Mónica Cristina Liberatori

Redes de Datos y sus Protocolos

Universidad Nacional de Mar del Plata

eudem

Redes de Datos y sus Protocolos

Mónica Cristina Liberatori

eudem

Liberatori, Mónica Cristina
Redes de datos y sus protocolos / Mónica Cristina Liberatori. - 1a ed . - Mar del Plata : EUDEM, 2018.
Libro digital, PDF

Archivo Digital: descarga
ISBN 978-987-4440-11-2

1. Ingeniería Informática. I. Título.
CDD 629.89

ISBN 978-987-4440-11-2

Este libro fue evaluado por el Dr. Jorge M. Finochietto

Fecha de edición: mayo 2018

Editorial de la Universidad Nacional de Mar del Plata
EUDEM / 3 de Febrero 2538 / Mar del Plata / Argentina

Libro
Universitario
Argentino

Dedico este libro a:

Mi esposo Marcelo,
Mis hijos Agustín, Matías y Santiago,
Mi madre Elvia,
Mis alumnos,
Mis profesores y compañeros de trabajo,
y a la memoria de mi Padre.
M. C. L.

INDICE

Prefacio

La comunicación de datos en redes y la interrelación entre las mismas es una de las tecnologías de mayor crecimiento en los últimos años. Una de las consecuencias de este crecimiento es la incorporación, en múltiples profesiones, de diversos cursos relacionados con este tema.

Esta obra pretende facilitar a los estudiantes afines a las carreras de ingeniería (electrónica, computación, comunicaciones e informática) la comprensión de diversos temas relacionados con la comunicación de datos en redes. Por este motivo, no sólo aborda principios básicos, de fundamental importancia para entender el diseño de la tecnología relacionada con el área, sino además provee una discusión detallada sobre los desarrollos más modernos, presentando ejemplos prácticos relacionados con estos temas y considerando los importantes desafíos subyacentes detrás de cada solución propuesta. En este sentido, el libro integra conceptos de comunicaciones y protocolos, haciendo hincapié en el análisis de diversas problemáticas y sus soluciones tecnológicas asociadas.

Para una mejor integración del conjunto de temas abordados, se dividió el texto en tres partes. La primera parte, de carácter general, presenta aspectos básicos conceptuales referidos a la transmisión de datos en redes, ya sea de área local como de área amplia. La segunda parte se orienta al estudio profundo de las redes de área local cableadas e inalámbricas, considerando pormenores tanto de su despliegue a nivel físico como de los protocolos que rigen la comunicación de estas redes de acceso. Por último, la tercera parte del texto aborda el tema de la Arquitectura TCP/IP, considerada en detalle a partir de su aspecto tradicional hasta las migraciones y modificaciones más recientemente conocidas de ambos protocolos.

La primera parte del libro consta de cuatro capítulos. El Capítulo I tiene carácter introductorio, ofreciendo una mirada a la historia de Internet, para luego desarrollar aspectos generales referidos a las redes de área amplia (WAN) y las redes de área local (LAN). En este capítulo se destaca la importancia de la estandarización de protocolos y los organismos reconocidos para ello, deteniéndose particularmente en la estructura de aquellos específicos para la estandarización de Internet. El Capítulo II aborda la temática de los modelos definidos para una comunicación de red y las arquitecturas desarrolladas en este mismo sentido. El Capítulo III pretende dejar una visión conceptual de la

tecnología que subyace detrás de las redes de área amplia, ya sean de conmutación de circuitos o de conmutación de paquetes. En este sentido, el capítulo presenta varias técnicas de multiplexado, cada una con sus propias características y entorno de aplicación, con ejemplos prácticos de cada tipo. También ofrece una descripción de los componentes principales de la estructura subyacente de ambos tipos de redes. Por último, el Capítulo IV introduce conceptos referidos a la transmisión de datos sobre medios guiados y no guiados. Se presentan los problemas más relevantes referidos a la transmisión de señales en un cable y la forma de compensarlos, definiéndose los parámetros que más frecuentemente aparecen en las especificaciones técnicas, junto con sus unidades y valores típicos. También se presentan las particularidades más relevantes del propio canal inalámbrico, para que se comprendan conceptualmente los desafíos enfrentados por estas redes. El capítulo describe los dos tipos de desvanecimiento presentes en entornos no guiados, desarrollando el modelo de canal de fading que domina la transmisión de datos en redes tipo WiFi. Finalmente, se mencionan los métodos apropiados para mitigar la degradación resultante del problema de desvanecimiento.

La segunda parte de la obra se desarrolla en cinco capítulos y comprende todos los aspectos más significativos de las redes LAN. El Capítulo V presenta el detalle de los protocolos que manejan la comunicación en redes LAN cableadas. Con este propósito, comienza explicando el estándar IEEE 802.3, tanto a nivel MAC como a nivel de capa física, sus alcances y limitaciones. El Capítulo VI desarrolla en orden cronológico de aparición las nuevas tecnologías que permitieron alcanzar velocidades muy superiores a las del estándar original IEEE 802.3. El Capítulo VII detalla las herramientas incorporadas a nivel de protocolo de acceso al medio para ayudar a superar los problemas de la comunicación, propios del canal inalámbrico. También se resaltan todos los aspectos administrativos que permiten que los elementos móviles funcionen en modo ahorro de potencia para no desgastar el tiempo de vida útil de sus baterías, uno de los principales desafíos de la movilidad. El Capítulo VIII desarrolla las técnicas ideadas para solucionar los problemas de transmisión en la capa física no guiada, desplegando una descripción detallada de cada una de las capas físicas IEEE 802.11 estandarizadas conocidas hasta hoy.

La tercera parte del libro desarrolla detalladamente los protocolos de la Arquitectura TCP/IP relacionados con el encaminamiento de datos a través de diferentes redes y el transporte confiable de la información. El Capítulo IX introduce los aspectos más importantes del Protocolo de Internet, conocido como IPv4. También se desarrollan los protocolos complementarios ARP, para comunicación en una red de difusión, e ICMP, para reporte de situaciones de error y provisión de mecanismos de testeo. El Capítulo X presenta la nueva versión del protocolo, IPv6, de incipiente despliegue actual, desarrollando con considerable detalle el nuevo esquema de direcciones propuesto, presentando a continuación los campos relevantes del encabezado. También se explica la configuración de dispositivos con este nuevo protocolo, detallando los mensajes relativos al protocolo ICMPv6, y el de Descubrimiento de Vecinos. El capítulo finaliza con una explicación sencilla de cada uno de los mecanismos más conocidos ideados para la transición de IPv4 a IPv6, por tratarse esta migración de uno de los

mayores desafíos con los que se enfrenta la Internet de nuestros días. El Capítulo XI comienza desarrollando el concepto de sockets y todas las llamadas al sistema necesarias para el establecimiento de una comunicación sobre la red. A continuación se presenta el protocolo UDP y, comparativamente, los aspectos más significativos del protocolo TCP. Dada su importancia conceptual, se presenta detalladamente toda la secuencia de sucesos relacionados con la apertura y cierre de una conexión TCP, finalizando el capítulo con un resumen sobre los mecanismos de ataque referidos a ciertas vulnerabilidades presentes en las fases mencionadas. El Capítulo XII presenta el protocolo TCP en acción, una vez que la conexión se ha establecido. Se desarrolla en detalle la estrategia de expiración (timeout) y retransmisión y el ajuste de los parámetros que la manejan. Se explica cómo afronta TCP situaciones de pérdida de segmentos, aparición de segmentos duplicados y casos de desorden en los segmentos recibidos, debiendo asegurar, en cualquier caso, una comunicación confiable. El Capítulo XIII explica ciertas reacciones que el protocolo TCP presenta frente a situaciones de congestión. En este capítulo, además se desarrollan los conceptos relacionados con las nuevas opciones del protocolo, incorporadas posteriormente para mejorar su comportamiento, adaptándolo a las tecnologías más modernas.

De este modo, a lo largo del texto se analizan los desafíos de la comunicación de datos en redes, incluyendo al mismo tiempo aspectos tecnológicos de implementación, detalles de estandarización por medio de protocolos y ejemplos prácticos para su mejor comprensión.

Agradecimientos

Me siento muy agradecida por todo el apoyo recibido durante la escritura de estas páginas. En particular, deseo expresar mi especial reconocimiento a mis colegas profesores, formadores de espíritus críticos, y a tantas generaciones de estudiantes, abiertos al desafío del conocimiento y siempre dispuestos a dar sugerencias para mejorar la calidad de las clases y los materiales de estudio. Agradezco las indicaciones sobre los errores, los consejos sobre diferentes visiones superadoras y las dudas, principales disparadoras de nuevas búsquedas del saber.

Parte I

Introducción

CAPÍTULO I

Introducción

La expansión continua de la comunicación de datos mediante redes en los últimos quince años, posiciona al tema en sí mismo como uno de los más importantes y de mayor generación de nuevas técnicas dentro del área de las comunicaciones.

No sólo este tipo de transmisión de datos ha tenido una expansión geográfica a nivel mundial, sino que también, creciendo casi a una velocidad comparable, se tradujo en la aparición de un enorme rango de aplicaciones para su utilización por un número cada vez mayor de usuarios. Esta evolución ha derivado en la aparición de nuevos desafíos tecnológicos, con respuestas de desarrollo ingeniosas, tanto en el terreno del hardware como del software.

Cabe mencionar, como uno de los motivos interesantes respecto de la explicación de este fenómeno, el uso cada vez más extendido de computadores personales y teléfonos celulares, con su propia variedad de métodos de acceso al medio sobre el que se establece la comunicación.

Existen diferentes formas de acceso a la red mundial conocida con el nombre de Internet. Algunas de ellas las reconocemos fácilmente desde el punto de vista de usuarios. Desde nuestros hogares en Argentina, lo más común es el acceso a través de la red de telefonía fija, cuando se contrata un servicio del tipo ADSL a alguna empresa de telefonía. Otra forma muy común es el acceso a través de la red de televisión por cable si el servicio se contrata a una compañía de cable. Últimamente, asistimos a la integración de la telefonía celular con este tipo masivo de accesos, en este caso ya sea por WiFi o por la red celular propiamente dicha.

Cada una de las formas mencionadas tiene como sostén su propia tecnología, siendo las mismas bastante diferentes entre sí. Sólo pensar en los diferentes medios por medio de los cuales se accede a Internet en cada caso, puede brindar una idea de las particularidades. Por ejemplo, el servicio ADSL es un acceso a través del par telefónico, en tanto que el servicio a través de la compañía de cable se realiza sobre un cable coaxial. En el caso de la telefonía celular, el acceso es por aire a través de las antenas de gran cobertura de las

distintas compañías, o a través de antenas WiFi, de baja cobertura e instalación privada.

Más allá de las diferencias, lo importante es que todas las formas proporcionan una conexión física a la red del proveedor de servicios de Internet. Los usuarios contratan una forma de acceso a Internet, mientras que el transporte de los datos intercambiados queda a cargo de los proveedores. Estos últimos conforman un escalón superior a ese servicio contratado, que se sostienen con su propia tecnología, muy diferente a la de acceso.

Tanto las redes, como los datos transportados por ellas, presentan su propia variedad. Una clasificación general nos permitiría distinguir datos entre texto, imágenes, audio y video, cada uno con sus características particulares. Los dos primeros son de naturaleza digital en su origen, en tanto que los últimos requieren de una conversión analógico-digital para una transmisión más eficiente. Se podrían destacar otras diferencias si consideramos la transmisión en tiempo real, con sus exigencias más estrictas. Por ejemplo, no es lo mismo enviar un mensaje de texto que mantener una comunicación por Skype.

En este sentido también observamos diferentes modos de transferencia. A lo largo de este texto se podrán encontrar ejemplos de modos de transmisión half duplex, donde la comunicación es de doble vía pero de a uno por vez, como es el caso de las redes de área local. En el modo full duplex en cambio, la transmisión y la recepción pueden realizarse de manera simultánea, como es el caso de la comunicación telefónica o de las redes locales conmutadas de alta velocidad.

Aparte de los modos mencionados, existen dos modos de transferencia muy usados en las redes de datos: broadcast y multicast. El primer caso es comúnmente usado en las redes locales como modo de aviso de algún mensaje desde uno de los equipos al resto de los que conforman la red. El segundo caso, se presenta para soporte de algunas aplicaciones particulares. Un ejemplo de este último tipo es el de los juegos en red, donde se debe conformar un grupo primero y la comunicación queda establecida entre los miembros de ese grupo solamente.

A lo largo de este capítulo se ofrecerá una mirada a la historia de Internet. Luego se abordarán los aspectos más generales referidos a los dos tipos de redes más importantes: las redes de área amplia y las de área local. Luego se definirá más formalmente el concepto de red y de protocolo de red, enumerando las funcionalidades comunes que poseen estos últimos. Por último, interesa destacar la importancia de la estandarización de protocolos y los organismos reconocidos mundialmente para ello.

1.1Breve Historia de Internet

El desarrollo de Internet ha significado una revolución en informática y comunicaciones. Se trata de una red para acceso a la información, que ofrece un mecanismo para la colaboración e interacción entre usuarios y máquinas, sin importar su ubicación geográfica.

Históricamente, surgió como resultado de una evolución tecnológica que comenzó con las primeras investigaciones sobre conmutación de paquetes.

El concepto de conmutación devino de las redes de telefonía ya instaladas. Simplificando, se lo puede relacionar con la necesidad de conectar dos puntos a través de una red, ayudándose mediante elementos intermedios, denominados conmutadores, que reciben datos por un canal de entrada y los re-envían por el canal de salida correspondiente, hacia otro elemento intermedio, y así sucesivamente hasta alcanzar el extremo final de la comunicación.

Por su parte, el concepto de paquete, surgió de la evolución de los elementos transportados en una red de telefonía. En principio, una red de telefonía es una red para transporte de señales de voz. Comúnmente, la señal analógica de voz para su transporte en la red de telefonía, se muestrea a razón de 8 $bits$ por muestra, a una frecuencia de muestreo de 8 KHz. De este modo, cada usuario de la red representa un canal de voz, que se puede interpretar como una señal digital de 64 $kbps$. Los conmutadores telefónicos son capaces de tomar varios canales de entrada de voz, y generar con ellos una señal de salida multiplexada, de mayor velocidad, para su transporte apropiado hacia otro elemento conmutador.

Inicialmente, sobre la red de telefonía fija, sólo se transportaban muestras de voz. Una muestra es información pura que es posible trasladar a través de la red si se acompaña de un mecanismo de sincronización adecuado. Se trata de las **redes de conmutación de circuitos**.

El concepto de paquete se aparta de este tipo de transporte, generando así otro tipo de redes. En este caso, a las muestras de información pura se agrega información adicional, llamada información de control, para mejorar la eficiencia de la comunicación. Por ejemplo, se podrían agregar bits para el control de errores, para identificación del destino final, para contemplar cuestiones de seguridad o para mejorar la calidad de la transmisión. En este sentido, los conmutadores de paquetes, a diferencia de los conmutadores originales para telefonía, deben ser capaces de revisar y comprender parte de esta información adicional, para poder tomar una decisión correcta referida al traslado de los datos. Se trata de las **redes de conmutación de paquetes**.

Leonard Kleinrock, profesor del Instituto de Tecnología de Massachusetts (MIT, Massachusetts Institute of Technology), fue quien publicó el primer documento sobre la teoría de conmutación de paquetes en julio de 1961, y el primer libro sobre el tema en 1964, explicando la factibilidad teórica de la comunicación por medio de redes de conmutación de paquetes.

Durante esos años, J.C.R. Licklider, director de la Agencia de Investigación de Proyectos Avanzados de Defensa (DARPA, Defense Advanced Research Projects Agency), también profesor del MIT, publicó el concepto de "*Red galáctica*", una red pensada como un conjunto de ordenadores interconectados globalmente, a través de los cuales todo el mundo podría acceder rápidamente a datos y programas desde cualquier sitio. En su esencia, el concepto de esta red era muy similar a la red de Internet de hoy en día.

El sucesor de Licklider en la dirección de DARPA, Lawrence G. Roberts, creó el plan para ARPANET en el año 1967, un nuevo proyecto que se basaba en el concepto de redes de paquetes. Debido al temprano desarrollo de Kleinrock sobre la teoría de conmutación de paquetes y a su trabajo en el análisis, diseño y

medición de este tipo de redes, su Centro de Investigaciones de la Universidad de California (UCLA, University of California, Los Angeles) fue seleccionado como el primer nodo de ARPANET. El proyecto de Doug Engelbart, del Instituto de Investigaciones de Standford (SRI, Standford Research Institute), fue elegido como segundo nodo, por su investigación sobre un sistema con características similares a lo que hoy conocemos como hipertexto. Para 1969, ya se habían añadido dos nodos más al proyecto original, uno en la Universidad de California en Santa Bárbara y otro en la Universidad de Utah.

Para poder conectar estos nodos, se debieron establecer normas comunes para la comunicación, que tuvieran el significado de un lenguaje universal, surgiendo así el concepto de protocolo. El protocolo utilizado por aquel entonces por las máquinas conectadas a ARPANET fue el Protocolo de Control de Red (NCP, Network Control Protocol). Con el paso del tiempo, NCP dio paso a un conjunto de protocolos más sofisticados, una arquitectura conocida como Protocolo de Control de Transmisión/Protocolo de Internet (TCP/IP, Transmission Control Protocol/Internet Protocol). Conviene destacar que la investigación sobre redes por aquel entonces, no sólo incorporaba trabajo sobre la red subyacente en cuanto a los protocolos de comunicación, sino también abarcaba aspectos relacionados con el uso de la red por parte de aplicaciones de usuarios, tradición que continúa hoy en día.

En octubre de 1972, se organizó una gran demostración pública del uso de ARPANET y también se introdujo la aplicación de mayor impacto inicial, motivado en la necesidad de los desarrolladores de ARPANET de contar con algún mecanismo sencillo de coordinación: el correo electrónico.

En ese mismo año, Robert Kahn, quien había desempeñando un importante papel en el diseño arquitectónico general de ARPANET, empezó a trabajar con la idea de redes de arquitectura abierta. Este concepto surgió debido a fallas registradas en el protocolo original de comunicación NCP de ARPANET. La principal desventaja del mismo era que no tenía capacidad de comunicación con otros componentes que no pertenecieran a la propia red ARPANET, ni podía ofrecer mayor fiabilidad entre extremos de la comunicación que la que ofrecía la red subyacente. Kahn relacionó que, para lograr la integración, sería necesario trabajar con los detalles de implementación de los sistemas operativos de los componentes. Por este motivo, comenzó a trabajar en conjunto con Vint Cerf, un profesor de Stanford, quien había estado involucrado de lleno en el diseño y desarrollo original de NCP y tenía conocimiento sobre las interfaces de los sistemas operativos existentes.

De este modo, armados con el enfoque arquitectónico de Kahn para la parte de comunicaciones, y con la experiencia de Cerf en NCP y sistemas operativos, crearon lo que se convertiría luego en la arquitectura TCP/IP.

A partir de 1977, la naturaleza descentralizada de ARPANET y la disponibilidad gratuita de los programas basados en TCP/IP fue lo que permitió que otro tipo de redes no vinculadas a ARPANET, comenzaran a conectarse, mezclándose una comunidad militar operativa con una comunidad científica. Aparecieron por entonces las primeras referencias a Internet como "*una serie de redes conectadas entre sí, específicamente aquellas que utilizan el protocolo*

TCP/IP". Internet es la abreviatura de *Interconnected Networks*, en español Redes Interconectadas o red de redes.

En 1983, el segmento militar de ARPANET se separó, conformando su propia red, denominada MILNET. Así, ya sin fines militares, ARPANET abrió sus puertas a universidades, empresas y todo tipo de instituciones. Desde ese momento, ARPANET y todas sus redes asociadas comenzaron a ser conocidas como Internet.

En 1984, la Fundación Nacional para la Ciencia (NSF, National Science Foundation) inició NSFNET, que adoptó también como arquitectura de comunicación a TCP/IP, sirviendo para conectar varias redes aisladas entre sí a la red ARPANET. También aparecieron nuevas redes, tales como USENET y BitNet. El crecimiento exponencial que experimentó NSFNET, así como el incremento continuo de su capacidad de transmisión de datos, determinó que la mayoría de los miembros de ARPANET terminaran conectándose a esta nueva red y, para 1989, ARPANET se declaró disuelta.

Por otra parte, desde 1973, Bob Metcalfe trabajaba en el proyecto *Ethernet* que se estaba desarrollando en el Centro de Investigación de Palo Alto (PARC, Palo Alto Research Center) de la empresa Xerox. Esta nueva forma de comunicación en redes pequeñas, sumado al objetivo de interconexión a la gran red, significó un verdadero desafío en cuanto al número creciente de máquinas a conectar. Entre los problemas a resolver, se encontraba la adaptación de un esquema de identificación de dispositivos que ya estaba presente en el protocolo IP y que consistía en un número de $32\ bits$ denominado dirección IP. A su vez, como para el ser humano es más fácil recordar nombres en lugar de números tan extensos, se puso en evidencia la necesidad de asociar una estructura de nombres a estas direcciones. Así surgió uno de los servicios más conocidos de Internet, el Sistema de Nombres de Dominio (DNS, Domain Name System).

Por otra parte, como NSFNET no sólo conectaba ordenadores en Estados Unidos, sino también en otros países, se empezó a trabajar sobre el concepto de dominio geográfico para las redes fuera de los Estados Unidos y en el concepto de una separación por nombre según la finalidad de la red anexada. Así aparecieron diversos dominios que hoy resultan muy familiares, tales como *gov*, por gobierno, *mil* por instituciones militares, *edu* por instituciones educativas, *com* para emprendimientos comerciales, *org* para instituciones sin fines de lucro y *net* para los ordenadores que servían de enlace entre las diferentes subredes. También surgieron los dominios identificadores de países, tales como *ar* para Argentina y *es* para dominios de España.

En 1989, Tim Berners Lee, investigador del Centro Europeo de Investigación Nuclear (CERN, Centre Européen de Recherche Nucléaire), inventó un sistema que facilitaba la interacción con Internet, para compartir y hallar datos con posibilidades de hipertexto y multimedia. Había nacido la *World Wide Web*, sobre la base del Protocolo de Transferencia de Hipertexto (HTTP, HyperText Transfer Protocol), el Lenguaje de Marcado de Hipertexto (HTML, Hyper Text Markup Language) y el concepto de Localizador Uniforme de Recursos (URL, Uniform Resource Locator).

En 1998, Internet contaba con alrededor de 50 millones de usuarios y 25 millones de servidores. Para ese año, compañías como Yahoo, eBay o Amazon tenían apenas 3 o 4 años de existencia, mientras Google recién estaba naciendo.

Hacia el año 2000, el acceso a la gran red empezó a abandonar el viejo sistema de discado telefónico, con tecnología de modem telefónico para pocas decenas de $kbps$, comenzando a ser reemplazado por accesos de banda ancha, posibles gracias a la aparición de nuevas tecnologías. Esta evolución se tradujo en velocidades en el orden de decenas de $Mbps$.

Para fines de la primera década del siglo XXI, ya se contaba con una base instalada de alrededor de 542 millones de servidores y 1.3 billones de usuarios. La telefonía móvil contabilizaba por entonces un número estimado de 3 billones de aparatos en uso. Alrededor del 15 por ciento de los mismos podía acceder a Internet. También para esa época, se estimaba alrededor de 1 billón el número de computadores personales en uso, la mayoría de ellos con acceso a Internet.

La progresión sigue en aumento y la variedad de dispositivos conectados es cada vez mayor. Sólo el tiempo nos dirá la manera en que la gran red se irá acomodando a las distintas expectativas.

1.2 Clasificación de las Redes

Según su extensión geográfica, los dispositivos que las conforman y las tecnologías específicas, se podría clasificar las redes de datos en dos grandes grupos: las Redes de Área Local (LAN, Local Area Networks) y las Redes de Área Amplia (WAN, Wide Area Networks).

1.2.1 Redes de Área Amplia WAN

Las redes WAN se caracterizan por su gran extensión geográfica. Son redes compuestas por dispositivos especiales, denominados nodos conmutadores o dispositivos de encaminamiento, en inglés *routers*. La finalidad principal de estas redes es el transporte de los datos, por lo que su funcionalidad primordial se relaciona con el área específica de enrutamiento, que se ofrece como servicio de conmutación. También ofrecen servicios de conexión o acceso.

Se trata de redes manejadas por los Proveedores de Servicio de Internet (ISP, Internet Service Providers). Un ISP es una compañía que ofrece acceso a Internet. Generalmente, la conexión del usuario con el ISP tiene lugar a través de un acceso telefónico, una conexión de banda ancha por ADSL, cable o algún servicio inalámbrico. Muchos ISP ofrecen servicios adicionales al propio acceso, tales como cuentas de correo electrónico o espacio para crear un sitio web propio. En Argentina encontramos empresas como Speedy (Telefónica de Argentina), Fibertel (Grupo Clarín) y Arnet (Telecom), entre otras.

El servicio de conexión ofrecido por el ISP, puede presentarse de dos maneras diferentes: orientado a la conexión o sin conexión.

El **servicio orientado a la conexión** se caracteriza por brindarse de manera explícita, mediante una fase de intercambio inicial en la que se establece

una sesión, antes de que comience la trasferencia de datos. Este tipo de servicio garantiza la reserva de recursos y la llegada de la información en orden. Se trata de un servicio confiable. Un ejemplo similar al de este tipo de servicio es el ofrecido por una empresa de telefonía para poder mantener una comunicación telefónica.

Por su parte, los **servicios sin conexión** no pasan por una fase inicial de conexión. Simplemente, una vez contratado el servicio, se puede comenzar a transmitir y recibir datos. Se trata de un servicio más rápido y simple, pero sin garantías ni confiabilidad. Un ejemplo de este servicio es el de la conexión a Internet desde nuestros hogares.

En redes WAN, el encaminamiento o servicio de conmutación, consiste en el proceso de seleccionar caminos o rutas en una red, a través de los cuales se transporta el tráfico para llegar al destino final. Varias clases de redes son capaces de llevar adelante este transporte. Entre las más conocidas, se pueden mencionar las redes de conmutación de circuitos y las redes de conmutación de paquetes.

Las **redes de conmutación de circuitos** se conforman mediante un sistema interconectado de centrales de conmutación telefónicas. Se trata de redes ideadas originalmente para la transmisión de voz. Tal como se presenta en la Fig. 1.1, su utilización por parte de cada usuario se traduce en el cumplimiento de tres fases:

- **Establecimiento**: en esta fase inicial se debe encontrar un camino hacia el destino final y reservar recursos para la comunicación. Este camino, también denominado circuito, garantiza recursos de ancho de banda y permanece establecido durante toda la comunicación. Se denomina circuito porque la red se comporta como si los nodos estuvieran conectados físicamente. El retardo de las señales es constante durante toda la comunicación, permaneciendo el canal reservado, aunque permanezca ocioso, hasta el momento de su liberación. De esta manera se asegura el servicio frente a potenciales usuarios competidores.

- **Intercambio de datos**: es en modo full duplex, sobre el camino dedicado a la comunicación.

- **Cierre**: es la fase final, cuyo objetivo es la liberación de los recursos para que queden disponibles a otros posibles usuarios.

La utilización de este tipo de redes adolece de un retardo inicial, debido al establecimiento de la conexión. Esta filosofía de reserva *a priori* de los recursos se puede interpretar como una manera de manejar las situaciones de congestión de la red: si no se aseguran los recursos para la comunicación, la red rechaza el intento. Además, en esta fase de establecimiento, para poder encontrar el camino hacia el destino, debe existir alguna manera de identificarlo, en este caso el número de teléfono que corresponda.

Para poder transmitir las muestras de voz, la red de conmutación de circuitos utiliza un sistema de multiplexado sincrónico, que impone cierta velocidad de transferencia y un retardo constante. La ventaja de una transferencia

continua se debe a la ausencia de información de control, asociada específicamente a la transmisión de paquetes. La desventaja es que la conexión no puede ser compartida por otros, pudiendo hasta denegarse la comunicación a los demás usuarios en situaciones de sobrecarga.

Hasta hace pocos años atrás la red se usaba únicamente para ofrecer un servicio de telefonía. Cuando comenzó a usarse para transportar datos, el valor máximo de la velocidad de transferencia era bastante limitado, debido a los filtros que originalmente se encontraban en la central de conmutación local, para limitar el ancho de banda de la señal analógica de voz. Más tarde, para mejorar la transmisión de datos, dichos filtros fueron retirados, quedando la velocidad limitada al propio par de telefonía y a la distancia desde el domicilio del abonado a la central local. Esto explica las diferencias de velocidades alcanzadas por tecnologías tipo ADSL frente a las de modem tradicional.

Por otra parte, la velocidad de transferencia constante, característica de este tipo de redes, no permite una transmisión eficiente en el caso de tráfico con naturaleza de ráfagas, distintivo de la transmisión de datos.

Señalización de una llamada de voz

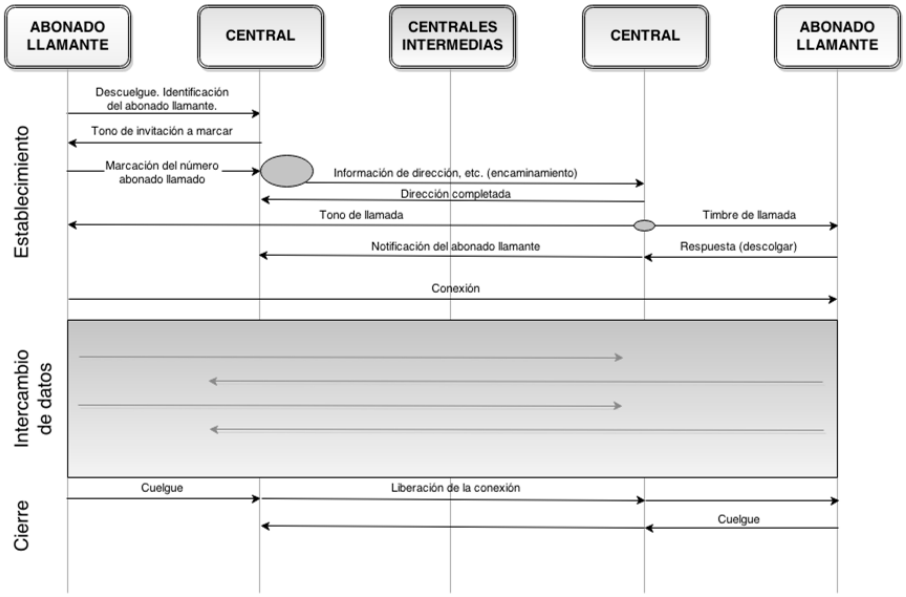


Figura 1.1 - Fases de una llamada. Red de Conmutación de Circuitos.

En contraste con las redes de conmutación de circuitos, surgieron las **redes de conmutación de paquetes**, mucho mejor adaptadas a la transmisión de ráfagas de datos. Este tipo de redes fueron diseñadas especialmente para el transporte de paquetes, permitiendo compartir el ancho de banda disponible entre múltiples sesiones de comunicación. El método de transmisión por conmutación de paquetes se conoce en inglés como *packet switching*. Se trata de la tecnología

de red fundamental que existe en Internet y en la mayoría de las redes de área local.

Un paquete se caracteriza por el agregado de información adicional: a la información propiamente dicha, representada por los datos, se agregan bits de control que conforman un encabezado, en inglés *header*. El conjunto de datos y encabezado se conoce como paquete. Una secuencia de paquetes transportada sobre este tipo de redes, tiene velocidad variable, siendo esta característica la más apropiada para la naturaleza de ráfagas de la transmisión de datos.

Tal como se aprecia en la Fig. 1.2, las redes de conmutación de paquetes están conformadas por nodos, denominados dispositivos de encaminamiento o *routers*. Se trata de elementos especiales, con funcionalidad específica, que se encuentran conectados entre sí por medio de enlaces, realizando tareas de cooperación para el traslado de los paquetes.

En cada nodo de la red, cada paquete ingresa por algún enlace de entrada, colocándose en una memoria temporal, a la espera de ser revisado. Se dice que el paquete se encuentra en una cola de entrada del dispositivo conmutador.

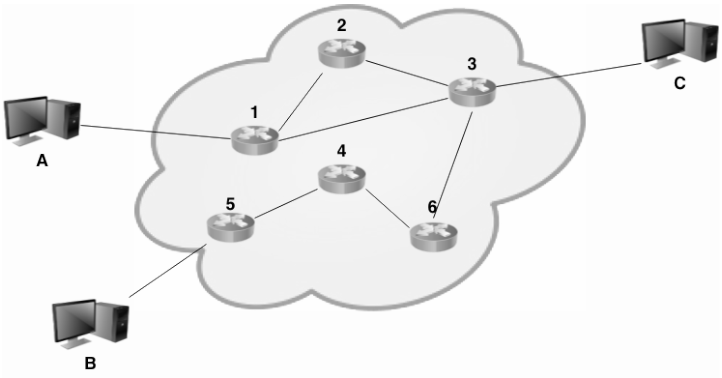


Figura 1.2 - Red de Conmutación de Paquetes.

Cuando el nodo revisa las colas de entrada, interpreta cierta información del encabezado de cada paquete, sirviéndole esta consulta para determinar sobre cuál enlace de salida debe re-enviar el paquete. La decisión de re-envío se realiza consultando una tabla almacenada en memoria, que se conoce como Tabla de Enrutamiento. Esta estrategia de almacenamiento y re-envío, conocida en inglés como de *store&forward*, genera un retardo variable por paquete, resultando su eficiencia dependiente de la carga de tráfico en la red.

A su vez, entre las redes de conmutación de paquetes, existen dos modos de ofrecer el servicio de transporte: el modo sin conexión, también conocido como conmutación de datagramas, y el modo orientado a la conexión, o conmutación de circuitos virtuales.

En el **modo sin conexión**, cada paquete es encaminado de manera independiente, según la información de direccionamiento de destino que se cargue en su propio encabezado. Esta independencia de decisión de enrutamiento por paquete puede resultar en la elección de diferentes rutas para cada paquete

perteneciente a una misma comunicación, llegando la información fuera de orden al destino. Se presenta un ejemplo en la Fig. 1.3, donde se puede apreciar que los paquetes que se transmiten desde el dispositivo A hacia el dispositivo B podrían encaminarse de manera independiente, por diferente camino. Lo mismo ocurre con las respuestas desde B hacia A.

Un ejemplo de este modo de transferencia de paquetes es propia de la red de Internet, donde se transportan datos utilizando el protocolo **IP**. Su principal ventaja es la capacidad de adaptación a situaciones de congestión: cuando un nodo se satura se generan pérdidas de paquetes en sus colas, pero la red puede adaptarse para modificar las rutas, de tal manera que los paquetes se dirijan a otros nodos, no saturados.

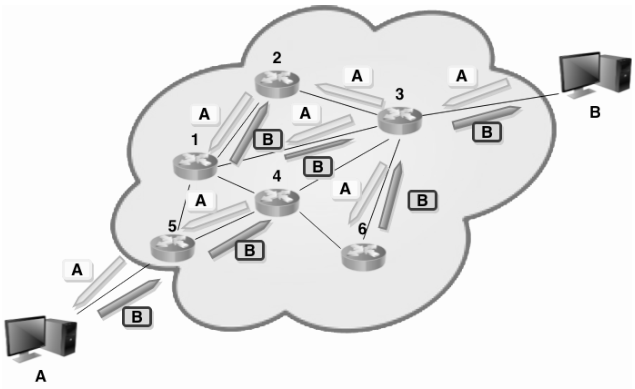


Figura 1.3 - Conmutación de paquetes en modo sin conexión.

En contraposición, en la Fig. 1.4 se presenta el **modo orientado a la conexión.** En este modo, la red inicia una conexión y pre-asigna recursos en cada nodo involucrado en el camino entre transmisor y receptor. Este camino, asignado en el inicio, se conoce como circuito virtual, por su semejanza a los circuitos de la red de conmutación de circuitos. Todos los paquetes de una misma comunicación incluyen una identificación de circuito virtual en su encabezado. La identificación será revisada en cada nodo del camino, con el propósito de que todos los paquetes de una misma conexión sigan la misma ruta. De este modo se conserva el orden de llegada a destino.

Se trata de redes confiables, pero presenta una desventaja, ya que si un nodo se satura y cae, quedando fuera de servicio, todas las conexiones que pasan por éste también caen y deberán restablecerse. Un ejemplo de estas redes es la vieja red de cajeros automáticos, que funcionaba bajo el estándar X.25, antes de su migración a TCP/IP.

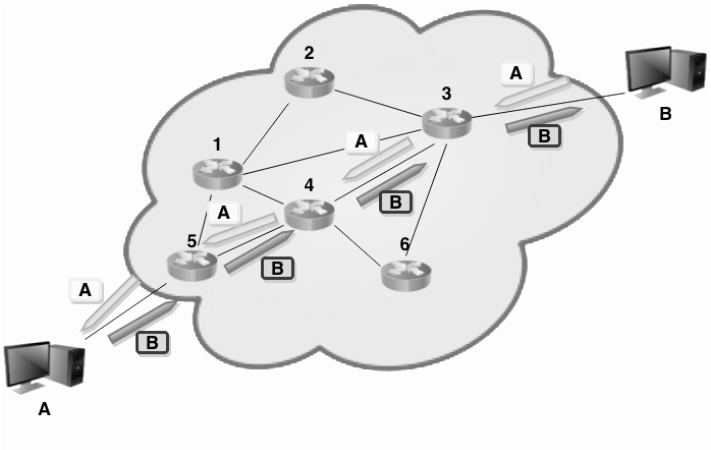


Figura 1.4 - Conmutación de paquetes en modo orientado a la conexión.

1.2.2 Redes de Área Local LAN

En contraste con las redes WAN, las redes LAN ocupan áreas geográficas pequeñas, por ejemplo un edificio o conjunto de edificios. Generalmente se trata de redes cuyo tendido obedece a la necesidad de compartir recursos, tales como impresoras, *scanners* y dispositivos de almacenamiento. Su mantenimiento queda a cargo de administradores, ya sean propietarios de las mismas o contratados para tal efecto.

Las redes LAN de mayor despliegue comercial son las conocidas con el nombre genérico de *Ethernet*. Su tendido se realiza sobre un cable del tipo par trenzado. Se trata de redes de alta velocidad, pudiendo llegar en la actualidad al orden de los $Gbps$. Se caracterizan por interconectar dispositivos tales como computadores personales, repetidores o *hubs*, puentes y conmutadores o *switches*, tal como se presenta en la Fig. 1.5. Generalmente un *router* es el dispositivo de salida de una red LAN a una red WAN.

Un *hub* es un elemento repetidor que ocasiona que la red se comporte como un *bus*, conociéndose con este nombre el caso de las redes LAN cuyos dispositivos comparten un medio, debiéndose establecer algún método de control para el acceso. El *hub* emula este comportamiento, repitiendo sobre todas sus bocas de salida, la señal que recibe por una boca de entrada.

Un *switch* es un dispositivo de mayor inteligencia que el *hub*, repitiendo mensajes entre sus puertos de acuerdo a cierto esquema de direccionamiento especial, propio de las redes LAN.

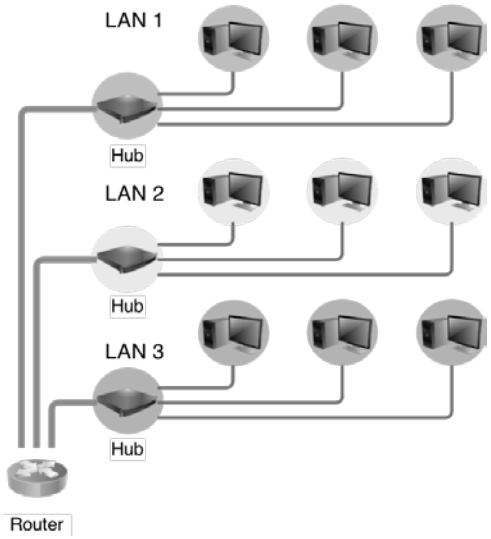


Figura 1.5 - Red LAN cableada.

La Fig. 1.6 presenta un ejemplo de red inalámbrica que cumple con el certificado de Fidelidad Inalámbrica (WiFi, Wireless Fidelity). Es un tipo de red LAN cuyo despliegue ha aumentado de manera sorprendente en los últimos años. Se trata de redes que conectan dispositivos inalámbricos y, en su modo más popular de instalación, poseen un nodo muy especial, conocido como Punto de Acceso (AP, Access Point), a través del cual pasan todas las comunicaciones.

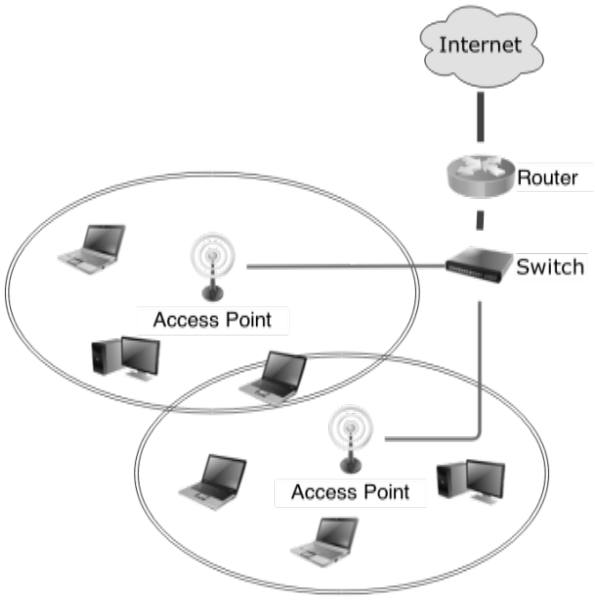


Figura 1.6 - Dos redes LAN WiFi conectadas a través de una LAN cableada.

En el caso de la figura, se puede observar un *switch* conectando ambos AP a través de un soporte cableado, disposición típica de muchas redes de este tipo. Es interesante observar en este ejemplo la disposición del AP como dispositivo de intermediación entre dos tipos de redes LAN. En este caso se dice que el AP realiza la función de un puente, también conocido como *bridge*.

1.2.3 Topologías de Red

La topología de red define su estructura física, o sea la manera en que se disponen los cables o enlaces que interconectan sus diversos elementos.

En general, encontramos las topologías presentadas en la Fig. 1.7, que también pueden servir para otro tipo de clasificación de las redes de datos:

- **Bus**: una de las topologías más sencillas que utiliza un único cable al que se conectan todos los componentes directamente. El cable debe terminarse apropiadamente en ambos extremos para evitar desadaptaciones. Todos los dispositivos comparten el mismo canal, por lo que debe existir una forma apropiada de ingreso al medio, quedando limitada tanto la cantidad de dispositivos como la longitud física de la red. La rotura del cable deja fuera de servicio el sistema.
- Ejemplo: LAN de cable coaxial.

- **Anillo**: conecta un elemento con el siguiente y el último con el primero. En este tipo de red la comunicación depende del paso de un paquete especial, denominado testigo o *token*, que se utiliza para ordenar la comunicación y permitir un acceso equitativo a todos los componentes. Si uno de los componentes falla o uno de los enlaces cae, la red queda fuera de servicio.
- Ejemplo: redes de fibra óptica como columna vertebral o *backbone* de red WAN.

- **Estrella**: conecta todos los cables con un punto central de concentración, por el que pasan todas las comunicaciones. Tiene como ventaja que, si un componente se desconecta o se rompe el cable que lo comunica, sólo ese equipo quedará fuera de la red. Su desventaja es que, si falla el nodo central, cae la red completa.
- Ejemplo: redes LAN tipo *Ethernet* con un conmutador tipo *switch* o un concentrador *hub* como elemento central.

- **Malla**: cada nodo se conecta con todos los demás, de tal manera que es posible llevar los mensajes de un nodo a otro por diferentes caminos. Al estar completamente conectada, se convierte en una red muy confiable en cuanto a una posible interrupción en las comunicaciones. Si la red tipo malla fuera cableada, una desventaja sería el costo, dada la cantidad de cable necesario para su instalación.

- Ejemplo: una red para control de una planta nuclear.

- **Árbol**: se trata de una topología centralizada, desarrollada a partir de un nodo raíz, a partir del cual se van desplegando los demás componentes como ramas. Los elementos de la red se ordenan en una estructura jerárquica, en donde se destaca un elemento predominante o raíz. El resto de los elementos comparte una relación tipo padre-hijo. El encaminamiento de los mensajes de este tipo de redes debe realizarse de tal manera de evitar lazos en la comunicación. Si falla un elemento podrían presentarse complicaciones, quedando parte de la estructura aislada, pero si falla la raíz, la propia red quedaría dividida en dos partes que no podrían comunicarse entre sí.
- Ejemplo: redes de sensores inalámbricos.

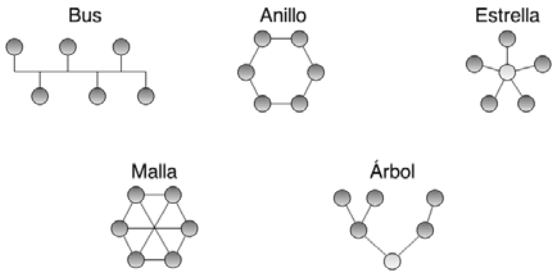


Figura 1.7 - Topologías de red.

1.3 Redes y Protocolos

Habiendo realizado una recorrida por diferentes clases de redes y topologías, conviene realizar una definición formal de red de datos y considerar algunos aspectos relacionados con su correcto funcionamiento.

Se define como **red de datos** a un grupo de dos o más elementos inteligentes que son capaces de comunicarse entre sí a través de algún medio e intercambiar datos de manera cooperativa.

Las redes conectan computadoras y los usuarios que las utilizan. Por ejemplo, en entornos de trabajo, los usuarios comparten recursos de redes LAN y pueden conectarse con otros usuarios, por ejemplo en otra red LAN, a través del acceso a una red WAN. Aparte de compartir datos fácil y rápidamente, pueden compartir dispositivos, tales como impresoras, y aprovechar más mecanismos de comunicación.

Al instalarse una red, primero debe ser planificada desde el punto de vista físico, atendiendo a las necesidades presentes y futuras que deba satisfacer. Una vez instalada físicamente, deberá configurarse para que funcione correctamente. Ya configurada correctamente, interesará que sea monitoreada para poder

anticiparse a posibles problemas de funcionamiento. Todo este trabajo se relaciona con la gestión de la red y es tarea de su administrador.

El uso eficiente de los recursos es una meta importante en cualquier clase de red y, en términos generales, se relaciona con las técnicas seleccionadas para compartir los mismos y los mecanismos utilizados para afrontar situaciones de congestión. De existir fallas, es importante considerar mecanismos de recuperación, sobre todo para evitar posibles pérdidas irreparables. También, dependiendo del tipo de información almacenada o transportada sobre la red, puede ser importante considerar herramientas de seguridad para la protección de los datos.

Por su parte, todo elemento conectado a la red debe contar con una interfaz de acceso apropiada, por ejemplo placas de red, con una antena o conector de red, en el caso de redes LAN.

A su vez, para que los distintos componentes de una red puedan dialogar entre sí de manera eficiente, debe existir una manera de identificarlos, es decir un esquema de direccionamiento apropiado al alcance de la comunicación.

Si se pretende una comunicación fiable, se deberán utilizar técnicas para control de errores. Si los componentes son de diferente capacidad operativa, probablemente utilicen técnicas de control de flujo para acomodar las diferencias en velocidad y/o capacidad de memoria.

Toda esta gestión del intercambio exige reglas de comunicación que deben ser respetadas por todas las partes.

Un **protocolo** define un conjunto de reglas, algoritmos, mensajes y otros mecanismos que habilitan a los elementos de una red a comunicarse de manera eficiente. Detrás de la definición de protocolo, yace la definición de un lenguaje común de entendimiento y la aceptación de un mismo conjunto de parámetros como convención. Por lo tanto, la definición de un protocolo exige el establecimiento de un formato para intercambio de mensajes y la precisión de las reglas que regirán ese intercambio. La elección del mismo debe ser previa a la comunicación y conocida por todas las partes involucradas en la misma.

Se pueden mencionar tres aspectos en la definición de un protocolo: sintáctico, semántico y de sincronismo de la comunicación. La especificación formal de estos tres aspectos es independiente de la implementación, que puede ser en *hardware* o *software*. El aspecto sintáctico se refiere a la especificación de formatos para los mensajes. La semántica se relaciona con la funcionalidad de control para la cual se ha diseñado el protocolo. Por su parte, el sincronismo define la sintonía de velocidades y secuencias particularmente utilizadas en la comunicación.

Para alcanzar un consenso general, un protocolo debe tener una especificación técnica con calidad de estándar. Los estándares son de conocimiento público, se los denomina protocolos abiertos para diferenciarlos de aquellos que no son públicos, conocidos como protocolos propietarios. Un ejemplo de los primeros es el protocolo de red IP. Los protocolos propietarios, en cambio, son protocolos con restricciones de uso, reglamentadas por patentes, y reforzadas por el mantenimiento de cláusulas secretas en cuanto a su implementación. Por ejemplo, un protocolo propietario es el que rige la comunicación por *Skype*.

Un protocolo de red es aquel específicamente diseñado para este tipo de comunicaciones. Su implementación consiste en un módulo de software con interfaces apropiadas para poder comunicarse con un entorno especial implementado en el sistema operativo de la máquina. Es decir que el sistema operativo debe poseer capacidad para comunicación sobre redes.

En general, los sistemas no utilizan un único protocolo, ya que se suele dividir el problema de la comunicación en módulos, para facilitar las tareas. Cada módulo puede tener asociado uno o más protocolos en un entorno de cooperación. Se suele denominar a este conjunto familia de protocolos o conjunto de protocolos. Una de las arquitecturas más conocido es la de TCP/IP.

1.4 Funcionalidad Asociada a Protocolos

Dado que un protocolo debe especificar las reglas que regulan la transmisión, deberá desarrollar varias de las siguientes funcionalidades, aunque no necesariamente todas:

- **Definición del formato de mensajes para el intercambio**: la existencia de un protocolo implica que los mensajes se trasladan de manera encapsulada, concepto que se asocia con la definición de paquete. El protocolo debe definir información adicional a los datos, denominada Información de Control de Protocolo (PCI, Protocol Control Information) o encabezado. El encabezado se asocia unívocamente a un protocolo, consistiendo en una serie de bits divididos en campos, cada uno con un significado particular asignado en su definición. Por otra parte, se denomina Unidad de Datos de Servicio (SDU, Service Data Unit) a la porción de información del mensaje. La SDU más el encabezado definido por el protocolo se denomina Unidad de Datos de Protocolo (PDU, Protocol Data Unit). El hecho de agregar un encabezado a la porción de datos se denomina encapsulado. En la Fig. 1.8 se presenta este concepto.

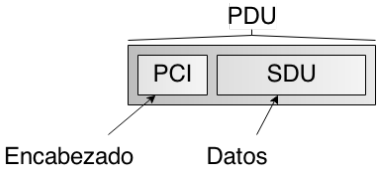


Figura 1.8 - Encapsulado.

- **Direccionamiento**: para poder realizar una comunicación en red se precisan identificadores, también llamados direcciones, tanto para la fuente de la información como para el destino de la misma. El examen de un campo de direcciones en el encabezado de un paquete permite al

elemento que lo procesa, determinar si va dirigido a éste o a otro dispositivo. Si el protocolo posee esta funcionalidad, muy probablemente defina además direcciones especiales. En este sentido, muchos protocolos precisan la dirección correspondiente a todos los bits en "1" como la dirección destino que pertenece a todas las estaciones de una red o dirección de *broadcast*. Esta dirección es de gran utilidad en las redes LAN para avisos de índole operativa, dirigidos a todos los integrantes de la red.

También existen direcciones en distintos niveles y con diferente alcance o significado.

A nivel de de comunicación en una red local, se usan las direcciones de fábrica de las placas de red o direcciones del nivel de Control de Acceso al Medio (MAC, Medium Access Control). Por ejemplo $00:13:49:00:01:02$ es una dirección MAC de $48\ bits$.

En cambio, a nivel de red, para que sea posible identificar cada máquina en Internet, se usan direcciones IP del tipo $170.210.36.4$, de $32\ bits$, con significado global. A su vez, dentro de un mismo dispositivo, para poder identificar las diferentes aplicaciones que simultáneamente se están comunicando, se utilizan números de puertos. Por ejemplo los servidores web se identifican con los puertos 80 y 8080, de $16\ bits$.

- **Control de errores**: muchos protocolos suelen agregar campos que resultan de la aplicación de algún código para detectar o corregir errores. Uno de los más utilizados es el código que resulta en la generación de bits de Chequeo de Redundancia Cíclica (CRC, Cyclic Redundancy Check). Con este campo adicional se puede trabajar en la detección de errores de transmisión, generando retransmisiones en caso de error.

  Algunos protocolos trabajan en la modalidad confiable, donde cada mensaje recibido sin errores es reconocido mediante el envío de otro mensaje especial a la fuente, conocido como de mensaje de Reconocimiento (ACK, Acknowledgment), distinguible por los valores de los bits de ciertos campos del encabezado. Bajo esta forma de transmisión, al enviar un mensaje y no recibir un ACK dentro de un tiempo determinado, conocido como tiempo de expiración o *timeout*, el transmisor supone que el mensaje no se recibió bien y procede a la retransmisión. Esta estrategia se conoce como de *timeout* y retransmisión.

- **Control de Acceso al Medio**: algunos protocolos se definen para generar un mecanismo de equilibrio de acceso en redes donde el medio es compartido. Algunas topologías que implican la existencia de un medio compartido son las de las redes tipo *bus*, estrella con elemento central tipo *hub*, redes inalámbricas o enlaces punto a punto de comunicación *half duplex*. En estos casos, el control de acceso puede ayudar a detectar o evitar las colisiones entre paquetes transmitidos en simultáneo por más de un dispositivo. Otras formas de ordenar el acceso podrían generarse a

través de un esquema de prioridades o por asignación de canales a cada usuario.

- **Control de Flujo**: es un mecanismo necesario para acomodar interlocutores con diferentes capacidades de procesamiento en cuanto a velocidad y memoria. Para ofrecer este control, muchos protocolos utilizan los propios mensajes de ACK para regular el flujo de la comunicación.
  Uno de los mecanismos más utilizados es el de Parada & Espera (*Stop & Wait*), que se usa en redes WiFi, condicionando la transmisión del cada mensaje a la llegada de un ACK para el mensaje previo.
  Por su parte, TCP usa un control de flujo más complejo, conocido como mecanismo de ventana deslizante, que permite transmitir y mantener en espera más de un mensaje por vez.

- **Control de Orden**: es aplicable en aquellas redes de tipo datagrama, donde los mensajes pueden llegar desordenados, debido a los retardos propios de la red o las diferencias de rutas tomadas por los paquetes en su camino hacia el destino. En respuesta a este problema, muchos protocolos agregan un campo en el encabezado, donde se coloca un número de secuencia que identifica al mensaje. Se debe tener en cuenta que se tratará de un campo con un número de bits limitado, por ejemplo $n\ bits$, razón por la cual sólo permitirá distinguir hasta 2^n mensajes diferentes que se encuentren en tránsito simultáneamente.
  El control de orden también permite distinguir copias de un mismo mensaje, ya que en los casos en los que se usan mecanismos de retransmisión, la pérdida de un ACK podría provocar un retransmisión innecesaria, generando la llegada de un mensaje duplicado, distinguible por la repetición de un número de secuencia.

- **Fragmentación**: en Internet existen redes capaces de transportar mensajes de distintos tamaños. Por ejemplo, una red LAN cableada tipo *Ethernet* puede transportar mensajes de hasta $1500\ bytes$ de datos, en tanto que una red de Modo de Transferencia Asincrónico (ATM, Asynchronous Transfer Mode) transporta mensajes, llamados celdas, de hasta $53\ bytes$. Se denomina Unidad de Transferencia Máxima (MTU, Maximum Transfer Unit) al tamaño máximo de mensaje que una red puede manejar. Teniendo en cuenta que la información debe atravesar una gran variedad de redes, con distintos valores de MTU, algunos protocolos incorporan mecanismos de fragmentación que deben ser complementados con mecanismos de re-ensamble para la recuperación de los datos fragmentados. Con este propósito, se deben adicionar campos especiales en los encabezados. Es de destacar que el mecanismo de fragmentación genera una disminución en el tamaño de la carga de datos, empeorando la relación de incidencia de la longitud del encabezado respecto de la del propio campo de datos.

- **Control de Conexión**: este tipo de control suele encontrarse presente en esquemas del tipo orientado a la conexión, donde se precisan mensajes específicos para el inicio y el cierre de la conexión. Con el propósito de distinguir estos mensajes de control, se incorporan bits especiales en el encabezado.

- **Multiplexado**: en comunicaciones se denomina multiplexor al dispositivo capaz de combinar varios canales en uno solo. Un de-multiplexor realiza la acción contraria, separando a partir un único mensaje, la información correspondiente a cada canal. En temas relacionados con protocolos, el concepto se refiere a la capacidad que posee un protocolo de marcar en su encabezado a cuál otro protocolo debe entregarse el mensaje cuando por encima de éste exista más de una posibilidad.

- **Encaminamiento**: en el caso de la comunicación entre sistemas que no se encuentren directamente conectados, se precisarán dispositivos intermedios a lo largo de la ruta al destino, cuya función es la generación de las acciones apropiadas para el re-envío de los paquetes hacia la dirección apropiada. Para poder cumplir la funcionalidad de encaminar correctamente, estos elementos deben conocer algo de la topología circundante, representada por otros nodos que los puedan ayudar a cumplir con la entrega. Las Tablas de Enrutamiento almacenan esta clase de información, construyéndose a partir del intercambio de mensajes entre nodos. Estos mensajes son generados por protocolos especiales, denominados protocolos de enrutamiento.

1.5 Estándares y su Organización

Toda vez que se enfrente el estudio de redes de datos, se deberán analizar estándares que regulan su funcionalidad y organizaciones que son las responsables de la generación de esos estándares. La necesidad de interconectar equipos con diferentes especificaciones de hardware o software pone en evidencia la importancia de estos estándares, pues ellos describen protocolos y tecnologías.

Se definen como sistemas abiertos, aquellos que son capaces de interactuar con otros de diferente tecnología. Estos sistemas se desarrollan en base a estándares universales, a diferencia de los sistemas propietarios que sólo pueden interactuar con otros sistemas similares. La definición de un estándar universal permite, por ejemplo, que equipos de diferentes fabricantes puedan compartir un entorno.

Para poder desarrollar estándares universales, se precisan organizaciones que coordinen las discusiones y la publicación de la documentación.

A lo largo del texto, se tratarán estándares desarrollados por algunas de las organizaciones mencionadas a continuación:

- **Organización Internacional para Estandarización (ISO,International Organization for Standardization)**: Esta organización no gubernamental fue creada en 1946. Sus miembros son organismos nacionales de máxima representatividad en el tema de estandarización, aceptándose sólo un miembro por país. Por ejemplo, IRAM de Argentina es miembro de la ISO. En este libro se desarrollará un modelo de comunicación en redes ideado por la ISO y conocido como Modelo de Referencia para Interconexión de Sistemas Abiertos (OSI, Open Sistem Interconnection). http://www.iso.org/

- **Instituto Nacional Americano de Estándares (ANSI, American National Standards Institute)**: Responsable de coordinar y publicar los estándares de tecnología de la información y computación en Estados Unidos. Es miembro de la ISO. Por ejemplo, ANSI C es un estándar para el lenguaje de programación C. http://www.ansi.org/

- **Instituto de Ingenieros Electricistas y Electrónicos (IEEE, Institute of Electrical and Electronics Engineers)**: Organización de profesionales de ingeniería eléctrica y electrónica. Uno de los estándares más conocidos de la IEEE es el proyecto IEEE 802, que permitió el desarrollo de tecnologías LAN tipo *Ethernet* y WiFi. http://www.ieee.org/index.html/

- **Alianza de Industrias Electrónicas (EIA, Electronic Industries Alliance)**: Asociación internacional de industrias cuyos estándares más conocidos se refieren al cableado de redes. http://www.eciaonline.org/eiastandards/

- **Unión Internacional de Telecomunicaciones – Sector de Estandarización para Telecomunicaciones (ITU-T, International Telecommunication Union – Telecommunication Standardization Sector)**: Organización internacional para desarrollo de estándares para la industria de las telecomunicaciones.
  http://www.itu.int/en/Pages/default.aspx

1.5.1 Organismos Específicos para Estandarización de Internet

Junto con el crecimiento de Internet, se hizo evidente la necesidad de una estructura formal de organismos para fortalecer cuestiones relacionadas con su arquitectura, estándares, políticas y muchas otras actividades.

En 1992 se creó la Sociedad de Internet (ISOC, Internet Society) para promover la evolución y crecimiento de Internet como estructura global de comunicaciones, y proveer coordinación global de actividades relacionadas con Internet. Se trata de una sociedad de profesionales con más de cien organizaciones y veinte mil miembros en ciento ochenta países. Provee liderazgo en el direccionamiento de cuestiones relativas al presente y futuro de Internet. También es la organización madre para los grupos responsables de los estándares de infraestructura de Internet y se ubica en la intersección del trabajo de grupos de desarrollo, políticas públicas y actividades de educación, tal como se representa en la Fig. 1.9.

La ISOC fiscaliza a la Junta de Arquitectura de Internet (IAB, Internet Architecture Board), que a su vez dirige la Fuerza de Tareas de Ingeniería de Internet (IETF, Internet Engineering Task Force) y la Fuerza de Tareas de Investigación de Internet (IRTF, Internet Research Task Force).

La IETF cumple su trabajo a través de varios grupos, cada uno responsable de desarrollar estándares y tecnologías en áreas tales como Internet, gerenciamiento y operaciones, enrutamiento, seguridad, transporte y aplicaciones en tiempo real. Cada área es manejada por un director, conformando en su conjunto el IESG que, a su vez, reporta ante la IAB.

La IRTF se dedica a las cuestiones relacionadas con el largo plazo para las tecnologías de TCP/IP e Internet. Es una organización más pequeña que la IETF aunque, al igual que ésta, se encuentra conformada por grupos de investigación. La IRTF es supervisada por el IRSG y la IAB.

Por otro lado, la necesidad de estandarizar determinados parámetros, por ejemplo identificadores de protocolos, o recursos globales tales como las direcciones IP, significó el surgimiento de una Autoridad de Asignación de Números de Internet (IANA, Internet Assigned Number Authority), que en 1998 fue sustituido por la Corporación de Internet para Asignación de Nombre y Números (ICANN, Internet Corporation for Assigned Names and Numbers).

IANA opera actualmente bajo ICANN, siendo aún responsable de la asignación de direcciones IP, a través de la entrega de grandes bloques de direcciones a los Registros Regionales de Internet (RIR, Regional Internet Registry) que realizan actividades de asignación a distintos ISP en una región particular del mundo. Existen cuatro de estos registros. APNIC atiende sólo Asia y el Pacífico, en tanto que ARIN se encarga de América del Norte, parte del Caribe y África sub-ecuatorial. Para América Latina y el resto del Caribe, el registro responsable es LACNIC, y para Europa, Medio Oriente, Asia Central, y África al norte del Ecuador, es RIPE NCC.

La asignación de nombres ya no es responsabilidad del IANA, sino que ICANN ha abierto el registro de nombres a muchas organizaciones en el nivel más alto de la jerarquía DNS.

En cuanto a la estandarización de protocolos, al principio se basó en un esquema de consensos: cualquier nueva propuesta se debía plasmar por escrito para ponerla a disposición del resto para su discusión. El objetivo era generar un Requerimiento de Comentarios, por eso estas presentaciones escritas se conocen con las siglas RFC (Requests for Comments). No siempre un RFC describe un

estándar, ya que muchos de estos requerimientos son meramente descriptivos, con el objetivo de clarificar conceptos.

Con la estructura inmensa de Internet actual, este esquema tan informal no daría resultados, aunque los estándares todavía se conocen con sus siglas originales RFC.

Actualmente, la responsabilidad principal de la creación de estándares es de la IETF, cuyos desarrollos se formalizan como un RFC, escritos que son publicados por el Editor de los RFC para su consideración por parte de la comunidad de Internet. Su descarga es gratuita y esta posibilidad se considera uno de los principales motivos de la explosión en el crecimiento de Internet.

No todos los RFC se convierten en estándares, sino que tienen diferentes categorías.

Existen RFC con categoría de estándar propuesto o de borrador, estos últimos conocidos como *draft*, que pueden haberse ya aprobados formalmente como estándares o encontrarse en vías de serlo. Otros son simples recomendaciones o documentos de información general. También existen propuestas en estado experimental.

Antes de que un RFC se pueda considerar estándar, debe publicarse como borrador, bajo ciertos lineamientos de creación que establece la IETF. Generalmente los escriben miembros de los grupos de trabajo de la IETF, aunque cualquier persona podría hacerlo. La revisión queda a consideración de quienes trabajan en otros grupos de trabajo de la IETF. Si la revisión arroja resultados favorables, el documento puede ser candidato a estándar y así pasar a la categoría de estándar propuesto, si el IESG lo dispone.

Aunque sea considerado como tal, estas propuestas deben revisarse mediante pruebas experimentales, probando su aptitud y aceptabilidad para la tecnología vigente. Si se sortean con éxito estas pruebas, el RFC se puede elevar de categoría estándar propuesto a estándar *draft*. Para llegar a alcanzar el estado de Estándar de Internet, la especificación debe ser tecnológicamente madura y ampliamente implementada.

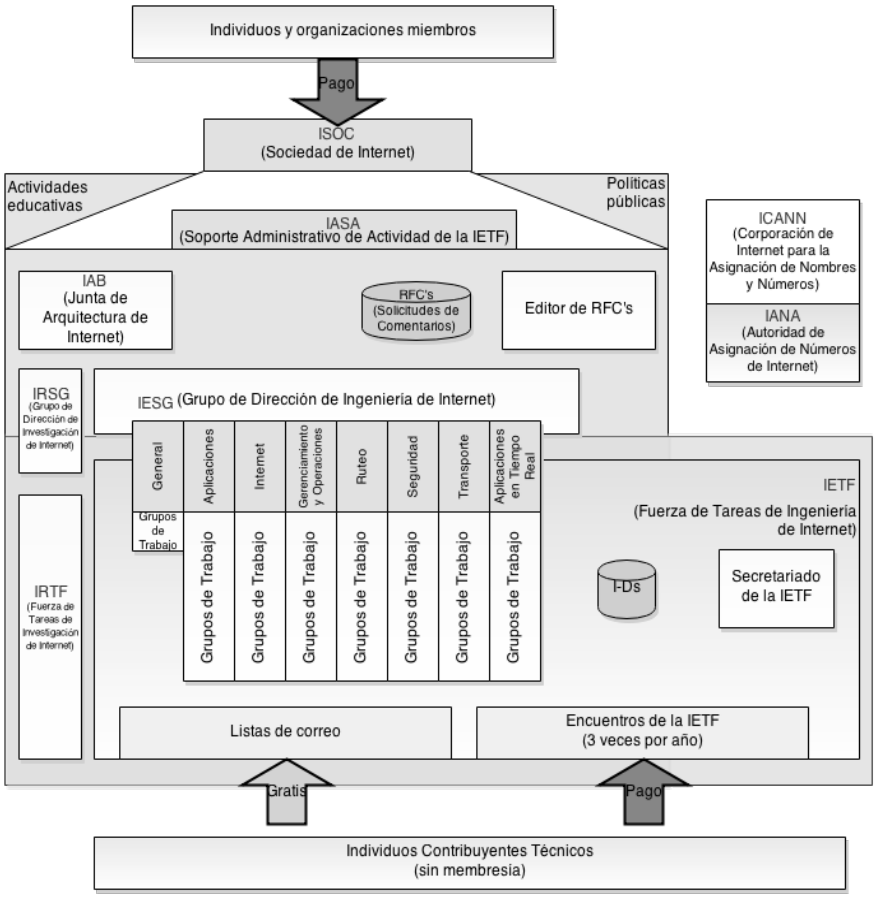


Figura 1.9 - Estructura organizacional de la ISOC.

Bibliografía

1. Tanenbaum, Andrew S., “Redes de Computadoras”, Tercera Edición. Prentice Hall Inc., 1996.
2. Stallings, William, “Comunicaciones y Redes de Computadores”. Sexta Edición. Prentice Hall Inc., 2000.
3. Commer, Douglas E., “Internetworking with TCP/IP”, Vol I Principles, Protocols, and Architecture. Prentice Hall Inc., 1995.
4. Cohen, Danny, “Computer History Museum Exhibits Internet History” http://www.computerhistory.org/internet_history/
5. Leiner, Barry M., Cerf, Vinton G., Clark, David D., Kahn, Robert E., Kleinrock, Leonard, Lynch, Daniel C., Postel, Jon, Roberts, Larry G., Wolff, Stephen, “Brief History of the Internet”. http://www.internetsociety.org/sites/default/files/Brief_History_of_the_Internet.pdf
6. Índice de RFC, http://www.ietf.org/rfc/
7. The Internet Engineering Task Force (IETF): http://www.ietf.org/

CAPÍTULO II

Modelo OSI y Arquitectura TCP/IP

En la década de los 80 se produjo un crecimiento caótico de las redes originado principalmente por la aparición de nuevas tecnologías y su desarrollo comercial. Como consecuencia, se presentó un problema debido a que no existía un lenguaje común para la comunicación y esto dificultaba enormemente la posibilidad de interconexión.

Para salvar tantos inconvenientes, la ISO conformó un comité para que generara un modelo estándar que rigiera este tipo de comunicaciones. Producto de estas investigaciones, casi a mitad de la década de los 80, surgió el modelo OSI, de casi nula implementación comercial.

Por su parte, durante la década de los 70, un grupo de ingenieros e investigadores, había desarrollado una arquitectura basada en protocolos de aplicación en la red ARPANET. Se trataba de la arquitectura que hoy se conoce como TCP/IP, siendo actualmente la más difundida para la comunicación en redes de datos. Casi al mismo tiempo que surgía el modelo OSI, TCP/IP se integró en la versión 4.2 de la distribución del sistema operativo UNIX de la Universidad de Berkeley. Enseguida se sumaron las integraciones en versiones comerciales de UNIX, y TCP/IP se convirtió en el estándar de Internet.

2.1 Arquitectura de Protocolos

En un entorno de comunicación en red, los sistemas no utilizan un único protocolo, sino un conjunto denominado familia, pila o arquitectura de protocolos, que actúan de manera cooperativa, debiendo ser capaces de comunicarse entre sí. Uno de los objetivos de diseño sería crear un entorno de división del trabajo, de tal manera que los protocolos se pudieran integrar de manera independiente, aunque constituyendo un conjunto en cooperación.

Dividir un problema complejo, como es el de la comunicación de datos en redes, en módulos simples, cada uno con una funcionalidad bien definida, ayuda a entender más fácilmente las dificultades existentes.

Las consecuencias de lograr una división óptima, se traduce en una serie de ventajas:

- **Documentación más sencilla de entender**: como consecuencia de dividir un problema complejo en partes más pequeñas y específicas, se facilita la descripción y el entendimiento, sobre todo para quienes trabajan en implementaciones.

- **Especialización**: la división puede generar más fácilmente expertos en aspectos particulares.

- **Facilidad de modificación**: una división óptima permite realizar modificaciones en las partes sin afectar al todo, ya sea en los casos en que se detecten dificultades, como en los que se propongan mejoras.

Cada módulo aportará lo suyo, ya sea en términos de *hardware* o de *software*, y será capaz de interactuar con otros módulos, de sus mismas características, ubicados en otros dispositivos en la red. Inclusive, bajo estas premisas, es posible lograr la comunicación entre dispositivos de diferentes características técnicas.

Uno de los interrogantes más difíciles de resolver se relaciona con decidir cuántas capas o módulos se deben definir y cuáles funcionalidades se asignarán a cada uno, para que el diseño de los correspondientes protocolos se convirtiera en una tarea de complejidad manejable.

De ese modo se diseñó la comunicación TCP/IP que, aunque estrictamente hablando, es una pila de protocolos, se puede asimilar a un modelo de comunicación conformado por capas.

A modo de ejemplo, imagínese dos sistemas A y B en distintos puntos de una red, cada uno con el mismo conjunto de capas, intentando comunicarse bajo las premisas descriptas. Se propone un modelo de 3 capas, separadas entre sí por las líneas punteadas que se presentan en la Fig. 2.1. Dentro de cada capa, un cubo indica la implementación en *software* de los protocolos. En cada sistema existe un flujo vertical de comunicación entre capas adyacentes. Entre ambos sistemas, el flujo de mensajes es horizontal entre capas del mismo nivel, también conocidas como capas o niveles pares.

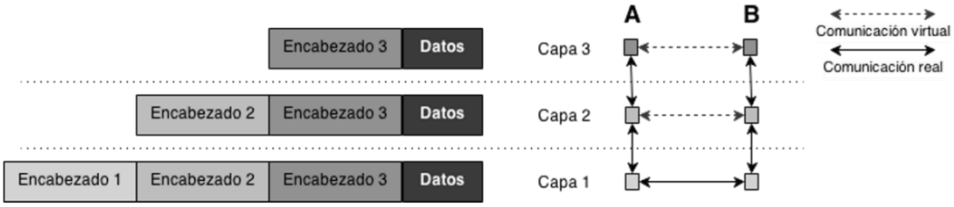


Figura 2.1 - Un modelo de comunicación por capas.

La comunicación entre sistemas funciona correctamente utilizando una técnica conocida como encapsulado. Este concepto implica agregar a un mensaje,

o conjunto de datos, un encabezado con información de control. La regla dentro de cada sistema, es que cada protocolo genera un mensaje que se encapsula dentro de un encabezado del protocolo de la capa inferior. En el caso del ejemplo, un mensaje generado en A, denominado Datos, llegará al final inferior de la pila de protocolos del sistema A encapsulado de la siguiente manera: $(Encabezado1/Encabezado2/Encabezado3/Datos$. Una vez transportado sobre el medio subyacente, arribará al sistema B, y subirá por la pila pasando por un proceso de des-encapsulado, donde cada protocolo de un nivel observará en el encabezado del protocolo del nivel par la información que se considera relevante a ese nivel. Luego entregará a la capa superior el mensaje con el encabezado que le corresponda. En el último nivel del sistema B, se entregará finalmente el mensaje $(Encabezado3/Datos)$, generado por A.

En este ejemplo, se ha dividido el trabajo relacionado con la comunicación de datos en tres capas, como una forma de aproximación al problema real. La capa más baja es la que más se relaciona con los detalles de la transmisión física de los datos. Funcionalidades más cercanas al *hardware* se relacionarían con este nivel, tales como codificación, velocidad de transmisión, conector de acceso a la red, niveles de tensión o sistema de señalización.

En contrapartida, el nivel superior es el más cercano a la interacción con el usuario. Se trata de las aplicaciones de *software* desarrolladas para la comunicación en red.

En el medio, se ha incluido un nivel que oficia de soporte para las aplicaciones, acomodando la comunicación a los diferentes tipos posibles de medios o métodos de acceso presentes en el nivel inferior, pero ocultando ese detalle al nivel superior.

Se podría decir que existe una comunicación real, asociada al flujo vertical de mensajes en cada sistema, y una comunicación virtual, indicada en el dibujo por las flechas punteadas horizontales. Esta última se denomina comunicación entre protocolos pares.

2.2 Modelo OSI

Considerando que estandarizar sistemas o protocolos permite desarrollar entornos inter-operativos, aún cuando provengan de diferentes fabricantes, en 1977 se formó un subcomité de la ISO con la misión de desarrollar una arquitectura para modelar la comunicación entre sistemas en una red. El modelo de referencia se llamó Modelo de Interconexión de Sistemas Abiertos (OSI, Open System Interconection) y fue publicado en el año 1984.

Como convenía a un problema complejo, era necesario dividir las tareas y asignar estándares por funcionalidad, es decir definir una arquitectura. En el modelo OSI las funciones se distribuyen entre un conjunto jerárquico de capas, necesarias para la comunicación entre entidades de distintos sistemas.

Cada capa debía apoyar su funcionalidad en la de la capa inferior. Las capas inferiores realizarían las funciones más primitivas, ocultando sus detalles

de implementación a las capas superiores. Este concepto de ocultamiento es el que permitiría que el reemplazo de una capa no afectara a las demás.

El gran desafío para el comité de la ISO fue encontrar el número de capas adecuado para que un gran problema se viese subdividido en problemas más sencillos, por sobre todo fáciles de describir e implementar.

Para enfrentar el dilema, el comité definió un conjunto básico de servicios que cada capa debería ser capaz de cumplir. Como premisa, se trabajó sobre la idea de que la división en capas debía ser tal que en una misma capa se agrupasen funcionalidades similares, siempre que el número total no resultase muy grande, para que la descripción del modelo, de la cual dependería finalmente cualquier implementación, fuese más sencilla.

También razonaron que el número total de capas establecería una relación de compromiso con respecto a la cantidad de bits de cualquier mensaje transportado. Como cada capa supondría con el agregado de un encabezado, con campos relacionados con la funcionalidad de la misma, si la cantidad de capas era muy grande, la relación entre los bits agregados como información de control con respecto a la cantidad de bits propios del mensaje, resultaría ineficiente. Es decir que debían encontrar una relación de *overhead* apropiada. Se denomina *overhead* a la relación en bits entre la longitud del mensaje propiamente dicho y la longitud total del paquete generado por el agregado de encabezados, procesamiento conocido con el nombre de encapsulado.

El modelo de referencia OSI se desarrolló sobre un conjunto de siete capas, numeradas de 1 a 7 según la posición de la capa en el modelo. Cuanto más bajo es el número, más cerca se encuentra la capa del *hardware* utilizado para implementar la comunicación en red. Por ejemplo, la Capa 1, denominada Capa Física, directamente es la de implementación en *hardware*, relacionándose su funcionalidad con técnicas de señalización, niveles de tensión, velocidades, anchos de banda y tipos de medios de transmisión, con sus respectivas particularidades. La Capa 2 ya posee funcionalidades que pueden desarrollarse en *hardware* y *software*, en tanto que la Capa 7 o Capa de Aplicación, es el nivel más cercano al usuario y a los programas que éste utiliza. Estos programas, a su vez, se comunican con el sistema operativo instalado en el dispositivo.

A medida que se recorre el modelo desde la Capa 1 a la Capa 7, se sube en el conjunto de capas, como se puede apreciar en la Fig. 2.2.

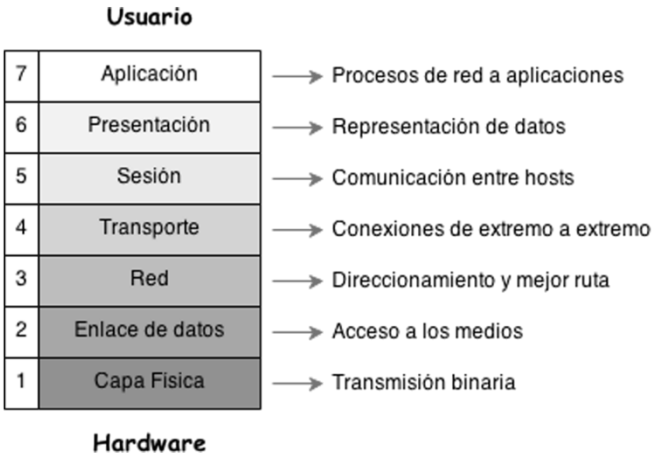


Figura 2.2 - Modelo OSI.

En el lenguaje descriptivo de OSI, una Capa N es una capa genérica que tiene por debajo la Capa (N-1) y por encima la Capa (N+1). Se dice que la Capa(N-1) ofrece sus servicios a la Capa N, mientras que ésta, a su vez, ofrece sus servicios a la Capa(N+1).

En este modelo, el mecanismo de comunicación entre capas adyacentes se denomina interfaz. El concepto de interfaz OSI no se parece al que usamos comúnmente hoy en día, puesto que se refiere a la forma en que se pasan los datos entre capas adyacentes. El concepto actual de interfaz tiene más relación con la conexión física y funcional de un dispositivo.

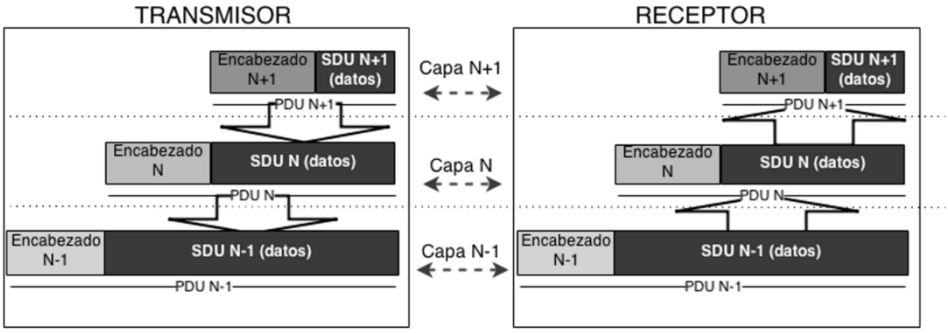


Figura 2.3 - Comunicación vertical entre capas del modelo OSI. Encapsulado.

Cuando se desea transmitir algo sobre la red, los mensajes de una aplicación recorren verticalmente la pila de capas del modelo que representa el dispositivo transmisor, desde arriba hacia abajo. En el extremo receptor, los mensajes se procesan también de manera vertical pero al revés, desde abajo hacia

arriba. Esta comunicación vertical es una comunicación real entre capas adyacentes.

Aparte de la comunicación real vertical, existe una comunicación virtual, de carácter horizontal, entre capas con la misma numeración, denominadas capas pares. Esta comunicación horizontal es lo que hace posible la transmisión de datos en redes, siempre y cuando los extremos que se están comunicando acuerden una serie de reglas, un lenguaje común para poder entenderse entre módulos concretos de *software* o *hardware*. Este es el concepto principal relacionado con la funcionalidad de de un protocolo.

Un protocolo reglamenta la comunicación horizontal, a través de la definición de mensajes denominados Unidades de Datos de Protocolos PDU. Estas unidades se componen de dos partes: el propio mensaje y un encabezado. En el proceso de transmisión, una Capa N genera una PDU, que se identifica como N-PDU, que se pasa a la capa adyacente inferior (N-1), donde conforma la (N-1)-SDU, es decir la porción de datos de la Capa (N-1). Esta capa debe conformar su propia (N-1)-PDU, adicionando a la (N-1)-SDU el encabezado propio. Este procedimiento es lo que se conoce como funcionalidad de encapsulado, como se puede apreciar en la Fig. 2.3.

De este modo, cada capa toma la PDU de la capa superior, SDU de su nivel, y le agrega su propio encabezado conformando una nueva PDU, que entrega luego a la capa adyacente inferior. El mecanismo se repite hasta llegar a la Capa Física, que termina recibiendo un mensaje de la Capa de Aplicación encapsulado por los encabezados de la Capa 7 hasta la Capa 2.

El modelo OSI definió las principales funciones asignadas a cada una de las capas:

- **Capa Física**: especifica el tipo de medio a utilizar, pudiendo tratarse de cable coaxial, fibra óptica, par trenzado o medio inalámbrico, entre los más comunes. También define la forma de adaptar los bits generados por las computadoras al propio medio en cuestión. Se trata de técnicas de codificación y señalización, ya sea banda base o pasa banda. Esta capa se encarga también de los detalles de operación de cables, conectores, transceptores, tarjetas de interfaz de red (NIC, Network Interface Cards) y de todo lo relacionado con la transmisión y recepción de los datos en la red. Describe la topología de la red y el tipo de transmisión, en general *half duplex* o *full duplex*, junto con los valores de tensión y la duración de los símbolos. En el caso de los conectores, especifica el tipo y cantidad de pines, describiendo las funciones de los mismos.
  Formalmente, el modelo OSI menciona la definición de características mecánicas (propiedades físicas de la conexión y del medio de transmisión que definen tipos de conectores o adaptadores), eléctricas (niveles de tensión, velocidades de transmisión), funcionales (definiciones de funciones de circuitos) y de procedimientos (secuencia de eventos) de la interfaz física.
  Las placas de red, los repetidores y transceptores son dispositivos de Capa Física.

- **Capa de Enlace de Datos**: su función es presentar a la capa superior, la Capa de Red, un enlace físico seguro, independientemente de la Capa Física existente. En el modelo OSI la Capa de Enlace se considera dividida en dos subcapas: la subcapa de Control de Acceso al Medio (MAC, Medium Access Control) y la subcapa de Control Lógico de Enlace (LLC, Logical Link Control).
  La MAC es la subcapa más cercana a la Capa Física y su misión es definir la forma de adaptación para el acceso al medio específico que se encuentre por debajo. Por ejemplo, en muchas redes se comparte el mismo medio físico debiéndose imponer una serie de reglas para el acceso equitativo, para evitar posibles conflictos.
  La subcapa LLC se define como una especie de interfaz entre la MAC y la Capa de Red, independiente del medio físico subyacente, proveyendo servicios a la Capa 3 pero ocultando los detalles de diferentes posibilidades a nivel de Capa 2.
  Las funciones de la Capa de Enlace se relacionan con activar, desactivar y mantener un enlace, controlar errores, controlar el flujo de la comunicación para acomodar diferencias de procesamiento entre capas pares, ordenar los mensajes en la entrega, manejar funcionalidades de re-transmisión y realizar tareas de delimitación y sincronismo.
  También, a nivel de Capa 2, se define un esquema de direccionamiento de significado local, denominado dirección de hardware o dirección MAC, que permite identificar unívocamente un dispositivo dentro de una red.
  Las Unidades de Datos de Protocolo a este nivel se denominan tramas.

- **Capa de Red**: se trata de la capa que define cómo es posible conectar diferentes redes entre sí. A diferencia de la Capa de Enlace, que se involucra en la comunicación entre dispositivos de la misma red, la Capa de Red se encarga de que sea posible la comunicación entre dispositivos, aún cuando los mismos se encuentren en distintas redes. Una de las principales funciones asociadas a esta capa es el enrutamiento, que se traduce en la posibilidad de manejar paquetes provenientes de distintas fuentes, observar ciertos campos a nivel de Capa de Red, y enviarlos de manera consistente hacia la red a la que deben arribar.
  En la arquitectura de red del modelo OSI existen sistemas finales, que corren aplicaciones, y sistemas intermedios, que sólo cumplen funciones de enrutamiento. Los dispositivos para interconexión de redes que operan en este nivel se conocen con el nombre de *routers*. Uno de sus principales trabajos consiste en comunicarse entre ellos por medio de protocolos de enrutamiento, para poder determinar automáticamente las mejores rutas a diferentes destinos. De este modo pueden realizar eficientemente la funcionalidad de enrutamiento.
  De todas maneras, cada sistema posee un identificador único en la red y se necesita una funcionalidad de red en todos. Por este motivo, cada dispositivo lleva asociada una dirección lógica, de significado global, independiente del *hardware* particular, que tiene carácter único,

permitiendo la comunicación con dispositivos fuera de la red local. Esta dirección es parte de la información de control de Capa de Red que usan los dispositivos intermedios para decidir hacia dónde encaminar un paquete.
El servicio ofrecido por esta capa puede ser orientado a la conexión o sin conexión. En el primer caso, la Capa de Red debe proveer los medios para establecer, mantener y liberar las conexiones de red.
En cualquier caso, en la Capa de Red se deberán ofrecer los medios necesarios para resolver diferencias cuando los paquetes atraviesan redes de distintas características. Un parámetro característico en estos casos es la Unidad de Trasferencia Máxima (MTU, Maximum Transfer Unit) que la red es capaz de transportar. Cuando el tamaño de un paquete es mayor que la MTU de la red subyacente, se hace necesario dividirlo en pedazos más pequeños, un proceso que se conoce con el nombre de fragmentación. Luego, para su entrega correcta, los fragmentos deben re-ensamblarse para conformar el mensaje original. Ambas tareas son funcionalidades propias de la Capa de Red.
Las Unidades de Datos de Protocolo de este nivel se suelen conocer con el nombre de paquetes o datagramas.

- **Capa de Transporte**: es el nivel donde se ofrece un control para el transporte de los datos entre los dos sistemas finales de la comunicación. Se dice que es la primera capa del tipo *end-to-end*, refiriéndose esta expresión a los extremos finales de la comunicación: fuente y destino de la información.
  Puede ofrecer servicios de dos tipos: orientado a la conexión o sin conexión. En el caso de servicio orientado a la conexión, la funcionalidad también puede encontrarse en la Capa de Red aunque, en la actualidad, es muy común hallarla en la Capa de Transporte por cuestiones de fiabilidad. En el segundo caso, la funcionalidad es apropiada para esquemas de comunicación de tipo transaccional, con intercambio de baja cantidad de volúmenes de datos.
  La misión principal de la Capa de Transporte es mantener un acceso uniforme a la red, independientemente del medio de comunicación disponible, especialmente con el objetivo de blindar el nivel superior respecto de los mecanismos de las redes subyacentes. Para lograrlo, la capa debe contar con todos los mecanismos de optimización de recursos que aseguren la calidad de la conexión. Por este motivo, la Capa de Transporte muchas veces cuenta con funcionalidades para manejo de errores, medición del retardo máximo permitido en una conexión, marcado de tráfico para prioridad, manejo de fallas y control de flujo.
  Como en las Capas 2 y 3, también a nivel de Capa de Transporte existe un esquema de direccionamiento, aunque usado para distinguir entre diferentes programas dentro de un mismo dispositivo. La existencia de este esquema hace posible que varias aplicaciones de usuario se encuentren conectadas a la red al mismo tiempo.

Las Unidades de Datos de Protocolo de este nivel se conocen como segmentos.

- **Capa de Sesión**: el modelo OSI plantea la comunicación entre sistemas finales como un diálogo que hay que organizar y sincronizar. La inclusión de una Capa de Sesión ofrece a los usuarios el acceso a la red, previa codificación de datos que realiza el nivel superior, permitiendo el establecimiento y desconexión de una sesión. El concepto de sesión se refiere al acceso remoto desde un terminal a un dispositivo, por ejemplo para transferencia de archivos.
  Las funciones de la Capa de Sesión se relacionan con organizar, sincronizar y administrar el intercambio de información entre entidades del nivel superior. Es función de esta capa la administración de testigos o *tokens* para controlar el orden del diálogo. Con estos elementos, se podrían definir puntos de comprobación o *checkpoints* durante una sesión. En caso de fallas, estos puntos servirían como marcas de sincronismo, permitiendo retomar una sesión desde un punto determinado, anterior al evento de error registrado, sin necesidad de arrancar de nuevo la sesión.
  En la actualidad, muchas de las funciones de esta capa se proveen a las capas superiores por medio de conjuntos de comandos conocidos como Interfaz de Programación de Aplicaciones (API, Application Program Interface). Las API definen servicios estandarizados para facilitar las comunicaciones sobre una red, permitiendo a los programadores de aplicaciones desprenderse de los detalles de implementación de los niveles inferiores. Por ejemplo, el término *socket* se aplica a una API para la familia de protocolos de TCP/IP provista usualmente por el sistema operativo.

- **Capa de Presentación**: esta capa pretende descargar de las aplicaciones la cuestión de la representación y manipulación de datos estructurados. En este sentido, define el formato de los datos que se van a intercambiar entre las aplicaciones para la resolución de diferencias sintácticas entre sistemas. Algunas de estas representaciones son típicas de ciertos dispositivos, otras veces la representación es diferente según el sistema operativo. Lo importante es la preservación de su significado entre ambos extremos de la comunicación.
  Dos funcionalidades solían asociarse a esta capa: compresión y cifrado de datos. Actualmente, la compresión es una funcionalidad agregada a modo de programa o aplicación, mientras que el cifrado de datos tiende a instalarse en niveles inferiores, de manera transparente al usuario.

- **Capa de Aplicación**: en esta capa se lleva a cabo el procesamiento final de la información a intercambiar. La Capa de Aplicación provee servicios a los programas usuario. Es la capa del modelo OSI más cercana al entorno usuario. Es responsable de la semántica de la información intercambiada. No todas las aplicaciones son susceptibles de

estandarizar, pero determinados procedimientos son comunes a todos los protocolos de aplicación. Por ejemplo, empezar y terminar una asociación entre procesos de aplicación podría ser una funcionalidad frecuente.

La comunicación entre capas pares en diferentes dispositivos es lógica, excepto a nivel de la Capa Física, donde la comunicación es por *hardware*. En la transmisión, cada capa conforma una PDU a su propio nivel y debe pasarla a la siguiente inferior. La capa inferior la recibe como parte del servicio que brinda a la superior, aunque en este nivel el mensaje se considera una SDU hasta que la capa inferior agregue su propio encabezado. Es decir que, en cada capa se produce un proceso de encapsulado de una SDU en una PDU.

Como se ha explicado, este proceso de encapsulado continúa hacia abajo hasta llegar a la Capa Física. En la Fig. 2.4 se ofrece un esquema de los mensajes en cada nivel del modelo. En este caso, la representación del proceso de encapsulado y des-encapsulado se grafica para dos sistemas directamente conectados, poniendo en relevancia la información de control en cada nivel del modelo OSI.

En la recepción, cada capa utilizará la información referida a su propio encabezado, subiendo a la capa superior el mensaje encapsulado a ese nivel. La comunicación virtual en el plano horizontal en cada capa se refiere a la interpretación de los encabezados provenientes de la capa par del lado transmisor.

La **comunicación directa** presentada en la Fig. 2.4 es aplicable en determinados entornos, entre máquinas de una red local. El objetivo principal de la comunicación en redes es ofrecer la posibilidad de interconexión entre redes. Una comunicación entre dispositivos alojados en diferentes redes es una **comunicación indirecta**. Este tipo de comunicación es transparente para los propios dispositivos finales que, simplemente, entregan los mensajes a su red local para que, de alguna manera, puedan arribar al destino final. El pasaje desde una red a otra precisa de dispositivos especiales, capaces de realizar el trabajo de re-envío. El proceso completo, realizado de manera conjunta y cooperativa, se denomina encaminamiento o enrutamiento.

Como se ha mencionado, la actividad de re-envío de mensajes depende de la funcionalidad de ruteo, típica de la Capa 3.

En el caso de una comunicación entre sistemas ubicados en diferentes redes, es decir conectados a través de nodos de re-envío o *routers*, el proceso de encapsulado inicial se cumple como se ha descripto. El mensaje se encapsula hacia abajo en la transmisión, contando con una dirección de destino externa y global, a nivel de Capa 3. Al ser reconocida como una dirección no local, se entiende que el mensaje debe ser pasado a un dispositivo intermedio, responsable del ruteo hacia la red destino. Esto se logra haciendo uso de una dirección apropiada a nivel de Capa 2. Así, el dispositivo de salida de la red local, recibirá el mensaje, lo des-encapsulará hacia arriba, hasta llegar al nivel de Capa de Red, donde se determinará si el mensaje, debido a su dirección destino, precisa nuevamente pasar por el proceso de re-envío. Tomada la decisión, el mensaje se volverá a encapsular, hacia abajo, y se enviará al siguiente *router* en la ruta. Luego de la eventual repetición de este proceso en varios dispositivos

intermedios, el mensaje arribará al *router* de entrada a la red destino, que procederá a su entrega directa.

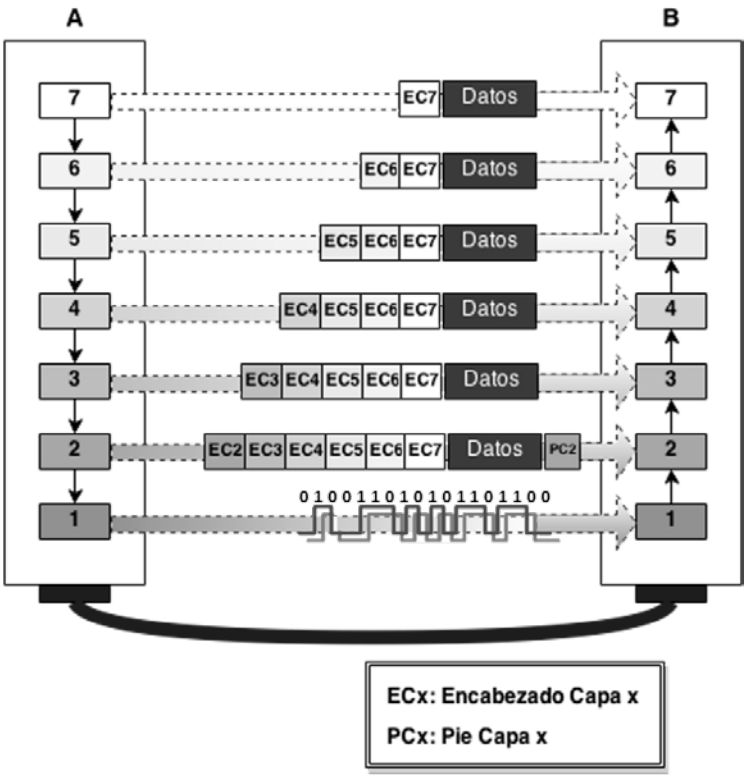


Figura 2.4 - Sistemas directamente conectados.

El proceso de comunicación indirecta se grafica en la Fig. 2.5 en el caso de dos dispositivos, A y B, en diferentes redes, conectadas entre sí por un *router*. Cuando la comunicación no es directa, a nivel de Capa de Red, es necesario que el encabezado del mensaje original cargue una dirección global para identificación del destino final B, que se señala como Dir_Global_B. Por otra parte, en la Capa de Enlace se agregará una dirección con significado local, referida como Dir_Local_RI, a los fines de identificación del *router* intermedio.

En el esquema simplificado de la Fig. 2.5, el único *router* en el camino, denominado Router Intermedio, recibe el mensaje por el vínculo físico con el Sistema A, y lo pasa a su propia Capa de Enlace, que reconoce en su encabezado su dirección local Dir_Local_RI. De este modo, el Router Intermedio asume que le corresponde procesar el mensaje, por ejemplo para chequeo de errores, y entregarlo a la Capa de Red, luego de remover el encabezado de Capa de Enlace. La Capa de Red revisa la dirección destino a su nivel, Dir_Global_B, para determinar si el destino final se encuentra en alguno de los vínculos locales del *router* o si debe re-enviarlo a otro *router*. Para tomar la decisión, consulta

información almacenada en memoria. Se trata de la información de ruteo que, en cierto modo, refleja la topología de la red. En el caso que el mensaje tuviera que ser re-enviado a otro *router* en su camino a destino, la Capa de Red mantiene la dirección global del destino, Dir_Global_B y baja el paquete a la Capa de Enlace, que tendrá que escribir en su encabezado la dirección local del siguiente *router*, antes de la entrega a la Capa Física para su transmisión. Este proceso se repetiría en cada nodo intermedio antes de llegar al destino final. El último *router* de la cadena reconocerá la Dir_Global_B como dependiente de una de sus conexiones de red, efectuando la entrega de manera directa. En este tramo de la comunicación, la dirección de Capa de Enlace deberá referirse al dispositivo receptor.

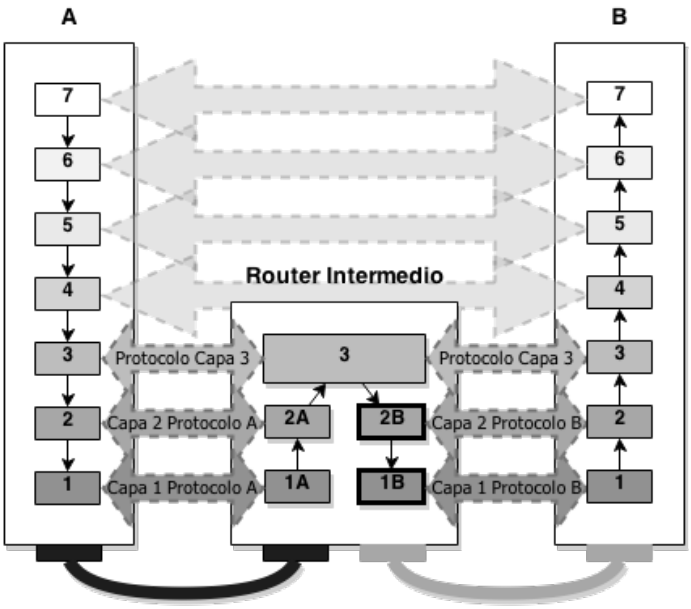


Figura 2.5 - Comunicación con *router* Intermedio.

Se puede observar que en los nodos intermedios no es necesario ningún procesamiento a niveles superiores a la Capa de Red. Por eso, a estos dispositivos especiales, se los modela como un conjunto de tres capas.

Al llegar a destino final, el mensaje es reconocido como propio por la dirección local de Capa de Enlace, Dir_Local_B, y la dirección global de Capa de Red, Dir_Global_B. Así va ascendiendo por las diversas capas del sistema B hasta que el mensaje original es entregado a la Capa de Aplicación de B.

Con esta descripción, se ha pretendido dejar claro un concepto: en el viaje desde fuente a destino, las direcciones globales del mensaje, presentes en el encabezado de Capa de Red, no cambian. Cuando la entrega es indirecta las direcciones de significado local, en el encabezado de Capa de Enlace, sí pueden cambiar en cada tramo de la comunicación.

Como se observa en la Fig. 2.5, las Capas Física, de Enlace y de Red, se comunican entre dispositivos directamente conectados, en tanto que el diálogo entre las capas por encima de la Capa de Red, es entre sistemas finales. De este modo, el protocolo usado en dichos niveles ha de ser común entre sistemas conectados directamente, pero cada enlace en el camino puede ser de diferente naturaleza. Esta propiedad demuestra uno de los puntos más fuertes del modelo en capas, la posibilidad de conectar diferentes redes entre sí, denominada capacidad de *internetworking*.

La publicación del modelo OSI en 1984, pareció abrir el camino para la estandarización de protocolos que permitieran la inter-operatividad entre equipos de distintos fabricantes pero, a pesar de las expectativas generadas por el trabajo de la ISO, los desarrollos de protocolos en base al modelo OSI no llegaron casi a hacerse realidad. Se podría mencionar el protocolo de nivel de red X.25 y el de mensajería X.400 como casi los únicos que tuvieron una implementación práctica importante.

Para la misma época, la comunicación mediante los protocolos TCP/IP se observaba como un logro experimental de un grupo de ingenieros e investigadores. Es de destacar que la primera versión de TCP data de 1973, aunque formalmente fue documentado en la RFC 675, a fines del año 1974. Por ende, TCP/IP no se basó en el modelo OSI.

A lo largo de la historia de Internet, sucedieron muchas circunstancias que permitieron el avance explosivo de los protocolos TCP/IP. Uno de los motivos más importantes fue su simplicidad, que resalta aún más cuando se compara con la complejidad conceptual del Modelo OSI, en cuyo detalle no hemos ahondado en esta presentación.

2.3 Arquitectura TCP/IP

A diferencia del modelo OSI, TCP/IP es en realidad una pila de protocolos. En términos comparativos, IP cumpliría la funcionalidad requerida por la Capa de Red de OSI, en tanto que TCP se correspondería con la Capa de Transporte, aunque en realidad la pila consta de más de un par de protocolos.

Como en el modelo OSI, en la arquitectura TCP/IP existe la idea de un protocolo ofreciendo sus servicios a los que se apoyan sobre él, fundamentalmente para poder hacer realidad la premisa de interconexión de redes. También se desarrolla la idea de comunicación virtual entre protocolos pares entre dispositivos conectados directamente para los niveles más bajos, y entre los extremos finales, para los niveles superiores. Otra característica que se sostiene es el concepto de encapsulado.

Dado que TCP/IP es un conjunto de protocolos, su funcionamiento se comprende estudiando dichos protocolos. Cada uno de ellos realiza una serie de funciones necesarias para implementar la comunicación en red, pero trabajan en cooperación, justamente para lograr este objetivo. Los protocolos más importantes de esta arquitectura son los que le dan el nombre, TCP e IP, y también el Protocolo de Datagrama de Usuario (UDP, User Datagram Protocol). Cada uno

de ellos soporta otros protocolos que apoyan su funcionalidad sobre este núcleo de la arquitectura.

Con fines comparativos, TCP/IP se podría modelar en 4 niveles con funcionalidades semejantes a sus pares del modelo OSI, como se puede apreciar en la Fig. 2.6.

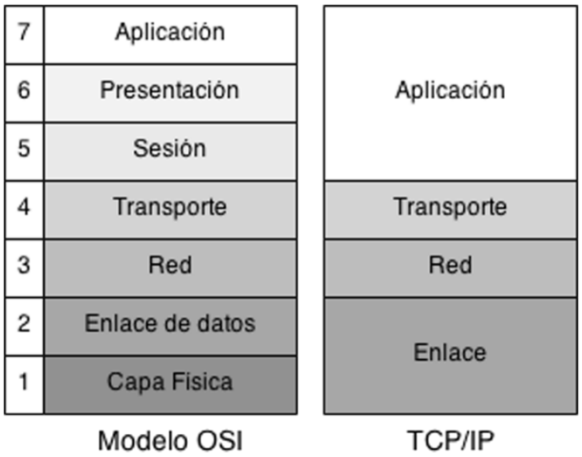


Figura 2.6 - Comparación Modelo OSI vs. Arquitectura TCP/IP

La estructura es más simple que la del Modelo OSI, tiene su paralelismo con éste, pero en el caso de implementaciones reales muchos protocolos no caben particularmente en una capa de la arquitectura convencional.

A continuación se presenta un resumen de las principales características de cada nivel:

- **Aplicación**: en este nivel la comunicación es entre procesos o aplicaciones que manejan datos de usuario y se los deben comunicar a otros procesos o aplicaciones en otro punto de la red. Se trata del nivel más alto de la arquitectura, comparable a las tres capas de mayor numeración del Modelo OSI, aunque muchas veces sería imposible establecer una relación directa entre protocolos en este nivel con funcionalidades específicas de una capa OSI. Protocolos tales como SMTP para transporte de mensajes de correo electrónico, FTP para transferencia de archivos, SSH para conexiones remotas seguras y HTTP para navegación en la web, operan en este nivel.

- **Transporte**: es el nivel de comunicación entre las máquinas en red que constituyen los extremos finales de la comunicación. Las mismas pueden encontrarse en redes diferentes conectadas a través de *routers*, o en la misma red. Los protocolos de este nivel, TCP y UDP, ofrecen a los del nivel superior una interfaz de acceso a la red uniforme, sin que importe

el tipo de conexión o red subyacente. Se les asocia funcionalidades relacionadas con el control de error y control de flujo, aunque TCP también permite manejar una conexión, porque es un protocolo del tipo orientado a la conexión. Es decir que TCP se encarga de brindar confiabilidad, abrir, mantener y cerrar conexiones solicitadas por aquellos protocolos de nivel superior que requieren sus servicios. Para ello, es capaz de manejar datos fuera de orden, errados o duplicados. Su funcionalidad incluye el control de congestión y el manejo de paquetes perdidos. Protocolos tales como HTTP apoyan su funcionalidad en TCP. Por su parte, UDP ofrece un servicio sin conexión, apto para aplicaciones transaccionales, como es el caso de DNS.

Tanto TCP como UDP incluyen un esquema de direccionamiento para identificar aplicaciones. Se trata de campos de encabezado, de $16\ bits$, conocidos como números de puerto.

- **Red**: existe un único protocolo a este nivel y su función es la de lograr la interconexión de redes. IP cuenta con capacidad de manejo de datagramas o paquetes y su misión es que los mismos se muevan hacia el destino, a través de diversas redes. El servicio de transmisión de datagramas IP es un servicio sin conexión, no confiable, pero que permite lograr uno de los objetivos más importantes de manera sencilla: la inter-conectividad. Como su principal trabajo es el ruteo, en su versión más antigua, el protocolo IPv4 posee un esquema de direccionamiento de tipo jerárquico, de $32\ bits$, conocido como esquema de direcciones IP. La versión más moderna IPv6 cuenta con un espacio de direcciones mucho más grande, de $128\ bits$. El servicio de ruteo es tipo salto a salto o *hop-by-hop*, con comunicación entre sistemas conectados directamente hasta llegar al *router* más cercano al destino final. IP puede cargar mensajes de muchos protocolos del nivel superior, identificados con un campo especial en el encabezado, denominado campo de número de protocolo, que le permite realizar multiplexado para poder entregar de manera correcta el mensaje encapsulado. Entre los protocolos encapsulados por IP, podemos mencionar: el Protocolo de Mensajes de Control de Internet (ICMP, Internet Control Message Protocol), y los protocolos TCP y UDP.

- **Enlace**: es el nivel que atiende las cuestiones relacionadas con el tipo de red local sobre la que se dirige la comunicación. En términos comparativos, los protocolos en este nivel de la pila se ubican en las Capas de Enlace y Física del modelo OSI. Es decir que aquí se manejan los detalles del medio de comunicación sobre el que se transmitirán y recibirán los datagramas generados por IP entre dos máquinas diferentes conectadas indirectamente o sobre el mismo enlace. La funcionalidad de los protocolos de este nivel puede desarrollarse en *hardware*, por ejemplo en las placas de red, y en *software*. Antes de enviar los datagramas sobre el medio físico específico, estos serán acondicionados con el agregado de un encabezado generado por protocolos de este nivel, por ejemplo para

agregar algún tipo de direccionamiento con significado local, a diferencia de IP cuyo direccionamiento es de significado global. Muchos protocolos de este nivel también agregan al final un campo para control de errores. Uno de los protocolos más antiguos que operan en este nivel es el Protocolo de Resolución de Direcciones (ARP, Address Resolution Protocol). En el caso de redes de acceso múltiple, este protocolo trabaja en conjunto con el protocolo IPv4, para relacionar direcciones IP con las correspondientes direcciones del nivel de enlace.

Como sucede en el modelo OSI, en la transmisión un mensaje atraviesa la pila de protocolos hacia abajo, agregándosele en cada caso información de control, denominada encabezado, como se puede apreciar en la Fig. 2.7. En el lado receptor, el proceso es inverso: a medida que los datos se mueven hacia arriba, cada protocolo obtiene información de los campos de control de su propio encabezado, y luego sube el mensaje encapsulado, sin el encabezado.

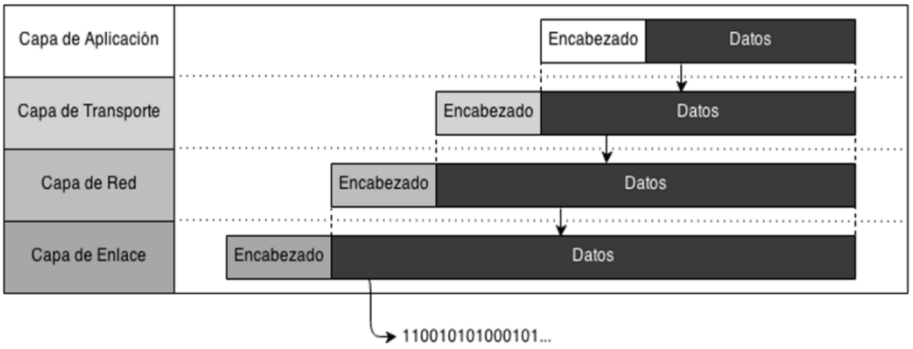


Figura 2.7 - Encapsulado TCP/IP.

La Fig. 2.8 presenta una comunicación indirecta entre dos sistemas, con arquitectura TCP/IP, a través de un *router* intermedio. Como en el caso del Modelo OSI, la comunicación horizontal entre los dos niveles más bajos, se realiza entre dispositivos comunicados directamente. Por su parte, los niveles de transporte y aplicación, establecen una comunicación horizontal entre extremos finales.

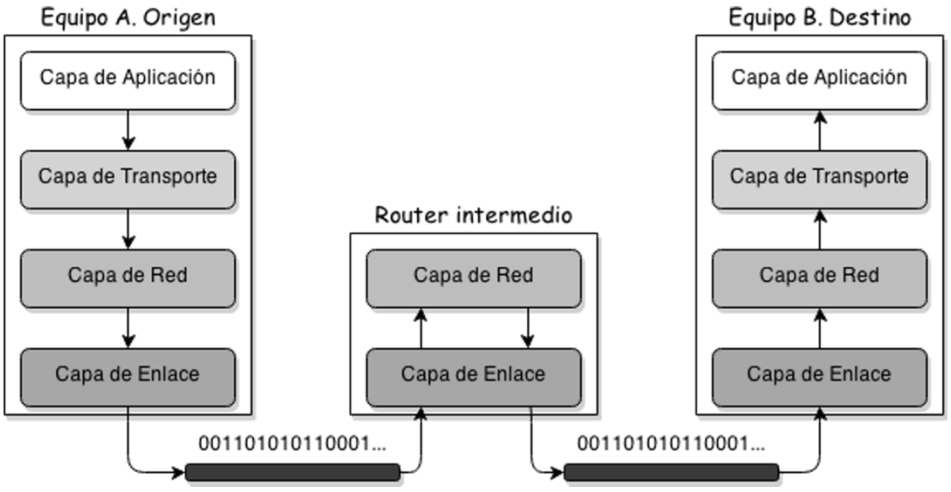


Figura 2.8 - Comunicación TCP/IP con *router* intermedio.

El software TCP/IP está integrado en la mayoría de los sistemas operativos actuales, tales como Unix, Linux, Windows, y Mac OS. En términos generales, se podría ubicar ambos protocolos, TCP e IP, como parte del sistema operativo, situándose el nivel de aplicación por encima, en el entorno de usuario, y el de enlace, algunas veces llamado interfaz de red, por debajo, integrado en el hardware. Ciertos parámetros de los protocolos se pueden ajustar de manera diferente según el sistema operativo que los contenga y aún pueden existir diferencias entre versiones del mismo sistema operativo. Esta posibilidad puede abrir la puerta a ciertas vulnerabilidades en términos de seguridad.

2.4 Operación Peer-to- Peer y Operación Cliente/Servidor

Generalmente se estudia detalladamente TCP/IP en término de sus protocolos porque dicha visión permite entender cómo trabaja la arquitectura. En el apartado anterior se han presentado los distintos niveles de la arquitectura TCP/IP y su equivalente en términos del modelo OSI. En ambos esquemas, la posibilidad de mover datos en una red abre la puerta a una variada oferta de servicios, provistos tanto a otros protocolos como a los propios usuarios finales.

Los protocolos más importantes de la arquitectura TCP/IP permiten desarrollar toda la funcionalidad necesaria para la inter-conectividad de redes, capacidad utilizada por los niveles superiores para ofrecer servicios a los usuarios finales. Se trata de las aplicaciones que tanto éxito han tenido en Internet.

Las redes se conectan a Internet para poder comunicarse con otras redes y compartir recursos. Esta última posibilidad precisa del diseño de aplicaciones de red, planteadas para realizar una gran variedad de tareas reconocidas genéricamente bajo la forma de ofrecer servicios. El diseño, en términos generales, puede estructurarse según dos modelos: cliente/servidor y *peer-to-*

peer. El modelo particular precisará de cierta responsabilidad asignada a cada dispositivo al momento de compartir recursos. En algunas redes, todos los dispositivos se comportan de la misma manera, mientras que en otras, cada uno tiene asignado un trabajo particular al momento de proveer servicios.

Aquellas redes configuradas con dispositivos pares, donde cada uno posee recursos que comparte con el resto, utilizando *software* de similares características, se denominan redes *peer-to-peer* (P2P) o entre pares. Las redes P2P permiten el intercambio directo de información entre los dispositivos interconectados, sin la coordinación de un servidor central. Se trata de aplicaciones distribuidas, donde cada actor trabaja como cliente o servidor, siendo capaz de re-enviar requerimientos. Algunas de las utilizaciones más difundidas de las redes P2P sirven para intercambiar archivos, desplegar juegos en red y actuar como base para la nueva tecnología Voz sobre IP (VoIP, Voice over IP). Por ejemplo, en Skype se considera un sistema P2P para conectar a todos los nodos en una red de manera dinámica para participar en el enrutamiento de tráfico, el procesamiento y las tareas intensivas de ancho de banda que, de otro modo, dependerían de la administración de servidores centrales.

Existen otras redes, que se definen dentro de un modelo conocido como cliente/servidor. Por diseño, muchos servicios típicos de Internet, se implementan en dos aspectos bien definidos: un número pequeño de máquinas muy poderosas, denominadas servidores, que proveen información, y un número mucho más importante de clientes, que la requieren. La arquitectura cliente/servidor puede ser pensada como un conjunto de computadoras conectadas por medio de una red de comunicaciones, pero con diferentes roles asignados. El modelo fue tenido en cuenta tanto en el diseño de protocolos TCP/IP como en de las aplicaciones apoyadas sobre los mismos. A modo de ejemplo, en el caso de la aplicación más popular de Internet, la *World Wide Web*, el navegador es en realidad un cliente HTTP con capacidad para iniciar requerimientos de algún recurso a un servidor HTTP, alojado en algún sitio Web. La red es el vínculo entre ambos. En términos generales, el servidor estará capacitado para atender muchos clientes al mismo tiempo.

En términos de *hardware* de red, un cliente debe entenderse como un computador utilizado por un usuario, que inicia requerimientos, enviando peticiones que viajan por la red. El servidor que responde a estos clientes es una máquina de gran potencia en términos de procesamiento, generalmente funcionando en lugares físicos protegidos, bajo la responsabilidad de un administrador. En términos de *software*, los conceptos cliente y servidor se aplican a los programas que se alojan en las máquinas cliente y servidor. Puede suceder que una máquina posea los dos tipos de aplicaciones, tanto cliente como servidor. Por otra parte, en términos de una transacción, se denomina cliente al que la inicia y servidor al que responde, enviando la información.

Es decir que, según el contexto, utilizaremos estas palabras con diferentes significados.

Las redes del tipo cliente/servidor proveen ventajas en cuestiones de escalabilidad y confiabilidad. Las redes P2P cuentan con la ventaja de su simplicidad y bajo costo. Las redes cliente/servidor son más confiables y responden mejor en cuanto a la escalabilidad, aunque su configuración es más

compleja a la hora del montaje. Estas características permitieron que las redes cliente/servidor tuvieran una evolución importante en entorno de redes grandes, siendo actualmente el esquema dominante.

Bibliografía

1. Tanenbaum, Andrew S., "Redes de Computadoras", Tercera Edición. Prentice Hall Inc., 1996.
2. Stallings, William, "Comunicaciones y Redes de Computadores". Sexta Edición. Prentice Hall Inc., 2000.
3. Stevens, Richard, "TCP/IP Illustrated, Volume 1: The Protocols", Addison-Wesley, 1994.

Problemas

1. Describa el modelo OSI incluyendo los términos: capa, servicio, *overhead*, ocultamiento, comunicación real, comunicación virtual, PDU, SDU e interfaz.

2. Indique cuáles protocolos de la Arquitectura TCP/IP forman parte del sistema operativo. Distinga las funcionalidades que se implementan en hardware de las que se implementan en software.

3. Indique si las siguientes afirmaciones son Verdaderas o Falsas, justificando su respuesta.

   a) La SDU de nivel (N+1) más la cabecera de nivel N conforman la PDU de la capa N.
   b) La definición de un servicio en el modelo OSI implica la especificación de las funcionalidades que se ofrecen, pero no cómo se hace.
   c) Encapsular significa agregar información de interés para la capa par.
   d) El protocolo TCP es un protocolo *end-to-end*, en tanto que IP es un protocolo de comunicación *hop-by-hop*.
   e) Considerando una PDU de capa de enlace de $500\ bytes$, una PDU de capa de red de $482\ bytes$ y una PDU de capa de transporte de $462\ bytes$. Considerando un encabezado a nivel de transporte de la misma longitud que el encabezado de red, un encabezado de capa de sesión de $8\ bytes$, otro a nivel de capa de presentación de $10\ bytes$ y que a nivel de capa de aplicación no hay encabezado, la proporción de encabezados (*overhead*) sobre la transmisión es de 15.2%.

4. Cite las principales características de una red desarrollada sobre un modelo *peer-to-peer*, comparando las prestaciones respecto del modelo cliente/servidor.

5. Investigue la asociación de los siguientes elementos de red con las capas del Modelo OSI, explicando su función:

   a) Repetidor o *Hub*.
   b) Puente o *Bridge*.
   c) Conmutador LAN o *Switch*.
   d) Dispositivo de encaminamiento o *Router*

CAPÍTULO III

Técnicas de Conmutación en Redes WAN

En los sistemas de comunicaciones, el multiplexado de varios canales sobre un único medio de transmisión mediante dispositivos especiales llamados multiplexores, es la combinación de dichos canales de entrada sobre una única línea de salida. El proceso inverso se denomina de-multiplexado de señales.

Existen varias técnicas de multiplexado, cada una con sus propias características y entorno de aplicación. Algunas de ellas son de uso extendido en las redes de telefonía y han dado lugar a diferentes estándares, conocidos como jerarquías en las redes de conmutación de telefonía digital: PDH (en dos versiones, europea y americana) y SONET (para mayor velocidad en redes ópticas sincrónicas).

Una variante asíncrona de los esquemas mencionados, adaptable a tráficos de naturaleza de ráfaga, se utiliza en conmutadores de redes de paquetes.

Este capítulo pretende dejar una visión conceptual de la tecnología que subyace detrás de ambos tipos de redes, de conmutación de circuitos y de conmutación de paquetes, ésta última en sus dos versiones: circuitos virtuales y datagramas.

El capítulo finaliza con una mirada sobre una de las funcionalidades más importantes en las redes de paquetes, el enrutamiento, sus diversos tipos y algunas de las características más sobresalientes de la evolución de las arquitecturas de los routers.

3.1 Multiplexado de Señales

En entornos de comunicación de redes, las técnicas de multiplexado permiten combinar varias señales, sean estas analógicas o digitales, en una señal o único flujo sobre un medio de comunicación. El objetivo es compartir el medio o canal para abaratar los costos asociados a la transmisión.

La primera red en la que se usaron técnicas de multiplexado fue la red de telefonía fija, logrando transportar sobre un mismo cable varias llamadas telefónicas. En esta red, la línea del usuario llega a la central de conmutación local sobre un par telefónico y es multiplexada junto con otras líneas telefónicas de su misma área. La señal multiplexada resultante se transporta hacia otra oficina de conmutación, reduciéndose de este modo sustancialmente la cantidad de cables necesarios y aumentando la distancia a la que puede llegar la línea del abonado.

Un dispositivo capaz de desarrollar técnicas de multiplexado se denomina multiplexor, abreviadamente MUX. El proceso inverso, denominado de-multiplexado, lo realiza un dispositivo llamado de-multiplexor o DEMUX.

Existen varios tipos de multiplexado. Entre los más usados, podemos mencionar: Multiplexado por División en el Espacio (SDM, Space Division Multiplexing), Multiplexado por División en Frecuencia (FDM, Frequency Division Multiplexing) y Multiplexado por División en el Tiempo (TDM, Time Division Multiplexing). Existe una variante de TDM, denominado TDM estadístico o asincrónico, que puede acomodar señales digitales de velocidad variable sobre un único canal, adaptándose muy bien al tráfico de ráfagas característico de las redes de datos.

3.1.1 Multiplexado por División en el Espacio, SDM

En redes cableadas, no se puede considerar SDM como un método de multiplexado ya que la técnica implica usar diferentes cables para cada canal. Un ejemplo de SDM en la red de telefonía fija es la forma de llegada de los pares de abonados a la central telefónica local: cada abonado dispone de un par en ese enlace punto a punto.

En cambio, en comunicaciones inalámbricas, SDM se puede desarrollar por medio de múltiples elementos radiantes que forman una antena de arreglo de fase. Un arreglo de fase es un arreglo de antenas en el que las fases relativas de las señales que alimentan las antenas, se varían de tal modo que el patrón de radiación presenta lóbulos reforzados o suprimidos en alguna dirección deseada o no deseada, respectivamente.

Un ejemplo moderno de SDM inalámbrico se presenta en la técnica de Entrada Múltiple Salida Múltiple (MIMO, Multiple Input Multiple Output), que introdujo mejoras en estándares para redes inalámbricas, estimándose que también será de aplicación en las redes 5G para telefonía celular. Se denomina MIMO al esquema de comunicaciones que usa múltiples antenas, tanto del lado transmisor, como del lado receptor. La base de división en el espacio consiste en

dividir una señal de gran velocidad en otras de menor velocidad, cada una de estas últimas transmitidas desde una antena diferente sobre el mismo canal, en la misma frecuencia. Si cada una de ellas arriba al receptor de manera que sea posible identificarlas claramente, y si el receptor posee suficiente información sobre el estado del canal, se las puede separar en canales paralelos. La identificación es posible por el aprovechamiento de la naturaleza de caminos múltiples propia de este tipo de canales inalámbricos.

3.1.2 Multiplexado por División en Frecuencia, FDM

El multiplexado por división en frecuencia FDM consiste en la combinación de varias señales sobre un único medio, de tal manera que cada una de estas señales va montada sobre su propia portadora, ocupando su propio canal. Se trata de una técnica de multiplexado analógica.

El canal de salida debe contar con una capacidad superior a la suma de los anchos de banda ocupados por las señales de entrada. Las frecuencias portadoras deben quedar lo suficientemente separadas como para evitar el solapamiento de los espectros de las señales transportadas. En este sentido, deben preverse regiones de guardas de seguridad entre canales, para evitar interferencias, tal como se observa en la Fig. 3.1.

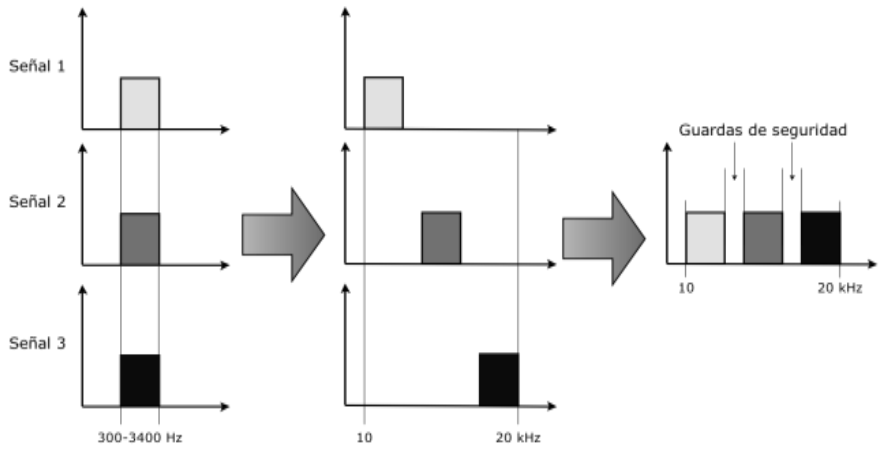


Figura 3.1 - Multiplexado por división en frecuencia FDM.

Por ejemplo, la señal de televisión por cable se entrega al usuario utilizando esta técnica. Sobre el cable coaxial se transmiten varios canales de televisión de manera simultánea sin que interfieran entre sí, permitiendo que el usuario, al sintonizar un canal, acceda a la señal deseada. Otros ejemplos de utilización de FDM como técnica para compartir un medio de transmisión se

encuentran en la comunicación por radio en AM y FM, en ciertos segmentos de la red original de telefonía y en la primera generación de telefonía celular.

Existe una variante de esta técnica, utilizada en redes ópticas, conocida como Multiplexado por División de Longitud de Onda (WDM, Wavelength Division Multiplexing). WDM involucra el multiplexado de cierto número de señales ópticas portadoras, cada una con su propia longitud de onda, sobre una única fibra óptica. Esto permite mayor eficiencia y también transmisión *full duplex*.

Un esquema muy popular en comunicaciones digitales de banda ancha es la versión Ortogonal de FDM (OFDM, Orthogonal Frequency Division Multiplexing). OFDM usa un método de modulación digital de muchas subportadoras muy próximas para transportar datos sobre varios canales paralelos. Sobre cada canal se usan esquemas de modulación convencionales, tales como Modulación Cuadratura de Amplitud (QAM, Quadrature Amplitude Modulation) o Modulación por Corrimiento de Fase (PSK, Phase Shift Keying), a baja velocidad, manteniendo la velocidad total similar a la que se lograría con esquemas convencionales de portadora única en el mismo ancho de banda. La restricción con respecto a FDM es que las subportadoras deben ser ortogonales. Esta relación entre frecuencias subportadoras implica que, aunque los espectros se solapen, pueden ser recibidos sin interferencias, ya que el espaciamiento entre subportadoras se mantiene igual a la inversa de la duración de un símbolo.

Del lado receptor, un banco de demoduladores traslada cada subportadora a banda base, para luego obtener el dato por filtrado de la señal sobre el período de un símbolo. Para cada subportadora, la contribución de las demás es nula ya que el espaciamiento entre ellas es la recíproca de la duración de un símbolo, presentando un número entero de ciclos durante el período de integración. De este modo se evita la interferencia por solapamiento.

Entre las ventajas de OFDM se pueden mencionar su robustez frente a interferencia co-canal de banda angosta, a la Interferencia entre Símbolos (ISI, Intersymbol Interference) y a los efectos de desvanecimiento, propios de los canales multi-camino. Por este motivo, es un método que se ha impuesto en las redes inalámbricas más modernas.

3.1.3 Multiplexado por División en el Tiempo, TDM.

Se trata de una técnica de multiplexado digital en la que un conjunto de señales comparte un mismo medio físico pero en diferentes instantes de tiempo. En TDM sincrónico o simplemente TDM, el tiempo se considera dividido en ranuras o *slots*, de tiempo fijo, uno para cada señal a ser multiplexada. Durante la ranura 1 se transmite una muestra codificada del canal 1, durante la ranura 2 una muestra de canal 2, y así sucesivamente hasta llegar a la última ranura o canal. Entonces, se recomienza con la primera ranura y se vuelven a barrer todos los canales. Para no perder muestras, la velocidad de barrido debe tener una estricta relación con la frecuencia de muestreo de las señales a transportar. Así, por ejemplo, si las señales son en banda base, limitadas a un ancho de banda de

3 KHz, deberán muestrearse al menos a 6 KHz, según lo establece el Teorema de Muestreo de Nyquist. Un esquema de dos canales para este tipo de señales, requeriría entonces un reloj de 12 KHz.

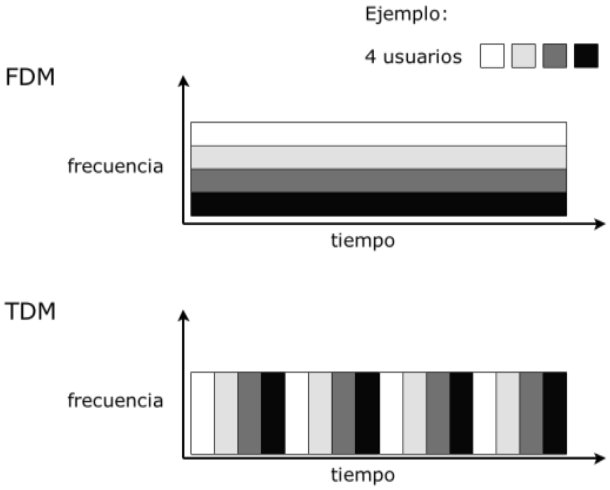


Figura 3.2 - FDM vs TDM

A diferencia de FDM, en TDM cada señal dispone, durante su tiempo de transmisión, del ancho de banda completo del canal de salida. En cambio, en FDM, cada señal dispone todo el tiempo de una porción del ancho de banda del canal de salida. La Fig. 3.2 es una representación gráfica de estos conceptos.

En TDM, sobre la salida física del MUX se conforma una trama. A pesar de llevar un nombre igual que el de los mensajes de Capa de Enlace, no se debe confundir con estos, ya que se trata de una serie de ranuras, una para cada canal, pudiendo transportar bits agregados con fines de sincronismo y control de errores. La Fig. 3.3 presenta una visión simplificada de la confluencia de ranuras provenientes de distintos canales, en una única trama a la salida del multiplexor.

En la entrega, el ordenamiento de los canales en la trama asegurará el reparto correspondiente, por medio de un elemento de-multiplexor. Para que el esquema funcione correctamente, es de fundamental importancia del mantenimiento del sincronismo entre ambos elementos. Muchos esquemas de este tipo utilizan códigos, previamente acordados, que se transmiten periódicamente, justamente para asegurar el mantenimiento del sincronismo entre ambos extremos.

Figura 3.3 - TDM sincrónico

A modo de ejemplo, imagine la posibilidad de transmitir, por medio de este esquema, treinta canales de voz sobre un mismo canal de salida. La señal de voz posee contenido espectral entre $300\,Hz$ y $3400\,Hz$. Al considerar su digitalización, se adopta un ancho de banda de $4\,kHz$ y, por lo tanto, una frecuencia de muestreo f_s de $8\,kHz$. Para la calidad pretendida en una transmisión telefónica, es suficiente una codificación de $8\,bits$ por muestra, por lo que cada canal de voz se convertirá en una señal digital de velocidad binaria $r_b = 64\,kbps$. A la salida del MUX, el canal de alta velocidad, se podría dividir en treinta ranuras de tiempo consecutivas, desde la correspondiente al canal 1 hasta la que se corresponde con el canal 30, conformando un conjunto de $240\,bits$. En un sistema real de multiplexado de 30 canales de voz, se transmiten dos canales más para sincronismo y señalización, resultando un conjunto de $256\,bits$, tal como se presenta en la Fig. 3.4.

La sucesión mencionada es lo que se conoce como trama, de repetición periódica cada $125\,\mu seg$, que sería el período de muestreo de las señales de voz. La velocidad a la salida del MUX es $64\frac{kbps}{canal} x\ 32\ canales = 256\frac{bits}{trama} \times 8\,kbps = 2048\,kbps$, o sea la suma de las velocidades de entrada. Este ejemplo se corresponde básicamente con la primera etapa de la jerarquía digital sincrónica, conocida con el nombre de Jerarquía Digital Plesiócrona (PDH, Plesiochronous Digital Hierarchy), que se usa en Argentina y Europa en la red de telefonía fija.

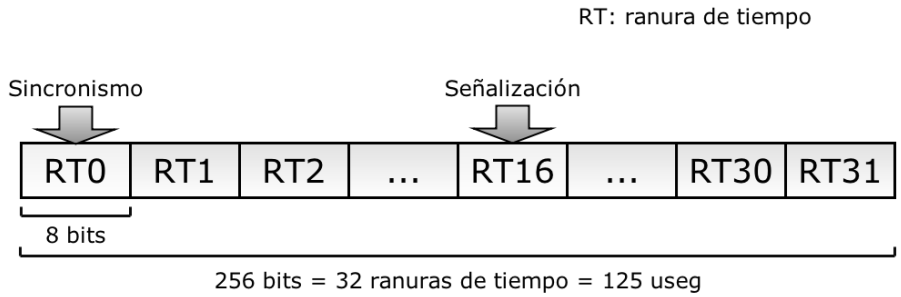


Figura 3.4 - Trama E1, 32 ranuras de tiempo.

Con la técnica de multiplexado de TDM, también es posible transportar señales de diferente velocidad en una única trama. Supóngase, por ejemplo, tres señales de $8\,kbps$,$16\,kbps$ y $24\,kbps$, respectivamente. Para su multiplexado sobre una señal de alta velocidad, la velocidad a la salida del MUX, debe ser $48\,kbps$, la suma de las velocidades de las tres fuentes de entrada. Pero, en este caso, para poder cumplir con el transporte correcto de todas las señales, no basta

con asignar a cada una de ellas una única ranura de tiempo. Para determinar el número de ranuras correcto para cada fuente, se debe reducir la relación de velocidades 8: 16: 24 a la forma más baja posible, en este caso 1: 2: 3. La suma de la relación reducida es 6, que representa la longitud mínima del ciclo repetitivo de asignaciones de ranuras en el multiplexor. Así, la trama a la salida del MUX, dentro de cada ciclo de 6 $ranuras$, se armará por asignación de 1 $ranura$ a la fuente de 8 $kbps$, 2 $ranuras$ a la de 16 $kbps$ y 3 $ranuras$ a la de 24 $kbps$, tal como se presenta en la Fig. 3.5.

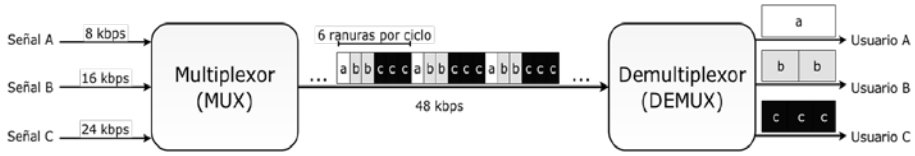


Figura 3.5 - TDM señales de distinta velocidad

Con esta técnica sincrónica, no es preciso un protocolo del nivel de enlace de datos, pero sí, como se ha observado, un método para mantener sincronismo MUX-DEMUX. La trama que viaja sobre el canal de alta velocidad, generalmente lleva un preámbulo por este motivo. Se trata de un patrón pre-especificado de bits.

TDM sincrónico es un esquema eficiente en el caso de fuentes predecibles, de tráfico pesado, de tal manera que las ranuras de tiempo estén siempre ocupadas. A continuación, se verán algunos ejemplos de jerarquías TDM sincrónicas cuya utilización es apta para redes de telefonía: PDH y SDH. Luego se presentará una variante TDM que se considera apropiada para el transporte de señales de datos.

3.1.4 Jerarquías TDM sincrónicas: PDH y SDH.
PDH

La Jerarquía Digital Plesiócrona (o casi sincrónica) es una tecnología de transmisión para el transporte de grandes volúmenes de datos sobre redes digitales tipo WAN. Su diseño permite la comunicación MUX-DEMUX sin necesidad de relojes perfectamente sincronizados.

Dado que las señales que se transmiten pueden correrse en el tiempo al llegar al receptor, por diferencias pequeñas entre los relojes, se deben utilizar técnicas de compensación. Con este propósito, PDH usa una técnica denominada agregado de bits o *bit stuff* : se insertan bits que no son de información, entre los datos, con el propósito de que los flujos de datos que no tienen exactamente la misma velocidad o cuyas velocidades no se hallan relacionadas racionalmente, se puedan llevar a una velocidad común. La localización de estos bits es informada

al extremo receptor, que los removerá, retornando los flujos a su velocidad original. De este modo se logra la recuperación exacta de los datos enviados.

La tecnología PDH permite el multiplexado de grupos de flujos de datos o *streams* a nivel binario, tomando un bit del primer flujo, a continuación un bit del segundo, y así sucesivamente. Los flujos pueden ser de diferente velocidad, pudiendo ocurrir que, al muestrear el siguiente bit, éste todavía no haya arribado. En este caso, se aplica la técnica mencionada de *bit stuff*, para mantener la velocidad a la salida.

La jerarquía PDH se basa en canales de $64\ kbps$. En cada nivel de multiplexado se aumenta el número de canales sobre el medio físico. Existen tres jerarquías PDH: la europea, la norteamericana y la japonesa. La primera usa la trama descrita en la norma G.732 de la ITU-T mientras que las jerarquías norteamericana utilizan la norma G.733. Por tratarse de tramas de diferente formato, para poder unir dos enlaces que utilicen diferente norma debe desarrollarse algún tipo de adaptación. En cualquier caso, siempre se convertirá la trama al formato usado por la jerarquía europea.

La jerarquía latinoamericana es la misma que la europea, que agrupa $30 + 2$ canales de $64Kbps$, para generar una trama que se conoce como $E1$, de $64\ \frac{kbps}{canal} x\ 32\ canales\ = 2048\ kbps$. Luego, con el multiplexado de 4 tramas $E1$, se consigue generar una trama $E2$, de $2048\ \frac{kbps}{canal} x\ 4\ canales\ = 8192\ kbps$. A su vez, cuatro tramas $E2$ conforman una trama $E3$ de $32.768\ Mbps$ y cuatro tramas $E3$ constituyen una trama $E4$ de $131.072\ Mbps$. En el último escalón la señal alcanza $524.288\ Mbps$. La Fig. 3.6 presenta un bosquejo de esta jerarquía de multiplexado, con su equivalente de canales de voz en cada nivel. Las diferencias numéricas de velocidad respecto a lo calculado previamente se deben al agregado de bits de control para funciones de sincronismo y administración.

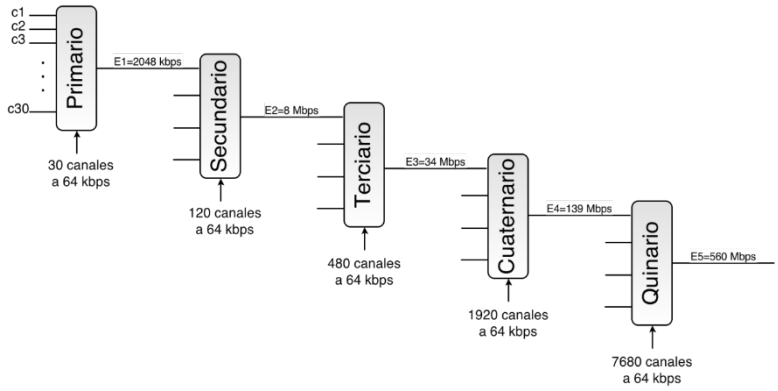


Figura 3.6 - Jerarquía Digital PDH.

La Fig. 3.7 presenta un detalle del armado de la trama de la jerarquía europea $E1$, de $2.048\ Mbps$. La especificación G.732 define un conjunto de $32\ ranuras$ de tiempo, de $8\ bits$ cada una, a una velocidad de $64\ kbps$. Tanto la ranura 0 como la ranura 16, se reservan para administración y señalización del

canal. Cada trama tiene una duración de $125\ \mu s$. La ley de *companding* utilizada es la ley A, especificada en la Recomendación G.711, con 256 niveles de cuantificación y una tasa de muestreo de $8000\ \frac{muestras}{seg}$.

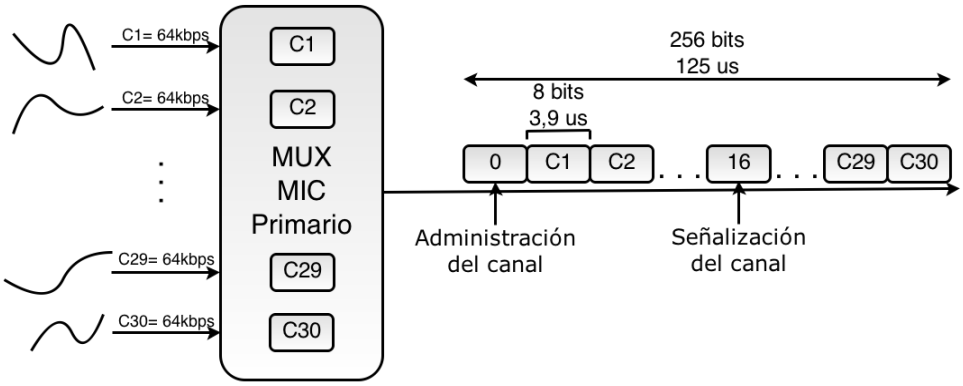


Figura 3.7 - Detalle E1.

Se destaca la existencia de dos portadoras, con diferente nominación, que utilizan PDH: $E1$ y $T1$. La segunda es la denominación norteamericana para un sistema similar de multiplexado, pero de 24 canales de voz en su primer nivel. La Tabla 3.1 presenta una comparación entre ambas jerarquías.

Aplicar cualquiera de estos esquemas a una red de telefonía supone un ingreso a una red con topología estrella, donde cada usuario se comunica con conexiones punto a punto con un elemento central, el correspondiente a su enlace local. A su vez, estos elementos se comunican con otros similares, a través de troncales PDH, en una estructura recursiva que conforma la jerarquía descripta. En la columna vertebral de esta estructura los enlaces pueden ser punto a punto, o conformar un anillo para proveer conexión redundante entre nodos.

En cualquiera de estos esquemas de multiplexado, el sincronismo depende de cada nivel de la jerarquía, restringiendo la identificación de una señal de menor orden, dentro de un flujo de mayor velocidad, al de-multiplexado final en la jerarquía. Por ejemplo, una tributaria de $140\ Mbps$ queda conformada por 64 señales independientes $E1$ de $2\ Mbps$, lo cual se traduce en $4 + 16 + 64 = 84$ circuitos de multiplexado, correspondientes a los respectivos niveles $E2 = 4E1$, $E3 = 16E1$ y $E4 = 64E1$. Para de-multiplexado de una señal de $140\ Mbps$ en señales $E1$ de $2\ Mbps$, se precisa esa misma cantidad de circuitos de de-multiplexado y sincronismo.

Esta estructura tan rígida, difícil de escalar, con capacidad limitada de administración y sin interconexión posible a nivel óptico, ha sido reemplazada por SONET y esquemas SDH que incluso permiten alcanzar mayores velocidades.

Tabla 3.1 - Comparación AT&T vs CCIT/PDH

	AT&T (Estados Unidos)		**CCIT / PDH** (Argentina)	
	Nº de Entradas	**Velocidad de Salida Mbps**	**Nº de Entradas**	**Velocidad de Salida Mbps**
Primer Nivel	24	1.544 – T1	30	2.048 –E1
Segundo Nivel	4	6.312 – T2	4	8.448 – E2
Tercer Nivel	7	44.736 – T3	4	34.368 – E3
Cuarto Nivel	6	274.176 – T4	4	139.264 – E4

SDH

Para superar las limitaciones de PDH, se desarrolló la Red Óptica Sincrónica (SONET, Synchronous Optical Networking), que sirvió como plataforma de diseño para la red universal de Jerarquía Digital Sincrónica (SDH, Synchronous Digital Hierarchy). SONET se ha desarrollado en EE.UU, Canadá, Corea, Taiwan y Hong Kong, con estándares definidos por el ANSI, en tanto que SDH se utiliza en el resto del mundo con estándares definidos por la ITU-T. Ambos esquemas se utilizan ampliamente en las transmisiones de compañías de telefonía, por su capacidad de manejo de múltiples canales telefónicos. Por ejemplo, la trama básica SDH STM-1 proporciona una capacidad de transmisión de 1.890 líneas telefónicas.

El propósito principal del desarrollo de estas nuevas tecnologías fue definir jerarquías para multiplexado de señales sobre un canal de fibra óptica, desarrolladas en reemplazo de PDH para evitar las limitaciones mencionadas en cuanto al sincronismo. Inicialmente, por cuestiones de convivencia con PDH, estas jerarquías se idearon para transportar de manera encapsulada señales tipo *T*1 o *T*3, codificadas en tiempo real en Modulación por Codificación de Pulsos (PCM, Pulse Code Modulation), no comprimidas.

SDH es la tecnología dominante en la capa física de transporte de voz en las redes ópticas actuales, permitiendo además el transporte de datos y de datos encapsulados en IP o en el Modo Asincrónico de Transferencia (ATM, Asynchronous Transfer Mode), y actuando como contenedor físico de mensajes de los niveles dos a cuatro del modelo OSI.

En SDH, la trama básica, que se conoce como Módulo de Transporte Sincrónico (STM-1, Synchronous Transport Module), opera a $155.520\ Mbps$. Se define también una estructura de multiplexado, mediante la cual una señal STM-1 puede transportar señales de menor velocidad de transmisión, formando parte de su carga útil.

Debido a que SDH no nace para sustituir a PDH, sino para ser usado conjuntamente como medio de transporte en los enlaces que requieren mayor capacidad, se previó una forma estándar para transportar tramas PDH dentro de tramas SDH (hasta 3 E3 en una STM-1). Pero, a diferencia de PDH, en SDH las señales se tratan en el multiplexor byte a byte, de manera sincrónica, extrayéndose las señales de sincronismo de una referencia común. Por este motivo es posible acceder de forma directa y simple a las señales multiplexadas, simplificando los problemas relativos a la vieja estructura de cobre.

Como se ha observado, en SDH cada trama se encapsula en una estructura denominada contenedor, que posee encabezados y campos de carga útil. Dicha estructura se puede asimilar a una distribución de bytes de 9 filas por 270 columnas: $9\ x\ 270\ =\ 2.430\ bytes\ =\ 19.440\ bits$. De este modo, todos los bytes de una columna pertenecen a una misma señal tributaria, siendo la carga útil transmitida: $9\ x\ 261\ =\ 2.249\ bytes\ =\ 18.792\ bits$. En el encabezado, que ocupa las primeras 9 columnas ($81\ bytes$), las primeras tres filas y las últimas cinco filas llevan información de control para las secciones de regeneración y secciones de multiplexado, respectivamente. Se trata de información para monitorear cuestiones de calidad, detectar fallas y gestionar alarmas, entre otras funciones. La fila intermedia es un puntero que apunta al comienzo de las señales tributarias. Estas tramas se transmiten a una velocidad de $8000\ \frac{tramas}{seg}\ =\ 8000\ x\ 19.440\ \frac{bits}{trama}\frac{tramas}{seg} = 155.52\ Mbps$.

Los niveles de jerarquía superior se forman por multiplexado a nivel de byte de varias estructuras STM-1 con una referencia de reloj común. Los niveles superiores se denominan STM-N, siendo N el nivel ocupado en la jerarquía. Actualmente están definidos para $N = 4$, $N = 16$, $N = 64$ y $N = 256$. La trama STM-N contiene $9 \times 270 \times N\ bytes$ y también su duración es de $125\ \mu s$.

En el caso de SONET, se hace referencia a la misma señal como portadora óptica (OC, Optical Carrier) o portadora sincrónica (STS, Synchronous Transport Signal), según que la señal sea eléctrica u óptica. La Tabla 3.2 presenta estos tipos y su equivalente con SDH. STM-0 no representa un nivel válido de SDH, se lo considera un método de transmisión a baja velocidad, para enlaces de radio y satélite. Como se observa en la Tabla, en el último escalón de la jerarquía se pueden alcanzar velocidades en el orden de $40\ Gbps$.

Tabla 3.2 - Jerarquías SDH/SONET

SDH	**SONET**	**Velocidad , Mbps**
STM-0	OC-1 / STS-1	51.84
STM-1	OC-3 / STS-3	155.52
STM-4	OC-12 / STS-12	622.08

STM-16	OC-48 / STS-48	2488.32
STM-64	OC-192 / STS-192	9953.28
STM-256	OC-768 / STS-768	39812.12

SONET también ofrece una trama básica adicional, STS-1 u OC-1, que opera a $51.84\ Mbps$, es decir a un tercio de la velocidad de STM-1. En STS-1, la trama es de $810\ bytes$, asimilable a una matriz de 9 filas por 90 columnas, tal como se presenta en la Fig. 3.8. Esta trama se transmite como una sucesión de $3\ bytes$ de encabezado con $87\ bytes$ de carga de datos, repetida nueve veces, con una duración total de $125\ \mu s$.

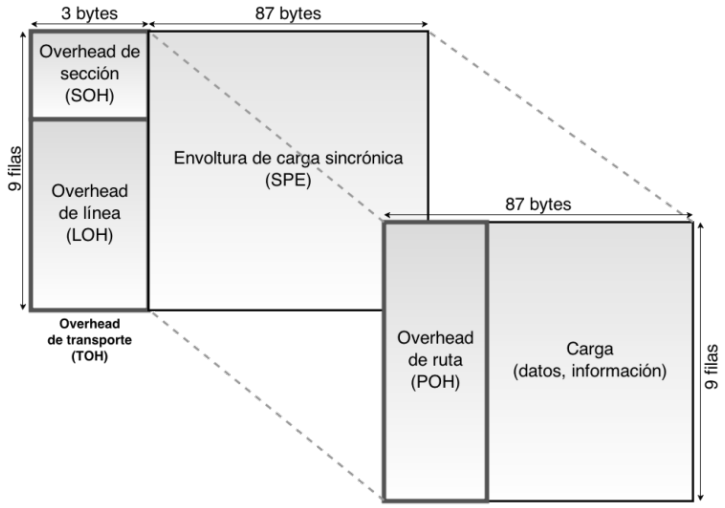


Figura 3.8 - Formato general de la trama SONET STS-1.

La trama posee dos secciones principales y tres niveles de encabezado, designados en la figura como *Overhead,* en una disposición similar a la de la trama STM-1. La sección de encabezado de transporte se divide en una parte para el nivel de sección y otra para nivel de línea. La parte de datos se denomina envoltura de carga sincrónica y contiene un nivel de encabezado de ruta, aparte de la propia carga.

Es la propia estructura de la red la que se relaciona con la información de control en los encabezados citados, tal como se muestra en la Fig. 3.9. Las señales electrónicas alimentan un multiplexor de inserción-extracción de origen (ADM, Add-Drop Multiplexer), donde se combinan en una única señal óptica. Este multiplexor proporciona la interfaz entre una red tributaria eléctrica y la red óptica. La señal óptica es transmitida hasta un repetidor o regenerador, que toma la señal óptica, la transforma en eléctrica, regenerándola para eliminar el ruido

que la ha contaminado en el trayecto, y la vuelve a modular en señal óptica. El regenerador SDH sustituye parte de la información de cabecera existente por información nueva.

La señal regenerada llega al multiplexor de inserción/extracción ADM. El ADM puede insertar señales que llegan de distintas fuentes en una ruta dada o extraer una señal de una ruta y redirigirla a otra sin necesidad de de-multiplexar toda la señal. El ADM usa la información de cabecera, tales como direcciones y punteros, para identificar los flujos individuales. Al final de la trayectoria se precisan terminadores apropiados para convertir la señal a un formato utilizable por los equipos receptores.

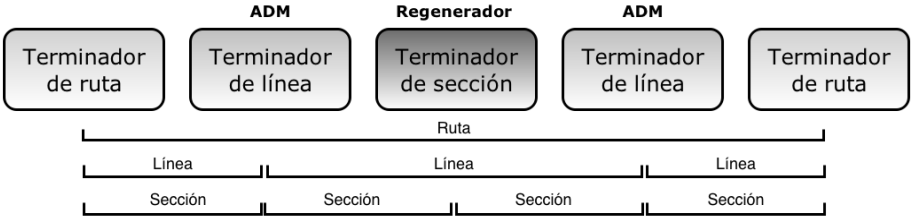


Figura 3.9 - Arquitectura SONET.

En esta estructura, una sección es físicamente un cable de fibra o enlace óptico terminado por un ADM o por un regenerador. La principal función de una sección es convertir señales eléctricas en ópticas y dar formato apropiado a las tramas SONET. Por su parte, una línea es la sucesión de una o más secciones y se origina o termina en elementos especiales ADM, que sincronizan y permiten el multiplexado de información de las tramas. La ruta o camino es entre extremos de la red.

Los dispositivos ADM se conectan en redes en anillo, de configuración unidireccional o bidireccional. En cada caso, se pueden añadir anillos extras para que la red pueda realizar auto-diagnóstico, siendo capaz de recuperarse por sí misma en el caso de fallas. Para servicios en redes WAN, las redes SDH actuales usan una combinación de anillos interconectados. Por ejemplo, una red SDH puede tener un anillo regional, varios anillos locales y muchos anillos de sitio para ofrecer servicio de área extensa.

3.1.5 TDM Asincrónico – Multiplexado por División en el Tiempo Estadístico.

Como se ha observado, el entramado en TDM sincrónico es necesario por cuestiones de sincronismo y administración de la red. Cada comunicación

queda identificada por la posición de su ranura de tiempo dentro de la trama. La información dentro de cada ranura no lleva encabezado.

Esta estructura rígida de TDM sincrónico, se traduce en una desventaja al momento de transmitir datos que provienen de equipos informáticos, debido a que esta correlación entre fuente y ranura de tiempo no es óptima desde el punto de vista de la naturaleza de este tipo de tráfico, caracterizada por ráfagas. Desde este punto de vista, si se utilizara multiplexado sincrónico para transporte de datos, muy probablemente en cada trama existirían ranuras vacías, con el consiguiente gasto innecesario de ancho de banda.

La naturaleza estadística del tráfico de datos, llevó a desarrollar otro tipo de técnica de multiplexado, cuya velocidad a la salida del multiplexor se ajustara a un valor promedio de las velocidades de las entradas. Se trata de un esquema que se conoce como TDM asincrónico, estadístico o a demanda.

En este esquema, la cuestión de varios dispositivos tratando de transmitir datos al mismo tiempo en las entradas del multiplexor, se soluciona por medio de memorias temporales de almacenamiento o *buffers*, que guardan los datos hasta que el medio se libere. A este esquema de transmisión se lo conoce como de almacenamiento y re-envío o *store & forward*, siendo típico de las redes de conmutación de paquetes. La situación se grafica en la Fig. 3. 10. La naturaleza estadística del tráfico debería ser tal que ningún canal acaparara el medio pues, de lo contrario, la transmisión se vería afectada por errores ocasionados por saturación en los buffers de entrada.

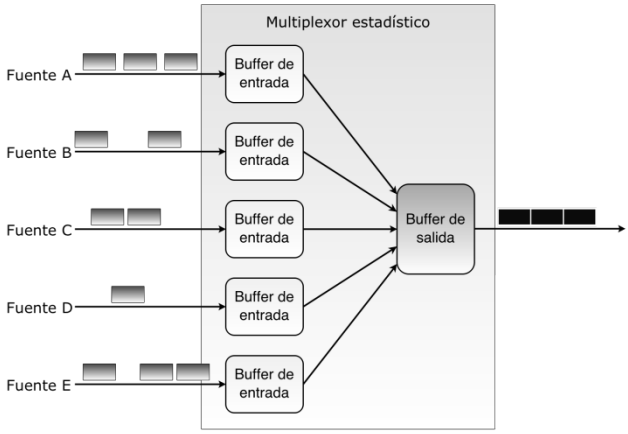


Figura 3.10 - TDM Estadístico, almacenamiento y re-envío.

El multiplexor muestrea los buffers de entrada y arma un paquete por vez con la información de aquellos buffers disponibles para transmitir. De este modo, se pierde la relación biunívoca entre la posición de los datos en la trama y el canal de comunicación de recepción correspondiente, típica de TDM sincrónico. Para salvar este detalle, asegurando una entrega correcta, se impone el agregado de

alguna información adicional bajo la forma de un esquema de direccionamiento, tal como se presenta en la Fig. 3.11.

Por otra parte, en TDM estadístico dejan de existir ranuras vacías en las tramas, pero al introducirse un nivel de almacenamiento o *buffering*, se condiciona el retardo en la entrega. También, el agregado de información para identificación de fuente y destino convierte al sistema en no transparente con respecto a los protocolos. Aún así, estas características de la transmisión permiten que la velocidad de la línea multiplexada, a la salida, pueda ser menor que la suma de las velocidades de las entradas. Evidentemente, el funcionamiento correcto dependerá de la distribución de la carga de transporte, debiéndose contemplar sobre todo las situaciones de demanda pico, siendo éste uno de los motivos principales del agregado de buffers en la estructura de entrada del multiplexor, ya que permiten el almacenamiento temporal de los datos en exceso. Esta forma de procesamiento constituye la base de los esquemas de conmutación de paquetes.

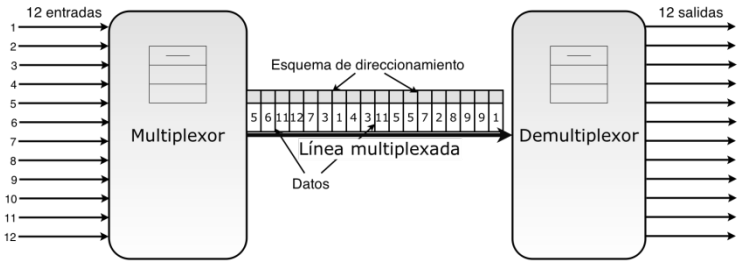


Figura 3.11 - TDM Estadístico, información adicional de direccionamiento.

Por supuesto que existirá un compromiso entre la capacidad de memoria utilizada y la velocidad de la línea. A menor cantidad de memoria, menor retardo de procesamiento y mayor podrá ser la velocidad de conmutación. Siempre la reducción de memoria implicará aumento de velocidad, del mismo modo que el aumento de la cantidad de memoria, conllevará un aumento del retardo.

Por estos motivos, es interesante analizar el comportamiento de este tipo de multiplexado bajo diferentes condiciones de carga.

Primero se definirán algunos parámetros que caracterizan el comportamiento sistema. Se conoce como I al número de fuentes de entrada al multiplexor, y R a la velocidad de cada fuente en bps. La entrada máxima total será el producto de ambas $I \times R$. Si M es la capacidad máxima de salida del multiplexor, medida en bps, el sistema se diseña para que esta capacidad sea menor que la suma de las capacidades de entrada:

$$M < I \times R \tag{3.1}$$

Por su parte, el tiempo que se emplea en transmitir un bit, dependiente de la velocidad de salida, se denomina tiempo de servicio $T_s = {}^{1}/_{M}$.

La capacidad de compresión del multiplexor se designa como el parámetro K, que resulta ser la relación entre la capacidad máxima a la salida y la velocidad de entrada máxima:

$$K = \frac{M}{I \times R} < 1. \tag{3.2}$$

En TDM sincrónico, se cumpliría la relación $K = 1$, dado que la condición de funcionamiento exige que la velocidad a la salida sea equivalente a la suma de las velocidades de entrada. En TDM estadístico, un valor de $K = 0.5$, se podría interpretar como el de un sistema que maneja el doble de dispositivos de entrada comparado con el caso sincrónico.

Si, al mismo tiempo, se caracteriza cada fuente de entrada con un número común, normalizado α, con el que se propone representar la fracción promedio de tiempo de transmisión, resulta ser:

$$0 < \alpha < 1 \tag{3.3}$$

La situación de funcionamiento que verificase la condición $K < \alpha$, implicaría que la entrada excede la capacidad del multiplexor.

Si se asimila el funcionamiento del multiplexor con el modelo de teoría de colas que rige para la atención cliente/servidor, el retardo de atención del servicio o procesamiento se puede considerar dividido en dos componentes: el tiempo en espera en cola y el propio tiempo de servicio. Este retardo depende de características propias del diseño del servidor, en este caso el multiplexor, y del patrón de tráfico que caracteriza la transacción, dependiente de las fuentes de entrada. Si se supone una distribución de llegada aleatoria y un tiempo de servicio constante, se pueden relacionar resultados de la teoría de colas para las expresiones de utilización del multiplexor.

Siendo λ la velocidad de llegada promedio de tráfico total al MUX, medida en bps, resulta:

$$\lambda = \alpha \times I \times R \tag{3.4}$$

Se designa como ρ a la utilización, definida como la fracción de la capacidad total utilizada:

$$\rho = \lambda \times T_s = \frac{\alpha \times I \times R}{M} = \frac{\alpha}{K} = \frac{\lambda}{M} \tag{3.5}$$

Se pueden graficar ciertos parámetros para poder comprender la relación de compromiso entre el tiempo de respuesta del sistema y la velocidad de salida del multiplexor. Se asume N como el tamaño de memoria en tramas, en función de la utilización ρ, obtenido de la teoría de colas como:

$$N = \frac{\rho^2}{2(1-\rho)} + \rho \tag{3.6}$$

Por ejemplo, si se eligen tamaños de tramas de 1000 $bits$ y se grafica el número promedio de tramas a almacenar, tal como muestra la Fig. 3.12, se puede observar que a mayor porcentaje de utilización, mayor es la necesidad de tamaño de memoria, porque la situación de competencia entre las líneas de entrada se empeora.

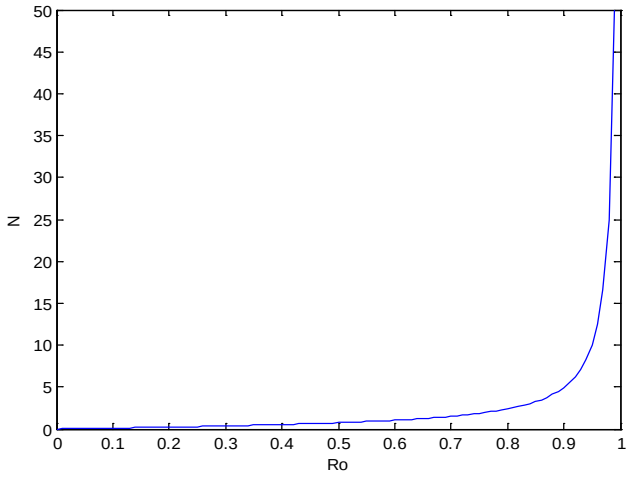


Figura 3.12 - Necesidad de memoria vs utilización.

También se podría graficar el retardo promedio que afecta a una trama, en función de la utilización, para diferentes valores de velocidad de salida del multiplexor:

$$T_r = \frac{T_s(2-\rho)}{2(1-\rho)} \tag{3.7}$$

En la Fig 3.13 se observa que, a medida que crece la utilización, aumenta el retardo, y también aumenta la necesidad de memoria. También, con una relación de utilización contante, a medida que disminuye la velocidad de salida también el retardo es creciente.

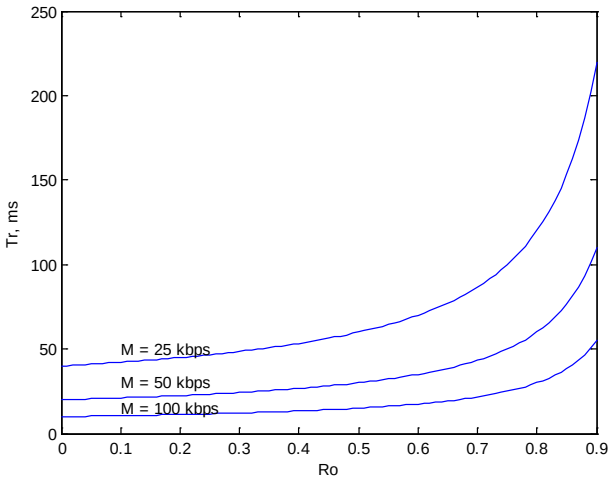


Figura 3.13 - Retardo promedio vs utilización

Por lo expuesto, es deseable una elección de parámetro de utilización, al momento de diseño o elección del multiplexor, en el orden del 80%. Este valor permite obtener una ganancia estadística, ya que la velocidad de salida se mantiene por debajo de la suma de las velocidades de entrada, con un retardo por paquete tal que la probabilidad de pérdidas de datos se mantenga dentro de límites aceptables. Dicha ganancia es mayor cuanto mayor es la naturaleza de ráfagas del tráfico de datos.

3.2 Redes de Conmutación de Circuitos

Desde la invención del teléfono, las redes telefónicas fueron creciendo tanto en usuarios como en extensión. Cuando comenzó la transmisión de datos sobre la arquitectura TCP/IP, ya existía una amplia red instalada de telefonía que cubría gran parte del mundo civilizado, motivo por el que se aprovecharon estas redes para el transporte de datos.

Se trata de redes de comunicación diseñadas primordialmente para transmisión de voz, que operan en base a elementos conmutadores que permiten seleccionar el destino de la comunicación deseada. Por este motivo se las conoce como redes conmutadas, y se las designa como Red de Telefonía Conmutada Pública (PSTN, Public Switched Telephone Network).

Desde sus comienzos, la telefonía utilizó conductores de cobre como medio de transmisión. Se trata de un par de conductores por cada usuario o equipo terminal telefónico, que finaliza en un centro de conmutación, conocido como central local u oficina final. Cada uno de estos conductores se denomina bucle de abonado, o línea de abonado. Mediante un micrófono, la voz se transforma en una señal eléctrica, que es transportada por el par telefónico hasta una central local y luego sobre la red conmutada. En el otro extremo, en el aparato receptor,

el auricular vuelve a convertir la señal en sonido. En la central, para que pueda establecerse la comunicación, el par del abonado llamante debe conectarse con el par del abonado llamado. Esta función se realiza mediante conmutadores, que están físicamente ubicados en centrales, conectados entre sí por un número cada vez más creciente de tecnologías que incluyen conexión cableada, transmisión por RF y fibra óptica. De esta manera, la red permite establecer un canal de comunicación dedicado entre los extremos abonados tal como se representa de manera simplificada en la Fig. 3.14.

Para poder hacer uso de las prestaciones ofrecidas por la red, la comunicación debe pasar por tres fases: establecimiento, transferencia y desconexión. En la fase de establecimiento se considera la creación del camino dedicado, por comunicación entre los nodos de la propia red, donde en cada enlace físico se reserva un canal lógico, por ejemplo por TDM. La fase de transferencia es la fase de intercambio de datos entre los extremos. La comunicación es *full duplex*, es decir está compuesta por dos comunicaciones unidireccionales, cada una de ellas transfiriendo las señales del micrófono de un extremo al auricular del otro extremo, ambas sobre el mismo par. La fase de desconexión se inicia a pedido de uno de los protagonistas de la comunicación y tiene como propósito la liberación de los recursos destinados a la misma, para su disponibilidad para otras comunicaciones. Desde el punto de vista de las redes de datos, se puede decir que la conmutación de circuitos ofrece un servicio orientado a la conexión.

En la fase de establecimiento, para indicar el destino al conmutador local, se debe realizar una operación denominada marcación, que consiste en indicar el número del destino, para que los conmutadores puedan efectuar las conexiones necesarias. En la actualidad, la marcación se efectúa por la superposición de dos tonos y se denomina marcación multi-tono. Esta función y otras, como el reconocimiento del tono de ocupado, la señal de llamada y todo aquello relacionado con la tarifación, forman parte de las llamadas funciones de señalización.

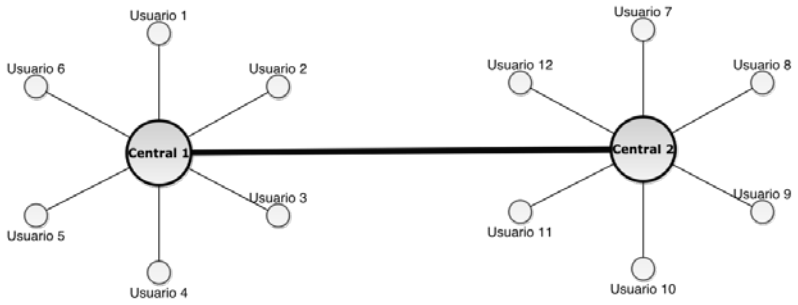


Figura 3.14 - Conexión entre centrales en Conmutación de Circuitos.

Los tres tipos de nodos que conforman la topología se conocen como:

- **Oficina final**: se localiza a nivel más bajo de la jerarquía, ya que se encarga de proveer acceso a la red a los abonados.

- **Tandem**: son nodos que conectan oficinas finales entre sí, proveyendo un punto de agregado de tráfico para comunicaciones entre ellas. A veces también pueden proveerles acceso al siguiente nivel de la jerarquía.

- **Tránsito**: se trata de nodos que proveen una interfaz a otro nivel de jerarquía. Generalmente agregan tráfico para su transporte a larga distancia.

Para conectar nodos conmutadores se pueden usar topologías tipo malla, con todos los nodos interconectados. Esta topología simplifica el transporte del tráfico entre nodos y evita cuellos de botella, aunque no es muy escalable cuando el número es elevado. Otro tipo de aproximación presenta la estructura de un árbol jerárquico, en la cual los nodos se van agregando desde la oficina final hacia el más alto nivel. En general, las redes de telefonía pública usan una combinación de ambos tipos de topologías, según el costo y patrones de tráfico esperados.

En redes telefónicas antiguas la conexión entre los abonados y la oficina central era aérea. En la actualidad, gran parte del cableado es subterráneo, no sometido a las inclemencias climáticas, construyéndose las oficinas centrales también bajo tierra para poder disminuir niveles de radiación. Miles de cables confluyen en la oficina central, provenientes de los usuarios del área geográfica que la misma sirve. Los pares de abonados se empaquetan en atados, que pueden cargar entre 25 y 600 pares. Una de las limitaciones de una oficina central es la cantidad total de pares que ingresan a la misma.

En dichas oficinas existen elementos conocidos como conmutadores de conexión cruzada, cuya función es conectar líneas de suscriptores y conmutar llamadas entre líneas de suscriptores que ingresan al dispositivo y aquellas que se conectan con otros dispositivos de cruce. Cada conexión necesita el establecimiento de un camino físico a través del conmutador que se dedica únicamente a la transferencia de señales entre los dos extremos.

Se puede esquematizar la central y sus abonados como una matriz, donde las filas y columnas representan a los abonados origen y destino respectivamente, como se observa en la Fig. 3.15. En este esquema, una unidad de control habilita los puntos de cruce para cada conexión. Si existen N abonados llamantes y N abonados llamados, deberán existir N^2 puntos de cruce o conexión para asegurar la posibilidad de conectar simultáneamente N abonados. Esta forma de multiplexado por división en el espacio se conoce como no bloqueante. Se trata de un esquema ineficiente debido al costo que implica el cableado de cada conmutador, que aumenta a su vez el costo de la central.

A modo de alternativa, existe un sistema más eficiente y económico que surge del estudio estadístico del tráfico telefónico, que considera establecer un grado de pérdidas de 1%, es decir la probabilidad de que, de cada 100 intentos de comunicación, 1 no pueda ofrecerse. De este modo, la cantidad de conmutadores se restringe notablemente, por ejemplo para 100 abonados puede reducirse a 15 conmutadores, con lo que simultáneamente sólo pueden cursarse 15 comunicaciones. Si se cursan 15 comunicaciones al mismo tiempo, y un

abonado de los 100 intenta comunicarse, recibirá el tono de ocupado, denegándose el servicio.

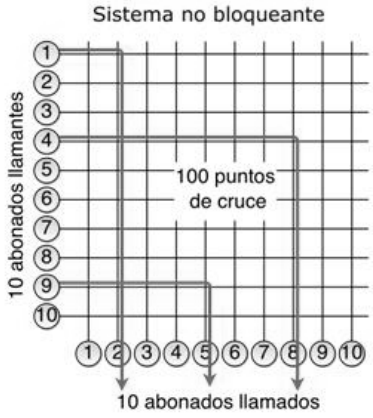


Figura 3.15 - Matriz de conexión por conmutación por división en el espacio.

A modo de ejemplo, en el esquema de la Fig. 3.16, cada abonado se encuentra representado por un número. Se observa que el abonado 5 está conectado con el abonado 2, y el 6 con el 1, pero el abonado 10 no puede conectarse con los abonados 3, 4 o 5. Además, cada comunicación implica 3 cruces o conexiones, complicando un poco la función de la unidad de control. A comparación del caso de un único conmutador de $10x10$, la operación se realiza por medio de conmutadores más pequeños, con menor cantidad de cruces, reemplazándose la situación de 100 puntos de cruce de conmutación no bloqueante, por 48 cruces del sistema bloqueante.

Cuando el abonado destino no pertenece a la misma central, la salida del conmutador debe ingresar a un canal del sistema de transporte, que finaliza en la entrada del conmutador de la central de destino, como se muestra en la Fig. 3.17. En dicha figura se han utilizado siglas para identificar diferentes equipos o componentes. Se designa con MID la caja de conexión que la compañía instala para ofrecer el servicio. Las siglas CO se refieren a la oficina central, donde confluyen los pares de abonados de un área. Las siglas terminadas en X designan oficinas de intercambio de diferentes jerarquías: FEX para no local, NEX para nacional e IEX para internacional. Las siglas IOT hacen referencia a las líneas troncales que conectan oficinas.

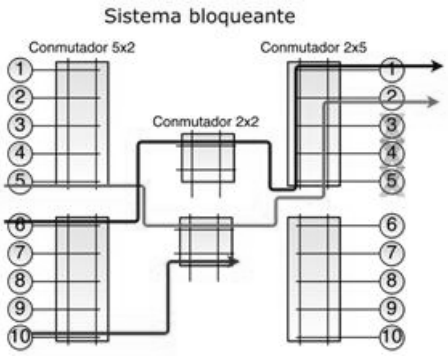


Figura 3.16 - Conmutación por División en el Espacio.

En una comunicación telefónica, los recursos y la ruta al destino deben reservarse y asignarse según sea la central a la cual pertenece el destino de la llamada. Para ello, la red se organiza en forma jerárquica, mediante un sistema de numeración que permite conocer el destino con la marcación de los primeros dígitos.

Para transportar la señal de voz entre conmutadores, las líneas individuales confluyen en un esquema jerárquico TDM, del tipo de los mencionados previamente. A la salida del multiplexor se juntan datos de múltiples canales de voz en líneas troncales de alta velocidad. En un principio, como hemos visto, este esquema era jerárquico por etapas, pero a medida que la tecnología fue avanzando, la funcionalidad de multiplexado se ha flexibilizado notablemente, pudiéndose realizar en el dispositivo multiplexor dentro del conmutador de cruce.

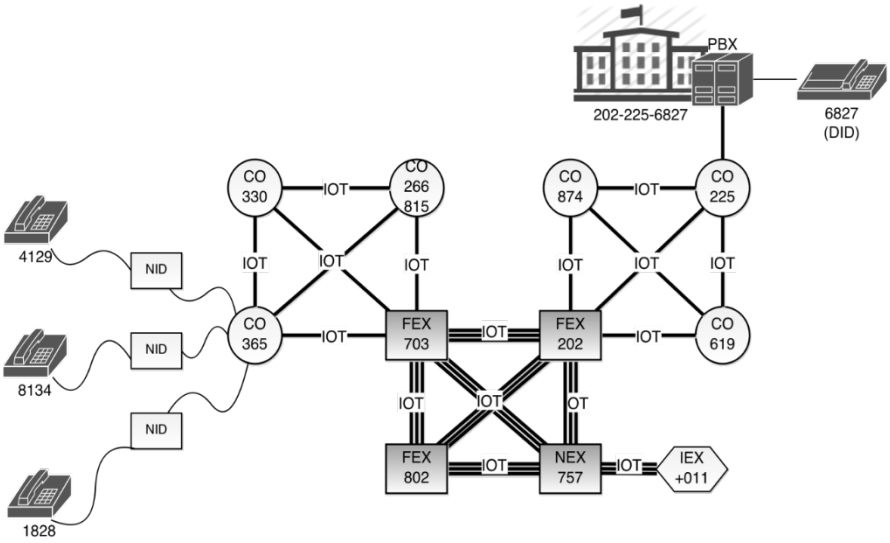


Figura 3.17 - Estructura jerárquica de la Red de Telefonía

En nuestro país, la numeración telefónica se organiza con un código de tres dígitos para el área, otro de tres dígitos de intercambio y, por último, cuatro dígitos para identificar al abonado. Técnicamente, un intercambio o *exchange* consiste en 10000 líneas de abonado dentro de un área geográfica local, todas confluyendo en una oficina final, sobre uno o más conmutadores de cruce dentro de esa oficina. Se trata de abonados cuyos números se encuentran dentro de un rango $xxx - 0000\ a\ xxx - 9999$. Dentro de la oficina final, cada intercambio es un conjunto de uno o más conmutadores de cruce que encaminan las llamadas según el número telefónico discado.

Un intercambio local es la oficina central más próxima al domicilio del abonado, representado por los tres dígitos a continuación del código de área. Un intercambio no local es aquel que se realiza fuera del área local, sobre troncales de alta velocidad, manejados por el proveedor local del servicio. Un intercambio nacional es la conexión entre el proveedor regional y el de larga distancia, para aquellas comunicaciones encaminadas según el código de área. Un intercambio internacional es la conexión entre proveedores nacionales de distintos países. En algunos países la red nacional es propiedad de un único operador, pero en la mayoría de los países existen operadores que gestionan el acceso local y otros que gestionan sus propias redes troncales.

A su vez, dentro de la propia red de telefonía, junto con los sistemas de transmisión y conmutación mencionados, convive un sistema de señalización.

Con un sistema de señalización se transmiten señales de control para gestionar la red y para establecer, mantener y finalizar las llamadas, intercambiando información entre el abonado y los conmutadores, entre los propios conmutadores, y entre los conmutadores y el centro de gestión de red. Entre las funciones de señalización se pueden destacar: la generación de tonos audibles para el abonado, tales como marcado, señal de llamada y señal de ocupado; la transmisión del número destino a la central de conmutación para enrutamiento; la comunicación de aviso de llamada no posible entre conmutadores; la comunicación de aviso de finalización de llamada entre conmutadores; la generación de señal audible en destino; la transmisión de información para facturación; la transmisión de información de estado de equipos y líneas para enrutamiento y mantenimiento y la transmisión de información para diagnóstico y aislamiento de fallos y el control de equipos especiales, por ejemplo, satélites.

Los tipos de señales, por su parte, se pude agrupar en cuatro categorías:

- **Señales de supervisión**: son señales binarias de control (activado/desactivado, verdadero/falso), tales como solicitud de servicio, respuesta, aviso y retorno a desocupado. Informan acerca de la disponibilidad del abonado llamado y de los recursos de la red necesarios.

- **Señales de direccionamiento**: identifican al abonado. Inicialmente se genera una señal de dirección por parte de un abonado origen cuando marca un número de teléfono. La dirección resultante se puede propagar

a través de la red para permitir el encaminamiento, y así localizar y hacer que suene el teléfono del abonado destino.

- **Señales de información**: proporcionan al abonado información acerca del estado de la llamada. Se trata de señales audibles que se emplean para el establecimiento y el cierre de la conexión.

- **Señales de gestión**: son para mantenimiento y funcionamiento general de la red. Pueden tener forma de mensajes, como por ejemplo una lista de rutas predefinidas enviadas a un conmutador, para la actualización de sus tablas de encaminamiento.

A su vez, existen cuatro tipos de señalización:

- **Señalización intra-canal**: es la tradicional. El mismo canal transporta señales de control y señales de información. De este modo, sigue toda la ruta de la llamada, compartiendo los recursos asignados a la misma, por eso tiene limitaciones en cuanto a la velocidad de transferencia.

- **Señalización intra-banda**: el mismo camino físico y la misma banda de frecuencia transportan señales de control e información en distinto tiempo. Es posible de realizar durante los silencios de la comunicación, pero resulta una señal de naturaleza discontinua.

- **Señalización fuera de banda**: se aprovecha el estrecho ancho de banda que no ocupa la señal de voz (el resto de los 4 kHz), sin importar que haya o no señal de voz en la línea. Para este tipo de señalización se precisa de electrónica adicional.

- **Señalización por canal común**: las señales de control se transportan sobre rutas físicamente independientes de los canales de voz. Se trata de canales de señalización comunes a varios canales de voz, conformando una red que se especializa en el transporte de mensajes cortos. Alguna de las ventajas son que se logra reducir retardo de establecimiento y se genera un esquema más adaptable a nuevas tecnologías.

La señalización por canal común ha traspasado los límites de la red de telefonía, usándose en redes inteligentes, servicios suplementarios y señalización en redes de telefonía celular. El Sistema de Señalización 7 (SS7, Signaling System 7) es el sistema actual de señalización por canal común, definido por la ITU-T en las recomendaciones de la serie Q.700. En SS7, la información de señalización se transfiere por medio de mensajes de hasta $200\,bytes$, cuya estructura la define el estándar. Con estos paquetes se puede realizar señalización relacionada con los circuitos, generada a demanda para el manejo básico de establecimiento y desconexión de una llamada, y señalización no relacionada. Esta última permite transferir datos libremente entre entidades, sin tener directa relación con el control de tráfico. La necesidad de estas transferencias surge de la

aparición de servicios suplementarios y de las consultas a bases de datos en el caso de redes de telefonía celular.

Esta forma de señalización por canal común tiene varias ventajas con respecto a la señalización intra-canal. Se trata de una señalización más flexible y, por lo tanto, con mejores perspectivas de evolución, permitiendo un establecimiento más rápido de las llamadas y mayor control sobre las mismas ya que no contiende con el propio tráfico de usuario.

En los últimos años, la red de telefonía fija, originalmente diseñada para transporte de voz, comenzó a transportar cada vez mayor cantidad de datos, obligando a las prestadoras a adecuar su tecnología a formatos totalmente digitales y nuevas formas de acceso a Internet. Los más modernos esquemas de telefonía, tales como VoIP, para transporte de señales de voz digitalizada sobre el protocolo IP, también revolucionaron este tipo de redes, dejando abiertos algunos interrogantes. La falta de regulación existente al respecto, se puede traducir en nuevas oportunidades de negocios y un desafío para las propias prestadoras, ya que esta comunicación no se encuentra tarifada.

3.3 Redes de Conmutación de Paquetes

Las redes de conmutación de paquetes transportan bloques de datos especiales, denominados paquetes, que tienen estructuras similares en lo que respecta a los encabezados, sin importar la información que transporten. Surgieron como alternativa para la transmisión de datos, ya que las redes de conmutación de circuitos operaban de manera ineficiente en el caso de tráfico en ráfagas.

A diferencia de las redes de conmutación de circuitos, la conmutación de paquetes permite el traslado de secuencias de paquetes a velocidad variable. Cuando estas secuencias arriban a los elementos de conmutación, se almacenan en *buffers*, colocándose en una cola para su posterior procesamiento. Esta técnica de almacenamiento y re-envío, resulta en un retardo variable en la transmisión, dependiente de la carga de tráfico de la red.

En una red de conmutación de paquetes, los recursos se manejan por multiplexado estadístico, también conocido como ancho de banda a demanda. Cada canal de entrada al multiplexor consiste de una secuencia de paquetes, que se re-envía de manera asincrónica, generalmente por uso de técnicas denominadas Primero en Entrar Primero en Salir (FIFO, First In First Out). Como los paquetes llevan información de usuario e información de control necesaria para enrutamiento, se abren las puertas a nuevas prestaciones para el re-envío, por ejemplo la posibilidad de adicionar esquemas de prioridad para el encaminamiento.

Se pueden clasificar las redes de conmutación de paquetes en dos grandes tipos: redes de datagramas y redes de circuitos virtuales. Como en cualquier red de conmutación, uno de los problemas más graves a enfrentar es el problema de la congestión. La forma de reacción ante este problema es diferente según el tipo

de red de conmutación de paquetes, resultando su adaptación mejor en un caso que en el otro.

3.3.1 Redes de Datagramas

En las redes de datagramas, cada paquete, también llamado datagrama, se encamina de manera independiente. Debido a esta forma de enrutamiento, basada en un proceso de decisión por cada paquete, los mismos pueden recibirse en desorden y hasta perderse antes de llegar a destino. Es responsabilidad de los extremos finales de la comunicación el mantener la secuencia de los paquetes recibidos. Cada paquete posee información sobre el destino, que es consultada en cada nodo de la red, para la selección del enlace de salida apropiado hacia otro nodo, que se encuentra más cerca del destino. La selección se realiza teniendo en cuenta la información almacenada en el nodo en una tabla de enrutamiento.

Debido a que la red de datagramas no ofrece un servicio confiable, la confiabilidad será funcionalidad de la capa de transporte. Se dice que son redes del tipo de mejor esfuerzo o *best effort*. La Fig. 3.18 presenta una red de este tipo, donde un dispositivo A transmite tres paquetes a otro dispositivo D y, dado que los mismos se encaminan de manera independiente, pueden llegar a destino por caminos diferentes y de manera desordenada.

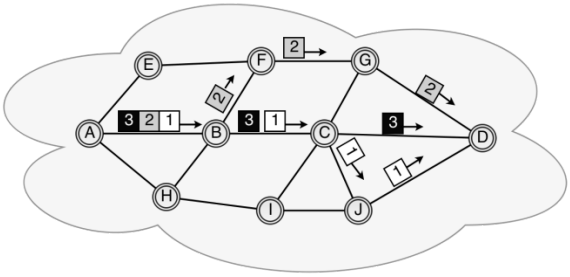


Figura 3.18 - Red de datagramas

Dado que su concepción es tan sencilla, las redes de datagramas son de fácil adaptación a una situación de congestión. Si un nodo falla, es posible desviar el tráfico que va dirigido a él, hacia otro nodo alternativo. Estos cambios de rutas exigen el trabajo de otros protocolos especiales, denominados protocolos de enrutamiento, cuya función no es el traslado de la información en sí, si no el establecimiento adaptivo de las rutas más eficientes entre distintos puntos de la red. En cuanto a la situación de congestión en sí misma, la filosofía de las redes de datagramas se apoya en que es preferible permitir que todos usen la red a costa de una degradación pareja del servicio, antes que permitir que algunos la usen, mientras a otros usuarios legítimos se les deniega el servicio. Se refiere a esta situación como de congestión no bloqueante.

Por su parte, las interfaces sobre una red de datagramas son más sencillas de implementar y el código que maneja la capa de red más simple. Debido a que la mayoría del tráfico de red es de tipo ráfaga, parecería poco eficiente reservar recursos de manera previa a la transmisión. Además, muchas aplicaciones no requieren entrega secuencial de paquetes y otras son tolerantes a las pérdidas de datos. Este podría ser el caso de la transmisión de mensajes de voz, donde los paquetes fuera de orden se pueden descartar pues se acepta algún porcentaje de pérdidas.

Por otra parte, los protocolos asociados a este tipo de servicios son protocolos sin estado, más sencillos. Mantener el estado de la comunicación por parte de la red puede ser muy difícil, dado la cantidad de comunicaciones que existen al mismo tiempo. Un ejemplo de protocolo de red que funciona bajo estas premisas es IP. Aquellas aplicaciones que no requieren más que un servicio *best effort* pueden apoyarse sobre esta protocolo de red por intermedio del protocolo de transporte UDP. Otras aplicaciones, tales como la navegación en la web, el servicio de correo o la transferencia de archivos, que precisan de confiabilidad en el transporte de los datos, se apoyan sobre el mismo servicio pero encapsuladas en el protocolo TCP.

3.3.2 Redes de Circuitos Virtuales.

En las redes de circuitos virtuales se establece una ruta previamente, al comienzo de la comunicación, y todos los paquetes siguen la misma ruta. Por este motivo, es necesaria una fase previa a la transferencia de información, denominada petición de llamada, en términos de redes de conmutación de circuitos, o de establecimiento del circuito virtual, donde se fija la ruta entre extremos. Se denomina circuito virtual porque parece ser un camino físico dedicado, pero en realidad es compartido por otras comunicaciones. También existe una fase final de liberación de recursos, o petición de liberación. La red garantiza el orden en la entrega y puede garantizar el control de errores, descargando de este modo estas tareas de los extremos finales de la comunicación.

De todas maneras, son redes cuyos elementos de conmutación son similares a los de las redes de datagramas, ya que también funcionan bajo la premisa *store&forward*. Se diferencian en su capacidad de adaptación a la situación de congestión, ya que la falla de un nodo genera la pérdida de todos los circuitos virtuales que pasan por él, resultando más complicado el proceso de recuperación.

A diferencia de la red de datagramas, la de circuitos virtuales puede rechazar una comunicación, de no poder asegurar de antemano los recursos necesarios para la misma. En casi todos los esquemas de este tipo, si la red acepta la llamada debe proveer el suficiente ancho de banda y, en todo caso, denegar nuevas llamadas si la calidad del servicio de las ya establecidas se degrada.

Durante la fase de establecimiento, se asignan números cortos llamados Identificadores de Circuitos Virtuales (VCI, Virtual Circuit Identifier) que se

incorporan a los encabezados y se consultan durante la fase de transferencia para realizar el re-envío, como se observa en la Fig. 3.19. Los nodos intermedios cargan sus tablas de enrutamiento con información para esa comunicación, quedando fija la relación entre un VCI y el puerto de salida del *router*. Por tratarse de números de pocos bits, se especula con que la conmutación podría ser más rápida respecto del caso de datagramas.

La red de circuitos virtuales se ideó para que su funcionamiento fuera similar al de las redes de conmutación de circuitos. Al establecerse una llamada, se fija la ruta que seguirán todos los paquetes de la misma, pudiendo la propia red garantizar un servicio confiable, con entrega de cada paquete en orden, sin pérdidas ni duplicados. También, al usarse una única ruta, se puede conocer el tamaño de paquete más pequeño aceptado por una red que forma parte de la misma, y evitar de este modo la fragmentación de los paquetes.

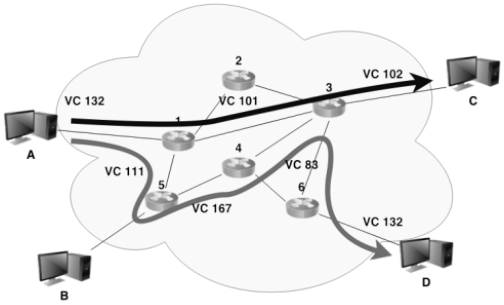


Figura 3.19 - Red de circuitos virtuales.

Debido toda la funcionalidad que se espera de la capa de red, en las redes de circuitos virtuales resultaría una capa de transporte de diseño más sencillo que en el caso de datagramas. Ejemplos de protocolos que funcionan bajo estas premisas son X.25, Frame Relay y ATM.

3.4 Conmutación de Circuitos vs Conmutación de Paquetes

La Tabla 3.3 presenta una comparación entre los tres tipos de redes de conmutación mencionados. Se tienen en cuenta aspectos relacionados con la elección de las rutas, el tipo de tráfico transportado, la necesidad de memoria en los conmutadores, la existencia de una fase inicial de establecimiento, la significación relativa en cuanto al retardo de conmutación, características en cuanto al bloqueo, el tipo de conmutadores, aspectos relacionados con la conversión de velocidades y la confiabilidad del sistema de transporte.

Tabla 3.3 - Comparación de Redes de Conmutación

Aspecto	Conmutación Circuitos	Conmutación de Paquetes	
		Red de Datagramas	**Circuitos Virtuales**
Ruta	Fija, establecida en el comienzo.	Independiente, decisión por paquete.	Fija, establecida en el comienzo.
Tipo de Tráfico	Inicialmente ideada para transporte de muestras de voz	Paquetes de datos	Paquetes de datos
Buffering	Conmutación rápida	Sí. Store&forward	Sí. Store&forward
Retardo Establecimiento	Sí	No	Sí
Retardo Conmutación	Nulo	Sí	Sí
Bloqueo	Red bloqueante si no se aseguran los recursos al inicio	Red no bloqueante. Congestión es degradación experimentada por todos los usuarios.	Red bloqueante si no se aseguran los recursos al inicio
Nodos	Conmutadores TDM	Routers, tecnología basada en TDM estadístico.	Routers, tecnología basada en TDM estadístico.
Velocidad	Constante, sin conversión de velocidades.	Conversión de velocidades. Ancho de banda a demanda.	Conversión de velocidades. Ancho de banda a demanda.
Confiabilidad	Depende de la red física	No confiable a nivel de red.	Confiable a nivel de red.

En el caso de usuarios de redes de Conmutación de Paquetes, estos contratan un tipo de acceso que, desde el punto de vista de la red, se pude denominar funcionamiento externo, por encontrarse en el extremo de la misma. Con el mismo razonamiento, se denomina funcionamiento interno al que se refiere a las redes que debe atravesar la comunicación y que no dependen del servicio contratado. En este sentido, se pueden dar todas las combinaciones posibles entre funcionamiento externo y funcionamiento interno. Lo cierto es que, desde el punto de vista del usuario, lo que importa es el servicio contratado, más allá de que éste deba pasar por otro tipo de redes en su camino al destino.

3.5 Redes de Conmutación de Paquetes – Enrutamiento

El proceso de seleccionar caminos para enviar tráfico a lo largo de una red se denomina encaminamiento o enrutamiento. Para realizarlo de manera eficiente se usan dispositivos especiales, denominados dispositivos de enrutamiento o *routers*. El proceso de re-envío y su eficiencia depende de la construcción de una tabla, denominada Tabla de Enrutamiento, que el *router* mantiene en su propia memoria.

No sólo los *routers* poseen **Tablas de Enrutamiento**. Toda máquina conectada a una red también la tiene. En ambos casos las tablas pueden construirse de dos maneras. Una posibilidad consiste en que el administrador, por medio de *scripts* que se ejecutan al inicializar el sistema o por medio de comandos ejecutados interactivamente, introduzca manualmente las entradas de la tabla. Esta técnica se denomina enrutamiento estático, debido a que la tabla de enrutamiento se construye cuando el dispositivo se enciende y no varía con el tiempo, excepto que el administrador la modifique. El enrutamiento estático es factible sólo en el caso de redes pequeñas.

La otra posibilidad es ejecutar en cada máquina un programa que actualice automática y periódicamente la tabla de enrutamiento, por información intercambiada con otros dispositivos. Este tipo de enrutamiento es típico en las redes administradas por los ISP. En estas redes se instalan protocolos de enrutamiento que permiten a los *routers* comunicarse entre sí, de manera de mantenerse al tanto de posibles cambios por fallas de otros nodos de la red y adaptar entonces la información de la tabla a esta nueva situación. Se dice que el enrutamiento es dinámico o adaptivo.

En cualquier caso, deben existir formas de evaluar la eficiencia de cada ruta en particular. Estos criterios, que pueden referirse a anchos de banda de los enlaces, tamaño de las colas de entrada o cantidad de enlaces entre fuente y destino, se plasman en números, denominados métricas, que se asocian a los enlaces de cada nodo. Una de las métricas más utilizadas es el número de saltos, medida como la cantidad de *routers* que los paquetes atraviesan entre fuente y destino de una ruta particular.

La Fig. 3.20 presenta un ejemplo de una red con métricas asociadas a cada enlace de cada nodo. Sobre cada enlace puede haber más de una métrica según el sentido en que se mueva la información. Por ejemplo, si se elige como métrica el ancho de banda del enlace y se pretendiera encaminar paquetes desde el nodo 1 al nodo 6, el camino óptimo sería el que parte del nodo 1 y pasa por los nodos 4 y 5. En cambio, el camino que se correspondería con la mejor ruta si la métrica se asociara a la cantidad de saltos entre los mismos nodos fuente y destino, partiría del nodo 1, pasaría por el 3 y finalmente llegaría al nodo 6.

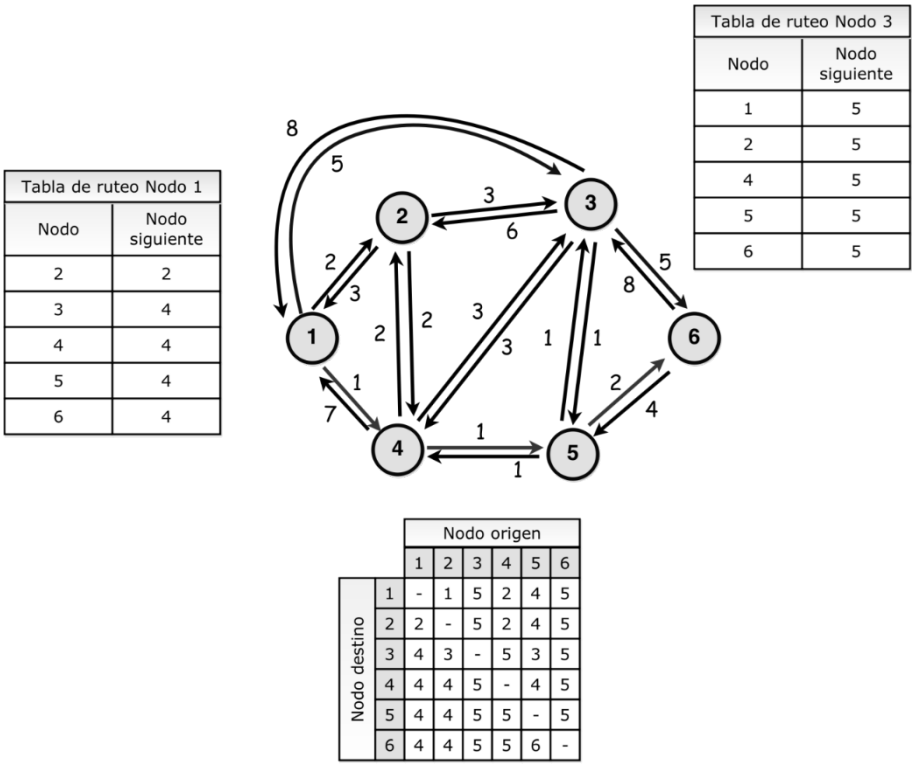


Figura 3.20 - Enrutamiento Estático

En ciertas redes pequeñas, de tráfico predecible y topología conocida, se podrían asociar métricas sobre cada enlace, tal como se presenta en la Fig. 3.20, obteniéndose una especie de fotografía del estado de la red. Con esta información, se podría consignar en una matriz, para cada nodo tomado como nodo origen, el nodo siguiente en la mejor ruta a cada nodo destino, tal como se consigna en la figura. Con la información aportada por esta matriz, se podrían armar tablas, una por cada nodo, donde se almacenarían los mejores caminos a un destino particular. En la figura se presentan las tablas de los nodos 1 y 3, derivadas de la información consignada en la matriz general. Como se puede observar, conociendo el estado referido a la red completa, es posible que cada nodo guarde información relativa sólo al el nodo siguiente en su mejor ruta hacia un destino particular.

El cálculo de las rutas óptimas se puede realizar aplicando algún algoritmo: Dijkstra o Bellamn-Ford, entre los más conocidos. Este cálculo es útil mientras la topología y la métrica, o alguna de ellas, no se vean alteradas, ya que las rutas permanecen fijas. Obviamente, ya que cada nodo no precisa almacenar la ruta completa a un destino, sino sólo el próximo nodo en la misma, las tablas en cada nodo deben ser consistentes con las de los demás.

Este tipo de **enrutamiento estático** no presenta diferencias si la red es de datagramas o de circuitos virtuales. Su ventaja es la simplicidad, aunque su

desventaja es muy importante, puesto que es una forma de enrutamiento inflexible ante fallos de congestión o de la red. Por este motivo, se ha observado que este tipo de enrutamiento es apropiado para redes pequeñas, confiables y con carga estacionaria.

En la mayoría de los casos, para poder adaptarse a la situación operativa de la red, las tablas de enrutamiento deben poder modificarse de manera dinámica, ya que las situaciones de falla y congestión condicionan las decisiones de enrutamiento. Para hacer posible la adaptabilidad, es necesario un intercambio de información entre *routers*, a través de algún **protocolo de enrutamiento**. Un protocolo de enrutamiento habilita a los *routers* a intercambiar mensajes sobre el estado de sus enlaces y sus propias Tablas de Enrutamiento, permitiendo la modificación dinámica de las mismas en base a esta información. Esta adaptabilidad aumenta la carga de procesamiento, así como también el tráfico en la red. Por este motivo, existe un compromiso entre la frecuencia de intercambio de mensajes y la velocidad de reacción ante los cambios.

Algunos ejemplos de protocolos de enrutamiento más conocidos: Protocolo de Información de Enrutamiento (RIP, Routing Information Protocol), Protocolo de Enrutamiento de Puerta de Enlace Enriquecido (EIGRP, Enhanced Interior Gateway Routing Protocol) y Abrir Primero el Camino Más Corto (OSPF, Open Shortest Path First). La información de próximo salto en la ruta se almacena en las tablas, como en el caso estático, pero el hecho de compartir información entre *routers* incrementa no sólo el uso del ancho de banda de la red, sino también la carga sobre el procesador y la propia memoria RAM de los *routers*.

En general, existen dos categorías de protocolos de enrutamiento dinámico: protocolos de Vector Distancia y protocolos de Estado de Enlace. EIGRP toma características de ambos tipos y por eso se considera un protocolo de enrutamiento híbrido.

Los protocolos de Vector Distancia, tales como RIP, se caracterizan por que la comunicación es periódica, entre *routers* vecinos, entre los que se intercambian las tablas de enrutamiento completas. Estos protocolos son de convergencia lenta y, por lo tanto, con cierta tendencia a generar lazos de enrutamiento. Utilizan algoritmos como el de Bellman-Ford para determinar el camino más corto. RIP usa la cuenta de saltos como métrica, otros protocolos generan una métrica dependiente del retardo y del ancho de banda.

Los protocolos de Estado de Enlace, tales como OSPF, se idearon para compensar las fallas de los anteriores. Estos protocolos mantienen varios tipos de información. Son capaces de armar tablas de los vecinos, como un listado de vecinos y las interfaces a las que se conectan. También mantienen tablas de topología, una especie de mapa de todos los enlaces de cierta área y sus estados. Por supuesto, además arman tablas de enrutamiento, con la mejor ruta a cada destino particular. De este modo, cada *router* conoce el estado de todos los enlaces de un área y puede entonces obtener en un instante una fotografía del estado de la red. Cuando un enlace cae, o se produce alguna falla, se genera un aviso que recorre todos los nodos para que las tablas de enrutamiento se reajusten a la nueva situación. Generalmente estos protocolos usan el ancho de banda como métrica de enlace y el algoritmo de Dijkstra para el cálculo de las rutas óptimas.

3.6 Redes de Conmutación de Paquetes – *Routers*

Los *routers* juegan un rol más que importante en los entornos ISP, para poder asegurar la provisión de los servicios contratados por los usuarios. El crecimiento del tráfico en Internet y la exigencia de condiciones relacionadas con la calidad de servicio, ha generado importantes cambios en el diseño de estos elementos.

Para poder entender las nuevas tendencias tecnológicas hay que tener presente los bloques funcionales de un *router*.

Se trata de un dispositivo que posee puertos de entrada por los que ingresan los paquetes que se des-encapsulan para ser entregados al nivel de red. Con información del encabezado de red, se realiza una búsqueda en la tabla de enrutamiento para determinar el *router* del próximo salto y el puerto de salida. También podría realizarse algún tipo de procesamiento sobre el encabezado del nivel de red, antes de su entrega a los puertos de salida. Para bajar el paquete sobre el nuevo enlace, se lo debe re-encapsular con el encabezado apropiado. Podría suceder que varios paquetes fueran entregados al mismo puerto a la vez, por este motivo los *routers* manejan colas de entrada y de salida, con características de almacenamiento temporal, funcionalidad que permite realizar conversión de velocidades.

De este modo, el *router* realiza principalmente dos funciones: enrutamiento y re-envío de paquetes. La funcionalidad de enrutamiento exige el procesamiento de mensajes de los protocolos de enrutamiento que les permiten cargar sus tablas y mantenerlas actualizadas. Se trata de una tarea relativamente sencilla, a cargo de un procesador de propósito general. En cambio, la tarea concerniente al re-envío depende de las condiciones de tráfico. Se podría realizar con un único procesador, pero entonces se vería limitada la velocidad de procesamiento de paquetes. Por este motivo, la tendencia actual es la de paralelizar el procesamiento de re-envío para poder alcanzar mayores velocidades.

En una implementación tradicional, desde el punto de vista del *hardware*, un *router* está conformado por interfaces de red de entrada/salida, una matriz de interconexión, una Unidad Central de Procesamiento (CPU, Central Processing Unit) y memoria. La matriz o bus de interconexión transporta los paquetes entre las interfaces y el procesador.

Muchos modelos de *router* proveen capacidades de procesamiento y memoria centralizada, tal como se presenta en la Fig. 3.21. En esta arquitectura básica, los paquetes arriban por las interfaces y se envían a la CPU que, a su vez, decide sobre cuáles interfaces deben ser re-enviados. Como la memoria se encuentra centralizada, la desventaja es que los paquetes se almacenan en ella, debiendo circular por el bus interno dos veces, generando verdaderos cuellos de botella.

En la segunda generación de *routers* que se presenta en la Fig. 3.22 se mejoró la arquitectura por distribución de las operaciones de re-envío de

paquetes. Se usan procesadores más rápidos y memorias *cache* para las rutas, junto con buffers para transmisión y recepción, en las propias interfaces. De este modo, en muchos casos el bus se circula una sola vez, gracias al caché de rutas en las propias interfaces. De todas maneras, esta arquitectura no es escalable para altas velocidades, ni siquiera mejorando la velocidad del procesador.

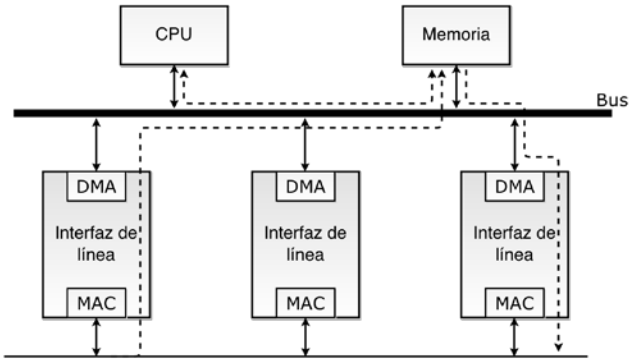


Figura 3.21 - Arquitectura tradicional básica de un router.

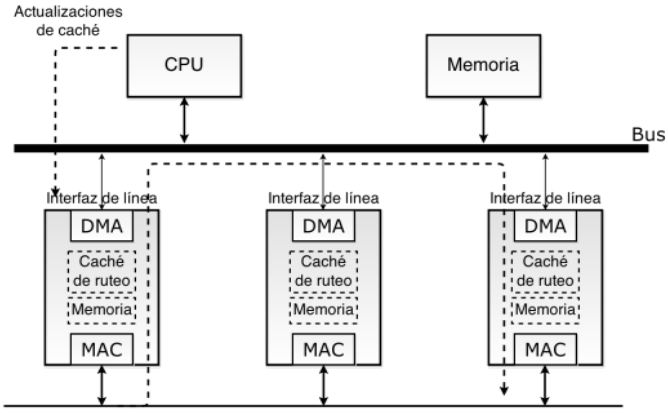


Figura 3.22 - Arquitectura mejorada de un router. Segunda generación.

Para mayores velocidades de procesamiento se usan arquitecturas con múltiples máquinas de re-envío, tal como la que se presenta en la Fig. 3.23. Las interfaces recogen los paquetes de entrada, se les separa la información relevante, que se envía a las máquinas para su validación y enrutamiento. Mientras la máquina hace su trabajo, el resto del paquete se almacena en un buffer de entrada paralelo y, cuando termina, envía la información procesada a la interfaz de salida apropiada. El resto del paquete se mueve entonces entre ambas interfaces a un buffer para su transmisión. De este modo, en las máquinas de re-envío se puede trabajar con varios encabezados en paralelo y por el bus no circula la carga de los

paquetes. Para mantener el orden entre los paquetes y sus encabezados, la asignación hacia las máquinas se hace en el modo cadena o *round robin*, mientras que el circuito de control de salida también trabaja de la misma manera. Se podría mejorar la velocidad si se trabajara en algún modo de balanceo de carga, siendo el circuito de control el encargado de preservar el orden.

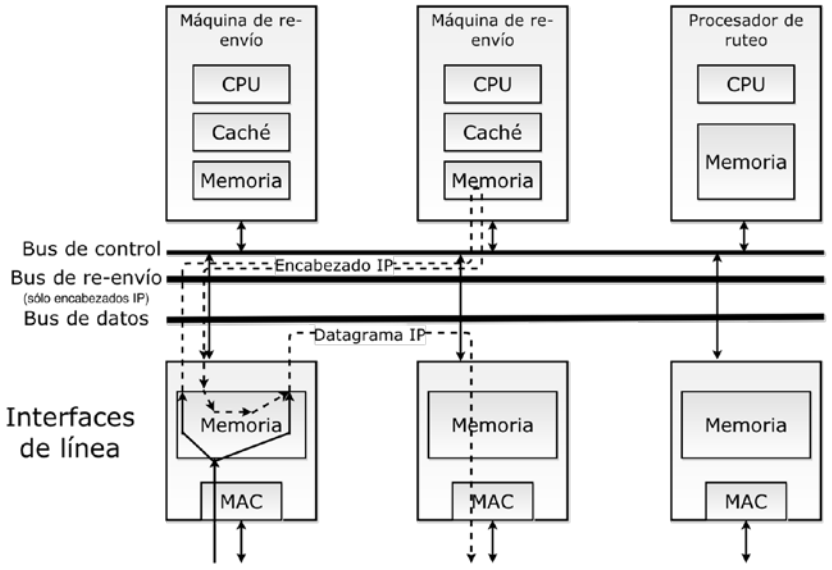


Figura 3.23 - Arquitectura de múltiples máquinas de re-envío.

En una tercera generación de *routers*, se reemplazó el bus compartido por un *switch* o unidad de interconexión entre interfaces, como se muestra en la Fig. 3.24. Esto traslada el cuello de botella al procesamiento de paquetes.

Un modelo de estos dispositivos de encaminamiento, conocido como multi-gigabit *router*, trabaja de manera que, cuando un paquete llega a las placas de línea, su cabecera es removida y se pasa a la máquina de re-envío, a través del *switch*. El resto del paquete permanece en la placa de entrada. La máquina de re-envío realiza un procesamiento en orden FIFO, procediendo a leer el encabezado para determinar cómo encaminar el paquete, luego actualiza el encabezado y lo envía, junto con instrucciones de re-envío, a la placa de entrada. La placa reconstruye el paquete y lo envía a la correspondiente placa de salida para su transmisión.

La mayoría de los *routers* con esta arquitectura, también poseen un procesador de control para funciones de administración, tales como generación de tablas y administración de enlaces.

Debido a la constante evolución de los entornos de red, empiezan a aparecer nuevas funcionalidades con exigencias para que los *routers* sean capaces de re-enviar tráfico agregado en rangos de velocidad cada vez mayores. Este desafío focaliza el interés en mejorar aspectos relacionados con el aumento de

ancho de banda interno, la velocidad de procesamiento de paquetes, la rapidez de consulta a las tablas de enrutamiento y los tiempos de acceso a memoria.

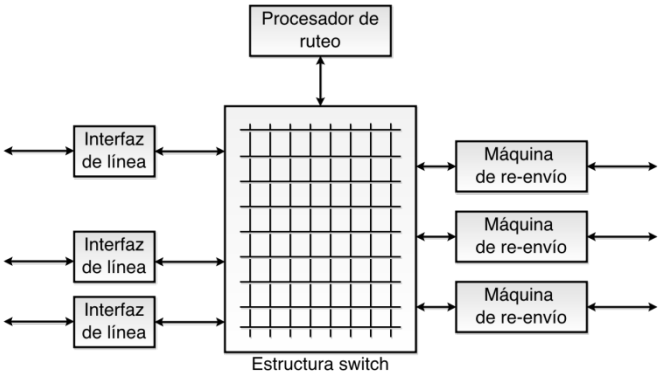


Figura 3.24 - Arquitectura de tercera generación.

A pesar de la cantidad de operaciones que un *router* debe realizar, pocas son necesarias en tiempo real, con lo cual, una solución de bajo costo para mejorar la eficiencia de funcionamiento puede consistir en la realización en software de las operaciones más lentas y el diseño en hardware de las funciones críticas. El esquema final dependerá de una relación de compromiso entre performance, complejidad y costo, tratándose de un área en continuo avance.

Bibliografía

1. Stallings, William, “Comunicaciones y Redes de Computadores”. Sexta Edición. Prentice Hall Inc., 2000.
2. Sklar, Bernard, “Digital Communications. Fundamentals and Applications”. Second Edition. Prentice Hall Inc., 1988.
3. Lindner, Hass, “TDM Techniques”, 2012. https://www.ict.tuwien.ac.at/lva/384.081/datacom/03-TDM_Techniques_ss2012_v5-2.pdf
4. InetDaemon.Com, “A Visual Guide to the Public Switched Telephone Network (PSTN)” http://www.inetdaemon.com/tutorials/telecom/pstn/visual_guide.shtml
5. Dryburgh, Lee and Hewett, Jeff, “Signaling System No. 7 (SS7/C7): Protocol, Architecture, and Services”. Cisco Press, 2004. http://www.informit.com/library/library.aspx?b=Signaling_System_No_7
6. Aweya, James, “IP Router Architectures: An Overview”. Nortel Networks. Ottawa, Canada, K1Y 4H7 http://www.cs.virginia.edu/~cs757/papers/awey99.pdf

Problemas

1. Calcule el ancho de banda ocupado por un conjunto de 32 fuentes de voz, asignando un espectro de 4 kHz para dicha señal, si se transmiten sobre un canal por medio de multiplexado FDM. Compare el ancho de banda obtenido con el que resulta de elegir un sistema de multiplexado tipo TDM sincrónico. ¿Cuál es la ventaja de este último respecto del primero?

2. Si sólo 6 computadoras se encuentran conectadas a la entrada de una jerarquía T1 ¿Cuál es la máxima velocidad a la que puede transmitir cada una?

3. ¿Cuál es la principal desventaja de TDM sincrónico desplegado en una jerarquía PDH? ¿Qué mejora introdujo SDH?

4. Un conjunto de 10 de computadoras se conectan en la entrada de un multiplexor TDM sincrónico. Cada computadora puede transmitir datos a 100 Mbps el 50% del tiempo. ¿Cuál es la mínima velocidad requerida a la salida del multiplexor? ¿Cómo se modifica la situación con un multiplexor estadístico?

5. ¿Cuál es la relación de compromiso más importante en el caso de TDM estadístico?

6. ¿Por qué puede haber conversión de velocidades en una red de paquetes?

7. Mencione las causas que motivaron la evolución entre las diferentes arquitecturas de *routers*.

CAPÍTULO IV

Medios de Transmisión

En este capítulo se aborda el estudio de diferentes medios de transmisión de uso extendido en redes de datos, especialmente en el caso de redes de área local. El propósito detrás de los conceptos aquí presentados es el de apreciar características y limitaciones de cada uno de los tipos, preparando el camino para los capítulos que conforman la segunda parte de este libro.

En general, los medios de transmisión se pueden clasificar en dos grandes grupos. Los medios guiados utilizan algún tipo de cable para el transporte, especialmente cable coaxial, par trenzado o fibra óptica. Los medios no guiados dependen de la existencia de una antena para generar las condiciones apropiadas de propagación. En la transmisión no guiada distinguimos la transmisión por radio, microondas y la comunicación satelital.

Debido a las diferencias notables entre los problemas referidos a la transmisión de datos sobre medios guiados y no guiados, se considera la puntualización de determinadas características de la comunicación en ambos casos.

En el caso de los medios guiados, se han introducido los parámetros principales que caracterizan la transmisión por cable de par trenzado, el de uso más extendido en el caso de redes LAN en la actualidad.

El capítulo también presenta las particularidades más relevantes del propio canal inalámbrico para dispositivos móviles, para que se comprendan conceptualmente los desafíos enfrentados por las redes LAN inalámbricas. La comunicación sobre este canal se aparta del modelo tradicional de canal de ancho de banda limitado inmerso en ruido blanco gaussiano, debido a la presencia de múltiples caminos entre transmisor y receptor. La dependencia de la presencia de objetos fijos o móviles condiciona la comunicación, generando efectos de desvanecimiento en la señal recibida. El capítulo presenta un modelo de canal adecuado bajo estas condiciones.

4.1 Introducción

En comunicaciones, se define como medio de transmisión al camino físico entre los extremos de transmisión y recepción. Este camino puede

extenderse mediante el uso de elementos especiales que permitan amplificar la señal, en el caso de transmisión analógica, o recuperarla y retransmitirla, en el caso de señales digitales. En cualquier caso, la transmisión se realiza mediante ondas electromagnéticas, que pueden viajar guiadas o no. Se habla de medios de transmisión guiados cuando las ondas viajan sobre un medio sólido o cable. Los medios de transmisión no guiados se caracterizan por la transmisión y recepción por medio de antenas sobre el espacio, también llamada transmisión inalámbrica.

En las redes LAN encontramos ambos tipos de medios, quedando los parámetros que definen cualquier transmisión determinados, no sólo por el tipo de señal intercambiada, sino también por el medio sobre el que la misma viaja. En medios guiados, es el propio medio el que define limitaciones en la transmisión. En medios inalámbricos, es la señal emitida la que impone las condiciones.

En ambos casos, uno de los objetivos más importantes es maximizar la distancia y la velocidad de transmisión. Existen factores relacionados tanto con el medio como con la señal que determinan estos límites máximos:

- Ancho de banda: es el rango de frecuencia que ocupa la señal en el espectro. En general, a mayor ancho de banda, mayor es la posibilidad de aumentar la velocidad de transmisión.

- Atenuación: se trata de la pérdida de energía de la señal en su viaje hacia el receptor. En medios guiados, la ley que rige esta pérdida es logarítmica, por lo que se suele expresar en decibeles (dB). En medios no guiados, la relación de pérdida depende de una manera más compleja de la distancia y, según el entorno, de factores atmosféricos. En cualquier caso, la atenuación es proporcional a la frecuencia de transmisión.

- Interferencias: son señales no deseadas, en bandas de frecuencia cercanas, que pueden generar problemas de distorsión o destrucción de la información. En medios guiados, pueden deberse a la inducción generada por cables adyacentes.

- Cantidad de receptores: Los medios guiados se pueden usar para comunicación directa entre transmisor y receptor a través de un enlace punto a punto. También pueden usarse como enlace compartido cuando existe más de un protagonista de la comunicación, como es el caso de redes LAN cableadas. En este caso, cada dispositivo agregado a la red dispone de su propio conector, cuya incorporación puede introducir fenómenos de distorsión o atenuación, afectando la distancia o la velocidad máxima alcanzable.

4.2 Medios de Transmisión Guiados

En términos generales, en los medios de transmisión guiados, la máxima velocidad alcanzada depende de la distancia y de la cantidad de receptores.

Un ejemplo de parámetros típicos de los cables más usados para aplicaciones punto a punto, se presenta en la Tabla 4.1.

Tabla 4.1 – Medios Guiados más comunes.

	Ancho de Banda	**Atenuación**	**Retardo Típico**	**Separación entre Repetidores**
Par Trenzado	0-3.5 kHz	0.2 dB/km a 1 kHz	50 µseg/km	2 km
Par Trenzado Múltiples Hilos	0-1 MHz	3 dB/km a 1 kHz	5 µseg/km	2 km
Cable Coaxial	0-500 MHz	7 dB/km a 10 MHz	4 µseg/km	1-9 km
Fibra Óptica	180-370 THz	0.2 dB/km- 0.5 dB/km	5 µseg/km	40 km

4.2.1 Par Trenzado

El cable más común y de menor costo es el Par Trenzado no Apantallado (UTP, Unshielded Twisted Pair). Existe una versión de par trenzado blindado mediante una hoja de aluminio que envuelve a los 4 pares, conocida como FTP, por *folded* o envuelto, que tiene mayor protección frente a interferencia electromagnética. Aún mejor resulta el cable STP, sigla cuya primera letra se refiere a blindaje o *shielded*, en el que además de la envoltura general, cada par es blindado en forma independiente. La Fig. 4.1 presenta las tres variantes mencionadas.

Como se puede observar en la Fig. 4.1, cada par consiste en dos cables de cobre cubiertos por un aislante que conforman un enlace de la comunicación y se encapsulan, junto a otros pares similares dentro de la misma envoltura. Debido a esta disposición, deben existir mecanismos para evitar el acoplamiento cruzado de señales eléctricas entre pares adyacentes, efecto conocido como *crosstalk*, causado por acoplamiento capacitivo entre cables. También es posible que se induzcan señales de ruido provenientes de otras fuentes causadas por radiación. Para atemperar estos efectos, cada par se trenza para reducir interferencias provenientes de pares adyacentes. Con el mismo objetivo, el trenzado tiene diferentes pasos de torsión para cada par.

El par trenzado es el cable de uso más extendido en redes de telefonía y en redes cableadas tipo LAN. En telefonía, se usa para conectar cada abonado a la central local (lazo o bucle de abonado). Tradicionalmente, este enlace fue

ideado para transportar sólo señal de voz, dentro de los $0 - 4\ kHz$ de ancho de banda. La evolución tecnológica permitió posteriormente el transporte de datos sobre el mismo par, en la banda por encima de $4\ KHz$, por medio del acceso conocido como ADSL.

Dada la diversidad de aplicaciones, la Asociación de industrias de las Telecomunicaciones, EIA/TIA, definió además varias categorías de cables tipo par trenzado, según sus prestaciones y características técnicas:

- **Categoría 1 y 2**: previstos para telefonía o transmisión de datos de baja velocidad, menos de $1\ Mbps$.

- **Categoría 3**: especificado hasta un ancho de banda de $16\ MHz$, posee niveles altos de atenuación y recaudos ante interferencia entre pares. Apto para transmisión en redes LAN a $10\ Mbps$, tipo *Ethernet* o 10 Base T.

- **Categoría 4**: especificado hasta $20\ MHz$. Los niveles de atenuación e interferencia entre pares lo hacen adecuado para transporte de señales hasta $16\ Mbps$. Prácticamente no tuvo instalación comercial.

- **Categoría 5**: cable de 4 pares para soporte de 100 BASE T, es decir para redes LAN de $100\ Mbps$ utilizando sólo dos pares.

- **Categoría 5e**: cable que cumple con especificaciones más rigurosas en cuanto a todos los tipos de interferencia entre pares, distorsión por retardo y pérdidas de retorno. Categoría apta para redes de alta velocidad, tipo LAN, que usan los 4 pares para transporte de señales de $1\ Gbps$.

- **Categoría 6**: soporta un ancho de banda de $250\ MHz$ y también permite funcionamiento a alta velocidad.

El par trenzado es el menos costoso de los medios de transmisión guiados y, aunque es más limitado en términos de velocidad y distancia, es mucho más flexible en cuanto a su manipulación, prefiriéndose para el tendido de distancias cortas, como es el caso de las redes LAN.

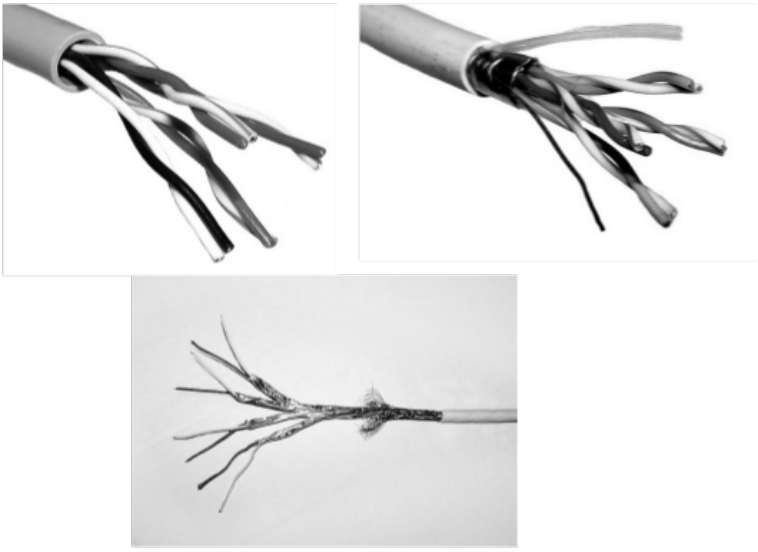


Figura 4.1 - UTP, FTP y STP.

4.2.2 Cable Coaxial

La principal limitación del par trenzado es su capacidad y un fenómeno que se produce a mayores frecuencias de transmisión, conocido como efecto *skin*. A medida que la velocidad de la señal transmitida aumenta, la corriente que circula en un cable tiende a hacerlo sobre la superficie más externa del conductor. De este modo, la sección transversal del conductor utilizable es cada vez, aumentando su resistencia eléctrica y, por consiguiente, la atenuación de la señal transmitida. También, a mayor frecuencia, mayores son las pérdidas por radiación. Por este motivo, para alcanzar altas velocidades se deben utilizar circuitos de recepción más complejos. Otra opción sería cambiar el medio de transmisión.

El cable coaxil, como el par trenzado, consta de dos cables pero dispuestos de una manera tal que permite operar en un rango más grande de frecuencias. Como se puede apreciar en la Fig. 4.2, un conductor cilíndrico interno se encuentra aislado de una malla externa por medio de un material dieléctrico. A su vez, el conductor externo se cubre de una funda para protección. Se conoce como apantallamiento a la disposición concéntrica de ambos conductores que permite reducir notablemente la vulnerabilidad frente a interferencias, desarrollar mejores distancias que el par trenzado y conectar varios receptores sobre un mismo cable compartido. De este modo, el conductor interno es efectivamente blindado frente a las interferencias externas y se reducen las pérdidas debidas a la radiación o el efecto *skin* debido a la presencia del conductor externo.

Este cable es muy utilizado para la distribución de señales de televisión y conexión a Internet en redes de TV por cable. Fue un medio guiado muy

utilizado en las redes conmutación de circuitos de larga distancia, pero paulatinamente ha sido reemplazado por tendidos de fibra óptica.

Figura 4.2 – Cable coaxial

En cuanto a las redes LAN, se trata del cable usado en las primeras instalaciones. Como se ha observado, el cable coaxial tiene una respuesta en frecuencia apropiada para transmisiones en bandas más altas y a mayores velocidades que el par trenzado y sus características constructivas lo hacen menos vulnerable a las interferencias, pero la flexibilidad en la instalación y a la versatilidad asociada a la topología de tendido del par trenzado, lo convirtieron en un reemplazo apropiado del cable coaxial.

4.2.3 Fibra Óptica

La transmisión sobre una fibra óptica difiere del par trenzado y del coaxial en que la señal transmitida es un haz de luz y el medio de transmisión no es cobre sino una fibra de vidrio. La ventaja es que las ondas de luz tienen un ancho de banda muy grande en comparación con las señales eléctricas y son inmunes a la interferencia electromagnética y al *crosstalk*. Una ventaja adicional es que son altamente seguras frente a posibles pinchaduras.

En una fibra óptica, cada señal se transmite sobre una única fibra de pocos micrones de diámetro, rodeada de un recubrimiento que puede ser otro cristal o plástico de diferentes propiedades ópticas. El recubrimiento oficia de reflector para confinar el haz de luz dentro de la hebra. Externamente y envolviendo uno o varios revestimientos, una cubierta de plástico ofrece protección contra factores ambientales agresivos. La transmisión y recepción de señales se realiza mediante diodos emisores de luz o diodos láser y fotodiodos.

Las principales ventajas comparativas de este medio guiado respecto de los previamente presentados, es su mayor capacidad en distancias mayores, su escaso tamaño y peso y su excelente comportamiento frente a la atenuación y la interferencia. Por estos motivos, su utilización es cada vez más común en grandes distancias en las redes de telefonía y presenta un futuro promisorio para redes LAN a medida que aumente la demanda de intercambio de información multimedia.

4.3 Medios de Transmisión No Guiados

En términos generales, en los medios de transmisión no guiados, la comunicación depende de la utilización de antenas. Cada antena tiene asociado un diagrama de radiación, existiendo dos grandes tipos de transmisión: omnidireccional y direccional. En el caso direccional, la antena es capaz de focalizar la energía electromagnética según la dirección de un haz, de tal manera que la antena receptora debe ubicarse alineada con la dirección del mismo. En el caso omnidireccional, la emisión no es concentrada y la señal puede recibirse sobre varias antenas separadas espacialmente.

Es importante destacar que, cuanto mayor es la frecuencia de la señal radiada, más fácil es generar un haz dirigido. Por este motivo se suele dividir el espectro en intervalos o bandas, cada uno con su propia característica de transmisión y posibilidades de aplicación:

- La banda de $30 - 300\ kHz$ se conoce con el nombre de banda Baja Frecuencia (LF, Low Frecuency). Una aplicación interesante de esta banda es la comunicación con submarinos navegando a unos 20 m de la superficie del agua. Con la comunicación en baja frecuencia se logra contrarrestar la atenuación sufrida por las ondas al viajar por el agua, evitándose el ascenso del buque a la superficie.

- La banda de $300\ kHz - 3\ MHz$ es la banda de Frecuencia Media (MF, Medium Frecuency) de utilización en radio AM comercial.

- La banda de $3 - 30\ MHz$ se conoce como banda de Alta Frecuencia (HF, High Frequency) suele utilizarse para transmisión de onda corta, modulada en AM o BLU. También en esta banda funciones los radares de vigilancia costera y, hacia el final de la misma, se aloja la denominada Banda Ciudadana, para transmisión libre de licencia.

- La banda de $30 - 300\ MHz$ es la banda de Muy Alta Frecuencia (VHF, Very High Frequency), utilizada para transmisión FM comercial, comunicación entre aviones civiles a la vista y asignación de canales marítimos internacionales.

- La banda $300\ MHz - 3\ GHz$ es la banda de Ultra Alta Frecuencia (UHF, Ultra High Frequency), a partir de la cual comienza la transmisión de microondas. Es la banda donde funcionan las redes LAN IEEE 802.11 b y g, las bandas de telefonía móvil, el Sistema de Posicionamiento Global (GPS, Global Position System), la Televisión Digital y la transmisión por *bluetooth*.

- La banda $3 - 30\ GHz$ es también de microondas, conocida como la banda de Frecuencia Super Alta (SHF, Super High Frequency), parte de

la cual se utiliza en el despliegue de redes LAN IEEE 802.11 a y n, comunicación satelital y transmisión de TV satelital.

4.3.1 Microondas Terrestres

La antena más utilizada es una parábola de diámetro cercano a los tres metros. Se trata de una transmisión direccional, con antenas ubicadas a alturas apreciables, para evitar obstáculos entre ellas, fijando una separación lo máxima posible. Existe una relación entre la altura de las antenas h, y la distancia d máxima de separación:

$$d = 7.14\sqrt{Kxh} \tag{4.1}$$

K es un factor de corrección que tiene en cuenta la desviación por la curvatura de la Tierra, que permite aumentar la distancia respecto del caso de línea directa. Un valor común es $K = 4/3$. Suponiendo antenas ubicadas a $50\ m$ de altura, resultaría una separación $d = 58.3\ km$.

Este tipo de comunicaciones suelen tenderse en reemplazo de alternativas guiadas de cable coaxial o fibra. La ventaja es que, a grandes distancias, necesitan de menor cantidad de repetidores que en el caso coaxial, aunque las antenas deben ubicarse de manera alineada.

Teniendo en cuenta la frecuencia f de transmisión, o su longitud de onda asociada λ, y la distancia de separación de las antenas d, se puede obtener la atenuación en la comunicación, expresada en dB a través de la siguiente relación:

$$L = 10\log\left(\frac{4\pi d}{\lambda}\right)^2 \tag{4.2}$$

Así, para el ejemplo previo, considerando una frecuencia de transmisión de $3\ GHz$, resultaría una atenuación $L \approx 36\ dB$, una ventaja considerable frente al cable coaxial o el par trenzado. Una desventaja puede ser la lluvia que, según a frecuencia de transmisión, genera un aumento en la atenuación. Otra desventaja puede ser la interferencia por solapamiento de áreas espaciales, por eso es muy importante la asignación por regulación del espectro.

4.3.2 Microondas Satelitales

En comunicaciones, un satélite oficia de repetidor entre dos estaciones base en tierra o entre una estación y un grupo de estaciones. El satélite recibe la señal por un canal ascendente o *uplink*, la amplifica o repite mediante un dispositivo denominado *transponder* y luego la retransmite en otra banda, sobre un canal descendente o *downlink*. El modelo apropiado para describir la

comunicación es el modelo tradicional, de un canal limitado en banda, corrupto con ruido aditivo blanco gausssiano (AWGN, Additive White Gaussian Noise) y propagación en el espacio libre.

Para que la comunicación sea eficiente, se exige que el satélite se mantenga sobre una órbita geoestacionaria, es decir que debe mantener su posición respecto de la tierra, con un período de rotación igual al de ésta y una altura fija de $35784\ m$. De este modo, las estaciones base no deben monitorizar el movimiento del satélite. Esta exigencia resulta en una comunicación de bajo costo, sobre todo en aquellas aplicaciones que requieren un gran número de antenas en Tierra, tales como la distribución de TV satelital.

El rango de frecuencias óptimo para la transmisión satelital arranca en $1\ GHz$, ya que por debajo de esta frecuencia el ruido atmosférico y el producido por otros dispositivos electrónicos y los fenómenos meteorológicos pueden dañar severamente la señal.

Para transmisiones punto a punto, el canal ascendente se ubica en la banda C $5.925 - 6.425\ GHz$ y el canal descendente en el rango $3.7 - 4.2\ GHz$. Debido a la saturación del espectro, se han desarrollado otras bandas. Por ejemplo la banda Ku $12/14\ GHz$, cuyo canal ascendente se ubica en $14 - 14.5\ GHz$. La desventaja de subir en el rango de frecuencia es la atenuación, pero los equipos terrestres son de menores dimensiones y más económicos.

Uno de los problemas asociados con la transmisión satelital es el retardo de propagación, debido a la gran distancia involucrada. Este retardo se encuentra en el orden de los $250\ mseg$, resultando apreciable en el caso de comunicaciones de voz. El retardo genera problemas en aspectos que se relacionan con el control de errores o el control del flujo de la comunicación, debiéndose tomar precauciones al respecto.

En cualquier caso, el papel de las antenas es muy importante en cuanto a radiar la energía electromagnética en la dirección correcta en el enlace directo (LOS, Line of Sight) y recibir la misma con supresión de radiación proveniente de otras direcciones, no deseada. Más allá de las ganancias asociadas a los patrones de radiación, que definen el factor de directividad de las antenas, la atenuación sigue la misma ley de la Ec. 4.2, aumentando con la separación física entre antenas, efecto que se puede compensar disminuyendo la frecuencia de la comunicación.

4.3.3 Radio

Se suele denominar ondas de radio, en contraposición con las microondas, a aquellas señales de frecuencia más baja utilizadas para transmisión no guiada en enlaces desplegados en lugar de una transmisión guiada. En el caso de las ondas de radio, la transmisión es omnidireccional, por lo que no utilizan antenas parabólicas alineadas, como el caso de las microondas. El rango de frecuencias comprende las bandas de UHF y VHF y es muy utilizado en el tendido de redes LAN inalámbricas.

Las ondas de radio son menos sensibles a la atenuación producida por la lluvia, siendo la distancia máxima ligeramente superior al alcance visual (LOS, Line of Sight) y la atenuación inferior debido a las longitudes de onda mayores.

El factor determinante en la transmisión de este tipo de ondas es la propagación multi-trayectoria, ocasionada por la reflexión con objetos o diferentes superficies, que produce efectos de desvanecimiento.

4.4 Problemas en Medios de Transmisión Guiados

El problema de ruido se reduce al acoplamiento de señales eléctricas aleatorias no deseadas. Puede ocurrir sobre circuitos de potencia o circuitos de señales, pero estos últimos son más vulnerables debido a que operan a velocidades mayores y se transmiten a menores niveles de tensión. La relación Señal a Ruido (SNR, Signal to Noise Ratio) describe cuánto ruido puede tolerar un circuito antes de que la señal deseada de información ya no pueda ser inteligible.

En industrias y fábricas es muy común la convivencia de equipos de diferente tecnología con redes de datos que, cuando la instalación no es adecuada, termina provocando problemas de compatibilidad electromagnética. Reducir el efecto negativo de esta forma de interferencia implica elegir adecuadamente los tipos de cables, su distribución, la topología y técnicas de protección.

Las principales fuentes de interferencias son aquellas generadas por acoplamiento capacitivo, acoplamiento inductivo y conducción a través de impedancia común. Esta última ocurre cuando las corrientes provenientes de dos circuitos diferentes pasan por una misma impedancia, por ejemplo en el camino de conexión a tierra común de dos circuitos.

El **acoplamiento capacitivo** se presenta por cercanía entre conductores e interacción entre campos eléctricos. En este caso, se habla de una fuente perturbadora o de ruido, cerca de la cual pasa un conductor que capta el ruido, trasladándolo hacia otra parte del circuito considerada como víctima. El efecto es debido a la capacitancia parásita entre conductores separados por un dieléctrico, que podría ser aire. En este caso, los cambios de tensión de un conductor crean un campo eléctrico que se puede acoplar con otro conductor cercano o inducir tensión en él. La tensión inducida es proporcional a las variaciones de tensión de la fuente perturbadora, a través de un coeficiente que es la propia capacidad parásita entre los cables.

El acoplamiento capacitivo aumenta proporcionalmente con la frecuencia, debido a la disminución de la reactancia capacitiva $X_c = {}^{1}/_{2\pi fC}$, y en proporción inversa a la separación entre cables. Las medidas para evitar este tipo de perturbaciones resultan en limitar la extensión de cables corriendo en paralelo, aumentando la distancia entre ellos. Lo ideal es blindar los conductores vulnerables

El **acoplamiento inductivo**, en cambio, se debe a un efecto basado en la corriente. Cualquier conductor circulado por una corriente lleva un campo magnético asociado. Una corriente variable puede inducir corriente en otro

circuito, incrementándose el efecto cuanto mayor sea el flujo de corriente, cuanto más rápida sea su fluctuación y cuanto más cerca se encuentren ambos conductores. El coeficiente de proporcionalidad en este caso es la inducción mutua entre ambos cables. Se puede reducir el acoplamiento magnético reduciendo el área del bucle representada por el cable víctima, por ejemplo trenzando el cable.

Otro tema importante es el de las **interferencias de RF**. Se trata de aquellas señales cuyas frecuencias (longitudes de onda) determinan un comportamiento de antena transmisora o receptora para los cables, debido a su propia longitud. El reemplazo de cables por fibra óptica o el blindaje de los mismos son técnicas muy usadas para reducir este tipo de interferencias.

Hacia el año 1990 empezaba a imponerse en el mercado el despliegue de redes LAN tipo Ethernet, al mismo tiempo que la industria de par trenzado había avanzado considerablemente en cuanto a técnicas de fabricación que proporcionaban aislamiento contra la diafonía interna e inmunidad contra las fuentes externas de ruido en las bandas de operación de estas redes. El cableado UTP fue el medio elegido para los diseños LAN de esa época pero, debido a las propias características constructivas, resulta ser más susceptible al ruido que el cable coaxial.

El ruido de modo común de motores eléctricos, transformadores e iluminación fluorescente, se encuentra presente en la mayoría de los entornos de instalación, presentándose como un problema debido a la falta de blindaje del cable UTP, ya que los cables actúan como antenas para estas señales.

Otro problema potencial es la interferencia por par adyacente, debida a la estrecha proximidad entre varios pares de cables. La diafonía, denominada en inglés *crosstalk* (XT), se presenta entre dos circuitos, cuando parte de las señales presentes en uno de ellos, considerado perturbador, aparece en el otro, considerado perturbado. La diafonía, en el caso de cables de pares trenzados se presenta generalmente por los acoplamientos magnéticos entre los elementos que componen los circuitos perturbador y perturbado o como consecuencia de desequilibrios de admitancia entre los hilos de ambos circuitos. Por ejemplo, este efecto lo podemos percibir en la línea telefónica cuando podemos escuchar otra conversación mientras llevamos adelante la propia, efecto que llamamos teléfono ligado.

Par balanceado

Minimizar este tipo de interferencia requiere una clase especial de señal, denominada diferencial que también impone un transmisor y un receptor especial.

La línea no balanceada también se conoce como línea no equilibrada. Se define considerando una línea de audio, en la que el retorno de la señal se produce a través de la malla exterior que cubre el conductor de ida, protegiéndolo contra interferencias electromagnéticas externas, aunque no eliminándolas completamente. Normalmente, las líneas no balanceadas no sirven en los casos en que se requiere una gran longitud de cable, ya que el efecto acumulativo de las interferencias puede producir tal nivel de distorsión que el sonido final sea de

pésima calidad. Un caso típico de línea no equilibrada es la implementada por un cable coaxial.

En una línea equilibrada o balanceada, existen dos conductores, uno de ellos denominado vivo, normalmente de color rojo, que porta la señal en fase, el otro denominado retorno, normalmente de color negro, porta la misma señal, pero desfasada 180º. Este par de conductores va cubierto por una malla conectada a masa. Con esta disposición, se logra mejorar la respuesta ante las interferencias que ofrece la línea no balanceada de audio. La diferencia entre ambos casos es considerable, pudiendo llegar a relaciones cercanas a los 80 dB.

En el receptor, para des-balancear la línea, hay que invertir la señal que porta la contrafase y sumarla a la que está fase. Esto genera el mismo efecto que restar ambas señales, logrando así duplicar la amplitud de la señal resultante. La mejora del par balanceado se fundamenta en que, si una interferencia logra atravesar la malla, induce el mismo transitorio en ambos conductores. Al invertir la contrafase, el transitorio queda invertido también, y al sumarlo con la fase, éste se anula.

Al tipo de interferencia descripta, se la conoce como señal de modo común. Precisamente, en muchas especificaciones aparece la relación de rechazo del modo común cuyo valor debe ser al menos de 80 dB. Los casos típicos de líneas balanceadas son las líneas de par trenzado.

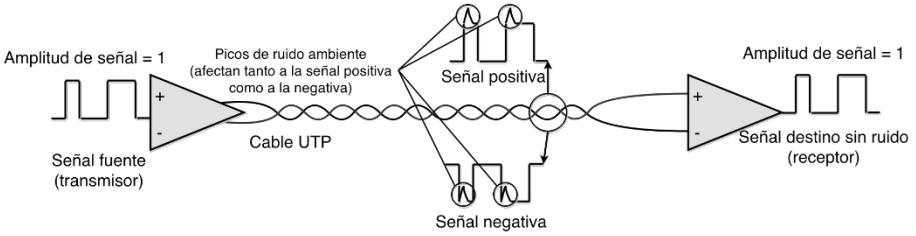


Figura 4.3 - Transmisión balanceada

Básicamente, la transmisión diferencial o balanceada funciona como se indica en la Fig. 4.3. En el modo diferencial, cuando el transmisor recibe una señal de la fuente, se crea un complemento de la señal o espejo, que se envía a través del par balanceado al receptor. El receptor diferencial recibe las señales de entrada balanceada, positiva y negativa, y utiliza la diferencia entre los dos para formar una señal de salida. Cualquier ruido inducido de modo común desde otras fuentes, implica que la misma señal de ruido se presente en los dos cables. El receptor procesa este ruido de modo común junto con la señal pero, puesto que la amplitud y la polaridad del ruido inducido son los mismos en ambos cables, el ruido se cancela y sólo la señal original es la que finalmente recibe el receptor. Esto se cumple tanto para los transmisores o receptores activos como para las soluciones pasivas (balun/transformador).

Por ejemplo, en la Fig. 4.3, se puede apreciar que el transmisor toma la señal de entrada y la envía equilibrada a lo largo del cable de par trenzado. El

receptor diferencial utiliza la diferencia entre las señales para producir la señal de salida. El resultado es una señal libre de ruido.

A su vez, en el caso de emisión de señales, el campo creado por uno de los conductores del par será igual, pero de signo opuesto al del otro conductor, por lo que se produce un efecto de cancelación sobre cualquier otro conductor, disminuyendo los efectos de la radiación del par.

Par trenzado

Como se ha mencionado, si dos cables recorren un mismo camino, la proximidad entre ellos creará acoplamiento entre ambos, tanto del tipo magnético como eléctrico.

Reducir el acoplamiento magnético, involucra limitar la longitud del recorrido de cables paralelos, o aumentar la distancia entre el cable perturbador y el cable víctima. Para minimizar el efecto de inducción, también se puede usar un cable de par trenzado, que reduce el área de exposición a la inducción y el efecto de la tensión inducida. Otra forma importante de protección la ofrece el blindaje

El acoplamiento eléctrico puede ser visualizado como un componente concentrado, representado por un capacitor, cuyo valor dependerá de la separación y el dieléctrico entre los conductores. Si otro par, por ejemplo otra línea de alambres paralelos, es interferido por este efecto, uno de los conductores estará más próximo que el otro del par y por lo tanto presentará una capacidad mayor. La corriente capacitiva entre cada uno de los conductores del par y tierra será distinta, con lo que en el receptor captará la diferencia entre los mismos.

Una forma simple de lograr que la separación sea la misma, es mediante el trenzado de los cables. De esta forma se alternan el par próximo con el lejano, para producir el mismo efecto, y poder cancelarlo en el receptor.

Análogamente, el campo magnético inducirá en cada par una tensión que depende del área encerrada entre el conductor y tierra. El par trenzado ayuda igualmente en este sentido para producir la cancelación de la interferencia.

Por lo expuesto, se debe usar par trenzado con transmisor y receptor balanceados, con iguales impedancias entre pares y tierra, para poder reducir las interferencias.

Adicionalmente, por cuestiones de seguridad, tanto de personas como equipos, debe preverse el contacto accidental con las líneas de electricidad de $220\ Volts$, de modo que en cada extremo debe haber un transformador que aísle los cables de red y los equipos, para que la eventual aparición de tensión de línea de la red eléctrica no se propague por la red de cables UTP.

4.4.1 Cables – Especificaciones Técnicas

Los trabajos de instalación de una red culminan con una verificación técnica, con el objetivo de asegurase el cumplimiento de los estándares. Se mencionan a continuación algunos de los ensayos que son posibles de realizar

sobre la instalación y parámetros cuyo significado se debe entender conceptualmente.

PÉRDIDAS DE RETORNO

Las pérdidas de retorno estructurales son una medida de la uniformidad en cuanto a la impedancia que presentan constructivamente los cables y el tendido de la red. Los cambios de impedancia más acentuados se producen en los puntos de interconexión, conectores de las áreas de trabajo y paneles de interconexión.

El primer efecto de estas desadaptaciones es aumentar la pérdida de inserción, lo que se traduce en menor potencia de señal en la salida del cable, agravando el problema de atenuación. También, la señal reflejada por desadaptación se sumará como ruido a la señal incidente. En casos de enlaces *full dúplex* este efecto es directo, como sucede en las redes *Gigabit Ethernet*, para las que se recomienda el uso de cable Categoría 5E o superior. Otro efecto indeseable, puede ser el de la señal re-reflejada, que viaja en el mismo sentido que la señal incidente, pero llega a destino retrasada respecto de esta.

Las pérdidas de retorno se expresan en dB y cuánto mayor es su valor, mejor es el cable desde el punto de vista de su fabricación. Lo que se trata de obtener es una medida de las discontinuidades de impedancia respecto al valor nominal ($100\,\Omega$ en par trenzado UTP), pero teniendo en cuenta conectores y uniones.

En definitiva, la pérdida de retorno se relaciona con la magnitud del coeficiente de reflexión expresado en dB. La reflexión de la señal modifica la forma de los pulsos en el receptor, aumentando la tasa de errores. Según el cable, los valores típicos por encima de los cuales debería estar trabajando el sistema varían. En el caso de cables Categoría 5E, $15\,dB$ es un valor típico.

ATENUACIÓN

Otro parámetro que se mide en dB . La atenuación depende fundamentalmente de la longitud y el diámetro del conductor. Aumenta en función de la frecuencia, principalmente por efecto *Skin* y pérdidas en el dieléctrico. El efecto *Skin* se produce en alta frecuencia, cuando la corriente tiende a circular por el borde de los conductores, disminuyendo el área efectiva de conducción y aumentando, de este modo, su resistencia, y la pérdida de señal, de manera proporcional a la raíz cuadrada de la frecuencia.

La atenuación también depende del diámetro del conductor y de si éste es multi-hilo o cobre sólido. Es mejor el conductor sólido ya que el multi-hilo presenta niveles de atenuación de entre 20% a 50% mayores.

La atenuación se especifica a una determinada temperatura ($20\,ºC$) debido a que los materiales dieléctricos utilizados para forrar los conductores, varían su comportamiento con el aumento de temperatura, aumentando las pérdidas y, por consiguiente, la atenuación. Por ejemplo, en cables Categoría 3 la atenuación aumenta a razón de $1.5\%/ºC$, en cables Categoría 5 el aumento es menor y se encuentra en el orden de $0.4\%/ºC$.

En cables UTP, la atenuación se mide sobre el mismo par, en ambos extremos.

Por ejemplo, la pérdida de inserción, o atenuación máxima permitida en una red LAN de $10\,Mbps$ 10 BASE T es $11.5dB$ a frecuencias entre 5 y $10\,MHz$. Este valor incluye conectores, *patch panels* (paneles de cruce) y pérdidas por reflexión debidas a desadaptación en el segmento.

NEXT

La diafonía o *crosstalk* se debe a la interferencia electromagnética de cada par de transmisión sobre los pares cercanos. Dado que el cableado más común para redes LAN consiste en cables de 4 pares, una de las mayores fuentes de ruido proviene de los pares adyacentes. El *crosstalk* depende de la frecuencia de la señal y de la geometría de los cables. Se mide como la potencia de la señal de interferencia respecto a la potencia de la señal transmitida.

La sigla NEXT es por Pérdida por Interferencia sobre Extremo Cercano (Near End CrossTalk Loss). Se trata de un parámetro que se mide sobre el mismo extremo, utilizando los pares de transmisión y recepción, tal como se indica en la Fig. 4.4. Se observa que, en la recepción, el efecto se atenúa según el largo del cable desde el extremo transmisor

La medición de NEXT tiene en cuenta las inducciones que se producen por proximidad entre los pares. Este efecto se traduce en una degradación de la relación Señal a Ruido SNR.

Para disminuir el NEXT se suele utilizar configuración balanceada en los pares. De este modo, toda interferencia que llegue a ambos conductores a la vez se cancelará debido a que el sistema admite sólo señales en modo diferencial. Lo mismo sucede cuando se emiten señales. El campo de un conductor será igual pero opuesto al del otro conductor y se producirá un efecto de cancelación impidiendo la emisión y por lo tanto eliminando las pérdidas.

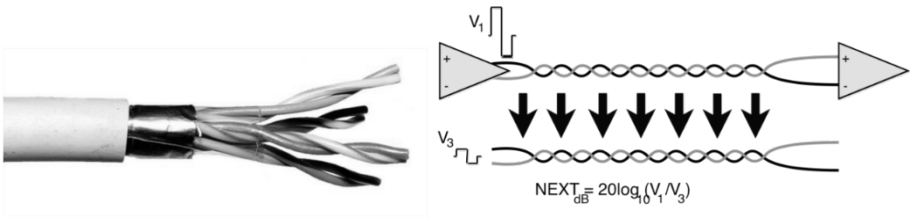


Figura 4.4 - Medición NEXT

Se debe utilizar una derivación central en los transformadores de acoplamiento para conexión a tierra. Los pares son trenzados para que los acoplamientos sean homogéneos y de este modo se cancelen en el transformador de entrada. A mayor trenzado, mejor es la cancelación y se pueden conseguir velocidades mayores.

El efecto NEXT se debe medir en el rango de frecuencia especificado para el cable.

FEXT/ELFEXT

La Interferencia de Extremo Lejano (FEXT, Far End Crosstalk) es el acoplamiento entre dos o más pares de transmisión, como se indica en la Fig. 4.5. Se puede expresar como FEXT o como Interferencia de Extremo Lejano de Igual Nivel (ELFEXT, Equal Level Far End Crosstalk), también presentada en la figura. Ambos acoplamiento se miden en dB, pero la medición de FEXT arrojará diferentes resultados, dependientes de la longitud del cable. Cuanto más corto sea el cable, mayor será el valor de FEXT. Por este motivo se define también la interferencia ELFEXT, ya que permite una medición independiente de la longitud.

Por lo tanto, FEXT y ELFEXT se refieren al mismo acoplamiento pero medido respecto de distintas referencias. FEXT se mide con respecto a una señal perturbadora. ELFEXT, en cambio, se mide con respecto a la señal perturbadora atenuada. Si se sustrae FEXT de ELFEXT, se obtiene como resultado la atenuación del canal. Es decir que ELFEXT implica un cálculo que normaliza los resultados de la medición del FEXT, ya que toma en cuenta la atenuación.

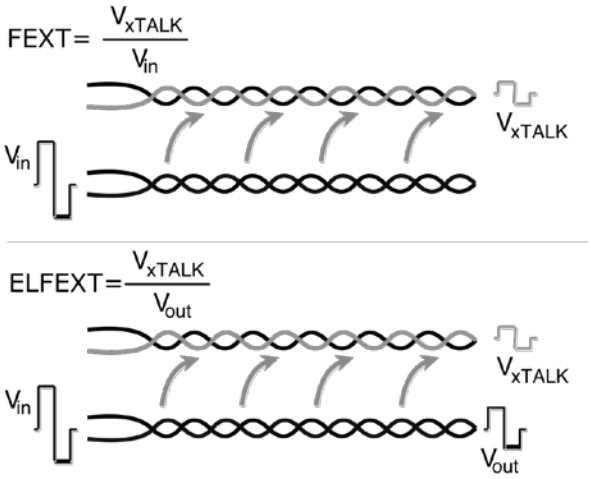


Figura 4.5- Medición *FEXT/ELFEXT*

PSNEXT

La Suma de Potencia NEXT (PSNEXT, Power Sum NEXT), aplicada al caso de un cable de 4 pares, es la suma algebraica del NEXT individual sobre cada uno de los pares, producidos por los otros tres. Se trata de un cálculo, no de una medición, tal como se indica en la Fig.4.6.

PSNEXT es importante para la calificación de cables para soporte de *Gigabit Ethernet*. Dado que PSNEXT es una medida de la diferencia de potencia de la señal entre pares perturbadores y un par perturbado, un número alto (menos *crosstalk*) es más deseable que un número más pequeño (más *crosstalk*). PSNEXT varía significativamente con la frecuencia.

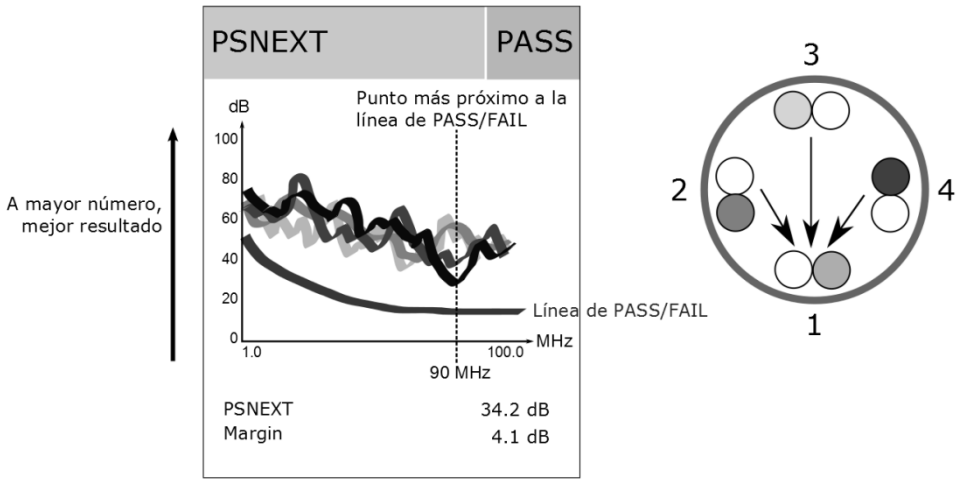


Figura 4.6 - Medición *PSNEXT*.

DC LOOP RESISTANCE

Se trata de la resistencia de lazo de continua. Es una medida de la pérdida lineal en continua de un conductor, que depende de la longitud del cable y aumenta con ésta. Un valor típico puede ser de $9.4\ \Omega$ cada $100\ m$. Usualmente tiene menos efecto que la pérdida de inserción.

DELAY SKEW

Para aprovechar el máximo ancho de banda en un cable UTP de 4 pares, los estándares para velocidades superiores a $10\ Mbps$, dividen la señal a transmitir entre los 4 pares. El receptor debe reconstruir la señal tomando lecturas de los 4 pares de manera simultánea. Por esta razón, es importante que las señales lleguen al extremo lejano a la vez, o con diferencias de tiempo mínimas.

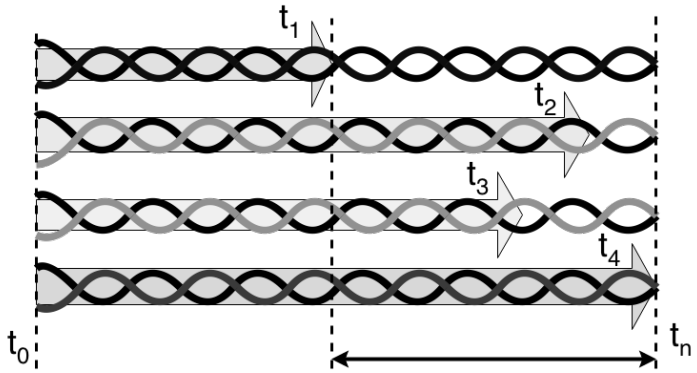


Figura 4.7 - Distorsión por Retardo.

Como se observa en la Fig. 4.7, cada par lleva asociado un retardo de propagación de la señal. La Distorsión por Retardo o *Delay Skew* considera la diferencia máxima en el retardo de propagación entre pares de un sistema.

El retardo típico para un cable UTP Categoría 5 es de 5.7 $nseg/m$. Se exige que no supere 1 μseg en un enlace de 100 m.

DESBALANCE DE CONTINUA

EIA/TIA 568A impone un límite superior de capacidad de conductores del mismo par de $5.6nF/100m$ para cable Categoría 5.

IMPEDANCIA CARACTERÍSTICA

Los cables Categoría 5 tienen una impedancia característica nominal de 100 Ω. La especificación requiere que la impedancia no se desvíe de este valor en más de $+/-$ 15 Ω para frecuencias por encima de 1 MHz.

4.5 Problemas en Medios de Transmisión No Guiados

Las comunicaciones inalámbricas se rigen por modelos de canal muy diferentes a los de las comunicaciones por medios de transmisión guiados, ya que se caracterizan por efectos propios tales como el **problema de desvanecimiento**.

En el caso de medios no guiados, la presencia de objetos reflectores en el entorno donde se encuentran transmisor y receptor, implica la existencia de múltiples caminos para la propagación de la señal. En el lado receptor este efecto se traduce en una superposición de copias de la señal transmitida, cada una arribando a través de un camino diferente, experimentando entonces una atenuación distinta, así como retardo y corrimientos de fase diferentes. El resultado puede ser destructivo o constructivo en el sentido de la potencia recibida, ya que puede resultar en nulos de señal o en efectos de amplificación,

según las circunstancias. Los nulos se conocen como efecto de desvanecimiento o *fading*.

Como oyentes de radio o usuarios de teléfonos celulares, muchas veces se puede notar este efecto con el movimiento del receptor. En las redes inalámbricas WLAN (Wireless LAN) sucede lo mismo. La movilidad, uno de los objetivos del desarrollo de estas redes, exige técnicas especiales para la compensación de un canal variable en el tiempo debido a la propia movilidad de los actores de la comunicación.

Trabajar sobre un modelo apropiado de canal de *fading*, permitió a los ingenieros comprender mejor el comportamiento de estas redes y hallar soluciones para mitigar el efecto del desvanecimiento. Generalmente, el modelo estadístico del canal incluye un cambio aleatorio en la amplitud y fase de la señal transmitida.

Una de las técnicas más usadas para combatir el efecto del fading se conoce con el nombre de diversidad o *diversity*, y consiste en transmitir la señal sobre canales múltiples, que experimenten desvanecimientos independientes, para luego combinarlos de manera coherente en el receptor. De este modo, se logra disminuir la probabilidad de sufrir el efecto destructivo del desvanecimiento, ya que resulta menos probable que todos los componentes de los canales múltiples experimenten *fading* al mismo tiempo. La técnica se puede realizar en el tiempo, en frecuencia o en el espacio.

Otro de los problemas deviene da la propia banda de funcionamiento de las redes LAN inalámbricas. Muchas de estas redes se pueden desplegar en el espectro Industrial Científico y Médico (ISM, Industrial, Scientific and Medical) de $2.4\ GHz$. En esta banda existen problemas de ruido e **interferencia** debido a la presencia de equipos interferentes, tales como hornos microondas o teléfonos inalámbricos.

También la presencia de otras redes inalámbricas en las cercanías de la propia puede ser fuente de interferencia. Para evitar el solapamiento entre diferentes redes se debe adoptar una disposición de canales de funcionamiento coherente para que los espectros no se vean alterados y la comunicación se mantenga dentro de los parámetros establecidos.

Los síntomas de interferencia pueden notarse en ciertas señales de advertencia, tales como el rango de alcance notablemente reducido, caídas abruptas en la velocidad de transferencia, pérdidas durante ciertos momentos del día o en determinadas ubicaciones y variaciones aleatorias en la potencia recibida.

Un problema propio de las redes inalámbricas es el fenómeno del **nodo oculto**. En un sistema de red fija o cableada en condiciones normales de funcionamiento, la transmisión por ese medio garantiza que todos los integrantes de la red reciban la información transportada, una vez que hayan accedido al cable. En una red inalámbrica, por cuestiones de alcance, algunas estaciones podrían no estar completamente conectadas entre sí. El problema del nodo oculto se refiere a la situación que se presenta en redes inalámbricas cuando existen nodos fuera del rango de alcance de otro, o de otros nodos. La situación se describe gráficamente en la Fig. 4.8. Debido a la disposición de los dispositivos, para el nodo A de la figura, el nodo C es un nodo escondido, ya que se encuentra fuera de su alcance. Si simultáneamente A y C transmitieran un mensaje a B se

produciría una colisión de ambos mensajes sobre B, muy difícil de detectar debido a la atenuación propia del medio.

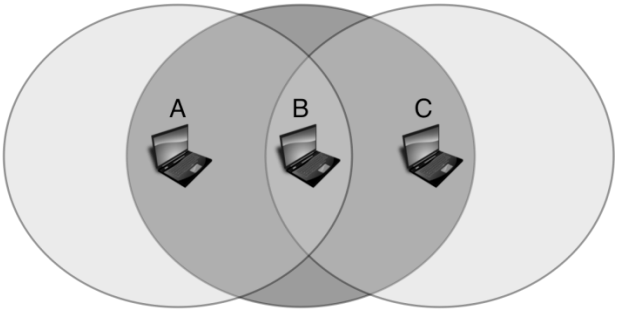


Figura 4.8 - Problema del Nodo Oculto.

Para solucionar este problema se implementa un protocolo conocido con el nombre de apretón de manos o *handshaking*, que complementa el método de acceso al medio elegido para este tipo de redes.

4.5.1 Propagación en canales de radio móvil - Fading

El punto de partida más común para la evaluación del comportamiento de los sistemas de comunicaciones es la suposición de un canal contaminado con Ruido Blanco Gaussiano Aditivo (AWGN, Additive White Gaussian Noise), con muestras de ruido estadísticamente independientes, que deforman muestras de datos libres de Interferencia Inter-simbólica (ISI, Intersymbol Interference).

Aunque la principal fuente de degradación es el ruido térmico generado en el receptor, muchas veces la interferencia externa captada por las antenas tiene un efecto más significativo que el ruido térmico. Este tipo de interferencias externas se puede caracterizar como de banda ancha y cuantificar por medio de un parámetro denominado temperatura de la antena. El ruido térmico usualmente posee una densidad espectral de potencia plana en la banda de la señal y se le asocia una función densidad de probabilidad (pdf, probability density function) Gaussiana, de valor medio nulo.

Al diseñar el sistema, el filtro transmisor generalmente obedece al cumplimiento de requerimientos regulatorios del espectro. Por su parte, la mayoría de las veces, el filtro receptor es un filtro adaptado. El filtrado introduce limitación en banda y distorsión de fase, induciendo ISI, por lo que es posible que se utilicen técnicas de ecualización y de conformado de señales, para mitigar el efecto de ISI inducido.

Si no se especifican las características de propagación del canal de radio, usualmente se infiere que la atenuación de la señal en función de la distancia tiene un comportamiento similar al de la propagación en el espacio libre. El modelo de

espacio libre considera la región entre transmisor y receptor libre de objetos que puedan absorber o reflejar las señales de radiofrecuencia, y a la atmósfera como un medio que se comporta de manera perfectamente uniforme y no absorbente. Por su parte, la tierra se considera infinitamente alejada de la señal que se está propagando o, equivalentemente, como si tuviera un coeficiente de reflexión despreciable. En este modelo idealizado, la atenuación de la energía de RF entre transmisor y receptor, se comporta de acuerdo a una ley inversa del cuadrado de la distancia entre ambos, siendo predecible la potencia de la señal recibida.

Para la mayoría de los canales prácticos inalámbricos de radio móvil, este modelo no es adecuado para describir el comportamiento del canal ni para predecir el comportamiento del sistema. Para poder analizar este tipo de canales se recurre al estudio estadístico complementado con mediciones prácticas. La idea es poder predecir la potencia mínima que se precisa radiar desde el lado transmisor para poder ofrecer un área de cobertura de calidad aceptable sobre el área de servicio establecida. También interesa cuantificar los parámetros de variabilidad de la señal, característicos de canales con desvanecimiento (*fading*).

Un ejemplo típico de este tipo de comunicación inalámbrica es la red de telefonía celular. Un modelo idealizado de esta red es el de un panal de abejas, donde cada celda posee una estación base conectada a un centro de conmutación mediante líneas dedicadas, con un alcance específico, que oficia de interfaz entre los usuarios y el sistema de radio celular. El concepto se aplica de manera bastante parecida a los esquemas de redes LAN inalámbricas.

Una de las características más importantes de este tipo de redes es que permite el movimiento de usuarios entre celdas sin pérdida de conectividad. Con este motivo, además de la división del espacio en celdas, el sistema se caracteriza por permitir la re-utilización de frecuencias: el uso del mismo canal de radio, sobre la misma portadora, en diferentes áreas separadas físicamente lo suficiente como para no interferir. De este modo es posible extender el área de cobertura mediante celdas.

Uno de los problemas más importantes de este canal de comunicaciones es que, debido al propio movimiento del usuario y al entorno circundante, puede no haber línea directa entre éste y la estación base, y la propagación establecerse mediante más de camino, dependiendo de los siguientes efectos:

- **Reflexión**: ocurre cuando la onda electromagnética impacta sobre una superficie suave de grandes dimensiones comparadas con la longitud de onda λ.

- **Difracción**: ocurre cuando el camino entre transmisor y receptor se ve obstruido por un cuerpo denso de grandes dimensiones comparadas con λ, haciendo que se generen ondas secundarias. En estos casos, la señal de RF viaja sin Línea de Vista (LOS, Line of Sight). A menudo se conoce como oscurecimiento (*shadowing*) porque la onda difractada puede alcanzar al receptor, aún cuando este se encuentre oscurecido por un obstáculo.

- **Scattering**: ocurre cuando la onda impacta o sobre una superficie grande y rugosa o sobre una superficie cuyas dimensiones están en el orden de λ o menos, causando que la energía reflejada se desparrame (*scatter*) en todas direcciones. En entornos urbanos, objetos como postes de luz, las señales de las calles y las copas de los árboles, pueden producir este fenómeno.

De este modo, en un sistema de comunicaciones móviles, la señal puede viajar desde transmisor a receptor sobre múltiples caminos de reflexión. Este fenómeno se conoce como propagación multi-camino. Su consecuencia más evidente son probables fluctuaciones en la amplitud, la fase y el ángulo de llegada de la señal recibida, dando lugar al efecto conocido como desvanecimiento multi-camino (*multipath fading*).

La propia movilidad de los actores agrega a este efecto posibles corrimientos en frecuencia de varias componentes de la señal, fenómeno conocido como corrimiento Doppler.

4.5.2 Escalas de Fading

Existen fundamentalmente dos tipos de efectos de *fading* que caracterizan las comunicaciones móviles. Se suele mencionar como *fading de pequeña escala* al efecto de desvanecimiento debido a pequeños cambios de posición, y como *fading de gran escala*, al provocado por movimientos en áreas grandes.

El *fading* de gran escala representa la atenuación promedio de la potencia de la señal, o pérdida de camino, debido al movimiento sobre áreas grandes. Este fenómeno es provocado por los contornos de los terrenos entre transmisor y receptor (colinas, bosques, grupos de edificios, etc.). A menudo se representa el receptor como ensombrecido por estos accidentes geográficos.

Por su parte, el *fading* de pequeña escala se refiere a cambios dramáticos de la amplitud y fase de la señal como resultado de pequeños cambios, del orden de la mitad de la longitud de onda, en el posicionamiento espacial entre transmisor y receptor. Este tipo de *fading* se manifiesta en dos mecanismos: desparramo en el tiempo de la señal, también llamado dispersión, y variación del comportamiento del canal en el tiempo.

Para aplicaciones de radio móvil, el canal varía en el tiempo por el movimiento entre transmisor y receptor, ya que se modifica el camino de propagación entre ellos. La relación de cambio de las condiciones de propagación se conoce con el nombre de rapidez de fading.

El *fading* de pequeña escala se denomina también *fading* de Rayleigh, debido a que, cuando existen múltiples caminos de reflexión y no existe señal directa, la envolvente de la señal recibida se puede describir estadísticamente con una función densidad de probabilidad tipo Rayleigh. Cuando se encuentra presente una señal componente dominante, tal como la de camino directo, la

envolvente de *fading* de pequeña escala se puede describir con una función distribución de probabilidad de Rician.

Una estación móvil desplazándose (*roaming*) sobre un área grande, debe procesar señales que experimentan ambos tipos de *fading*: el de pequeña escala superpuesto al de gran escala.

Las dos manifestaciones de *fading* de pequeña escala, dispersión en el tiempo de la señal y naturaleza variable en el tiempo del canal, se pueden examinar en dos dominios: tiempo y frecuencia. Para la dispersión de la señal, los tipos de degradación por *fading* se caracterizan por ser selectivos en frecuencia o no selectivos en frecuencia (plano o *flat*). Para el canal variante en el tiempo, la categorización es por rápido o lento, *fast fading* o *slow fading*.

Se puede pensar el efecto superpuesto de ambos componentes de *fading* a través de un modelo. También se lo puede observar por medio de una medición de potencia en función del desplazamiento de la antena receptora.

Suponiendo que se desprecia la degradación debida al ruido, cualquier señal recibida $r(t)$ se puede describir en términos de la señal transmitida $s(t)$ y la respuesta al impulso del canal $h_c(t)$, a través de su convolución:

$$r(t) = s(t) * h_c(t) \tag{4.3}$$

En el caso de radio móvil, $r(t)$ se puede expresar en término de dos variables aleatorias componentes:

$$r(t) = m(t)x\, r_0(t) \tag{4.4}$$

En la ecuación, $m(t)$ es la componente de *fading* de gran escala y $r_0(t)$ la componente de *fading* de pequeña escala. Algunas veces referimos a $m(t)$ como el *fading* log-normal, porque la magnitud de $m(t)$ se describe con una pdf log-normal o, equivalentemente, la magnitud medida en dB se asocia a una pdf Gaussiana. Por su parte, $r_0(t)$ se refiere como *fading* de Rayleigh.

En la Fig. 4.9, se grafica la potencia de la señal recibida $r(t)$ en función del desplazamiento de la antena, típicamente medido en unidades de λ. Se puede identificar el *fading* de pequeña escala superpuesto al de gran escala. El desplazamiento típico de la antena entre nulos de la señal de pequeña escala es aproximadamente de $\lambda/2$. Si se removiera el valor medio $m(t)$, se podría apreciar sólo el *fading* de pequeña escala $r_0(t)$.

En general, el modelado y diseño de sistemas que incorporan técnicas para mitigar los efectos del desvanecimiento representan mucho mayor desafío que aquellos cuya única fuente de degradación es el ruido AWGN.

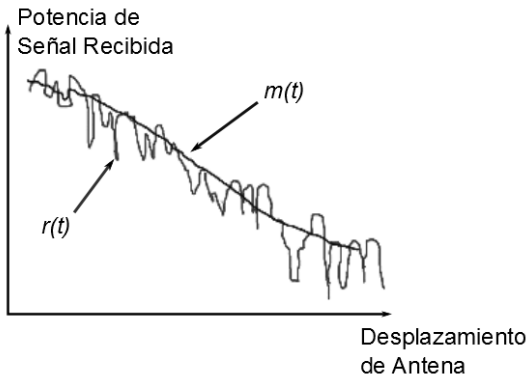


Figura 4.9 - Fading de grande y pequeña escala.

4.5.3 Fading de Gran Escala

El efecto de este tipo de *fading* puede ser apreciado por un usuario moviéndose dentro de un edificio o entrando al mismo o al dar vuelta una esquina. Interesa calcular en este caso una estimación de las pérdidas generadas por el propio entorno de la comunicación.

Okumura realizó una de las primeras mediciones más entendibles de las pérdidas de camino para un rango muy grande de altura de antenas y de distancias de cobertura. Hata, por su parte, transformó las mediciones realizadas por Okumura en fórmulas paramétricas, expresando la pérdida promedio de camino, en función de la distancia d entre transmisor y receptor, y una distancia de referencia d_o, correspondiente a un punto localizado en el campo lejano de la antena. Las pérdidas resultan en relación exponencial n-ésima con respecto a la distancia:

$$\overline{L_p(d)} \propto (\frac{d}{d_o})^n. \tag{4.5}$$

Valores típicos para d_o van desde $1\ km$ para grandes celdas, $100\ m$ para microceldas y $1\ m$ para canales dentro de edificios. El valor de n depende de la frecuencia, de la altura de las antenas y del entorno de propagación, adoptándose valores como $n = 2$ en el espacio libre. En presencia de entornos que asemejan guías de onda, como podría ser el caso de las calles de una ciudad, n puede ser menor. Cuando hay obstrucciones, es mayor.

De todas maneras, la expresión no deja de referirse a un valor promedio, por lo que no resulta adecuada para describir una situación o camino particular. Es necesario proveer variaciones alrededor de la media, ya que el entorno en diferentes sitios puede ser distinto para similar separación entre transmisor y receptor. Se ha demostrado que, para cualquier valor de d, $Lp\ (d)$ es una VA con

distribución log-normal alrededor de su valor medio $\overline{L_p(d)}$, que se puede expresar como:

$$Lp\,(d) = L_s(d_o)(dB) + 10nlog\left({}^{d}/_{d_o}\right) + X_\sigma(dB) \tag{4.6}$$

Así, el fading de gran escala, al seguir una distribución log-normal, expresado en dB se convierte en una distribución Gaussiana. Por este motivo, los efectos del oscurecimiento se incorporan a la estimación de las pérdidas con la adición de una VA Gaussiana de valor medio nulo y desviación estándar σ. En la expresión (4.6), X_σ es la VA Gaussiana expresada en dB, con desviación estándar σ, también expresada en dB. Se trata de una VA dependiente del sitio y de la distancia, que usualmente presenta valores entre 6 y 10 dB. La pérdida $L_s(d_o)$ se puede medir o estimar con la Ec. (4.2). El valor de n se adopta según el entorno.

De este modo, los parámetros que se necesitan para describir estadísticamente la pérdida debida al *fading* de gran escala, para una localización arbitraria, con una separación específica entre transmisor y receptor, son: la distancia de referencia d_o, el exponente de pérdidas n, y la desviación estándar σ de X_σ.

4.5.4 Fading de Pequeña Escala - Modelo de Canal Variante en el Tiempo

El movimiento introduce un factor de variación en el tiempo de la propia estructura del medio. Como consecuencia de estas variaciones, la respuesta del canal a cualquier señal transmitida es variable en el tiempo. Si se repitiera la transmisión de un pulso, siempre se verían variaciones en la señal recibida, variaciones consideradas impredecibles para el sistema de recepción. Se dice que la respuesta al impulso del canal varía en el tiempo y la mejor manera de describir su comportamiento es mediante un modelo estadístico.

Se desarrollará una expresión apropiada para el efecto de *fading* de pequeña escala, que se ha expresado como $r_0(t)$ en la Ec (4.4). En el análisis se supone que el efecto de fading de gran escala $m(t)$ es constante y se fija en el valor 1, debido a que la antena permanece dentro de una trayectoria limitada.

Sin considerar el efecto de ruido, suponiendo que la antena se mueve y que existen n múltiples caminos por efecto del *scatter*, cada uno con su propio retardo de propagación variante en el tiempo $\tau_n(t)$, afectado por un factor de ganancia o de pérdida también dependiente del tiempo $\alpha_n(t)$, la señal recibida se podría expresar como:

$$r(t) = \sum_n \alpha_n(t)s[t - \tau_n(t)] \tag{4.7}$$

En la expresión, $s(t)$ es la señal transmitida:

$$s(t) = Re\{g(t)e^{j2\pi f_c t}\} \tag{4.8}$$

Reemplazando esta expresión en la anterior, resulta:

$$r(t) = Re(\{\sum_n \alpha_n(t) g[t - \tau_n(t)]\}\, e^{j2\pi f_c (t - \tau_n(t))}) \tag{4.9}$$

$$r(t) = Re(\{\sum_n \alpha_n(t) e^{-j2\pi f_c (\tau_n(t))} g[t - \tau_n(t)]\}\, e^{j2\pi f_c t}) \tag{4.10}$$

Se puede deducir que la señal recibida en banda base $z(t)$, tendrá la forma:

$$z(t) = Re(\{\sum_n \alpha_n(t) e^{-j2\pi f_c (\tau_n(t))} g[t - \tau_n(t)]\}) \tag{4.11}$$

Si se considera el caso de una transmisión de una portadora no modulada de frecuencia f_c , sería como considerar todo el tiempo $g(t) = 1$. Bajo esta suposición, la señal recibida en banda base sería:

$$z(t) = \sum_n \alpha_n(t) e^{-j2\pi f_c (\tau_n(t))} = \sum_n \alpha_n(t) e^{-j\theta_n(t)} \tag{4.12}$$

Es decir que la señal en banda base $z(t)$ está conformada por la suma de fasores variantes en el tiempo, con amplitudes $\alpha_n(t)$ y fases $\theta_n(t) = 2\pi f_c \tau_n(t)$. En general, generar un gran cambio en $\alpha_n(t)$ implica cambios drásticos en el entorno físico, pero no sucede lo mismo con $\theta_n(t)$. Las fases cambiarán en 2π radianes cada vez que $\tau_n(t)$ varíe en $1/f_c$. Se trata de retardos muy pequeños, por ejemplo para $f_c = 2.4\ GHz$ es apenas de $0.42\ ns$. En el espacio libre, esto se corresponde con una distancia de $12.5\ cm$. Es decir que la fase $\theta_n(t)$ puede cambiar de manera significativa con relativamente poco cambio en el retardo de propagación. En este caso, cuando dos componentes *multipath* de la señal difieren en $6.25\ cm$ en cuanto a la longitud de sus caminos, una componente llega con 180 º de diferencia de fase respecto de la otra. Obviamente, algunas veces se adicionan componentes de manera constructiva y otras de manera destructiva. El efecto final es una variación de amplitud, denominado desvanecimiento o *fading* de $z(t)$. El modelo de propagación multi-camino detrás de $r(t)$ o de $z(t)$ es el

modelo de un canal con *fading*, donde las variaciones en la amplitud de la señal recibida se deben fundamentalmente a las variaciones de las fases $\{\theta_n(t)\}$.

En la Ec (4.12), cada señal reflejada tiene un corrimiento de fase y cierta atenuación con respecto a la señal deseada. A su vez, las señales reflejadas se pueden describir en términos de sus componentes ortogonales:

$$\alpha_n(t)e^{-j\theta_n(t)} = x_n(t) + jy_n(t) \tag{4.13}$$

Si el número de componentes reflejadas es grande y ninguna es dominante, las componentes tendrán una distribución Gaussiana. Estas componentes son las que manejan la amplitud del *fading* de pequeña escala $r_0(t)$ de la Ec. (4.2) que, en términos de portadora no modulada, se traduce en la señal $z(t)$ de la Ec. (4.12). Es decir que:

$$r_0(t) = \sqrt{x_r^2(t) + y_r^2(t)} \tag{4.14}$$

Si la señal recibida se compone de múltiples rayos reflejados más una componente importante de señal directa, la amplitud de la envolvente recibida se asocia a una función densidad de probabilidad de Rician, conociéndose como desvanecimiento de Rician:

$$p(r_0) = \frac{r_0}{\sigma^2} e^{\frac{-({r_0}^2 + A^2)}{2\sigma^2}} I_0\left(\frac{r_0 A}{\sigma^2}\right), \text{ para } r_0 \geq 0, A \geq 0 \tag{4.15}$$

Aunque la envolvente varíe dinámicamente en el tiempo debido al propio movimiento, en un tiempo fijo es una variable aleatoria proveniente de un ensamble de números reales positivos. Por este motivo resulta apropiada la descripción mediante la pdf de la Ec. (4.15) que no es función de tiempo. En esta ecuación, A representa el pico de magnitud de la señal no desvanecida e $I_0(*)$ es la función de Bessel modificada de primera clase y orden cero. La distribución de Rician muchas veces se describe en términos de un parámetro K, definido como la relación de potencia entre la componente directa y la potencia de la señal *multipath*, es decir $K = \frac{A}{2\sigma^2}$.

A medida que la componente de línea directa se acerca a cero, la función densidad de probabilidad de Rician se aproxima a una función densidad de probabilidad de Rayleigh:

$$p(r) = \frac{r}{\sigma^2} e^{\frac{-r}{2\sigma^2}}, \text{ para } r \geq 0 \tag{4.16}$$

En la expresión, r es la amplitud de la envolvente de la señal recibida y $2\sigma^2$ es la potencia promedio de pre-detección de la señal *multipath*. La componente de Rayleigh se denomina a veces componente difusa, *scatter* o aleatoria. Es decir que, para un enlace único, representa la pdf asociada con el peor caso de potencia promedio recibida.

En lo que sigue del texto, se supondrá que la pérdida en la relación señal ruido debida al *fading* sigue el modelo de Rayleigh descripto y que la señal propagada, se encuentra en la banda UHF ($300\ MHz$ a $3\ GHz$).

La Fig. 4. 10 ilustra el caso de la respuesta de un canal *multipath* a la transmisión de un pulso muy angosto.

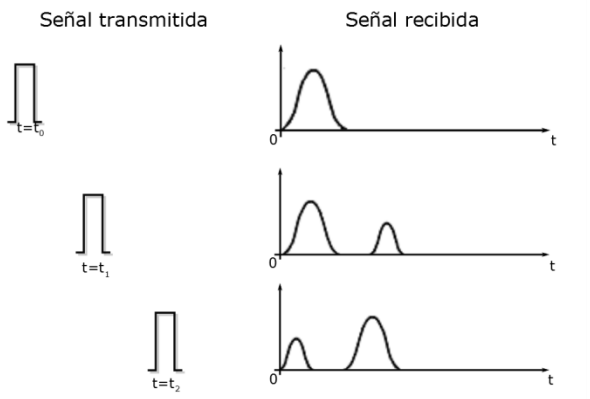


Fig. 4.10 – Respuesta de un Canal Variante en el Tiempo

El tipo de *fading* de pequeña escala descripto produce efectos como los presentados en dicha figura. Es importante destacar que el tiempo del eje de abscisas de cada respuesta es en realidad el retardo, pensado como el efecto de desparramo en el tiempo resultante de la propia respuesta del canal al impulso. Se debe distinguir este tiempo respecto del tiempo de transmisión, relacionado con el movimiento entre dispositivos o el comportamiento del canal variable en el tiempo. Por ejemplo, en la figura se han graficado tres posiciones de la antena de recepción que, suponiendo un movimiento a velocidad constante, equivalen a tres diferentes tiempos de transmisión. Se puede observar la diferencia en los patrones de respuesta, tanto en lo que respecta al retardo como a la magnitud, el número de réplicas y la potencia recibida.

Estos mecanismos de *fading* de pequeña escala producen determinados efectos de degradación que pueden ser estudiados tanto en el dominio del tiempo como en el dominio de la frecuencia.

Así, el desparramo en el tiempo estudiado en el dominio del tiempo, se traduce en desvanecimientos selectivos en frecuencia o planos (no selectivos), según que el retardo sea mayor o menor que el tiempo correspondiente al símbolo transmitido. En estos casos importa el desparramo del retardo (*delay spread*). El mismo efecto estudiado en el dominio de la frecuencia, se presenta también con

naturaleza selectiva o no selectiva en frecuencia según que el ancho de banda de coherencia sea menor o mayor que la velocidad a la que se transmiten los símbolos.

Por su parte, el mecanismo de variación en el tiempo debido al movimiento se podrá caracterizar en el dominio del tiempo con un tiempo de coherencia asociado al canal, provocándose un desvanecimiento rápido o lento, según que este parámetro sea menor o mayor respecto del tiempo de duración de un símbolo. De la misma manera, visto en el dominio de la frecuencia, el desvanecimiento se caracterizará por ser rápido o lento según que la velocidad de desvanecimiento propia del canal sea mayor o menor que la velocidad a la que se transmiten los símbolos.

Estudiaremos con mayor detalle estos efectos, sus parámetros asociados y las maneras existentes para mitigarlos en el capítulo correspondiente la capa física de las redes LAN inalámbricas

Bibliografía

1. Stalling, Williams, "Comunicaciones y Redes de Computadoras". Prentice Hall. Sexta Edición, 2000.
2. Hassall, Fred, "Data Communications, Computer Networks and Open Systems". Addison – Wesley. Fourth Edition, 1995.
3. Haykin, Simon, "Communication Systems". John Wiley & Sons. Fourth Edition, 2001.
4. Proakis, John, Salehi, Masoud, "Communication Systems Engineering". Prentice Hall, 1994.
5. Siemon, Network Cabling Solutions. "Cableado Apantallado y Blindado - Inmunidad al Ruido, Conexión a Tierra y el Mito de la Antena". http://www.siemon.com/la/learning/Screened_and_Shielded_Guide/Screened_and_Shielded_Guide_1_Overview_and_History.asp
6. Fluke, Application Note, "Electrical noise and transients". http://support.fluke.com/find-sales/download/asset/2403183_a_w.pdf
7. Cassiolato, César, "Instalaciones Fieldbus: Acoplamientos Capacitivo e Inductivo". SMAR Equipamentos Industriais Ltda. 2012. http://www.smar.com/newsletter/marketing/index152.html
8. Panduit White Paper "The evolution of Copper Cabling Systems from Cat5 to Cat5e to Cat6", 2004. http://www.gocsc.com/UserFiles/File/Panduit/Panduit098765.pdf
9. Extron Electronic, "UTP Technology" http://www.extron.com/download/files/whitepaper/cat5-white.pdf
10. Perez, Pedro F., "Parámetros de Cableado de Cobre" http://www1.frm.utn.edu.ar/medidase2/varios/parametros_redes1.pdf
11. Sklar, Bernard, "The Characterization of Fading Channels". http://faraday.ee.emu.edu.tr/ee569/art_sklar5_fading.pdf

Problemas

1. Realice una comparación entre las principales características constructivas y de funcionamiento entre los medios de transmisión guiados explicados en este capítulo.

2. Repita el problema anterior para las tres formas de transmisión no guiadas expuestas en este capítulo.

3. ¿Cuáles son las principales fuentes de ruido en el caso de los medios guiados? ¿Qué medidas se pueden tomar para evitarlas?

4. Explique los principales parámetros de comportamiento de un cable, explicitando las unidades de medida en cada caso.

5. Explique conceptualmente los parámetros que caracterizan el efecto de desvanecimiento en el canal inalámbrico.

6. Resalte diferencias conceptuales entre un canal de aire tradicional, contaminado con ruido blanco gaussiano, y un canal inalámbrico con problemas de desvanecimiento.

Parte II

Redes LAN

CAPÍTULO V

Métodos de Acceso al Medio LAN Cableadas

En este capítulo se aborda el estudio de las redes LAN, introduciendo conceptos relacionados con redes de difusión de acceso múltiple. Se trata el control del acceso al medio desde una perspectiva histórica, para poder comprender cómo fue posible arribar a la filosofía de acceso de las redes actuales. En este sentido, se presentan los protocolos originales ALOHA y sus derivados de detección de portadora. Se explorará cómo evolucionó la eficiencia de estos métodos hasta llegar a la mejora de la detección de colisiones con CSMA/CD, base de las redes Ethernet y del estándar IEEE 802.3, al que se dedicará un estudio más profundo. El capítulo finaliza con las especificaciones estandarizadas de cableado estructurado.

5.1 Métodos de Acceso al Medio

En el Modelo OSI, la capa de enlace se considera dividida en dos subcapas: MAC y LLC. La subcapa MAC, como las siglas de su nombre lo indican, provee mecanismos de control de acceso al medio. Es la subcapa que hace posible que varias computadoras se comuniquen dentro de un entorno de red de acceso múltiple, sobre un único medio compartido. Además, provee un esquema de direccionamiento de significado local y es la interfaz entre la subcapa LLC y la capa física.

Los sistemas de acceso múltiple y los canales de difusión se encuentran íntimamente relacionados a la subcapa MAC. Las redes de difusión tienen un solo canal de difusión compartido por todas las máquinas de la red. En estos canales, los mensajes que envía un dispositivo son recibidos por todos las demás. Un campo de dirección dentro del mensaje especifica el dispositivo al que va dirigido. Al recibir un mensaje, cada máquina verifica el campo de dirección y, si es la propia, lo procesa. De otra manera, lo ignora.

En este tipo de redes, el objetivo de los mecanismos asociados al acceso por competencia es el de ofrecer un servicio de comunicaciones equitativo entre los participantes. Con este propósito se imponen reglas que ordenan la comunicación bajo la implementación de un protocolo. A su vez, el protocolo incorpora algoritmos que regulan el flujo de la comunicación.

Los primeros métodos de acceso a un medio compartido surgieron como respuesta a los problemas planteados en las comunicaciones satelitales. Al principio, estos sistemas funcionaron de manera centralizada, con base en una estación en tierra o en el satélite mismo. La debilidad inherente de todo sistema centralizado, abrió el camino para la investigación de sistemas de control distribuido para el acceso. Los primeros esquemas de acceso múltiple se basaron en asignaciones fijas o periódicas, independientemente de las necesidades. Se los conoce como esquemas de reparto estático. En capítulos previos se han presentado ejemplos de este tipo, tales como FDM y TDM. Más tarde comenzaron a aparecer los esquemas de asignación por demanda que resultaron ser más eficientes en el caso de tráfico en ráfagas. Se trata de esquemas de reparto dinámico del canal.

Algunos ejemplos de redes con soporte de medio físico compartido se presentan en las redes de topología *bus*, en las de topología estrella con repetidor central, en las redes inalámbricas o en enlaces punto a punto tipo *half duplex*. En este tipo de redes, el acceso simultáneo de dos máquinas genera una situación de solapamiento de señales en el tiempo, situación conocida como colisión. Las colisiones se traducen en errores que imposibilitan la recepción adecuada. Los protocolos de acceso múltiple pueden detectar o tratar de evitar estas colisiones, mediante diferentes mecanismos adecuados, ya sea de contienda o de reserva del canal.

4.1.1 Aloha Puro

El sistema ALOHA, también conocido como ALOHAnet, fue desarrollado en la Universidad de Hawaii en el año 1971, para ser operado sobre una red de paquetes inalámbrica que debía conectar varios sitios repartidos en diversas islas. En esos años, no existían estándares de aplicación comercial para la asignación de frecuencias o canales para comunicaciones entre computadoras, pero existían redes de tendido sobre cable e inalámbricas.

El objetivo de los investigadores de la Universidad de Hawaii, se centró en utilizar un equipo de radio comercial de bajo costo para conectar usuarios del campus central de la Universidad y otros de otras islas de Hawaii, con una computadora central de tiempo compartido en el campus. La versión original definía dos frecuencias de transmisión y una configuración estrella con *hub* central. Los usuarios del sistema enviaban paquetes hacia el *hub*, sobre el canal que se denominó entrante o *inbound*. El repetidor los re-enviaba a todos los dispositivos, por *broadcast* sobre el canal saliente o *outbound*. Para identificar el destinatario del paquete, el esquema ALOHA original incorporó un sistema de direcciones.

Por otra parte, si los datos se recibían correctamente en el *hub*, el usuario recibía como respuesta una trama de reconocimiento, denominada ACK. Si no se recibía la trama ACK dentro de un tiempo previamente establecido, la máquina transmisora medía un intervalo de tiempo extra, de carácter aleatorio, para luego proceder a una retransmisión. De este modo, la no recepción de una trama de ACK se interpretaba como una situación en que dos máquinas habían tratado de acceder al medio al mismo tiempo, evento que se denominó colisión.

Cabe destacar que el reloj asociado a la recepción de la trama especial de ACK, se debía ajustar a los tiempos de ida y vuelta relacionados con la red particular. Además, como el evento de colisión confirmaba la intención de dos o más dispositivos intentando acceder al mismo tiempo al medio, el agregado de un tiempo aleatorio previo a la retransmisión, disminuía la probabilidad de una nueva colisión. Esta forma de retransmisión se encontrará en muchos protocolos, asociándosela a técnicas de retroceso o *back off* que influyen bastante en la eficiencia de funcionamiento de los mismos.

La necesidad de un mecanismo de control se debía a que todos los usuarios se comunicaban con el *hub* en la misma frecuencia. Básicamente, ALOHA estableció que cada estación o nodo de la red transmitiera toda vez que se tuvieran datos, manejando los eventos de colisión con las tramas de ACK y las posteriores retransmisiones. Este mecanismo tan sencillo, se conoció como ALOHA puro y sentó las bases para el control de acceso al medio en redes tipo *Ethernet* y WiFi.

En ALOHA, una máquina podía encontrarse en uno de los modos siguientes, como se observa en la Fig. 5.1:

- **Modo Transmisión**: los usuarios transmiten cuando disponen de datos.

- **Modo Escucha**: luego de transmitir, se escucha el medio, habilitando el sistema de recepción, en espera de una trama de ACK. Puede haber colisiones.

- **Modo Retransmisión**: cuando se recibe una trama de indicación de error, denominada NACK, o no se recibe la trama de ACK, se asume que hubo colisión. Si hay colisión, se retransmite luego de un tiempo aleatorio. Como se mencionó antes, el tiempo es aleatorio porque se supone que, si hubo una colisión, otro usuario está tratando de acceder al canal y también detectará la situación. Si ambos retransmitieran de inmediato, la probabilidad de colisión sería mayor.

- **Modo Timeout**: si no se recibe ACK (reconocido) ni NACK (no reconocido) dentro de cierto tiempo límite, se pasa al modo retransmisión.

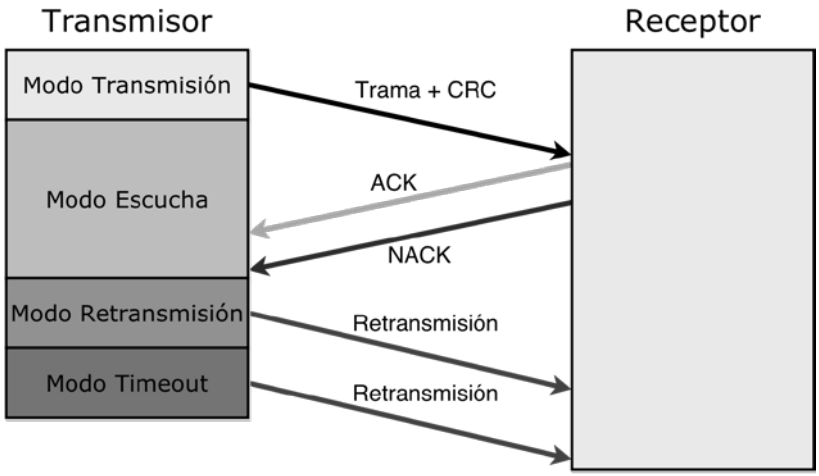


Figura 5.1 - ALOHA

En la Fig. 5.1 se presentan los cuatro modos mencionados. El modo de retransmisión se puede disparar por la recepción de una trama NACK (recepción incorrecta) o luego de haberse agotado el tiempo límite en espera de una trama ACK. Se destaca en la figura que la trama de mensaje se acompaña de algún esquema de detección de errores, por ejemplo un código de redundancia cíclica CRC, cuya verificación genera tramas ACK o NACK.

A fines de comparación con otros métodos de acceso, interesa conocer la eficiencia de este protocolo pero, para simplificar las relaciones, se deben realizar ciertas suposiciones:

- Todas las tramas tienen la misma longitud $L\ bits/trama$.
- Las máquinas no pueden generar tramas mientras están transmitiendo o retransmitiendo.
- El grupo de máquinas transmite o retransmite siguiendo una Distribución de Poisson. Se trata de una distribución de probabilidad discreta que expresa, a partir de una frecuencia de ocurrencia media, la probabilidad de que ocurra un determinado número de eventos durante cierto período de tiempo.

Se define T como el tiempo, medido en segundos, necesario para transmitir una trama. Resulta $T = L/r_b$, siendo r_b la velocidad sobre el canal medida en bps.

Se define λ_{ok} como la velocidad de llegada de tramas aceptadas sin errores, medida en $tramas/seg$. Si λ_r es velocidad de llegada de tramas rechazadas, entonces la velocidad de tramas total es $\lambda = \lambda_{ok} + \lambda_r$.

De este modo, la probabilidad de éxito de una transmisión, que se simboliza como P_s, se puede medir como la relación entre λ_{ok} y λ

$$P_s = \frac{\lambda_{ok}}{\lambda} \tag{5.1}$$

El producto $\rho' = \lambda_{ok} L$, representa la cantidad de bps recibidos sin errores durante el tiempo de transmisión de una trama. Si se divide este producto por r_b, se obtiene una relación normalizada, que mide la eficiencia del sistema:

$$\rho = \frac{\lambda_{ok} L}{r_b} = \lambda_{ok} T \tag{5.2}$$

Del mismo modo, la relación G representa la cantidad de bits de tráfico total normalizado recibido durante el tiempo de transmisión de una trama.

$$G = \frac{\lambda L}{r_b} = \lambda T \tag{5.3}$$

Bajo estas suposiciones, para que una estación pueda realizar una transmisión exitosa, necesita disponer del medio durante $T\ seg$, pero si otra estación transmite dentro de los $T\ seg$ previos o los $T\ seg$ posteriores, se producirá una colisión. Teniendo en cuenta esta premisa, se puede hablar de la existencia de una ventana de colisión, también conocida como ventana de contienda, de duración $2T$, tiempo durante el cual la transmisión de un mensaje es vulnerable a la interferencia por el acceso de otros dispositivos. Es decir que, una vez que un dispositivo accede al medio, para poder realizar una transmisión exitosa precisaría que el resto de las máquinas no transmitiera durante ese tiempo, tal como se presenta en la Fig. 5.2.

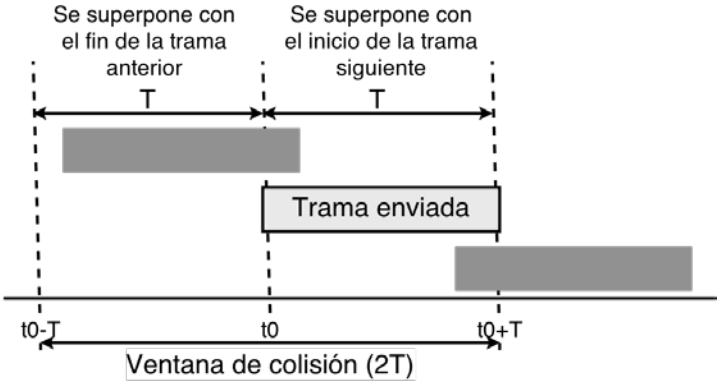


Figura 5.2 - Vulnerabilidad en ALOHA puro.

Bajo el supuesto de que la estadística de llegada de los mensajes, en el caso de usuarios no relacionados, se puede modelar como un proceso de Poisson,

la probabilidad de que lleguen k mensajes durante un intervalo de $T\ seg$ y a una velocidad promedio λ será:

$$P(k) = \frac{(\lambda T)^k}{k!} e^{-\lambda T} \tag{5.4}$$

Aplicando esta distribución al sistema de acceso desarrollado en ALOHA, teniendo en cuenta que el período de vulnerabilidad es $2T$, se podría medir la probabilidad de éxito como aquella en la que ningún otro usuario transmita ($k = 0$) durante el intervalo vulnerable ($2T$):

$$P_s = \frac{\lambda_{ok}}{\lambda} = P(0) = e^{-2\lambda T} \tag{5.5}$$

Se podría re-escribir esta expresión como:

$$\lambda_{ok} = \lambda\, e^{-2\lambda T} \tag{5.6}$$

Si se multiplica ambos miembros de la Ec. (5.6) por T, resulta:

$$\lambda_{ok} T = \lambda\, T e^{-2\lambda T} \tag{5.7}$$

A la izquierda de la Ec. (5.7), aparece la expresión de eficiencia normalizada y, la derecha, la de tráfico total normalizado. Es decir que la expresión anterior es equivalente a:

$$\rho = G e^{-2G} \tag{5.8}$$

De estas relaciones, comparando la Ec (5.5) con la Ec. (5.8), también surge que la probabilidad de éxito se puede expresar de manera equivalente como:

$$P_s = \frac{\lambda_{ok}}{\lambda} = e^{-2\lambda T} = \frac{\rho}{G} \tag{5.9}$$

A partir de la expresión de la probabilidad de éxito, se puede derivar la expresión de probabilidad de colisión para el esquema ALOHA:

$$P_{coll} = 1 - P_s = 1 - \frac{\rho}{G} \tag{5.10}$$

En la Fig. 5.3 se grafica la expresión de la eficiencia del sistema presentada en la Ec. (5.8). Se puede observar que el máximo valor de eficiencia se encuentra derivando la expresión e igualando el resultado a cero:

$$\frac{\partial \rho}{\partial G} = e^{-2G} + G(-2)e^{-2G} = 0 \tag{5.11}$$

Es decir que en el máximo, para el cual se cumple

$$e^{-2G}(1 - 2G) = 0 \tag{5.12}$$

resulta presentarse en $G = 1/2$, que se corresponde con un valor de eficiencia: $\rho = 0.5/e$.

Entonces, el valor máximo de eficiencia es $\rho = 0.184$ tramas por tiempo de trama, o sea que en *ALOHA* puro sólo durante 18.4% del tiempo, como máximo, se pueden realizar transmisiones exitosas. Evidentemente, la eficiencia de este esquema dejaba mucho que desear.

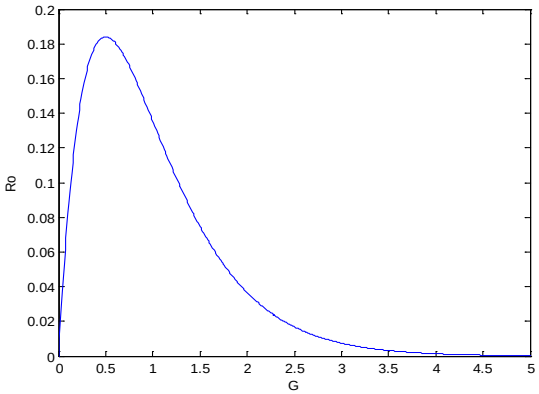


Figura 5.3 - Eficiencia ALOHA puro.

5.1.2 Aloha Con Ranuras

Debido a la baja eficiencia del esquema ALOHA puro, se empezaron a estudiar alternativas para mejorarlo. Entre ellas, surgió el esquema ALOHA con ranuras, variante que logró duplicar la capacidad de ALOHA original. En ALOHA con ranuras, el tiempo se considera dividido en ranuras de duración T . Se trata de un esquema sincronizado, de modo tal que la transmisión sólo se puede realizar al comienzo de una ranura. Un reloj centralizado envía la señal de reloj a las estaciones que están intentando acceder al medio. Estas sólo tienen permitido enviar sus tramas inmediatamente después de recibir la señal de reloj. Si hay una

sola estación con intención de emitir un mensaje, se garantiza que nunca habrá una colisión para ese paquete. De este modo, al imponer tiempos de inicio de la transmisión, el esquema ALOHA puro se convierte en discreto, lográndose reducir a la mitad el tiempo de vulnerabilidad T, ya que sólo colisionarán las tramas que se transmitan en la misma ranura.

En ALOHA con ranuras, la probabilidad de éxito se reduce a:

$$P_s = \frac{\lambda_{ok}}{\lambda} = P(0) = e^{-\lambda T} \tag{5.13}$$

Trabajando la expresión como en el caso anterior, se obtiene la relación de eficiencia:

$$\rho = Ge^{-G} \tag{5.14}$$

En la Fig. 5.4 se grafica la expresión de la eficiencia del sistema. Se puede observar que el máximo valor de eficiencia se corresponde con:

$$\frac{\partial \rho}{\partial G} = e^{-G} + G(-1)e^{-G} = 0 \tag{5.15}$$

Es decir que resulta

$$e^{-G}(1 - G) = 0 \tag{5.16}$$

y el punto máximo se alcanza con $G = 1$, siendo $\rho = 1/e$.

Entonces, el valor máximo de eficiencia, es $\rho = 0.368$ tramas por tiempo de trama. En ALOHA con ranuras, como máximo un 36.8% del tiempo se puede utilizar para transmisiones exitosas, el doble que en el caso de ALOHA puro.

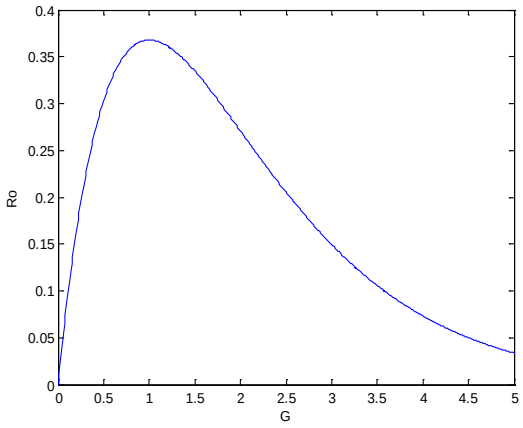


Figura 5.4 - Eficiencia ALOHA con ranuras.

Interesa calcular el número promedio de intentos que debe realizar una máquina para poder ocupar el medio y su relación con la carga sobre el sistema. La probabilidad de tomar el medio P_x luego de x intentos se puede calcular como la probabilidad de no haber podido hacerlo durante $(x-1)$ intentos previos y lograrlo en el siguiente. Según la Distribución de Poisson:

$$P_x = P(k \neq 0)^{x-1} P(k = 0) = (1 - e^{-G})^{x-1} e^{-G} \tag{5.17}$$

El número promedio de intentos necesarios para poder tomar el medio luego de x intentos es:

$$E_x = \sum_{x=1} x P_x = \sum_{x=1} x(1 - e^{-G})^{x-1} e^{-G} = e^{-G} \sum_{x=1} x(1 - e^{-G})^{x-1} \tag{5.18}$$

Denominando $y = (1 - e^{-G})$, resulta:

$$E_x = e^{-G}(1\, y^0 + 2y^1 + 3y^2 + \cdots . = e^{-G}(1 - 1 - e^{-G})^{-2} = e^{-G} e^{2G} = e^{G} \tag{5.19}$$

La expresión anterior denota una dependencia exponencial entre la cantidad promedio de intentos y la carga del sistema. Esta dependencia no es una propiedad deseable en ningún método de acceso.

5.1.3 CSMA – Acceso Múltiple por Detección de Portadora.

El método de Acceso Múltiple por Detección de Portadora (CSMA, Carrier Sense Multiple Access) es un método de acceso al medio de naturaleza probabilística. En CSMA, cuando un dispositivo tiene datos para enviar, a diferencia de lo que ocurría en ALOHA puro, antes de transmitir verifica que ninguna otra máquina se encuentre ocupando el medio compartido. Para ello, la etapa receptora trata de detectar la presencia de señal portadora proveniente de otra máquina que estuviera ocupando el medio. De encontrarlo ocupado, se abstiene de transmitir hasta que éste se desocupe.

En estos esquemas, el acceso múltiple se relaciona con el medio compartido, o sea con la posibilidad de que cuando una de las máquinas transmite, todas pueden recibir esa transmisión. Existen algunas variantes del modo de acceso CSMA que se presentan a continuación.

CSMA persistente-1

En este método, cuando la estación está lista para transmitir, primero chequea el medio. Si lo encuentra ocupado, lo continúa escuchando de manera

continua hasta que el mismo se desocupa. Cuando esto sucede, transmite con probabilidad 1.

Luego de una transmisión, en el caso que haya colisión, se espera un tiempo aleatorio antes de volver a intentarlo. La colisión se detecta por ausencia de la recepción de una trama de ACK.

El éxito en poder detectar lo que sucede en el medio, sin confundir una situación de medio libre cuando en realidad está ocupado, tiene que ver con el tiempo de propagación de la señal en la red. Este tiempo, a su vez, se relaciona con la longitud de la red. El caso más relevante es el de separación entre extremos más alejados, que sería el correspondiente al mayor tiempo de propagación. En este sentido, surge uno de las primeras decisiones de diseño, ya que la longitud de la red debe ser de tamaño apropiado para evitar detectar falsos libres y generar así nuevas colisiones.

Se presenta un diagrama de flujo representativo del funcionamiento de este método en la Fig. 5.5.

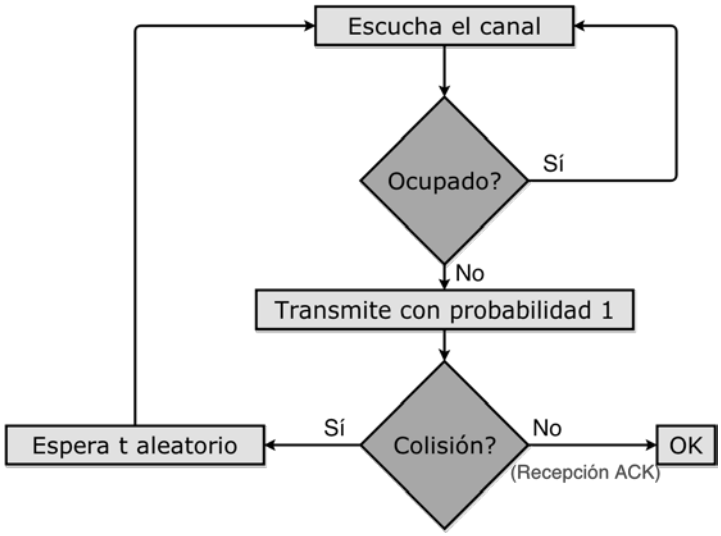


Figura 5. 5 - CSMA persistente-1.

CSMA No persistente

La Fig. 5.6 ofrece un diagrama de flujo del método no persistente. La diferencia con el anterior es que, si el canal está ocupado, espera un tiempo aleatorio antes de volver a detectar. Este detalle mejora la eficiencia, aunque aumenta el retardo promedio de acceso, en comparación con CSMA persistente-1.

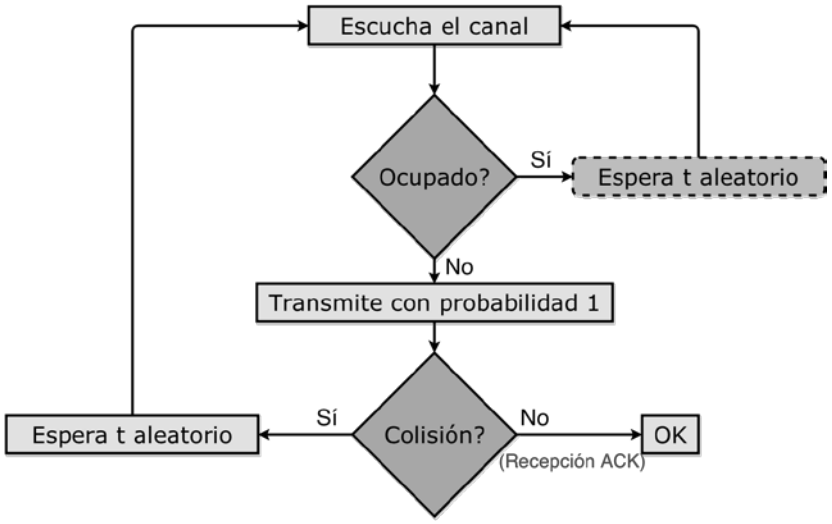


Figura 5.6 - CSMA No persistente.

CSMA persistente-p

En este caso, el transmisor también detecta el medio para ver si está ocupado. La diferencia es que, cuando el medio se encuentra libre, el dispositivo transmite con probabilidad p, como se observa en la Fig. 5.7. Se trata de un sistema de uso común en esquemas de acceso con ranuras en el tiempo.

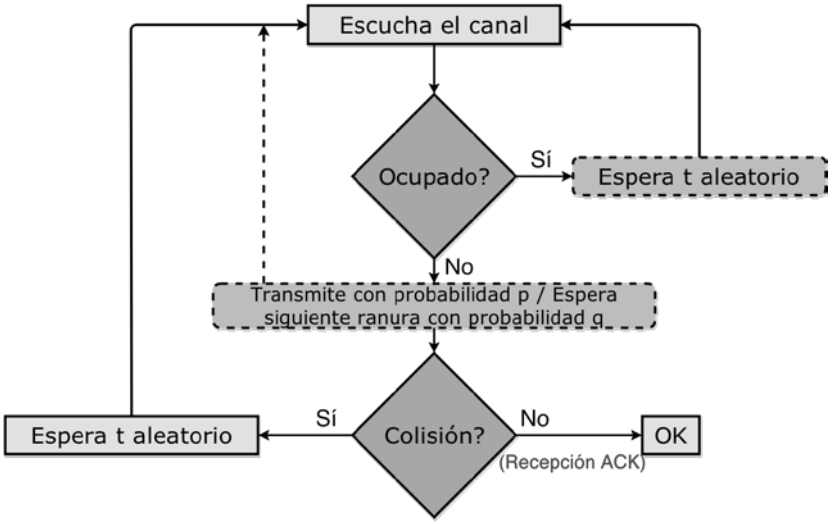


Figura 5.7 - CSMA persistente-p

Si se decide no transmitir, con probabilidad $(1 - p)$, el sistema espera la siguiente ranura y transmite con probabilidad p si el medio está libre. El proceso se repite hasta que se logra transmitir el mensaje.

Con este método se pretende minimizar las colisiones y el tiempo en el que el canal está desocupado. Se comporta peor que CSMA persistente en el caso de tráfico moderado, pero en situaciones de congestión evita el efecto de aumento promedio en cola de espera. Este tipo de esquemas es de aplicación en sistemas WiFi.

CSMA/CD – Acceso Múltiple por Detección de Portadora y Detección de Colisiones

Las mejoras introducidas por CSMA, respecto de los sistemas ALOHA, se deben fundamentalmente a la detección de medio ocupado. Una mejora posible al método CSMA consiste en detectar las colisiones y, de producirse, abortar la transmisión. Este nuevo método se conoce como Método de Acceso Múltiple por Detección de Portadora/ Detección de Colisiones (CSMA/CD, Carrier Sense Multiple Access/Collision Detection).

Como se indica en la Fig. 5.8, el estudio del comportamiento de las redes CSMA demostró que la actividad en estas redes presenta períodos alternativos de transmisión, contienda e inactividad. Se produce colisión cuando dos o más estaciones transmiten a la vez. En todos los esquemas explicados previamente, la detección de colisión dependía del arribo de una trama de ACK. En el esquema CSMA/CD no se implementa de esa manera, no existen tramas de ACK cuyo objetivo sea la detección de colisiones.

Sin la existencia de tramas de ACK, igualmente se puede detectar colisión, por comparación entre la señal recibida y la señal transmitida. Esto significa que, a la vez que transmite, la estación debe escuchar el medio, habilitando al mismo tiempo su sistema de recepción. Si una estación detecta colisión, aborta la transmisión y emite una señal corta de interferencia para que todas las demás se abstengan de transmitir. Luego espera un tiempo aleatorio, denominado de *back off* o de retroceso, y vuelve a escuchar el medio. Se trata del método de acceso utilizado por las redes cableadas tipo *Ethernet*.

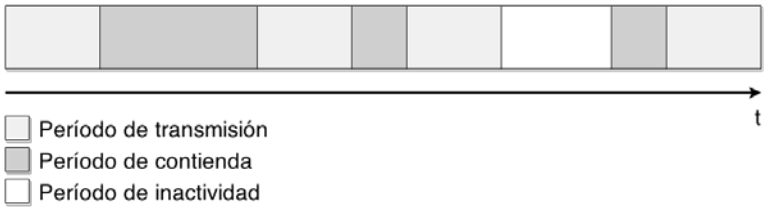


Figura 5.8 - Situación en el tiempo de una red de Acceso Múltiple.

Del análisis del método expuesto, surgen una serie de interrogantes. Por ejemplo, ¿En cuánto tiempo se darán cuenta las estaciones transmisoras que ocurrió una colisión? Se debe tener presente que este tiempo se relaciona con lo que tarda la señal en propagarse de una estación a otra. También: ¿Importa sólo el tiempo de ida? ¿Importa el de ida y vuelta? ¿Por cuánto tiempo debe quedarse

escuchando una estación transmisora para estar segura de que no hubo colisión? Se intentará dar respuestas a estas preguntas a través de un ejemplo, tal como se presenta en la Fig. 5.9.

Supongamos una red como la de la Fig. 5.9, con topología *bus*, donde intentan comunicarse cuatro máquinas. La topología *bus* implica que todas las máquinas se conectan al mismo cable, todas pueden ver lo que el resto transmite, conformando una red de difusión. En el ejemplo, A y D son las máquinas que más alejadas se encuentran entre sí. Si se supone que A desea transmitir una trama, lo que debe hacer primero es recibir señal proveniente del medio, para verificar si el mismo se encuentra ocupado o no. Si en un principio, A encuentra el medio libre, tal como se presenta en la figura, empieza la transmisión.

Si las máquinas B y C también desean transmitir casi al mismo tiempo que A, surge que, al haber tomado el medio la máquina A, cuando la máquina B habilite su circuito de recepción encontrará el medio ocupado con la señal proveniente de A. La máquina B es la primera en entender que el medio no está libre, debido a que se encuentra más cercana físicamente al dispositivo A. Por su parte la máquina C, que se encuentra un poco más alejada, no puede darse cuenta que el medio está ocupado hasta que la señal transmitida por A no haya recorrido el trayecto entre A y C. Si durante este tiempo, la máquina C detectara el medio, lo encontraría libre y empezaría a transmitir, tal como se indica en la Fig. 5.9.

Cuando las tramas transmitidas por A y C se encuentren en el *bus*, se producirá una colisión, una superposición de ambas señales en el cable que se interpretará como ruido. Los dispositivos A y C se darán cuenta de la colisión cuando puedan detectar por sus respectivos circuitos receptores que la señal que están transmitiendo no coincide con la recibida. Esto conlleva un tiempo relacionado con el tiempo de propagación de la señal en el cable.

Por este motivo, en este ejemplo, el primero en enterarse de la colisión es el dispositivo C, por encontrarse más cerca físicamente del lugar donde la misma se produce. La reacción de C consistirá en abortar inmediatamente la transmisión, aunque también dispondrá el envío de una trama especial, más corta que cualquiera de las permitidas, para avisar al resto de esta situación de colisión. Se conoce a esta trama como trama de interferencia o *jamming*.

En el peor caso, el tiempo que se demora en comprender que hubo una colisión depende del retardo de propagación extremo a extremo de la red, pues sólo es función de cuán alejados se encuentran entre sí los dispositivos de los extremos del *bus*. Por este motivo, este tiempo no puede ser mayor que dos veces el retardo de propagación extremo a extremo. Es decir que, la detección de colisión depende del llamado tiempo de ida y vuelta asociado a la red.

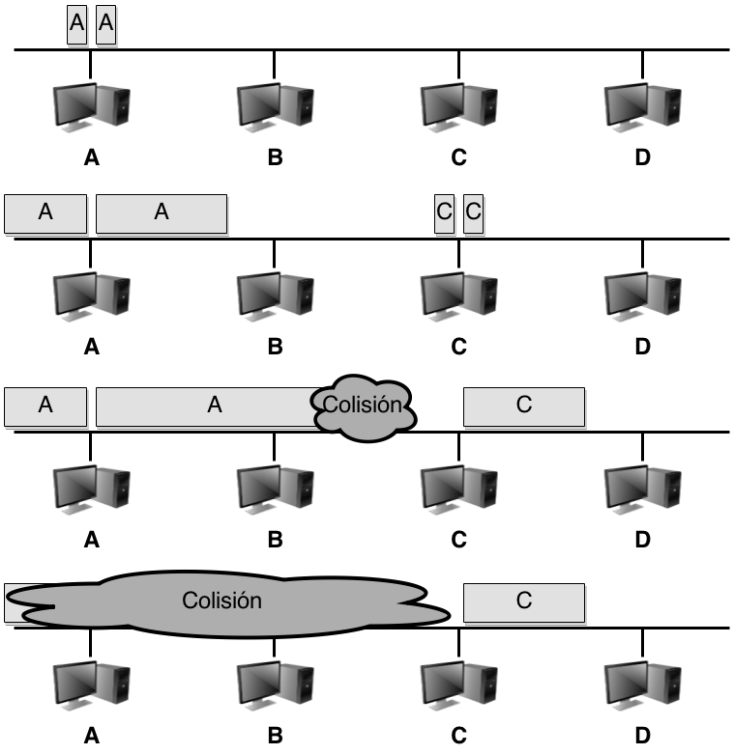


Figura 5.9 - Detección de Colisión.

Por ejemplo, si las máquinas más alejadas entre sí, A y D, estuviesen separadas por una distancia $d = 200m$, suponiendo la velocidad en el cable $v = 2x10^8 m/seg$, llamando retardo de propagación a la relación $\tau_d = \frac{d}{v}$, el tiempo de ida y vuelta resultaría:

$$2\tau_d = \frac{2x200}{2x10^8} = 2\ \mu seg \tag{5.20}$$

Cualquier estación con intenciones de transmitir estará obligada a escuchar el medio al menos durante ese tiempo para poder enterarse si hubo colisión o no. Esta restricción pone en juego otro parámetro: la longitud de las tramas. En este sentido, las tramas deben ser lo suficientemente largas como para permitir la detección de la colisión antes de que finalice su propia transmisión. Si se denomina L a la longitud de las tramas en bits y r_b a la velocidad de transmisión, la relación $T_{Tx} = L/r_b$ se conoce como tiempo de transmisión de las tramas. Para que el esquema funcione, debería verificarse que:

$$T_{Tx} > 2\tau_d \tag{5.21}$$

Un problema adicional en este método de detección de colisiones es la propia atenuación del cable, de tal manera que la suma de las señales de dos estaciones muy alejadas supere cierto umbral pre-establecido, justamente para que la detección sea posible. La solución a este problema podría encontrarse en la limitación del diámetro máximo de la red, o sea de la distancia máxima entre las máquinas más alejadas.

5.2 IEEE 802.3

Se trata de un estándar para una LAN cableada que utiliza CSMA/CD como método de acceso al medio. En dicho estándar se define no sólo el método MAC sino también se describe la capa física recomendada.

En los años 70, la aparición de ALOHAnet inspiró a desarrolladores de Xerox PARC, quienes patentaron en 1975 una tecnología para redes de computadoras tipo LAN que denominaron *Ethernet*. En los 80, el IEEE comenzó el proyecto 802 para estandarizar redes LAN. Un grupo de investigadores creadores de *Ethernet*, denominado DIX-group (DEC, Intel, Xerox), remitieron al Instituto un libro "Blue Book" con la especificación CSMA/CD para que fuera tenida en cuenta como posible candidata. Competía con las especificaciones Token Ring de IBM y Token Bus de General Motors. Finalmente se procedió a la estandarización de los tres tipos de LAN, aunque con el tiempo fue el estándar de *Ethernet* el que se impusiera en el mercado. El estándar IEEE 802.3 CSMA/CD se aprobó a fines de 1982, aunque el IEEE lo publicó como tal en 1985. Es decir que la historia del estándar se liga absolutamente con la aparición de *Ethernet*. Es por eso que se suelen llamar *Ethernet* a las redes cableadas que derivan de este estándar.

Ethernet se adaptó muy bien y rápido a las necesidades de los mercados, sobre todo cuando se incorporó el par trenzado como medio físico de transmisión. Por tanto, para fines de los 80, era la tecnología que dominaba comercialmente. Además, se trata de un estándar muy flexible, sencillo y fácil de entender.

A nivel físico se han considerado varias implementaciones. Todas ellas se estandarizaron para funcionar a $10\ Mbps$. A nivel MAC, el estándar define una trama, especificando los campos de la misma, y un método de acceso. *Ethernet* y el estándar IEEE 802.3 difieren únicamente en la definición de un campo en la cabecera de la trama. A pesar de ello, son compatibles, pudiendo coexistir en una misma red física dispositivos con placas de red que responden a ambas implementaciones.

5.2.1 Capa Física IEEE 802.3

El comité IEEE 802.3 definió cuatro configuraciones físicas alternativas. Para distinguir las posibles implementaciones desarrolló una nueva notación que se representa en varios campos: <velocidad de transmisión en Mbps><método de señalización><longitud máxima del segmento en m>. Las alternativas más

importantes resultaron ser 10 BASE 5, 10 BASE 2 y, posteriormente, 10 BASE T. Todas ellas funcionan a $10\,Mbps$, sobre segmentos de 500 y 200 m en el caso de las dos primeras. 10 BASE T no siguió la notación original, refiriéndose la letra final al medio de transmisión: T es por par trenzado o *twisted pair*.

El estándar IEEE 10 BASE 5, también conocido como *Ethernet* grueso o *thick Ethernet*, es la *Ethernet* original. Se trata de una topología tipo *bus*, con un cable coaxial que conecta todas las computadoras entre sí. En cada extremo final del cable debe existir un terminador adaptador para evitar reflexiones. En 10 BASE 5, cada computadora se conecta al cable con un dispositivo llamado transceptor. El cable coaxial usado es relativamente grueso y rígido, lo que dificulta el tendido de la red. Sin embargo, se trata de un medio muy resistente a interferencias externas y con pocas pérdidas. El cable se reconoce por su denominación RG8 / RG11 y tiene una impedancia característica de $50\,ohms$.

Tal como se presenta en la Fig. 5.10, en la vieja red 10 BASE 5, la señal era tomada desde el *bus* mediante conectores tipo vampiro, entre los cuales debía respetarse una distancia mínima de $2.5\,m$, con marcas en el cable para las derivaciones. El conector atravesaba el *bus* con una especie de clavija para hacer contacto con el núcleo del cable coaxial. La lógica de la detección de portadora y detección de colisiones, se encontraba en la propia derivación vampiro. El estándar denominaba Unidad de Acoplamiento al Medio (MAU, Media Attachment Unit) a la unidad con el conector al coaxial, e Interfaz de Unidad de Acoplamiento (AUI, Attachment Unit Interface) al conector con la placa de red.

En 10 BASE 5 se estableció la máxima longitud de cada segmento en $500\,m$, pero la red se podía extender con 4 repetidores hasta una longitud máxima de $2500\,m$. También, por eficiencia de funcionamiento, el número máximo de dispositivos conectados por segmento no podía superar las 100 unidades. En parte, de estas disposiciones deviene la regla $5 - 4 - 3$, norma que limita el tamaño de estas redes. La misma establece que puede haber hasta 5 segmentos conectados en serie en una red, en la que se pueden usar hasta 4 repetidores y no más de 3 segmentos pueden tener estaciones intentando comunicarse.

Los inconvenientes que presentaba el tendido 10 BASE 5 se relacionaban con la rigidez del cable y con la inflexibilidad de la propia red, en el sentido de que es difícil realizar cambios en la instalación una vez montada. También existían dificultades ante la rotura del cable, pues la red quedaba inoperativa y resultaba complejo encontrar la falla. La Fig. 5.11 presenta algunos componentes necesarios para el tendido de redes 10 BASE 5 y 10 BASE 2.

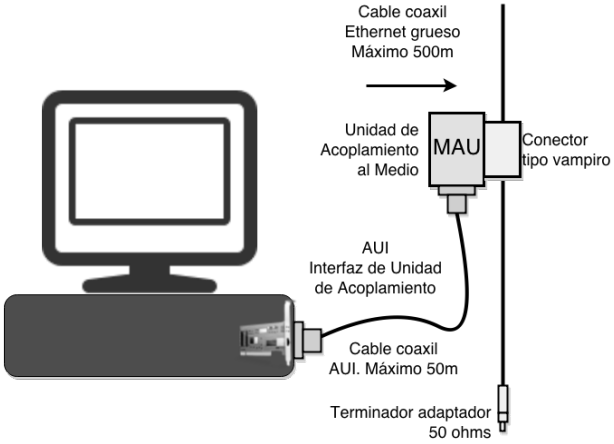


Figura 5.10 - IEEE 802.3. 10 BASE 5.

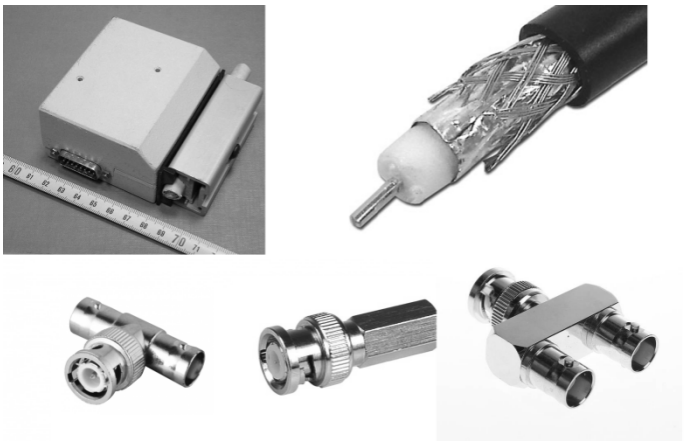


Figura 5.11. Transceptor 10 BASE 5, conectores y cable coaxial de 50Ω.

El estándar con capa física IEEE 10 BASE 2, también conocido como *Ethernet* fino o *thin Ethernet*, era una variante de 10 BASE 5 que usaba cable coaxial fino RG-58A/U o similar, terminado con un conector BNC en cada extremo, y con cada dispositivo agregado a la red con conectores T, tal como se muestra en la Fig. 5.12. Durante muchos años fue el de mayor despliegue en este tipo de redes de 10 $Mbps$, pero fue desplazado con la aparición del par trenzado.

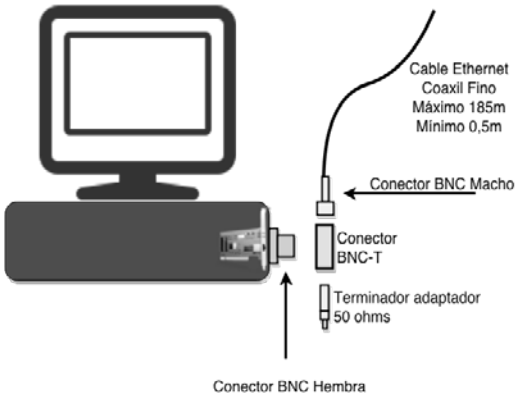


Figura 5.12 - Conectores 10 BASE 2.

10 BASE 2, al igual que 10 BASE 5 utiliza codificación para la transmisión de señales en banda base. Es transmisión en modo *half duplex*, como 10 BASE 5, es decir que sólo una estación puede transmitir a la vez. De lo contrario, se produce una colisión.

En 10 BASE 2 se permite hasta cinco segmentos de $185\ m$ de longitud cada uno, pudiéndose instalar hasta 30 estaciones en cada segmento individual en sólo tres de estos segmentos. Como en 10 BASE 5, las redes 10 BASE 2 no pueden ampliarse o modificarse sin interrumpir temporalmente el servicio al resto de usuario. También son redes vulnerables a la interrupción por fallas en el cable.

La formalización de IEEE 10 BASE T, en los años 90, significó una revolución en el tendido de redes LAN cableadas. *Ethernet* 10 BASE T fue el primer estándar para LAN que consideró las recomendaciones realizadas para un sistema de cableado estándar o cableado estructurado: ANSI/TIA/EIA-568-A. La especificación 10 BASE T es compatible con las versiones del estándar IEEE 802.3 para cable coaxial anteriormente mencionadas, con el propósito permitir una migración suave de la tecnología. Aparte del tipo de cable utilizado, se introdujo una innovación en cuanto a la topología de la red. La topología de 10 BASE T es del tipo estrella, tendida con cable del tipo Par Trenzado no Apantallado (UTP, Unshielded Twisted Pair) de categoría 3 o superior, que se usa para conectar las estaciones a un *hub* 10 BASE T. Los *hubs* son repetidores multi-puerto, tal como se aprecia en la Fig. 5.13. Estos dispositivos toman las señales provenientes de un cable de entrada y las repiten en todas las bocas de salida. Cada puerto en el *hub* provee un punto de conexión por UTP a un dispositivo de la LAN.

Lo más significativo de esta LAN 10 BASE T es que la apariencia física de la red es la de una estrella, pero sigue operando como un *bus*. Cada máquina se conecta al *hub* mediante un par de cables, uno para transmisión y otro para recepción. Cuando el dispositivo envía información, lo hace por el cable transmisor, mientras escucha por el receptor con la finalidad de detectar colisiones. La restricción más significativa, respecto de las redes de cable coaxial,

es que la longitud total del cable, desde el *hub* hasta cualquier dispositivo, no debe exceder los 100 *m*.

En referencia a la Fig. 5.13, una computadora se comunica con el *hub*, transmitiendo por los pines 1 y 2 de un conector RJ 45. La recepción se realiza sobre los pines 3 y 6 del mismo conector. Para que todo funcione correctamente, los pines de transmisión del conector del lado de la máquina, deben estar conectados a los pines de recepción del otro conector y viceversa. De este modo, la conexión exige de un cruce interno en el cable, que se conoce como cable cruzado o *crossover*, aunque el estándar recomienda que los cruces de señales se realicen internamente dentro del puerto del *hub*, no en el cable. Así se simplifica la tarea del tendido de la red, permitiendo utilizar cables rectos, sin cruce, para conectar los dispositivos, no siendo necesario asegurarse que los alambres en los cables estén correctamente cruzados. Cuando el cruce se realiza dentro del puerto del *hub*, el estándar requiere que el puerto sea marcado con una **X**.

Actualmente, las placas de red que funcionan bajo estándares más modernos, a mayor velocidad y en entornos conmutados, cuentan con un mecanismo denominado de auto-negociación para poder salvar este tipo de detalles de conexión.

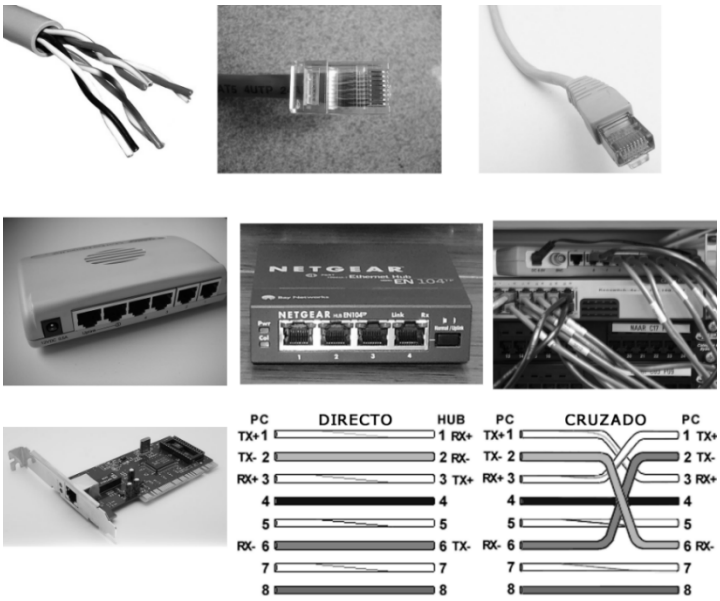


Figura 5.13 - Cables, conectores, hub, placa de red y cable cruzado 10 Base T.

5.2.2 Subcapa MAC IEEE 802.3 - Tramas

Habiendo mencionado las características de la capa física y del método de acceso al medio del estándar IEEE 802.3, resta entrar en algunos detalles de la

subcapa MAC. En este sentido, el estándar define el formato de una trama cuyo encabezado se describirá en este apartado.

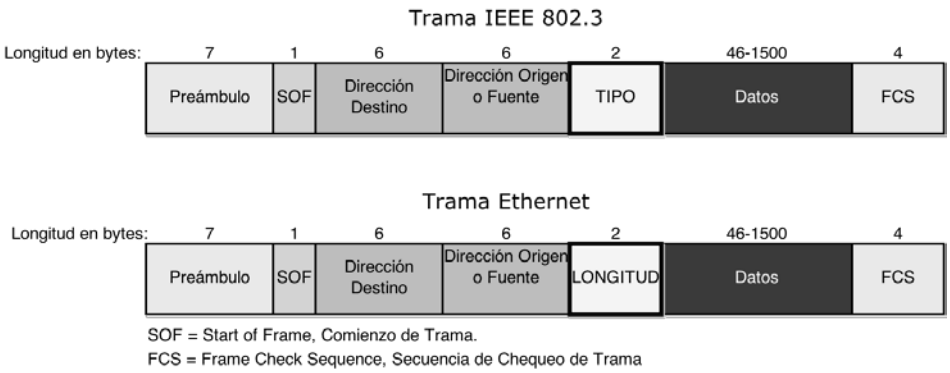


Figura 5.14 - Trama IEEE 802.3 vs Trama *Ethernet*.

La Figura 5.14 presenta los formatos de las tramas MAC IEEE 802.3 y *Ethernet*, con sus cabeceras, sus tamaños máximo y mínimo de carga de datos e información adicional al final. Se observa que ambos formatos difieren en la definición de un único campo. Sin embargo, ambos tipos de tramas y las placas que las generan, pueden convivir en una misma red sin confusiones, según se explicará a continuación.

Las tramas se transmiten en banda base, por medio de una codificación Manchester, que asegura una transición por cada bit, ya sea éste un "0" o un "1", reflejándose en un espectro con contenido de continua nulo, aunque el ancho de banda del mismo es mayor que la velocidad de señalización, apareciendo su primer nulo en $2r_b$, es decir $20\ Mhz$.

El formato de la trama incluye los siguientes campos:

- **Preámbulo** ($7\ bytes$): se podría decir que el preámbulo pertenece en realidad a la capa física, pues su funcionalidad es la de sincronismo en la recepción. Se trata de una serie de $7\ bytes$ de bits alternantes en formato Manchester. Cada uno tiene una duración de $0.1\ \mu seg$, dado que la velocidad es de $10\ Mbps$. De este modo, se origina una onda cuadrada de período $0.2\ \mu seg$, presente durante $5.6\ \mu seg$ que permite a las unidades MAC receptoras detectar fácilmente una nueva trama y sincronizarse al recibir esta señal.

- **Comienzo de Trama** (SOF, Start of Frame)($1\ byte$): se trata de una palabra de $8\ bits$, cuyos 7 primeros son alternantes codificados en Manchester, mientras que el último bit es igual al anterior: (10101011). De este modo, se presenta una violación de código que sirve para indicar el comienzo de la trama.

- **Dirección Destino** (6 $bytes$): 48 $bits$ que indican la dirección MAC de la estación a la cual va dirigida la trama. El dispositivo receptor compara esta dirección con la propia para poder decidir si una trama en particular va dirigida para sí mismo. Todos los dispositivos con este tipo de placa de red tienen cargada una dirección, también de 48 $bits$, grabada desde su fabricación en la memoria ROM de la placa de red.
  En términos generales, en el campo de dirección destino se pueden colocar tres tipos de destinos, según el caso: *unicast*, *multicast* o *broadcast*. En el caso de dirección destino *unicast*, el destino es único. En el caso *multicast*, el destino es un conjunto de estaciones de la LAN. Se trata de grupos, conformados especialmente, para algún propósito particular. En cambio, cuando la transmisión es por *broadcast*, el destino es todo el conjunto de estaciones de esa LAN. La dirección de *broadcast* es una dirección muy especial, ya que el campo de dirección destino se completa en este caso con todos "1".

- **Dirección Origen o Fuente** (6 $bytes$): 48 $bits$ que indican la dirección MAC de la estación que transmite la trama. Es la dirección que viene grabada en la memoria ROM de la placa, desde fábrica.

- **Longitud** (2 $bytes$): en tramas 802.3, en este campo se indica la longitud del campo de datos, expresada en $bytes$. En referencia a la Fig. 5.14, se puede observar que existe una longitud mínima y una longitud máxima para el campo de datos. La longitud mínima se debe a la necesidad de detectar colisiones al mismo tiempo que se realiza la transmisión. Este tiempo se relaciona con el tiempo de ida y vuelta de la Ec. (5.21). Considerando la extensión de 10 BASE 5 con cuatro repetidores, la separación máxima entre dos estaciones de la red es de $2500\,m$. El tiempo que demora la señal en recorrer esta distancia, adoptando una velocidad conservadora en el cable, es $\tau_d = \frac{d}{v} = \frac{2500m}{1.\frac{10^8 m}{s}} = 25\,\mu seg$, es decir que el tiempo de ida y vuelta, en el peor de los casos, para la detección de colisión es de $2\tau_d = 50\,\mu seg$. El estándar considera $2\tau_d = 51.2\,\mu seg$ ya que da un margen para el procesamiento de los cuatro repetidores que pueden existir en la distancia máxima permitida en 10 BASE 5. Durante todo este tiempo cualquier estación deberá estar transmitiendo y escuchando a la vez, por lo que el tamaño de la trama mínima deberá cumplir $2\tau_d \leq T_{Tx} = L/r_b = L/10\,Mbps$, donde L correspondería en este caso a la longitud mínima de la trama en bits, que resulta $51.2\mu seg \leq L/10Mbps$, o sea $L \geq 512$ bits. Esta longitud representa 64 $bytes$ repartidos entre los campos del encabezado y el de datos. En el encabezado contamos 6 $bytes$ de dirección destino, 6 de dirección fuente, 2 de longitud y 4 del la cola o *trailer* para el control de errores. Por lo tanto, de los 64 $bytes$, 18 $bytes$ son del protocolo, siendo lo que resta la longitud mínima de datos, de 46 $bytes$.

También, el estándar precisa una longitud máxima de $1500\ bytes$ para el campo de datos. Se trata de un valor máximo establecido para evitar el acaparamiento del medio por parte de cualquier estación que haya logrado acceder al mismo.

- **Tipo** ($2\ bytes$): en tramas *Ethernet* se trata de un código pre-definido que señala el protocolo encapsulado por la trama. Por ejemplo, uno de los códigos más conocidos es $0x800$, para indicar datagrama IP encapsulado.
  Dado los valores estandarizados que lleva el campo tipo, no es posible que se confunda esa información con la del campo longitud. Todos los códigos pre-definidos que puedan aparecer en el campo tipo se corresponden a equivalentes decimales superiores a 1500, que es el tamaño máximo de la trama. Por ejemplo, como se ha mencionado, el campo tipo para IP es el número $0x800$ en formato hexadecimal, que se corresponde con 2048 en decimal. De esta manera es posible la convivencia entre tramas *Ethernet* y 802.3 en la misma red. Esta posibilidad resulta equivalente a decir que las placas *Ethernet* y las placas IEEE 802.3 son compatibles, pudiendo compartir ambos tipos el entorno de una misma LAN.

- **Secuencia de Chequeo de Trama** (FCS, Frame Check Sequence) ($4\ bytes$): se trata de los bits de chequeo de un código cíclico CRC que cubre todos los campos de la trama, con excepción del preámbulo y el campo del comienzo SOF. El polinomio generador del CRC está normalizado y tiene grado 32, por eso son 4 los bytes del *trailer*.
  En la transmisión, estos cuatro bytes se presentan al final de la trama debido a la forma de generación, en el caso de la versión sistemática. En la recepción, se calcula el *checksum* y se detecta si hubo errores. Si no los hubo, la trama se analiza, verificándose si la dirección destino es propia tipo *unicast*, en cuyo caso debe ser la propia del receptor. También se pueden levantar tramas *multicast* si el dispositivo pertenece a ese grupo. Las tramas de *broadcast* es obligatorio recibirlas.
  Si hubo errores, la trama se descarta, sin avisar al transmisor. No existen tramas especiales de ACK, como en los esquemas de acceso al medio que se presentaron al principio. Esto conduce a reflexionar sobre qué sucederá en el caso de una transmisión con errores, ya que no existe una realimentación en este sentido entre la transmisión y la recepción. Por ahora sólo se adelantará que, en los casos en los que sea necesario, la confiabilidad quedará en manos de las capas superiores.

5.2.3 Subcapa MAC IEEE 802.3 – Direcciones MAC

Las direcciones MAC presentes en el encabezado descripto, son identificadores de $48\ bits$ que, en el caso del campo del campo de dirección

fuente, se corresponden de forma única a una placa de red o NIC. Las direcciones MAC se conocen también como direcciones físicas o de alcance local.

Existen tres numeraciones definidas por el IEEE para el nivel de enlace, todas diseñadas para ser identificadores globalmente únicos: MAC-48, EUI-48 y EUI-64.

En el caso de las direcciones MAC de $48\ bits$, el vendedor del equipamiento adquiere un Identificador Organizacionalmente Único (OUI, Organizationally Unique Identifier). Se trata de un identificador único de $24\ bits$ asignado por el propio IEEE. El OUI forma la primera mitad de la dirección física de la placa. Por su parte, el fabricante le asigna otro número único a los $24\ bits$ de menor orden. Como el OUI identifica al vendedor, puede ser útil cuando suceden determinados problemas con las placas de red.

Por tratarse de una dirección de tantos bits, se elige una representación especial para su formato. La dirección completa de $48\ bits$ se separa en 6 grupos de $8\ bits$ delimitados por ":". A su vez, cada grupo se divide en dos grupos de $4\ bits$, también conocido como *nibble*. Cada *nibble* se expresa en formato hexadecimal.

Por ejemplo, en la dirección MAC $00:a0:c9:14:c8:29$, el prefijo $00:a0:c9$ indica que el fabricante es Intel Corporation. El número $14:c8:29$ es la identificación que dicho fabricante asignó a esta placa en particular. Se trata de direcciones únicas a nivel mundial, puesto que son escritas directamente en el hardware en su momento de fabricación, grabándose en el chip de la memoria ROM de manera permanente. De todas maneras, cuando se inicia el equipo, la NIC copia la dirección a la memoria RAM.

Una trama *unicast* contiene la dirección única local MAC del dispositivo receptor en el campo dirección destino y, en el campo dirección fuente, la dirección única local MAC del dispositivo transmisor.

Una trama *broadcast* tiene como dirección destino, una dirección especial, que se conforma por medio de una cadena de "1" binarios. En formato MAC, esta dirección se expresa como $ff:ff:ff:ff:ff:ff$. Las tramas de *broadcast* llevan la dirección única local MAC del dispositivo transmisor en el campo dirección fuente.

Una trama *multicast* tiene como dirección destino una dirección especial que se puede distinguir observando el bit más significativo del byte más significativo. Si este bit es "0", la dirección es *unicast*. Si este bit es "1" y el resto también, la dirección es *broadcast*. En cualquier otro caso, se trata de una dirección *multicast*. El byte más significativo es el de más a la izquierda en la dirección. El bit más significativo es el de más a la derecha de dicho byte, conociéndoselo como bit distintivo entre dirección de grupo y dirección individual. Así definidos, algunos ejemplos posibles de direcciones *multicast* son $01:00:cc:cc:dd:dd$ o $09:00:aa:aa:bb:bb$, en tanto que direcciones como $00:01:44:55:66:77$ ó $08:00:22:33:44:55$ son ejemplos de direcciones *unicast*.

La manera más sencilla de obtener la dirección MAC de un dispositivo en un entorno con sistema operativo tipo Windows, es abrir una terminal desde

la línea de comandos con *cmd*, y luego usar la instrucción *ipconfig /all*, o también se puede usar el comando *getmac*. En entornos *Linux*, el comando es *ifconfig -a*.

5.2.4 Subcapa MAC IEEE 802.3 – Algoritmo de Retroceso Exponencial Binario

Una situación de colisión implica que dos o más computadoras estarían intentando transmitir al mismo tiempo. Si, una vez detectada la colisión, los protagonistas intentan retransmitir inmediatamente, muy probablemente ocurrirán nuevas colisiones. Si se elige un valor aleatorio de tiempo de espera, dentro de un rango razonable, se puede evitar este tipo de situaciones sin incurrir en demasiado retardo en el acceso.

El algoritmo de retroceso exponencial binario es un algoritmo de aplicación para métodos de acceso al medio, en casos donde se quiera evitar la congestión. Su objetivo es separar las retransmisiones en el tiempo. Aplicado a redes *Ethernet* sirve para retrasar el proceso de retransmisión en el caso de colisiones.

Cuando sucede una colisión, el protocolo divide el tiempo posterior en ranuras discretas cuya duración es $2\tau_d = 51.2\ \mu seg$, equivalente al tamaño de una trama mínima. El algoritmo establece que, luego de la primera colisión, la estación debe esperar 0 ó 1 ranuras antes de reintentar. Luego de la segunda colisión, la estación elige un número aleatorio de ranuras, entre $0, 1, 2$ ó 3 ranuras, antes de reintentar. Luego de la tercera colisión, la estación espera un número elegido al azar entre 0 y $(2^3 - 1)$ ranuras antes de reintentar. En general, luego de i colisiones, se escoge un número aleatorio de ranuras de espera en el rango 0 a $(2^i - 1)$. Tras 10 colisiones, el número aleatorio de ranuras permanece fijo entre 0 y 1023. Luego de 16 colisiones, el controlador de la placa considera que no es posible la comunicación y abandona el intento de acceso. La recuperación queda en manos de capas superiores.

Este algoritmo permite una adaptación dinámica en el caso de varias estaciones tratando de transmitir a la vez. Si el intervalo de espera aleatoria fuera corto, la probabilidad de colisión no se vería mejorada en aquellos casos donde muchas estaciones se encuentran tratando de acceder al medio. Aumentar el tiempo de espera permite disminuir la probabilidad de colisión, pero puede aumentar considerablemente el retardo en el tiempo de acceso. El algoritmo trata de adaptarse dinámicamente a la carga de tráfico, asegurando un retardo pequeño en el caso de pocas estaciones en situación de colisión, pero también tiempos razonables en el caso de muchas estaciones.

5.2.5 Subcapa MAC IEEE 802.3 – Algoritmos de Transmisión y Recepción

Se puede graficar el método de acceso al medio por medio de diagramas de flujo que resumen los pasos a tener en cuenta en el caso de transmisión y recepción de tramas, como se indica en las Fig. 5.15 y Fig. 5.16, respectivamente.

Se observa en el diagrama de transmisión que, una vez que se ha detectado que el medio se encuentra libre, no se transmite de inmediato. Los dispositivos deben respetar un tiempo mínimo entre transmisiones de tramas que se conoce como Espacio entre Tramas (IFG, Inter Frame Gap). Se trata de un tiempo breve, entre tramas, que permite a los dispositivos prepararse para la recepción de la siguiente trama. Para las redes de $10\ Mbps$, este tiempo se establece en $9.6\ \mu s$, equivalente a $96\ bits$, que se corresponden con $64\ bits$ del preámbulo y $32\ bits$ de la secuencia de *jamming*.

En el caso de detectar colisión durante la transmisión, la estación abortará y enviará una trama especial de $32\ bits$ que se conoce como secuencia de *jamming*. El propósito es que todos los dispositivos se enteren de la colisión. También se observa que, luego de una colisión, se activa el algoritmo de retroceso exponencial binario.

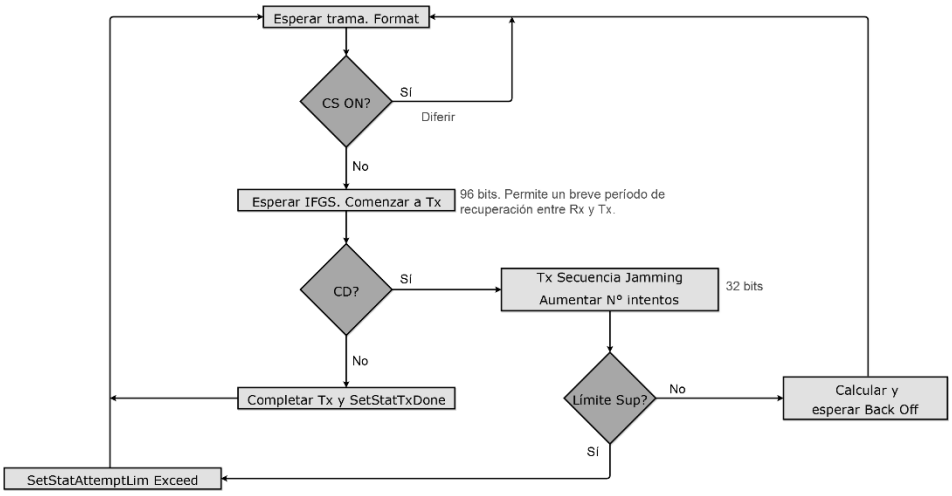


Figura 5.15 - Transmisión de tramas.

En el caso de la recepción, interesa verificar que la trama no contenga errores y que sea dirigida al dispositivo receptor, a un grupo al que éste pertenezca o a todos los dispositivos de la red.

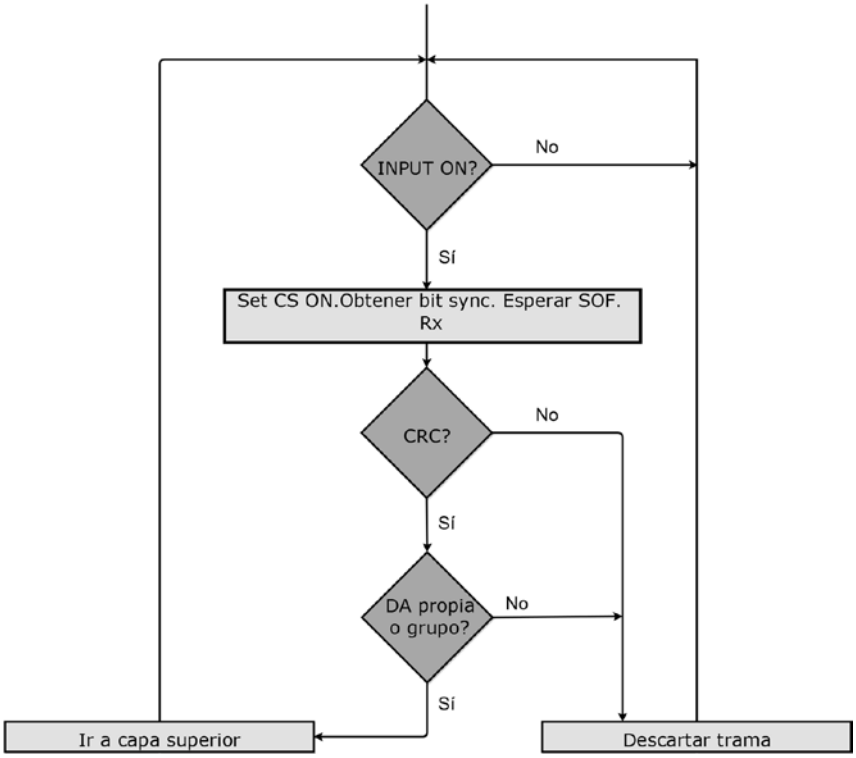


Figura 4.16 - Recepción de tramas

5.2.6 Subcapa MAC IEEE 802.3 – Eficiencia

Es de interés encontrar una medida de eficiencia del método de acceso al medio del protocolo IEEE 802.3 con el fin de comparar este método con otros previos. En estos casos, se puede pensar la eficiencia como la relación que existe, para cada estación, entre el tiempo que le lleva bajar una trama al medio o tiempo de transmisión T_{Tx} , y el tiempo total acumulado, que incluye el tiempo que demoró en promedio en acceder al medio, debido al propio método de acceso.

Deberían realizarse varias simplificaciones para poder llegar a una aproximación más fácilmente. Por ejemplo, ya que el análisis riguroso del algoritmo de retroceso exponencial binario es complejo, se asume una probabilidad constante de retransmisión en cada ranura. También se supone un modelo de k dispositivos, siempre listos para transmitir. El tiempo de una ranura es $2T = 51.2\ \mu seg$, ya que se trata del tiempo de duración que mínimamente debe tener una trama para que el método sea efectivo. Se supondrá que cada estación transmite durante una ranura de contienda $2T$ con probabilidad p.

Interesa calcular la probabilidad de que una estación adquiera el canal durante el tiempo de duración de una ranura T. Se trata de la probabilidad A, que refleja la posibilidad de que una estación ocupe el medio con probabilidad p en el tiempo de duración de una ranura, mientras las restantes $(k-1)$ estaciones no

ocupan el medio en ese mismo tiempo, con probabilidad $(1-p)$. Resulta así que A tiene una distribución binomial:

$$A = \binom{k}{1} p\,(1-p)^{k-1} \tag{5.22}$$

El valor máximo de A se alcanza cuando la probabilidad de acceso es la misma para todas las estaciones, es decir $p = 1/k$, resultando entonces:

$$A_{máx} = k\,\frac{1}{k}\,(1-\frac{1}{k})^{k-1} = (1-\frac{1}{k})^{k-1} \tag{5.23}$$

Si el número de estaciones es muy grande, se puede plantear:

$$A_{máx} = \lim_{k\to\infty}\left(1-\frac{1}{k}\right)^{k-1} = \frac{1}{e} \tag{5.24}$$

O sea que, bajo este supuesto, la probabilidad máxima de que una estación ocupe el medio durante el tiempo de duración de una ranura es de 36.8%:

$$A_{máx} = \tfrac{1}{e} = 0.368. \tag{5.25}$$

La duración promedio del período de contienda depende de esta probabilidad, ya que si se denomina P_j a la probabilidad de que un período de contienda consista de j ranuras, resulta:

$$P_j = A\,(1-A)^{j-1} \tag{5.26}$$

La Ec. 5.26 resulta de considerar que, durante las primeras $(j-1)$ ranuras, el dispositivo no puede tomar el medio y recién lo toma en la ranura j con probabilidad A. De este modo, el número promedio de ranuras por período de contienda es:

$$E = \sum j.P_j = \sum j\ A\,(1-A)^{j-1} = 1.A + 2.A\,(1-A) + 3.A\,(1-A)2 + \cdots = 1/A \tag{5.27}$$

Considerando el valor de $A_{máx}$, el número promedio de ranuras por período de contienda resulta ser $E = e = 2.71$ que, traducido en unidades de tiempo se convierte en $2Te = 138.75\ \mu seg$.

Considerando estos resultados, se puede medir la eficiencia del método CSMA/CD como:

$$\rho = \frac{T_{Tx}}{T_{Tx} + 2Te} = \frac{\frac{L}{r_b}}{\frac{L}{r_b} + 2Te} = \frac{1}{1 + \frac{2Ter_b}{L}} = \frac{1}{1 + \frac{2der_b}{c.L}} \tag{5.28}$$

Obsérvese con detención los factores de los que depende el comportamiento de la red. En la ecuación de rendimiento (Ec. 5.28), d es la distancia máxima entre dos estaciones, r_b la velocidad binaria, L la longitud de las tramas en bits y c la velocidad de propagación en el cable. De todos estos factores, los únicos modificables en términos de diseño de nuevas redes son d, r_b y L. El estándar IEEE 802.3 fija $r_b = 10Mbps$ y $d_{máx} = 2500m$. Si se adopta $c = 2.10^8 m/s$, tomando la longitud de las tramas como parámetro, se pueden graficar curvas de rendimiento como las presentadas en la Fig. 5.17. Al graficar, no se ha considerado el valor máximo $A_{máx}$, sino que se ha expresado el valor de la probabilidad A en términos de la cantidad de estaciones k, tal como se presenta en la Ec. 5.23. Por este motivo, cuando la cantidad de estaciones es muy grande, las curvas tienden a establecerse en un valor constante.

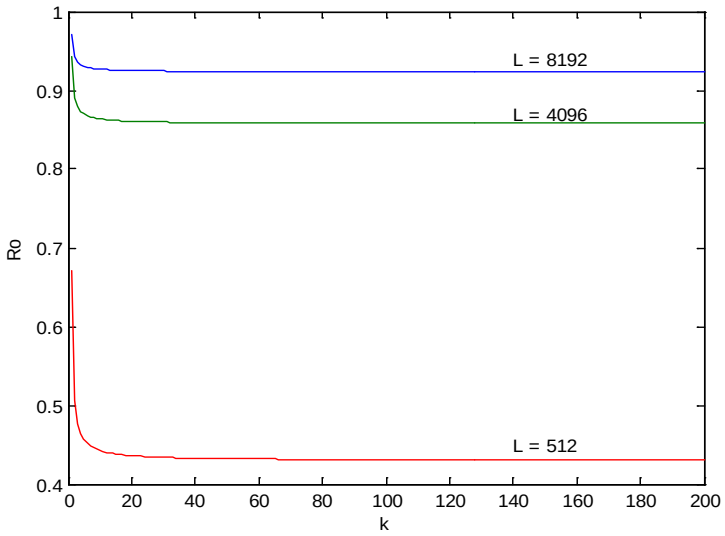


Figura 5.17 - Eficiencia $\rho = \frac{T_{Tx}}{T_{Tx}+2Te} = \frac{\frac{L}{r_b}}{\frac{L}{r_b}+2(dmáx/c)(1-\frac{1}{k})^{k-1}}$

Como se puede observar en la Fig. 5.17, el sistema se comporta mejor cuando las tramas son más largas.

5.2.7 Subcapa MAC IEEE 802.3 – Reglas de Instalación

Al instalar una red tipo *Ethernet*, conviene dejar claro que existen recomendaciones para su buen funcionamiento.

Un dominio de colisión en una red de este tipo, se refiere a un único sistema *half duplex Ethernet* cuyos elementos, ya sea cables, repetidores, estaciones o interfaces, forman parte del mismo dominio de señalización en el tiempo. Se trata de un entorno donde, si dos o más dispositivos transmiten al mismo tiempo se genera una situación de colisión. Un dominio de colisión puede implicar varios segmentos si los mismos se encuentras unidos por repetidores. Un ejemplo típico de dominio de colisión lo constituyen las redes tipo *bus*, en sus versiones 10 BASE 5 y 10 BASE 2. En las redes de par trenzado con topología tipo estrella, un dominio de colisión queda representado por un *hub* y todas las máquinas que se conectan con sus puertos.

Una disposición, conocida como regla 5-4-3 es aplicable a un dominio de colisión. En la Figura 5.18, se puede observar en el lado derecho una red inválida, con 6 saltos entre dispositivos finales, conectados mediante *hubs*. También, en la misma figura, se puede observar una red válida del lado izquierdo, en donde aparece un nuevo elemento, denominado conmutador de red LAN o *switch*, colocado en medio del camino entre 4 *hubs*. El *switch* permite dividir la red en más de un dominio de colisión. Por el momento, se definirá la funcionalidad de un *switch* como la de un dispositivo con un único dominio de colisión por cada puerto de conexión. Se trata de un elemento con componentes en capa física y a nivel de enlace, a diferencia de un *hub*, que es un dispositivo sólo de capa física.

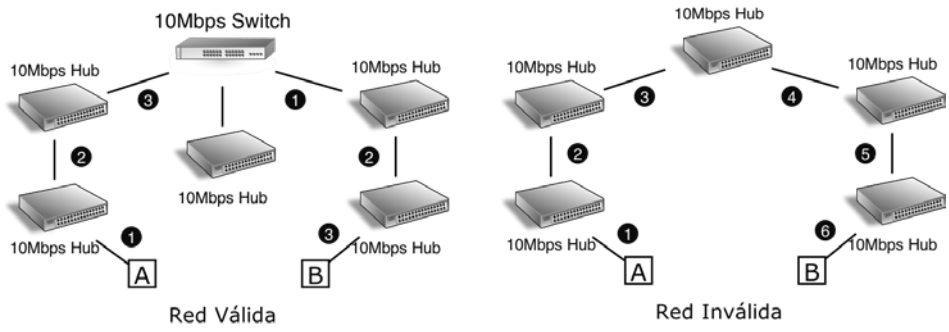


Figura 5.18 - Tendido válido e inválido de red *Ethernet*.

IEEE 802.3 y *Ethernet* implementan la regla 5-4-3, limitando el número de repetidores y segmentos en accesos compartidos en topología de árbol o apilado de repetidores. La regla considera dos tipos de segmentos físicos: los que tienen usuarios conectados y los que no los tienen. Estos últimos se usan para conectar repetidores entre sí. La regla dicta que entre dos nodos cualquiera de la red, sólo puede haber un máximo de cinco segmentos, conectados a través de cuatro repetidores o concentradores, y solamente tres de los cinco segmentos pueden contener conexiones de usuarios. Esta regla asegura que una señal

enviada sobre la LAN alcance cada parte de la red dentro de un tiempo específico. La limitación en la cantidad de repetidores se debe a que cada uno de ellos añade una pequeña cantidad de tiempo al proceso. En definitiva, la regla 5-4-3 establece que, entre dos equipos de la red, no podrá haber más de cuatro repetidores y cinco segmentos de cable. Además, dos de los cinco segmentos sólo pueden ser empleados para la interconexión entre repetidores.

Si se observa la Fig. 5.18, la red de la derecha es inválida porque se forma una cadena que viola la regla 5-4-3. Sin embargo, colocar un *switch* en una cadena de *hubs*, tiene el efecto de volver la cuenta a cero, como sucede en la red de la izquierda de la figura. Por eso se dice que los *switchs* sirven para segmentar redes LAN.

También, en las redes de este tipo deben respetarse las reglas de cruzado de cables mencionadas previamente. Un cableado incorrecto, una red superpoblada por exceso de nodos o una extensión excesiva de la red en cuanto a distancia, seguramente conducirán a problemas de funcionamiento.

5.3 Administración de una red LAN tipo *Ethernet*

A medida que evolucionó la tecnología de de los dispositivos conectados a una red, en cuanto a su velocidad de procesamiento, comenzaron a aparecer problemas de congestión en las redes LAN. La congestión es un fenómeno estadístico que es función de los patrones de tráfico, manifestándose a través de una serie de síntomas, entre los más notables, el retardo de acceso.

Cuando existe sobrecarga de corto plazo, la red LAN distribuye la carga a lo largo del tiempo. Si la carga es liviana, el tiempo promedio de transmisión de una trama será corto. Cuando existen picos importantes de carga, aumentará el retardo promedio, también llamado tiempo de servicio. Esta situación se traducirá en un efecto que hará que la red parezca más lenta para la percepción del propio usuario.

Es difícil medir los tiempos de servicio porque sólo se puede realizar usando software especializado, configurable sobre cada placa, pero sí es posible medir otros parámetros de operación de una LAN. Algunos de estos parámetros son medidos de manera automática por controladores estándar y *software* de aplicación. Otros, requieren equipamientos especiales para monitoreo, tales como analizadores de protocolo y monitores remotos.

Entre las métricas más importantes para medir una situación de congestión, se pueden mencionar la utilización del canal, la cantidad de colisiones, la degradación de la performance de las aplicaciones y la insatisfacción del usuario.

La utilización del canal se refiere al porcentaje de tiempo en que el canal está ocupado transmitiendo datos de manera exitosa. Se relaciona directamente con la carga ofrecida, aunque muchas variables influyen en la consideración de un valor de utilización aceptable. El número de estaciones en la LAN, el comportamiento de las aplicaciones, los patrones de tráfico, la distribución estadística de la longitud de las tramas y el tamaño de la red, entre otras

circunstancias, afectan los valores de utilización. A pesar de esto, de acuerdo a la experiencia, se pueden usar algunos niveles de utilización como reglas prácticas para determinar si la situación de la LAN se acerca a la congestión. Importa también el momento en que se haya tomado la apreciación. Valores por encima del $10-20\%$ promediados sobre una jornada laboral de 8 horas, o del $20-30\%$ promediados sobre la peor hora del día, así como valores superiores al 50%, promediados sobre los peores 15 minutos del día, son indicadores de problemas de congestión.

En términos más generales, como se puede observar en la Fig. 5.19, cuando los tiempos de respuesta son cortos, la utilización se encuentra por debajo del 35% y la performance de la red es la esperada. Ni bien nos aproximamos al 35%, la red comienza a hacerse más lenta. Superado este valor, se puede hablar de congestión, con tiempos de respuesta muy lentos.

En cuanto a la cantidad de colisiones, un incremento de este valor es una medida indirecta de la carga ofrecida a la LAN. En el caso de redes con muchos usuarios y grandes volúmenes de tráfico, aumenta la contienda por el ancho de banda disponible, aumentando la cantidad de colisiones. Se conoce como relación de colisiones al número de tramas que colisionan respecto del valor total, considerándose 10% como un valor razonable. En realidad, las colisiones consumen pequeños porcentajes de la capacidad del canal, cuando las condiciones de carga son entre moderadas y altas. Por este motivo, generalmente se da más importancia a otras estadísticas.

Con respecto a la degradación de la performance de las aplicaciones, en una LAN que se encuentra en situación de congestión, las transferencias de archivos se hacen más lentas y los accesos a los servidores se presentan de manera impredecible. Si la carga es muy elevada, es posible que comience a haber fallas tales como desconexiones, tiempos de espera agotados, caídas y necesidad de reinicios. Lo importante de resaltar en estos casos es que no necesariamente una degradación implica congestión de la propia red. Podrían existir otros problemas de comunicación debido a la performance del sistema sobre el cual funciona la aplicación, ya sea por fallas de la CPU, la memoria y/o el acceso al disco de los servidores y también por el número de usuarios. Es decir que la LAN, muchas veces no es el único cuello de botella posible.

La medida de la insatisfacción del usuario es muy importante para el administrador. Si los usuarios no están satisfechos con la performance del sistema, ninguna de las estadísticas que se les ofrezcan será capaz de convencerlos. La reacción de los usuarios es la métrica más importante de la performance de una red aunque no indique un problema de congestión sobre la LAN, ya que el concepto de red, para el usuario, incluye aplicaciones, servidores, pilas de protocolos, dispositivos de interconexión y muchas otras cuestiones que no tienen que ver directamente con la propia red.

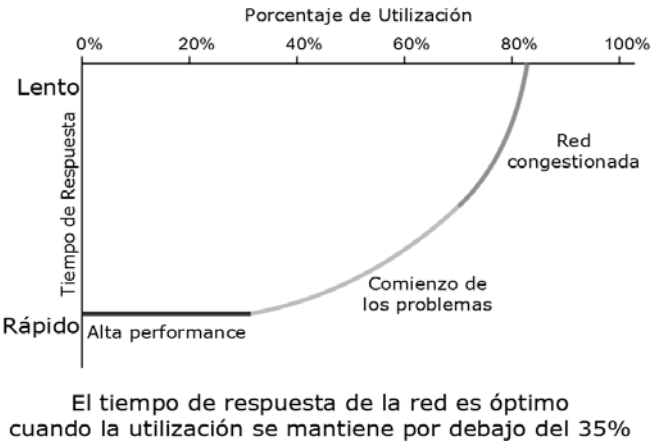


Figura 5.19 - Tiempo de respuesta vs porcentaje de utilización

Una solución a una situación de congestión instalada, podría ser aumentar el ancho de banda. Si una red LAN de 10 $Mbps$ tiene una utilización promedio del 50% durante horas, esto se puede interpretar como una indicación fuerte de congestión, lográndose bajar este valor al 5%, si simplemente se pudiera aumentar la capacidad a 100 $Mbps$. Esto resuelve el problema aunque genera dificultades porque, partiendo de una base instalada 10 BASE T, probablemente deberíamos re-cablear la red y cambiar las placas de los equipos, aumentando los costos y generando interrupciones.

Otra solución a una situación de congestión instalada podría ser segmentar la red. Segmentar una LAN significa dividirla en diferentes dominios de colisión. De este modo, se genera más de un dominio de colisión *Ethernet* para el mismo número de usuarios y dispositivos. La segmentación debe considerar aspectos tales como la distribución del patrón de tráfico, para que la mayor carga de cada grupo quede dentro de su propio segmento. Una solución de segmentación exige la instalación de nuevos dispositivos: *bridges* y/o *switches.*

5.4 Cableado Estructurado

A medida que el uso de redes cableadas se extendía, comenzó a notarse la necesidad de desarrollar un cableado genérico para transporte de voz y datos en red.

Proveer un servicio de red eficiente y flexible es una tarea complicada a medida que aumenta el tamaño de la red. Al diseñar un sistema de cableado es importante planificar pensando a futuro, en cuanto al propio crecimiento y la aparición de nuevas tecnologías.

Un sistema de cableado estructurado se basa en una serie de segmentos de cable, instalados de acuerdo a los estándares. El sistema involucra una jerarquía basada en cables de *backbone* o cableado vertical, que transportan

señales entre armarios de distintos pisos, y cableado horizontal, que se distribuye en cada piso, desde el armario a cada una de las estaciones, tal como se presenta en la Fig. 5.20.

En Octubre de 1995, la Asociación de Industria Electrónica/Asociación de Industria de las Telecomunicaciones (Electronic Industry Association/Telecommunications Industry Association) publicó el primero de estos estándares para cableado de telecomunicaciones en instalaciones comerciales, EIA/TIA 568A. El objetivo fue proveer un sistema de cableado genérico, independiente de los vendedores, para soporte de voz y datos. En el año 2000, se publicó el estándar EIA/TIA 568B, conteniendo las modificaciones a la norma anterior que contemplaran cambios tecnológicos.

TIA/EIA 568 lista seis elementos básicos de un sistema de cableado estructurado:

- Posibilidades de entrada al edificio.
- Espacio para los equipos.
- Cableado vertical o de *backbone*.
- Armario de telecomunicaciones.
- Cableado horizontal.
- Áreas de Trabajo.

El estándar especifica un sistema de *backbone* con topología estrella que no tiene más de dos niveles de jerarquía dentro del edificio. Es decir que un cable no debería pasar por más de un dispositivo de conexión cruzada entre el Cruzado Principal (MC, Main Cross-connect) en la habitación de equipos, y el Cruzado Horizontal (HC, Horizontal Cross-connect), localizado en el armario. Se eligió una topología estrella por las ventajas que la misma presenta.

La evolución de la norma original, define actualmente los siguientes componentes del cableado estándar: ccableado de par trenzado y fibra óptica, tomas de 8 pines para conexión de componentes a la red, elementos denominados *patch pannels* donde termina el cableado y es posible administrar la topología de manera flexible y también efectuar tareas de mantenimiento y de expansión y, por último, conexiones de cruce entre distintos cables sobre el propio *patch pannel*. La Fig. 5.21 presenta alguno de los elementos mencionados.

El cable más común y de menor costo es el Par Trenzado no Apantallado (UTP, Unshielded Twisted Pair) aunque, como se ha visto en el capítulo previo, existen versiones blindadas. También, EIA/TIA definió varias categorías de cables, según sus prestaciones y características técnicas.

La Fig. 5.23 presenta la denominación de pines del conector RJ45 y la coloración de pares según la norma.

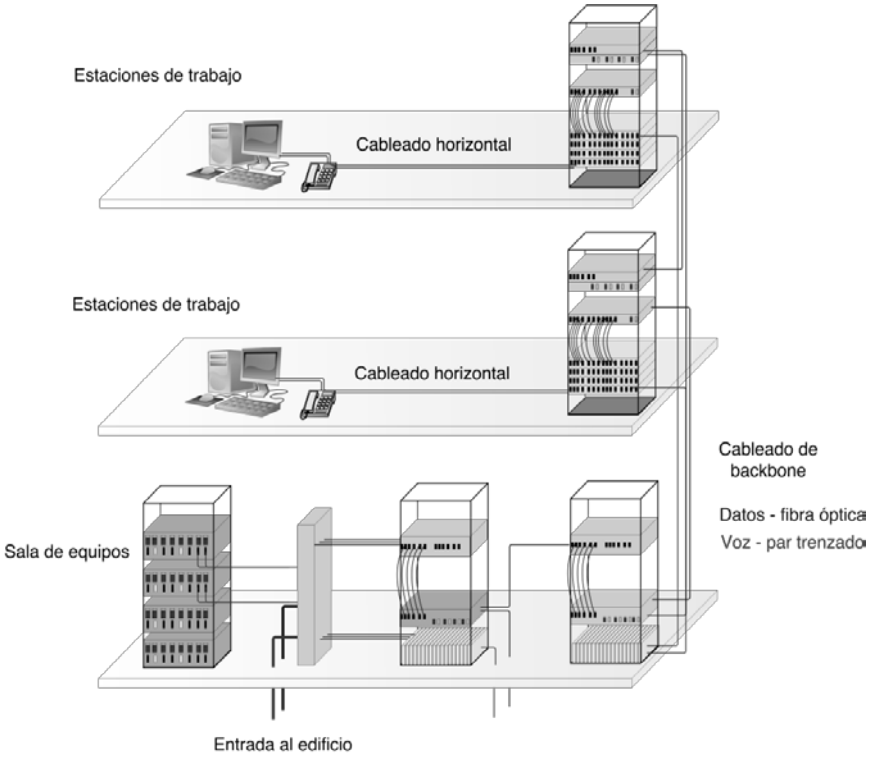


Figura 5.20 - Elementos de Cableado Estructurado.

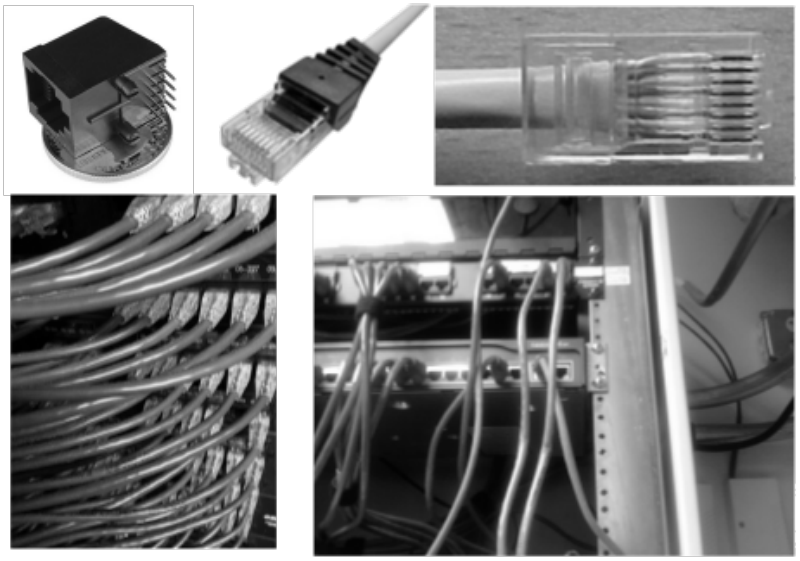


Figura 5.21 - Toma de 8 pines, Conector RJ45 y Patch Pannel.

Por otra parte, los estándares establecen que, en cada piso de un edificio, debe existir al menos un distribuidor, a partir del cual se instala el cable hasta el toma cercano a cada estación de trabajo. El distribuidor puede ser un *hub* o un *switch*. Este cableado es el que se conoce como cableado horizontal o *channel link*. La longitud máxima total de este canal no debe superar los 100 metros, como se grafica en la Fig. 5.23.

Es importante destacar que, al momento de tender una red, también es recomendable tener en cuenta la documentación. Siempre será útil dibujar un plano del tendido, por piso, con identificadores apropiados, y llevar en paralelo una anotación con los detalles de los identificadores. Es deseable etiquetar los cables en ambos extremos al momento de la instalación. Un plano de cableado y la documentación apropiada se convierten en herramientas fundamentales para la administración y mantenimiento de la red, sobre todo en el caso de fallas de funcionamiento.

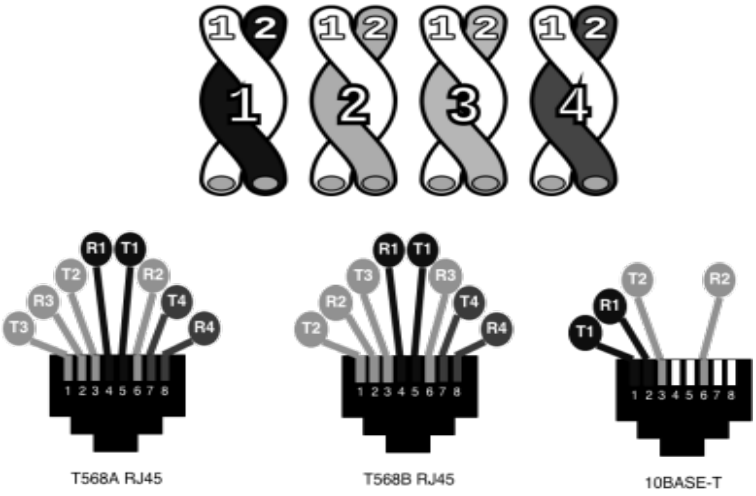


Figura 5.22 - Numeración e identificación de pares y de conectores RJ45

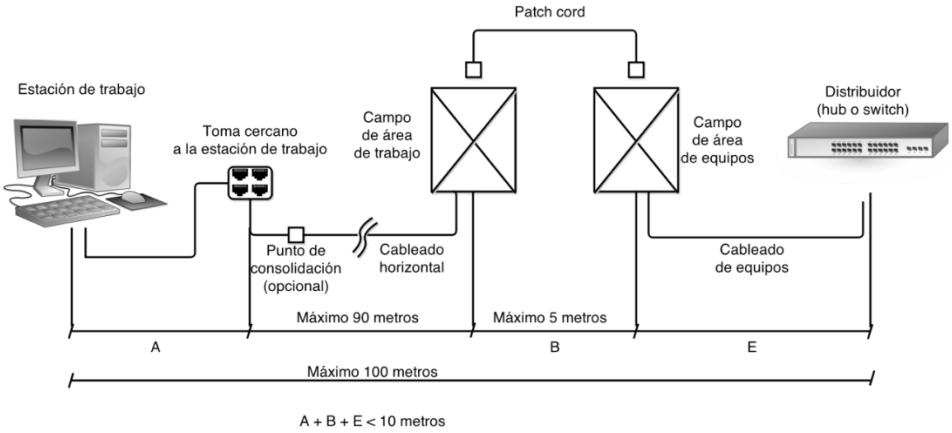


Figura 5.23 - *Channel Link.*

Bibliografía

1. Stallings, William, “Comunicaciones y Redes de Computadores”. Sexta Edición. Prentice Hall Inc., 2000.
2. Sklar, Bernard, “Digital Communications. Fundamentals and Applications”. Second Edition. Prentice Hall Inc., 1988.
3. Castiñeira Moreira, Jorge and Farrell, Patrick Guy, “Essentials of Error-Control Coding”. Wiley, September 2006.
4. Castiñeira Moreira, Jorge and Farrell, Patrick Guy, “Codificación para el Control de Errores”. Eudem, 2012.

Problemas

1. Un grupo de estaciones comparte un canal Aloha puro de $64\ kbps$. Cada estación transmite exitosamente, en promedio, un paquete cada $10\ seg$. Cada paquete es de $3500\ bits$. ¿Cuál es el número máximo de estaciones que pueden compartir este canal?

2. Suponiendo que la transmisión y retransmisión de paquetes puedan ser descriptos como procesos de Poisson, calcule la probabilidad de que una transmisión de un paquete de datos colisione con otro en un sistema Aloha con ranuras. Suponga un tráfico total $\lambda = 20\ paquetes/seg$ y que la duración total de un paquete es $T = 15\ ms$.

3. Explique brevemente las diferencias entre IEEE 802.3 CSMA/CD y Ethernet. En cualquier caso: ¿Dónde está la capa física y dónde la capa de enlace en una Ethernet o CSMA/CD? Cuál de las capas maneja la diferencia entre los protocolos Ethernet y CSMA/CD? ¿Cómo lo hace?

4. En CSMA/CD ¿Solamente las estaciones que se ven involucradas en una colisión se enteran de lo sucedido? Luego de una colisión, una estación transmisora arranca un algoritmo de retroceso exponencial binario ¿En cuánto tiempo ganará el medio?

5. Una red CSMA/CD ha sido diseñada para operar a $1\ Gbps$ a través de un cable de $1\ km$, con una velocidad de propagación de $200\ m/\mu seg$). Calcule el tamaño de la trama mínima con que se puede operar. ¿Cuál sería el tamaño máximo?

6. ¿Qué distingue los distintos protocolos CSMA? ¿Cuál de ellos tiene mejor eficiencia y porqué? Calcular la eficiencia de IEEE 802.3 para tramas de $1000\ bytes$ de longitud.

7. Explique el significado de los siguientes términos en el contexto CSMA/CD:

   a) Tiempo de ranura.
   b) Secuencia de interferencia.
   c) Espacio entre tramas, IFS.
   d) Modo de Difusión.
   e) Detección de Portadora.
   f) Dominio de colisión.
   g) Regla 5-4-3.

8. Explique los parámetros que se utilizan para medir la performance de operación de una LAN.

9. Responda si las siguientes aseveraciones son Verdaderas o Falsas. En todos los casos justifique su respuesta, realizando cálculos cuando corresponda.

   a) Si la longitud promedio de las tramas en LAN 802.3 supera en 8 veces la longitud de la trama mínima, la eficiencia del sistema supera el 80%.
   b) Sean dos dispositivos A y B sobre el mismo segmento Ethernet de $10\ Mbps$. El retardo entre ellos es equivalente a la duración de $225\ bits$. El nodo A comienza a transmitir una trama y, antes de que finalice, el nodo B comienza su propia transmisión. A detecta situación de colisión y no puede terminar de transmitir la trama mínima.
   c) En 10 BASE 5, la longitud máxima de $2500\ m$ establece un preámbulo de $64\ bits$.
   d) En 10 BASE T, la distancia máxima entre dos estaciones conectadas a través de un *hub* es de 100 m.
   e) La detección de colisiones en 10 BASE T es igual que en 10 BASE 5.
   f) Una dirección de *broadcast* no puede ser Dirección MAC fuente.
   g) La máxima cantidad de máquinas en una red 10 BASE T armada con 5 *hubs* de 8 bocas cada uno, es de 40.
   h) Para enviar $12\ bytes$ de datos encapsulados en una trama Ethernet, se la debe rellenar con $34\ bytes$.

CAPÍTULO VI

Redes LAN Cableadas de Alta Velocidad

Las mejoras en capacidades de almacenamiento y de procesamiento de las computadoras, sumado al incremento en la cantidad de usuarios y al desarrollo de nuevas aplicaciones asociadas a consumos intensivos de ancho de banda, impusieron la necesidad de mayores velocidades en las redes cableadas. Al principio, el propósito de diseño se centró en mantener el mismo método de acceso al medio y formato de tramas que en el estándar IEEE 802.3, pero a mayor velocidad. Fast Ethernet o Ethernet de alta velocidad es el nombre de una serie de estándares de IEEE de redes Ethernet de 100 Mbps. Por su parte, Gigabit Ethernet es el nombre que se da a las redes capaces de alcanzar 1000 Mbps.

En IEEE 802.3, la distancia máxima entre estaciones, definida para 10 BASE 5 de cable coaxial de cinco segmentos con cuatro repetidores, se definió en 2500 m. Esta distancia implicaba un tiempo de ranura para la velocidad elegida, imponiendo un tamaño de trama mínima. Si el objetivo de una nueva tecnología fuera aumentar diez veces la velocidad, manteniendo el retardo de propagación según los valores establecidos, se debería disminuir la longitud máxima de la red a la décima parte.

Por su parte, en una arquitectura de hub 10 BASE T, los segmentos se reducen a dos como máximo, conectados a través del propio hub, cada uno de 100 m. O sea que la distancia máxima entre dos estaciones es de unos 200 m, obteniéndose una longitud máxima comparable a la que se precisaría al aumentar diez veces la velocidad. En este sentido, se podría razonar que existe una ventaja, en términos de distancia, que favorecería el aumento de velocidad.

La contracara de esta ventaja es que los cables UTP utilizados en 10 BASE T son de Categoría 3, con ancho de banda suficiente en las distancias mencionadas como para acomodar la velocidad de 10 Mbps en formato Manchester. En esta topología se usan sólo dos pares, uno para transmitir y otro para recibir, pero en modo half duplex. De todos modos, el ancho de banda del cable Categoría 3 es insuficiente para transmitir a una velocidad diez veces mayor, a menos que se modifique la codificación o se utilicen más pares. Por su

parte, los cables más modernos de Categoría 5, con ancho de banda de unos 100 MHz, no permitirían la transmisión de 100 Mbps con la misma codificación Manchester, por lo que igualmente se debería modificar el esquema de codificación.

Todos estos temas serán abordados en el desarrollo de este capítulo. Se mencionarán las modificaciones que debieron realizarse para lograr la velocidad de 100 Mbps y de 1Gbps desarrolladas en las llamadas redes Fast Ethernet y Gigabit Ethernet. El capítulo culmina analizando las características técnicas de un conmutador LAN comercial y las capacidades que se precisan para poder configurar una red LAN virtual.

6.1 100 BASE T4

Fue una de las primeras implementaciones de *Fast Ethernet* ideadas para par trenzado. El estándar 100 BASE T4, requería la utilización de los cuatro pares de cable trenzado, que podían ser de Categoría 3, igual que en la red original de 10 $Mbps$. De los cuatro pares, un par se reservaba para transmitir, otro para recibir, y los otros dos conmutaban entre transmisión y recepción, de tal manera que la comunicación se establecía transmitiendo en simultáneo sobre tres pares. La Fig. 6.1 representa la conexión entre una computadora ó Equipo Terminal de Datos (DTE, Data Terminal Equipment) y un *hub* ó Equipo de Circuito de Datos (DCE, Data Circuit Equipment), a través del cable UTP de cuatro pares. Por definición, en el estándar 100 BASE T4, los pares 3 y 4 son bidireccionales, en tanto que los pares 1 y 2 son unidireccionales.

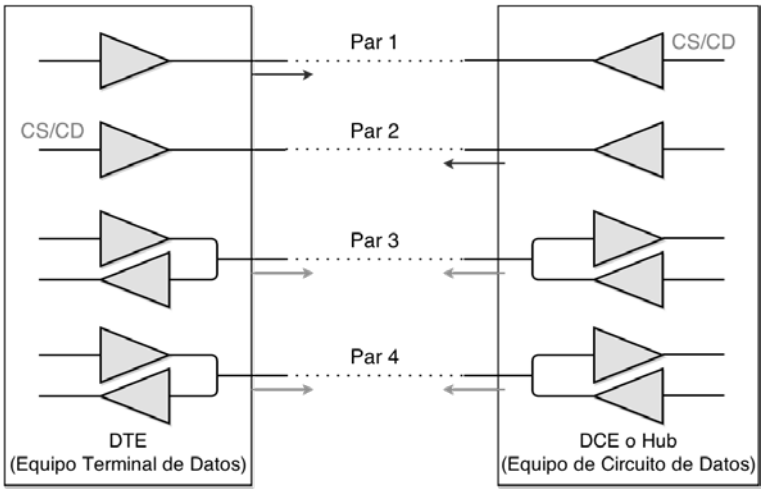


Figura 6.1 - Pares UTP en 100 BASE T4.

La propuesta consideraba que, cuando el DTE transmitiera hacia el *hub*, se utilizaran los pares 1, 3 y 4, separando el par 2 para detectar colisiones. En el

caso del *hub*, se transmitía por los pares 2, 3 y 4, en tanto que el par 1 se utilizaba para detectar colisiones. En cualquier caso, los pares 1 y 2 se usaban para detección de colisiones y detección de señal portadora, manteniendo el método de acceso *half duplex*, característico de 10 BASE T.

Es decir que 100 BASE T4 mantuvo el uso de cables Categoría 3 UTP y conectores son RJ45 pero, a diferencia de 10 BASE T, utilizaba los cuatro pares disponibles, de allí la denominación T4. En 100 BASE T4, para manejar un flujo de $100\ Mbps$, los diseñadores pensaron en dividirlo en tres partes, usándose tres pares para transmitir en cada dirección. Si la información se bajara directamente sobre los tres pares, cada par debería trasladar un flujo de $33.33 Mbps$. Si para ello se empleara la misma codificación banda base tipo Manchester, como en 10 BASE T, se excedería el límite de ancho de banda del cable UTP Categoría 3. Por este motivo, se impone algún tipo de codificación para adaptación de la señal al medio.

Con este propósito, el estándar asignó una codificación de tres niveles conocida como 8B6T, donde cada bloque de $8\ bits$ es convertido en $6\ símbolos$ ternarios. Un símbolo ternario puede adoptar para su transmisión uno entre tres niveles: $(+V), (0) y (-V)$. Como la transmisión multinivel debe ser de velocidad apropiada para el ancho de banda del cable, se eligió una relación de equivalencia de tiempos entre la duración de una palabra de $8\ bits$ y la duración de dos símbolos ternarios. Si se simboliza como T_b a la duración de un bit y D la de un símbolo ternario, se debe verificar que la duración de $8\ bits$ sea comparable a la duración de dos símbolos ternarios, o sea:

$$8T_b = 2D \tag{6.1}$$

Si se tiene en cuenta que la inversa de T_b es la velocidad binaria correspondiente al flujo de la información $r_b = 100 Mbps$, y que la inversa de D, denominada r, se corresponde con la velocidad de los símbolos en el medio, se verifica la siguiente relación:

$$r = \frac{1}{4} r_b = 25\ Msímbolos/seg \tag{6.2}$$

Se observa que esta velocidad de señalización cumple perfectamente con el ancho de banda del cable, de aproximadamente $30 Mhz$.

Por otro lado, la correspondencia respecto de la codificación 8B6T, entre palabras de $1\ byte$ y palabras de $6\ símbolos$ ternarios ha de ser unívoca. Agrupando de a 8 bits, se pueden generar hasta $2^8 = 256$ palabras diferentes. Como con 6 símbolos ternarios se pueden componer hasta $3^6 = 729$ palabras, sobrarían 473 palabras ternarias.

El criterio aplicado en el estándar fue que, ya que sobraban palabras de entre las 729 posibles, lo prudente sería elegir aquellas que mejor se adaptaran a la transmisión. Generalmente, la idea principal es eliminar en lo posible la

componente de continua y elegir aquellas palabras que posean al menos dos transiciones, para asegurar la sincronización del receptor. Este propósito lo veremos repetidamente en otros estándares.

Se denomina peso de una palabra a la suma de las tensiones normalizadas de sus símbolos. Así, una palabra de 6 símbolos ternarios, por ejemplo $(+ + + - - -)$ tendría peso (0), una palabra $(0 + + - - -)$ tendría peso (-1), y una palabra $(+ + + - -0)$ tendría peso $(+1)$. Para fortalecer la idea de eliminar la componente de continua, el estándar eligió seleccionar aquellas palabras de peso (0) ó $(+1)$.

Sólo 267 palabras de entre las 729 cumplen esta condición. Si se eliminan de este grupo aquellas que tienen menos de dos transiciones, quedarían 262 palabras. Si de estas 262 palabras se eliminan las que comienzan con cuatro ceros consecutivos, restarían 256 palabras, exactamente la cantidad que se precisa para codificar las palabras de 8 $bits$ que proceden del flujo de 100 $Mbps$. A modo de ejemplo, en la Tabla 6.1 se presentan las primeras 20 palabras del código.

Tabla 6.1 - Primeras palabras del Código 8B6T

Datos	Código	Datos	Código
00	-+00-+	0A	0-+0+-
01	0-+-+0	0B	0-+-0+
02	0-+0-+	0C	-++-0+
03	0-++0-	0D	+0-+-0
04	-+0+0-	0E	+0-0+-
05	+0--+0	0F	+0- -0+
06	+0-0-+	10	0--+0+
07	+0-+0-	11	-0-0++
08	-+00+-	12	-0-+0+
09	0-++-0	13	-0-++0

Dado que existen palabras con peso $(+1)$, si se envía una secuencia de estas palabras, el valor medio de la señal tiende a moverse de cero, causando errores en la recepción. Para evitar esta situación, siempre que aparezca una secuencia de este tipo, el estándar establece que los símbolos se deben invertir de forma alternada, de manera de asegurar valor medio nulo. Por ejemplo, la secuencia: $(0 + + + - -), (0 + + + - -), (0 + + + - -), (0 + + + - -)$, se transmitiría como $(0 + + + - -), (0 - - - ++), (0 + + + - -), (0 - - - ++)$ En el receptor debe aplicarse el mismo procedimiento para poder decodificar correctamente la secuencia.

Es posible describir esta codificación mediante un diagrama de estados, tal como se presenta en la Fig. 6.2. En este diagrama, W representa el peso de una palabra a transmitir y Sum se refiere a dos posibles estados, según el peso de la palabra transmitida. Se parte del estado $Sum = 0$, generándose un cambio de estado cada vez que el peso de una palabra es $(+1)$. Si la máquina sale del estado

de equilibrio $Sum = 0$, la palabra se transmite tal cual es cuando su peso es positivo o nulo. Si proviene de un estado de desequilibrio de continua, desde $Sum = 1$, la palabra se transmite invertida si su peso no es nulo, de otro modo se transmite sin cambios.

Adicionalmente, para reducir la latencia de decodificación, cada byte se transmite según la secuencia presentada en la Fig. 6.3. Tres bytes de datos provenientes del flujo de $100\,Mbps$, se transmiten codificados de manera consecutiva como tres palabras ternarias, una sobre cada uno de los tres pares de cable. El cuarto byte se transmite codificado sobre el primero de los pares y la sucesión se repite. La transmisión sobre el par se realiza a $25\,Msímbolos/seg$. Como entre pares existe una latencia o retardo, equivalente a la duración de dos símbolos ternarios, la secuencia de símbolos recibidos en cada par se puede decodificar independientemente, teniendo en cuenta la compensación de las líneas.

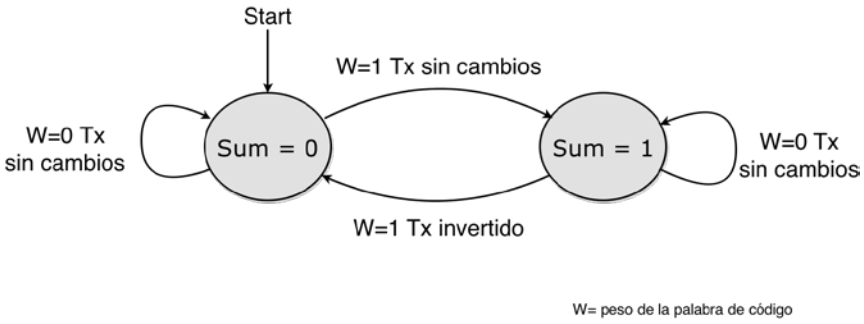


Figura 6.2 - Diagrama de Estados para la codificación final.

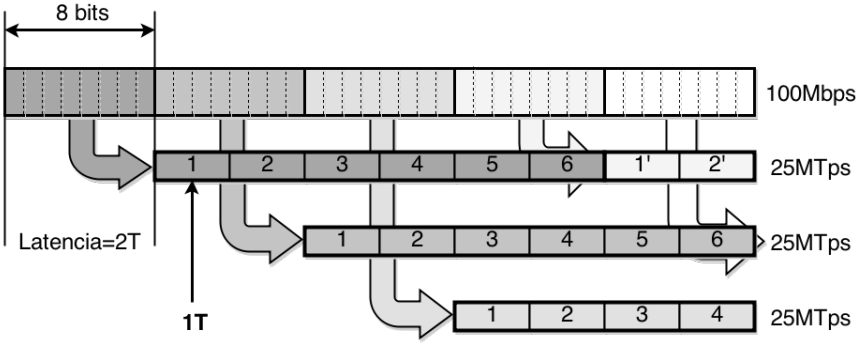


Figura 6.3 - Flujo de 100 Mbps dividido en tres flujos de 25 Msímbolos/seg.

Como se mencionó previamente, mientras el sistema transmite, se detecta colisión por la presencia de señal en el par que corresponda. Como la señal transmitida tiene un nivel muy alto, podría inducir señal no deseada en el par de

detección de colisión. Esta interferencia, que se denomina NEXT, puede ser interpretada por el dispositivo transmisor como señal recibida y, por lo tanto, como una colisión. Para minimizar el efecto de NEXT, el preámbulo en cada trama se codifica en dos niveles $(+1)$ y (-1), en lugar de tres niveles como es el caso de los datos. Esto permite separar las variaciones en amplitud en el extremo receptor, discriminando más fácilmente entre la transmisión de una trama y la posible presencia de interferencia.

El preámbulo es un patrón fijo que viaja en cada par. El Comienzo de Flujo (SOS, Start of Stream) se compone, a su vez, de dos palabras de dos niveles: SOS-1 y Delimitador de Comienzo de Trama (SFD, Start Frame Delimiter). Como estas palabras se transmiten sobre los tres pares, el comienzo de una trama requiere la detección de tres palabras SFD, y la forma en que son enviadas, asegura que se deberían producir al menos cuatro errores en los símbolos para no detectar el comienzo de la trama.

La codificación con memoria que posee 8B6T permite una capacidad extra para detección de errores. Se ha observado que el progreso de la transmisión siempre estará en uno de dos posibles estados Sum. Al final de la transmisión de una trama, luego del CRC, se transmiten sobre cada par una palabra diferente según el par y según la condición en la que se encuentre la máquina de estado de la codificación. Estas palabras, conocidas como Fin del Flujo (EOS, End of Stream), se presentan en la Tabla 6.2.

Tabla 6.2 – Códigos EOS basado en el estado de cada par

	Sum = 0	Sum = 1
E1 (Par 4)	------	++++++
E2 (Par 1)	----	++++
E3 (Par 3)	--	++

La diferencia en la longitud de cada palabra EOS, obedece a la compensación final para compensar la latencia, logrando así que las tres señales arriben al mismo tiempo. Esto asegura una detección confiable del final de una trama, permitiéndose pequeñas variaciones del retardo por cada par.

Aunque 100 BASE T4 fue diseñado para funcionar sobre el mismo cable UTP que 10 BASE T, tecnología que en aquel momento dominaba el mercado, no pudo imponerse comercialmente, probablemente porque las nuevas tecnologías de $100\ Mbps$ proponían la utilización de un cable de categoría superior que era menos vulnerable a las interferencias.

6.2 100 BASE T2

La idea detrás de este estándar fue lograr la transmisión de 100 $Mbps$ *full duplex* sobre dos pares del cable Categoría 3. Esto significaba todo un desafío, debido a la atenuación y el alto nivel de interferencia propios del medio. Lo cierto es que, para la época de su aparición, la tecnología ya había desarrollado herramientas de procesamiento de señales digitales lo suficientemente avanzadas.

La transmisión 100 BASE T2 usa dos pares para transmitir y recibir simultáneamente sobre ambos. Es decir que la principal diferencia de este estándar es que se aparta del modo de funcionamiento de los previos, pasando a modo *full duplex*.

La transmisión se realiza a razón de 4 bits por símbolo, agrupamiento conocido como *nibble*. Primero, los grupos de 4 bits se expanden a 2 símbolos de 3 bits a través de un procedimiento bastante complejo de mezclado o *scrambling* que se realiza mediante un Registro de Desplazamiento de Realimentación Lineal (LFSR, Lineal Feedback Shift Register). El objetivo es desparramar el espectro de emisión de la señal en el cable. De este modo, el mapeo de la codificación no es una función constante en el tiempo, debido al mezclador, por lo que la señal así obtenida es una secuencia pseudo-aleatoria. Esta señal parece ruido y, de este modo, la potencia radiada que resultaría en interferencia en el otro par, no estaría correlacionada con los datos que se transmiten. Se dice que existe una de-correlación espacial que colabora en la recepción para distinguir entre señal deseada y ruido, disminuyendo el efecto del NEXT.

La Fig. 6.4 presenta un sencillo mezclador tipo LFSR de 3 elementos que genera un código pseudo-aleatorio de 7 bits: 0010111. Se puede comprobar esta secuencia, cargando el registro de tres bits con el estado inicial $(b_1 b_2 b_3) = 111$ y siguiendo su evolución, como se presenta en la Tabla 6.3.

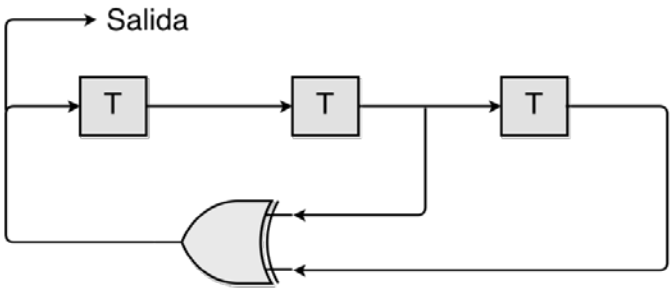


Figura 6.4 - Scrambler LFSR de 3 elementos.

Tabla 6.3 – Evolución LFSR de la Fig. 6.4.

$(b_1 b_2 b_3)_n$	$(b_1 b_2 b_3)_{n+1}$	Salida
111	011	0

011	001	0
001	100	1
100	010	0
010	101	1
101	110	1
110	111	1

A continuación del *scrambling*, se realiza un mapeo de los símbolos a una constelación tipo PAM-5. De este modo, durante cada intervalo de modulación un grupo de 4 bits o *nibble* se codifica en un par de señales *quinarias* que se transmiten simultáneamente sobre los dos pares en ambas direcciones. La codificación mencionada se conoce como 4B2Q, su propósito es controlar errores, reducir la velocidad de señalización para adaptarla al ancho de banda del cable y proveer palabras extra para señalización de control, el mismo propósito que buscaba la codificación 8B6T de 100 BASE T4.

El mapeo PAM-5 de los símbolos a determinadas formas de onda tiene como objetivo el conformado del espectro para el ancho de banda disponible, la reducción de radiación y la provisión de suficientes transiciones para sincronismo, preservándose a la vez el balance de continua. De todas maneras, en los receptores es necesario agregar filtros digitales para cancelación de eco, cancelación de NEXT, ecualización y supresión de interferencia.

Al igual que 100 BASE T4, 100 BASE T2 no llegó a desarrollarse comercialmente, pero la tecnología propuesta para codificación y procesamiento de las señales sentó las bases para 1000 BASE T, cuyos detalles se abordarán más adelante.

5.3 100 BASE TX

100 BASE TX utiliza los mismos pares que 10 BASE T en transmisión y recepción, es decir 2 pares, pero la diferencia es que el cable debe ser STP Categoría 5 o superior. Es el protocolo de $100\ Mbps$ que se impuso en todas las instalaciones nuevas, posteriores a 10 BASE T. La distancia máxima sigue siendo $100\ m$ y su funcionamiento es inherentemente *full duplex*. En realidad, 100 BASE TX puede funcionar en modo *half duplex* o en modo *full duplex*. Se incorpora el modo de funcionamiento *half duplex* por compatibilidad hacia atrás con dispositivos 10 BASE T. El modo de funcionamiento *full duplex* es opcional y permite la comunicación de doble vía simultánea en ambos pares del cable, pero solamente entre dos estaciones, lográndose de este modo una velocidad efectiva de $200\ Mbps$.

En el modo *half duplex* el elemento concentrador *hub* de la topología estrella se comporta como mero repetidor, pero en el modo *full duplex* se precisa un elemento con mayor capacidad, capaz de entenderse con otro elemento a nivel MAC: un *switch*. El *switch* puede ofrece un enlace punto a punto de par trenzado

por cada puerto en que tenga conectada una estación de la LAN. Se habla de redes micro-segmentadas.

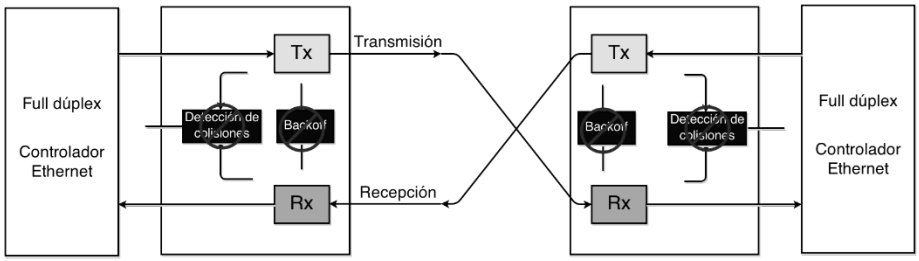


Figura 6.5 - 100 BASE TX en modo full duplex.

Al trabajar en este modo, aparecen algunas ventajas. La modalidad de comunicación punto a punto que permite un *switch*, hace innecesario el método de acceso al medio CSMA/CD característico de la forma de transmisión *half duplex*. De este modo, deja de tener sentido un tamaño máximo de segmento, porque ya no existe la limitación de tiempo requerida para la detección de colisiones mientras se transmite una trama. Tampoco se precisaría un mecanismo de *back off*, tal como se indica en la Fig. 6.5. Sin embargo, en el modo *full duplex* se debe respetar el tiempo entre transmisiones de tramas sucesivas para asegurar que las interfaces en los extremos de cada enlace puedan mantener la velocidad máxima de tramas.

A nivel de capa física, la codificación es doble: 4B5B y MLT-3. En la codificación 4B5B, presentada en la Tabla 6.4, se mapean bloques de 4 bits de datos o *nibbles*, a palabras de 5 bits. De este modo, existen 2^5 posibles palabras de código y 2^4 *nibbles*, por lo que sobran 16 símbolos.

Las palabras que sobran se disponen para ser usadas como códigos de control. Entre estas palabras de control se reservan:

- **IDLE** (11111), que se transmite cuando no hay datos a transmitir, para no perder el sincronismo entre ambos extremos.
- **J/K** (11000/10001), son las señales de comienzo de trama que se envían para indicar que termina la señal de IDLE y comienzan los datos.
- **T/R** (01101/00111), son las señales de fin de trama que se envían para indicar que terminan los datos y comienza la señal de IDLE.
- **H** (00100), es la señal de error usada para forzar errores de señalización.

Tabla 6.4 - Codificación 4B5B

Datos	Símbolo 5 bits	
0000	11110	0
0001	01001	1
0010	10100	2
0011	10101	3
0100	01010	4
0101	01011	5
0110	01110	6
0111	01111	7
1000	10010	8
1001	10011	9
1010	10110	A
1011	10111	B
1100	11010	C
1101	11011	D
1110	11100	E
1111	11101	F

Control	Símbolo 5 bits
IDLE	11111
J	11000
K	10001
T	01101
R	00111
S	11001
QUIET	00000
HALT	00100

Por razones de sincronismo, las palabras que pertenecen al código, se eligen entre aquellas palabras que tienen al menos una transición cada dos bits. Obviamente, la recepción de cualquier otra palabra que no sea de datos o de control, es indicación de error.

Al codificar de este modo, se aumenta la velocidad de señalización en $5/4$, llegando a $125Mbps$. Es decir que se precisa otro esquema de codificación que permita acomodar la velocidad de señalización al ancho de banda del cable Categoría 5, de alrededor de $100\ MHz$. Con este propósito, se agrega un esquema de codificación, conocido como MLT-3, que es similar a la codificación Sin Retorno a Cero Invertido (NRZI, Non Return to Zero Inverted) en cuanto a que posee sus mismas ventajas, pero usa tres niveles y, por lo tanto, menor ancho de banda.

La Fig. 6.6 presenta una comparación entre la codificación NRZI y MLT-3. La codificación NRZI es una codificación binaria, en la que un bit de datos en "1" se transmite como una transición en el comienzo de la duración del bit. Si el bit de datos es "0", no hay transición. La desventaja que posee esta codificación es la dificultad de recuperación de la señal para sincronismo en condiciones de cadenas largas de bits "0".

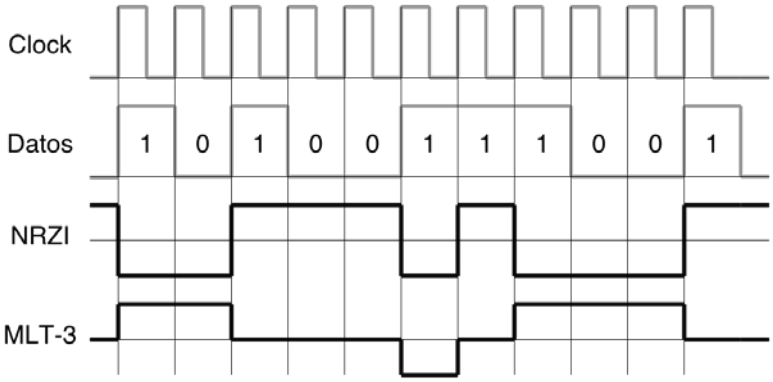


Figura 6.6 - Codificación NRZI y MLT-3

MLT-3 es un tipo de codificación en donde la señal va pasando de manera cíclica por los niveles $(-1), (0), (+1)$, y (0). Se mueve al siguiente estado cuando se transmite un "1" y permanece en el mismo estado al transmitirse un "0". Se podría describir el funcionamiento de la codificación MLT-3, mediante un Diagrama de Estados, tal como presenta la Fig. 6.7.

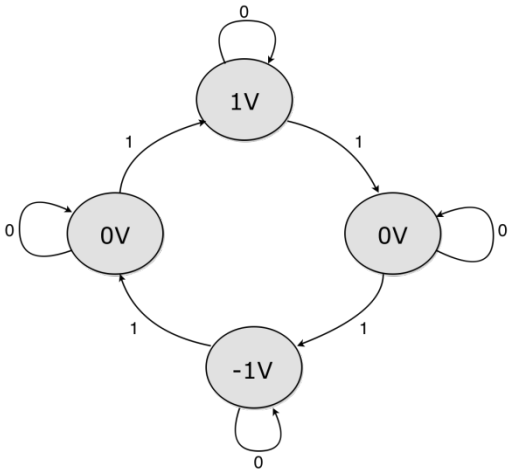


Figura 6.7 - Diagrama de Estados MLT-3

Como se observa, MLT-3 es una codificación tipo senoidal de parada y continuar (*stop&go sine wave*), donde "0" se codifica como *stop* y "1" como *go*, pasando por los tres niveles. Así, se requieren 4 transiciones para completar un ciclo: "$(-V)(0)(+V)(0)(-V)$". Es decir que la frecuencia fundamental se reduce en $1/4$.

A modo de ejemplo, si se deseara transmitir un byte de información ¨0*e*¨, es decir ¨0000 1110¨, la codificación 4B5B lo transformaría en ¨11110 11100¨,

según se indica en la Tabla 6.4. A continuación, la codificación MLT-3 convertiría esta palabra en la secuencia "$(\ 0)(+V)(\ 0)(-V)(-V)(\ 0)(+V)(\ 0)(\ 0)(\ 0)$".

Por otra parte, MLT-3 ofrece una capacidad de detección de errores extra, puesto que cualquier palabra es una secuencia de "$(-V)(\ 0)(+V)$", en ese orden. De este modo, se podrían detectar fácilmente errores en el caso de patrones inválidos y reemplazarlos por un símbolo especial de trama corrupta para evitar su procesamiento.

Es interesante destacar que la codificación MLT-3 de la señal de Control IDLE, "11111", genera una secuencia "$(+V)\ (\ 0)\ (-V)\ (\ 0)\ (+V)$", que reduce la velocidad original a $r = \frac{125Mbps}{4} = 31.25MHz$. Esta señal produce una irradiación muy fuerte, mayor que los niveles permitidos para la aprobación de la Comisión Federal de Comunicaciones (FCC, Federal Communications Commission). Por ello es necesario distribuir la potencia irradiada mediante *scramblers*, similares a los descriptos en el apartado de 100 BASE T2.

Por último, se destaca que el modo opcional de funcionamiento *full duplex* quedó especificado en un suplemento del estándar original, conocido como IEEE 802.3x. Allí se describe un conjunto de mecanismos que permiten realizar control de flujo sobre los enlaces en modo *full duplex*, conocidos como MAC Control y PAUSE, sobre los que nos explayaremos más adelante. Ambos extremos del enlace trabajando en modo *full duplex* deben ser configurados correctamente, para lo cual el estándar recomienda la utilización de un mecanismo de Auto Negociación sobre el que también nos detendremos más adelante.

6.4 Puente

Los puentes o *bridges*, son dispositivos de comunicaciones que operan principalmente a nivel de enlace del Modelo OSI y sirven para conectar no más de dos o tres redes entre sí. Estuvieron disponibles comercialmente a partir de principio de los años 80, presentando grandes diferencias respecto de los *hubs*, que son simples dispositivos repetidores.

Un puente sirve para conectar dos redes, ya sea del mismo tipo o de diferente tipo, permitiendo el re-envío de tramas entre las mismas. La tecnología de los años 80 presentó comercialmente dos clases de puentes: transparentes y de ruteo de fuente. Los primeros se encontraban principalmente en entornos *Ethernet*, mientras que los puentes de ruteo de fuente se diseñaron para entornos *Token Ring*. En este último caso, todo el camino hacia el destino se encontraba explicitado en la trama. En cambio, en entornos con puentes transparentes, la trama se re-envía hacia el destino siguiendo la modalidad de un salto por vez.

Al trabajar en el nivel 2 del Modelo OSI, un puente puede cumplir funcionalidades relacionadas con el control de flujo, el control de errores, el esquema de direccionamiento físico y el método de acceso al medio. El principio de funcionamiento exige analizar las tramas entrantes, efectuar decisiones de

encaminamiento basadas en la información obtenida de las mismas (a nivel capa de enlace), para luego re-enviarlas a su destino.

Aparte de las funcionalidades expuestas, estos dispositivos son capaces de adicionar otras muy importantes, tales como el filtrado de direcciones MAC. Por ejemplo, se podría configurar un puente para rechazar (no re-enviar) tramas según reglas basadas en determinados campos. Así, un puente puede rechazar tramas provenientes de una red determinada, según el valor del campo Tipo de la trama, o filtrar tramas de *broadcast* o de *multicast*. Al re-enviar sólo cierto porcentaje de tráfico, se logra disminuir la carga en los segmentos conectados.

Evidentemente, al realizar esta funcionalidad de filtrado, un puente se presenta como un dispositivo eficaz en cuanto a la separación de dominios de colisión. Al segmentar dominios, los puentes permiten extender la longitud de una red LAN, conectando estaciones alejadas que antes no podrían acceder.

Físicamente, se trata de dispositivos de pocos puertos de Entrada/Salida (I/O, Input/Output), cuya versión transparente fue la base de desarrollo de los *switches*. La denominación de transparencia se debe al hecho de que su presencia y operación no implica carga de procesamiento para las estaciones. Cuando un puente transparente se enciende, puede aprender la localización de las máquinas pertenecientes a cada una de las redes que conecta, analizando las direcciones fuente de las tramas entrantes. Por ejemplo, si un puente levanta una trama proveniente de la máquina A sobre el puerto 1, entiende que la máquina A es alcanzable a través del segmento conectado al puerto 1. De esta manera, construye una tabla mediante un proceso conocido como auto-aprendizaje o *self-learning*. Posteriormente, el puente utiliza dicha tabla como base de consulta para el re-envío de tramas.

Cuando el puente recibe una trama en una de sus interfaces, busca la dirección destino en la tabla. Si dicha dirección se encuentra en la tabla, la trama se re-envía por el puerto que indica la misma. Si no se encuentra, la trama se re-envía por todos los puertos del puente, excepto el de entrada de la trama. Esta fase de inundación es típica durante el proceso de *self-learning*. De todas maneras, las tramas de *broadcast* y *multicast* también se re-envían por inundación.

La actualización de la tabla por cambios en la red se realiza por medio de un temporizador de inactividad, para evitar que la misma crezca indefinidamente. Si durante ese intervalo no se reciben tramas de una estación, se elimina la entrada correspondiente de la tabla. Si se recibe una trama de una estación que hubiera sido eliminada, el puente pasa nuevamente por el proceso de *self-learning*.

De esta manera, se logra aislar el tráfico entre segmentos, reduciendo la carga sobre cada segmento individual, produciendo un efecto de filtrado que ocurre cuando las direcciones MAC fuente y MAC destino se encuentran sobre el mismo puerto o red LAN. El filtrado generalmente mejora los tiempos de respuesta de la red para el usuario, según el volumen de tráfico entre segmentos y la velocidad de generación de tramas de *broadcast* y *multicast.*

6.5 Switch

Un *switch* es un dispositivo de conexión que trabaja como un *bridge* a nivel de capa de enlace, permitiendo el intercambio simultáneo de tramas entre un gran número de estaciones conectadas, una por cada uno de sus puertos de I/O.

Conceptualmente es similar a un conjunto de puentes funcionando en paralelo con un *bus* interno común. Como en el caso del puente, el *switch* inspecciona cada trama entrante, lee su dirección destino y determina el puerto de salida apropiado consultando en una tabla interna. Si dicho puerto se encuentra disponible, la trama se re-envía de inmediato. Esta aproximación se conoce con el nombre *cut-through* y permitió reducir la latencia inherente a la mayoría de las arquitecturas de puente, que son del tipo *store&forward*. Si el puerto de salida no está disponible, el *switch* almacena la trama y la re-envía cuando éste se libera. De esta manera, la forma de trabajo se adapta entre conmutación *cut-through* y conmutación *store&forward*, de manera automática, en una base por trama.

Con fines comparativos, es importante destacar que un *router* es un dispositivo que tiene funcionalidades a nivel 3 del Modelo OSI. El *router* crea y mantiene Tablas de Enrutamiento estática o dinámicamente y toma decisiones de re-envío basándose en una dirección de red destino y en el contenido de la Tabla. En este sentido, comparado con un *router*, el *switch* es un dispositivo mucho más rápido en cuanto a la conmutación pues trabaja en un nivel más bajo del modelo OSI.

De todos modos, mientras que un *switch* es un dispositivo de baja latencia, gran eficiencia y fácil de administrar o configurar, el *router* precisa mucho más control administrativo, ofrece mejor aislamiento de tráfico y puede tener capacidades relacionadas con la protección de la red, por ejemplo la posibilidad de configurar un *firewall* por seguridad o un servicio para Traducción de Direcciones de Red (NAT, Network Address Translation).

Al igual que un *bridge*, el *switch* aprende de manera dinámica cuáles dispositivos de la LAN se encuentran en cada puerto, identificándolos por su dirección MAC. La importancia de un aprendizaje dinámico se traduce en la posibilidad de que los dispositivos puedan cambiar de puerto o cambiar el hardware de interfaz. En este sentido, los cambios de topología se puede interpretar como cambios o movimientos entre puertos, ya que en este tipo de entornos existe una relación única entre dirección *unicast* propia de la interfaz de un dispositivo y el puerto del *switch* al cual el mismo se encuentra conectado.

Tanto el filtrado de tramas como la regeneración de las tramas re-enviadas permite que un *switch* divida la red en diferentes dominios de colisión, pudiendo de este modo extender la misma a mayor distancia, agregar nuevos nodos y bajar considerablemente la cantidad de colisiones. Justamente, se trata de dispositivo diseñado para resolver problemas de rendimiento en la red, permitiendo agregar mayor ancho de banda, acelerar la salida de tramas, reducir tiempos de espera y bajar el costo por puerto. Al segmentar la red en pequeños dominios de colisión, uno por cada puerto, elimina la competencia por el medio.

Muchas veces, en las redes LAN con topología estrella con un *switch* como elemento central, se habla de de un entorno micro-segmentado. Se denomina micro-segmentación a la situación de tener un único dispositivo por cada puerto de I/O, al que se le puede ofrecer la capacidad completa de la LAN

para su propia utilización, evitándose una de las causas primarias de congestión en este tipo de redes: la contienda por el medio. Como contrapartida, la carga de congestión se traslada al propio *switch*.

En un entorno micro-segmentado, sólo un dispositivo en cualquier momento puede encontrarse intentando usar el cable, dado que existe una única estación conectada a cada puerto del *switch*. Sólo esa estación transmite al *switch* mediante el par de transmisión y únicamente el *switch* le contesta mediante el par de recepción. No existe contienda por el medio y, por lo tanto, no se precisa ni detectar colisiones, ni ejecutar el algoritmo de retroceso. Se trata del modo de operación *full duplex*, donde se ignora cualquier detección del medio, previo a la transmisión, o de colisiones, luego de haber accedido al medio.

Para comprender mejor los alcances de la tecnología de un *switch*, se analizan a continuación las características técnicas de un *switch Nway Fast Ethernet*:

- **24 port 10/100BaseT/TX (Auto Negociación)**: se trata de un *switch* de 24 puertos de I/O, capaz de conectar la misma cantidad de dispositivos, trabajando ya sea en *10 BASE T, 100 BASE T* ó *100 BASE TX*. La adaptación es automática por poseer capacidades de Auto Negociación. Por cada puerto del *switch*, éste negociará con el dispositivo conectado en el otro extremo del cable, la velocidad y el modo de transmisión.
- **Auto detección Full/Half-dúplex en todos los puertos**: esta capacidad también es parte del proceso de Auto Negociación, que se produce en el inicio, cuando el dispositivo conectado a un puerto se enciende.
- **Todos los puertos soportan la función Auto MDIX**: se refiere a la posibilidad de cruce interno automático en todos los puertos, para poder hacer posible que los pines de transmisión se conecten con sus pares de recepción y viceversa, sin necesidad de cruzar los cables.
- **Soporte de hasta 8K MAC de filtrado de direcciones y funcionalidad de envejecimiento**: se refiere a la capacidad de memoria para la tabla de direcciones MAC. También, a cada línea de la tabla se le asocia un tiempo de vida, pudiendo eliminarse entradas antiguas de manera automática. Si la dirección MAC por la cual se creó la línea permanece inactiva por ese tiempo, la línea se borra.
- **Memoria de buffer 2.5 Mbits**: Capacidad de memoria para tramas entrantes.
- **Re-envío no bloqueante a la velocidad del cable y filtrado**: se refiere a que la capacidad del bus interno es la suma de las velocidades de todos los puertos.
- **Arquitectura Store&Forward y prevención de bloqueo de cabeza de línea**: tanto el tráfico entrante como el saliente se almacenan en colas en buffers. Generalmente, el manejo de estas colas es de tipo FIFO. Cuando las tramas recibidas deben entregarse a un puerto sobrecargado ocurre una situación que se conoce como bloqueo de cabeza de línea. Si las colas se atienden por ronda, cada puerto requiere acceso al *bus* y, si el árbitro central del mismo se lo otorga, ningún otro puerto puede acceder hasta que el puerto receptor haya recibido la trama completa. Si varios intentan

transmitir sobre un mismo puerto de salida, se puede producir una situación de bloqueo. Un manejo de buffers inteligente, apoyado en algoritmos apropiados, con colas para tramas con prioridad, pueden evitar estas situaciones.

- **Prevención de Tormentas de Broadcast**: en algunos *switches*, luego de cierto número de tramas de *broadcast* consecutivas recibidas en un puerto particular, las siguientes tramas del mismo tipo arribadas durante un intervalo de tiempo predefinido, por ejemplo $800\ ms$, se descartan. Cualquier otra trama que no sea de *broadcast* recibida dentro de ese intervalo, fija el contador de protección de tormenta de *broadcast* en cero. En algunos switches, se habilita la prevención cuando el tráfico de *broadcast* excede cierto porcentaje de la carga de la red, por ejemplo 5%.
- **Soporte de control de flujo IEEE 802.3x para FDX y de control de flujo presión trasera para HDX**: significa que el switch tiene capacidad de control de flujo para ambos tipos de funcionamiento: *half duplex* y *full duplex*.
- **Auto aprendizaje, esquemas de re-envío y filtrado**: quiere decir que el *switch* aprende por sí mismo la vinculación de direcciones MAC con números de puerto.
- **Soporte de interfaz SMII/GMII retardo I/O**: soporte de interfaz *Gigabit* y capacidad para acomodar diferencias de retardos.
- **Ancho de banda dedicado full duplex 200 Mbps en cada puerto**: Todos los puertos soportan transmisión *full duplex*.

6.6 Mecanismo de Auto Negociación

Se trata de una tecnología propuesta por la IEEE en el año 1994 debido a la necesidad de un mecanismo que permitiera acomodar, en una misma red, diferentes tipos de dispositivos. Permite que dispositivos que comparten un segmento, se configuren de manera automática en el modo de operación mejor posible. Al incorporarse esta posibilidad en la propia placa de red, no sólo se alivia el trabajo del administrador, sino que también es transparente para el usuario.

La especificación sobre el mecanismo de Auto Negociación se publicó por primera vez en 1995 como parte del suplemento *Fast Ethernet* del estándar IEEE 802.3u. Se basaba en un sistema de configuración conocido como *NWay* que había sido ideado por ingenieros de National Semiconductor con el propósito de que las estaciones *Ethernet* intercambiaran información sobre sus capacidades en el segmento de enlace donde se encontraran.

Las premisas definidas en la especificación incluyen la restricción de que la operación sólo es realizable en un enlace entre dos dispositivos, ocurriendo solamente en el inicio, al momento en que un dispositivo se enciende o se conecta mediante un cable. Así, la negociación es entre dos partes, la máquina y el *switch*. Cada uno advierte sus propias capacidades al otro y el protocolo selecciona un denominador común de funcionamiento. Es importante observar que, con un *hub*

como elemento central, es imposible tener esta posibilidad debido a que el repetidor no posee inteligencia a nivel de capa de enlace.

Para poder llegar a un acuerdo, se intercambia información entre ambos dispositivos por medio de un sistema de señalización especial, denominado Ráfagas de Pulsos Rápidos de Enlace (FLP, Fast Link Pulse Burst). FLP es una versión modificada de las señales de Pulso de Enlace Normal (NLP, Normal Link Pulse) que se usan para verificar la integridad del enlace en 10 BASE T cuando no se transmiten datos.

Cuando los dispositivos 10 BASE T no se encuentran transmitiendo o recibiendo tramas, envían pulsos unipolares de $100\ nseg$ de duración nominal, con un ancho máximo de pulso de $200\ nseg$, que se generan a intervalos de $16\ mseg$, con una tolerancia de variación en el tiempo de $8\ mseg$ ($f_{máx} = 125\ Hz, f_{mín} = 62.5\ Hz$), tal como se muestra en la Fig. 6.8. En términos genéricos, se los conoce como Chequeo de Integridad del Enlace (LIT, Link Integrity Test). Si un dispositivo no recibe ni una trama ni dos de estos pulsos por un intervalo tiempo de entre $50\ y\ 150\ mseg$, detecta falla del enlace.

Los pulsos NLP que se mencionan en la especificación de Auto Negociación son también unipolares y con la misma duración nominal de $100\ nseg$, pero cada pulso LIT es en realidad una ráfaga de entre $17\ a\ 33$ pulsos separados en intervalos de $125\ \mu seg$. Cada pulso de la ráfaga se conoce como FLP. El intervalo de tiempo entre el comienzo de cada ráfaga es de $16\ mseg$, con la misma tolerancia que en el caso NLP. De este modo, cualquier dispositivo 10 BASE T puede interpretar los pulsos como si se tratara de los NLP. Así se provee compatibilidad hacia atrás con equipos 10 BASE T viejos que no reconozcan el mecanismo de Auto Negociación.

Los pulsos FLP codifican palabras de $16\ bits$. Entre cada par de dos pulsos NLP consecutivos, o sea a $62{,}5\mu seg$ del primero de ellos, puede estar presente un pulso adicional positivo que indica la presencia de un "1" lógico cuando está presente y un "0" cuando está ausente. Cada ráfaga contiene una palabra de $16\ bits$ que se conoce como Palabra de Código de Enlace (LCW, Link Code Word). En conjunto, la ráfaga FLP contiene 33 posiciones de pulso. De estos, los 17 pulsos en posición impar representan información de sincronismo, y los 16 en las posiciones pares son los datos LCW. Sus bits se numeran de 0 a 15.

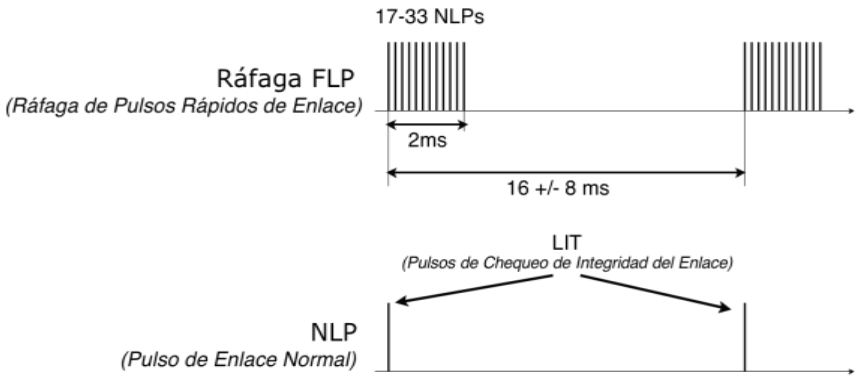


Figura 6.8 - LIT, NLP, FLP.

Cuando el dispositivo se inicializa, el protocolo de Auto Negociación envía tantos mensajes de 16 $bits$ como sean necesarios para completar la negociación. En la Fig. 6.9 se presentan los 16 bits de LCW, etiquetados como D0 a D15. Los bits D0 a D4 constituyen un Campo Selector que identifica el tipo de tecnología LAN. Para *Ethernet*, la representación correspondiente es el primer bit (S0) en "1" y el resto de los bits en "0".

Los 8 bits D5 a D12, identificados también como A0 a A7, se denominan Habilidad de Tecnología y se usan para indicar el soporte físico. Si un dispositivo soporta más de una capa física, enciende el bit que corresponde en "1", tal como presenta la figura.

El bit D13, por su parte, es el Indicador Remoto de Falla y puede ser usado, por ejemplo, en el caso que una de las partes detecte errores en la recepción.

El bit D14 es el de indicación de reconocimiento o ACK de los mensajes recibidos. Su existencia se debe a que los mensajes se retransmiten hasta que se recibe un ACK y este bit se levanta luego de recibir tres mensajes idénticos.

El bit D15 sirve para indicar Página Siguiente, usado en el caso de mensajes especiales que advertirán tecnologías no contempladas en el listado, por ejemplo: comandos específicos de fabricante o 10 Gigabit Ethernet.

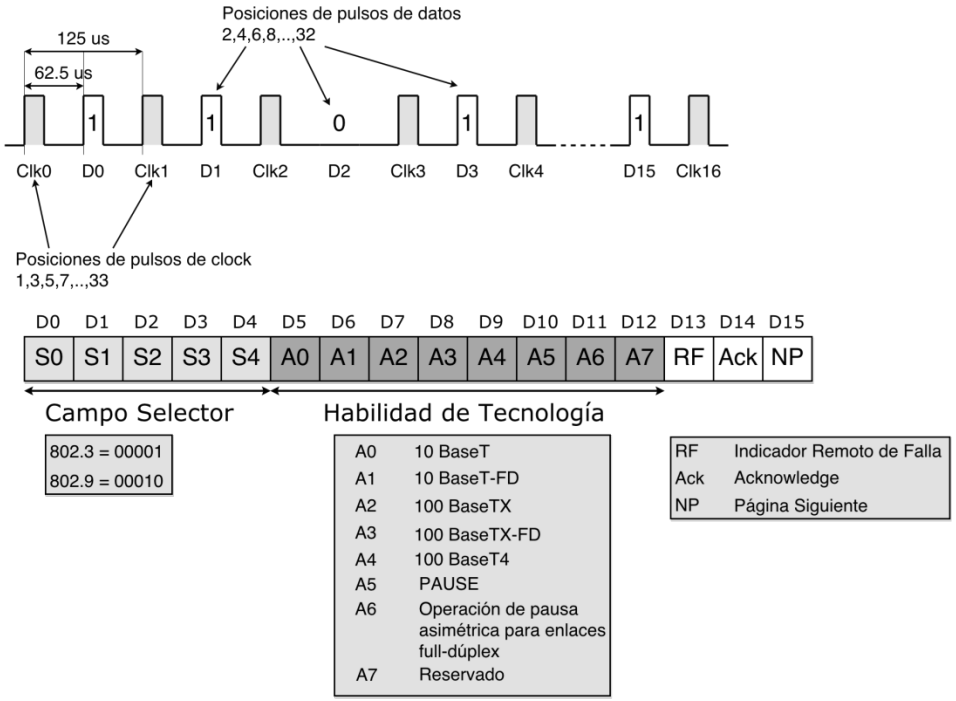


Figura 6.9 - Palabra LCW

Una vez completada la Auto Negociación, no se transmiten más ráfagas FLP, aunque el sistema monitorea el enlace y puede detectar cuando se cae, de tal manera que cuando vuelva a estar activo, se realiza el procedimiento nuevamente.

El protocolo define una serie de prioridades para que el dispositivo seleccione el conjunto común más alto de capacidades, tal como se presenta en la Tabla 6.5.

Tabla 6.5 - Prioridades de Auto Negociación.

Prioridad	**Tecnología**
1	1000 BASE T – FULL DUPLEX
2	1000 BASE T – HALF DUPLEX
3	100 BASE T2 – FULL DUPLEX
4	100 BASE TX – FULL DUPLEX
5	100 BASE T2 – HALF DUPLEX
6	100 BASE T4
7	100 BASE TX – HALF DUPLEX
8	10 BASE T – FULL DUPLEX
9	10 BASE T – HALF DUPLEX

El soporte de Auto Negociación es opcional para la mayoría de los sistemas *Ethernet*, pero la tecnología moderna de *switch* lo tiene incorporado casi sin excepciones.

6.7 Mecanismo de Control de Flujo

Un *switch* es un dispositivo típico de una red LAN que permite conectar computadoras en topología estrella. Se trata de un elemento con mayor inteligencia que el *hub*. Un *hub* es un mero repetidor que se puede modelar simplemente como un dispositivo de capa física, que no posee funcionalidad en otro nivel, puesto que su uso se restringe a entornos de redes CSMA/CD de baja velocidad, en modo *half duplex*. Un *switch*, por su parte, en un dispositivo más avanzado, ya que posee funcionalidad a nivel de capa de enlace. Esta inteligencia adicional le permite enviar tramas sólo a los dispositivos cuya dirección MAC destino reconoce.

Como se ha explicado, un *switch* posee una estructura interna, denominada genéricamente *bus*, con capacidad para manejar el flujo de tráfico para el que se supone estar diseñado. Asociado con cada puerto de Entrada/Salida (I/O, Input/Output) también posee *buffers*. A pesar de esta estructura, en enlaces especialmente cargados como los de *full duplex*, podría suceder una situación de congestión. Un mecanismo de control de flujo puede colaborar en el manejo de estas situaciones.

Se han desarrollado varios mecanismos de control de flujo del tipo propietarios para segmentos *half duplex* que incluyen el uso de ráfagas cortas de portadora que el *switch* envía cuando los *buffers* de sus puertos se llenan, para que las estaciones en el segmento *half duplex* dejen de transmitir datos. Esto no funciona en enlaces *full duplex*, donde el algoritmo CSMA/CD no es de aplicación, ya que se ignora la detección de portadora.

En el suplemento Full Duplex 802.3x se explicitan las especificaciones opcionales Control de MAC y PAUSE.

Control MAC del suplemento 802.3x provee un mecanismo para manipulación y control en tiempo real de la transmisión y recepción de tramas. No se usa para funciones en tiempo no real, como por ejemplo en la configuración de interfaces.

Las tramas de Control MAC se identifican con un valor especial en el campo Tipo: $"0x8808"$. Estas tramas pueden perderse, dañarse, retardarse o descartarse, dado que *Ethernet* no posee un mecanismo de transporte confiable. Como se puede apreciar en la Fig. 6.10, el campo de datos es de tamaño fijo, igual al mínimo de una trama *Ethernet*, 46 bytes. Los primeros dos bytes del mensaje son el campo de Código de Operaciones, y los dos bytes siguientes constituyen el campo de Tiempo de Pausa.

El sistema PAUSE de control de flujo para enlaces *full duplex* utiliza estas tramas especiales con el valor del campo Código de Operaciones en $"0x0001"$. La dirección destino de la trama es una dirección especial *multicast*, $01:80:c2:00:00:01$, reservada por el estándar IEEE 802.1D para *switches*, para

simplificar el proceso de control de flujo, ya que no es necesario pasar por ningún mecanismo de descubrimiento de direcciones entre los dos extremos del enlace. En la operación normal del *switch*, las tramas con dirección destino *broadcast* o *multicast* son retransmitidas por todos los puertos, excepto en el caso de esta dirección especial, que el *switch* interpreta como propia.

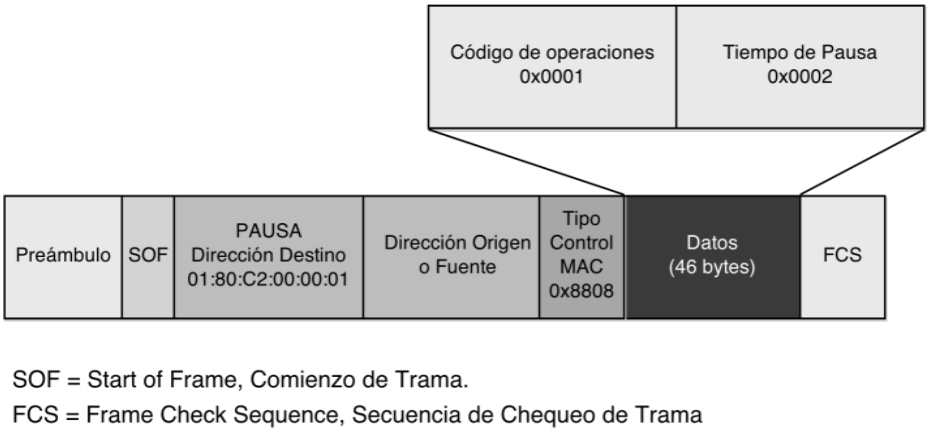


Figura 6.10 - Control de Flujo, tramas PAUSE

En el campo de Tiempo de Pausa, el transmisor escribe el tiempo de parada que se requiere al receptor. Este tiempo se mide en unidades especiales, denominadas *quanta*, de 512 bits de duración, interpretados como $5.12\ \mu seg$ a $100\ Mbps$. Si una trama PAUSE ha anunciado un período de tiempo de parada pero, antes de que el mismo expire, llega otra trama con otro valor para ese tiempo, este parámetro reemplaza al anterior. De este modo se puede restablecer una comunicación inmediatamente enviando una trama con dicho parámetro en "0" . El valor más alto establece una espera de $336\ mseg$.

Con este mecanismo se logran aliviar muchos problemas de congestión que se trasladan al *switch* trabajando en modo *full duplex*.

6.8 Gigabit Ethernet

Gigabit Ethernet es una ampliación del estándar *Ethernet* para poder transmitir a la velocidad de 1 $Gbps$ Una de las implementaciones de *Gigabit Ethernet* es 1000 BASE TX, ideado para uso sobre 2 pares del cable Categoría 6. Como contrapartida, 1000 BASE T puede transmitir sobre cables de Categoría 5 de amplia base ya instalada, pero utilizando los 4 pares.

1000 BASE T fue explicitado en la revisión IEEE 802.3ab y aprobado en 1999. Apoyó su tecnología en detalles técnicos de estándares previos. 100 BASE T4 le aportó técnicas de codificación redundante para señales de control y la posibilidad de transmitir y recibir señales multinivel sobre cuatro pares. De 100

BASE T2 se tomaron todas las técnicas de Procesamiento Digital de Señales (DSP, Digital Signal Proccessing), codificación multinivel, transmisión *full duplex*, cancelación de eco y NEXT que dicho estándar proponía.

En la Fig. 6.11 se presenta un esquema de la circuitería de capa física. Comparada con una transmisión *half duplex*, la transmisión *full duplex* sobre los cuatro pares minimiza la velocidad de señalización a la mitad sobre cada par. El costo de esta reducción de ancho de banda exige el uso de circuitos especiales, denominados circuitos híbridos, cuya tarea es separar la señal transmitida en el receptor. Aunque estos circuitos tengan muy buena relación de minimización de la señal transmitida acoplada en la recepción, siempre existen residuos que deben ser anulados mediante circuitos eliminadores de eco.

El estándar propone transmitir sobre cada par una señal de $250\ Mbps$ en ambas direcciones, logrando llegar a la velocidad de $1\ Gbps$ debido a la transmisión simultánea sobre los cuatro pares. Debido a las limitaciones de ancho de banda del cable de Categoría 5, la transmisión sobre cada par se realiza a $125\ Msímbolos/seg$, aunque en vez de usar cuatro niveles de señal para reducir la velocidad a la mitad, se utilizan cinco niveles, reservando un nivel para señalización extra. Este esquema de codificación se conoce como PAM-5. También se utilizan técnicas de conformación de pulso para adaptar el espectro de la señal al canal, mientras que los efectos del ruido y de la interferencia entre pares, se compensan con una codificación para Corrección de Errores hacia Adelante (FEC, Forward Error Correction) muy compleja, de 8 estados y 4 dimensiones: 4D 8 State Trellis FEC. La complejidad justamente se compensa con una transmisión en 4 dimensiones, o sea sobre los cuatro pares.

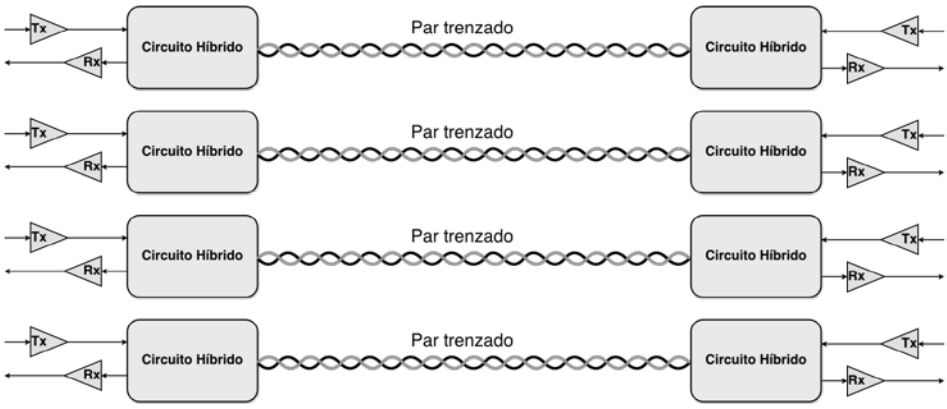


Figura 6.11 - Capa Física Gigabit Ethernet.

Por otro lado, las posibilidades abiertas por las nuevas tecnologías DSP se reflejan en el uso de ecualizadores cuya contribución a la eliminación de interferencias permite llegar a valores de Probabilidad Binaria de Error en el orden de 10^{-10}.

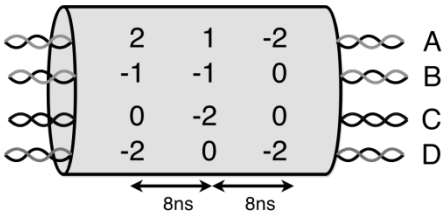
El estándar define una Interfaz Independiente del Medio (GMII, Gigabit Media Independent Interface) entre la subcapa MAC y la capa física propiamente dicha. Los datos que maneja ésta se deben sincronizar a 125 MHz debido al ancho de banda del cable. Para manejar un flujo de información de 1 $Gbps$ durante la transmisión, el nivel físico estaría recibiendo 1 de entre 256 palabras (8 $bits$) cada 8 ns. Por este motivo es ineludible una codificación en varios niveles.

La elección de un esquema de codificación de varios niveles incrementa la complejidad del sistema. Si se eligiera codificar con MLT-3, se enviaría 1 símbolo (de 1 entre 3 niveles posibles) sobre cada par. Esta elección permitiría el transporte de un universo de $3^4 = 81$ palabras posibles. Esta codificación no resultaría suficiente para cubrir 256 palabras provenientes del flujo de información a 1 $Gbps$.

Si la elección incluyera 4 niveles de codificación, se cubrirían las $4^4 = 256$ palabras posibles, pero no sobrarían combinaciones para palabras de control.

Al codificar en 5 niveles $(-2, -1, 0, +1, +2)$, se obtienen $5^4 = 625$ palabras posibles. De este modo, se podría obtener hasta el 100% de redundancia y sobrarían 113 símbolos para señales de control. Por otra parte, si no se usaran los símbolos (-2) y $(+2)$ para transmitir y sólo se usara dos pares, la salida sería como en 100 BASE T. Esta es una de las características que permite implementaciones duales 100/1000 BASE T más sencillas.

De este modo, el esquema de codificación traduce palabras de 8 $bits$ en una trasmisión simultánea de 4 $símbolos$ sobre los 4 $pares$ del cable. En la Fig. 6.12 se presenta un ejemplo de la transmisión de la codificación PAM-5 sobre los cuatro pares y el mapeo de los respectivos niveles en señales de tensión. El flujo original repartido de este modo, se convierte a la velocidad deseada: 125 $Mbaud\ x\ 2\ bits/baud\ x\ 4\ pares\ =\ 1\ Gbps$.



Símbolo	Mapeo del nivel en señal de tensión (volts)
-2	-1
-1	-0,5
0	0
+1	+0,5
+2	+1

Figura 6.12 - 4D PAM-5.

La capa física consiste esencialmente de dos subcapas: la superior de Codificación Física (PCS, Physical Coding Sublayer) y la inferior de Acoplamiento de Acceso al Medio (PMA, Physical Medium Attachment).

El transmisor PCS contiene varios bloques funcionales que convierten el flujo de datos de 8 $bits$ a símbolos PAM-5 para su entrega al PMA. Para reducir la interferencia electromagnética se utilizan codificadores pseudo-aleatorios implementados con registros de desplazamiento y, para mejorar las condiciones de error, los datos se codifican mediante un codificador convolucional. Así, a partir de la información a transmitir, se genera una secuencia pseudo-aleatoria de 8 $bits$ que se entrega a un bloque de mapeo de bits a símbolos quinarios. Con dos bits de esta secuencia, se alimenta el codificador convolucional, para generar un bit adicional, que también se entrega al bloque de conversión.

Este conversor toma 9 $bits$ de entrada y los transforma a símbolos quinarios. Su funcionalidad divide los símbolos en subconjuntos pares e impares, para combatir el ruido y la atenuación. La Fig. 6.13 presenta, a modo de ejemplo, una constelación de dimensión 5×5 de símbolos quinarios para poder observar el efecto de la interferencia y de la atenuación en la detección. Se debe tener en cuenta que la constelación real es de $5 \times 5 \times 5 \times 5$ ya que la transmisión sobre los cuatro pares genera un vector de 4 dimensiones. Cada punto de la constelación representa una palabra de 8 $bits$ o una palabra de 4 símbolos quinarios que, al ser transmitida y sufrir el efecto del ruido y la atenuación, empieza a desdibujarse, acercándose a las vecinas y generando errores de detección. Si se separan los símbolos en grupos de pares e impares, se puede aumentar la distancia entre símbolos adyacentes, reduciendo así la probabilidad de error.

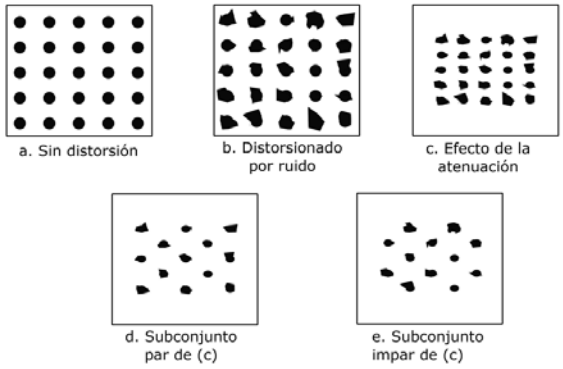


Figura 6.13 - 2D PAM-5, Efectos de Ruido y Atenuación

Por ejemplo, si se piensa en una constelación de dos dimensiones, dividir los símbolos entre pares e impares, aumentaría la distancia efectiva entre ellos al doble. Para entender mejor el concepto, se supone una constelación 2D PAM-5 de $d_{mín} = 1$, como la que se muestra en la Fig. 6.14. Si sobre esta constelación separamos los símbolos en dos subconjuntos, de tal manera que se designara con

X los símbolos con componente $A_n = \{-1, +1\}$, e Y a aquellos símbolos tal que $B_n = \{-2, 0, +2\}$, se puede redibujar una constelación par, tal como se presenta en la Fig. 6.15, lográndose aumentar la distancia a $d_{mín} = \sqrt{2}$. Si en esta constelación se observa sólo el subconjunto de los símbolos YY, se notará que la distancia entre ellos no supera el valor $d_{mín} = 2$, sucediendo lo mismo para el subconjunto XX.

Los símbolos que restan, se pueden ordenar en otra constelación, como se indica la Fig. 6.16, con la misma distancia que la constelación anterior $d_{mín} = \sqrt{2}$. Si de esta constelación impar se separaran los símbolos que comienzan con Y, se obtendría un subconjunto de $d_{mín} = 2$. De manera similar sucede con el subconjunto de símbolos que comienzan con X.

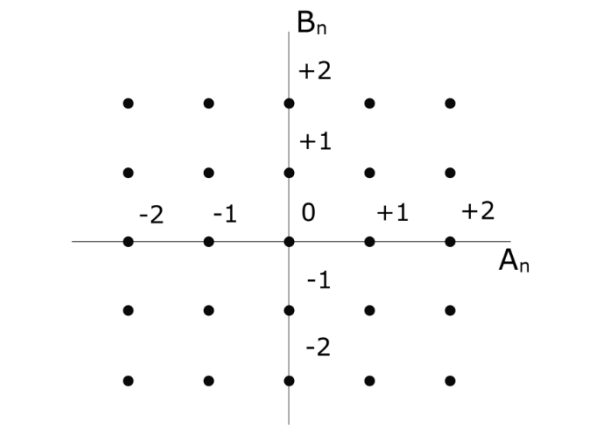


Figura 6.14 - Constelación 2D PAM-5 de $d_{mín} = 1$

En el ejemplo desarrollado, no se debe olvidar que se trata de una constelación *2D*, pero en el caso de *Gigabit Ethernet* esta constelación es de 4 dimensiones, ya que cualquiera de los 4 pares puede estar transportando un símbolo X o Y. Es decir que, en cada transmisión, existen 16 combinaciones posibles trasladándose entre ambos extremos. Este conjunto puede ser reducido a 8 manteniendo una $d_{mín} = 2$ entre los símbolos sobre los 4 pares, de la misma manera que se observó en el ejemplo.

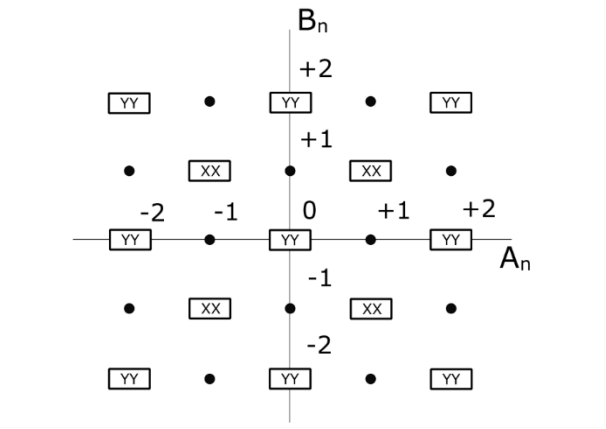


Figura 6.15- Subconjunto constelación par de 2D PAM-5.

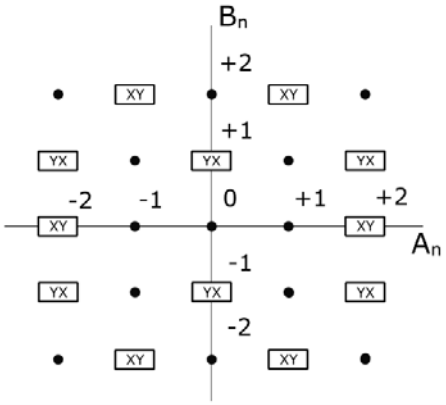


Figura 6.16 - Subconjunto constelación impar de 2D PAM-5

A partir de esta separación, se pueden armar subconjuntos de constelaciones 4D, agrupando los símbolos en ocho conjuntos, de la siguiente manera: $D0 = \{ XXXX, YYYY\}$; $D1 = \{XXXY, YYYX\}$; $D2 = \{XXYY, YYXX\}$; $D3 = \{XXYX, YYXY\}$; $D4 = \{XYYX, YXXY\}$; $D5 = \{XYYY, YXXX\}$; $D6 = \{XYXY\ , YXYX\}$; $D7 = \{XYXX, YXYY\}$.

De este modo, dentro de las sub-constelaciones, la distancia aumenta a $d_{mín} = 2$, sucediendo lo mismo entre dos símbolos de diferentes sub-constelaciones. Por ejemplo, si se considera $D4$ y se codificara $XYYX = (+1, -2, +2, -1)$ y $YXXY = (0, -1, +1, 0)$, se puede calcular la distancia entre ambos símbolos como:

$$d(XYYX - YXXY) = \sqrt{(+1 - 0)^2 + (-2 + 1)^2 + (+2 - 1)^2 + (-1 - 0)^2} = \sqrt{1 + 1 + 1 + 1} = 2 \quad (6.3)$$

Hay símbolos más separados aún, por ejemplo para el conjunto $D0$, $XXXX = (-1, +1, -1, -1)$ y $YYYY = (+2, +2, -2, -2)$, la distancia es:

$$d(XXXX - YYYY) = \sqrt{(-1 - 2)^2 + (+1 - 2)^2 + (-1 + 2)^2 + (-1 + 2)^2} = \sqrt{9 + 1 + 1 + 1} = 3.16 \quad (6.4)$$

Esta forma de definir el mapeo de los símbolos a las sub-constelaciones determina las transiciones del Trellis del esquema de codificación convolucional que se utiliza para la transmisión. Cuando mayor distancia exista entre las secuencias a transmitir, más eficiente será la decodificación en términos de la probabilidad binaria de error P_{be}. El beneficio que la complejidad de este sistema presenta es una ganancia de código efectiva de $6dB$ que compensa lo que se pierde por transmitir más de tres niveles manteniendo el rango de tensión constante en $2\ Volts$.

Una vez codificado en PAM-5, cada palabra se pasa al bloque PMA para su procesamiento. En este bloque se encuentran un ecualizador adaptivo para compensar las distorsiones del canal, un cancelador de eco y de NEXT para manejar interferencias entre pares y transmisión full duplex, un control automático de ganancia para la recepción y un compensador de nivel de continua.

A nivel MAC de *Gigabit Ethernet*, también se permite la forma de operación *half duplex*, aunque se presenta un inconveniente. Dado que el tiempo necesario para transmitir una trama es inversamente proporcional a r_b, a una velocidad de $100\ Mbps$ la trama mínima se transmite en aproximadamente $1/10$ del tiempo de ranura de *Ethernet* 10 BASE T. Es muy difícil detectar colisiones en este caso, pero la solución consiste en reducir a la décima parte el diámetro de la red: de $2500\ m$ a $200\ m$ aproximadamente, distancia máxima para las redes de par trenzado. En el caso de *Gigabit Ethernet* se presenta el mismo problema, pero reducir el diámetro a $20\ m$ sería una solución impráctica. En este caso, se prefirió mantener el diámetro constante pero modificar el tamaño de la trama mínima, llevándolo de 64 a 512 bytes. Aquellas tramas que son más cortas que la longitud mínima, se rellenan con un campo sin datos, de longitud variable, llamado Campo de Extensión, que se remueve en el proceso de recepción, tal como se presenta en la Fig. 6.17.

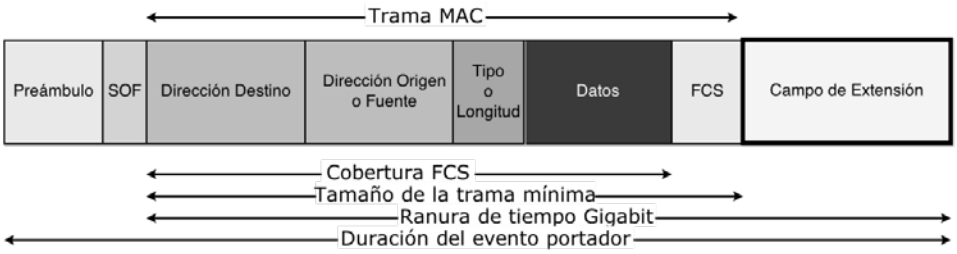


Figura 6.17 - Modificación del tamaño de la trama mínima para modo half duplex.

El campo de extensión soluciona el tema de compatibilidad hacia atrás, pero genera consumo de ancho de banda ya que para paquetes de tamaño mínimo

implica un agregado de 448 $bits$ de relleno. Una extensión de este concepto permite mejorar la eficiencia: la posibilidad de transmitir tramas por ráfagas o *frame bursting*. Cuando un dispositivo tiene varias tramas para transmitir, rellena la primera con el campo de extensión si es necesario, y transmite el resto de las tramas una a continuación de otra, respetando el tiempo mínimo entre tramas IFG hasta llegar al tamaño máximo permitido. De esta manera se aprovecha mejor el mecanismo de extensión. La Fig. 6.18 presenta los campos más relevantes de este tipo de transmisión.

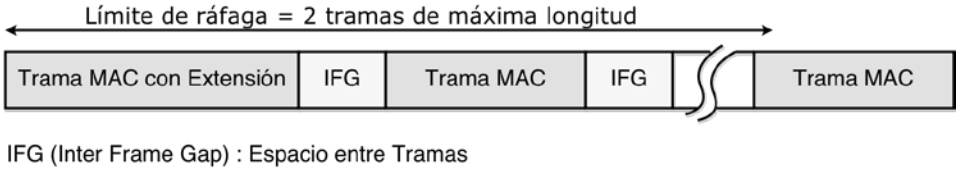


Figura 6.18 - Ráfaga de tramas

En cuanto a la operación *full duplex*, aunque la misma es opcional, es una de las más utilizadas. No implica contienda por el medio, ni colisiones, ni retransmisiones, ni bits de extensión. No sólo duplica el ancho de banda, sino que además mejora el tiempo disponible para transmitir. La única restricción es que debe existir un mínimo espacio entre tramas, IFG. La operación *full duplex* requiere la implementación de la capacidad opcional de Control de Flujo, que permite a un nodo congestionado requerir a su par que cese la transmisión de tramas por cierto período de tiempo, como ya se ha explicado.

6.9 V-LAN

Una LAN virtual, también llamada V-LAN, es una red lógica dentro de una red física. El objetivo es la agrupación de integrantes de equipos de trabajo con los mismos intereses, cuyos datos no se deberían mezclar con otros de la misma red. Un ejemplo podría ser que cada departamento dentro de una empresa sea una red virtual dentro de la red de área local física. Se podría lograr este efecto de separación por medio de *routers*, pero aumentarían considerablemente los costos de instalación.

La posibilidad ofrecida por el tendido de una LAN virtual es la de definir una red LAN de tal manera que sus estaciones de trabajo no dependen de su posición física para poder compartir recursos con otros dispositivos que pertenezcan a la misma LAN. Los equipos de la red virtual se comportan como conectados a un mismo *switch*, aunque, en realidad, pueden estar conectados físicamente a diferentes segmentos de una red de área local, tal como se presenta en la Fig. 6.19. Para poder lograr esto, se precisa la intervención de un administrador en una etapa previa de configuración. Una de las ventajas más

interesantes es que, según como se configure, el traslado de una máquina perteneciente a una red V-LAN dentro de la red física no implica cambiar la configuración de la misma.

Las redes V-LAN son construcciones esencialmente de capa 2. Un *switch* es el dispositivo capaz de crear varias redes V-LAN en una misma red física, proveyendo segmentación a nivel 2. El tráfico por encima de este nivel no puede cruzar los límites de la VLAN, a menos que haya rutas especialmente definidas entre redes virtuales LAN. Por tanto, es una forma más de separar dominios de *broadcast* sin precisar la instalación de un *router*.

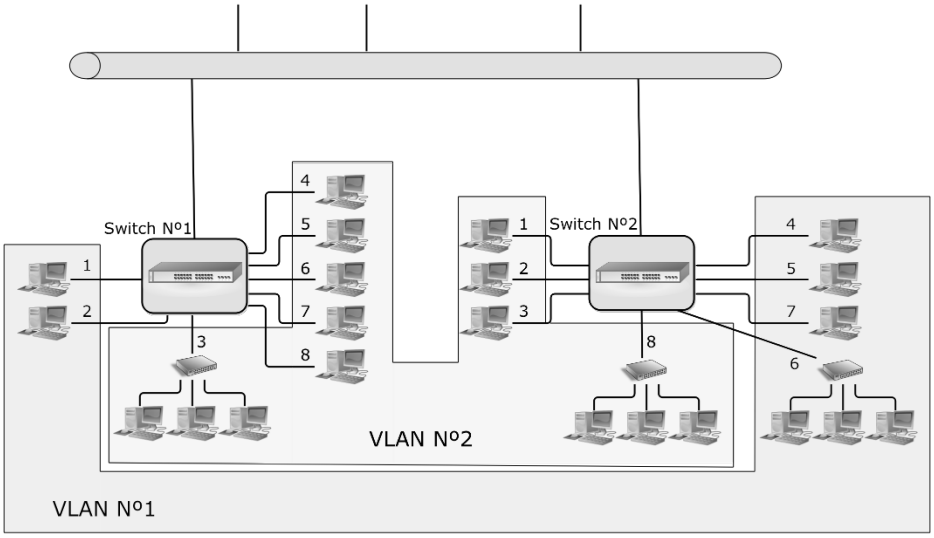


Figura 6.19 - Armado VLAN

En términos generales, las redes V-LAN se pueden clasificar en dos tipos según el nivel de la jerarquía de protocolos en el que operen:

- **V-LAN de nivel 1, por puerto**: también conocida como conmutación por puerto o *port switching*. En este tipo de redes, se especifican los puertos del *switch* que pertenecen a la V-LAN. Los miembros de la V-LAN son los que se conecten a esos puertos. Si el usuario se mueve físicamente, habría que reconfigurar la red virtual. Es decir que, este tipo de V-LAN no permite la movilidad de los usuarios. En este caso no se requiere modificación del encabezado de las tramas. Es el propio *switch* el que mantiene la información apropiada para el re-envío de tramas, de acuerdo al número de puerto por el que ingresan las mismas. A este tipo de configuración se suele acudir por motivos de seguridad, conociéndosela también como red V-LAN estática.
  - **V-LAN de nivel 2, por direcciones MAC**: Los equipos se asignan a una red V-LAN de acuerdo a su dirección MAC. Si el usuario cambia de lugar, moviéndose entre puertos de un mismo *switch* o entre *switches*, no

hay que reconfigurar el dispositivo. En esta clase de V-LAN, cuando se recibe una trama por un puerto, se utiliza la dirección MAC fuente para determinar la asociación. Se conoce esta asociación como red V-LAN dinámica.

El principal inconveniente de este tipo de redes V-LAN es que, a medida que aumenta el número de usuarios, se complica la configuración, recargando la tarea del administrador.

Cualquier *switch* con soporte VLAN establece la LAN virtual por dos tipos de mecanismos: etiquetado de tramas o filtrado. En ambos casos, se examina la trama para poder decidir su re-envío. En el caso de etiquetado, la trama lleva una marca que sirve para la decisión. En el caso de filtrado, existe una tabla en el *switch* y diferentes atributos de la trama son los que permiten la decisión. El problema de este último mecanismo es que es menos escalable que el primero, ya que cada trama necesita ser referenciada a la tabla de filtrado. Por este motivo, se considera el etiquetado de tramas como el mecanismo más eficiente de despliegue.

Cuando un entorno VLAN incluye múltiples switches se usa el método *VLAN trunking* entre ellos. Con *VLAN trunking* los *switches* etiquetan cada trama que envían a otros *switches*, para que los receptores sepan a cuál VLAN pertenece la trama. Todos los dispositivos de interconexión que soportan V-LAN con etiquetado de tramas deben seguir la norma IEEE 802.1Q que especifica el funcionamiento y administración de redes virtuales. El protocolo presenta un mecanismo que permite a múltiples redes compartir de forma transparente el mismo medio físico, sin problemas de interferencia entre ellas.

La norma IEEE 802.1Q provee un método de etiquetado con información V-LAN que modifica el formato de la trama *Ethernet*. El estándar añade 4 $bytes$ al encabezado *Ethernet* original, entre el campo dirección MAC fuente y el campo Tipo, como se puede ver en la Fig. 6.20. El proceso de insertar esta etiqueta 802.1Q en la trama Ethernet resulta en una alteración de la misma, con el consecuente re-cálculo del campo CRC para control de errores. Este trabajo es realizado automáticamente por el primer *switch*, justo antes de enviar la trama sobre el puerto que corresponda. Los *switches* intermedios no recalculan el identificador, pero el *switch* final lo remueve.

El valor del campo TPID de la Fig. 6.20 lleva el número "$0x8100$" para señalar el cambio en el formato de la trama. Los siguientes tres campos, Prioridad, CFI y VLAN ID se conocen con el nombre de Información de Control de Etiqueta (TCI, Tag Control Information). El campo de 3 $bits$ señalado como PRI significa prioridad y sirve para dar tratamiento especial a aquellos servicios sensibles a la latencia en el tiempo, tales como VoIP. La bandera CFI es el Identificador de Formato Canónico. Cuando este bit se configura en "0", indica que la información se debe leer en el formato canónico de Ethernet: de derecha a izquierda o los bits de menor peso primero. Para *switches* Ethernet debe estar siempre en "0".

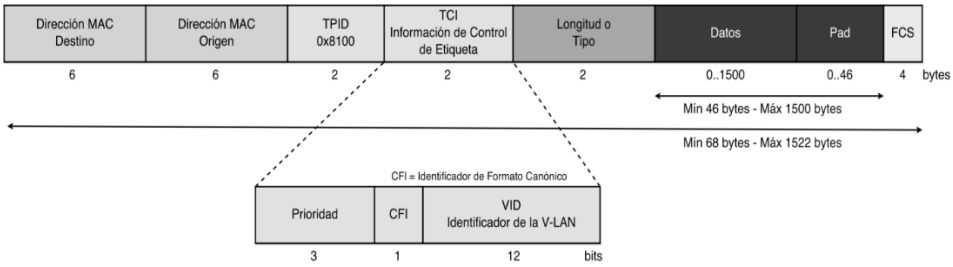


Figura 6.20 - Trama Ethernet para V-LAN

El campo más importante es el Identificador de la V-LAN, VID, de 12 bits, distintivo de cada grupo de trabajo. Se trata de un número en el rango $1\ a\ 4094$ inclusive. Los valores 0 y 4095 son de uso reservado y no deberían utilizarse. La primer red VLAN, con $VID = 1$, es la VLAN default, sobre la cual trabajarán los puertos del *switch*, a menos que se configure otro número.

En un entorno como el descripto, una interfaz Ethernet puede funcionar como puerto de acceso o como puerto troncal, pero no como ambos al mismo tiempo. Un puerto de acceso tiene sólo un identificador de VLAN asociado. Un puerto troncal es capaz de soportar la configuración de más de una VLAN y, por lo tanto, puede transportar tráfico para más de una red virtual al mismo tiempo. Generalmente es un enlace punto a punto entre *switches* que transporta el tráfico de varias redes virtuales sobre el mismo cable, permitiendo extender la propia red virtual.

Como se ve, la configuración de una red VLAN permite definir una red LAN por encima de la red física, ofreciendo ventajas en cuanto a flexibilidad, seguridad y segmentación. La flexibilidad se traduce en la posibilidad de efectuar cambios de arquitectura facilitados simplemente por una configuración apropiada. A su vez, el análisis de las tramas en los *switchs* debido al agregado de un encapsulado adicional podría mejorar las condiciones de seguridad en determinados entornos. Por otra parte, una configuración adaptada a cada red particular puede colaborar en el reparto de tráfico por medio de una segmentación apropiada.

Bibliografía

1. Stallings, William, “Comunicaciones y Redes de Computadores”. Sexta Edición. Prentice Hall Inc., 2000.
2. Halsall, Fred, “Data communications, Computer Networks and Open Systems”. Fourth Edition. Addison-Wesley, 1996.
3. Castiñeira Moreira, Jorge and Farrell, Patrick Guy, “Codificación para el Control de Errores”. Eudem, 2012.
4. Zarlink Semiconductor Inc., “2B1Q Line Code Tutorial”, MSAN-127, Application Note http://www.datasheetarchive.com/dlmain/Datasheets-41/DSA-803739.pdf
5. Noseworthy, Bob, “Gigabit Ethernet - Focus - Physical Coding Sublayer (PCS) Functional Basics and Overview” 1998. https://www.iol.unh.edu/sites/default/files/knowledgebase/ge/pcs.pdf
6. Gigabit Ethernet Alliance, “Gigabit Ethernet, 1000 BASE T Whitepaper”, 1997. http://www.dveo.com/pdf/GEA1000BASET1197_rev-wp.pdf
7. Healey, Adam, “1000BASE-T Technology Overview”, 1997
8. Knickerbocker, Jason, “Gigabit Ethernet Over Copper Performance”, Marvell Whitepaper, 2003. http://www.ic-on-line.cn/
9. Patwardhan, Shriram, “Gigabit Ethernet over Copper: Hardware Architecture and Operation”, Power Solutions, 2001. www.dell.com/powersolutions
10. Moorthy, Vijay, “Gigabit Ethernet”, 1997. http://www.cs.wustl.edu/~jain/cis788-97/ftp/gigabit_ethernet/#BaseT
11. Thaler, Patricia, Finn, Norman, Fedyk, Parsons Don, Glenn, Gray, Eric, “IEEE 802.1Q
12. Media Access Control Bridges and Virtual Bridged Local Area Networks”, 2013. IETF Tutorial. http://www.ietf.org/meeting/86/tutorials/86-IEEE-8021-Thaler.pdf
13. Cisco, “Configuring Access and Trunk Interfaces”, http://www.cisco.com/en/US/docs/switches/datacenter/nexus5000/sw/configuration/nxos/Cisco_Nexus_5000_Series_NX-OS_Software_Configuration_Guide_chapter9.html

Problemas

1. Compare los tres estándares de acceso tipo 100 BASE, de acuerdo al tipo de cable, cantidad de pares utilizados, codificación, detección de colisiones y capacidad full duplex.

2. ¿Qué características determinan la utilización de MLT-3 en 100 BASE TX? ¿Paraqué se usa la codificación 4B5B?

3. Detalle cómo funciona la secuencia de Auto Negociación, sus objetivos y su capacidad de compatibilidad hacia atrás.

4. ¿Qué significa que un entorno es micro-segmentado? ¿Cuáles son las diferencias entre el modo de funcionamiento *store&forward* y el modo *cut-through*? ¿Cómo funciona el mecanismo de auto aprendizaje? ¿Por qué se agrega un mecanismo de control de flujo para transmisión full duplex?

5. ¿Por qué 1000 BASE T se transmite en 5 niveles? ¿Cómo se compensa el acercamiento entre niveles de decisión sin aumentar la potencia? ¿Cómo se alcanza la velocidad de $1\ Gbps$? ¿Por qué no se modifica el diámetro máximo de la red con respecto a las redes de 100 Mbps? ¿Presenta alguna modificación a nivel MAC?

6. Indique si las siguientes afirmaciones son Verdaderas o Falsas. Justifique su respuesta.

a) La segmentación de una LAN implica la utilización de dispositivos tipo *switches* o puentes.
b) En un entorno LAN con *switches*, los cambios de topología podrían interpretarse como movimientos entre puertos.
c) Cada símbolo en 100 BASE T4 equivale al tiempo de $3\ bits$.
d) La trama en 100 BASE TX es igual que la de Ethernet 10 BASE T.
e) La sincronización en 100 BASE TX depende del preámbulo y el bit de comienzo de datos.
f) La técnica de *scrambling* sirve para evitar interferencias porque ensanchan el espectro.
g) 100 BASE TX permite usar $200\ Mbps$ en el cable.
h) 1000 BASE T usa alternativamente los 4 pares para transmisión y recepción.
i) En 1000 BASE T, la mayor cantidad de niveles obliga a aumentar el nivel de señal para mantener la P_{be}.
j) En 100 BASE T4 se utiliza señalización 8B6T para asegurar nivel nulo de continua.

7. ¿Por qué en una VLAN el cambio de puerto de conexión de una máquina al *switch* no afecta su conectividad? ¿Qué diferencias existen entre un puerto de acceso y un puerto troncal?

CAPÍTULO VII

Redes LAN Inalámbricas

Una red de área local inalámbrica es un sistema de comunicación muy flexible y, por este motivo, cada vez más utilizado como alternativa a las redes de área local cableadas, o a modo de extensión geográfica de las mismas. Además de haberse desplegado en muchos ámbitos relacionados con la producción, estas redes se están haciendo muy populares en los hogares, para compartir el acceso a Internet entre varias computadoras.

Las redes inalámbricas utilizan tecnologías de radiofrecuencia con el propósito de permitir movilidad de los usuarios. Aparte de esta movilidad, una de las principales ventajas de las redes inalámbricas, es la ausencia o minimización de la fase previa de tendido del cableado.

En este capítulo se presentarán los problemas relacionados con la comunicación debido a las características del canal inalámbrico, así como las herramientas incorporadas a nivel de protocolo de acceso al medio para ayudar a superarlos. También se resaltarán todos los aspectos administrativos que permiten que los elementos móviles funcionen en modo ahorro de potencia para no desgastar el tiempo de vida útil de sus baterías.

El capítulo comienza mencionando los estándares más conocidos y la arquitectura general de una red inalámbrica. Luego se abordan las características más relevantes que se incorporaron al método de acceso para enfrentar los desafíos propios de la transmisión en el medio no guiado. También se presenta de manera detallada el formato general de las tramas y los formatos particulares para el caso de las tramas de control y administración. Finalmente, se discuten los aspectos relacionados a la búsqueda de una red para poder incorporarse a la misma, las funcionalidades agregadas para el mantenimiento del sincronismo de las móviles con respecto a la red, la forma de autenticarse ante la misma y los mecanismos provistos para funcionar en modo ahorro de potencia.

7.1 Estándares IEEE 802.11

La utilización de medios de comunicación inalámbricos se encuentra en continua expansión. Uno de los ejemplos del desarrollo explosivo de este tipo de comunicación el sistema de telefonía celular. La gran ventaja de la comunicación inalámbrica es su flexibilidad y la posibilidad de brindar movilidad al usuario, quien no precisa utilizar un medio físico cableado para lograr la conexión. El medio de comunicación en la transmisión inalámbrica es el de la propagación de ondas electromagnéticas que son transmitidas y recibidas utilizando equipos de transmisión y recepción provistos de antenas.

En términos de las redes de datos inalámbricas, el estándar más aceptado es el que se conoce como IEEE 802.11. Se trata de un estándar descriptivo de redes tipo LAN inalámbricas, también conocidas como WLAN (Wireless LAN). En los últimos tiempos, se impuso la denominación de tecnología WiFi a la denominación tradicional del protocolo IEEE 802.11. En realidad, WiFi es un proceso de certificación de equipos que ofrecen niveles de compatibilidad de manufacturación con el propio estándar. La Alianza WiFi define productos tipo de red de área local inalámbrica, basados en el estándar IEEE 802.11. Es decir que se puede usar el término WiFi como sinónimo de WLAN. Sólo aquellos productos que completan las pruebas de certificación de interoperabilidad exigidas por la Alianza WiFi pueden etiquetarse como *WiFi CERTIFIED*.

Cualquier dispositivo que pueda usar certificado WiFi, ya sea una PC, un teléfono, una *notebook*, una *netbook* o una *tablet*, puede conectarse a los recursos de una red, por ejemplo para tener acceso a Internet, por medio de un Punto de Acceso (AP, Access Point). Un AP puede tener un área de cobertura de varias decenas de metros puertas adentro de alguna edificación, aunque fuera de edificios, sin obstáculos importantes, puede llegar a mayor distancia. Al ser el medio inalámbrico, puede resultar un acceso menos seguro que por una conexión cableada, motivo por el cual se han adoptado nuevos protocolos de cifrado de datos y de autenticación de usuarios.

En la transmisión inalámbrica para redes de datos se utilizan las regiones del espectro radioeléctrico conocidas como banda Industrial, Científica y Médica (ISM, Industrial, Scientific and Medical), que comienza en 2.4 GHz.

En cierto sentido, el estándar IEEE 802.11 presenta similitudes con el estándar de redes fijas IEEE 802.3, conocido también como estándar Ethernet. Sin embargo, hay razones para introducir significativos cambios en el caso de las redes inalámbricas, sobre todo por aquello que tiene relevancia respecto de las características físicas del enlace. Una coincidencia entre ambos protocolos es que a las placas de red inalámbricas también se les asignan direcciones MAC de 48 $bits$, de manera que pueden ser vistas como placas de interfaz del estándar Ethernet. Las direcciones MAC de 802.11 también se almacenan en tablas caché del Protocolo de Resolución de Direcciones (ARP, Address Resolution Protocol) y son en general indistinguibles de las que vienen grabadas en las placas Ethernet.

Sin embargo, existen dispositivos de IEEE 802.11, tales como los Puntos de Acceso, que lucen de manera diferente y presentan diferencias significativas con respecto a los dispositivos característicos conectados a las redes Ethernet.

La Tabla 7.1 describe los estándares IEEE 802.11 más conocidos, resaltando sus principales parámetros de funcionamiento. Se observa que todos los protocolos mencionados trabajan en la banda de $2.4\ GHz$, excepto IEEE 802.11a, definido específicamente para trabajar en la banda de $5\ GHz$, y 802.11n que puede trabajar en ambas bandas.

En América, la banda ISM de $2.4\ GHz$ para IEEE 802.11b se considera dividida en 11 canales de $5\ MHz$ de ancho de banda cada uno, el primero con frecuencia central $2.1417\ GHz$ y el último centrado en $2.467\ GHz$. Para IEEE 802.11g y 802.11n, el primer canal posee su frecuencia central en $2.1412\ GHz$ y el último en $2.462\ GHz$. Bajo la misma norma regulatoria americana, los 8 canales para IEEE 802.11a en la banda de $5\ GHz$, se encuentran separados cada $20\ MHz$. Los canales recomendados para redes inalámbricas en la banda de $2.4\ GHz$ son 1, 6, y 11 dado que se considerarían canales no solapados, como se verá más adelante. Para la banda de $5\ GHz$ se recomienda el uso de los canales 36, 40, 44 y 48, ya que supuestamente son menos vulnerables a interferencias debido a que las redes en estos canales funcionan con menos potencia que los del otro extremo de la banda.

Tabla 7.1 - Estándares IEEE 802.11 más conocidos

IEEE Standard	Velocidad	Banda de Frecuencia	Comentario
802.11	1 Mbps / 2 Mbps	2.4 GHz	1997 FHSS/DHSS
802.11a	Hasta 54 Mbps	5 GHz	1999 OFDM
802.11b	5.5 Mbps / 11 Mbps	2.4 GHz	Equipamiento comercial HR/DSSS
802.11g	Hasta 54 Mbps	2.4 GHz	Equipamiento comercial OFDM
802.11n	600 Mbps	2.4GHz y 5 GHz	2009 Equipamiento comercial OFDM MIMO

El estándar IEEE 802.11 es el original, que describe una capa física que utiliza técnicas de Espectro Esparcido por Salto en Frecuencia (FHSS, Frequency Hopping Spread Spectrum) y de Espectro Esparcido por Secuencia Directa (DSSS, Direct Sequence Spread Spectrum) para obtener una velocidad de transferencia de datos de $1\ Mbps$ y hasta $2Mbps$.

El estándar IEEE 802.11a, describe la capa física que utiliza Multiplexado por División en Frecuencia Ortogonal (OFDM, Orthogonal Frequency Division Multiplexing) para alcanzar velocidades de hasta $54\ Mbps$ en la banda de $5\ GHz$.

IEEE 802.11b refiere una transmisión del tipo Alta Velocidad/Espectro Esparcido por Secuencia Directa (HR/DSSS, High Rate/Direct Sequence Spread Spectrum), tratándose del primer protocolo estándar de desarrollo comercial para

transmisión inalámbrica de datos en redes que operan en la banda ISM. Esta banda es de uso libre siempre que los dispositivos sean de baja potencia.

Luego fue estandarizado IEEE 802.11g, de gran aceptación comercial, que usa OFDM para alcanzar velocidades de 54 $Mbps$ en la misma banda que 802.11b.

El más moderno de todos los protocolos es IEEE 802.11n, que puede trabajar en las dos bandas de 2.4 GHz y 5 GHz, permitiendo alcanzar mayores velocidades que sus predecesores.

La banda usada no necesariamente determina la velocidad máxima. Por ejemplo, un dispositivo 802.11a en la banda de 5 GHz puede llegar hasta una velocidad de 54 $Mbps$, igual que otro dispositivo 802.11g trabajando en la banda de 2.4 GHz. Muchas veces es el entorno lo que se debe considerar al desplegar la red. La banda de 5 GHz es menos vulnerable a la interferencia por que la mayoría de los dispositivos, tales como teléfonos inalámbricos, hornos microonda, computadoras en WLAN y dispositivos Bluetooth, funcionan en la banda de 2.4 GHz. La contrapartida es que, a mayor frecuencia, menor es el rango de cobertura. El éxito que promete el despliegue en la banda de 5 GHz parece tener que ver más con el ancho de banda disponible, mayor que el ofrecido en 2.4 GHz, haciéndolo apropiado para aplicaciones que requieren ancho de banda de manera ininterrumpida, tales como la transferencia en tiempo real de música y video.

La Fig. 7.1 presenta la ubicación de los estándares mencionados anteriormente en el Modelo OSI. Como se puede observar, la variedad de protocolos extendidos a partir del original IEEE 802 a nivel de capa física, permite imaginar la evolución tecnológica de las técnicas de comunicaciones con el objetivo de lograr mayor velocidad de transmisión en un medio tan ruidoso como el canal inalámbrico.

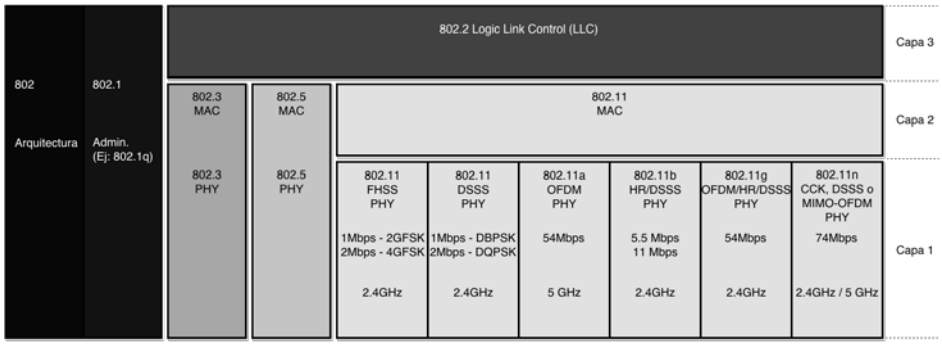


Figura 7.1 - Ubicación de los estándares IEEE 802 en el modelo OSI.

Por su parte, la capa física se divide en dos subcapas, para distribuir las funcionalidades complejas que se deben implementar a este nivel. La subcapa de Procedimiento de Convergencia de la Capa Física (PLCP, Physical Layer Convergence Procedure) traslada las tramas MAC al medio, agregando algunos campos a nivel de capa física. El sistema Dependiente del Medio Físico (PMD,

Physical Medium Dependent) es la subcapa que realmente transmite y recibe esas tramas. Más adelante nos referiremos a estas subcapas con mayor detalle.

7.2 Arquitectura WLAN

La arquitectura básica más utilizada de una red LAN inalámbrica es semejante a la desplegada en telefonía celular. En esta arquitectura, el sistema se subdivide en celdas, cada una controlada por una estación base, denominada en este caso AP.

Los componentes físicos de la red WLAN son:

- **Medio Inalámbrico**: sobre el mismo se mueven tramas, con todos los desafíos que ello implica, dadas las características del canal.

- **Estaciones Móviles**: dispositivos con placas de red inalámbricas.

- **Puntos de Acceso, AP**: generalmente realizan una función de puente, traduciendo tramas entre la red cableada, que suele conocerse como la columna vertebral o *backbone*, y la LAN inalámbrica. El más común de los *backbones* es una red *Ethernet*.

- **Sistema de Distribución**: es la red formada por varios AP pertenecientes a una misma administración. Los AP deben comunicarse entre sí, sobre la red cableada, para seguir el movimiento de las estaciones móviles, permitiendo que su desplazamiento en la red sea transparente al usuario. La comunicación con las estaciones móviles se rige según lo establecido por IEEE 802.11, pero la comunicación entre ellos sobre la red cableada no se ha estandarizado.

Como se puede apreciar en la Fig. 7.2, un grupo de estaciones comunicadas en modo inalámbrico constituyen lo que se denomina un Conjunto de Servicio Básico (BSS, Basic Service Set), dando forma a la unidad elemental que define una red inalámbrica. La región espacial que determina el rango de operación válido de la red tiene límites difusos, debido a la naturaleza misma del proceso de comunicación. Si una estación se encuentra en un BSS, se puede comunicar con las demás que se encuentren en el mismo BSS.

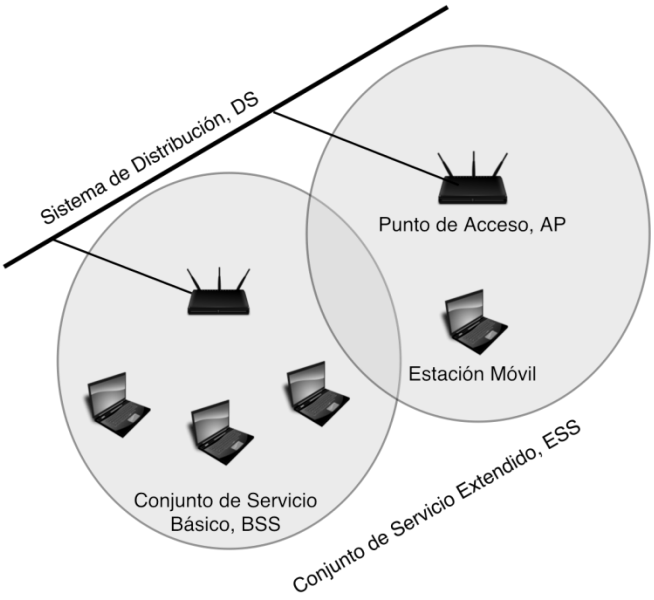


Figura 7.2 - Arquitectura WLAN. BSS, Sistema de Distribución y ESS

Como se presenta en la Fig. 7.3, existen dos maneras de desplegar un BSS: red Independiente, también conocida como red *ad hoc* o IBSS, y red de Infraestructura.

Las redes Independientes son redes desplegadas sólo entre estaciones móviles, donde cada estación se comunica con las otras de manera independiente. Generalmente se trata de pocas máquinas que se deben comunicar con algún propósito especial por un período corto de tiempo.

Por su parte, las redes de Infraestructura incluyen un AP que implica un manejo centralizado de la comunicación. Se habla de comunicación multi-salto o *multihop*, donde los dispositivos se pueden comunicar entre sí sólo a través del AP. En este modo, la restricción se centra en la distancia al AP, no en la cercanía o lejanía existente entre estaciones. La característica principal es que los AP pueden contribuir con funcionalidades de ahorro de potencia, en tanto que en las redes *ad hoc* la funcionalidad de ahorro de potencia tiene características distribuidas.

En un BSS de Infraestructura, cada estación debe realizar un proceso de asociación con el AP, que es un paso semejante a la conexión física que se hace en la red fija por medio de un conector de cable. El proceso de asociación es iniciado por la estación, siendo el AP el encargado de administrar tal solicitud, pudiendo acceder o denegar la petición. Una restricción importante de destacar es que cada estación puede asociarse solamente con un único AP.

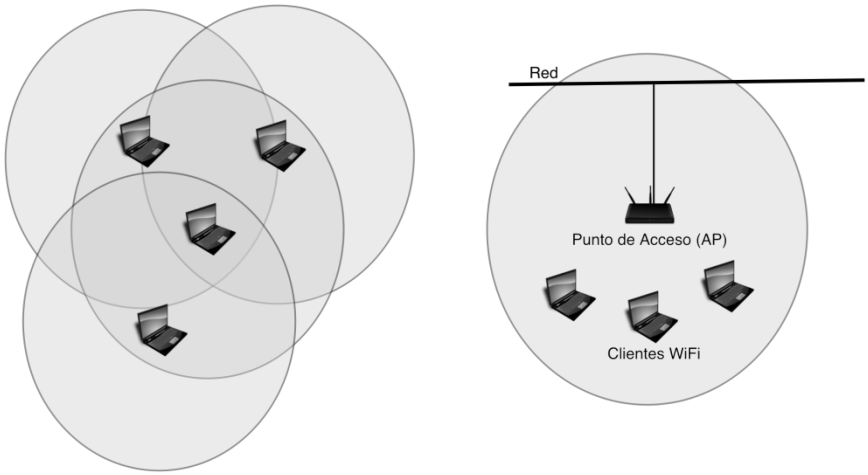


Figura 7.3 - Red ad-hoc (IBSS) y red de Infraestructura.

El conjunto completo de LAN inalámbricas interconectadas se conoce como Conjunto de Servicio Extendido (ESS, Extended Service Set). Se trata del dominio completo de un administrador, que incluye los diferentes BSS, cada uno con su respectivo AP, y el sistema de distribución. En redes de Infraestructura, los AP utilizan el sistema de distribución para ofrecer un servicio de integración, ya que permiten conectar la WLAN a una red que no funcione bajo el estándar 802.11 intercambiar tramas.

Además del proceso de registro frente al AP, denominado asociación, al moverse dentro de un mismo ESS, la estación móvil detecta potencia y, si ésta detección se traduce en bajo nivel, puede llegar a cambiar de AP. Cambiar de AP significa mudarse del BSS en el que se había registrado. Para poder hacerlo, debe desasociarse del AP actual y re-asociarse frente a otro AP. En el sistema de distribución, los AP nuevo y viejo deben comunicarse entre sí, para actualizar la información de la estación re-asociada y permitir así el intercambio correcto de tramas de manera transparente al usuario. Es oportuno aclarar que, previo a un proceso de asociación, el protocolo exige una fase de autenticación de los equipos móviles frente al AP, proveyendo además un esquema de cifrado de datos a modo de protección contra ataques.

El estándar denomina movilidad sin transición a la posibilidad de que las estaciones se muevan dentro del alcance del AP al que se encuentran asociadas, es decir que se permite el movimiento dentro de un BSS. También se asegura la transición entre BSS, o sea la movilidad entre puntos de acceso que pertenecen al mismo ESS. Si se transita desde un AP a otro, dentro del mismo ESS, la estación debe re-asociarse. Como se ha observado antes, en este caso los AP involucrados, tanto el de origen como el de destino, deben colaborar entre sí, informando sobre los cambios. Todavía no existe un protocolo estandarizado para esta comunicación entre AP aunque, en términos generales, se lo denomina Protocolo entre AP (IAPP, Inter Access Point Protocol), entendiendo que se trata de protocolos de carácter propietario.

El estándar no permite la movilidad entre diferentes ESS, por el consecuente problema de mantenimiento de conexiones a nivel superior.

7.3 MAC IEEE 802.11

Más allá de las funcionalidades de la subcapa MAC que hemos estudiado con respecto al estándar IEEE 802.3, la MAC 802.11 realiza otras funciones mucho más amplias, generalmente relacionadas con los problemas de transmisión que presenta el medio inalámbrico y con aquellas previsiones que han de tomarse para evitar el consumo excesivo de las baterías de los dispositivos móviles. Por ejemplo, la subcapa MAC IEEE 802.11 posee funcionalidad para permitir cerrar por momentos el sistema de recepción y aún así no perder conectividad. También a este nivel se puede realizar fragmentación, retransmisión de tramas y reconocimiento de tramas llegadas sin errores. Algunas de estas funcionalidades exigen la definición de tramas especiales, como es el caso de la trama de reconocimiento denominada ACK.

El estándar IEEE 802.11 puede adaptarse a distintas capas físicas por debajo de la subcapa MAC. De este modo, es posible que en el medio físico existan diferentes velocidades, debidas a diferentes esquemas de modulación, distintos canales o portadoras y diferentes características de detección del medio. Una manera de acomodar tantas capas físicas diversas y poder adaptarse a las condiciones variables del canal es el agregando, a nivel de la MAC, de la capacidad de fragmentación. Dado que en entornos inalámbricos la P_{be} es muy grande, la probabilidad de que una trama se contamine con ruido o colisione con otra aumenta de manera proporcional al tamaño de la misma, motivo por el cual los diseñadores del estándar supusieron que la posibilidad de transmitir tramas de menor tamaño permitiría reducir el costo asociado a la retransmisión.

Por su parte, la cuestión económica de la preservación del tiempo de vida de las baterías exigió el agregado de mecanismos de ahorro de potencia que generaron funcionalidades extras de administración y la aparición de tramas especiales, tales como el mensaje conocido como Faro o *Beacon.*

También, la necesidad de evitar el acceso a usuarios no autorizados y la posibilidad de bloquear el husmeo o *sniffing* de datos privados, determinó el agregado de un mecanismo de autenticación y la definición de un protocolo de cifrado, denominado Privacidad Cableada Equivalente (WEP, Wired Equivalent Privacy) en el protocolo original. Este algoritmo fue violado antes de la aparición oficial del estándar, situación que obligó posteriormente a adicionar esquemas de seguridad más apropiados.

A su vez, la necesidad de satisfacer diferentes entornos, con distintas posibilidades de acceso, fijó la definición de tres tipos de WLAN, sobre los que se brindarán mayores detalles más adelante: Infraestructura, Ad-Hoc y Puente Inalámbrico.

Por otra parte, frente a la creciente expectativa respecto del transporte de tráfico multimedia en tiempo real, el estándar previó dos métodos de acceso al medio diferentes: la Función de Coordinación Distribuida (DCF, Distributed

Coordination Function) y la Función de Coordinación de Punto (PCF, Point Coordination Function).

PCF provee un servicio libre de contienda, tratándose de un mecanismo de acceso que se sirve de puntos de coordinación en los AP, por lo que sólo puede desplegarse en redes tipo Infraestructura. Su agregado en el estándar se debe a la aparición de nuevas aplicaciones, que poseen limitaciones de tiempo más exigentes y suponen cierto grado de calidad de servicio. Como se trata de un modo opcional, sólo pocos AP o tarjetas de red lo implementan. Por este motivo, PCF no será objeto de desarrollo en este libro.

DCF es la base del mecanismo de acceso por contienda. El método de acceso al medio se conoce como Acceso Múltiple por Detección de Portadora/ Prevención de Colisión (CSMA/CA, Carrier Sense Multiple Access/Collision Avoidance). DCF prevé asegurar la movilidad con cierto grado de solapamiento, debido al hecho de que puedan existir distintas redes en la misma área de trabajo y espacio de canales. Este re-uso del medio compartido debió acompañarse con un esquema de acceso apropiado, por eso se dotó al modo DCF con el método CSMA/CA. En el esquema propuesto, se detecta el medio antes de transmitir y se agrega un algoritmo para evitar colisiones que se basa en un mecanismo de *back off* aleatorio.

La detección de portadora se vuelve más compleja en un medio inalámbrico, ya que la señal sólo pueden escucharla los dispositivos que se encuentran en el área de alcance. En general, los transmisores y receptores de la red inalámbrica son de un solo sentido o *half duplex*, de manera que no pueden transmitir y recibir al mismo tiempo, haciéndolo sólo de manera alternada. Una parte de la solución a este problema es la modificación del esquema original, permitiendo un mecanismo RTS/CTS para enfrentar el problema de nodo oculto. En este caso, el protocolo propone el uso de tramas especiales, conocidas como Requerimiento de Transmisión (RTS, Request To Send) y Listo para la Transmisión (CTS, Clear To Send), como mecanismo de reserva del medio y disminución de la probabilidad de colisión.

Por ejemplo, si la estación A de la Fig. 7.4 transmitiera una trama RTS al AP indicándole su deseo de iniciar una comunicación, generaría un silenciamiento de parte de todas las estaciones que la reciban, es decir aquellas estaciones que están al alcance de A. En este ejemplo, sólo el AP escucha este requerimiento pues es el único dispositivo que se encuentra dentro del alcance de A. El AP responderá con una trama CTS, indicando autorización para transmitir. Esta trama será recibida tanto por A como por el resto de las estaciones dentro de su alcance, en este caso B y C. Por lo tanto, B y C se abstendrán de transmitir hasta que la estación A haya enviado su trama de datos y recibido la trama de reconocimiento ACK correspondiente por parte del AP.

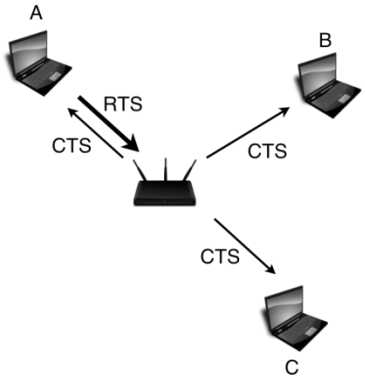


Figura 7.4 - Solución al problema del Nodo Oculto.

El procedimiento de transmisión por medio de la señalización RTS/CTS es eficiente sólo en aquellos casos de alto grado de colisión o en redes con muchas estaciones, pues de por sí significa un consumo excesivo de recursos y aumento de latencia, debido a que para poder intercambiar dos tramas, la de datos y su ACK, se precisan intercambiar cuatro tramas. El procedimiento puede ser ajustado controlando el umbral de RTS, que es esencialmente un ajuste del tamaño de las tramas sujetas al procedimiento en cuestión, configurable por parte del administrador.

Para mejorar la detección de portadora, se agregó un mecanismo de detección virtual a la propia detección física del medio. Por medio de un campo en el encabezado de la trama, conocido como campo de Duración/ID, en determinadas circunstancias se fija el valor de un Vector de Asignación de Red (NAV,Network Allocation Vector), donde se consigna la duración de un intercambio, para que el resto de los dispositivos se abstenga de acceder durante ese tiempo.

Por otra parte, como el medio es muy ruidoso y con probables fuentes de interferencia, ya sea de otros equipamientos o de otros canales, los diseñadores del protocolo eligieron dotar de confiabilidad a la transmisión de tramas *unicast*, a través del mecanismo de ACK y retransmisión. Además, las tramas de *unicast* se transmiten bajo un esquema de bloqueo de contienda, denominado Operación Atómica: una vez que se accede al medio y se transmite la trama de datos, si ésta se recibe correctamente, el receptor genera una trama de ACK, que puede ser enviada con prioridad, sin tener que contender por el medio, pues para el resto de los dispositivos se sostiene un bloqueo de contienda. Se trata de una operación que implica la transmisión de varias tramas, pero que se realiza como una operación única, donde sólo la primera trama debe contender por el medio. Cabe aclarar que este mecanismo no tiene lugar en el caso de transmisión de tramas tipo *broadcast*.

7.3.1 Método de Acceso CSMA/CA

El mecanismo básico de acceso, llamado DCF, es un acceso por detección de portadora acompañado de un mecanismo para evitar la colisión que se conoce como CSMA/CA. En el entorno *Ethernet* se ha presentado un mecanismo parecido, denominado CSMA/CD. Ambos protocolos funcionan muy bien en entornos no excesivamente cargados ya que permiten a las estaciones transmitir con el mínimo retardo, aunque pueda haber colisiones. Las situaciones de colisión se deben identificar porque, en dichos casos, la MAC o alguna capa superior, debe retransmitir las tramas o los mensajes encapsulados en ellas.

Mientras que el mecanismo de detección de colisiones funciona muy bien en redes cableadas, no se puede aplicar a entornos inalámbricos, principalmente por dos motivos. Tal y como se plantea en Ethernet, para poder implementar este mecanismo en el entorno inalámbrico, se requeriría la construcción de un dispositivo con capacidad para transmitir y recibir al mismo tiempo, incrementándose su precio significativamente. Por otro lado, en un entorno inalámbrico, por cuestiones de alcance de las señales en el medio, no todas las estaciones se pueden ver entre sí, siendo esta premisa la base supuesta para la detección de colisiones en las redes cableadas.

Por estas razones, IEEE 802.11 propone un manejo diferente de las colisiones, que se conoce como mecanismo para evitar la colisión (CA) acompañado de un esquema de reconocimiento ACK positivo cuando las tramas se reciben correctamente.

Así, cuando una estación se encuentra en estado de transmisión, detecta el medio. Si éste se encuentra ocupado, difiere. Diferir significa que no lo ocupa directamente, sino se abstiene por un tiempo. Si luego el medio se encuentra libre durante un tiempo especificado por el protocolo, entonces recién puede comenzar a transmitir. Esta situación de diferir se observará repetidas veces en el acceso, ya sea que el medio se encuentre libre como en el caso en que se encuentre ocupado, y es la que se relaciona con el concepto de evitar la colisión.

Por su parte, la estación receptora coloca en *buffer* la trama en tránsito, chequea el código de redundancia cíclica que la acompaña y, si éste se verifica como correcto, puede disponer de la trama si le corresponde y aparte enviar un ACK al transmisor. Cuando el transmisor recibe la trama de ACK, lo interpreta como indicador de que no hubo colisiones. Si no recibe el ACK, debe retransmitir la trama enviada hasta que reciba un ACK o desistir si lo ha intentado cierto número de veces sin éxito.

7.3.2 Detección Virtual de la Portadora

Como se ha explicado, el protocolo agrega un mecanismo de detección virtual al mecanismo de detección física del medio. Ambos resultan útiles para reducir la probabilidad de colisión.

La detección física de la portadora depende del medio y del esquema de modulación, tratándose de una funcionalidad más compleja que en el caso de

LAN cableadas, porque para transmitir y recibir al mismo tiempo, en las WLAN se encarece bastante la electrónica, sin suficientes garantías de operación, entre otras cosas por la presencia de nodos potencialmente ocultos. Por esto motivo se asiste la detección física con la detección virtual. Esta última se realiza a través de un campo en el encabezado de la trama que anticipa la duración en el tiempo de la transmisión a la que pertenece la trama. En realidad, se trata de un mecanismo de reserva del medio por cierto tiempo anunciado en ese campo. Las estaciones que escuchen el medio y lean el campo de Duración, ajustarán sus propios NAV de acuerdo a los valores recibidos, aplazando el acceso al medio en base a este tiempo. Desde el punto de vista de la electrónica, se produce un cierre del circuito de recepción por ese tiempo.

El valor del NAV se puede interpretar como el de un contador que se va disminuyendo hasta llegar a cero, momento en que la estación sale del modo de ahorro de potencia en el que había entrado, para volver a detectar el medio nuevamente, re-habilitando sus circuitos de recepción. De este modo, la detección virtual se convierte en una abstracción lógica que limita la necesidad de la detección física de la portadora, con el propósito de ahorrar potencia.

7.3.3 Espaciamiento entre tramas – IFS

El estándar define cuatro tipos de tiempos de Espaciamiento entre Tramas (IFS, Inter Frame Spacing), que se usan como un esquema de prioridades para acceso sin contienda o como una medida de lo que se debe diferir en el acceso al tratar de evitar la colisión:

- **Espaciamiento Corto entre Tramas SIFS (Short Inter Frame Space)**: se definió para manejar las transmisiones de mayor prioridad. Tal es el caso de las tramas de pedido/respuesta del tipo RTS/CTS para evitar el problema de nodo oculto y las transmisiones de tramas de datos *unicast* que llevan asociadas sendas tramas ACK.
  Por ejemplo, luego de la transmisión de una trama RTS, la trama CTS deberá ocupar el medio de manera inmediata, una vez que se haya agotado este tiempo SIFS. Esto quiere decir que la trama CTS no debe contender por el medio, pues tiene prioridad frente al resto de las posibles transmisiones de los demás integrantes de la red. Así, SIFS se usa para sincronizar transmisiones que pertenecen a un mismo diálogo.
  Como se ha mencionado, el esquema se conoce Operación Atómica, bloqueándose la contienda por el medio mediante un mecanismo interno de prioridades basado en SIFS. El mecanismo es también aplicable al envío de tramas de datos *unicast*, con reconocimiento positivo ACK en respuesta de las tramas transmitidas recibidas correctamente. SIFS se calcula como el tiempo que el transmisor precisa para conmutar a modo recepción para decodificar una trama entrante, fijándose por ejemplo en $10\ \mu seg$ en IEEE 802.11n trabajando en la banda de $2.4\ GHz$

- **Espaciamiento entre tramas en el modo PCF PIFS (Point Coordination IFS):** se definió para que fuera utilizado por el AP, llamado Punto de Coordinación en este modo, para permitirle ganar el acceso al medio antes que a cualquier otra estación. Su valor se corresponde con el de SIFS más una ranura de tiempo, siendo de $25\ \mu seg$ en IEEE 802.11n en cualquiera de las dos bandas de transmisión.

- **Espaciamiento entre tramas en el modo DCF DIFS (Distributed IFS)**: se definió para que las estaciones lo usen al momento de intentar realizar una transmisión. Su valor es el de PIFS más una ranura de tiempo, $34\ \mu seg$ en IEEE 802.11n en la banda de $5\ GHz$. Se trata del tiempo mínimo en que el medio debe estar desocupado en el modo de contienda antes de poder comenzar a transmitir. Siempre es mayor que SIFS.

- **Espaciamiento entre tramas Extendido EIFS (Extended IFS)**: Es el mayor de todos espaciamientos definidos y se usa en el caso en el que una estación ha recibido una trama que no es capaz de interpretar por haber sido recibida con errores. Su utilidad es evitar que una estación que no pueda entender la información del campo de Duración, indispensable para el mecanismo de detección virtual, colisione con la trama siguiente en el caso de una Operación Atómica.

7.3.4 DCF

La mayor parte del tráfico comúnmente se intercambia con contienda y utilizando DCF. El mecanismo DCF permite la interacción de varias estaciones, sin necesidad de control central, por lo que puede aplicarse tanto a redes Ad-hoc como a las de Infraestructura.

En la Fig. 7.5 se presenta el esquema del método de acceso al medio impuesto por DCF, que es CSMA/CA. La figura pretende graficar lo que sucede en el dominio del tiempo con una estación que intenta acceder al medio pero lo encuentra ocupado.

Antes de intentar la transmisión, la estación detecta si el canal está libre. De encontrar libre el canal, la estación debe verificar que esta situación se mantenga durante un período de tiempo DIFS, definido por el estándar, para luego poder transmitir. Si al momento de la detección, el canal se encuentra ocupado, la estación demora el acceso utilizando un algoritmo de espera exponencial para evitar colisiones.

Para explicar claramente lo que debería suceder en cada caso, el estándar define una serie de reglas:

- **Regla 1**: Si el canal permanece libre por un tiempo mayor a DIFS, entonces la estación puede comenzar inmediatamente la transmisión. La decisión que determina el estado libre del canal se toma usando ambos métodos de detección: el de portadora física y el basado en el ajuste de NAV.

- **Regla 2**: Si el canal está ocupado, la estación debe diferir el acceso, esperando que el medio se mantenga libre durante DIFS, para luego prepararse para el procedimiento de *backoff* exponencial, a través de la definición de una Ventana de Contienda (CW, Contention Window).

- **Regla Adicional 1**: El manejo de control de errores en la transmisión es responsabilidad de la estación transmisora que, luego de la emisión de una trama, espera el correspondiente ACK y debe prepararse a retransmitir hasta que la trama llegue correctamente. Las tramas de ACK son los únicos indicadores de éxito y la retransmisión se relaciona con la ausencia de estas tramas en el entorno de Operaciones Atómicas. Toda recepción correcta de tramas *unicast* se asocia a la recepción de una trama ACK. Existe un contador de retransmisiones que se incrementa cuando se falla al ganar acceso al medio o cuando no se recibe una trama de ACK, luego de haber transmitido los datos.

- **Regla Adicional 2**: Las secuencias multi-trama deben actualizar el valor del NAV en cada paso del procedimiento de transmisión. Esto significa que el ajuste del NAV se realiza sobre una base por trama.

- **Regla Adicional 3**: Las tramas de ACK, CTS (en transmisiones RTS/CTS) y los fragmentos, en secuencias de fragmentos, son las tramas de máxima prioridad. Ellas se transmiten luego del tiempo SIFS y no precisan contender por el medio. Si una estación gana el control del medio, quiere decir que ya ha enviado la primera trama de una secuencia. Las tramas adicionales y sus respectivos ACK, se enviarán teniendo en cuenta el SIFS, bloqueando de este modo la contienda con otras estaciones. A su vez, las tramas adicionales de la Operación Atómica actualizan el NAV para el tiempo que resta de la misma, para el resto de las estaciones.

- **Regla Adicional 4**: Las secuencias de tramas extendidas se requieren para tramas de niveles superiores que son más largas que los umbrales configurados. Existe un umbral para disparar el mecanismo RTS/CTS y otro para disparar el mecanismo de fragmentación. Aquellas tramas mayores en tamaño que el umbral RTS configurado, deben transmitirse con el procedimiento RTS/CTS y aquellas tramas que superan el umbral de fragmentación deberán transmitirse fragmentadas.

A partir del conjunto de reglas presentado, se puede deducir que las operaciones atómicas comienzan luego de que el medio haya estado libre durante DIFS, pero sus pasos intermedios se manejan con SIFS, pues tienen prioridad.

En la Fig. 7.6 se presenta el método CSMA/CA mediante una representación de diagrama de flujo. El diagrama pretende resaltar las diversas situaciones de acceso al medio explicadas en las reglas anteriores.

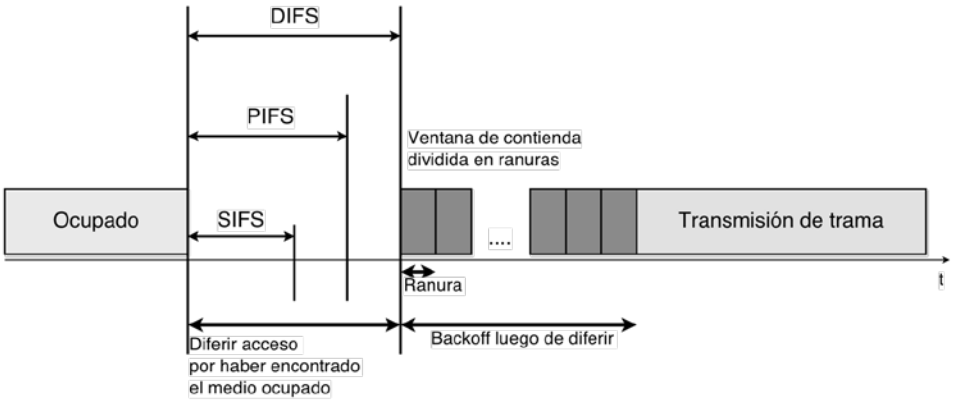


Figura 7.5 - CSMA/CA en el caso de medio ocupado.

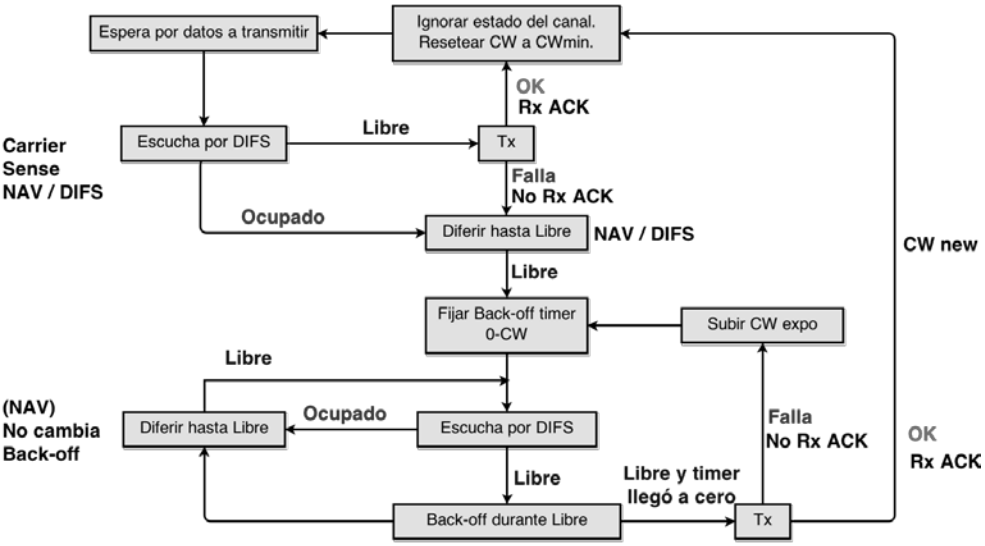


Figura 7.6 - Diagrama de Flujo CSMA/CA

La Ventana de Contienda CW es un período de tiempo que se considera dividido en intervalos de igual duración, denominados ranuras, que son más cortos cuanto más rápida sea la capa física que se utilice. Cada estación adopta aleatoriamente un número posible de ranuras, cuyo conjunto conforma la CW, y espera a que pase ese intervalo de tiempo para empezar a transmitir. Si hubiera varias estaciones intentando acceder al mismo tiempo, o en la misma ventana de contienda, aquella que por este proceso aleatorio haya adoptado el número aleatorio menor de ranuras, es la que accede al medio.

Un contador de retroceso cuenta las ranuras, a partir del número elegido aleatoriamente, disminuyéndolo hasta llegar a cero. El decremento se produce sólo si al detectar portadora durante esa ranura, el medio permanece desocupado. Si en cambio, lo encuentra ocupado, congela el valor y no lo disminuye sino hasta la próxima ranura que lo encuentre desocupado.

Cuando el contador de retroceso llega a cero, la estación puede transmitir una trama. Si la transmisión no es exitosa, entonces se configura otro valor para CW, duplicando el previo. Así, a la primera retransmisión, luego de un primer intento, la ventana pasa de ser elegida aleatoriamente dentro de un rango de 31 ranuras a otro rango de 63 ranuras. En el caso de nuevas retransmisiones, el proceso continúa hasta que se alcanza un valor límite, por ejemplo de 1023 ranuras en el caso del estándar que utiliza DSSS en capa física. La ventana de contienda se regresa al valor mínimo cuando se transmite exitosamente, o cuando el contador de retransmisiones asociado llega al máximo, descartándose la trama.

Este mecanismo reduce la probabilidad de colisión cuando existen múltiples estaciones intentando acceder al medio.

La Fig. 7.7 presenta el caso de dos estaciones comunicándose y una tercera intentando acceder al medio. Se puede apreciar que, luego que el medio permanece desocupado durante un intervalo DIFS, la estación que había encontrado el canal ocupado, determina un número aleatorio de ranuras para la ventana de contienda y empieza a disminuir el contador asociado a la misma, siempre que el medio se encuentre desocupado. Cuando llega a cero, puede iniciar la transmisión. En la figura también se resaltan los tiempos que interesan en el acceso: DIFS para canal desocupado, SIFS dentro de la Operación Atómica y CW luego de haber encontrado ocupado el canal.

En la Fig. 7.8 se presenta un intercambio de datos por medio del mecanismo de protección para el problema del nodo oculto. En la última línea de tiempo de la figura se grafican los tiempos cubiertos por los campos de Duración, que imponen el valor de NAV, anunciados en cada trama intercambiada. Sobre la línea se representan los valores de NAV para las tramas del transmisor y, por debajo de ella, para las tramas del receptor, en el caso de las tramas RTS y CTS.

El transmisor, luego de un tiempo de contienda que no se presenta en la figura, logra acceder al medio y baja la trama de RTS. En este caso, todo el intercambio, una vez que se ha accedido al medio, se convierte en una *Operación Atómica*. Es decir que se bloquea el mecanismo de contienda, con prioridades definidas por los tiempos cortos SIFS.

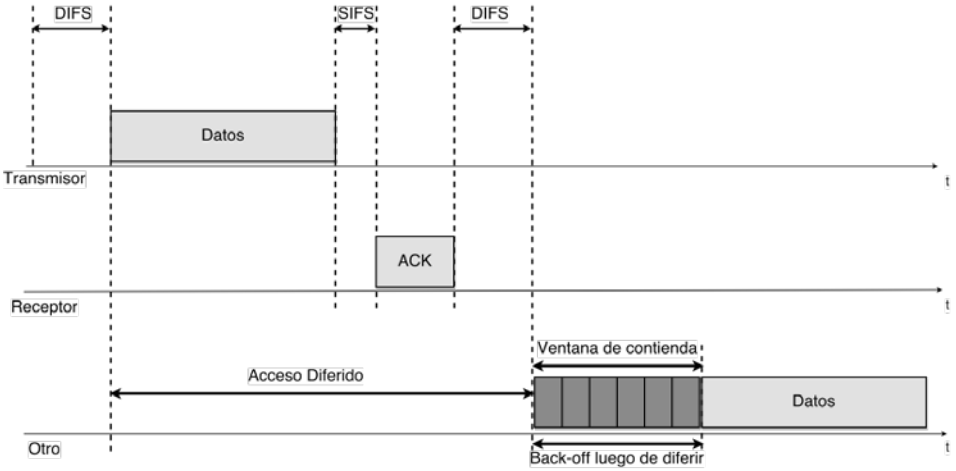


Figura 7.7 - CSMA/CA para una tercera estación que encuentra el medio ocupado.

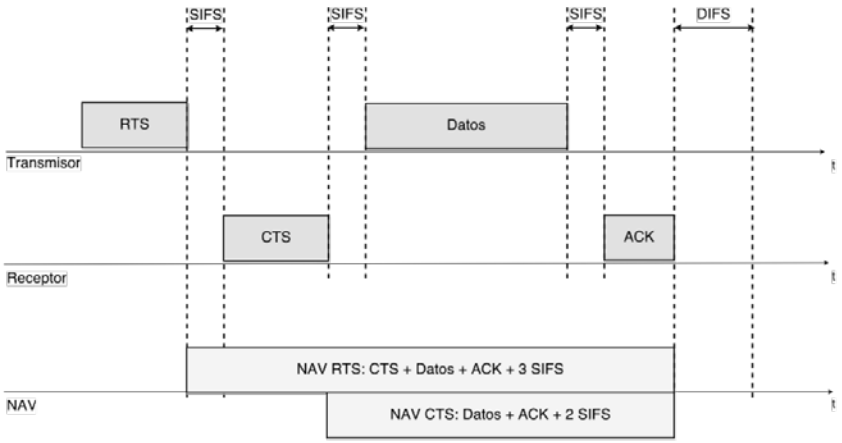


Figura 7.8 - Intercambio de datos con RTS/CTS y NAV asociados

Como se observa en la Fig. 7.8, el campo de Duración de la trama RTS lleva escrito un número que es equivalente en microsegundos a todo el tiempo necesario para que el receptor conteste con la trama CTS, el propio transmisor le envíe la trama de datos y el receptor finalmente la reconozca, de no mediar errores, mediante una trama de ACK. El valor incluye los tiempos intermedios SIFS.

Cuando el receptor recibe la trama RTS y verifica que no tiene errores, debe contestar con una trama CTS. Para eso dispone del medio sin necesidad de contender por el mismo, debiendo bajar al mismo la trama CTS ni bien finalice el período SIFS, como se aprecia en la figura. El campo de Duración de la trama CTS lleva escrito el tiempo, en microsegundos, que se demorará en el envío de los datos y la edición del ACK, incluyendo los tiempos intermedios.

Una vez que el transmisor recibe la trama CTS, dispone de un pequeño tiempo SIFS para apropiarse del medio con la trama de datos, cuyo campo de Duración, no especificado en la figura, cubre la emisión del ACK más la duración de un SIFS.

Finalmente, el receptor verifica los datos y, si estos no contienen errores, edita la trama de ACK, luego de un tiempo SIFS. El campo de Duración de la trama de ACK es nulo porque es la última trama de la Operación Atómica.

Debe quedar claro que el campo de Duración se ajusta en una base por trama, teniendo en cuenta lo que falta para completar la operación. Esto es así para contemplar el caso de aquellas estaciones que detectan portadora en el medio, mientras se está llevando adelante una Operación Atómica. De este modo, cada trama transporta en su campo Duración el tiempo que resta para completar la Operación Atómica.

La Fig. 7.9 presenta otro caso de Operación Atómica, donde se combina el mecanismo RTS/CTS con una transmisión fragmentada. Se ha mencionado que, frente a problemas de interferencia esporádica, o durante períodos cortos de tiempo, la fragmentación de tramas ofrece una solución, permitiendo que el deterioro de la información se localice en algunos fragmentos solamente, y no en la trama completa, lo que obligaría a su retransmisión y, por lo tanto, a un uso del medio más prolongado e ineficiente.

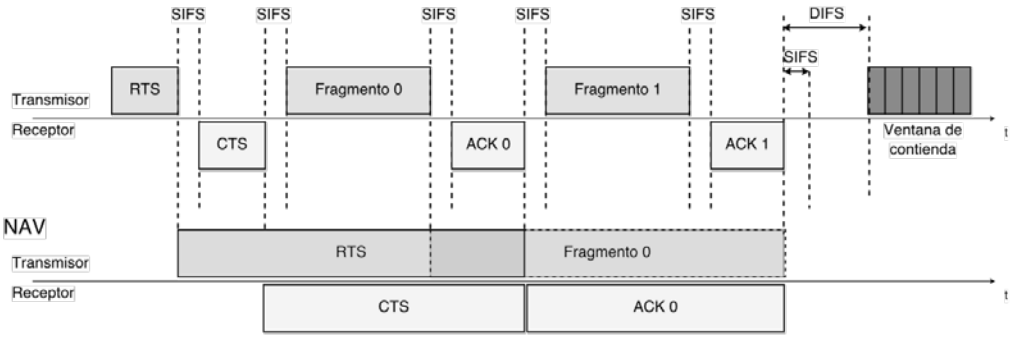


Figura 7.9 - Mecanismos de RTS/CTS y Fragmentación.

La fragmentación se produce cuando la trama excede el umbral de fragmentación ajustado por el administrador de la red. En general el umbral de fragmentación se ajusta al mismo valor que el del mecanismo RTS/CTS. Así, el proceso de comunicación con requerimiento de transmisión y aviso de habilitación para la transmisión RTS/CTS envuelve la transmisión de una única trama fragmentada.

Esta transmisión conjunta se puede apreciar en la Fig. 7.9, en la que se han agregado además los valores equivalentes de los campos de Duración para las tramas RTS, CTS, el primer fragmento, denominado fragmento 0 y su correspondiente ACK. Observar que, en este caso, la trama de RTS cubre el tiempo que falta para la transmisión de CTS, el primer fragmento y su ACK. La

trama de CTS cubre el tiempo de transmisión del primer fragmento y su ACK. Por su parte, el fragmento 0 cubre el tiempo de transmisión de su propio ACK y el que resta hasta el ACK 1, correspondiente al fragmento que le sigue. Como la transmisión incluye más de un fragmento, el ACK de cada fragmento diferente del último, cubre con su NAV la transmisión completa del fragmento que sigue.

En la transmisión fragmentada, cada fragmento ha de ser reconocido. Es de destacar que todos los fragmentos de la misma trama comparten un identificador único, que es el número de secuencia de la trama original, pero además poseen un identificador de fragmento, que se utiliza para el rearmado de la trama original. También se señaliza, por medio de un bit en el encabezado del propio fragmento, si existen más fragmentos a continuación o si se trata del último fragmento de esa transmisión.

7.4 Formato de la Trama IEEE 802.11

En la Fig. 7.10 se presentan los campos de la trama MAC 802.11. Dada la gran variedad de funciones que se deben cumplir a nivel MAC, se definen tres tipos de tramas: de datos, de control y de administración. Cada uno de estos tipos se divide a su vez en subtipos.

Aparte del encabezado MAC, existe un encabezado a nivel físico. Este encabezado se compone de un preámbulo y un encabezado propiamente dicho, conocido como encabezado PLCP. El preámbulo es parecido al de Ethernet, una serie de unos y ceros alternantes que finaliza con algún campo de delimitación de comienzo de trama SFD. Al igual que en Ethernet, el preámbulo se usa para sincronismo pero también puede ayudar en la selección de la antena apropiada, si el sistema utilizara *diversity*, tema que ampliaremos en el siguiente capítulo.

El encabezado PLCP depende de la capa física particular que haya por debajo de la MAC. En general, posee campos que identifican la velocidad, el tiempo de duración de la PDU de capa física, para que sea más fácil encontrar el final de la trama, y un código CRC exclusivo para control de errores del encabezado de capa física. Estos campos se transmiten a baja velocidad por cuestiones de compatibilidad hacia atrás con protocolos más viejos. También es para asegurarse de que el receptor use el mecanismo de demodulación correcto, ya que el mismo varía con la velocidad.

Observando la Fig. 7.10, se puede apreciar la cantidad de campos adicionales de la trama 802.11 comparada con la de 802.3. Este *overhead* tan importante, denota la complejidad de la MAC 802.11, descubriendo la riqueza de la funcionalidad asociada al protocolo.

La trama posee un Campo de Control con posibilidad de distinguir entre tipos y subtipos de trama, aclarar el sentido de traslado de la trama en el medio inalámbrico, colaborar en la fragmentación y en la posibilidad de realizar ahorro de potencia y marcar tramas retransmitidas y tramas cifradas.

El campo de Duración se utiliza para detección virtual en la mayoría de las operaciones. En una operación particular, invocada para recuperar información almacenda en el AP luego de que la estación haya entrado en el modo

ahorro de potencia, cumple el papel de identificador de asociación del dispositivo móvil.

Notablemente, existen cuatro campos de Dirección, frente a IEEE 802.3 que utiliza sólo dos. Este detalle permite entrever la variedad existente en las comunicaciones. Tres de estos campos aparecen uno a continuación del otro, pues son los de aparición frecuente. El cuarto se encuentra separado ya que su uso no es habitual, sólo aparece en un caso especial de configuración WLAN.

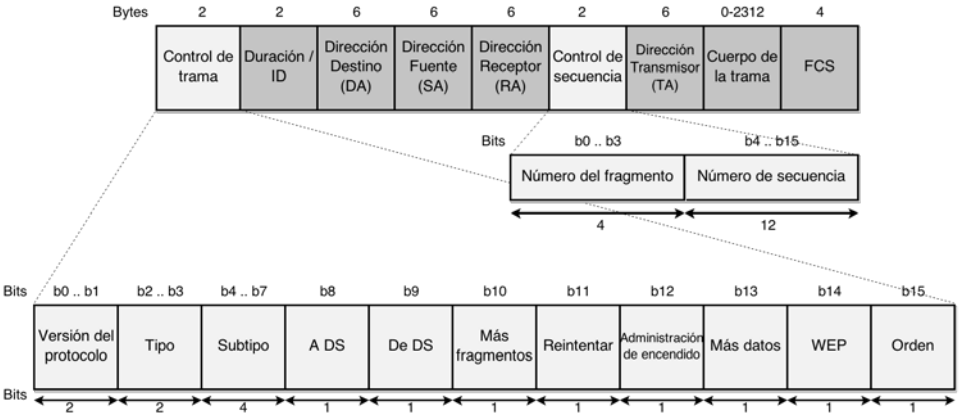


Figura 7.10 - Encabezado y Trailer de la trama IEEE 802.11.

7.4.1 Campo de Control

El *Campo de Control* de trama tiene un tamaño de dos bytes y es el primer campo de la trama MAC, al inicio del encabezado, como se aprecia en la Fig. 7.10.

- **Versión**: los dos primeros bits ($bit\ 0$ y $bit\ 1$) definen la versión de protocolo. Existe un único protocolo MAC 802.11, aunque puede ir asociado a distintas capas físicas. Por este motivo, el valor de estos bits es “00”.
- **Tipo**: los dos bits a continuación ($bit\ 2$ y $bit\ 3$) definen el tipo de trama. El tipo “00” se destina a tramas de administración, el tipo “01” se refiere al grupo de las tramas de control, el tipo “10” se usa para distinguir las tramas de datos, mientras que el tipo “11” es de uso reservado.
- **Subtipo**: los siguientes cuatro bits ($bit\ 4$, $bit\ 5$, $bit\ 6$ y $bit\ 7$), distinguen el subtipo dentro de un tipo de trama, sirviendo para destacar diferentes tramas. Así, por ejemplo, el tipo “01”, subtipo “1011”, define a una trama de control RTS, en tanto que el subtipo “1100”, marca una trama de control CTS. Todos los tipos y subtipos definidos se presentan en la Tabla 7.2.

- **A DS, DE DS**: (*bit* 8 y *bit* 9) definen la dirección de los mensajes en relación al sistema de distribución DS. Si alguno de los dos bits es ajustado a "1", se opera en modo de Infraestructura, indicándose si la trama se dirige hacia el AP o proviene de éste. Parte del mecanismo de entrega es el medio de la red Ethernet, donde debe existir alguna manera de relacionar cada AP con las estaciones móviles asociadas a los mismos, y de esta manera poder entregar las tramas correctamente. En este caso, los AP ofician de puente entre la LAN cableada y la WLAN.
  El mecanismo de asociación permite reconocer dónde se encuentra cada móvil. A su vez, existe un mecanismo de aviso de asociación entre puntos de acceso, que se implementa a través de un protocolo propietario IAPP. El hecho de que el protocolo no se encuentre estandarizado, genera la restricción de tener que utilizar AP del mismo fabricante al momento de instalar un ESS.
  En cuanto a los otros valores posibles de estos bits, en todas las tramas de administración y control y en el caso de redes *ad hoc*, los bits se ajustan en "00".
  En el caso particular de puente inalámbrico, donde se configura una WLAN entre dos AP a modo de enlace entre redes, ambos bits se ajustan en "11". Más adelante, en este capítulo, se presentan ejemplos referidos al uso de este campo.
- **Más fragmentos**: (*bit* 10) es la bandera que sirve para indicar, en una transmisión fragmentada, la existencia de otros fragmentos. Ajustada en "1", quiere decir que existen fragmentos adicionales, que se transmitirán luego del presente.
- **Reintentar**: (*bit* 11) es el bit de retransmisión. Con esta bandera, el transmisor avisa al receptor que está enviando una trama retransmitida debido a algún evento de error. De este modo, se evitan las tramas duplicadas.
- **Administración de Encendido**: (*bit* 12) es el indicador de modo de Ahorro de Potencia (PS, Power Save) luego de finalizar una Operación Atómica. Es la bandera que levantan las estaciones móviles para avisar que se pasan a modo de ahorro de potencia.
- **Más Datos**: (*bit* 13) indica almacenamiento de datos en el AP, que se está comunicando con las unidades móviles que utilizan el modo de ahorro de potencia. Esta bandera es levantada por el AP para avisar a las móviles que han entrado en modo PS que tiene más tramas almacenadas.
- **WEP**: (*bit* 14) indica el uso del algoritmo de cifrado de datos WEP.
- **Orden**: (*bit* 15) este bit ajustado a "1" indica que la información debe entregarse en orden estricto.

En la Tabla 7.2 se ofrece un resumen de los tipos y subtipos de tramas definidos para la MAC 802.11, con una breve explicación en cada caso. Más adelante, se desarrollará con mayor detalle cada una de las tramas de administración y control en el modo con contienda DCF. La presentación

resumida de la Tabla 7.2 permite adelantar al lector la variedad de funciones administrativas agregadas a las tramas MAC 802.11

Algunas de las tramas de administración permiten a las estaciones móviles asociarse a una red WLAN, alejarse de la misma y volver a asociarse al mismo AP o a otro. También permiten a las móviles buscar de manera activa una WLAN para asociarse, para autenticarse frente al AP o para anunciarle que se va a pasar al modo de ahorro de potencia. Otras tramas son propias del AP y se utilizan para anunciar la WLAN y sus capacidades a las móviles que desean asociarse, colaborando también en cuestiones relacionadas con el sincronismo y el almacenamiento de tramas para aquellas que ya están asociadas.

En particular, más adelante estudiaremos con detalle los campos de la trama de Beacon, que es una de las más importantes para el correcto funcionamiento de la WLAN y ofrece la posibilidad de que las estaciones móviles entren en el modo de ahorro de potencia.

Tabla 7.2 - Tipos y Subtipos de Tramas 802.11

Tipo	***Subtipo***	***Significado***	***Utilidad***
00	*0000*	Requerimiento de Asociación	La móvil lo transmite para asociarse a una WLAN.
00	*0001*	Respuesta de Asociación	En respuesta a la trama anterior.
00	*0010*	Requerimiento de Re-asociación	La móvil lo transmite cuando se aleja del AP al que se encuentra asociada, entrando en el alcance de otro AP.
00	*0011*	Respuesta de Re-asociación	Respuesta a la trama anterior.
00	*0100*	Requerimiento de Sondeo	Las estaciones móviles usan estas tramas cuando buscan redes WLAN para asociarse.
00	*0101*	Respuesta de Sondeo	Si se encuentra una red con la trama de Requerimiento de Sondeo, la respuesta se realiza con esta trama.
00	*1000*	Trama Beacon	Contiene toda la información sobre la WLAN. Estas tramas se transmiten de manera periódica, para anunciar la presencia de una WLAN. En una red Infraestructura estas tramas las transmiten los AP. En una red tipo IBSS, la generación de tramas Beacon es una funcionalidad distribuida entre las móviles que conforman la red.
00	*1001*	ATIM	Anuncio de Mapa de Indicación de Tráfico IBSS. Cuando una móvil en una red *ad hoc* tiene tramas almacenadas para otra estación que se encuentra en modo de baja potencia, envía una trama ATIM

Tipo	***Subtipo***	***Significado***	***Utilidad***
			durante el período de entrega para notificar al receptor que tiene datos almacenados.
00	*1010*	Des-asociación	Para des-asociarse de una WLAN.
00	*1011*	Autenticación	Para autenticarse ante una WLAN.
00	*1100*	De-autenticación	Para des-autenticarse.
01	*1010*	PS Poll	Cuando una estación despierta, luego de haber estado en modo PS, transmite una trama PS Poll dirigida al AP, para solicitarle todas las tramas que éste haya almacenado para ella, mientras estaba en modo de ahorro de potencia.
01	*1011*	RTS	Trama especial para mitigar el problema de nodo oculto. Emitida por una móvil hacia el AP, solicitando iniciar una transmisión de datos.
01	*1100*	CTS	Contestación a RTS por parte del AP. El mecanismo se dispara cuando el tamaño de las tramas supera un valor configurable por el administrador.
01	*1101*	ACK	Trama emitida en respuesta a aquellas tramas *unicast* que han arribado sin errores.
01	*1110*	CF-End	Cuando finaliza el período libre de contienda CF, el AP emite una trama se este tipo para liberar a las estaciones de las reglas de acceso del modo PCF. Entonces empieza el servicio basado en contienda DCF.
01	*1111*	CF-End+ACK	Es igual a la trama de CF-End pero también transporta el ACK para la anterior trama de datos transmitida.
10	*0000*	Datos	Trama de datos.
10	*0001*	Datos + CF-ACK	Esta trama combina las transmisiones de datos con un ACK en el modo PCF.
10	*0010*	Datos+ CF-Poll	Esta trama puede ser enviada solamente por el AP durante el período libre de contienda. Se utiliza para transmitir datos a una estación móvil y requerir una trama pendiente de la móvil.

Tipo	***Subtipo***	***Significado***	***Utilidad***
10	0011	CF-Ack+CF-Poll	Esta trama es un ACK de la última trama del AP y requiere una trama protegida para ingresar en la lista sondeo o *polling*.
10	0100	NullData	Se trata de tramas sin datos que se utilizan para indicar al AP un cambio de estado, en el caso de que la móvil vaya a entrar en modo PS y no tenga datos para transmitir. De este modo, las móviles le piden al AP que comience a almacenar tramas mientras ellas van a modo ahorro de potencia.
10	0101	CF-ACK	Es la trama ACK en el modo PCF cuando no hay datos para transmitir.
10	0110	CF-Poll	Durante el período libre de contienda, las estaciones pueden transmitir solamente si el AP solicita la transmisión con la trama CF-Poll. Cada trama CF-Poll es una licencia de transmisión. Las tramas múltiples pueden ser transmitidas únicamente si el AP envía múltiples peticiones *poll*.
10	0111	Data+CF-Ack+CF-Poll	Esta trama reúne la transmisión de datos, el ACK de otra trama transmitida y la posibilidad de sondeo o *polling*, todo junto en una misma trama para lograr mayor eficiencia. La transmite el AP.

Las tramas que llevan las siglas CF en la Tabla 7.2, se corresponden con el modo libre de contienda, sobre el que no nos explayaremos. Como ya se ha aclarado, el modo PCF no ha sido ampliamente implementado, aunque es posible que productos basados en PCF lleguen al mercado en algún momento. Su mayor ventaja es la combinación de varias funcionalidades en una sola trama.

Cabe aclarar que las tramas con destino *broadcast* o *multicast* en la Dirección 1 y las tramas de administración con dirección de *broadcast* en este mismo campo (Beacon, Requerimiento de Sondeo e IBSS ATIM) no pueden fragmentarse, no requieren ACK y no se pueden retransmitir. Se trata de la transmisión de una sola trama según las reglas de control de acceso. Por ese motivo, el campo NAV en dichas tramas se coloca en 0. Se trata de transmisiones no confiables.

7.4.2 Campo de Duración/ID

El campo de Duración/ID es el siguiente al de campo de Control de trama. El principal propósito de este campo es actualizar el reloj asociado al NAV de otras estaciones. También las tramas de de administración PS-Poll usan este campo como un identificador de asociación. En los períodos libres de contienda, en cambio, el campo se usa como indicador de comienzo del proceso PCF. La Fig. 7.11 ilustra las tres posibilidades mencionadas.

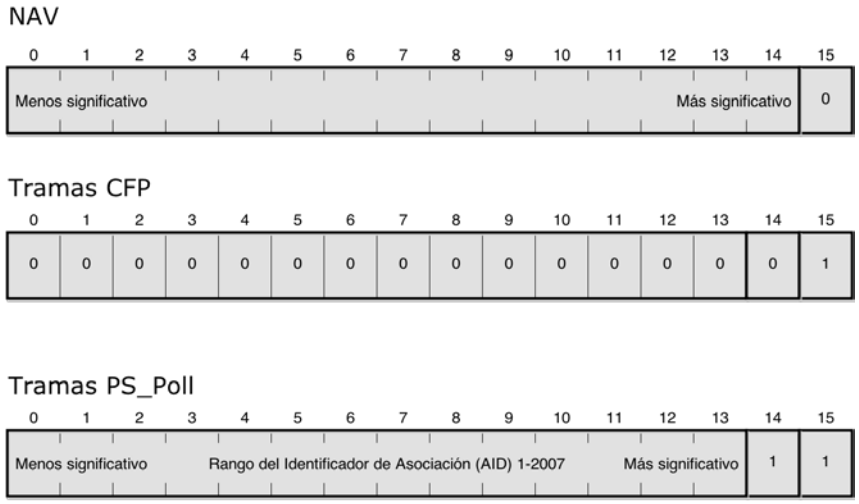


Figura 7.11 - Campo de Duración/ID.

Cuando el bit 15 es "0", el campo de Duración/ID tiene el significado de una duración en el tiempo y se usa para ajustar el valor del NAV. El valor dentro del campo representa el número de microsegundos que se espera que el medio permanezca ocupado durante la transmisión. Todas las estaciones móviles que intentan acceder al medio, lo deben escuchar para actualizar su propio NAV a este valor. Así configuran el tiempo en que deberán bloquearse con respecto al acceso.

En los períodos libres de contienda, el bit 15 es "1" y el bit 14 es "0". Todos los otros bits son también "0", con lo que el campo completo se interpreta como un NAV de valor 32.768. Esto permite a las estaciones que no recibieron el mensaje de la trama Beacon que anuncia el período libre de contienda, que actualicen su propio NAV con un valor lo suficientemente grande como para evitar provocar interferencia en el período libre de contienda. Cabe aclarar que la trama de Beacon se transmite por *broadcast*, siendo su transmisión no confiable ya que no queda sujeta a la operación con ACK.

En el tercer caso, los $bits$ 14 y 15 se ajustan ambos a "1". Este valor define un modo Sondeo por Ahorro de Potencia (PS-Poll). Cuando las estaciones móviles entran en modo PS, deben despertar periódicamente para enterarse si

tienen información almacenada para ellas en el AP. Cuando una estación se re-enciende, luego de un período de apagado para reducción de consumo, envía una trama PS-Poll para recuperar las tramas almacenados en el AP. En estas tramas, el campo Duración/ID tiene el significado de un identificador de asociación, AID (Association ID), que determina a cuál de los servicios básicos BSS pertenece la estación. Se trata de un número entre 1 y 2007 que la móvil recibe al asociarse, es el número con que el AP la identifica.

7.4.3 Campos de Direcciones

Las direcciones 802.11 son todas de 48 $bits$, como en Ethernet. Curiosamente, a diferencia de la MAC IEEE 802.3 que sólo posee dos campos de direcciones, uno para la fuente de la información y otro para el receptor de la misma, en la MAC IEEE 802.11 pueden aparecer hasta cuatro campos de direcciones. Tres de estos campos se presentan de manera contigua y el cuarto aparece luego del campo de Control de Secuencia. Esto se debe a que, la mayoría de las veces, sólo se usan tres direcciones. La cuarta se reserva para un caso especial.

En el estándar IEEE 802.11, distingue tres protagonistas de la comunicación:

- **Transmisor**: es la estación o AP que baja la trama al medio inalámbrico, aunque no necesariamente es el que la genera.

- **Receptor**: puede ser un intermediario, por ejemplo el AP. En todo caso, es el receptor de la trama en el medio inalámbrico.

- **Destino**: es el que procesa la trama para pasarla a niveles superiores. La dirección de destino es el identificador IEEE MAC que corresponde a la estación que destinará la información a las capas superiores para su procesamiento final.

La dirección de fuente de una trama identifica la fuente generadora de la información, Por tratarse de una dirección única, posee el primer bit menos significativo del primer byte en "0", como en el caso Ethernet. La dirección del receptor es el identificador IEEE MAC de 48 $bits$ que indica la estación inalámbrica que va a procesar la trama en recepción.

Por ejemplo, en el caso de transmitirse tramas a un AP conectado a una red cableada, en la que podría existir una estación que reciba la información que se envía, la dirección de receptor es la que identifica a la placa inalámbrica del AP, mientras que la dirección de destino identifica al nodo final a quien se envía la información en la red cableada.

Como en el caso Ethernet, el bit menos significativo del primer byte, cuando es "0", significa que la dirección es la de una estación particular o *unicast*. Si el bit mencionado es "1", la dirección es de tipo *multicast*. Por último, una

dirección con todos los bits en "1" es una dirección de *broadcast* y se transmite a todas las estaciones de la red.

En general, la Dirección 1 es utilizada para el receptor de la trama, la Dirección 2 se destina para el transmisor, y la Dirección 3 se requiere para filtrado en el receptor.

En un intercambio normal de tramas, se hará referencia en alguna de las direcciones al identificador del BSS. Esta característica es debido a que podría haber varias redes inalámbricas en la misma área, aunque las estaciones se asocian a una de ellas por vez. En la redes de Infraestructura, las estaciones reconocen un Identificador de Servicio Básico (BSSID, Basic Service Set Identifier) que se corresponde con la identificación de la interfaz inalámbrica del correspondiente AP, es decir que se trata de la MAC de la placa inalámbrica del AP. En redes *ad hoc* este número se genera aleatoriamente al crearse la WLAN.

La Tabla 7.3 presenta las relaciones entre los bits To DS y From DS del campo de Control de la trama, con el significado de las direcciones. La primera línea de la Tabla 7.3 representa el caso de redes *ad hoc*, donde se usan las Direcciones 1, 2 y 3, tal como se aprecia. Como en este tipo de redes los interlocutores son todas estaciones móviles, las Direcciones 1 y 2 se consideran directamente como las de Destino (DA) y Fuente (SA) de la información. La Dirección 3 es un filtro, el número aleatorio que identifica a la red.

La segunda y tercer línea de la Tabla 7.3, representan el caso de la comunicación en redes de Infraestructura. La segunda línea trata el caso de una trama generada por una estación móvil, que debe pasar por el AP obligatoriamente. Por este motivo, la Dirección 1 es la propia MAC inalámbrica del AP (BSSID) que actúa como receptor y se usa como identificador del BSS. La Dirección 2 es la MAC de la estación móvil que genera la trama (SA), que se puede pensar como fuente o transmisor. La Dirección 3 es la de destino (DA), que podría ser otra estación móvil o una fija en la red cableada. Un ejemplo de este caso se presenta en la Fig. 7.12. Claramente se observa la diferencia entre receptor (RA) y destino (DA), y la coincidencia entre fuente (SA) y transmisor (TA).

Tabla 7.3 - Relación entre los bits To DS y From DS con las Direcciones MAC 802.11.

Función	**To DS**	**From DS**	**Dirección 1**	**Dirección 2**	**Dirección 3**	**Dirección 4**
IBSS	0	0	Destino (DA)	Fuente (SA)	BSSID	No usada
Hacia el AP (Infraestructura)	1	0	BSSID	Fuente (SA)	Destino (DA)	No usada
Desde el AP (Infraestructura)	0	1	Destino (DA)	BSSID	Fuente (SA)	No usada
Puente Inalámbrico WDS	1	1	Receptor	Transmisor	Destino (DA)	Fuente (SA)

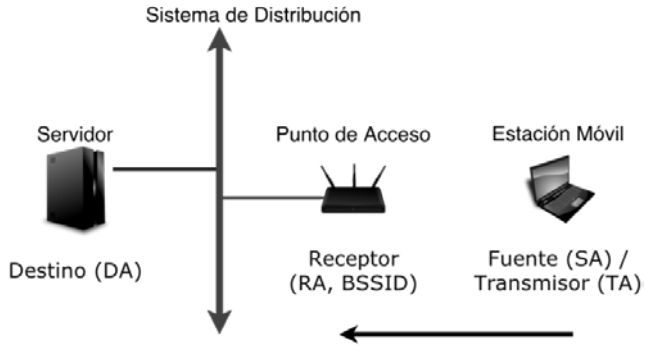


Figura 7.12 - Comunicación hacia Sistema de Distribución (To DS).

La tercera línea de la Tabla representa el caso de comunicación desde el AP a alguna de las estaciones móviles asociadas. En este caso, la Dirección 1 es la de la estación móvil, destino y receptor de la trama. La Dirección 2 es la MAC del AP que vuelca la trama al medio inalámbrico. La Dirección 3 es la de la fuente de esta trama, que podría ser otra móvil o una estación fija en la red cableada. La situación se representa en la Fig. 7.13. Claramente se observa la coincidencia entre receptor (RA) y destino (DA), y la diferenciación entre fuente (SA) y transmisor (TA).

La cuarta línea de la Tabla se refiere a un caso muy especial, conocido Puente Inalámbrico (WDS, Wireless Distribution System), graficado en la Fig. 7.14. En este caso, los dos bits del campo de Control a los que se ha hecho referencia antes (TO DS, FROM DS), están en "1". La comunicación inalámbrica es entre puntos de acceso. Se ha representado el caso de un cliente en la red cableada, detrás de uno de los AP, y un servidor en la otra red cableada, detrás del otro AP. Los AP se comunican entre sí utilizando el protocolo 802.11. En este caso se utilizan las cuatro direcciones porque hay cuatro protagonistas de la comunicación. La Dirección 1 es la del AP receptor de la trama en el medio inalámbrico (RA), la Dirección 2 es la del AP transmisor de la trama en ese mismo medio (TA). La Dirección 3 es la del destino de la trama, en este caso el servidor (DA). Por último, la Dirección 4 es la del cliente, que en este ejemplo oficia de fuente o generador de la trama en tránsito (SA). En este caso se utilizan los cuatro campos de direcciones.

WDS puede funcionar también en un modo de Repetidor Inalámbrico, en el que los AP se pueden comunicar entre sí y con estaciones móviles, generalmente con el propósito de extender la red. Una de las mayores desventajas de este tipo de despliegue es que la velocidad se reduce a la mitad, por tener que repartirse en la funcionalidad de repetidor que cumple el AP. No es una forma de utilización todavía estandarizada, pero están empezando a aparecer en el mercado dispositivos inalámbricos para redes tipo malla o *mesh*, al mismo tiempo que existe un grupo de la IEEE trabajando en este sentido.

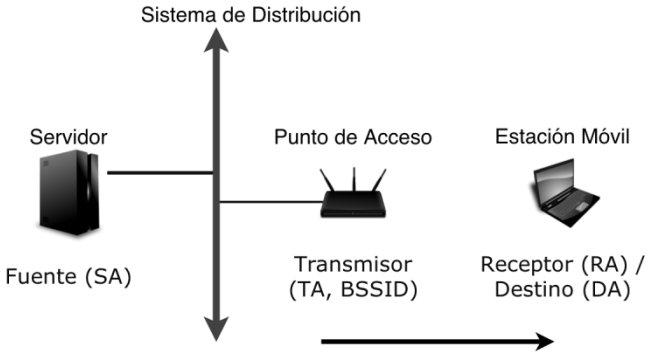


Figura 7.13 - Comunicación desde el Sistema de Distribución (From DS).

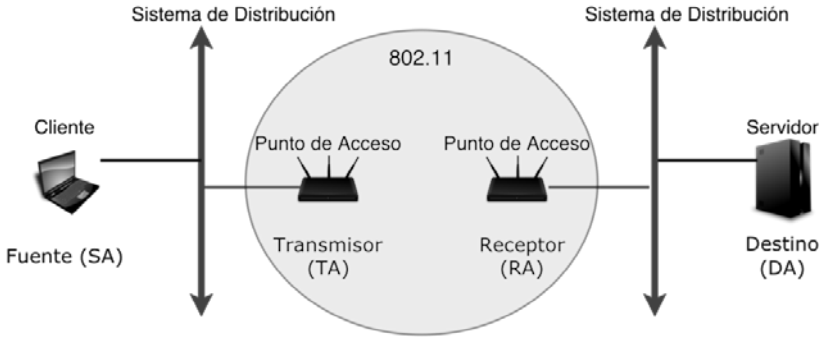


Figura 7.14 - Comunicación puente inalámbrico.

7.4.4 Campo de Control de Secuencia

El campo de Control de Secuencia se usa para la fragmentación y para la detección y descarte de tramas duplicadas. Está conformado por 16 $bits$, tal como se presenta en la Fig. 7.15. Los 12 $bits$ más significativos se usan para la numeración de las secuencias (módulo 4096) y los 4 $bits$ restantes se destinan a la numeración del fragmento.

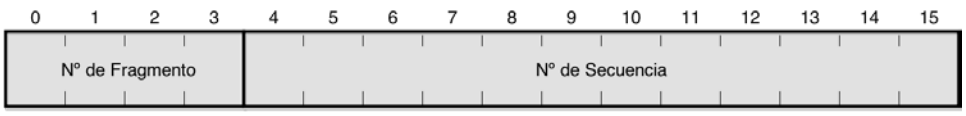


Figura 7.15 - Campo de Control de Secuencia.

Cada trama recibe un número de secuencia. Este número comienza en cero, y se incrementa en uno con cada trama posterior, para numerarla antes de ser enviada a la MAC para su transmisión. Cuando se realiza la fragmentación de tramas de niveles más altos, cada fragmento tiene el mismo número de secuencia. Cuando se produce una retransmisión, el número de secuencia no se cambia. Lo que diferencia a los fragmentos en una trama fragmentada es el número de fragmento, que se inicia en cero y se incrementa en uno con cada fragmento siguiente. El bit del campo de Control indicador de la existencia de más fragmentos se ajusta en "1" cuando la estación transmisora se dispone a enviar más fragmentos. En el último fragmento su valor es "0".

7.4.5 Campo de Datos y Campo de Redundancia Cíclica

El cuerpo de la trama, o campo de Datos, se encuentra limitado en 802.11 a una longitud máxima de que depende de la capa física en cuestión. Debe tener la posibilidad de expandirse en el caso de encapsulado por seguridad, para agregar información de encabezado adicional.

Las tramas de datos presentan variantes según que el servicio sea basado en contienda o sin contienda. Para mejorar la eficiencia, las tramas del servicio libre de contienda (CF) pueden especificar subtipos, permitiéndose el reconocimiento de otras tramas junto con la propia trama de datos.

Las tramas del subtipo Datos sólo se transmiten en los períodos de acceso basados en contienda. Su propósito es simplemente trasladar datos entre estaciones.

Las tramas Null son tramas sin datos, que anuncian cambios de estado de las móviles cuando las mismas deban anunciar su paso a modo PS y carecen de datos para transmitir

El campo de Control de Errores, FCS, está formado por los bits de chequeo de un código cíclico, igual que en el estándar IEEE 802.3. Todos los encabezados y el campo de datos están afectados por este control de errores. A pesar de usar el mismo método que en 802.3, como los encabezados son diferentes, el FCS debe ser calculado nuevamente en los AP cada vez que se pasan tramas entre la WLAN y la Ethernet.

En 802.3, cuando el receptor recalcula el FCS, contrastándolo con el CRC recibido, si coinciden, se acepta la trama y se la pasa a capas superiores. En caso contrario, se descarta. En 802.11, la recepción de tramas *unicast* con chequeo correcto se completa con la transmisión de una trama ACK**.** En caso de errores, se procede a su descarte. No hay tramas especiales de NACK, sino que el transmisor admite la retransmisión cuando se vence un tiempo asociado a la trama en espera del ACK.

7.5 Tramas de Control IEEE 802.11

Estas tramas asisten el trabajo de entrega de las tramas de datos, administrando el acceso al medio inalámbrico y proveyendo a la subcapa MAC con funciones de confiabilidad. Se trata de tramas de tipo “01” que comparten el mismo campo de Control. Excepto por el campo tipo o el campo subtipo, muy frecuentemente todos los bits se ajustan en “0”.

Como el sistema de distribución no envía ni recibe este tipo de tramas, ambos bits ToDS y FromDS, se ajustan en "0". A su vez, como estas tramas no se fragmentan ni se colocan en cola para retransmisión, el bit de más fragmentos y el bit de retransmisión también se ajustan en “0”. En cuanto al bit de indicador de más datos, sólo se usa en las tramas de administración y de datos, por lo que su valor es “0” en las tramas de Control. Como las tramas de Control no pueden ir cifradas y se usan como parte de intercambios atómicos, tampoco pueden transmitirse fuera de orden, por lo que los bits de cifrado y el bit de orden se ajustan a “0”. El único bit que podría no ser nulo es el bit de PS, ya que se ajusta al estado en el que se encontrará el transmisor luego de la conclusión del intercambio actual de tramas.

Existen cuatro tramas de Control: RTS, CTS, ACK y PS-Poll:

- **RTS**: las tramas de RTS se usan para ganar acceso al medio para el caso de transmisiones de tramas *unicast* que superan un umbral configurado por el administrador. El formato de las tramas RTS se presenta en la Fig. 7.16, con un detalle del campo de Control. Estas tramas no llevan datos y sólo precisan dos direcciones, para identificar al que recibe la trama como dirección destino (DA), y al que la genera como dirección fuente (SA). El campo de duración se mide en μseg y es el valor del tiempo que falta para completar la Operación Atómica: duración de la trama CTS, duración de la trama de Datos, duración del ACK y duración de tres tiempos SIFS intermedios entre las mencionadas.

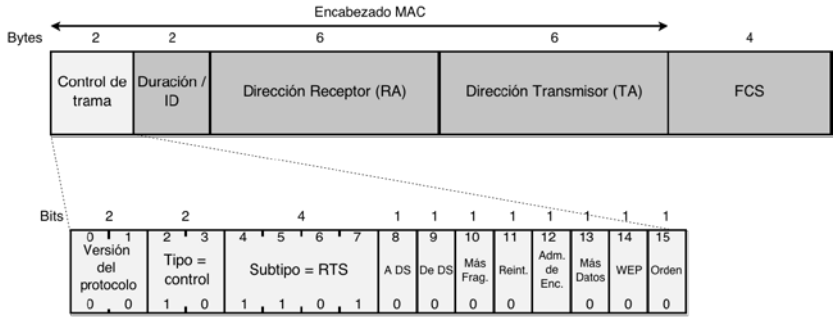


Figura 7.16 - Trama RTS.

- **CTS**: tal como se aprecia en la Fig. 7.17, la trama CTS presenta una sola dirección, la dirección destino DA, que es la dirección SA de la trama

RTS. El campo de Duración es el correspondiente de RTS, al que se le resta la Duración de la propia trama CTS y un intervalo SIFS.

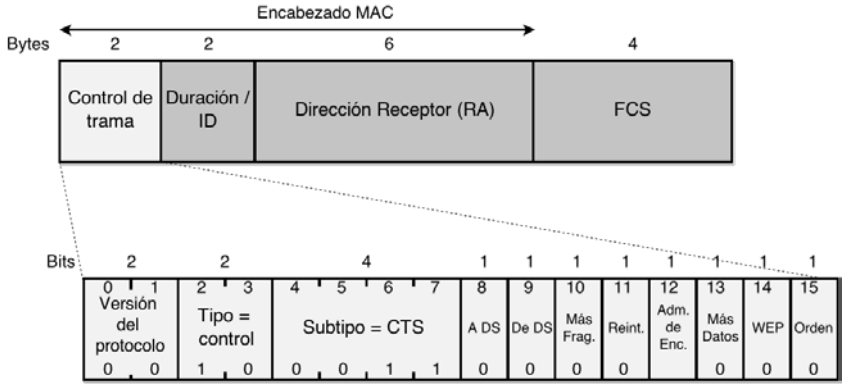


Figura 7.17 - Trama CTS.

- **ACK**: esta trama tiene la misma longitud que la de CTS. En la dirección DA se copia la dirección SA de la trama cuyo reconocimiento se está haciendo. El campo de Duración lleva un valor nulo en el caso de tratarse de la trama final de una Operación Atómica. Si la transmisión es fragmentada y la trama de ACK se corresponde al reconocimiento de un fragmento intermedio, entonces el campo de Duración se ajustará de manera de cubrir el tiempo que lleve transmitir el siguiente fragmento y su ACK, considerando además los tiempos intermedios.

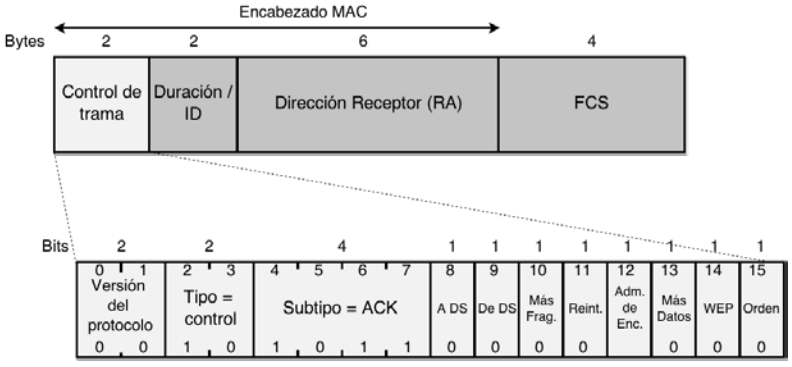


Figura 7.18 - Trama ACK.

- **PS-Poll**: en IEEE 802.11 se intenta maximizar el tiempo de vida de las baterías de los dispositivos móviles cerrando el transceptor de radio. Esta situación se suele referenciar diciendo que los dispositivos “duermen” periódicamente. En esos períodos, los puntos de acceso colocan en

buffers las tramas *unicast* que lleguen para las estaciones que duermen. A su vez, de manera periódica, los AP anuncian el almacenamiento de estas tramas a través de un campo especial de las tramas Beacon. Cuando una estación móvil despierta, re-encendiendo el transceptor, comienza a recibir tramas Beacon y debe revisar el campo especial de dichas tramas para enterarse si el AP le ha almacenado información mientras se encontraba en modo ahorro de potencia. De verificarse esta situación, la estación móvil debe transmitir una trama PS-Poll para solicitarle al AP la entrega de las tramas almacenadas.
El formato de la trama se presenta en la Fig. 7.19. Aparte del detalle del campo de Control, se puede apreciar que el campo de Duración en estas tramas lleva el Identificador de Asociación AID. Se trata de un número entre 1 y 2007 que la móvil recibe del AP en el proceso de asociación. En la trama PS-Poll, luego del AID aparece la dirección del AP, referenciada como BSSID, y la dirección del transmisor de la trama. Todos los que reciben la trama de PS-Poll, actualizan el valor de NAV automáticamente a $1\ SIFS + 1\ ACK$, para que el AP pueda contestar sin colisión.

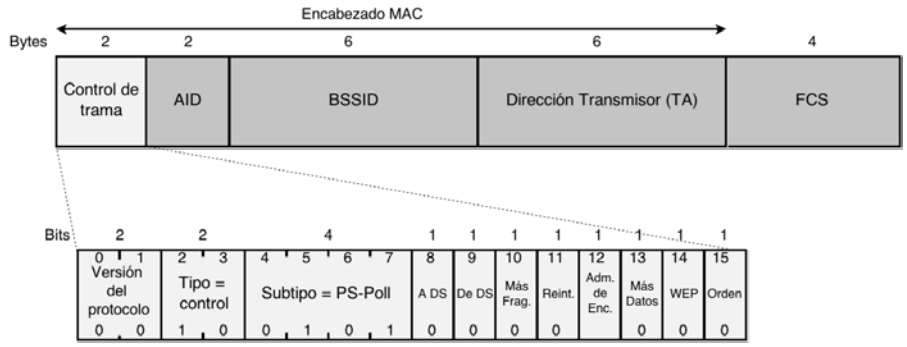


Figura 7.19 - Tramas PS-Poll.

El detalle de la existencia de un NAV implícito se refiere a las dos posibilidades que pueden suceder en el caso que el AP reciba la trama de PS-Poll.
En la Fig. 7.20 se observa el caso en que el AP está disponible para satisfacer el requerimiento, ante la solicitud de las tramas almacenadas. La trama PS-Poll contiene el AID de la estación en el campo del NAV. El propósito de esta trama es que el AP verifique si todavía tiene tramas en el *buffer* para esa móvil. Todas las estaciones que escuchen la trama de PS-Poll deben actualizar el NAV al valor implícito. Aunque dicho NAV es muy corto para la transmisión de la trama de datos, el AP adquiere el medio y las demás estaciones difieren, realizándose la Operación Atómica como indica la Fig. 7.20. La transmisión de los datos actualiza el NAV para proteger la transmisión del ACK final.

El otro caso que se puede presentar se grafica en la Fig. 7.21, donde se puede apreciar que el AP responde a la trama PS-Poll con un ACK, probablemente por encontrarse ocupado en otras tareas. Un tiempo más tarde, el AP deberá competir por el medio para poder responder con las tramas de datos que correspondan a este requerimiento. En ese caso, la estación móvil solicitante no puede volver a dormir hasta recibir las tramas que el AP guarda para ella o hasta que reciba una trama Beacon en la cual su propio bit en el Mapa Indicador de Tráfico (TIM, Traffic Indication Map) le indique que en el AP no hay tramas para ese AID.
Podría darse el caso de que haya más de una trama almacenada en el *buffer* del AP para la móvil en cuestión. Al asociarse cada estación, las móviles anuncian el tiempo durante el que irán a dormir habitualmente. El AP acepta la asociación si puede asegurarles cierta cantidad de *buffer* disponible para el almacenamiento de tramas durante ese tiempo. Una vez agotado este tiempo, el AP puede descartar las tramas que no haya podido entregar al dispositivo móvil.

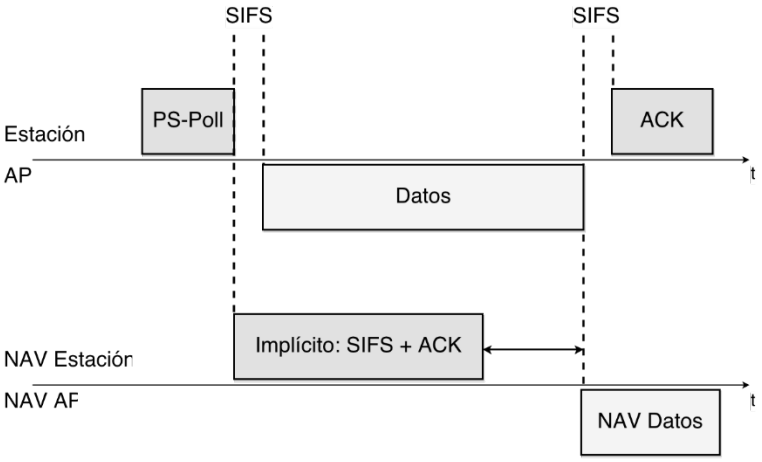


Figura 7.20 - Respuesta a PS-Poll no diferida.

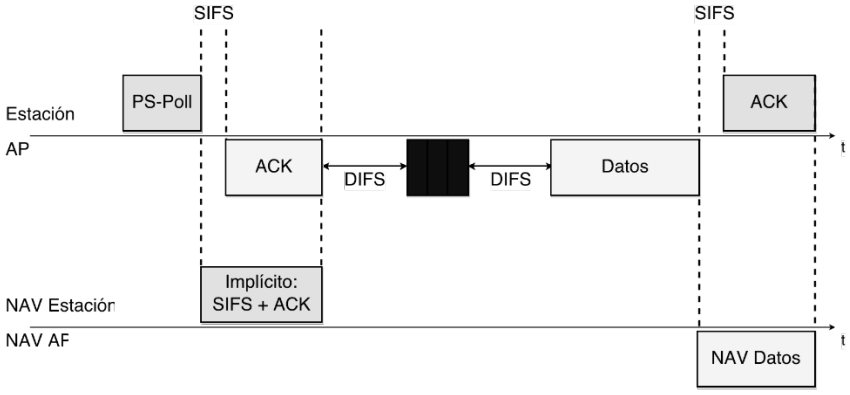


Figura 7.21 - Respuesta a PS-Poll diferida.

7.6 Tramas de Administración IEEE 802.11

La funcionalidad administrativa es un componente muy importante de la MAC 802.11. Existen servicios que exigen la definición de tramas que no existen en una red tipo Ethernet cableada. Por ejemplo, conectar un equipo a una red cableada se traduce en tener autorización del administrador para enchufar un conector en el lugar apropiado. En una red inalámbrica esta situación es un poco más complicada. Las estaciones móviles deben primero localizar una red compatible, luego se deben autenticar a modo de autorización para conectarse a la WLAN y, finalmente, deben asociarse con el AP para poder comenzar a intercambiar datos. Todos estos pasos se asocian a diferentes tipos de tramas.

Las tramas de administración IEEE 802.11 comparten una estructura que se muestra en la Fig. 7.22. El encabezado es el mismo para todas pues no depende del subtipo. Como se observa en la figura, las tramas de administración transportan elementos de información. Se trata de datos etiquetados, con una estructura definida, que sirven para comunicar información entre los distintos actores de la comunicación. Algunos de estos elementos son campos de longitud fija y otros tienen longitud variable. Estos últimos se etiquetan con un campo de tipo y otro de longitud. IEEE 802.11 fija el orden de estos elementos, pero no todos son obligatorios. De este modo es posible definir nuevos elementos de información de manera compatible a los más antiguos, aunque las implementaciones más viejas no los reconocerán.

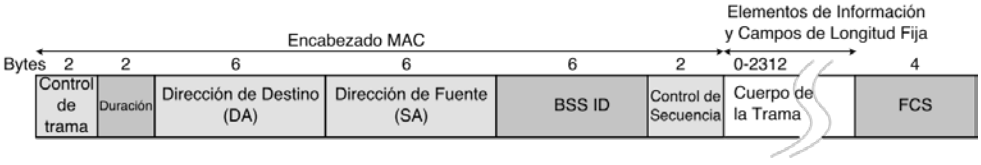


Figura 7.22 - Trama de Administración

En las tramas de administración, la Dirección 1 sigue siendo la de destino. Algunas tramas de este tipo se usan para mantener ciertas propiedades dentro de un mismo BSS. Luego de recibir una trama de administración, las móviles deben inspeccionar el BSSID. La lectura de este campo es usada para limitar el efecto de tramas administrativas de *broadcast* y *multicast*, pues sólo aquellas reconocidas como pertenecientes al propio BSSID de la móvil, serán entregadas a la capa de administración MAC para su procesamiento. La única excepción a esta regla es la trama de Beacon, usada para anunciar redes IEEE 802.11. Recordar que los AP asignan la MAC de su interfaz inalámbrica al BSSID, en tanto que en redes *ad hoc* se usa un número generado aleatoriamente al momento de la creación de la red.

Es oportuno aclarar que las tramas de administración usan el campo de Duración de igual modo que el resto de las tramas, respetando los tiempos en el caso de Operaciones Atómicas. También pueden llevar un valor “0” en dicho

campo en el caso de tramas de *broadcast* o *multicast* ya que estas últimas no se asocian a un ACK.

7.6.1 Trama Beacon

Las tramas Beacon anuncian la existencia de la WLAN y son muy importantes para las tareas de mantenimiento de la red. Se transmiten a intervalos regulares para que las móviles que se encuentren en cercanía de la WLAN puedan enterarse de su existencia, identificarla y conocer los parámetros necesarios para incorporarse a la misma. En redes de Infraestructura, el AP es el responsable de transmitir estas tramas, quedando el BSS definido por el área dentro de la cual estas tramas pueden recibirse.

La Fig. 7.23 muestra la mayoría de los campos presentes en una trama Beacon, así como el orden en que el estándar exige en cuanto a su disposición. No todos los elementos se presentan en todas las tramas de este tipo, ya que algunos son excluyentes entre sí y la presencia de otros depende de cómo se haya configurado la red. Por ejemplo, los elementos Conjunto de Parámetros FH y Conjunto de Parámetros DS no pueden ir nunca juntos en una trama Beacon porque se refieren a la capa física subyacente, que sólo puede estar trabajando en una de estas dos modalidades de espectro esparcido.

Otro elemento opcional es el Conjunto de Parámetros CF, que se presenta solamente cuando los AP tienen capacidades PCF. Por su parte, el elemento TIM se usa únicamente en tramas Beacon generadas por los AP, ya que estos son los únicos elementos capaces de almacenar tramas.

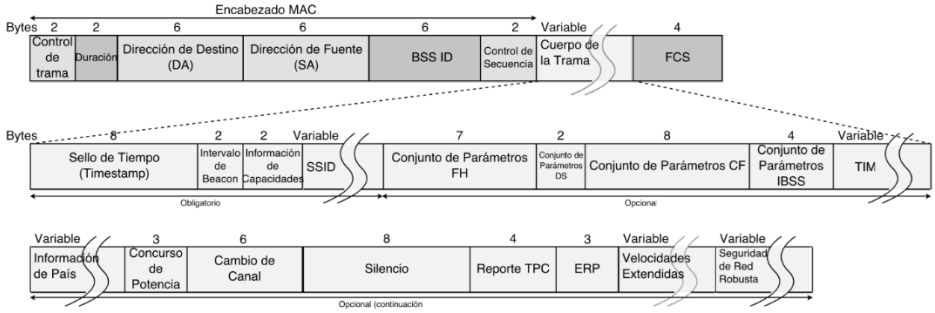


Figura 7.23 - Trama Beacon.

Para dar un ejemplo práctico, primero se definirán los elementos fijos y los elementos de información presentes en una trama real, levantada con un software tipo *sniffer*, que se presenta en la Fig. 7.24.

El primer elemento fijo que obligatoriamente debe cargar una trama Beacon es el **Sello de Tiempo (Timestamp)**, de 8 $bytes$. Este elemento colabora con la funcionalidad de sincronización que se agrega a nivel MAC, ya que las

estaciones, luego de hallar una WLAN y asociarse a ella, deben contar con algún mecanismo para permanecer en contacto con la red, aún después de salir del modo ahorro de potencia. Para ello, en el estándar se define la Función de Sincronismo en el Tiempo (TSF, Timing Syncrhonization Function) que cada estación mantiene de manera individual. Se trata de un reloj local sincronizado con el TSF de las demás estaciones en el mismo BSS. La trama de Beacon anuncia su valor periódicamente en el elemento Sello de Tiempo. El AP, guardián del tiempo maestro, transmite en este elemento el número de microsegundos que ha permanecido activo. Como se trata de un valor creciente transportado en un campo de $64\ bits$, si llegara al máximo, debería comenzar de nuevo en cero, aunque para ello tendrían que transcurrir $580.000\ años$. En redes *ad hoc*, la funcionalidad de sincronización es distribuida.

IEEE 802.11Trama de Administración Inalámbrica

Parámetros Fijos (12 bytes)

Sello de Tiempo (timestamp):	0x0000007D1C552122	
Intervalo de Beacon:	0.102400 [segundos]	
Información de Capacidades:	0x0431	
	.... …. …. …1	Capacidades ESS: el Transmisor es un AP
	…. …. …. ..0.	Estado IBSS: el Transmisor pertenece a una BSS
	…. ..0. …. 00..	Capacidades de participación CFP: sin coordinador de punto en AP (0x0000)
	…. …. …1 ….	Privacidad: AP/Estación soporta WEP
	…. …. ..1. ….	Preámbulo corto: permitido
	…. …. .0.. ….	PBCC: modulación PBCC no permitida
	…. …. 0… ….	Agilidad de Canal: no en uso
	…. …0 …. ….	Administración de Espectro: no requerido
	…. .1.. …. ….	Tiempo de Ranura Corta: en uso
	…. 0… …. ….	APSD: no implementado
	..0. …. …. ….	DSSS-OFDM: no permitido
	.0.. …. …. ….	ACK de Bloque Retardado: no implementado
	..0. …. …. ….	ACK de Bloque Inmediato: no implementado

Parámetros etiquetados (54 bytes)

Conjunto de Parámetros SSID:	"RED_LIC"
Número de Etiqueta:	0 (Conjunto de Parámetros SSID)
Longitud de Etiqueta:	7
Interpretación de Etiqueta:	RED_LIC

Velocidades Soportadas:	1.0(B) 2.0(B) 5.5(B) 11.0(B) 22.0
Número de Etiqueta:	1 (Velocidades Soportadas)
Longitud de Etiqueta:	5
Interpretación de Etiqueta:	Velocidades Soportadas: 1.0(B) 2.0(B) 5.5(B) 11.0(B) 22.0

Conjunto de Parámetros DS:	Canal Actual: 6
Número de Etiqueta:	3 (Conjunto de Parámetros DS)
Longitud de Etiqueta:	1
Interpretación de Etiqueta:	Canal Actual: 6

Mapa de Indicación de Tráfico (TIM):	DTIM 0 de 3 mapas de bits vacíos
Número de Etiqueta:	5 (Mapa de Indicación de Tráfico (TIM))
Longitud de TIM:	4
Cantidad DTIM:	0
Período DTIM:	3
Control de Mapa de Bits:	0x00 (mcast: 0, bitmap offset 0)

Información de País	Código de País: US, Cualquier Entorno
Número de Etiqueta:	7 (Información de País)
Longitud de Etiqueta:	6
Interpretación de Etiqueta:	Código de País: US, Cualquier Entorno
Canal de Inicio:	1
Canales:	11
Máxima Potencia de Transmisión	30dBm

Velocidades Soportadas Extendido:	6.0 9.0 12.0 18.0 24.0 36.0 48.0 54.0
Número de Etiqueta:	50 (Velocidades Soportadas Extendido)
Longitud de Etiqueta:	8
Interpretación de Etiqueta:	Velocidades Soportadas: 6.0 9.0 12.0 18.0 24.0 36.0 48.0 54.0 [Mbps]

Vendedor Específico:	GlobalSu
Número de Etiqueta:	221 (Vendedor Específico)
Longitud de Etiqueta:	6
Vendedor:	GlobalSu
Interpretación de Etiqueta:	Sin interpretación

Figura 7.24 - Trama Beacon

A continuación del Sello de Tiempo, aparece otro elemento fijo obligatorio de 2 $bytes$, denominado **Intervalo de Beacon (Beacon Interval)**. Debido a que los Beacons son tramas periódicas que anuncian la existencia de una red IEEE 802.11, transportando información sobre parámetros del BSS y sobre las tramas almacenadas en *buffer* en el AP para aquellas móviles en modo de ahorro de potencia, las estaciones no sólo tienen la obligación de recibir estas tramas especiales, sino que además, para hacerlo correctamente, deben conocer cada cuánto tiempo se transmiten. El campo Intervalo de Beacon se expresa en unidades de $1024\ \mu seg$, siendo común un valor de 100, equivalente a $0.1\ seg$, como es el caso del ejemplo de la Fig. 7.24.

Otro elemento fijo obligatorio es el de **Información de Capacidades (Capability Information)**. Se trata de un campo de 2 $bytes$ cuyo detalle se observa en la Fig. 7.25.

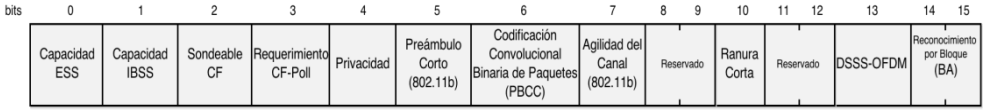


Figura 7.25 - Elemento de Información de Capacidades

Cada bit del elemento es una bandera que anuncia una función particular de la red. De este modo, las móviles pueden saber si soportan las características del BSS. De no ser así, no pueden asociarse. Se trata del siguiente grupo de banderas:

- **Capacidad ESS y IBSS**: se trata de los dos primeros bits, que son mutuamente excluyentes. Si la red es del tipo Infraestructura el bit ESS está en ¨1¨ y el bit IBSS en ¨0¨. En el caso de redes *ad hoc* es al revés.
- **Sondeable CF y Requerimiento CF-Poll:** se trata de los bits de sondeo, cuya interpretación varía según que el uso sea por parte de las móviles o de un AP. La Tabla 7.4 presenta un resumen de estas interpretaciones.

Tabla 7.4 – Bits de CF en Información de Capacidades

Sondeable CF	**Req. CF-Poll**	**Móvil**	**AP**
0	0	No soporta sondeo	No soporta PCF
0	1	Soporta sondeo pero no requiere ser puesta en la lista de sondeo.	Usa PCF para entrega, pero no soporta sondeo

1	0	Soporta sondeo requiere ser puesta en una posición en la lista de sondeo.	Usa PCF para entrega y sondeo
1	1	Soporta sondeo pero requiere nunca ser sondeada (se la tratará como si no soportara operación CF)	Reservado

- **Privacidad**: bandera que indica el uso de WEP para el cifrado de los datos.
- **Preámbulo Corto**: fue agregado para soporte de una forma modificada de espectro esparcido de secuencia directa que usa IEEE 802.11b, denominado DSSS de alta velocidad.
- **Codificación Convolucional Binaria de Paquetes (PBCC):** también fue agregado por causa de 802.11b, para indicar que se usa un esquema de modulación especial.
- **Agilidad del Canal (Channel Agility)**: indica DSSS de alta velocidad con opción de canal ágil. IEEE 802.11b incluye esta capacidad para evitar interferencias con redes FHSS. Cuando se utiliza agilidad del canal, las redes 802.11b saltan periódicamente de canal con una frecuencia y tiempo de permanencia en cada canal ajustados para evitar conflictos con otras redes en la misma área.
- **Ranura Corta**: se refiere al soporte de 802.11g que permite el uso de un preámbulo de menor duración a nivel de capa física.
- **DSSS-OFDM**: indica el uso opcional de DSSS-OFDM en 802.11g.
- **Reconocimiento por bloque (BA, Block Acknowledgment)**: se trata de los últimos dos bits, que han cobrado el significado definido en IEEE 802.11e. El reconocimiento por bloque fue ideado para mejorar la eficiencia de la MAC. La enmienda 802.11n lo ratificó y mejoró, haciéndolo obligatorio para dispositivos que cumplan con ese estándar. El mecanismo BA consiste de una fase inicial donde se negocian capacidades con el receptor de las tramas, tales como tamaño de *buffer* de bloque. En esta fase, mediante una trama especial, denominada ADDBA, el transmisor especifica los números de secuencia inicial y final que el receptor debería esperar recibir. Si el receptor lo acepta, comienza la fase de envío de varias tramas, con reconocimiento BA. Finalmente, el acuerdo finaliza con una nueva trama denominada DELBA.

Si se observa el ejemplo presentado en la Fig. 7.24, el campo de Información de Capacidades indica que el AP posee capacidades de encriptado WEP, Preámbulo Corto 802.11b y Ranura Corta 802.11g.

En referencia a la misma figura, a continuación del elemento descripto, el *sniffer* presenta un título referido a Parámetros Etiquetados. Se trata de los elementos de longitud variable que poseen un formato especial, de tres campos: etiqueta de identificación, longitud de elemento e información propiamente

dicha, de longitud variable. Se suele denominar TLV a este tipo de codificación. La sigla es por Tipo, Longitud, Valor.

El primero de los elementos etiquetados es de aparición obligatoria. Se trata de la **Identidad del Conjunto Servicio (SSID, Service Set Identity)**, que no es otra cosa que el nombre que el administrador le ha asignado a la red. Este nombre es compartido por todos los BSS dentro de un ESS. Es un elemento de longitud variable de 0 a 32 $bytes$. El caso de 0 $bytes$ es especial, se denomina SSID de *broadcast*, y se usa en las tramas de Requerimiento de Sondeo cuando una móvil intenta descubrir todas las redes IEEE 802.11 que existan en su área. En el ejemplo de la Fig. 7.24, el nombre es de siete caracteres, "RED_LIC", de ahí que el campo de longitud lleve el número 7.

Luego del nombre de la red aparece el elemento referido a las **Velocidades Soportadas (Supported Rates)**. Cuando las móviles intentan asociarse a una red deben conocer la velocidad o velocidades soportadas por la misma. Algunas son opcionales y otras obligatorias. La Fig. 7.26 presenta el detalle de este elemento. En cada byte se codifica el valor de velocidad en los 7 $bits$ de menor orden. Con respecto al más significativo, cuando está en ¨1¨, indica que la velocidad es obligatoria, en cambio cuando está en ¨0¨ se trata de una velocidad opcional. Se pueden codificar hasta 8 velocidades. En la revisión inicial de IEEE 802.11, los 7 $bits$ se codificaban como múltiplos de 500 $kbps$. Así, el valor 2 se interpretaba como velocidad de 1 $Mbps$, 4 codificaba la velocidad de 2 $Mbps$, el número 11 se asociaba a 5.5 $Mbps$ y 22 a 11 $Mbps$. De este modo, la máxima velocidad codificable en este elemento era de 63.5 $Mbps$, bastante menor a las velocidades soportadas por nuevos estándares. Por este motivo, la IEEE cambió la interpretación mencionada a partir de la definición de un nuevo elemento, etiquetado con el número 50, y conocido como **Velocidades Soportadas Extendido (Extended Supported Rates)**, que fue estandarizado para manejar más de ocho velocidades.

En la trama Beacon de nuestro ejemplo, el AP soporta 5 velocidades: 1 $Mbps$, 2 $Mbps$, 5.5 $Mbps$, 11 $Mbps$ y 22 $Mbps$. Más adelante, a continuación de otros elementos, también aparece el nuevo elemento extendido, anunciando que puede trabajar a 6 $Mbps$, 9 $Mbps$, 12 $Mbps$, 18 $Mbps$, 24 $Mbps$, 36 $Mbps$, 48 $Mbps$ y 54 $Mbps$.

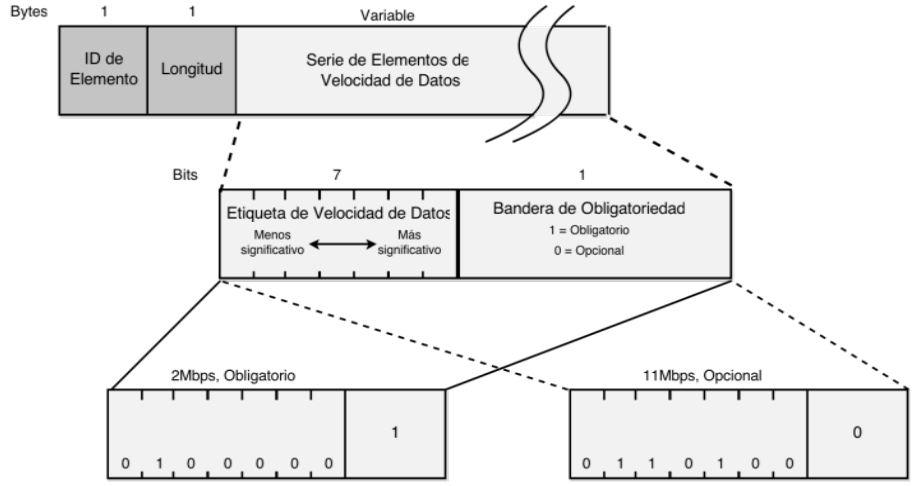


Fig. 7.26 - Velocidades soportadas.

El elemento siguiente en la Fig. 7.24 es el **Conjunto de Parámetros DS (*DS* Parameter Set)**. Se trata del único parámetro que se necesita para trabajar con espectro esparcido en modo DS: el número de canal. La red del ejemplo se ha desplegado en el canal 6.

A continuación sigue uno de los elementos más importantes de la trama Beacon. Se trata del **Mapa Indicador de Tráfico (TIM)**, cuya composición se detalla en la Fig. 7.27. Como se explicó anteriormente, el AP almacena tramas para las estaciones móviles que se encuentran en modo de ahorro de potencia. Periódicamente, el AP intenta entregar estas tramas a las estaciones que corresponda. Parte de esta operación, cuyo propósito principal es evitar el deterioro de la vida útil de las baterías, se realiza enviando el TIM. En este mapa se indica cuáles estaciones tienen tramas almacenadas en el AP esperando ser entregadas.

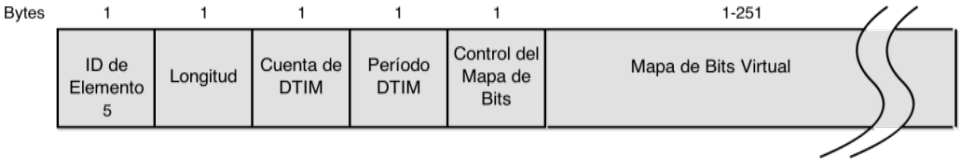


Fig. 7.27 - TIM

Sus principales campos son:

- **Cuenta de DTIM (DTIM Count)**: el TIM cuenta con este campo de $1\,byte$ que representa el número de tramas Beacon que serán transmitidas antes de la siguiente trama de DTIM. Estas tramas son tramas especiales que indican que prontamente se entregarán tramas de *broadcast* y *multicast* almacenadas en el AP para las móviles que hayan

entrado en modo ahorro de potencia. No todas las tramas Beacon son tramas DTIM, por este motivo existe este campo y el que le sigue. El contador de DTIM se disminuye en una unidad en cada trama Beacon, hasta llegar a 0 en la propia trama DTIM.

- **Período DTIM (DTIM Period)**: este campo de 1 $byte$ indica la cantidad de intervalos de Beacon entre tramas DTIM.
- **Control del Mapa de Bits (Bitmap Control)**: es un campo de 1 $byte$ que, junto con el que sigue, constituyen el TIM. Se divide en dos subcampos: el bit 0 se usa para indicación del estado de tráfico de la AID 0, reservada para tráfico *multicast*, los restantes 7 $bits$ son el corrimiento u *offset* del mapa de bits. El *offset* se puede usar para transmitir una parte del mapa de bits virtual como una forma de comprimir la información transmitida. Por medio de este sub-campo, que se relaciona con el comienzo del mapa, y el campo de Longitud, las estaciones pueden deducir cuál parte del mapa virtual se ha incluido, sin que haya que transmitir el estado de los 2008 bits representativos de cada AID.
- **Mapa de Bits Virtual**: debería tratarse de una estructura de 2008 bits, cada uno en relación con un AID, es decir con una móvil asociada a la red. Un bit en alto, en dicho mapa, indicaría que hay tramas almacenadas para ese AID. Gracias a la presencia del campo previo se puede comprimir la información y no enviar un campo tan largo.

En el caso de la trama Beacon del ejemplo de la Fig. 7.24, el AP comunica que el período entre tramas DTIM es de 3 intervalos de Beacon y también que es inminente el envío de tramas *broadcast* y *multicast* almacenadas, ya que la Cuenta de DTIM es nula. El campo de Longitud es de 4 $bytes$ y el *offset* nulo, indicando que el campo de mapa virtual se ha comprimido en 1 $byte$.

Luego del TIM, en la trama del ejemplo, aparece el elemento **País (Country)**. Cuando fue diseñado el estándar original, se tuvieron en cuenta las regulaciones en vigencia en los países más industrializados. Al irse agregando nuevos países a la aceptación de la norma, en vez de revisar lo establecido cada vez que un país se agregaba, se armó una nueva especificación para describir las restricciones vigentes en los mismos. Esta es la misión de este elemento cuyo detalle se presenta en la Fig. 7.28.

El elemento *Country* lleva al principio un identificador de país, seguido de una serie de descriptores de 3 $bytes$ para desarrollar las regulaciones:

- **Identificador del País**: es un campo de 3 $bytes$, los primeros dos son el código ISO del país y el tercero codifica las regulaciones para este tipo de redes, dentro y fuera de los edificios. En la trama del ejemplo, el país es USA, país donde se fabrica el AP, y la regulación *Any*, palabra clave que corresponde a la descripción de la regulación.
- **Descriptores**: cada descriptor a continuación, es un conjunto de 3 $bytes$. El primer byte presenta el número del primer canal de la banda: canal 1. Luego viene el número de canales, que es el tamaño de la banda sujeto a restricciones de potencia, en este caso 11. El último byte informa la

potencia máxima de transmisión, expresada en dBm. Un dBm es una unidad de medida de potencia expresada en decibelios (dB) relativa a un miliwatt (mW), muy utilizada en aparatos de comunicaciones. La trama Beacon de la Fig. 7.24 indica para este campo el valor de potencia máxima de $30\,dBm = 10\log\frac{P}{1mW} => P = 1W$. En particular, el elemento debe tener un número par de bytes, si no se agregará un byte de relleno.

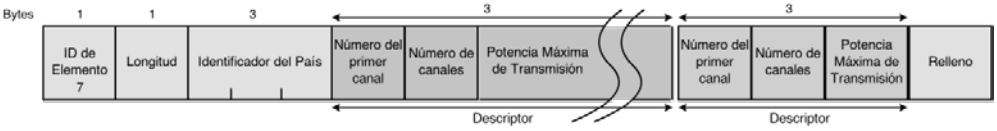


Figura 7.28 - Elemento Country

7.6.2 Elementos Fijos de las Tramas de Administración

En el ejemplo de la trama Beacon de la Fig. 7.24 se han desarrollado diversos tipos de elementos de longitud fija presentes, tales como Intervalo de Beacon, Información de Capacidades y Sello de Tiempo. En esta sección se mencionarán otros elementos de longitud fija que pueden aparecer en las tramas de administración:

- **Número de Algoritmo de Autenticación**: se trata de un elemento de $2\,bytes$ de longitud que define, mediante un número, el algoritmo a utilizar en la fase de autenticación. El protocolo original sólo definía dos valores: "0" para autenticación Sistema Abierto definido en IEEE 802.1X, y "1" para Autenticación de Clave Compartida que quedara obsoleta a partir de la definición del estándar IEEE 802.11i.

- **Número de Secuencia de la Transacción de Autenticación**: durante la fase de autenticación se intercambian varias tramas entre el AP y la móvil que se está autenticando. Cada trama lleva un número de secuencia en este campo de $1\,byte$. El propósito es evitar posteriores ataques de repetición (*replay*).

- **Dirección del AP actual**: este campo de $6\,bytes$ lleva la dirección del último AP con el que la estación móvil ha estado asociada. Se usa para facilitar el proceso de re-asociación cuando la móvil cambia de punto de acceso.

- **Intervalo de Escucha (Listen Interval)**: cuando una móvil se asocia a una WLAN debe anunciar la cantidad de tiempo que usará para cerrar el circuito transceptor cuando pase al modo PS, posibilidad que permite ahorrar vida útil de la batería. Este elemento de $2\,bytes$ indica la

cantidad de intervalos de Beacon que la móvil permanecerá dormida. Desde el punto de vista del AP, este tiempo se traduce en recursos que el AP deberá disponer para esa móvil, cada vez que ésta anuncie que entra en modo PS. Podría darse el caso que el AP se rehúse a la asociación por resultar la misma demasiado intensiva en cuanto a los recursos solicitados.

- **Identificador de Asociación (Association ID)**: se trata de un número de $2\ bytes$ que se le asigna a la estación como identificador al momento de asociarse. Puede tratarse de un número en el rango $1 - 2007$.

- **Código de Razón**: es un número de $2\ bytes$ que se carga en tramas des-asociación o de de-autenticación, explicando el motivo de la situación. La Tabla 7.5 presenta los códigos definidos.

- **Código de Estado**: es un número de dos bytes que indica el éxito o fracaso de una operación. La Tabla 7.6 presenta los códigos definidos.

Tabla 7.5 - Códigos de Razón definidos.

Código	Significado
0	Reservado.
1	No especificado.
2	Autenticación previa no válida.
3	Móvil ha abandonado BSS o ESS y es de-autenticada.
4	Reloj de inactividad ha expirado y la estación se ha desasociado.
5	Des-asociación debida a falta de recursos en el AP.
6	Trama con campo tipo o subtipo incorrecto recibido de una estación no autenticada.
7	Trama con campo tipo o subtipo incorrecto recibido de una estación no asociada.
8	Móvil ha abandonado BSS o ESS y es desasociada.
9	Pedido de asociación o re-asociación solicitado antes de autenticación.
10	Desasociado por valor inaceptable en el elemento Capacidad de Potencia. IEEE 802.11h
11	Desasociado por valor inaceptable en el elemento Canales Soportados. IEEE 802.11h
12	Reservado
13	Elemento de información inválido. IEEE 802.11i
14	Falla en el Chequeo de Integridad del Mensaje. IEEE 802.11i
15	Tiempo expirado en el apretón de manos de cuatro vías para claves. IEEE 802.11i
16	Tiempo expirado para clave de grupo. IEEE 802.11i
17	El elemento de información del apretón de manos de cuatro vías para claves tiene diferentes parámetros de seguridad que el conjunto inicial. IEEE 802.11i

Código	Significado
18	Grupo de Cifrado inválido. IEEE 802.11i
19	Cifrado entre pares inválido. IEEE 802.11i
20	Protocolo de Administración Autenticación y Clave inválido. IEEE 802.11i
21	Elemento de información RSN no soportado. IEEE 802.11i
22	Capacidades inválidas en el elemento RSN
23	Falla de Autenticación 802.11X. IEEE 802.11i
24	Conjunto de Cifrado propuesto rechazado por políticas configuradas. IEEE 802.11i
25	Reservado.

Tabla 7.6 - Códigos de Estado definidos.

Código	Significado
0	Operación completamente exitosa.
1	No especificado.
2 -9	Reservado.
10	El conjunto requerido de Capacidades no puede ser soportado.
11	Re-asociación denegada; asociación previa no pudo ser identificada y transferida.
12	Asociación denegada por motivo no especificado en el estándar 802.11
13	Algoritmo de Autenticación requerido no soportado.
14	Número de Secuencia de Autenticación no esperado.
15	Denegación de Autenticación. Respuesta al desafío ha fallado.
16	Denegación de Autenticación. La siguiente trama en la secuencia no arribó en la ventana esperada.
17	Denegación de Asociación. El AP tiene restricciones en los recursos.
18	Denegación de Asociación. La móvil no soporta las velocidades requeridas en el BSS.
19	Denegación de Asociación. La móvil no soporta Preámbulo Corto. 802.11b
20	Denegación de Asociación. La móvil no soporta PBCC. 802.11b
21	Denegación de Asociación. La móvil no soporta Agilidad de Canal. 802.11b
22	Denegación de Asociación. Se requiere Administración del Espectro. 80211h
23	Denegación de Asociación. Valor inaceptable de Capacidad de Potencia. 802.11h
24	Denegación de Asociación. Valor inaceptable de Canales Soportados. 802.11h
25	Denegación de Asociación. La móvil no soporta Ranura Corta. 802.11g
26	Denegación de Asociación. La móvil no soporta DSSS-OFDM. 802.11g
27-39	Reservado
40	Elemento de información inválido. 802.11i

Código	Significado
41	Cifrado de grupo broadcast/multicast inválido. 802.11i
42	Cifrado por pares inválido. 802.11i
43	Protocolo de Autenticación y Manejo de Clave inválido. 802.11i
44	La versión del elemento de información RSN no es soportada. 802.11i
45	Capacidad RSN no soportada. 802.11i
46	Conjunto de Cifrado rechazado por política administrativa. 802.11i
47-fin	Reservado.

7.6.3 Elementos de Información de las Tramas de Administración

En el ejemplo de la trama Beacon ofrecido, se pudo observar que, aparte de los elementos fijos, aparecen una serie de elementos de longitud variable, con una estructura tipo TLV, por ejemplo: SSID, Velocidades Soportadas, Velocidades Soportadas Extendido, Vendedor, Conjunto de Parámetros DS, TIM y País. A continuación se presentan otros elementos de longitud variable que también pueden estar presentes en las tramas de administración:

- **Conjunto de Parámetros FH (*FH Parameter Set*)**: este elemento contiene los parámetros necesarios para asociarse a una red que utilice espectro esparcido por FH, tal como se muestra en la Fig.7.29. Es excluyente con respecto al elemento *DS Parameter Set*. En el campo Tiempo de Residencia (*Dwell Time*) se anuncia cantidad de tiempo que se debe permanecer en cada frecuencia de la secuencia, expresado en unidades de $1024\ \mu seg$. El campo Conjunto de Salto (*Hop Set*) se refiere al conjunto de patrones en uso, en tanto que el Patrón de Salto (*Hop Pattern*) indica el patrón específico de ese conjunto. El último campo del elemento presenta el Índice de Salto (*Hop Index*), que explicita la frecuencia actual de trabajo dentro del patrón.

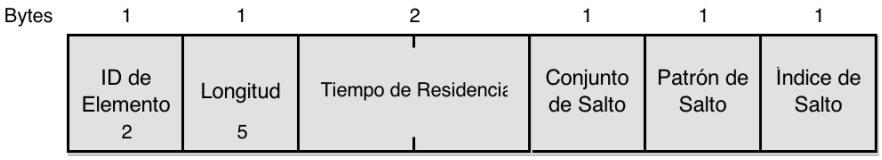


Figura 7.29 - Parámetros para FH.

- **Conjunto de Parámetros CF (*CF Parameter Set*)**: se transmite en tramas Beacon generados por el AP que soporta operación libre de contienda

- **Conjunto de Parámetros IBSS (*IBSS Parameter Set*)**: es un elemento necesario en redes *ad hoc*. Las redes tipo IBSS requieren de un único parámetro, conocido como Ventana de Mapa de Indicación de Anuncio de Tráfico ATIM. Esta ventana indica la cantidad de unidades de tiempo,

medidas en porciones de 1024 μseg, que existirá entre tramas ATIM de la red IBSS desplegada.

- **Parámetros de Patrón de Salto (*Hopping Pattern Parameters*) y Tabla de Patrón de Salto (*Hopping Pattern Table*)**: estos elementos se idearon para cumplir con patrones que verifiquen las restricciones en nuevos países que se vayan incorporando a esta tecnología de redes.

- **Requerimiento (*Request*)**: en las tramas de Requerimiento de Sondeo se puede transportar este tipo de elemento para pedirle a la red cierto tipo de elementos de información que se presentan como una lista de número enteros de 1 $byte$ cuya interpretación es directa para el AP que reconoce este elemento.

- **Texto Desafío**: se trata de un texto de longitud variable, que el AP le envía a la estación móvil en el proceso de autenticación. Si la móvil conoce la clave compartida, lo devolverá en la trama siguiente cifrado de manera correcta.

- **Restricción de Potencia (*Power Constraint*)**: este elemento posee un campo de 1 $byte$ en la sección de valor, a través del cual, en el caso de que haya algún tipo de restricciones aparte de las regulatorias, la red le anuncia a las estaciones el valor de potencia máxima transmitida. Se trata de un número entero expresado en dB que se debe restar al máximo local regulado. Así, si el máximo fuera 10 dBm, y el elemento informa el valor 2, la estación debería ajustar el máximo de potencia transmitida a 8 dBm.

- **Capacidad de Potencia**: es un elemento que permite a las estaciones móviles anunciar su capacidad de transmisión mínima y máxima, expresada en unidades enteras de dBm, codificadas en 2 $bytes$ a continuación de los campos tipo y longitud.

- **Requerimiento de Control de Potencia Transmitida (*TPC Request*)**: este elemento se utiliza para solicitar información de administración del enlace. No tiene datos asociados, por lo que el campo longitud se codifica en cero.

- **Reporte TPC (*TPC Report*)**: el elemento sirve para que las móviles ajusten la potencia de transmisión, a través de una estimación de la atenuación. Se trata de dos valores, cada uno de 1 $byte$. El primer byte del elemento es la potencia transmitida de la trama que contiene este elemento, expresada en dBm. El segundo valor representa el margen del enlace, un número expresado en dB de margen de seguridad.

- **Canales soportados**: es un elemento que carga campos denominados descriptores de sub-banda. Cada descriptor se compone de un número de canal, que es el más bajo de la sub-banda soportada, seguido del número de canales en la sub-banda. Por ejemplo, si un dispositivo soportara los canales 40 a 52, lo codificaría con los números 40 y 12.

- **Anuncio de Cambio de Canal (*Channel Switch Announcement*)**: elemento agregado en función del estándar IEEE 802.11h, que permite que la red cambie dinámicamente de canal. Posee tres campos de información de 1 $byte$: Modo Cambio de Canal (*Channel Switch Mode*), Número del Nuevo Canal (*New Channel Number*) y Cuenta de Cambio de Canal (*Channel Switch Count*). Si el primer campo del elemento indica un valor "1", las estaciones deben detener su transmisión de tramas hasta que se produzca el cambio de canal. Las transiciones de canal se pueden programar, por lo cual no sólo se anuncia el nuevo canal a ocupar, sino también la cantidad de intervalos de tramas Beacon que faltan para el cambio. La cuenta también puede anunciar un valor "0", indicando que la transición puede ocurrir sin previo aviso.

- **Requerimiento de Medición y Reporte de Medición**: se trata de dos elementos clave en 802.11h, ya que sirven para monitoreo del canal y ajuste de niveles de potencia.

- **Silencio (*Quiet*)**: uno de los grandes problemas de señales interferentes se presenta en entornos con radares militares. Para poder adelantarse a la presencia de interferencia, el AP puede usar el elemento *Quiet*, para cerrar el canal y mejorar la calidad de las mediciones. El elemento posee cuatro campos. El primer campo, de 1 $byte$, se denomina *Quiet Count* e indica el número de intervalos de Beacon que faltan para que empiece el período de silencio. El campo que sigue se denomina *Quiet Period*, es también de 1 $byte$ e indica el número de intervalos de Beacon que habrá entre períodos de silencio, para el caso de que los mismos sean programados. *Quiet Duration* es el tercer campo, de 2 $bytes$, e indica el número de unidades de $1024\ \mu seg$ que durará el período de silencio. Se describe de este modo por que el período de silencio puede ser menor que un intervalo de Beacon. Por último, un campo de 2 $bytes$, denominado *Quiet Offset*, es el número de unidades de $1024\ \mu seg$, luego del intervalo de Beacon, a partir de la cual comenzará el siguiente período de silencio. Se debe especificar porque no necesariamente debe comenzar con el intervalo de Beacon y porque puede ser menor que éste.

- **Selección Dinámica de Frecuencia IBSS (*DFS IBSS, Dinamic Frequency Selection IBSS*)**: en redes Infraestructura existe un algoritmo de selección dinámica de frecuencia que es manejado por el AP, pero en redes independientes debe designarse una estación para tal efecto. Esta estación transmitirá el elemento de información DFS que se presenta en

la Fig. 7.30, denominado IBSS DFS. Luego del encabezado, se escribe la dirección de $6\,bytes$ de la MAC de la estación responsable del algoritmo. A continuación se escribe el intervalo de recuperación de $1\,byte$. Lo más importante de la trama es una serie de mapas de los canales, con información sobre lo que se detecta en cada uno. Para cada canal, se comienza con el número de canal de $1\,byte$, seguido de un campo de banderas de diferente significado, también de $1\,byte$. El bit BSS se enciende si se detecta tráfico de otra red durante el período de medición. El bit Preámbulo OFDM se enciende si se detecta una secuencia de entrenamiento corta de IEEE 802.11a sin el resto de la trama. Unidentified Signal es 1 bit que se enciende cuando la potencia recibida es alta pero no se puede clasificar si la señal es de otra red IEEE 802.11. El bit Radar se utiliza para indicar la presencia de una señal de radar y el bit Unmeasured se enciende si no se pudo medir.

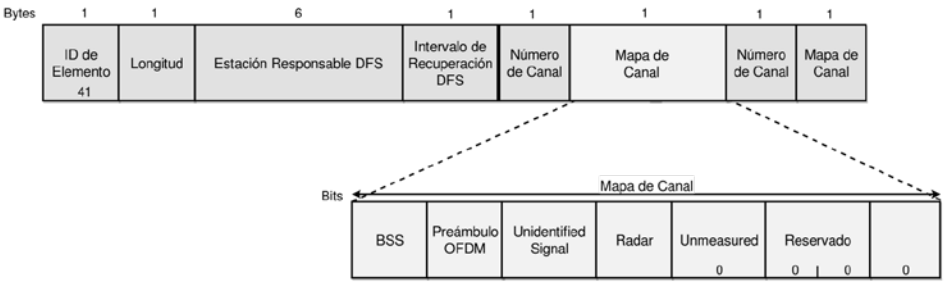


Fig. 7.30 - Elemento IBSS DFS

- **Información de Velocidad Extendida de Capa Física (*ERP Information*)**: IEEE 802.11g definió una capa física de velocidad extendida que se conoce como ERP. El elemento ERP se define para compatibilidad hacia atrás. Se trata de $1\,byte$ con tres flags definidos. El flag Non-ERP Present se enciende cuando una estación más antigua que 802.11g se asocial a la red o cuando la propia red 802.11g se solapa con otra red de otra capa física. El flag Use Protection se enciende cuando se encuentra presente una estación sin capacidades 802.11g, por compatibilidad hacia atrás. Por último, el flag Barker Preamble Mode se enciende cuando se asocian estaciones que no manejan el preámbulo corto.
  Este elemento deja entrever los problemas que se presentarán al mezclarse en la misma WLAN dispositivos de distintas capacidades.

- **Red de Seguridad Robusta (Robust Security Network)**: el agregado de condiciones de seguridad más robustas, definidas en 802.11i, hizo necesaria la definición de nuevos elementos de información para comunicación de nuevas capacidades entre dispositivos. El más

importante se conoce como elemento RSN y se presenta en la Fig. 7.31. El campo de Versión es 1. El segundo campo se denomina Conjunto de Cifrado de Grupo y se refiere a la protección brindada a las tramas de *broadcast* y *multicast*. El AP debe elegir un cifrado de grupo compatible con todas las estaciones. El identificador de cifrado de grupo es un número de 4 $bytes$ que comienza con un OUI de vendedor y sigue con un número, cuyo significado se presenta en la Tabla 7.7.

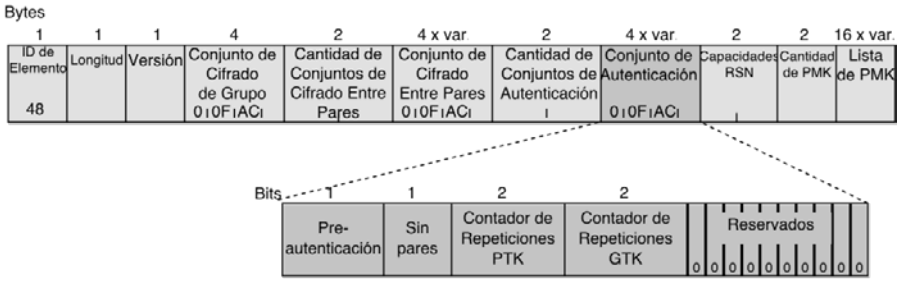


Fig. 7.31 - Elemento RSN

Tabla 7.7 - Conjunto de Cifrado de Grupo.

OUI	Conjunto	Definición
00:0F:AC	0	Usar Conjunto de Cifrado de Grupo (sólo para cifrados pares)
00:0F:AC	1	WEP-40
00:0F:AC	2	TKIP
00:0F:AC	3	Reservado
00:0F:AC	4	CCMP (default 802.11i)
00:0F:AC	5	WEP-104
Vendedor	Cualquier valor	Definido por vendedor

WEP es el algoritmo criptográfico para cifrado de datos que originalmente se propuso para IEEE 802.11, el número que lo acompaña es la cantidad de bits de la clave. El Protocolo de Integridad de la Clave Temporal (TKIP, Temporal Key Integrity Protocol) fue aprobado en 2002 por la Alianza WiFi con el nombre de Acceso de WiFi Protegido (WPA, Wi-Fi Protected Access). TKIP usaba el mismo algoritmo de cifrado WEP, pero incorporaba técnicas para comprobación de la integridad del mensaje, rotación de claves y otras que desalentaron muchos ataques. La versión final de TKIP, que incluía 802.1X y el Algoritmo Avanzado para Encriptado de Datos Estándar (AES, Advanced Encryption Standard) basado en CCMP, fue aprobada en 2004 junto con la publicación de 802.11i con el nombre WPA2, también

conocida como RSN. Aunque esta versión incluye cifrado de datos seguro, autenticación más fuerte y control de acceso a nivel de administración, para 2011 ya contaba con reportes de seguridad.
A continuación del campo de cifrado de grupo, aparecen una serie de campos de Cifrado entre Pares para tramas *unicast*, codificadas como un contador de $2\ bytes$ y una serie de descriptores. Si el selector del conjunto es "0" quiere decir que sólo se soporta cifrado de grupo.
Luego siguen una serie de descriptores del método de Autenticación de codificación parecidos a los explicados previamente. IEEE 802.11i reemplazó el método de autenticación original, agregando un protocolo de cuatro vías que permite derivar una Clave Temporal entre Pares (PTK, Pairwise Transient Key) y otra para descifrado de tramas *multicast* y *broadcast* GTK (Group Temporal Key).
En el campo de capacidades RSN se describen, mediante banderas, las capacidades del transmisor en cuanto a pre-autenticación para ganar tiempo en procesos de re-asociación, contadores *replay* para diferentes niveles de prioridad definidos en extensiones emergentes para calidad del servicio y una lista de Claves Maestras (PMK, Pairwise Master Key) para *caching* en el AP.
La explicación exhaustiva de todos los algoritmos mencionados excede el detalle de este libro, aunque su mera mención deja entrever las mejoras adoptadas en los últimos años en términos de cifrado y autenticación.

7.6.4 Trama Requerimiento de Sondeo (Probe Request)

Estas tramas son emitidas por las estaciones móviles cuando investigan áreas en búsqueda de presencia redes IEEE 802.11. Su formato se presenta en la Fig. 7.32. Todos sus campos son obligatorios.

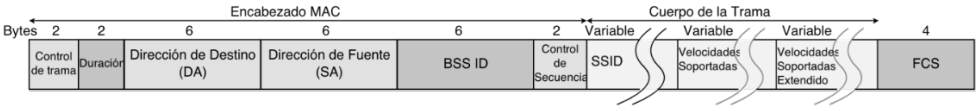


Figura 7.32 - Trama Probe Request

Estas tramas poseen dos campos:

- **SSID**: las estaciones que escuchen estas tramas, pueden usar la información para determinar si la móvil puede incorporarse a la red. La móvil explicita su intención de agregarse a una red particular, cuando identifica el SSID, pero también podría escribir en este campo un SSID de *broadcast*, indicando que no tiene preferencias por ninguna red en particular. En las tramas de sondeo también puede aparecer un elemento Request, que se usa para solicitarle a la red ciertos elementos de información.

- **Velocidades Soportadas, Velocidades Soportadas Extendido**: para poder asociarse, es necesario que la móvil sea capaz de soportar las velocidades requeridas por la red.

7.6.5 Trama Respuesta de Sondeo (Probe Response)

Estas tramas se generan en respuesta a las de Requerimiento de Sondeo. La estación que transmitió la última trama Beacon es la encargada de emitirlas. En redes de Infraestructura, el AP es el responsable de su emisión, en redes *ad hoc* la responsabilidad es distribuida. El formato de la trama se presenta en la Fig. 7.33. Algunos de sus campos son mutuamente excluyentes.

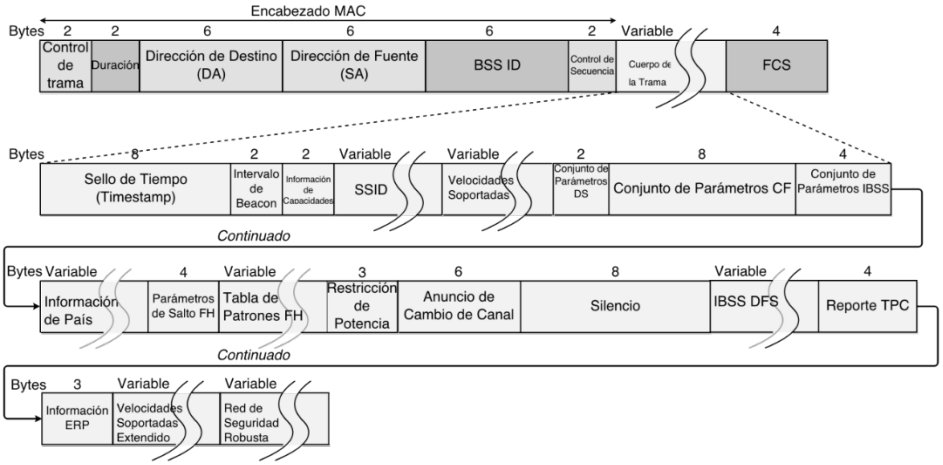


Figura 7.33 - Trama Probe Response

Estas tramas llevan los mismos parámetros que una trama Beacon, excepto por el elemento TIM, ya que las móviles que transmiten las tramas de Requerimiento de Sondeo todavía no se encuentran asociadas a la red y esta información no les incumbe.

7.6.6 Tramas de Autenticación y Asociación

Antes de poder asociarse a una red WLAN, las estaciones pasan por un proceso de autenticación. El protocolo original define dos versiones en este sentido: autenticación abierta (sin autenticación) o con clave compartida. Durante esta fase, se intercambian tramas como la presentada en la Fig. 7.34. El intercambio forma parte del protocolo IEEE 802.11b pero, como se ha mencionado, IEEE 802.11i incorporó nuevos mecanismos de autenticación mucho más robustos.

La trama de Autenticación carga el elemento fijo Número de Algortimo de Autenticación. Se trata de un número de 2 $bytes$ que identifica el tipo de autenticación usada en el proceso previo a la asociación. El protocolo original sólo define dos valores: "0" para lo que se denomina Sistemas Abiertos (*Open System*) o que no tienen autenticación, y "1" para el método de autenticación de Clave Compartida.

A continuación aparece otro elemento fijo, el Número de Secuencia de la Transacción de Autenticación, también de 2 $bytes$. Sirve para seguir el progreso del intercambio de autenticación. Este elemento es preciso porque la autenticación es un proceso de varias etapas, donde el AP envía un desafío a la móvil y ésta debe responder correctamente para poder luego ser asociada. Este campo puede tomar valores desde 1 a 65.535. Nunca puede valer "0".

El elemento Código de Estado es el indicador de éxito o fracaso en una operación. Cuando es "0" significa éxito. Por último, aparece el elemento de longitud variable, definido para la autenticación por clave compartida. Se trata del Texto de Desafío que se intercambia entre el AP y la móvil, para que el AP pueda comprobar si la estación que se está autenticando conoce la clave compartida. El AP le pasa a la móvil el texto sin cifrar y ésta lo devuelve cifrado. Si la clave es correcta, el AP podrá descifrarlo correctamente y así autenticar a la móvil.

Aparte de la trama de autenticación, existen tramas de De-autenticación que sirven para finalizar una relación de autenticación. Las mismas incluyen un único elemento de longitud fija: el Código de Razón.

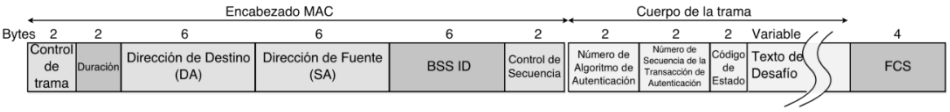


Figura 7.34 - Trama de Autenticación

Una vez que la estación móvil ha encontrado una red y se ha autenticado ante la misma, debe intentar incorporarse enviando tramas de Requerimiento de Asociación, tales como la de la Fig. 7.35. En esta trama, se presenta el elemento Información de Capacidades, el SSID, las Velocidades Soportadas/Velocidades Soportadas Extendido y el Intervalo de Escucha. El AP precisa conocer este último dato para saber durante cuánto tiempo deberá almacenar tramas para la estación cuando la misma entre en modo PS. De este modo puede estimar los recursos necesarios. Este elemento también está presente en tramas de Re-Asociación.

Las tramas para asociación también podrían cargar elementos tales como Capacidad de Potencia, Canales Soportados y RSN.

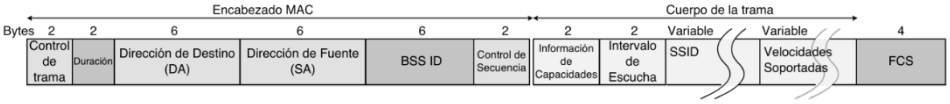


Figura 7.35 - Requerimiento de Asociación.

Por otra parte, las estaciones móviles que se mueven entre distintos AP, dentro de un mismo ESS, precisan re-asociarse con la red, antes de poder seguir usando el sistema de distribución. Lo mismo sucede cuando se alejan de un AP y luego regresan a su área de cobertura. En estos casos se utiliza la trama de Re-asociación, que se presenta en la Fig. 7.36.

La trama de Re- Asociación difiere de la trama de Asociación en un campo. Se trata del elemento Dirección del AP actual. En este elemento, las estaciones móviles escriben la dirección MAC de 6 *bytes* del AP actual al que se encuentran asociadas. Como se ha explicado, esta información se utiliza para facilitar procesos de asociación y re-asociación, ya que cuando se establece una asociación con un AP diferente, esta referencia al AP actual sirve para la transferencia de asociación entre ambos AP y el intercambio de tramas en almacenamiento que se deberán entregar a la móvil que se mueve entre diferentes servicios básicos.

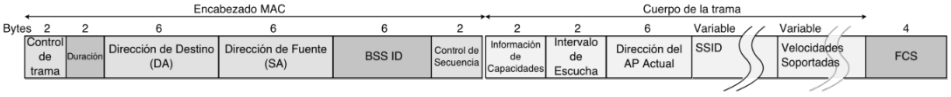


Figura 7.36 - Trama de Requerimiento de Re-Asociación.

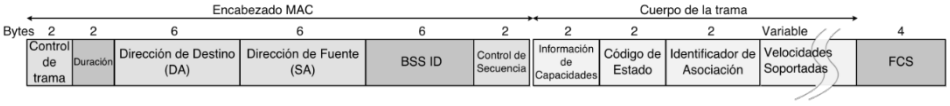


Figura 7.37 - Trama de Respuesta de Asociación o de Re-Asociación.

Las tramas de Respuesta de Asociación y de Respuesta de Re-asociación surgen como contestación a los requerimientos del mismo nombre por parte de las móviles. Tienen el formato que se presenta en la Fig. 7.37. Como parte de la respuesta, el AP asigna un identificador de asociación a la móvil.

7.7 Administración IEEE 802.11

Las características particulares de las redes inalámbricas implican detalles de administración muy diferentes al caso de las redes cableadas. Por ejemplo, existe el peligro de accesos de parte de actores no autorizados y además

el medio, por su propia naturaleza, ofrece poca confiabilidad en la comunicación. Por otra parte, el problema del consumo de potencia y autonomía de baterías en las estaciones móviles es crítico.

Las características administrativas del protocolo se diseñaron para reducir el efecto de estos problemas. La funcionalidad de administración se ideó como una cuestión cooperativa entre las móviles y la red. Parte del control de acciones que influyen sobre estas características pueden ser ajustadas en los *drivers* provistos por los fabricantes para accionar placas de red o los propios AP.

El estándar IEEE 802.11 incluye tres componentes en cuanto a su arquitectura de administración, como se puede apreciar en la Fig. 7.38:

- La Entidad de Administración de la capa MAC (MLME, MAC Layer Managment Entity).
- La Entidad de Administración de la capa Física (PLME, Physical-Layer Managment Entity).
- La Entidad de Administración del Sistema (SME, System Managment Entity).

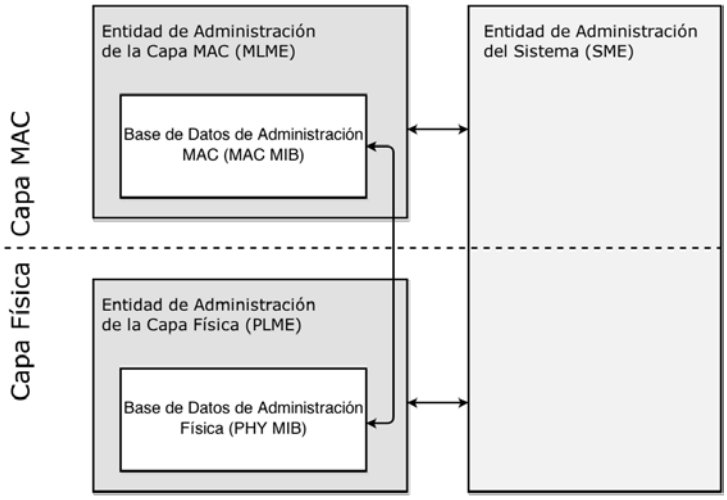


Figura 7.38 - Arquitectura de Administración 802.11.

IEEE 802.11 no especifica formalmente la SME, pues es la manera en que usuarios y *drivers* interactúan con la interfaz de red y colectan información sobre su estado. La Base de Datos de Administración (MIB, Management Information Base) tiene objetos, disponibles por requerimiento, sobre reportes de estado, aunque también existen otros que pueden realizar acciones.

Se definen tres interfaces entre los componentes. La SME puede acceder a las Bases de Datos de MAC y capa física a través de sus respectivas interfaces. Si algunos cambios efectuados en la MAC requieren cambios en capa física, los mismos se realizan a través de la interfaz entre MLME y PLME.

7.7.1 Exploración - Scanning

Antes de asociarse a una red, es preciso encontrarla e identificarla, en un proceso que se conoce como exploración (*scanning*). Existen parámetros relacionados con este proceso, algunos especificados por el usuario, aunque muchas implementaciones usan los valores por default.

Se trata de:

- **Tipo de BSS**: ya sea que se esté buscando una red Independiente o una de Infraestructura, o de ambos tipos, se puede especificar en esta fase.

- **BSSID**: se puede buscar una red particular, mencionando la dirección MAC WiFi del AP, o se puede buscar cualquiera red, situación que se denomina exploración *broadcast*. En este último caso, los resultados de la búsqueda incluirán todos los BSS en el área.

- **SSID**: se trata del Nombre de Red. Generalmente se le dice nombre porque se asocia a un conjunto de caracteres con formato de lectura. Es el nombre con que el administrador bautiza la red. Si los clientes desean hallar cualquier red, este nombre se ajusta a *broadcast*.

- **Tipo de Exploración**: puede ser activo o pasivo. La exploración activa requiere la emisión de tramas de Requerimiento de Sondeo editadas por la móvil que está buscando una red. La exploración pasiva ahorra consumo de batería y sencillamente consiste en escuchar, esperando la recepción de tramas Beacon.

- **Lista de Canales*:*** se trata de la lista de canales que las estaciones pueden especificar para buscar redes de manera activa o pasiva. En el caso de espectro esparcido DS se trata de un verdadero listado de canales, pero en espectro esparcido FH es un patrón de salto.

- **Retardo de Sondeo**: es un tiempo, medido en μseg, anterior a iniciar una exploración activa sobre un canal. Este retardo asegura que un canal vacío o poco cargado genere una situación de bloqueo de la búsqueda.
- **Tiempo de Canal Mínimo y Tiempo de Canal Máximo**: son valores que especifican el tiempo mínimo y máximo de búsqueda en un canal.

La exploración pasiva ahorra batería porque no requiere transmisiones. La estación sólo escucha, en espera de una trama Beacon, sobre cada canal de la lista. La llegada de un Beacon implica almacenamiento y análisis de la información aportada sobre el BSS que lo haya enviado. Esta búsqueda se realiza sobre todos los canales y finaliza con un listado de servicios básicos encontrados y sus capacidades.

En la exploración activa, la estación transmite tramas de Requerimiento de Sondeo sobre cada canal, según las siguientes reglas:

1. En cada canal, la móvil espera una indicación de trama entrante o el tiempo especificado en el parámetro retardo de sondeo. Este parámetro evita que un canal sin uso bloquee todo el procedimiento. Si recibe una trama, es indicación de que el canal está en uso y puede sondearse de manera activa.

2. Para editar las tramas de Requerimiento de Sondeo, la móvil aplica el procedimiento de acceso DCF.

3. Espera que finalice el tiempo mínimo de estadía en un canal. Si el medio no se apreció como ocupado, se concluye que no hay red y se procede a pasar al sondeo del siguiente canal. Si el medio se apreció como ocupado durante el tiempo de estadía mínimo en el canal, la móvil espera hasta el tiempo máximo, y procesa cualquier trama de Respuesta de Sondeo que le llegue.

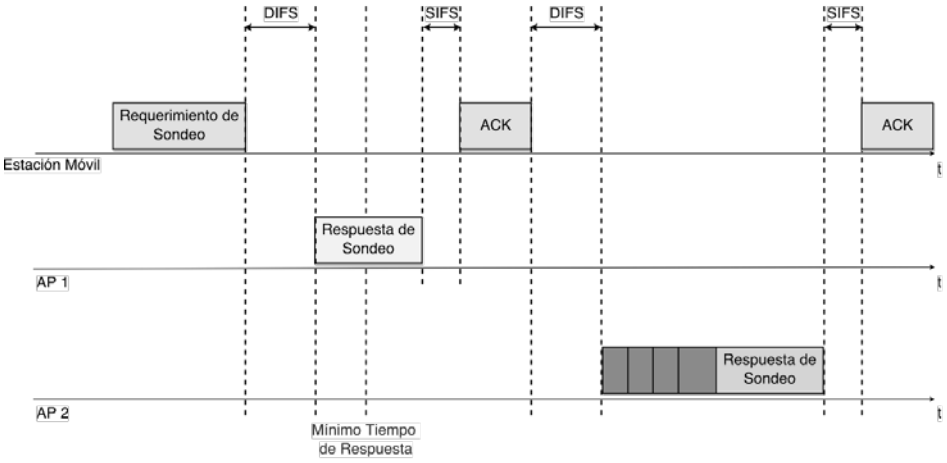


Figura 7.39 - Relaciones de Tiempo Exploración Activa.

La Fig. 7.39 presenta las tramas y tiempos de relevancia en una exploración activa. Vale la pena destacar que en redes Infraestructura, el AP se encarga de transmitir tramas Beacons o Respuestas de Sondeo, pero en redes *ad hoc* esta emisión se convierte en una responsabilidad distribuida. También, como se aprecia en la figura, las tramas de respuesta a la exploración activa son del tipo *unicast*, por lo que quedan sujetas al reconocimiento por ACK. Si la trama de Requerimiento es de tipo *broadcast*, es posible que se reciba más de una trama de respuesta.

7.7.2 Reporte de Exploración

Se trata de un informe que se genera una vez finalizada la búsqueda. El reporte lista todos los BSS descubiertos y sus parámetros. Además del BSSID, el SSID y el tipo de BSS, los parámetros también incluyen:

- **Intervalo de Beacon**: se trata de un número entero que anuncia el tiempo entre tramas Beacon.

- **Período DTIM**: también es un número entero, necesario ya que las tramas DTIM forman parte del mecanismo de ahorro de potencia.

- **Parámetros de Tiempo**: existen dos valores que ayudan a las móviles a sincronizar su propio reloj con el que utiliza el BSS. El elemento Sello de Tiempo, ya presentado, indica el valor del reloj recibido por la estación que realiza la búsqueda. A este valor se le debe sumar un *offset*, propio del tiempo de procesamiento de la móvil. Así, la estación quede habilitada a adaptarse a la información de tiempo que le sirve para incorporarse a un BSS.

- **Parámetros de Capa Física, CF e IBSS**.

- **Conjunto de Velocidades BSS**: lista de velocidades que deben soportar las estaciones que deseen incorporarse a la red. Las móviles deben ser capaces de recibir datos a cualquier velocidad de las listadas.

7.7.3 Incorporación a la WLAN

Luego de obtener los resultados de la exploración, la estación debe elegir con cuál red asociarse. Se trata de una decisión específica de la implementación y puede exigir intervención del usuario. Los criterios que se tienen en cuenta para la elección, en general involucran el nivel de potencia y la fortaleza de la señal. Se trata de un proceso interno a la móvil, que implica la adaptación de parámetros locales a los del BSS.

Incorporarse a un BSS seleccionado requiere adaptación a los parámetros locales requeridos por el BSS y a los parámetros de capa física, sincronizar la información de tiempo de la estación al resto de la red, adaptarse al método de cifrado, poseer capacidades de alta velocidad y adoptar el intervalo de Beacon y el período de DTIM anunciados.

7.7.4 Autenticación

IEEE 802.11 exige a las móviles probar su identidad antes de la asociación. No es una autenticación de doble vía, ya que sólo se autentica la móvil ante el AP.

El Sistema Abierto no es en realidad un método de autenticación ya que consiste en el intercambio de dos tramas, una de requerimiento y otra de respuesta con un código de estado. No hay cifrado ni algoritmos de autenticación, sólo un permiso o una denegación basados en una lista de direcciones MAC permitidas, configuradas por el administrador.

En el caso de Clave Compartida se usa WEP y se requiere de una clave común que tiene que haber sido entregada previamente. De este modo, se envía un texto desafío al cliente y la respuesta prueba la posesión de la clave, como se muestra en la Fig. 7.40.

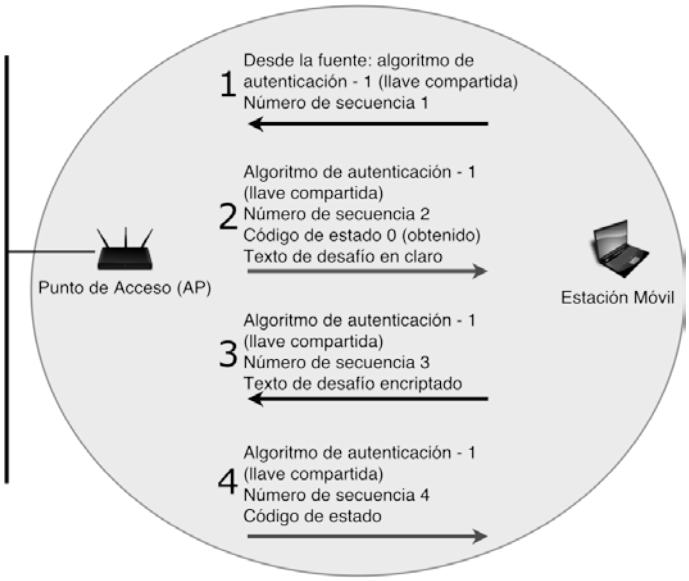


Figura 7.40 - Autenticación de Clave Compartida.

El AP envía a la móvil un texto de desafío para su cifrado. El texto desafío se compone de 128 $bytes$ generados con el generador de clave de WEP, usando un vector de inicialización aleatorio. La móvil lo debe cifrar usando la clave compartida que previamente le ha sido entregada. Cuando el AP recibe el texto cifrado, procede a descifrarlo y, si obtiene el texto desafío original, responde con un código de estado de éxito. Este tipo de autenticación puede vulnerarse con un ataque muy sencillo y por eso no es recomendable. IEEE 802.11i no lo permite porque, como hemos visto, define un entorno de RSN, de excelentes características de seguridad.

7.7.5 Asociación

Una vez completado el proceso de autenticación, las estaciones se pueden asociar a la red para poder comenzar a intercambiar tramas de datos. La asociación permite al sistema de distribución saber cuál es la localización de cada móvil, de tal manera que las tramas que vayan destinadas a las móviles se puedan entregar al AP correcto.

El proceso de asociación lo inicia la móvil, editando una trama de Requerimiento de Asociación, con destino *unicast*, que debe ser respondida por una trama de ACK. En el caso de red de Infraestructura, el AP debe decidir si puede garantizar la asociación, y habitualmente lo hace según el espacio de memoria que debería reservar para el almacenamiento de tramas, tal como lo plantea el Intervalo de Escucha que la móvil le anuncia. Si es capaz de garantizar los recursos, contesta con una trama de Respuesta de Asociación con un código de estado de éxito y el AID elegido para la móvil.

Luego que se ha completado la asociación, el AP debe registrar a la móvil ante la red. Una forma de hacerlo es enviando una trama ARP gratuita, que permite reconocer la MAC de la estación asociada al puerto del *switch* conectado al AP. Este es un proceso propio de una red de Infraestructura y, para que funcione correctamente, queda prohibida la asociación a más de un AP por vez.

En el caso de re-asociación, la móvil se mueve desde un AP viejo a uno nuevo, como se presenta en la Fig. 7.41. Es como un proceso de asociación, pero entre los AP, en la red cableada, deben comunicarse este cambio.

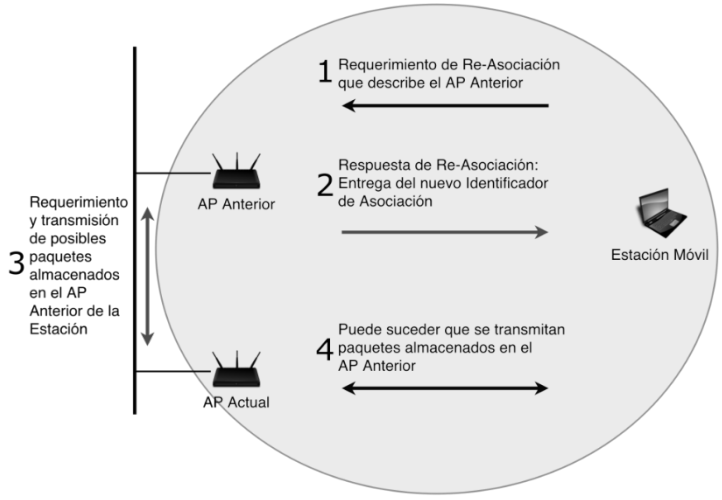


Figura 7.41 - Re-Asociación.

Generalmente el proceso de re-asociación se inicia porque la móvil detecta una señal mejor proveniente de un nuevo AP. Antes de poder re-asociarse, si la móvil no se ha autenticado previamente, debe hacerlo en este momento. La

trama de Requerimiento de Re-Asociación difiere de la de asociación en que contiene un campo con la MAC del AP viejo. El nuevo AP se debe conectar con el viejo AP a través de un protocolo entre puntos de acceso de carácter propietario. Si se comprueba la asociación vieja, entonces se puede proceder a la nueva. Por su parte, el viejo AP debe enviar al nuevo AP cualquier trama que haya almacenado para la móvil, para que le sean entregadas.

7.7.6 Conservación de Potencia

La posibilidad de apagar el transceptor puede llevar a ahorrar considerable cantidad de potencia y, de ese modo, alargar la vida útil de las baterías. Cuando el transceptor se apaga se dice que la estación se va a dormir o se encuentra en modo ahorro de potencia PS. Cuando se prende, la estación se despierta. El objetivo de IEEE 802.11 es minimizar este último tiempo, facilitando la situación de PS, sin sacrificar conectividad.

En redes Infraestructura, al pasar todo el tráfico por el AP, éste se convierte en el punto ideal para el almacenamiento de tramas. Los AP conocen la localización de las móviles, siempre están activos, almacenan el estado de administración de potencia de cada estación asociada a ellos y pueden distinguir si una trama se debe bajar al medio inalámbrico o si se debe almacenar, según el estado del destino de la misma. Periódicamente, el AP debe avisar el almacenamiento de dichas tramas a los destinatarios de las mismas. Este anuncio periódico también colabora con el ahorro de potencia, ya que requiere menos potencia encender el receptor periódicamente, para escuchar los anuncios referidos, que transmitir tramas preguntando de manera periódica. Así, solo se encenderá el transmisor luego de haberse informado de la existencia de tramas para la estación, almacenadas en el AP.

En el Requerimiento de Asociación las móviles anuncian su propio Intervalo de Escucha como el número de intervalos de Beacon en los que se encontrará en modo PS. Este parámetro es usado por el AP para estimar los recursos necesarios para soportar la asociación. De esta manera, al aceptarlo, se le asegura a la móvil que se almacenarán las tramas que lleguen para ella, por lo menos durante el tiempo que se encuentre en modo PS, pudiendo descartar dichas tramas sino pudieron ser entregadas, sólo luego del agotamiento de este tiempo. Este posible descarte de tramas es silencioso: no se notifica a la móvil.

El enlace entre las tramas almacenadas y la móvil particular es su propio AID. Las tramas de *multicast* y *broadcast* almacenadas se asocian al AID "0".

Mediante el TIM en las tramas Beacon se indican las estaciones con tramas en *buffer* del AP. Cuando las móviles despiertan, reciben el Beacon y revisan el TIM. Si tienen tramas en el AP, para poder recibirlas, deben transmitir una trama PS-Poll por cada trama almacenada. A su vez, la trama entregada debe ser objeto de un ACK, antes de poder ser descartada del *buffer*. Si existe más de una trama en el *buffer* del AP, éste lo indica con el bit Más Datos del campo de Control. De este modo, las móviles transmitirán nuevas tramas PS-Poll hasta que el estado de dicho bit sea "0".

Luego de la transmisión de una trama PS-Poll, la móvil debe permanecer despierta hasta concluir la Operación Atómica, o hasta que el bit que corresponde a su AID no aparezca más en el TIM de la trama Beacon. Una vez que todo el tráfico almacenado en el AP se le haya entregado a la móvil o haya sido descartado, ésta puede volver a dormir.

En la Fig. 7.42 se presenta un ejemplo de este procedimiento. Allí el medio está ocupado por un AP y dos estaciones asociadas.

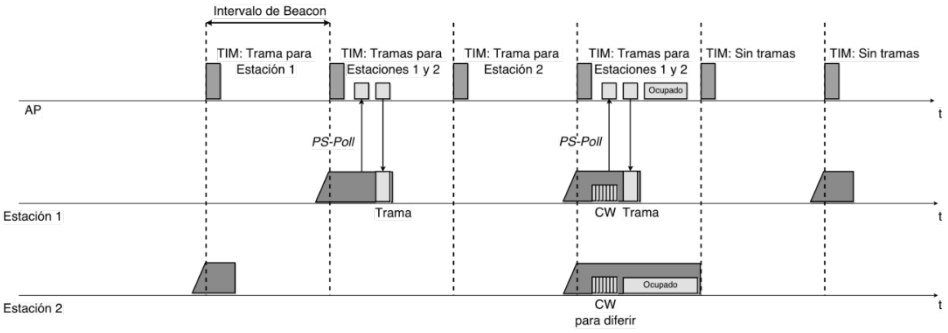


Figura 7.42. Proceso de manejo de potencia.

Se ha escalado cada línea de tiempo con el intervalo de Beacon. Cada Beacon lleva la información del TIM. La estación 1 tiene un Intervalo de Escucha "2", o sea dos intervalos de Beacon, y la estación 2 ha anunciado uno de "3". Se aprecia en la figura los tiempos en que las estaciones despiertan graficado como un proceso de forma inicial del tipo rampa.

En el primer intervalo de Beacon, el TIM anuncia tramas para la estación 1. La estación 2 despierta, pero como en el TIM no se anuncia nada para ella, puede volver a dormir. En el segundo intervalo de Beacon, el TIM indica tramas para las estaciones 1 y 2, pero solo la 1 está despierta, por lo que transmite una trama PS-Poll y recibe la trama almacenada en respuesta. Luego la estación 1 puede volver a dormir. Durante el tercer intervalo de Beacon, ambas móviles duermen.

En el cuarto intervalo, ambas están despiertas y el TIM las señala a las dos. Ambas se preparan para transmitir una trama PS-Poll luego de que termine la ventana de contienda. En este caso, gana la estación 1 y entonces puede recibir la trama en *buffer* para ella. La estación 2 debe diferir, pero debe permanecer despierta hasta poder acceder al medio. En el ejemplo, se marca como *ocupado* la presencia de una tercera estación, que obliga a la estación 2 a permanecer despierta hasta el siguiente TIM. Si el AP se queda sin lugar en el *buffer*, puede descartar las tramas almacenadas para la estación 2, porque ya ha cumplido su compromiso. De hecho, el TIM en el quinto intervalo de Beacon no indica tramas en *buffer* para la estación 2 y esta puede retornar entonces al modo PS.

IEEE 802.11 establece que los AP utilicen una función de envejecimiento para determinar cuándo las tramas en *buffer* se pueden descartar. Cuánto tiempo

más allá del Intervalo de Escucha se mantendrá la información, es más cuestión de la implementación que del estándar.

Las tramas *broadcast* y *multicast* en almacenamiento no pueden recogerse con el mecanismo de requerimiento, estableciéndose un mecanismo especial para ellas. Se almacenan siempre que una estación asociada al AP se encuentre en modo PS y se lo realiza asociándolas con el AID "0" , como se lo marca en el TIM. Cada BSS tiene asociado un parámetro, conocido como el período DTIM. Luego de cierto número de intervalos de Beacon, se transmite un TIM especial, conocido como DTIM. En la propia trama Beacon existe un elemento contador de intervalos de Beacon que se va disminuyendo hasta llegar a "0" en la trama DTIM. Luego de esta trama, el tráfico *broadcast* y *multicast* se transmite en secuencia, con el bit Más Datos del campo de Control en alto indicando si hay más tramas. La transmisión se hace por CSMA/CA, por lo que puede ser que el AP elija diferir el procesamiento de tramas PS-Poll, hasta luego de haber transmitido las tramas de *broadcast* y *multicast* almacenadas.

O sea que, para recibir tramas de *broadcast* y *multicast*, la estación móvil debe despertarse para las transmisiones DTIM. De todos modos, el estándar no obliga a hacerlo. Por ejemplo, existe un modo de baja potencia extremo, donde se descarta la recepción de este tipo de tramas.

En un IBSS el mecanismo de ahorro de potencia no es tan eficiente como en una red Infraestructura. En estas redes importa más que el transmisor se asegure que el receptor esté presente. De este modo, los receptores no pueden dormir tanto como en las redes de Infraestructura. Al no tener un nodo coordinador, estas redes usan un sistema distribuido y mensajes denominados Mensajes de Indicación de Anuncio de Tráfico (ATIM, Announcement Traffic Indication Messages) para prevenir que otras estaciones duerman. Todas las estaciones en una IBSS deben escuchar el ATIM durante períodos de tiempo especificados luego de las transmisiones de las tramas Beacon.

En las redes IBSS, cuando una estación ha almacenado tramas para otra, puede enviar un mensaje de ATIM como notificación. La Fig. 7.43 se presenta el caso en que una estación A ha almacenado una trama para la estación C, por eso le envía un mensaje *unicast* ATIM durante la ventana de ATIM. El mensaje notifica a la estación C de que no puede entrar en modo PS porque la estación A tiene que entregarle tramas que ha almacenado para ella. En este ejemplo, la estación B sí puede entrar en modo PS, porque no ha recibido un mensaje ATIM. En el caso ATIM *multicast*, este mensaje notifica a un grupo, como se presenta en la misma figura. Entonces, ninguno de los del grupo puede entrar en modo PS.

También, en estas redes se define un tiempo especial, denominado ventana de ATIM, que comienza en el momento en que se espera la transmisión del Beacon y finaliza luego de un período especificado en la creación de la IBSS. Durante esta ventana todos deben permanecer activos y su valor es el único parámetro requerido para crear una red IBSS. Ajustado en "0" evita cualquier modo de ahorro de potencia. No importa si el Beacon se retrasa porque el medio se encuentre ocupado, este tiempo permanece fijo y comienza en el momento que finaliza el tiempo que mide los intervalos entre Beacons. Entonces, para monitorear lo que sucede en la ventana de ATIM, las estaciones deben despertar antes de la transmisión del Beacon.

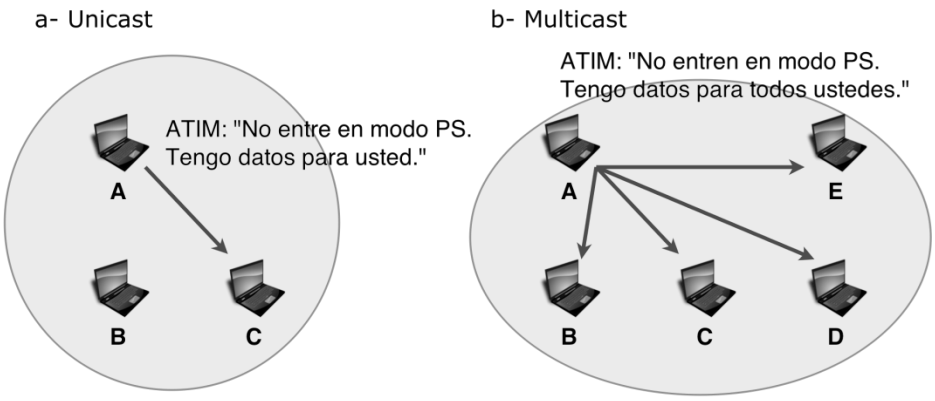


Figura 7.43 - ATIM, redes IBSS.

Puede suceder que una móvil de una red IBSS se encuentre en una de cuatro situaciones posibles: que la estación haya transmitido un ATIM, que haya recibido un ATIM, que ni haya transmitido ni haya recibido ATIM, o que haya transmitido y recibido a la vez un mensaje ATIM. Obviamente, las móviles que transmiten un mensaje ATIM no deben dormir porque es un indicador de que están intentando transmitir tramas almacenadas. Por su parte, las móviles que reciben un mensaje ATIM deben evitar entrar en modo PS, porque existe tráfico que se ha almacenado para ellas y deben recibirlo. En el tercer caso, la estación está activa. Sin embargo, si la móvil no ha recibido ni transmitido un ATIM, tiene permitido ir a modo PS.

Sólo ciertas tramas de control y administración se pueden transmitir durante la ventana de ATIM: Beacon, RTS, CTS, ACK y ATIM. Particularmente, las tramas ATIM *unicast*, precisan de un ACK. Por su parte, las tramas de *broadcast* y *multicast* almacendas, se transmiten luego de la ventana de ATIM, seguidas de las tramas *unicast* que se habían anunciado en el mensaje ATIM. A continuación, las estaciones pueden transmitir tramas a otras estaciones que estén activas. Una estación está activa si ha transmitido un Beacon, un mensaje ATIM o no es capaz de ir a modo PS.

7.7.7 Sincronismo

La información de sincronismo y su distribución es muy importante en redes IEEE 802.11. Además de mantener un reloj local, cada estación en un BSS mantiene una copia de una función, denominada TSF, que no es otra cosa que el reloj local sincronizado con la TSF de cada una de las otras estaciones en el BSS. El TSF se basa en un reloj de $1MHz$.

En las tramas Beacon, periódicamente se anuncia el valor de la TSF al resto de la estaciones. El valor de tiempo que se escribe en el elemento Sello de

Tiempo es el del momento en que el primer bit del mismo se envía a la capa física para su transmisión.

En redes de Infraestructura, los AP son los responsables de mantener el tiempo de la función TSF. Las estaciones lo aceptan como válido, pero agregan un pequeño *offset* al valor recibido, para tener en cuenta el procesamiento local de antena y transceptor. A pesar de que la transmisión de tramas Beacon no es confiable, el sistema es lo suficientemente robusto como para permitir que las estaciones puedan no recibir una trama Beacon pero, aún así, continuar sincronizadas con la TSF global.

Los parámetros referidos también se distribuyen en las tramas de Respuesta de Sondeo, para ayudar a las estaciones que realizan sondeos activos en el proceso de adaptación al BSS.

En el caso de las redes IBSS, no existe un punto de coordinación central sino que el proceso de generación de la trama Beacon es distribuido, y el mantenimiento de la función TSF es una tarea más de este proceso. Si se considera el tiempo dividido en segmentos equivalentes a un intervalo de Beacon, los momentos en los que se supone se debería transmitir una trama Beacon se conocen como Tiempo de Transmisión del Faro Objetivo (TBTT, Target Beacon Transmission Time).

Al aproximarse el tiempo TBTT, todas las estaciones en la red IBSS se preparan para transmitir un Beacon, suspendiendo todas las transmisiones. Todas las estaciones de la IBSS generan un valor aleatorio de ranuras, cuyo rango va desde cero hasta el doble de la mínima ventana de contienda. Luego del tiempo TBTT, todas las estaciones encienden el reloj contador de ranuras y comienzan a disminuirlo. Si reciben una trama Beacon antes de llegar a cero, cancelan la transmisión del Beacon que tenían pendiente.

En redes IBSS, la generación de esta trama tiene mucha relación con la administración de potencia. Las tramas Beacon deben generarse durante los períodos activos cercanos al intervalo de Beacon, con todas las estaciones disponibles para procesarlo. La móvil que transmite la trama Beacon no puede entrar en modo PS hasta el final del siguiente período activo. Esto asegura que al menos una estación se encuentre despierta y pueda responder a requerimientos de estaciones nuevas que sondean en búsqueda de redes.

Como en una red IBSS no existe un reloj centralizado, el objetivo principal es sincronizar todos los relojes al tiempo del reloj más rápido. Por eso, cuando se recibe una trama Beacon, su Sello de Tiempo se ajusta en compensación de los retardos de procesamiento y luego se compara con el TSF local. Si el valor así generado es mayor (posterior) al local, entonces el local se ajusta a éste.

Bibliografía

1. Official IEEE Standards Association web site
   http://standards.ieee.org/about/get/802/802.11.html

2. Official industry association web site http://www.wi-fi.org/
3. Ergen, Mustafa, "802.11 Tutorial". Department of Electrical Engineering and Computer Science, June 2002. http://wow.eecs.berkeley.edu/ergen/docs/ieee.pdf
4. Gast, Matthew, "802.11 Wireless Networks: The Definitive Guide". O'Reilly, 2002.
5. Brenner, Pablo, "A Technical Tutorial on the IEEE 802.11 Protocol". BreezeCom Wireless Communication, 1997. http://www.sss-mag.com/pdf/802_11tut.pdf
6. Mei Yen Cheong, "Management Operations of the IEEE 802.11". S-72.333 Postgraduate Seminar on Radio Communications Helsinki University of Technology, 2004. http://www.comlab.hut.fi/opetus/333/2004slides/topic26.pdf
7. Jui Hung Yeh, Jyh-Cheng Chen and Chi-Chen Lee "WLAN Standards: in particular the IEEE 802.11 family". IEEE Potentials, 2004.

Problemas

1. Compare el método de acceso al medio de IEEE 802.3 con el de IEE 802.11. ¿Qué es la ventana de contienda en el método de acceso al medio 802.11? ¿Cómo se determina y modifica?

2. Mencione las posibilidades de movilidad permitidas en una WLAN.

3. Explique los principales desafíos relacionados con la comunicación en redes WLAN y las funcionalidades agregadas por el protocolo IEEE 802.11 para mitigarlos.

4. En un sistema de distribución existe un protocolo que no se encuentra estandarizado ¿Cuál es? ¿Por qué supone usted que esto es así?

5. En el campo NAV de la primera trama de una operación atómica transmitida con protección contra nodo oculto, se anuncia un tiempo equivalente a 257,4 µseg. Obtener el tamaño en bytes de la trama de datos transmitida, sabiendo que la red es del tipo 802.11g, funcionando a la velocidad máxima, con los siguientes parámetros característicos: ranura de 20 µseg, SIFS de 10 µseg, tamaño de la trama RTS de 24 bytes.

6. Un BSS depende de un punto de acceso (AP) cuya dirección MAC inalámbrica es 00:13:46:fd:b1:5b, la MAC cableada es 00:1f: d0:19:50:c0. Se genera una trama desde una laptop con MAC 00:19:7d:ef:79:e6, dirigida a un Servidor en la red cableada cuya MAC es 00:1f: 20:a1:c2:44. ¿Cuáles serán las direcciones en los campos Address 1, 2 y 3 de la trama transmitida?

7. ¿Por qué las tramas PS-Poll suponen un NAV implícito?

8. Enumere los cuatro tipos de tramas de Control y explique la funcionalidad asociada a cada una de ellas.

9. Nombre al menos cinco elementos que transporta una trama Beacon, explicando la funcionalidad de cada uno.

10. ¿Durante cuánto tiempo un AP almacena tramas para una máquina móvil? ¿Cómo anuncia esta situación? ¿Qué significa una valor DTIMperiod=3 y DTIMcount=0?

11. Explique los parámetros que caracterizan la fase de exploración en la búsqueda de una red WLAN.

12. ¿Cuál es la información más relevante intercambiada en un proceso de asociación y para qué sirve? ¿Cuándo es necesario re-asociarse y cómo es este mecanismo? ¿Existe alguna diferencia con el proceso de asociación?

13. Nombre y explique al menos dos funcionalidades que contribuyen al modo ahorro de potencia.

14. Enumere al menos 5 parámetros de configuración para una red inalámbrica.

15. ¿Cómo se genera un Beacon en redes ad-hoc?

16. Indicar si las siguientes afirmaciones son Verdaderas o Falsas. Justificar su respuesta.

a) En las redes inalámbricas se agrega el concepto de Operación Atómica para contrarrestar el problema de *multipath fading*.
b) El problema del nodo oculto se soluciona agregando el campo NAV en el encabezado de la trama.
c) En 802.11b si un dispositivo transmite una trama de 1500 bytes a 11 Mbps, tomando una ventana de contienda de 15 slots en promedio (20us/slot), esto agrega cierto tiempo de demora en el acceso haciendo que la velocidad efectiva para este volumen de datos se vea reducida en alrededor de 30%.
d) Una red celular GSM es semejante a una red WiFi. La red GSM es una red de células, cada una con una antena base y radio de cobertura variable desde decenas de metros a decenas de kilómetros, en las bandas de 900MHz y 1800 MHz. En la red GSM es posible la movilidad entre células (*roaming*) de amplia cobertura. La red GSM ofrece transferencia de datos a 57.6 kbps. Cada canal GSM tiene 200 kHz de ancho de banda y, a su vez se divide en 8 slots de tiempo, asignándose uno por usuario para el acceso al medio.
e) Instalando una red WiFi se pueden enfrentar una serie de problemas relacionados con la presencia de electrodomésticos en la banda de operación, interferencias por el uso del mismo canal por parte de una red cercana, señales débiles por la presencia de obstáculos, e ingreso de usuarios no autorizados. En este sentido, una configuración aconsejable sería:
- Dirección IP por DHCP.
- Canal de Radio: 11.
- Potencia de Transmisión: 100mW.
- SSID: Mi_Red.
- Velocidad: 11 Mbps.
- Intervalo de Beacon: 10ms.
- RTS / CTS: 2346.
- Fragmentation: 2346.
- Cifrado: WEP.
- Autenticación: Clave Compartida.

f) Al ingresar en una red WiFi, un usuario recibió una serie de datos en una trama de Beacon que examinó con un sniffer, concluyendo que esta trama contenía errores. Los principales campos de la trama fueron los siguientes:
- Tipo/Subtipo Beacon frame (8)

- Frame Control: Versión (11) Tipo (00) Subtipo (8) ToDS/FromDS (10) MF (1) Retransmisión (1) PS (1) Más Datos (1) WEP (1) Orden (0)
- Duration: 1800
- Dirección Destino: Broadcast (ff:ff:ff:ff:ff:ff)
- Dirección Fuente: D-Link_3c:20:f9 (00:11:95:3c:20:f9)
- BSSID: D-Link_3c:20:f9 (00:11:95:3c:20:f9)
- Número de Fragmento: 0
- Número de Secuencia: 3790

g) Una trama PS-Poll lleva en su campo NAV un valor decimal igual a 2500 anunciando que ocupará el medio durante 2500 useg.

h) En una trama IEE 802.11 se leen tres direcciones, de las que se puede inferir que se trata de una trama con el bit TO DS en alto. Las direcciones en cuestión son (en el mismo orden de presentación en la trama): BSS Id (00:13:46:fd:b1:5b) Dirección Fuente (00:19:7d:ef:79:e6) Dirección Destino (00:13:46:fd:b1:5b).

i) El Intervalo de Beacon es un dato que se anexa junto al Sello de Tiempo en una trama Beacon para mantener el sincronismo de la red.

j) Una máquina que salga del estado de ahorro de potencia y observe una trama Beacon con su propia identificación en el TIM, no puede volver a dormir hasta no solicitar las tramas que el AP haya almacenado para ella.

CAPÍTULO VIII

Capa Física WLAN

Una vez que instalada una red LAN cableada, si la misma ha sido diseñada teniendo en cuenta las reglas que surgen de las especificaciones, se comporta de manera predecible y no es complicado actualizarla, en cuanto a su capacidad, con la tecnología actual de soporte basada en dispositivos switch.

En el caso de las redes LAN inalámbricas, la cuestión es mucho más dinámica y complicada. La propagación de ondas de RF adolece de los problemas típicos de la transmisión no guiada: interferencias, atenuación, zonas de sombras donde la señal no puede penetrar debido a la presencia de objetos en el camino de propagación y por la interferencia en el camino entre las señales directas y las reflejadas por objetos.

Algunos de los problemas mencionados son dependientes de la frecuencia de trabajo. Por ejemplo, las señales de mayor frecuencia se atenúan más rápido y son absorbidas fácilmente por objetos presentes en el camino de propagación. Otros problemas se relacionan con las restricciones generadas por las regulaciones de la parte del espectro donde se encuentran asignadas las WLAN. En estos casos, mejorar la velocidad se convierte en un problema cada vez más complejo en términos de técnicas de modulación, codificación y diversidad.

Habiéndose comprendido la funcionalidad que el estándar IEEE 802.11 le otorga a la capa MAC para que la misma colabore lo más eficientemente posible con la comunicación en un medio tan poco predecible, es muy interesante conocer las técnicas ideadas para solucionar los problemas de transmisión en la capa física.

Este capítulo introduce los problemas más comunes de la transmisión inalámbrica con sus posibles soluciones. Por este motivo, la primera parte describe los aspectos más distintivos del problema de desvanecimiento Rayleigh que afecta a sistemas de comunicación móvil, mencionando los métodos apropiados para mitigar la degradación resultante.

El capítulo se completa con una descripción detallada de cada una de las capas físicas IEEE 802.11 estandarizadas conocidas hasta hoy.

8.1 Fading de Pequeña Escala – Degradación y Efectos

En capítulos previos hemos explicado el problema particular de los canales inalámbricos en entornos móviles respecto de la propagación multi-camino. También hemos presentado el modelo estadístico característico de la envolvente de una señal afectada por este fenómeno de desvanecimiento. Restaría abordar el tema de los parámetros que caracterizan los efectos de desparramo en el tiempo y de variación del canal, su relación con la velocidad de transmisión y la degradación que se produce en consecuencia en cada caso. Justamente, a partir de un modelo de canal como el estudiado y considerando las componentes en recepción como no correlacionadas, Bello definió determinadas funciones a partir de las cuales es posible caracterizar el canal mediante parámetros apropiados.

Comprendidos dichos parámetros resultará más sencillo posteriormente entender las diversas técnicas de mitigación del desvanecimiento.

8.1.1 Desparramo en el Tiempo y Selectividad en Frecuencia

En la Fig. 8.1 se presenta una de las funciones definidas por Bello: el perfil de intensidad *multipath* $S(\tau)$ en función del retardo τ. La función $S(\tau)$ se refiere a la variación de la potencia promedio recibida a partir de un impulso transmitido, en función del retardo. En este caso, el retardo representa el retardo de propagación que excede al retardo del arribo de la primera señal en el receptor. A parir de este perfil, es posible definir el retardo en exceso T_m como el intervalo de tiempo total durante el cual llegan reflexiones con contenido de energía significativo, por ejemplo 10 o 20 dB por debajo de la componente mayor. Por supuesto que, para un sistema ideal con retardo en exceso nulo, $S(\tau)$ sería un impulso ideal con peso igual a la potencia total promedio de la señal recibida.

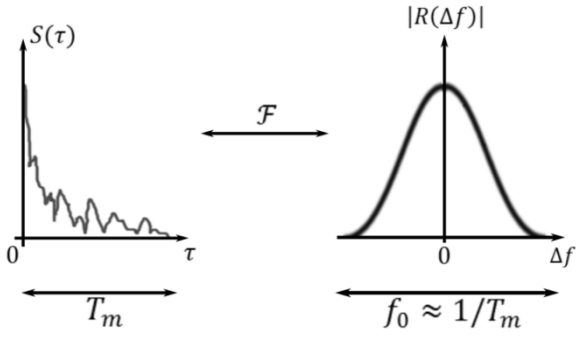


Figura 8.1 - Perfil de Intensidad y Correlación en Frecuencia

En el canal de *fading*, la relación entre el retardo en exceso T_m y la duración del símbolo transmitido T_s deviene en diferentes categorías de

degradación. El efecto más importante del desparramo del retardo es la interferencia entre símbolos ISI.

Si la duración del símbolo es lo suficientemente mayor que el retardo en exceso $T_m < T_s$, típicamente 10 veces sería lo ideal, se podría esperar un canal libre de ISI o canal no selectivo en frecuencia (plano). En este caso, los componentes multi-camino arriban al receptor dentro de la duración del símbolo, sin provocar interferencias con símbolos adyacentes, aunque el efecto de degradación puede provocar una disminución significativa de la relación SNR. Es posible mitigar el efecto mediante alguna técnica de diversidad y usando códigos de corrección de error.

Por otra parte, si $T_m > T_s$ se dice que el canal exhibe un *fading* selectivo en frecuencia que provoca la misma clase de distorsión que el ISI causado por filtrado. Es posible aplicar diferentes técnicas para mitigar este efecto, ya que se pueden discriminar las componentes multi-camino del lado receptor.

El fenómeno se puede observar también en el dominio de la frecuencia, tal como se presenta en la Fig. 8.1. La Transformada de Fourier de la función $S(\tau)$ es una función $|R(\Delta f)|$, conocida como función correlación de frecuencia espaciada. $|R(\Delta f)|$ representa la correlación entre las respuestas del canal a dos señales como una función de la diferencia de frecuencia entre ambas. Se puede pensar como la función de transferencia en frecuencia del canal, que es como ver la manifestación de desparramo en el tiempo como resultado de un proceso de filtrado. Conocer $|R(\Delta f)|$ ayuda a responder la pregunta sobre cuál es la correlación entre señales recibidas que están separadas en frecuencia por Δf. Se puede medir por la transmisión de un par de sinusoides con esa diferencia en frecuencia, correlacionando sus respuestas y repitiendo la medición muchas veces mientras se va aumentando la separación en frecuencia.

En ancho de banda de coherencia f_0 es una medida estadística del rango de frecuencias sobre el cual el canal permite el paso de componentes aproximadamente con la misma ganancia y fase lineal, es decir el ancho de banda con posibilidades de correlación fuerte en amplitud. Así, todas las componentes de frecuencia de una señal dentro de esta banda sufrirán desvanecimiento en simultáneo o no lo sufrirán. Se verifica la relación $f_0 \cong {}^{1}/_{T_m}$.

El retardo en exceso máximo T_m , por sí mismo, no necesariamente es el mejor indicador de cómo se comportará un sistema específico sobre un canal, porque otro canal, con el mismo valor de T_m puede exhibir un perfil diferente de intensidad de la señal $S(\tau)$. Por este motivo, una medida más útil del desparramo del retardo es su valor raíz cuadrático medio (*rms*), $\sigma_t = (\overline{\tau^2} + (\bar{\tau}^2)^{1/2}$, donde $\bar{\tau}$ es el retardo en exceso promedio y $\overline{\tau^2}$ es su valor cuadrático medio. No existe una relación exacta entre f_0 y σ_t pero se puede derivar una a partir del análisis de señales mediante transformada de Fourier. La relación más popular es menos estricta en cuanto al valor de correlación, resultando un valor $f_0 \cong {}^{1}/_{5\sigma_\tau}$. Los valores de f_0 y σ_t se relacionan solamente con las características *multipath* del canal, no con la velocidad de señalización. Esta última sólo determina el ancho de banda de transmisión W.

Se dice que un canal es selectivo en frecuencia si $f_0 < {}^{1}/_{T_s} \sim W$, siendo W el ancho de banda de la señal. En este caso, existirán componentes de la señal que caerán fuera del ancho de banda de coherencia del canal y serán afectadas de manera diferente (e independiente) de aquellas componentes dentro del ancho de banda de coherencia.

En el caso $f_0 > W$, el *fading* es no selectivo en frecuencia o *flat fading*, no se introduce distorsión ISI por el canal, pero se genera una degradación de la relación SNR. De este modo, f_0 fija un límite superior en la velocidad de transmisión a la que se podría llegar sin necesidad de ecualizadores.

De todas maneras, en el caso de *flat fading*, puede suceder que, por momentos, haya *fading* selectivo debido al cambio en el canal, producto del movimiento. Una de las consecuencias más peligrosas en estos casos es la pérdida de componentes de baja frecuencia, con su consiguiente ausencia de valores confiables para establecer sincronismo o muestrear la fase de la portadora en los momentos precisos. Lo cierto es que a medida que f_0 sea bastante más grande que W, o T_m bastante menor que T_s, se corre menos riesgo de que el canal presente por mucho tiempo esas condiciones.

8.1.2 Variación en el Tiempo y Velocidad del Desvanecimiento

Como se ha explicado, el canal es variable en el tiempo debido al movimiento relativo entre transmisor y receptor, o al movimiento de objetos entre ellos. Suponiendo que todos los generadores de *scatter* que forman el canal son estacionarios, siempre que el movimiento cese, la amplitud y fase en recepción permanecerán constantes y el canal aparecerá como variable en el tiempo sólo cuando se reanude el movimiento. El hecho de que sea variante en el tiempo es lo mismo que decir que es variante en el espacio. La Fig. 8.2 muestra la función $R(\Delta t)$ de correlación en el tiempo espaciado, que especifica la correlación entre la respuesta del canal a una sinusoide transmitida en un instante y la respuesta a la misma sinusoide transmitida un tiempo Δt posterior. Se trata de la función dual de $R(\Delta f)$ ya que describe una relación matemática similar, pero en términos de otros parámetros.

Tal como se indica en la figura, el tiempo de coherencia T_0 es una medida del promedio del tiempo que se espera que el canal permanezca invariante. Para medir la correlación, en este caso se usan señales de banda angosta o sinusoides de una armónica. La función de correlación y el tiempo de coherencia nos dan una idea de la rapidez del *fading*. Por ejemplo, para un canal ideal invariante en el tiempo, donde los protagonistas no se mueven, la respuesta del canal tendrá alta correlación para cualquier Δt, y $R(\Delta t)$ será una función constante.

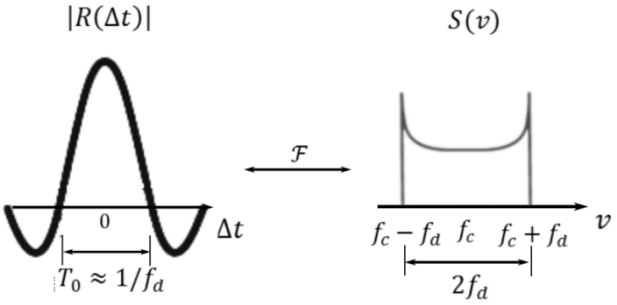


Figura 8.2 – Correlación en el Tiempo y Espectro Doppler

Se describe como degradación *fast fading* cuando se cumple $T_o < T_s$, que se corresponde al caso en que el tiempo durante el cual el canal se comporta de manera correlacionada es corto comparado con la duración del símbolo. O sea que sería esperable que el carácter de *fading* del canal cambie varias veces mientras se transmite un símbolo, provocando distorsión similar al ISI porque los componentes de la señal recibidos no se encuentran altamente correlacionados en el tiempo. Esto genera distorsión que termina traduciéndose en una pérdida de SNR que afecta la P_{be} . Los pulsos distorsionados también causan problemas de sincronismo y dificultades en las definiciones al momento de diseñar los filtros adaptados.

Por otro lado, cuando $T_o > T_s$, se habla de una degradación lenta o *slow fading*, donde no existirá la distorsión señalada en el caso anterior. Se puede esperar que el canal permanezca invariante durante el tiempo de transmisión de cada símbolo. La degradación principal en este caso será el deterioro de la SNR, tal como ocurre en el caso de *flat fading*.

La Fig. 8.2 también muestra la densidad espectral de potencia *Doppler* $S(v)$, graficada en función del corrimiento en frecuencia *Doppler* v. Esta función muestra la manera en que una única frecuencia o impulso en el espectro (sinusoide pura) es desparramada en frecuencia cuando atraviesa el canal y es la función dual de $S(\tau)$. El modelo que se sostiene para canales inalámbricos dentro de edificios asume $S(v)$ plano, aunque en ambientes de alto nivel de desvanecimiento, la función tiene la forma de un bowl tal como se representa en la figura.

Los extremos de valores altos se producen cuando el *scatter* se coloca directamente delante o detrás de la antena en movimiento. En este caso la magnitud del corrimiento en frecuencia es $f_d = {}^{V}/_{\lambda}$, donde V es la velocidad relativa y λ la longitud de onda de la señal. La frecuencia f_d es positiva cuando transmisor y receptor se mueven el uno hacia el otro, y negativa cuando se alejan. El hecho de que componentes Doppler que arriben exactamente a 0º y 180º tengan una densidad espectral de potencia infinita, no es un problema, dado que la probabilidad de que las componentes arriben a exactamente esos ángulos es cero.

$S(v)$ y $R(\Delta t)$ son pares transformados. Conocer $S(v)$ nos permite deducir cuánto ensanchamiento espectral se le impone a la señal en función de la velocidad de cambio en el estado del canal. El ancho del espectro de potencia Doppler se conoce como desparramo Doppler o ancho de banda de *fading* y se denota como f_d y verifica $f_d \sim {}^{1}/_{T_o}$, por eso también se toma como la relación de cambio típica del canal.

Se producirá una degradación del tipo *fast fading* cuando se verifique $f_d > W$ (o $T_s > T_o$) y *slow fading* en caso contrario, $f_d < W$. Para evitar la distorsión causada por *fast fading*, se debe procurar que el canal exhiba *slow fading*, es decir que la velocidad de señalización debe exceder la del *fading* del canal. También se puede decir que f_d impone un límite a la velocidad inferior del canal.

Debido a la dispersión, hemos visto que el ancho de banda de coherencia f_0 implica un límite superior en la velocidad de la señal $f_o > W$ para que no exista distorsión por selectividad en frecuencia. Similarmente, debido al desparramo Doppler, la velocidad de cambio del canal f_d, pone un límite inferior en la velocidad de la señal para que por encima de la misma no haya distorsión por *fast fading*, $f_d < W$. Para mitigar los efectos de *fast fading*, lo mejor sería $W \gg f_d$. De no cumplirse la condición, el efecto se traduce en tasas de errores demasiado elevadas, imposibles de sortear aumentando la SNR.

Veamos un ejemplo numérico de los parámetros presentados hasta aquí. Para entornos en el interior de edificios, la variación en el tiempo del canal se deberá principalmente al movimiento respecto de los objetos que, típicamente, se producirá a la velocidad a la cual la persona se traslada, estimada en 1.2 ${}^{km}/_{h} = 0.33\ {}^{m}/_{s}$. Este movimiento generará cambios en la amplitud de la señal recibida debido a los cambios en los caminos de reflexión existentes entre transmisor y receptor. Si se considera la frecuencia de trabajo de 2.4 GHz, típica de muchos entornos WLAN, la longitud de onda correspondiente sería del orden de $\lambda = {}^{c}/_{f_c} = {}^{3x\ 10^8 m/seg}/_{2.4\ GHz} = 0.125\ m$.

Si se considera la velocidad del movimiento, el desparramo Doppler en frecuencia se puede calcular como $f_d = {}^{v}/_{\lambda} = {}^{0.33}/_{0.125} = 2.64\ Hz$. Este valor representa el peor corrimiento en frecuencia debido al propio movimiento. Su inversa permite estimar el tiempo durante el cual el canal permanecerá esencialmente invariante en el tiempo $T_0 = {}^{1}/_{f_d} = 0.378\ seg$. Desde el punto de vista de la transmisión de tramas en una WLAN, se impone un límite al tiempo de duración de las mismas cuando se trabaja en esta banda. Una trama típica de $1500\ bytes$ en 802.11b, transmitida a la velocidad máxima, tendrá un tiempo de duración de ${}^{1500x8\ bits}/_{11\ Mbps} = 1.090\ mseg$, mucho menor que el valor calculado antes.

En cuanto al desparramo del retardo, se podrían estimar los valores mínimo y máximo en un entorno interior a un edificio, considerando un rayo de línea directa de $20\ m$ y el camino de propagación más largo, por reflexión, de

$150\,m$. Estos valores se traducen en parámetros $T_{m,min} = {20\,m}/{3x10^8 m/seg} = 66.7\,nseg$ y $T_{m,máx} = {150\,m}/{3x10^8 m/seg} = 500\,nseg$. Se puede interpretar que el desparramo del retardo, debería ser inferior a $T_m < 500\,nseg - 66.7\,nseg = 433.3\,nseg$. Se puede interpretar el resultado razonando que, para asegurarse de trabajar con un canal uniforme en cuanto a su respuesta en frecuencia al *fading*, la duración para los símbolos transmitidos deberá ser significativamente superior a este valor.

8.2 Mitigación del Fading

La Fig 8.3 presenta curvas de P_{be} en diferentes condiciones. La curva de más a la izquierda representa el comportamiento esperado en el caso de cualquier sistema de modulación apropiado y un canal contaminado con AWGN. La curva del medio representa el comportamiento en caso de degradación por pérdida de SNR, característico del caso de degradación *flat fading* o de *slow fading*, cuando no hay componente de rayo directo. La curva superior, donde P_{be} alcanza valores cercanos a 0.5, muestra los efectos distorsivos severos del *fading* selectivo en frecuencia o *fast fading*.

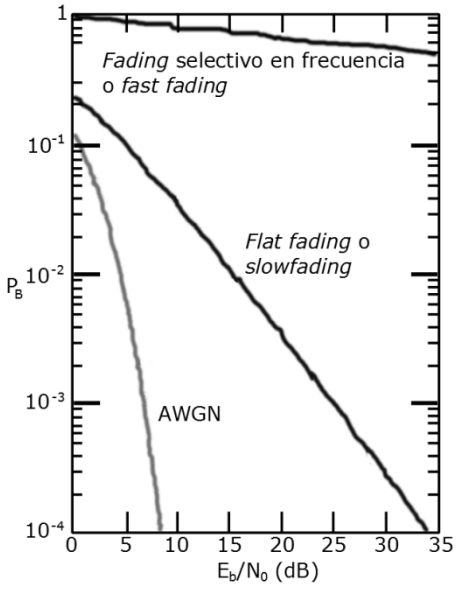


Figura 8.3 - P_{be} en diferentes condiciones.

La forma de contrarrestar estos efectos se denomina mitigación y depende del tipo de *fading* presente. La idea es acercar la curva de P_{be} de arriba a la del medio. A partir de allí, para obtener resultados cercanos a la curva de

AWGN, se deben adicionar técnicas de diversidad y códigos para corrección de errores muy poderosos.

La Tabla 8.1 presenta un resumen de los tipos de mitigación básicos en cada caso.

Tabla 8.1 - Tipos básicos de mitigación.

Para combatir distorsión	Para combatir pérdida en SNR
Distorsión por canal selectivo en frecuencia ▪ Ecualización Adaptiva ▪ Espectro Esparcido DS o FH ▪ OFDM ▪ Portadora Piloto	**Flat fading y slow fading** ▪ Diversidad para tener estimaciones no correlacionadas adicionales. ▪ Codificación FEC
Distorsión por fast fading ▪ Modulación Robusta ▪ Redundancia de la señal para aumentar la velocidad. ▪ Codificación y entrelazamiento	**Tipos de Diversidad** ▪ Tiempo (entrelazamiento) ▪ Frecuencia (expansión de ancho de banda, SSDS o SSFH con receptor rake) ▪ Espacial (antenas) ▪ Polarización

8.2.1 Mitigación Para Combatir la Distorsión por Fading Selectivo en Frecuencia

Una técnica de **Ecualización** puede combatir el ISI introducido por *fading* selectivo en frecuencia, moviendo la curva superior de la Fig. 8.3 hacia abajo, es decir, mejorando la Pbe para la misma SNR. La ecualización implica restaurar la energía del símbolo que ha sido dispersada dentro de su intervalo original de tiempo, para contrarrestar la respuesta en frecuencia del canal. En la práctica se trata de un filtro tal que, la combinación filtro y canal, presenta una respuesta plana y de fase lineal. La fase lineal se logra diseñando el filtro como el complejo conjugado de la versión espejada en el tiempo del pulso dispersado, es decir como filtro adaptado. Para que el ecualizador pueda adaptarse a los cambios en el tiempo del canal, se debe usar un tipo de ecualización adaptiva, que ofrece mitigación y diversidad al mismo tiempo.

Un Ecualizador de Decisión por Realimentación (DFE, Decision Feedback Equalizer) posee dos secciones. Una sección es de realimentación hacia adelante, con la forma de un filtro lineal transversal cuya cantidad de etapas y ganancias se eligen para combinar de manera coherente toda la energía actual del pulso. La otra sección es de realimentación hacia atrás, que remueve la energía que permanece de símbolos previamente detectados. Funciona de tal manera que,

una vez que un símbolo ha sido detectado, el ISI que induce sobre símbolos posteriores, se puede estimar y sustraer antes de la detección de los mismos. Los ajustes de las ganancias se pueden realizar para cumplir un criterio, por ejemplo el de Error Cuadrático Medio (MSE, Mean Squared Error) es uno de los más usados.

Otro tipo de ecualizador, denominado Ecualizador de Estimación de Secuencia de Máxima Probabilidad (MLSE, Maximum Likelihood Sequence Estimation) detecta por secuencia más probable, en vez de por símbolo más probable. Típicamente se implementan siguiendo la decodificación ofrecida por el algoritmo de Viterbi para minimizar la probabilidad de una secuencia equivocada.

Las **Técnicas de Espectro Esparcido** son aplicables para rechazo de interferencia, justamente porque la distorsión por ISI que genera el *fading selectivo en frecuencia* es una forma de interferencia. La técnica DSSS también se puede usar en los casos particulares de *flat fading* que ocasionalmente muestran distorsión selectiva en frecuencia cuando el nulo de la función transferencia del canal ocurre en el centro de la banda. DSSS mitiga este tipo de distorsiones por que incrementa el ancho de banda sobre muchos lóbulos de la respuesta selectiva en frecuencia.

FHSS se puede usar para mitigar el *fading selectivo en frecuencia*, siempre que la velocidad de salto sea por lo menos la misma que la de los símbolos. Se evita la pérdida *multipath* por cambios rápidos en las bandas de frecuencia de transmisión, evitando así la interferencia por cambio de la posición de la banda de recepción, antes de la llegada de la señal *multipath*.

En **OFDM**, la señal a transmitir, de alta velocidad, se divide en N grupos de símbolos, de tal manera que cada grupo contiene una secuencia de símbolos de menor velocidad que la original, justamente reducida en $1/N$. La banda de transmisión se divide en N sub-bandas de portadoras ortogonales, cada una modulada por un grupo diferente. El objetivo es reducir la velocidad de la señal sobre cada portadora para que sea menor frente al ancho de coherencia del canal f_0, verificándose la condición $W < f_0$ en cada canal.

La técnica denominada **Señal Piloto** se refiere al caso de una señal agregada con la que se pretende mejorar la detección coherente. Se puede implementar en el dominio de la frecuencia como un tono dentro de la banda, o en el dominio del tiempo, como una secuencia piloto que puede también proveer información del estado del canal y, de este modo, mejorar el comportamiento en el caso de *fading*.

8.2.2 Mitigación Para Combatir la Distorsión por Fast Fading

En este caso, funcionará cualquier esquema de **Modulación robusta,** no coherente o diferencialmente coherente, que no requiera enganche de fase. Se trata de evitar la utilización de técnicas de modulación de amplitud, debido a que el desvanecimiento puede causar grandes variaciones en esta característica de la señal.

Adicionar redundancia implica aumentar la velocidad de los símbolos $W = \frac{1}{T_s}$, para que el ancho de banda sea mayor que el de fading $W > f_d \approx \frac{1}{T_0}$. El agregado de entrelazado en muchos esquemas de corrección de errores también ayuda, porque permite reducir la relación E_b/N_0 necesaria.

En los casos de *fast fading* y *fading selectivo* ocurriendo en simultáneo, OFDM puede combatir el *fading selectivo*, pero el *fast fading* lo degradará por que el desparramo Doppler corrompe las relaciones de ortogonalidad entre sub-portadoras. Se pueden usar **Técnicas Especiales de Filtrado** en estos casos, para formateo en el dominio del tiempo y extensión de la duración, con el objeto de reducir las bandas laterales espectrales del conjunto de señales y ayudar a preservar la ortogonalidad. El proceso introduce ISI conocido e interferencia de canal adyacente que luego se puede remover con un ecualizador de procesamiento posterior y un filtro de cancelación.

8.2.3 Mitigación Para Combatir la pérdida de SNR

Luego de la aplicación de las técnicas de mitigación para prevenir distorsiones de la señal, el siguiente paso sirve para aproximarnos a una curva de P_{be} de características similares a AWGN. Los métodos disponibles que sirven para proveer al receptor de muestras no correlacionadas de la señal se conocen con el nombre de diversidad (*diversity*). La cuestión acá es la de-correlación, porque no serviría de mucho tener copias adicionales igualmente pobres en calidad.

Las **Técnicas de Diversidad en el Tiempo** permiten transmitir la señal sobre cierta cantidad de intervalos de tiempo diferentes con separación al menos de T_0 entre ellos. Por ejemplo, el entrelazado de datos (*interleaving*), a menudo usado en combinación con técnicas de control de error, es una forma de diversidad en el tiempo. Con esta combinación se logran excelentes resultados frente al caso de errores en ráfaga.

Cuando se usan **Técnicas de Diversidad en Frecuencia** se transmite la señal sobre cierto número de portadoras diferentes con separación en frecuencia de, por lo menos, f_0. La expansión del ancho de banda es una forma de este tipo de diversidad. Llevando el ancho de banda de la señal W a un valor mucho mayor que f_0, se provee al receptor con varias réplicas de la señal independientes del *fading*. El problema es que podría generarse distorsión selectiva en frecuencia que debe mitigarse con alguna forma de ecualización. OFDM y espectro esparcido pueden contarse entre estas técnicas.

Las **Técnicas de Diversidad Espacial** incluyen el uso múltiples antenas transmisoras y/o múltiples antenas receptoras. Si existen varias antenas receptoras, seguramente el esquema es combinado con alguna otra técnica de diversidad. Se usa procesamiento de señales para elegir la mejor salida de entre todas las antenas o para combinar coherentemente todas las salidas.

En el caso de **Diversidad por Polarización** se transmiten y reciben múltiples versiones de una señal por medio de antenas con diferente polarización.

Las técnicas más comunes para combinar señales provenientes de algún esquema de diversidad son: selección, realimentación, relación máxima e igual ganancia. En los casos de diversidad espacial, la técnica de selección consiste en el muestreo de las señales de las antenas receptoras, para enviar la mejor al demodulador. Es una técnica sencilla pero, al no hacer uso de todas las señales disponibles, no es óptima.

En el caso de realimentación, las señales recibidas se muestrean siguiendo una secuencia fija hasta encontrar una que exceda un umbral previamente establecido. La señal elegida continúa siendo tal hasta que su valor no caiga por debajo del umbral, en cuyo caso se reanuda el muestreo. También se trata de una técnica sencilla de implementar, aunque no es la de mejor performance.

La combinación de relación máxima consiste en pesar todas las señales recibidas de acuerdo con la SNR correspondiente a cada una, y luego sumarlas. Esta técnica se acompaña con algoritmos para ajustes de ganancias y retardos, similares a los usados en el caso de ecualizadores. La ventaja es que se puede generar una relación SNR promedio aceptable, aún cuando ninguna de las componentes posea una relación aceptable. La combinación de igual ganancia es similar, pero el peso para todas las señales es el mismo, siendo su eficiencia un poco menor que en el caso de relación máxima.

8.3 Capa Física IEEE 802.11

La versión original IEEE 802.11 fue publicada en 1997. El estándar especificaba dos velocidades de transmisión de 1 y 2 $Mbps$, y definía técnicas específicas a nivel de capa física para lograr estas velocidades.

Como se ha mencionado, los mayores problemas a nivel físico provienen de la propia naturaleza del medio de transmisión y se relacionan con interferencias, reflexiones y asignación de ancho de banda. Debido a la complejidad de la transmisión, la capa física fue dividida en dos subcapas: PLCP y PMD. La subcapa PLCP fue diseñada para preparar las unidades transmitidas o recibidas por las técnicas de acceso definidas a nivel MAC. La subcapa PMD se encarga de transmitir las señales moduladas y demodular las señales recibidas por medio de la antena.

Para enfrentar los problemas de interferencia, el estándar original IEEE 802.11 propuso tres técnicas posibles para transmitir sobre una red inalámbrica: FHSS, DSSS y transmisión por infrarrojo. La propuesta de transmisión en la banda Infrarroja, no tuvo éxito y no fue implementada comercialmente. Como se ha comentado, inicialmente la banda de trabajo elegida fue la banda de RF no licenciada de $2.4GHz$, conocida como ISM. La Federación de Comunicaciones de Estados Unidos permite un máximo de potencia transmitida de $1W$ en dicha banda, pero los equipos 802.11 raramente superan los 100 mW.

IEEE 802.11, en su versión DSSS, usa el código Barker (en hexadecimal $0x5BB$) como secuencia pseudoaletaoria de 11 $bits$ a combinar con cada dato transmitido. Así, los datos a 1 $Mbps$ se modulan con el código Barker generando

una secuencia a 11 $Mbps$ que luego se modula en DBPSK para su transmisión final en un ancho de banda de 22 MHz. Por su parte, si se requiere transmitir a mayor velocidad, los datos a 2 $Mbps$ se modulan con el mismo código Barker, generando una secuencia a 22 $Mbps$, que luego se modula en DQPSK para su transmisión final en un ancho de banda de 22 MHz. Es decir que, en cualquier caso, el ancho de banda de transmisión se establece en 22 MHz.

IEEE 802.11b deriva de la versión DSSS de IEEE 802.11. En 802.11b, los datos pueden generarse a 5.5 $Mbps$ y a 11 $Mbps$, pero un esquema especial de modulación, conocida como Modulación de Código Complementario (CCK, Complementary Code Keying), a la que luego sigue una modulación DQPSK, permite mantener el ancho de banda de transmisión en 22 MHz. La capa física se conoce con el nombre HR/DS o HR/DSSS, donde la sigla HR significa alta velocidad. También 802.11b definió una extensión opcional, conocida como Codificación Convolucional Binaria de Paquete (PBCC, Packet Binary Convolutional Coding) para poder transmitir velocidades de 22 $Mbps$ y 33 $Mbps$, aunque la mayoría de los equipos en el mercado no lo implementan.

En la banda ISM existen 14 canales de transmisión, cada uno de 5 MHz de ancho de banda. Dado que la asignación de canales de trabajo para una WLAN ha de ser tal que las redes no se interfieran entre sí, una asignación muy común entre redes próximas es la de los canales 1, 6 y 11. Dado que cada canal tiene asignado un ancho de banda de 5 MHz, estos canales se encuentran separados por bandas de 25 MHz, dando lugar al acomodamiento sin solapamientos del ancho de banda de 22 MHz asignado a cualquier WLAN.

IEEE 802.11a trabaja en la banda de 5 GHz, utilizando 52 subportadoras OFDM para alcanzar una velocidad máxima de 54 $Mbps$. No puede operar en conjunto con equipos del estándar 802.11b ya que estos trabajan en la banda de 2.4 GHz.

IEEE 802.11g fue ratificado en el año 2003, como una evolución de 802.11b. Trabaja también en la banda de 2.4 GHz pero opera a una velocidad teórica máxima de 54 $Mbps$. Es compatible con el estándar 802.11b y utiliza los mismos canales. Para alcanzar las velocidades más altas, incorpora una capa física tipo OFDM pero en la banda de ISM.

IEEE 802.11n es el más moderno de los estándares desarrollados comercialmente. Fue diseñado para superar las velocidades alcanzadas por 802.11g, objetivo que logra utilizando múltiples señales y antenas con una tecnología denominada MIMO.

En este capítulo se revisarán las técnicas utilizadas por cada uno de los estándares mencionados, deteniéndonos en los detalles más relevantes de las respectivas capas físicas.

8.3.1 IEEE 802.11 – Espectro Esparcido por Salto en Frecuencia

La técnica de Espectro Esparcido por Salto en Frecuencia fue el primer paso en la evolución de este tipo de tecnología. La idea detrás de esparcir el

espectro es evitar la interferencia de banda angosta, así como la interferencia mutua entre transmisores cercanos, para alcanzar velocidades de hasta 2 $Mbps$.

En el caso de IEEE 802.11, la técnica FHSS interpreta la banda de ISM de 2.4 Ghz como una serie de canales separados cada 1 MHz. El canal 1 tiene su frecuencia central en 2.401 GHz, el canal 2 en 2.402 GHz, y así hasta el canal 95 en 2.495 GHz.

El transmisor debe cambiar de canal al menos 2.5 veces por segundo, es decir cada 400 $mseg$ o menos. Los patrones de salto de frecuencia se definen como 3 conjuntos de 26 secuencias cada uno. Se trata de secuencias reguladas por una ley pseudo-aleatoria, cuya técnica de salto se apoya en la existencia de sintetizadores de frecuencia controlables digitalmente, conocidos como Osciladores Con control Numérico (NCO, Numerical Controlled Oscillators) o sintetizadores por Lazo de Enganche de Fase (PLL, Phase Locked Loop). En caso de redes desplegadas entre varios AP, los conjuntos de frecuencias de salto se eligen de tal modo que las secuencias de cada uno signifiquen un mínimo de interferencia mutua. Cuando las secuencias no tienen ninguna frecuencia que se solape, se las considera ortogonales.

En general, FHSS es una tecnología simple y de bajo costo que no posee grandes exigencias en el consumo de potencia, aunque su funcionamiento tiene requisitos de sincronización importantes. El estándar especifica que, en cada salto de portadora, la señal se module en GFSK (*Gaussian Frequency Shift Keying*, Modulación en Frecuencia con formato Gaussiano) binario para 1 $Mbps$ y GFSK cuaternario para lograr alcanzar 2 $Mbps$ de velocidad de transmisión de datos. GFSK es modulación en frecuencia que utiliza un filtro Gaussiano en banda base, para que los saltos de frecuencia se presenten de manera más suavizada, presentando un espectro más compacto.

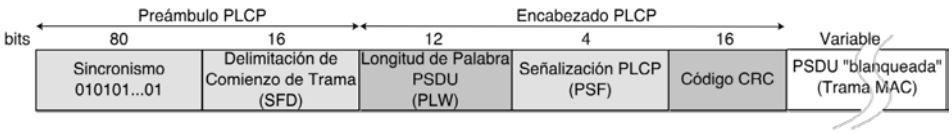


Figura 8.4 - Capa Física IEEE 802.11 FHSS.

Las tramas MAC llevan un preámbulo PLCP para sincronismo, igual que en *Ethernet*. En el caso 802.11 con capa Física FHSS, el preámbulo se compone de un campo de Sincronismo y uno de Delimitación de Comienzo de Trama (SFD), tal como se observa en la Fig. 8.4. La presencia de la señal de sincronismo indica que es inminente la llegada de una trama. También puede permitir, a aquellas estaciones con múltiples antenas, seleccionar la antena con la señal más fuerte para poder combatir el *fading multipath* con mayor eficiencia. El campo de Sincronismo consta de 80 $bits$ de una secuencia alternante que es utilizada por los receptores para prepararse a recibir los datos. El Delimitador de Comienzo de Trama SFD es un campo fijo de 16 $bits$, establecido por el estándar en el valor "0000 1100 1011 1101".

A continuación, el encabezado PLCP transporta parámetros específicos de la capa física. Un campo de Longitud de la Palabra PSDU (PLW, PSDU Length Word) de 12 $bits$ señala la longitud de de la trama MAC. Según este número, la trama máxima podría llegar hasta 4.095 $bytes$. Luego se presenta el campo de Señalización (PSF, PLCP Signaling Field) de 4 $bits$, que lleva el primer bit fijo en "0" y codifica en los tres restantes la velocidad a la que se transmite la trama MAC. Este anuncio sirve para que el receptor prepare el demodulador adecuado a la recepción de datos a 1 $Mbps$ o 2 $Mbps$.

A su vez, el encabezado PLCP se protege con un código CRC de 16 $bits$.

Cabe aclarar que los campos PLCP se transmiten a siempre a 1 $Mbps$. Por su parte, los datos de la PLCP, o sea la propia trama MAC, pasan a través de un mezclador antes de su envío, para que su espectro se parezca a AWGN. Luego, en recepción, se debe revertir el proceso.

Algunos parámetros característicos de la Capa Física FHSS son:

Tabla 8.2 – Parámetros IEEE 802.11 FHSS

Ranura	**SIFS**	**CW**	**Preámbulo**	**Encabezado PLCP**	**Trama MAC Máxima**
50 µseg	28 µseg	15 – 1023 ranuras	96 µseg a 1 $Mbps$	32 µseg a 1 $Mbps$	4095 $bytes$

8.3.2 IEEE 802.11 – Espectro Esparcido por Secuencia Directa

En IEEE 802.11, DSSS también operaba originalmente a 1 $Mbps$ y 2 $Mbps$, aunque se trata de una técnica potencialmente mucho mejor que FHSS para alcanzar mayores velocidades. Por este motivo se impuso como tecnología a partir de la aparición de IEEE 802.11b en 1999, permitiendo alcanzar velocidades de 5.5 $Mbps$ y 11 $Mbps$ con ciertas modificaciones.

La técnica DSSS combina la secuencia de datos de baja velocidad con otra secuencia de mayor velocidad, conocida como secuencia de *chips*. La característica de estos *chips* es que son bits de duración mucho menor que los bits de datos. En general, las secuencias de *chips* provienen de códigos pseudo-aleatorios, también conocidos como códigos PN, combinándose con los datos como se muestra en la Fig. 8.5. La combinación se puede interpretar como la suma EXOR de cada bit de datos con una secuencia de *chips*. Si en el receptor se realiza la suma EXOR con la misma secuencia *PN* sincronizada, se pueden recuperar los bits de datos transmitidos.

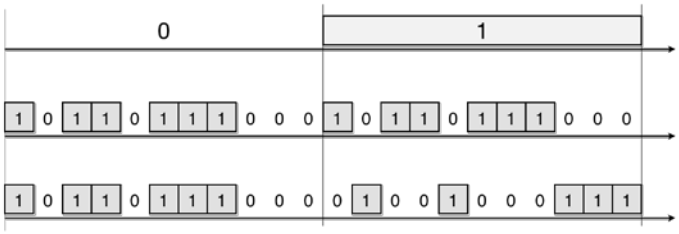


Figura 8.5 - Secuencia PN en IEEE 802.11 DSSS.

El efecto espectral de esta operación es un desparramo de la potencia de la señal sobre un ancho de banda mucho mayor. La relación de desparramo es el número de *chips* usados para cada bit de datos. A mayor cantidad de *chips*, más aumenta el ancho de banda, pero también el costo del equipamiento RF necesario.

IEEE 802.11 adoptó una secuencia de $11\ chips$, conocida como secuencia de Barker, por sus propiedades de auto-correlación, que le otorgan mejor respuesta en recepción frente al problema de la variación del retardo por multi-camino. La secuencia de Barker es fija: $\{+1, -1, +1, +1, -1, +1, +1, +1, -1, -1, -1\}$, también expresada en hexadecimal como $0x5BB$. De esta manera, una serie de datos a $1\ Mbps$ que se modula con el código Barker, se transforma en una secuencia a $11\ Mbps$, mientras que una serie de datos a $2\ Mbps$ genera una secuencia a $22\ Mbps$.

El estándar 802.11b, establece una modulación BPSK para el flujo de datos de $1\ Mbps$, luego de la combinación con el código de Barker. De esta manera, el ancho de banda final que ocupa la señal transmitida en el espectro es de $22\ MHz$. En el caso de $2\ Mbps$, se establece una modulación DQPSK, de manera que transmisión final también ocupa un ancho de banda de $22\ MHz$.

Cuando la red se presenta en cercanía de otros AP, la asignación de canales de trabajo ha de ser tal que no se interfieran entre sí. Una asignación muy común es la de los canales 1, 6 y 11 que comienzan en 2.412, 2.437 y 2.462 respectivamente. Estos canales se encuentran separados por bandas de $25\ MHz$, espacio suficiente para acomodar el espectro de $22\ MHz$ de la WLAN.

Como en FHSS, en el caso DSSS también existe un mezclador que sirve para remover secuencias largas de "$1´s$" o "$0´s$" de la cadena de datos transmitidos. De este modo se minimizan problemas de sincronismo, aplicándose la operación a la trama completa, incluyendo el encabezado de capa física.

Para este tipo de capa física, la PLCP agrega un encabezado de 6 campos a la trama MAC, tal como se muestra en la Fig. 8.6.

A diferencia de FHSS, en DSSS el Preámbulo se transmite modulado en DSSS. El campo de Sincronismo es de $128\ bits$, en este caso todos "1" y el campo SFD lleva una palabra diferente de la de FHSS: "0000 0101 1100 1111". A continuación, el campo de Señalización se usa para interpretar la velocidad de la trama encapsulada, codificando con "$0x0A$" el caso de $1\ Mbps$ y "$0x14$" para $2\ Mbps$. El campo Servicio es reservado y se ajusta a todos "0". El campo Longitud es el número de μseg requeridos para transmitir la trama. Por último, un código CRC de 16 $bits$ protege el encabezado PLCP.

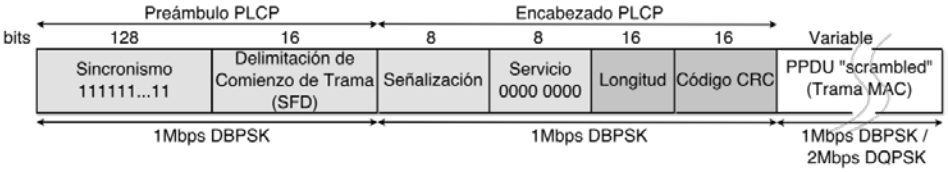


Figura 8.6 - Capa Física IEEE 802.11 DSSS.

Para la transmisión a 1 $Mbps$, se agrega el encabezado PLCP a la trama MAC y todo el conjunto se pasa por el *mezclador*, modulándose la secuencia resultante en *DBPSK*. Para transmitir a 2 $Mbps$, el preámbulo y encabezado PLCP se modulan en DBPSK a 1 $Mbps$, luego la capa física usa modulación DQPSK para proveer servicio a 2 $Mbps$.

Se presentan a continuación los principales parámetros para la capa física DSSS:

Tabla 8.3 – Parámetros IEEE 802.11 DSSS

Ranura	**SIFS**	**CW**	**Preámbulo**	**Encabezado PLCP**	**Trama MAC Máxima**
20 µseg	10 µseg	31 – 1023 ranuras	144 µseg a 1 $Mbps$	48 µseg a 1 $Mbps$	8191 $bytes$.

8.4 Capa Física IEEE 802.11b – Secuencia Directa de Alta Velocidad

La versión original IEEE 802.11 sólo permitía alcanzar una velocidad como máximo de 2 $Mbps$. Por este motivo, en el año 1999 se publicó la extensión IEEE 802.11b, que permitía la transmisión hasta 11 $Mbps$. Este aumento de la velocidad tuvo mucha repercusión en la instalación comercial de las redes WLAN. Para poder lograrlo, el nuevo estándar mantuvo la estructura de canales de DSSS, aunque cambió el esquema de modulación, que se conoce como HR/DSSS.

Se descartaron los esquemas de modulación de más de cuatro fases dado que es difícil discriminar corrimientos de fase más finos en presencia de interferencia *multipath*. En su lugar se propuso la incorporación de una codificación denominada CCK.

Un Código Binario Complementario es un subconjunto de una clase de códigos conocidos como Códigos Polifase. CCK es un Código Polifase Complementario. Las secuencias binarias complementarias se componen de

secuencias de igual longitud, finita, que tienen la propiedad de que el número de pares de elementos similares con cierta separación en una serie es igual al número de pares de elementos no similares con la misma separación en la otra. Un ejemplo lo demuestra muy fácilmente. Si la secuencia 1 es $\{-1,-1,-1,1,1,1,-1,1\}$ y la secuencia 2 $\{-1,-1,-1,1,-1,-1,1,-1\}$, recorridas por pares se transforman en $\{=,=,\neq,=,=,\neq,\neq\}$ y $\{=,=,\neq,\neq,=,\neq,\neq\}$. La secuencia 1 tiene 4 pares de elementos similares con una separación de 1 y 3 pares de elementos diferentes con una separación de 1. La secuencia 2 tiene 3 pares de elementos similares y 4 pares de elementos diferentes con una separación de1. Si se considera una separación de 2 elementos se obtienen los mismos resultados. Para una separación de 3 elementos los valores siguen siendo complementarios, 1 y 5 para la secuencia 1 y 5 y 1 para la secuencia2.

En términos matemáticos, si se consideran dos secuencias de n elementos a_i y b_i, con $1 \leq i \leq n$, se dice que son complementarias cuando, al calcular sus respectivas series de autocorrelación, $c_j = \sum_{i=1}^{n-j} a_i a_{i+j}$ y $d_j = \sum_{i=1}^{n-j} b_i b_{i+j}$, se verifica $c_j + d_j = 0$ para $j \neq 0$ y $c_0 + d_0 = 2n$. Esta propiedad resulta muy útil a la hora de transmitir información de manera digital.

Un Código Polifase Complementario también tiene propiedades complementarias, pero sus elementos poseen parámetros de fase. Por ejemplo, el código definido para 802.11b de alta velocidad tiene elementos en el conjunto $\{1,-1,j,-j\}$ y se ha demostrado que se trata de un código con buenas propiedades de distancia, que favorecen la mejora de la P_{be} en entornos inalámbricos.

Para que se pudiera mantener el mismo ancho de banda y distribución de canales que en IEEE 802.11, se definió una longitud de código de 8 $chips$ a una velocidad de 11 $Mchip/seg$. Dicho de otro modo, al establecer la velocidad por símbolo en $\frac{11\frac{Mchip}{seg}}{8\frac{chip}{símb}} = 1.375\ símb/seg$, se mantiene el ancho de banda ocupado por cada canal.

Las palabras código que conforman un símbolo de 8 $chips$ $c = (c_0\ c_1\ c_2\ c_3\ c_4 c_5\ c_6\ c_7\ )$, se definen a partir de la siguiente expresión:

$$c = \begin{Bmatrix} e^{j(\varphi_1+\varphi_2+\varphi_3+\varphi_4)}, e^{j(\varphi_1+\varphi_3+\varphi_4)}, e^{j(\varphi_1+\varphi_2+\varphi_4)}, -e^{j(\varphi_1+\varphi_4)}, \\ e^{j(\varphi_1+\varphi_2+\varphi_3)}, e^{j(\varphi_1+\varphi_3)}, -e^{j(\varphi_1+\varphi_2)}, e^{j(\varphi_1)} \end{Bmatrix} \tag{8.1}$$

La misma expresión se usa en el caso de 5.5 $Mbps$ o de 11 $Mbps$, la única diferencia es que las palabras utilizadas en 5.5 $Mbps$ son un subconjunto de las de 11 $Mbps$. Las fases $\varphi_1\ a\ \varphi_4$ se determinan a partir de los datos. En el caso de mayor velocidad, cada símbolo representa 8 $bits$ de información, pero en 5.5 $Mbps$ se trata sólo 4 $bits$. La principal diferencia con el caso DSSS es que el código de Barker es fijo, en tanto que CCK usa una palabra código derivada de la propia secuencia de datos para transportar información y esparcir el espectro.

Tabla 8.4 – Modulación CCK para 11 Mbps

Dibit	Parámetro de Fase	Dibit(d_{i+1},d_i)	Fase DQPSK
(d_1,d_0)	ϕ_1	00	0
(d_3,d_2)	ϕ_2	01	π
(d_5,d_4)	ϕ_3	10	$\pi/2$
(d_7,d_6)	ϕ_4	11	$-\pi/2$

A modo de ejemplo, se presenta la modulación de una palabra para $11Mbps$. Se usan los bits de los datos para definir los parámetros de fase $\phi 1$ a $\phi 4$, de acuerdo a la Tabla 8.4. También la tabla presenta la codificación DQPSK, según el par de valores presente. Así, si se presenta una palabra $(d_7, d_6, d_5, d_4, d_3, d_2, d_1, d_0) = (1\ 0\ 1\ 1\ 0\ 1\ 0\ 1)$, la misma define, de acuerdo a la Tabla, el conjunto ordenado de parámetros de fase $(\varphi 4, \varphi 3, \varphi 2, \varphi 1) = (\frac{\pi}{2}, -\frac{\pi}{2}, \pi, \pi)$. A partir de esta asignación, la palabra código resulta:

$$c = \begin{Bmatrix} e^{j(\pi+\pi-\pi/2+\pi/2)}, e^{j(\pi-\pi/2+\pi/2)}, e^{j(\pi+\pi+\pi/2)}, -e^{j(\pi+\pi/2)}, \\ e^{j(\pi+\pi-\pi/2)}, e^{j(\pi-\pi/2)}, -e^{j(\pi+\pi)}, e^{j(\pi)} \end{Bmatrix} \tag{8.2}$$

$$c = \begin{Bmatrix} e^{j(2\pi)}, e^{j(\pi)}, e^{j(5\pi/2)}, -e^{j(3\pi/2)}, \\ e^{j(3\pi/2)}, e^{j(\pi/2)}, -e^{j(2\pi)}, e^{j(\pi)} \end{Bmatrix} \tag{8.3}$$

$$c = \{1, -1, j, j, -j, j, -1, -1\} \tag{8.4}$$

Se observa que el parámetro de fase $\phi 1$ se encuentra presente en los 8 $chips$ de la palabra código, de tal manera que, esencialmente, rota el vector completo. Esta y otras características son importantes a la hora de la implementación ya que permiten distinguir, del lado receptor, entre diferentes codificaciones de manera más sencilla, aún en presencia de interferencia y *fading*.

Transmisión en 5.5 Mbps con CCK: Se basa en la modulación DQPSK, que transmite 2 $bits$ por período de símbolo, codificado como cuatro posibles corrimientos de fase. Usando CCK, las palabras símbolos transportan en sí mismas información adicional. En 5.5 $Mbps$ se codifican 4 $bits$ de datos en un símbolo: 2 $bits$ se transportan por medio de DQPSK convencional y los otros 2 $bits$ se transportan a través del contenido de las palabras codificadas, como se ilustra en la Fig. 8.7.

Como se observa en la figura, la trama MAC encapsulada por la trama PCLP, es dividida en una secuencia de bloques de 4 $bits$. Cada uno de estos bloques, a su vez, se divide en dos sub-bloques de 2 $bits$. El primer sub-bloque

de 2 $bits$ se codifica en DQPSK con la particularidad de que símbolos pares o impares usan diferente corrimiento de fase, según se indica en la Tabla 8.5.

Con el segundo sub-bloque de 2 bits, se selecciona una de entre cuatro palabras código según se presenta en la Tabla 8.6. Estas palabras son las que surgen de la codificación CCK mencionada.

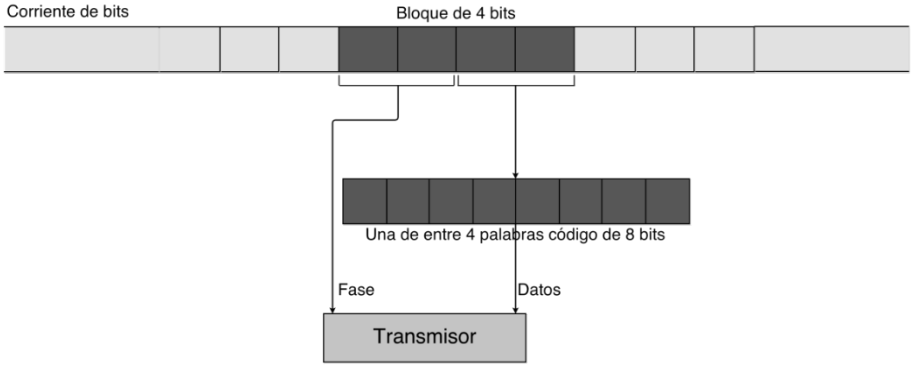


Figura 8.7 – IEEE 802.11b CCK a 5.5 Mbps

Tabla 8.5 – Símbolo DQPSK del sub-bloque de 2 bits de mayor orden

Dibit	Fase símbolo par	Fase símbolo impar
00	0	π
01	π/2	-π/2
11	π	0
10	-π/2	π/2

Tabla 8.6 – Palabras Código para 5 Mbps.

Dibit	Palabra Código
00	$i, 1, i, -1, i, 1, -i, 1$
01	$-i, -1, -i, 1, 1, 1, -i, 1$
11	$i, -1, i, 1, -i, 1, i, 1$
10	$-i, 1, -i, -1, -i, 1, i, 1$

Como sucede en 802.11, la velocidad de los chips es de 11 $Mchips/seg$. De esta manera, la duración de un chip es de $\frac{1}{11}$ μseg y la duración de una palabra de 8 chips $\frac{8}{11}$ μseg. De estas relaciones surge la velocidad de los símbolos : $\frac{11}{8} =$ 1.375 $Msímbolos/seg$. Como cada símbolo codifica la información proveniente de 4 $bits$, la velocidad alcanzada es de $4\frac{bits}{símbolo}1.375\frac{Msímbolos}{seg} = 5.5Mbps$.

Transmisión en 11 Mbps con CCK: En 11 $Mbps$ se toma la trama MAC y se divide la cadena de bits en bloques de 8 $bits$. El modulador usa 6 $bits$ de cada byte para elegir 1 entre 64 Códigos Complementarios Polifase de 8 $chips$ ortogonales. Los otros dos bits se usan para rotar la palabra código completa (0º, 90 º, 180 º o 270 º), con modulación DQPSK, tal como se muestra en la Fig. 8.8.

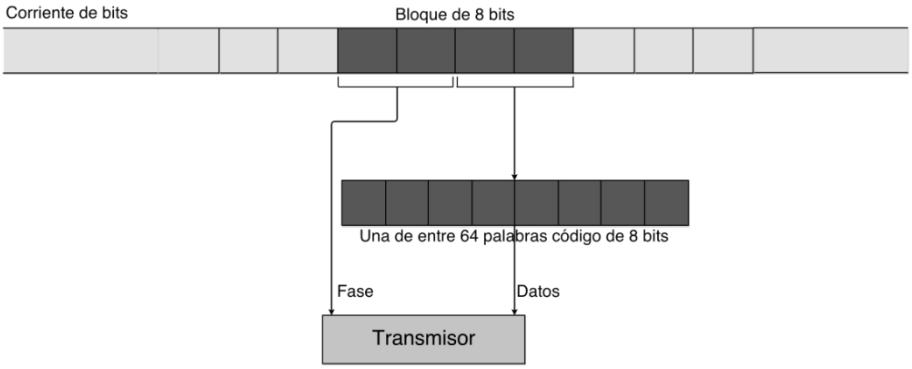


Figura 8.8 - IEEE 802.11b CCK a 11Mbps

Como en el caso anterior, la velocidad de los chips es de 11 $Mchips/seg$, aunque cada símbolo codifica la información proveniente de 8 bits, por lo que la velocidad alcanzada es de $8\frac{bits}{símbolo}1.375\frac{Msímbolos}{seg} = 11Mbps$.

Un esquema sencillo del transmisor y receptor de HR/DSSS se presenta en la Fig. 8.9. De acuerdo a la velocidad de transmisión seleccionada, la salida de un *scrambler* se divide en grupos de 4 u 8 $bits$, siendo un multiplexor de entrada serie/salida paralelo, el encargado de presentar estos grupos a la velocidad de 1.375 $Msímb/seg$. Como se ha explicado, en el caso de 11 $Mbps$, 6 $bits$ de salida del multiplexor se usan para seleccionar uno de 64 códigos complejos que se ingresan a un modulador diferencial. En el caso de 5.5 $Mbps$, la selección se realiza con 2 $bits$, para elegir una entre 4 palabras de la Tabla 8.6. En ambos casos, los 2 $bits$ restantes se usan para modular en DQPSK, siendo su efecto el de rotar la palabra código compleja de 8 $chips$. Las salidas del modulador diferencial son las salidas fase y cuadratura de acuerdo a la ecuación de generación de códigos complejos c , mencionada previamente.

En el receptor, la señal modulada en CCK se convierte a formato digital por demodulación coherente. En el caso de mayor velocidad, la determinación del código transmitido se realiza mediante un banco de 64 correladores, una vez obtenidos los 6 $bits$ de datos. En el caso de 5.5 $Mbps$ se utiliza un banco de 4 correladores. Es aquí donde se ponen en juego las propiedades de autocorrelación de los códigos complementarios. Los restantes 2 $bits$ en ambos casos, se determinan a partir de la fase DQPSK del símbolo.

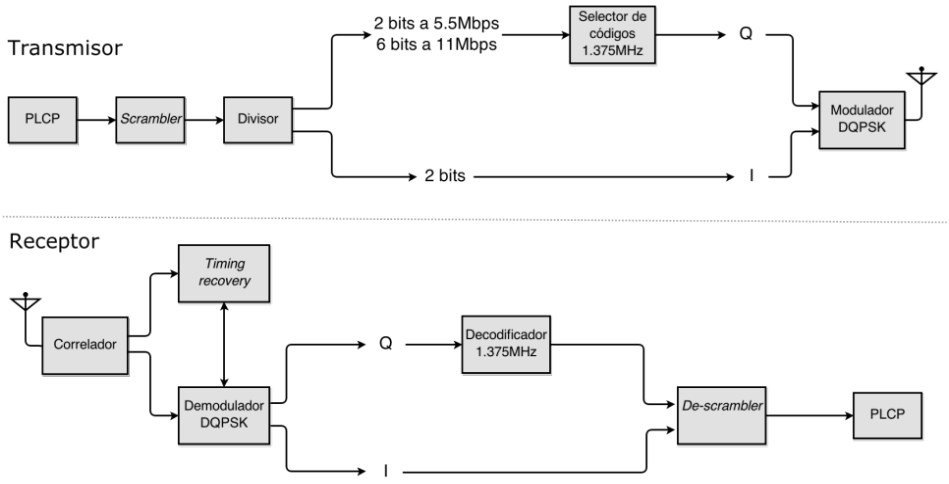


Figura 8.9 - HR/DSSS

En cuanto a la trama de capa física, el aumento de velocidad puso en discusión la duración del preámbulo, debido al *overhead* que el mismo representa.

Por ejemplo, cada trama MAC de datos exige en respuesta una trama ACK cuya duración es $14\,bytes\,x\,8\frac{bits}{byte}x\,\left(\frac{1}{11Mbps}\right) = 10.18\mu seg,$ mucho menor que la duración de preámbulo y encabezado PLCP a $1\,Mbps$, que es de $192\,\mu seg$. Por otro lado, a $11\,Mbps$, la transmisión de una trama de $1500\,bytes$ consumiría $1500\,bytes\,x\,8\frac{bits}{bytes}\,x\left(\frac{1}{11Mbps}\right) = 1090.90\mu seg$. Si consideráramos sumar al tiempo de transmisión propiamente dicho, el correspondiente a la trama de ACK y el del SIFS intermedio ($10\mu seg$), estaríamos en el orden de los $1111\mu seg$. Si cada una de las tramas mencionadas se transporta en capa física con un encabezado a nivel físico de $(144bits + 48bits)x1\,Mbps = 192\,\mu seg$, resulta que un porcentaje importante del tiempo de transmisión, a la velocidad máxima de $11\,Mbps$, se consume en el encabezado de capa física.

Por este motivo se intentó acortar el formato de la trama a nivel físico, ofreciendo la posibilidad de usar un preámbulo corto de $72\,bits$ para reducir el *overhead* de capa física ya que, de este modo, se reduce bastante el tiempo consumido en la transmisión del encabezado físico. La Fig. 8.10 muestra el preámbulo original largo y la posibilidad de preámbulo corto para IEEE 802.11b. Observar en la propia figura, los esquemas de modulación y velocidades referenciados.

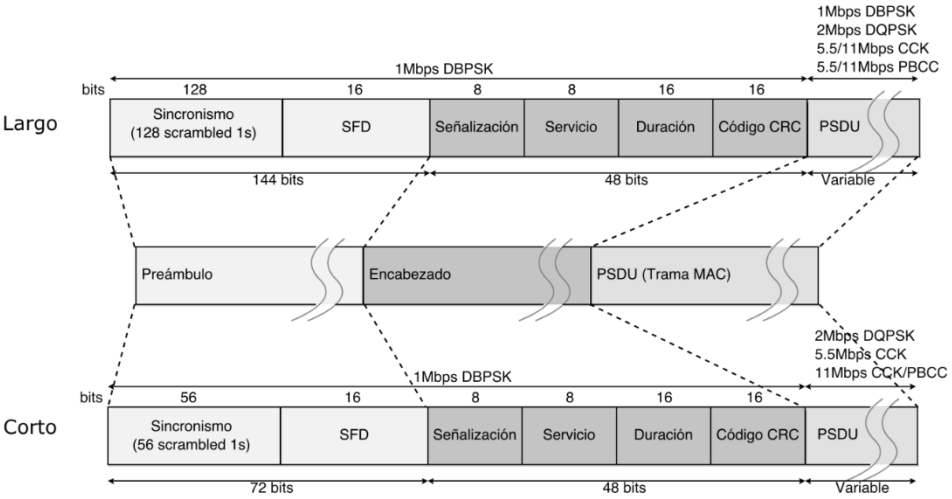


Figura 8.10 - Trama PLCP HR/DSSS.

En el caso de Preámbulo Largo, la cabecera y el preámbulo se transmiten a 1 $Mbps$ con modulación DBPSK. Con el Preámbulo Corto, el campo para sincronismo es una serie de 56 $bits$ en ¨0¨ procesados por un *scrambler*. En el preámbulo largo, SFD es "1111 0011 1010 0000" y en el corto, su valor complementario, "0000 1100 0101 1111". En el caso de preámbulo corto, el resto de la cabecera PLCP se transmite a 2 $Mbps$ con modulación DQPSK y transporta los mismos campos que en el caso de preámbulo largo.

Para finalizar, se ofrece un resumen de los parámetros que caracterizan HR/DSSS son:

Tabla 8.7 – Parámetros IEEE 802.11 HR/DSSS

Ranura	**SIFS**	**CW**	**Preámbulo**	**Encabezado PLCP**	**Trama MAC Máxima**
20 μseg	10 μseg	31 – 1023 ranuras	144 μseg largo 72 μseg corto Ambos a 1 $Mbps$	24 μsega 2 $Mbps$	4095 $bytes$.

8.5 Capa Física IEEE 802.11a – OFDM 5 GHz

Estandarizado en 1999, el protocolo IEEE 802.11a fue diseñado para la banda de Infraestructura de Información Nacional No Licenciada (U-NII, Unlicensed National Information Infrastructure) asignada en 5 GHz en USA. La

banda ofrece mayor espectro que la de 2.4 GHz y menos densidad de dispositivos interferentes. A pesar de ello, debido a que es utilizada principalmente por sistemas de radar militares, derivó en la aparición del estándar IEEE 802.11h para Europa en el año 2003 y su equivalente IEEE 802.11j para Japón. IEEE 802.11h ofrece la capacidad de gestionar dinámicamente tanto la frecuencia como la potencia de transmisión, pudiéndose adaptar a situaciones particulares de interferencia.

De todas maneras, comparado con los sistemas que trabajan en la banda de 2.4 GHz, el estándar IEEE 802.11a ofrece la posibilidad de desplegar más de tres canales y alcanzar velocidades más altas, del orden de $54\ Mbps$. Una desventaja importante es que no es compatible con IEEE 802.11b, justamente por operar en otra banda del espectro.

Como ya se ha visto, un problema importante de los canales inalámbricos usados en transmisiones de alta velocidad es que presentan efectos severos de *fading* selectivo en frecuencia, resultando difícil la aplicación de técnicas de ecualización, en términos de performance y complejidad, sobre una única señal portadora de banda ancha. Para mitigar este problema, IEEE 802.11a se aparta de las técnicas tradicionales de modulación con una única portadora, incorporando una nueva técnica conocida como OFDM.

Si se divide un flujo de bits de alta velocidad r_b en un gran número N_s de flujos paralelos de baja velocidad $\frac{r_b}{N_s}$, y luego se realiza la transmisión simultánea de estos N_s flujos sobre los correspondientes N_s sub-canales, con ancho de banda menor que el ancho de banda de coherencia del canal, se logra mitigar el efecto de *fading*. De este modo, un símbolo se convierte en un conjunto de bits que se utiliza para modular en simultáneo varias sub-portadoras. Si la velocidad de los símbolos y el espaciamiento entre sub-portadoras se eligen de tal manera que estas últimas sean ortogonales entre sí sobre el tiempo de duración de un símbolo, el esquema se conoce como OFDM. Con estas premisas, las sub-portadoras individuales, de menor ancho de banda, experimentarán *flat fading*, fácilmente compensable. Si a esta mejora se agrega la propiedad de relaciones ortogonales entre frecuencias sub-portadoras, se asegura una separación precisa en la recepción.

Esta nueva técnica salva uno de los problemas de FDM tradicional: el desperdicio del ancho de banda de transmisión en bandas de guarda entre canales para evitar la interferencia. Por este motivo, en OFDM se seleccionan canales que se solapan en frecuencia pero no interfieren entre sí debido a las relaciones ortogonales, lográndose una mayor eficiencia espectral o espectro más compacto. En este sentido, cada sub-portadora presentará un pico de amplitud en la frecuencia central, y nulos en los picos de amplitud de las demás. También mejora la performance con respecto a la interferencia, disminuyendo la complejidad de implementación respecto del caso de una única portadora con ecualización, y presentando excelente rechazo a la interferencia de banda angosta ya que, de estar presente, la misma sólo afectaría un pequeño porcentaje de canales.

La relación ortogonal entre frecuencias sub-portadoras se traduce en relaciones matemática precisas. Así, si un símbolo OFDM dura $T\ seg$, las primeras sub-portadoras serán de frecuencia $f_0 = 1/T$, $f_1 = 2/T$, $f_2 = 3/T$ y así

sucesivamente. Es interesante destacar que se podrían utilizar otras combinaciones de frecuencias ortogonales, tales como $f_0 = 1/T$, $f_1 = 2/T$, $f_2 = 4/T$ o $f_0 = 1/T$, $f_1 = 3/T$, $f_2 = 5/T$, pero estas no serían eficientes desde el punto de vista del ancho de banda, ya que la señal a la salida OFDM resulta ser la suma algebraica de las señales provenientes de todas las sub-portadoras, siendo las técnicas de modulación más utilizadas para la transmisión BPSK, QPSK y QAM. Una consecuencia de este esquema de transmisión es que el receptor puede ser construido como un banco de demoduladores que traslade cada sub-portadora a banda base y luego integre sobre el período correspondiente para recuperar el dato transmitido. Esta técnica de recepción asegura interferencia nula entre portadoras, debido al número entero de ciclos por período, producto de la relación ortogonal entre las frecuencias de las mismas.

Obviamente, cuanto mayor es la cantidad de sub-portadoras transmitidas en paralelo, mejor funciona OFDM, pero la implementación original hasta aquí comentada se vuelve bastante más compleja. Este detalle significó una postergación de las implementaciones prácticas de OFDM hasta la aparición de nuevas técnicas de procesamiento digital en *hardware* que permitieron generar la misma señal y recuperarla de manera más sencilla. Una de las técnicas más utilizadas es la Transformada Rápida de Fourier (FFT, Fast Fourier Transform). La FFT es una variación de la Transformada Discreta de Fourier (DFT, Discrete Fourier Transform), descubierta en los años 60 como una manera de aumentar considerablemente la velocidad de cálculo de la transformada, permitiendo un análisis mucho más práctico. La señal digitalizada se trabaja como un conjunto de muestras para el cálculo de la FFT. De este modo se obtiene una versión digital del espectro de la misma. Esta característica convierte a la FFT en una herramienta importante para separar las sub-portadoras de una señal OFDM. Por su parte, la Transformada Rápida Inversa de Fourier IFFT, realiza el cálculo inverso al de la transformada FFT. Tomando las portadoras individuales con modulación digital y aplicándoles el procesamiento IFFT, se puede obtener una única señal compuesta para su transmisión, equivalente a la señal OFDM. Del lado receptor, aplicando la FFT se logra separar todas las señales para volver a obtener la secuencia de datos original.

La equivalencia entre ambas operaciones se puede ver si se desarrolla matemáticamente una señal OFDM correspondiente al esquema original de generación. Dicha señal se puede expresar como la suma de sub-portadoras moduladas en PSK o QAM. Si se designa con d_i al símbolo QAM, N_s al número de subportadoras, T a la duración de un símbolo y f_c a la frecuencia de la portadora, se puede describir un símbolo OFDM que empiece en $t = t_s$ como:

$$s(t) = \left\{ \sum_{i=0}^{i=N_s} d_i \ \exp j2\pi(f_c - \frac{i}{T})(t - t_s) \right\}, t_s \leq t \leq t_s + T$$
$$s(t) = 0, t_s > t \ y \ t > t_s + T \tag{8.5}$$

La relación ortogonal entre frecuencias sub-portadoras también se puede ver desde otro punto de vista. De acuerdo a la ecuación (8.5), cada símbolo contiene sub-portadoras que son distintas de cero sobre el intervalo de tiempo T.

O sea que el espectro de un único símbolo es la convolución de un grupo de deltas de Dirac localizadas en la frecuencia de la sub-portadora con el espectro de un pulso cuadrado de duración T. El espectro del pulso tiene la forma de una función $sinc(\pi fT)$, con nulos en todas las frecuencias múltiplos de $1/T$. La relación ortogonal permite que en los máximos de cada sub-espectro, el resto de los sub-espectros presenten nulos. Si el receptor OFDM calculara los valores espectrales en dichos puntos, podría de-modular cada sub-portadora sin interferencia de las demás. Se trata del criterio de Nyquist visto desde el punto del dominio de la frecuencia, pensando que se trata de evitar la interferencia entre portadoras (ICI) en lugar de la interferencia entre símbolos (ISI).

Muchas veces, por comodidad, en lugar de la Ec. (8.5) se utiliza la expresión en banda base equivalente:

$$s(t) = \left\{ \sum_{i=0}^{i=N_s} d_i \ \exp j2\pi(\frac{i}{T})(t - t_s) \right\}, t_s \leq t \leq t_s + T$$
$$s(t) = 0, t_s > t \ y \ t > t_s + T \tag{8.6}$$

En esta última representación, las componentes real e imaginaria se corresponden con las componentes fase y cuadratura de la modulación, que luego se multiplicarán por el $\cos(2\pi f_c t)$ y $sen(2\pi f_c t)$ a la frecuencia que corresponda, para generar, mediante la suma, la señal OFDM. En la realidad, cada portadora podrían modularse de manera diferente, presentando diferencias de amplitud y fase entre ellas, pero todas presentarán un número entero de ciclos en el intervalo T y el número de ciclos entre portadoras adyacentes diferirá en 1 exactamente, para que sean ortogonales.

Por otra parte, la expresión de la señal compleja OFDM en banda base es semejante a la Transformada Inversa de Fourier de N_s símbolos de entrada. El equivalente discreto es la IDFT, que resulta de reemplazar t en la expresión anterior por el número de muestra n:

$$s(n) = \left\{ \sum_{i=0}^{N_s - 1} d_i \ \exp(\frac{j2\pi in}{N}) \right\} \tag{8.7}$$

Una de las mejores contribuciones para el desarrollo de OFDM es, justamente, la posibilidad de usar la Transformada Discreta de Fourier para realizar la modulación y demodulación de la señal en banda base. Esta novedad en la implementación original de multi-portadoras analógicas permitió utilizar un esquema digital, eliminando bancos de osciladores y demoduladores coherentes, reduciendo la complejidad y haciendo posible sistemas modernos de OFDM de bajo costo.

A pesar de esta importante innovación, un esquema de esta naturaleza no podía mantener relaciones ortogonales perfectas entre frecuencias de sub-portadoras cuando el canal es del tipo inalámbrico. Por este motivo, al principio, los problemas de ISI e ICI se mitigaban por medio del uso de tiempos de guarda entre símbolos y filtrado coseno levantado.

Un esquema simplificado de transmisión y recepción se presenta en la Fig. 8.11. La serie de datos entrantes se agrupa en conjuntos de cierta cantidad de bits, según la constelación correspondiente a la modulación en cuestión, por ejemplo 4 bits en el caso de 16 QAM o 5 bits para 32 QAM. Previo a este paso puede aplicarse técnicas de corrección de errores y entrelazado para obtener robustez en el caso de errores por ráfagas. A la salida del modulador, se trabaja con números complejos en formato serie, a los que se aplica una conversión a paralelo para poder aplicarles la operación de la transformada inversa de la FFT (IFFT). Los datos transformados se agrupan nuevamente, según la cantidad de sub-portadoras requeridas, para luego ser multiplexados a un formato serie. En este punto, los datos ya se encuentran modulados en OFDM y listos para ser transmitidos, pero se convierten a analógicos y modulan en RF a la frecuencia de transmisión. Previo a la modulación, a los fines de preservar las relaciones ortogonales, es posible agregar un tiempo de guarda a cada símbolo. Por supuesto, el receptor realiza la operación inversa, utilizándose luego de la FFT, antes de la conversión serie, una ecualización muy sencilla, de ganancia ajustable a la información obtenida del canal, para corregir la distorsión.

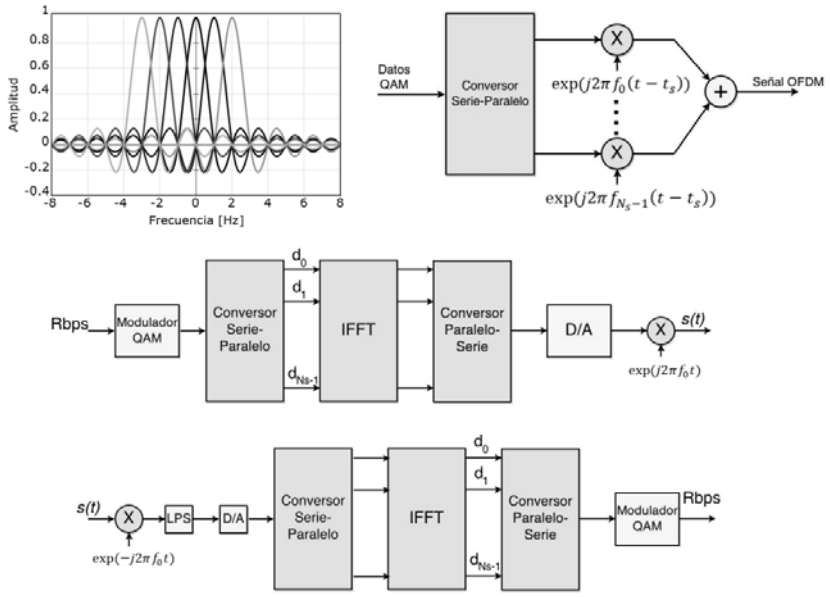


Figura 8.11 - Principio de OFDM. Generación y Recepción.

En principio, un tiempo de guarda es simplemente un tiempo de silencio entre dos símbolos consecutivos. Para reducir el ISI, se debe introducir un tiempo de guarda más largo que el desparramo de retardo esperado, para que los componentes *multipath* de un símbolo no interfieran con el siguiente. Dicho en otras palabras, el tiempo de guarda ofrece un tiempo suficiente para que las señales multi-camino provenientes del símbolo previo, hayan terminado antes que se reciba la información correspondiente al símbolo actual.

En la práctica, si el tiempo de guarda se generara como ausencia de señal, podrían presentarse problemas de ICI. Para eliminar este último efecto, el símbolo OFDM se extiende cíclicamente dentro del intervalo de guarda: una réplica de la última parte del símbolo OFDM se copia al frente del mismo. En estos casos se llama prefijo cíclico al tiempo de guarda.

La aparición de técnicas para el agregado del prefijo cíclico fue otro hito en la implementación de OFDM. Como se ha mencionado, el prefijo cíclico ocupa el mismo intervalo de tiempo que el período de guarda, pero asegura que las réplicas retrasadas de los símbolos OFDM siempre presenten un número entero de ciclos dentro del intervalo de la FFT, siempre que el retardo sea menor que el tiempo de guarda. Como resultado de esto, las señales *multipath* con retardos menores al tiempo de guarda no pueden causar ICI.

En la Fig. 8.12 se muestra el efecto producido por el prefijo cíclico. La selección del valor del tiempo de guarda es crítico para el correcto funcionamiento de OFDM. Se puede demostrar matemáticamente que el prefijo cíclico hace que el símbolo OFDM aparezca como periódico en el receptor, aún cuando pudiera existir un retardo propio del canal de *fading*. Esta propiedad convierte en periódica la operación de convolución entre la función transferencia del canal y la señal transmitida, eliminando de este modo la interferencia entre sub-portadoras y permitiendo mantener la relación ortogonal entre las mismas.

Obviamente, el tiempo de guarda reduce la velocidad efectiva de transmisión ya que alarga el tiempo de duración del símbolo aunque la posibilidad de lograr ICI nulo compensa la reducción .

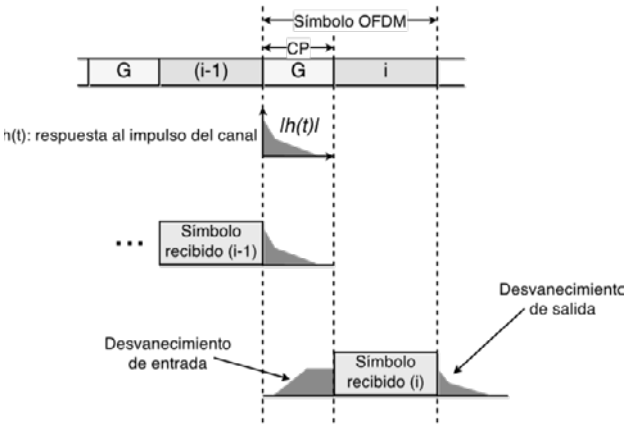


Figura 8.12 – Intervalo de Guarda.

Otra mejora agregada al sistema es la técnica de ventana (*windowing*), aplicada a símbolos OFDM individuales, ya que permite obtener un espectro más compacto. Una de las técnicas de ventana más usadas se basa en formatos tipo coseno levantado, que logra que la señal caiga a cero suavemente en los límites con otros símbolos. De este modo, se evitan las transiciones abruptas, logrando el efecto deseado sobre el espectro.

También es común el agregado bits de relleno (*padding*) al comienzo o al final de las transmisiones, sobre todo en el caso de corrección de errores tipo FEC convolucional, para permitir al codificador el regreso al estado inicial. De este modo, con los bits de relleno, se logra una transición suavizada de potencia desde el principio de la transmisión o hacia el final de la misma.

Una ventaja interesante de OFDM se presenta en términos de la ecualización, al comparar el comportamiento de esta técnica de transmisión con el de sistemas de portadora única, en canales selectivos en frecuencia. El procesamiento de la señal en el receptor es muy simple en el caso de OFDM, ya que la ecualización se reduce al procesamiento sobre una sub-portadora de menor ancho de banda.

En cuanto a los esquemas de estimación de la respuesta al impulso del canal, en OFDM se opta por insertar portadoras piloto dentro de la transmisión de un símbolo, en lugar de usar secuencias de entrenamiento entre símbolos consecutivos. Estas portadoras piloto también se usan para sincronización en tiempo y en frecuencia.

Una de las mayores desventajas de OFDM es debida a la propia generación de la señal, que presenta una alta relación de potencia pico a potencia promedio de la señal transmitida (PAPR, Peak-to-Average Power Ratio), comparada con el caso de una única portadora. En la práctica, este efecto se traduce en un empeoramiento de la relación de señal a ruido de cuantificación en los conversores analógico/digital y digital /analógico y en la pérdida de eficiencia del amplificador de potencia de transmisión. Problemas derivados de estos efectos empeoran notablemente la tasa de error, proponiéndose varias técnicas para mejorarlos.

Muchos son los parámetros a tener en cuenta al diseñar un esquema OFDM. Entre ellos, se puede mencionar el número de portadoras, el espaciamiento entre las mismas, el tipo de modulación, la duración de los símbolos, el tiempo de guarda entre ellos para evitar el ISI y la técnica de corrección de errores apropiada. La selección depende del ancho de banda disponible, la velocidad de transmisión que se desee alcanzar, el desparramo del retardo que resulte tolerable y las características del propio canal de transmisión. Generalmente, el ancho de banda es fijado por las autoridades de regulación y el retardo en exceso es fuertemente dependiente del entorno. En la mayoría de los edificios es de 40 a 70 $nseg$ aunque podría llegar a valores tan altos como 200 $nseg$. Una estimación práctica sería que el intervalo de guarda se eligiera entre 2 a 4 veces el retardo en exceso. Por ejemplo, en el caso de IEEE 802.11a se seleccionó un tiempo de guarda de 800 $nseg$.

A su vez, la duración de los símbolos debería ser grande, cuanto más grande, más sub-portadoras podrían entrar dentro de la duración del símbolo, pero esto aumenta la carga de procesamiento en ambos extremos. Un valor práctico podría ser al menos 5 veces el tiempo de guarda. En este sentido, IEEE 802.11a adopta el tiempo de símbolo de $4\ \mu seg$. Debido a que el espaciamiento entre sub-portadoras se encuentra inversamente relacionado con el tiempo de integración de la FFT, IEEE 802.11a tiene un tiempo de integración de $4\ \mu seg - 0.8\ \mu seg = 3.2\ \mu seg$. Su inversa establece el espaciamiento entre sub-portadoras $\frac{1}{3.2\mu seg} = 0.3125\ MHz$.

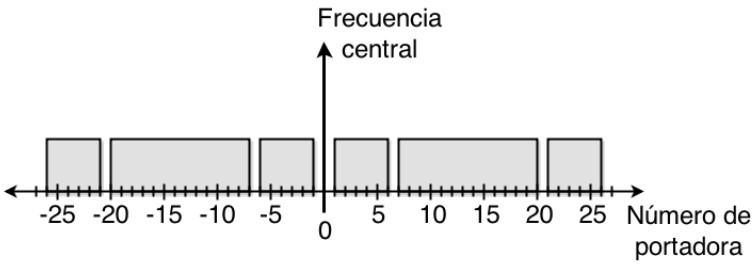


Figura 8.13 - Canal OFDM de IEEE 802.11a.

En la banda de $5\ GHz$ elegida para las redes IEEE 802.11a, cada canal tiene $5\ MHz$ de ancho de banda, pero sólo se pueden usar uno cada cuatro de estos canales. El motivo es que las redes IEEE 802.11a ocupan 20 MHz en el espectro. Por ejemplo, si se elige desplegar una red en el canal 40 de esta banda, cuya frecuencia central es $5.2\ GHz$, el ancho de banda ocupado se extenderá $10\ MHz$ para cada lado, entre $5190 MHz$ y 5210MHz, utilizando 2 canales en cada dirección: 38, 39, 41, 42.

La banda de $5\ GHz$, permite la elección de 24 canales de $20\ MHz$ de ancho de banda, sin solapamiento, pero debido a posibles interferencias con sistemas de radar, sólo se usan los primeros cuatro y los últimos cinco. En el medio, se definen canales para dispositivos capaces de seleccionar de manera dinámica la frecuencia y que controlan la potencia de transmisión, que generalmente no se usan para redes WiFi.

La elección de canales de $20\ MHz$ permite buena velocidad, hasta $54\ Mbps$ por canal, así como un número razonable de canales en la banda. Cada canal de $20\ MHz$ está compuesto por 52 subportadoras espaciadas en $0.3125\ MHz$ .Cuatro subportadoras se usan como piloto para monitoreo de corrimientos e ICI, las otras 48 son para transmitir datos. Tal como se aprecia en la Fig. 8.13, las subportadoras se numeran de -26 a 26. La portadora 0 no se usa por razones de procesamiento. Las portadoras pilotos, canales -21, -7, 7 y 21, transportan una secuencia fija con una modulación convencional.

La capacidad total se mide multiplicando el número de sub-canales por la cantidad de bits por cada subcanal. Por ejemplo, con 6 $bits$, si la modulación fuera 64 QAM, y 48 subcanales se obtiene una capacidad de $48\ x\ 6 = 288\ bits$

por canal de $20\ MHz$. Si se tiene en cuenta el codificador convolucional asociado, resultan $288 \times \frac{3}{4} = 216$ bits por símbolo. La velocidad surge de dividir la cantidad de bits por símbolo por la duración $\frac{216\ bits}{4\mu seg} = 54\ Mbps$.

En cuanto a los esquemas de modulación, IEEE 802.11a usa distintos esquemas de modulación, según el grupo de cuatro velocidades en el que se trabaje: 6 y 9 $Mbps$, 12 y 18 $Mbps$, 24 y 36 $Mbps$, y 48 y 54 $Mbps$. El grupo de menor velocidad usa BPSK codificando 1 bit por sub-canal, o 48 $bits$ por símbolo. La codificación convolucional hace que la mitad, o un cuarto de los bits, sean de redundancia, es decir que quedan sólo 24 o 36 $bits$ de datos por símbolo. El grupo que sigue usa QPSK, codificando 2 $bits$ por sub-canal, o sea 96 $bits$ por símbolo, que terminan representando 48 o 72 $bits$ de datos, debido a la codificación convolucional. El tercer grupo usa 16-QAM y codificación convolucional de $R_c = 1/2$ y $R_c = 3/4$. El grupo de mayor velocidad usa 64-QAM con convolucionales de $R_c = 2/3$ y $R_c = 3/4$. La Tabla 8.8 resume las modulaciones y codificaciones para cada velocidad.

Tabla 8.8 – Métodos de Modulación IEEE 802.11a

Velocidad Mbps	**Modulación y Relación de Código**	**Bits codificados por portadora**	**Bits codificados por símbolo**	**Bits de datos por símbolo**
6	*BPSK* $R_c = 1/2$	1	48	24
9	*BPSK* $R_c = 3/4$	1	48	36
12	*QPSK* $R_c = 1/2$	2	96	48
18	*QPSK* $R_c = 3/4$	2	96	72
24	*16-QAM* $R_c = 1/2$	4	192	96
36	*16-QAM* $R_c = 3/4$	4	192	144
48	*64-QAM* $R_c = 2/3$	6	288	192
54	*16-QAM* $R_c = 3/4$	6	288	216

En esencia, cada canal es una suma multiplexada de 48 secuencias de datos separados. La secuencia de bits debe ser mapeada de la forma correcta. Se podría usar un esquema de *round-robin*: primer bit a la primer sub-portadora, segundo bit a la segunda y así sucesivamente. Sin embargo, 802.11a usa reglas de entremezclado (*interleaving*) para asegurarse que los bits en secuencia se

transmitan en sub-portadoras bien separadas y que las secuencias mapeen en diferentes puntos de la constelación.

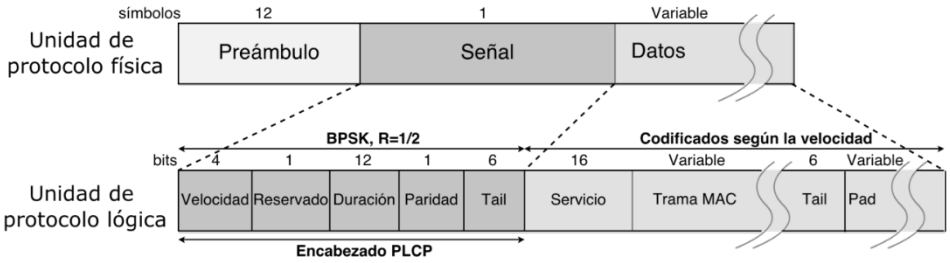


Figura 8.14 – Preámbulo y Encabezado PLCP de trama IEEE 802.11a.

El formato de trama de capa física se presenta en la Fig. 8.14. Un Preámbulo de 12 *símbolos* OFDM precede al encabezado de capa física. Ambos se modulan por BPSK para OFDM, a $6\,Mbps$ con una codificación convolucional de $R_c = 1/2$. El resto se modula y codifica de acuerdo a la velocidad de transmisión.

La Fig. 8.15 presenta el comienzo de la trama, con los intervalos de tiempo de guarda y *windowing* utilizados. El preámbulo se compone de $12\,símbolos$ OFDM, dura $16\,\mu seg$ y está dividido dos secuencias de entrenamiento. Los primeros $10\,símbolos$ forman una secuencia de entrenamiento corta, para que el receptor enganche la señal, seleccione una antena apropiada, si está usando más de una, y sincronice a la velocidad de los símbolos que siguen. A continuación, la secuencia de entrenamiento larga sirve para sincronización fina y se protege con un intervalo de guarda.

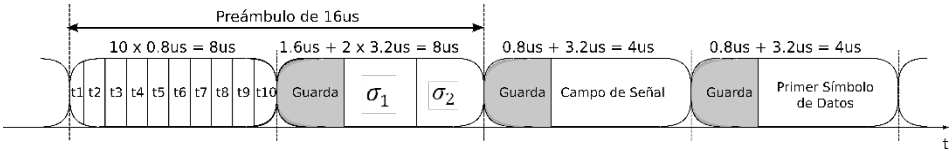


Figura 8.15 - Guardas.

Luego, un símbolo OFDM transporta el campo Señal y otro número variable de símbolos llevan el final del encabezado PLCP, la carga MAC, y el final de la trama. Todos los símbolos se filtran mediante una ventana tipo coseno modificado para asegurar transiciones suaves.

El campo Señal, codifica en sus primeros 4 $bits$ los valores de velocidad (6, 9, 12, 18, 24, 36, 48 o 54 $Mbps$), anunciando el tamaño en bytes de la trama MAC en el campo Duración de 12 $bits$. Luego aparece un bit de paridad par, que

protege a los 16 $bits$ previos. Los 6 $bits$ del campo Tail sirven para finalizar la codificación convolucional.

El encabezado de capa física finaliza con el campo Servicio de 16 $bits$. Curiosamente, este campo se transmite en el campo de datos, antes de la trama MAC, a la velocidad de transmisión elegida. Los primeros 6 $bits$ del campo de Servicio son ¨0¨, ya que su misión es inicializar el mezclador al que se ingresa la propia trama MAC. Los siguientes bits de este campo tienen carácter reservado. A continuación de la propia trama MAC, aparece el *trailer* de capa física, que se compone de dos campos. Los 6 $bits$ del campo Tail sirven para finalizar la codificación convolucional. El campo de bits de relleno se precisa para que la longitud del campo de datos sea un múltiplo del tamaño de bloque de bits de datos transmitidos. El tamaño de este bloque, a su vez, depende del método de modulación seleccionado.

Los parámetros característicos de IEEE 802.11a son:

Tabla 8.9 – Parámetros IEEE 802.11a OFDM 5 GHz

Ranura	**SIFS**	**CW**	**Preámbulo**	**Encabezado PLCP**	**Trama MAC Máxima**
9 μseg	16 μseg	$15 - 1023$ ranuras	16 μseg	4 μseg	4095 $bytes$.

8.6 Capa Física IEEE 802.11g – OFDM 2.4 GHz

IEEE 802.11a logró aumentar la velocidad de operación de las redes WLAN desde 11 $Mbps$ a 54 $Mbps$, aunque su operación en la banda de 5 GHz no proveía interoperabilidad con dispositivos IEEE 802.11 y IEEE 802.11b. La convergencia entre IEEE 802.11a y IEEE 802.11b se concretó con la publicación del estándar IEEE 802.11g. Con este estándar se lograron velocidades comparables a 802.11a, pero operando con mejor alcance, en la banda de 2.4 GHz de 802.11b.

Mientras IEEE 802.11b usa tecnología DS de alta velocidad, IEEE 802.11g usa la misma tecnología por compatibilidad hacia atrás, pero también puede modular los datos en OFDM en la banda de ISM de 2.4 GHz, permitiendo llegar a velocidades de hasta 54 $Mbps$. El uso de estas tecnologías se provee a través de cuatro capas físicas diferentes, definidas en el estándar como Capas Físicas de Velocidad Extendida (ERP, Extended Rate Physicals). De este modo, transmisor y receptor pueden seleccionar una de las cuatro capas, siempre que ambos la soporten.

Las especificaciones de capa física, bajo la denominación ERP son:

- **ERP-DSSS/CCK*:*** Modalidad compatible hacia atrás con IEEE 802.11 (1 y 2 $Mbps$) y IEEE 802.11b (5.5 y 11 $Mbps$). Obligatorio, con preámbulo corto.

- ***ERP-OFDM:*** Es esencialmente IEEE 802.11a en la banda ISM de $2.4\,GHz$, para soporte de velocidades $6, 9, 12, 18, 24, 36, 48,$ y $54\,Mbps$. Cuenta con un tiempo de ranura opcional de $9\,useg$ cuando en el BSS sólo hay dispositivos ERP. Es obligatorio.
- **ERP-DSSS/PBCC*:*** Extensión opcional al estándar PBCC de IEEE 802.11b, para $22\,Mbps$ y $33\,Mbps$. Opcional, generalmente no implementado comercialmente. Usa una codificación convolucional de 256 estados.

- **DSSS-OFDM:** Codificación de encabezados DSSS y de carga OFDM para compatibilidad con IEEE 802.11b. Opcional y no muy implementado. Se trata de modulación híbrida: DSSS para preámbulo y encabezado, y OFDM para los datos.

Los modos obligatorios son como las capas físicas existentes, aunque con pequeños cambios, para asegurar compatibilidad hacia atrás y coexistencia con equipos más viejos. Dado que IEEE 802.11b implementa la especificación original DSSS de IEEE 802.11 y HR/DSSS con CCK que le es propia, IEEE 802.11g adopta ambos estándares para poder coexistir con ellos.

Uno de los desafíos que implicó la definición del estándar IEEE 802.11g fue solucionar los problemas que se presentan cuando estos dispositivos deben convivir con otros IEEE 802.11b. Un mezcla de dispositivos dependiendo del mismo AP obliga a trabajar a la velocidad de IEEE 802.11 b para que la comunicación sea posible. Por ejemplo, las tramas Beacon no pueden transmitirse sino a la mayor velocidad permitida en 802.11b.

Una solución para evitar este tipo de interferencias, al mismo tiempo que se permite a los dispositivos IEEE 802.11g aprovechar al máximo sus capacidades, es la posibilidad de aplicar un Mecanismo de Protección, definido para este estándar para evitar la interferencia de dispositivos IEEE 802.11b durante la transmisión de tramas OFDM y sus respectivas tramas de ACK.

En la Fig. 8.16 se presenta este mecanismo, también conocido con el nombre de CTS a uno mismo (*CTS-to-self*). Cuando una estación IEEE 802.11g precisa proteger una transmisión, emite una trama CTS con dirección MAC destino propia. Para que los dispositivos 802.11b puedan recibirla, esta trama se transmite bajo las especificaciones de la capa física de este protocolo: modulación PSK en $1\,Mbps$ o $2\,Mbps$, o CCK en $5.5\,Mbps$ u $11\,Mbps$. Cualquier estación IEEE 802.11b entenderá estas tramas y podrá actualizar el NAV como corresponde. De este modo, el transmisor se envía una trama CTS a sí mismo, con el objetivo de actualizar el NAV y anunciar al resto cuánto tiempo ocupará el medio. Una vez anunciada la transmisión, el dispositivo IEEE 802.11g puede transmitir su trama y recibir el ACK a la máxima velocidad OFDM. El inconveniente de este método es que la eficiencia puede bajar considerablemente, ya que a los tiempos de transmisión normales habrá que agregarle el tiempo que se consume en transmitir la trama CTS.

Existe un segundo mecanismo, que también se presenta en la Fig. 8.16. Se trata de un intercambio RTS/CTS, más robusto en términos de nodo

escondido, pero que afecta bastante la eficiencia de transmisión. Como en el caso anterior, el tiempo demorado en transmitir las tramas para reservar el medio, puede ser comparable al propio tiempo de transmisión.

Los mecanismos de protección se activarán en el caso de redes con ambos dispositivos y en el caso de redes diferentes colindantes trabajando en el mismo canal. La protección se anuncia en los elementos ERP de la trama Beacon, es decir que en redes de infraestructura es el AP el encargado de decidir la aplicación del mecanismo, en tanto que en redes *ad hoc* el dispositivo encargado de emitir el Beacon debe anunciar el mecanismo. La presencia de dispositivos IEEE 802.11b dispara su utilización.

Cabe aclarar que, en el elemento ERP, existe un bit denominado Usar Protección (*Use Protection*) que les indica a las estaciones el uso de protección cuando está en alto. En el mismo elemento ERP, el bit de Modo de Preámbulo Barker (*Barker Preamble Mode*) se puede usar para indicar a las estaciones el uso de preámbulo corto o largo. La existencia de una única estación que no sea capaz de usar preámbulo corto, obliga a que todas usen preámbulo largo.

No se requiere protección para ERP-PBCC ni para DSSS-OFDM. Ambos comienzan con una cabecera compatible con 802.11b y las estaciones pueden leer el NAV sin necesidad de editar tramas extra. El costo en este caso es que los encabezados son mucho más lentos.

En todos los casos, menos ERP-OFDM, el preámbulo corto es de $120\ bits$, incluyendo el encabezado, bastante más pequeño que el preámbulo largo de $192\ bits$. La diferencia en tiempo es de $96\ \mu seg$. La duración del preámbulo corto es de $96\ \mu seg$ porque los primeros $72\ bits$ del preámbulo se transmiten a $1\ Mbps$, en tanto que los $48\ bits$ del encabezado PLCP se transmiten a $2\ Mbps$. En el caso ERP-OFDM sólo se define un preámbulo y encabezado de $40\ bits$ y $20\ \mu seg$ de duración, igual que en IEEE 802.11a. La Tabla 8.10 presenta un resumen de las posibilidades mencionadas.

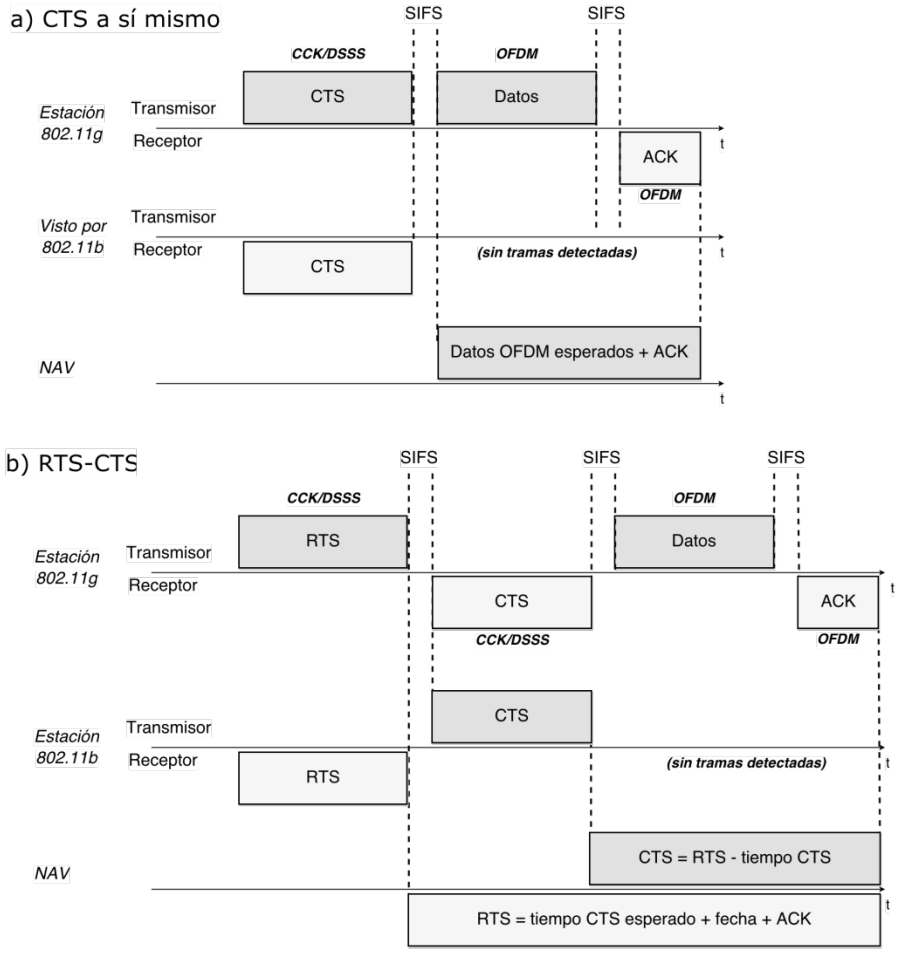


Figura 8.16 - Mecanismos de protección: CTS a sí mismo y RTS/CTS.

Tabla 8.10 – Encabezados PLCP IEEE 802.11g

Capa Física	**Velocidades Soportadas (Mbps)**	**Preámbulo + Encabezado (tiempo)**	**Preámbulo + Encabezado (bits)**
ERP-DSSS (obligatorio)	1, 2, 5.5, 11	Largo 192 μseg	Largo 192 bits

		Corto 96 μseg	Corto 120 bits
ERP-OFDM (obligatorio)	6, 9, 12, 18, 24, 36, 48, 54	20 μseg	40 bits (sólo de encabezado)
ERP-PBCC (opcional)	1, 2, 5.5, 11, 22, 33	Largo 192 μseg Corto 96 μseg	Largo 192 bits Corto 120 bits
DSSS-OFDM (opcional)	6, 9, 12, 18, 24, 36, 48, 54	Largo 192 μseg Corto 96 μseg	Largo 192 bits Corto 120 bits

En IEEE 802.11g, el entramado de capa física es bastante variado, debido a todos los formatos de transmisión posibles. En el caso de ERP-OFDM, el encabezado a nivel físico se presenta en la Fig. 8.17. Es muy parecido al formato de 802.11a, aunque la diferencia es que la trama es seguida por un intervalo de silencio de 6 μseg, que se conoce con el nombre de extensión de la señal. El motivo de este intervalo es lograr que los cálculos de tiempo y velocidades sean idénticos a los de IEEE 802.11a. IEEE 802.11g fija un intervalo SIFS de 10 μseg por compatibilidad con 802.11b, pero agrega estos 6 μseg como extensión de la propia trama para dar tiempo a la finalización correcta de todo el proceso de decodificación. En este sentido, es como si se copiara los 16 μseg que destina IEEE 802.11a para su propio tiempo SIFS. El resto de los campos de capa física son idénticos a los de 802.11a.

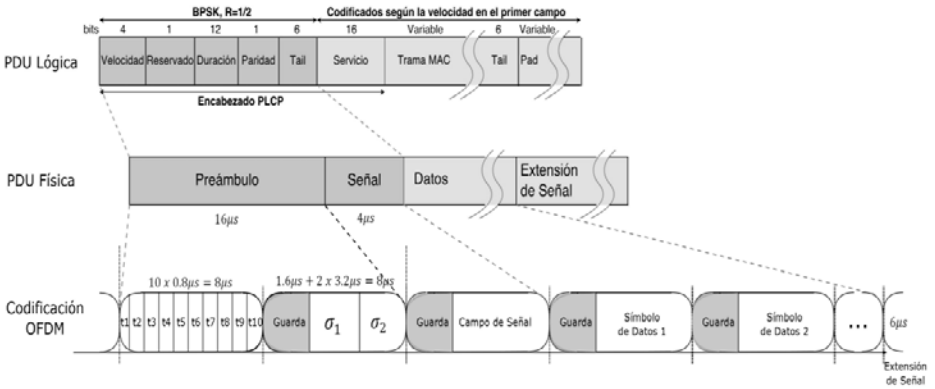


Figura 8.17 – Capa Física ERP-OFDM.

En el modo ERP-DSSS/CCK, los encabezados de capa física son idénticos a los de 802.11b, excepto que ahora el campo Señal agregará el listado de las nuevas velocidades alcanzadas.

Al comparar los diferentes estándares presentados hasta aquí, se observa que a nivel físico existen tres tipos de preámbulos diferentes.

En IEEE 802.11b puede transmitirse un preámbulo largo de $192\ \mu seg$, incluyendo el encabezado, o un preámbulo corto de $96\ \mu seg$. En IEEE 802.11a y en IEEE 802.11g es posible usar un preámbulo OFDM de $20\ \mu seg$ de duración. La diferencia con IEEE 802.11a es que la trama 802.11g va seguida de un tiempo ocioso de $6\ \mu seg$, denominado extensión de la señal, utilizado para ensamblar la trama para su transmisión y para que los cálculos de tiempos y velocidades sean los mismos que en el caso IEEE 802.11a. IEEE 802.11g define un tiempo SIFS de $10\ \mu seg$, por compatibilidad con IEEE 802.11b. En cambio IEEE 802.11a define un intervalo SIFS de $16\ \mu seg$.

El efecto del agregado a nivel de capa física se puede estimar si se considera la transmisión de una trama de $1500\ bytes$, con un encabezado MAC de $34\ bytes$, y la trama ACK asociada.

En el caso IEEE 802.11b a $11\ Mbps$, suponiendo que no hay contienda por el medio, transmitiendo con preámbulo largo, $SIFS = 10\ \mu seg$ y $DIFS = 2xT_{ranura} + SIFS = 50\mu seg$, el tiempo total de transmisión se puede estimar como:

$$DIFS + HPream_{largo} + 1534\ bytes\ x\ \frac{8}{11\ Mbps} + SIFS + HPream_{largo} + 14\ x\ \frac{8}{11\ Mbps} = 50\mu seg + 2x192\mu seg + 1115.63\mu seg + 10\mu seg + 10.18\mu seg \cong 1570\ \mu seg \quad (8.8)$$

En este caso $1500\ bytes$ transmitidos en $1570\ \mu seg$ representan una velocidad real de transmisión de $7.64\ Mbps$. Es decir que el agregado del preámbulo largo implica una reducción de casi 30% de la velocidad.

Por otra parte, si se usara un preámbulo corto en las mismas condiciones, los $1500\ bytes$ de datos se transmitirían en $1378\ \mu seg$, representando una velocidad efectiva de $8.7\ Mbps$. En este caso. La reducción de velocidad es de de alrededor del 20%.

Con el encabezado OFDM de IEEE 802.11g a $54\ Mbps$, $SIFS = 10\ \mu seg$, $DIFS = 2xT_{ranura} + SIFS = 28\ \mu seg$ y preámbulo de $20\ \mu seg$, considerando el agregado de $6\ bits$ para codificación y la máxima velocidad de transmisión:

$$DIFS + 20\ \mu seg\ + \frac{1534x8}{54\ Mbps} + 6\ \mu seg + 10\ \mu seg + 20\ \mu seg + \frac{14x8}{54\ Mbps} + 6\ \mu seg\ \cong 319\ \mu seg \quad (8.9)$$

y la velocidad efectiva será de $37.61\ Mbps$, que se traduce en una reducción cercana al 30%.

En cambio, si la transmisión del ejemplo se realizara bajo la premisa de un mecanismo de protección CTS, por compartir la red con dispositivos 802.11b, se debe usar el mismo valor $DIFS$ de 802.11b, resultando:

$$50\,\mu seg + 192\,\mu seg\ + 14\,x\ \frac{8}{11Mbps} + 10\,\mu seg + 20\,\mu seg + 57\,símb\ x4\ \frac{\mu seg}{símb} + 6\,\mu seg\ + 10\,\mu seg + 20\,\mu seg + \ 1\,x\ \frac{4\,\mu seg}{símb} + 6\,\mu seg = 557\,\mu seg \tag{8.10}$$

En este último caso, se ha considerado la transmisión de la trama CTS con preámbulo largo, resultando una velocidad efectiva reducida drásticamente a $21.54\ Mbps$.

Además del *overhead* mencionado, es interesante tomar en cuenta la elección de los parámetros relacionados con el método de acceso al medio. Como hemos explicado, el método usado es CSMA/CA y el tiempo de demora de acceso al medio se mide en ranuras, cuyo valor fija el estándar. En IEEE 802.11b, la duración de la ranura es de $20\ \mu s$. En IEEE 802.11g funcionando en modo OFDM es de $9\ \mu s$.

En el primer caso, la transmisión de una trama de $1500\ bytes$ a $11\ Mbps$, insumirá un tiempo de $1115.63 \mu seg$. Tomando una ventana de contienda de 15 ranuras en promedio, se agregan $300\ \mu s$ a la transmisión, haciendo que la velocidad efectiva sea $8.5\ Mbps$. O sea que el tiempo de *backoff* reduce la velocidad efectiva en alrededor de 23%. En el caso de IEEE 802.11g, considerando la ranura de $9 \mu s$ y el mismo tiempo promedio de contienda, la transmisión de la misma trama a $54\ Mbps$ insumirá un tiempo de $362\ \mu s$. La velocidad efectiva cae a $33\ Mbps$, resultando en una reducción cercana al 40%.

En la Tabla siguiente se presentan los parámetros característicos de IEEE 802.11g:

Tabla 8.11 – Parámetros IEEE 802.11g OFDM 2.4 GHz

Ranura	**SIFS**	**CW**	**Preámbulo**	**Tiempo de Extensión**	**Trama MAC Máxima**
9 µseg (OFDM) 20 µseg(HR/DS)	10 µseg	15 – 1023 ranuras	20 µseg	6 µseg	4095 $bytes$.

8.7 Capa Física IEEE 802.11n – OFDM MIMO 2.4 GHz y 5 GHz

El estándar IEEE 802.11n ofrece muchos beneficios respecto de las tecnologías previas, mejorando fundamentalmente aspectos relativos a la

velocidad y confiabilidad, incluyendo cambios a nivel de capa física y en la MAC. La velocidad es mejorable si se trabaja desde el punto de vista físico con nuevos esquemas de diversidad o, desde un punto de vista MAC, si se intenta mejorar la eficiencia de transmisión de las tramas.

Curiosamente, el estándar 802.11n fue ratificado en 2009 como extensión de los estándares 802.11a/g, agregando MIMO en la capa física, aunque había comenzado a comercializarse desde 2007, teniendo en cuenta simplemente las especificaciones propuestas en el borrador (*draft*). Seguramente, esto se debió a que los AP ya usaban para ese entonces antenas múltiples, pero bajo un concepto diferente: la elección de la mejor antena para transmitir o recibir una trama, pero siempre una antena por vez.

La técnica innovadora más importante usada por este estándar en capa física se conoce con el nombre de Entrada Múltiple Salida Múltiple (**MIMO**, Multiple Input Multiple Output) y se refiere a la existencia de más de una antena del lado transmisor o del lado receptor. El propósito detrás de esta técnica es mejorar la relación SNR del lado receptor.

A fines de los 90, comenzaron a surgir nuevos estudios en comunicaciones inalámbricas que incluían múltiples antenas tanto del lado receptor como del lado transmisor, prometiendo mejor performance sobre todo en entornos con muchos problemas de *scattering*, como es el caso de las WLAN 802.11 en 2.4 o 5 GHz. Lo revolucionario del sistema era la propuesta de utilizar el espacio como una variable más del problema. Esto se tradujo en la división espacial de caminos entre transmisor y receptor, de tal modo que agregando antenas, se producía un escalamiento lineal de la capacidad de la comunicación, aunque las antenas transmitieran y recibieran en la misma banda de frecuencia al mismo tiempo. Sencillamente expresado, en vez de mitigar el efecto del *fading* multicamino, la técnica intenta justamente aprovechar la propia naturaleza del medio no guiado.

Un esquema convencional de radiocomunicaciones utiliza una antena para transmisión y otra para recepción. Desde el punto de vista del canal, se denomina Entrada Única Salida Única (SISO, Single Input Single Output) a este tipo sistemas. Su comportamiento se rige de acuerdo al Teorema de Shannon, que presenta una relación entre la capacidad C de un canal, su ancho de banda B y la relación SNR presente. El Teorema establece un límite sobre la cantidad de datos de información sin errores que puede ser transmitida con un ancho de banda establecido, en presencia de ruido interferente AWGN:

$$C = B\ log_2(1 + SNR) \tag{8.11}$$

Es decir que el aumento de la capacidad es una función logarítmica de la relación SNR. Esto significa que, para lograr aumentos apreciables en la capacidad, la potencia debería aumentarse enormemente ya que la relación no es lineal.

Un sistema MIMO transmite múltiples señales de RF al mismo tiempo, intentando obtener ventaja del efecto *multipath*. A cada señal que se envía desde

una antena, usando un transmisor propio, se la denomina flujo espacial (*spatial stream*). La separación entre las antenas transmisoras implica diferente camino hacia destino para cada flujo. Este efecto se conoce como **diversidad en el espacio**. Cada transmisor podría manejar un flujo de datos diferente del resto. Del lado receptor, también cada antena cuenta con su propio circuito de recepción que decodifica la señal entrante de manera independiente. Cada uno recibe la combinación de las señales transmitidas. Se logra mejorar la SNR, pero se deben utilizar herramientas matemáticas complejas para lograr este objetivo.

En el caso de un sistema MIMO, típicamente existirán m antenas transmisoras y n antenas receptoras, tal como se presenta en la Fig. 8.18. A través del mismo canal, cada antena no sólo recibe los componentes directos destinados para ella, sino también los componentes indirectos transmitidos a otras antenas. El canal se supone caracterizado por determinados parámetros. Por ejemplo, la conexión directa entre la antena transmisora número 1 y la antena receptora del mismo número, se especifica con el parámetro h_{11}. El parámetro h_{21}caracteriza la conexión indirecta entre la antena de transmisión 1 y la de recepción 2. Considerando todas las transiciones, se puede obtener una matriz de transición, de dimensiones nxm:

$$H = \begin{pmatrix} h_{11} & h_{12} \ldots . & h_{1m} \\ h_{21} & h_{22} \ldots . & h_{2m} \\ h_{n1} & h_{n2} & h_{nm} \end{pmatrix} \tag{8.12}$$

La pretensión de MIMO es lograr una relación menos dramática entre la mejora de la capacidad del canal y la potencia transmitida. El costo se traduce en un aumento en la complejidad del sistema. Desde un punto de vista de mitigación de *fading*, MIMO se relaciona con técnicas de diversidad. Es una manera de aumentar las dimensiones de la comunicación, ya que el agregado de múltiples antenas, del lado transmisor o del lado receptor provee nuevos conjuntos de caminos independientes. Esta posibilidad se puede explotar de varias maneras.

Una manera de usar MIMO aprovechando técnicas ya conocidas para mejorar la relación SNR se basa en coordinar las señales transmitidas por más de una antena para mejorar la calidad de la señal recibida. Esta técnica se conoce con el nombre de **Formateo del Haz en Transmisión** (***transmit beamforming***). Se suele usar en los casos en los que se cuenta con una única antena de recepción y existen pocas obstrucciones o superficies receptoras en el medio.

La manera de lograr una mejora de la relación *SNR* en esta técnica consiste en ajustar las fases de las antenas transmisoras para lograr un máximo de potencia recibida del lado de la antena receptora. Este ajuste es muy difícil de lograr sin información proveniente del receptor. Esta realimentación es posible sólo en dispositivos IEEE 802.11n y debe hacerse toda vez que cambie el canal, o que haya un movimiento relativo significativo en cuanto a la longitud de onda de trabajo entre los dispositivos involucrados.

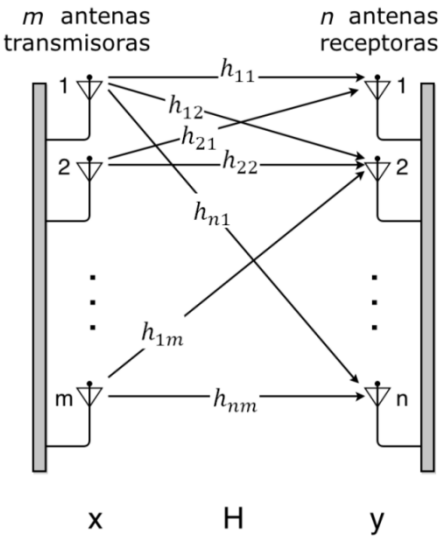


Figura 8.18 - Sistema MIMO.

El estándar IEEE 802.11n define algunos métodos para esta técnica. Un método, conocido como *beamforming* explícito, requiere que el receptor mida el canal y transmita esta información al transmisor, para que éste ajuste los parámetros de transmisión adecuadamente. En otro método, conocido como *beamforming* implícito, es el transmisor el que mide el canal y usa estas mediciones para el ajuste de los parámetros. Este método es un poco más complejo, puesto que requiere un intercambio adicional entre ambos extremos de la comunicación para el ajuste preciso de la misma.

El haz de transmisión se formatea por el ajuste de las fases de las señales transmitidas por las múltiples antenas, de tal manera de lograr el efecto de una única antena direccional de alta ganancia sobre el sistema receptor.

En general, son los AP los dispositivos capaces de realizar este trabajo, enviando el mismo flujo espacial hacia el cliente. En el caso de redes WLAN, la técnica es un tanto limitada, ya que sólo es posible de ser aplicada en transmisiones *unicast*, dirigidas a un único receptor. En este sentido, las tramas Beacon no se verían beneficiadas por esta técnica, impidiendo la mejora en lo que respecta al área de cobertura de la red.

La técnica de **Diversidad Espacial** transmite o recibe flujos redundantes de información en paralelo sobre los distintos caminos, para mejorar la confiabilidad y el rango, basándose en que es improbable que todos los caminos se degraden al mismo tiempo. No se mejora la velocidad puesto que el propósito es lograr una transmisión más robusta. Los datos se dividen en flujos redundantes y cada flujo espacial se transmite desde su propia antena, con un transmisor propio.

Consideremos, por ejemplo, el arreglo de la Fig. 8.19, también conocido como **Diversidad en Recepción**, que consta una antena transmisora y dos receptoras, emulando un sistema **SIMO**. Con este esquema, cada antena recibe

una copia de la señal transmitida, modificada por el canal. Las ganancias del canal, modeladas por medio de parámetros h_{ij}, son números complejos que representan la atenuación en amplitud y el corrimiento de fase sobre el canal. El receptor mide las ganancias del canal según una secuencia de entrenamiento agregada en el preámbulo de la trama. La ganancia es diferente sobre cada sub-portadora en el caso de *fading* selectivo en frecuencia, así como también sobre cada antena. La cuestión es cómo combinar las dos señales recibidas. Se consideran dos técnicas de diversidad en este sentido.

El método más sencillo consiste en usar la antena con mayor señal, o sea la de mayor relación SNR, para la recepción del paquete e ignorar las demás. Este método se denomina ***Combinación Selectiva*** y es la forma en que trabajan los AP con antenas múltiples en 802.11a/g. Es un método que contribuye con la confiabilidad de la transmisión, pero no aprovecha la potencia recibida de las antenas no seleccionadas.

El mejor método consiste en sumar las señales de ambas antenas, pero esto no se puede realizar por simple superposición, pues se potenciarían los efectos del *fading multipath*. Más aún, el mejor aprovechamiento significaría que cada señal deberían retrasarse, hasta que todas se encuentren en fase. Esto permitiría sumarlas de manera coherente, como se observa en la Fig. 8.19. Para poder trabajar de esta manera, cada antena debe ir acompañada de su propia cadena de RF. Es decir que se aumenta la complejidad del hardware y el consumo de potencia, aunque se logra una mejor performance. Las señales se pesan por sus propias relaciones SNR, dando menor peso a aquella señal con mayor contaminación de ruido, para no amplificar el efecto de la señal indeseada. El resultado se conoce como ***Combinación de Máxima Relación***. Es importante notar que, en presencia de *fading* selectivo en frecuencia, el proceso se lleva a cabo sobre cada sub-portadora de acuerdo a la ganancia específica por canal.

En el ejemplo presentado en la figura, las dos ganancias del canal tienen magnitudes de valor 3 y 2, con potencia esperada de ruido de valor 1, es decir SNR de 9 y 4 respectivamente. El receptor de *Máxima Relación* escala cada una de las señales de las antenas por su magnitud normalizada al total, retrasa las señales a una fase de referencia común, y luego las suma. El resultado tiene magnitud $\sqrt{13}$ y la suma pesada normalizada de ruido todavía mantiene el valor de potencia media en 1. Es decir que se logra que la señal combinada posea una SNR de 13.

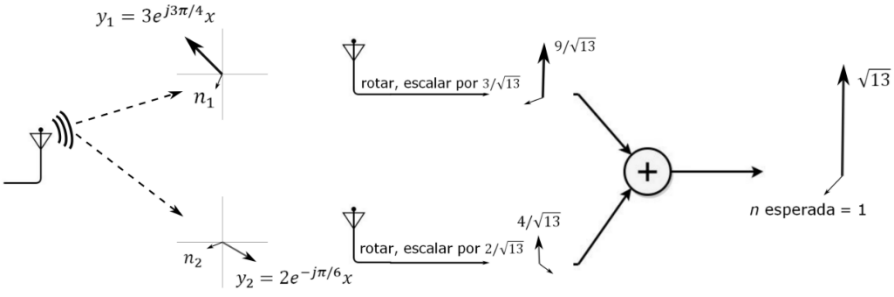


Figura 8.19 - Diversidad Espacial.

SIMO tiene la ventaja de ser relativamente fácil de implementar aunque la desventaja es que la carga de procesamiento se encuentra en el receptor, aumentando costos y consumo de batería.

La **Diversidad en Transmisión** implica la existencia de más antenas de transmisión. El escenario más sencillo utiliza dos antenas transmisoras y una receptora. Se conoce como **MISO** o $2x1$. En este caso, se transmiten los mismos datos de forma redundante sobre ambas antenas. El equivalente de ***Combinación Selectiva*** simplemente selecciona la mejor antena para la transmisión. En el caso de ***Combinación de Máxima Relación***, el equivalente es parecido al caso de *beamforming*. El transmisor pre-codifica las señales, retrasando sus fases de manera de lograr una combinación constructiva del lado receptor, y ponderándolas de tal manera que la potencia transmitida se distribuya a cada camino espacial según su propia relación SNR. La desventaja de la diversidad del lado transmisor comparado con la misma técnica del lado receptor, es que el transmisor debe conocer el canal con antelación, ya sea para seleccionar la antena de transmisión o para pre-codificar las señales. La carga de procesamiento se presenta del lado transmisor, pero ello requiere alguna forma de realimentación dinámica del lado receptor, ya que el canal es variante en el tiempo.

Cuando existe más de una antena, tanto del lado transmisor como del lado receptor, el sistema se conoce como **MIMO**. Estos esquemas se usan para mejorar la robustez y la velocidad de transmisión, pero es necesario aplicar algún tipo de codificación sobre los canales para separar los datos provenientes de diferentes caminos.

Una de las técnicas usadas en MIMO es la utilización del sistema como un **Multiplexor Espacial**, para aumentar la capacidad del canal. En este caso, el número de antenas receptoras debe ser igual o mayor que el número de antenas transmisoras.

Según hemos visto, al existir una variedad de caminos de transmisión, cada uno con su propia característica en cuanto al canal, la señal recibida en cada antena será una combinación de las transmitidas afectadas por sus propios coeficientes, representativos del estado del canal. Para poder recuperar el flujo de datos transmitido se debe realizar un trabajo considerable desde el punto de vista del procesamiento de la señal recibida. Por ejemplo, el decodificador MIMO debe estimar los parámetros individuales h_{ij} para poder hallar la matriz de transferencia. El principio básico de la recepción consiste en, una vez estimada la matriz H, reconstruir los flujos transmitidos por multiplicación del vector recibido con la inversa de la matriz, H^{-1}. Se trata de resolver un conjunto de n ecuaciones lineales, para encontrar n variables, siendo n el número de antenas transmisoras.

Para poder utilizar correctamente el esquema de multiplexado es preciso agregar codificación a los canales, para que el receptor pueda detectar correctamente los datos.

La Codificación Espacio Tiempo habilita la transmisión de múltiples copias de un flujo de datos sobre cierta cantidad de antenas. En recepción, se combinan todas las copias recibidas, de manera óptima, para poder extraer la

mayor cantidad de información posible. La transmisión de múltiples copias ayuda a contrarrestar los problemas del canal, ya que el flujo de datos se codifica por bloques, distribuyéndose entre las antenas transmisoras, combinando de esta forma diversidad en el tiempo y en el espacio.

Esta posibilidad distingue el esquema claramente frente a los de diversidad, donde la mejora se produce sobre la relación SNR y, aunque el aumento mantiene una relación lineal con este parámetro, se mantiene la relación logarítmica respecto de la capacidad, tal como se resume en la Tabla 8.12.

Tabla 8.12 – Comparación entre esquemas de antenas.

Método	**Capacidad**
SISO	$C = B\ log_2(1 + SNR)$.
Diversidad (1xN) o (Nx1)	$C = B\ log_2(1 + N\ x\ SNR)$.
Diversidad (NxN)	$C = B\ log_2(1 + N^2\ x\ SNR)$.
Multiplexado Espacial	$C = B\ x\ N\ log_2(1 + SNR)$.

El objetivo principal detrás de la utilización de MIMO y de las técnicas usadas en IEEE 802.11n fue elevar la velocidad máxima lograda hasta entonces, de 54 $Mbps$, a centenares de $Mbps$. Desafortunadamente, cada incremento de velocidad que se había logrado hasta la aparición de IEEE 802.11n era a expensas de una pérdida de rango. Por ejemplo, la velocidad máxima de 54 $Mbps$ se alcanza mediante la técnica de modulación 64-QAM, muy eficiente desde el punto de vista espectral, pero se precisa de una mayor relación SNR si se la compara con una técnica más simple, tal como la de BPSK que se usa para la velocidad de 1 $Mbps$. El resultado es una pérdida de rango, ya que el enlace se vuelve más vulnerable a la interferencia co-canal, lo que reduce la capacidad total del sistema. La solución para aumentar la velocidad e incrementar el rango se obtuvo combinando MIMO con OFDM.

En MIMO-OFDM se incrementa la capacidad del enlace por transmisión simultánea de múltiples flujos de datos por medio de múltiples antenas de transmisión y de recepción.

Cuando en Julio de 2003, se creó el grupo de tareas de 802.11n con el objetivo de definir un nuevo estándar para WLAN, las propuestas presentadas coincidieron en tres aspectos: la utilización de MIMO-OFDM, la utilización de canales de 20 y 40 MHz, y técnicas de agregado de paquetes. En el año 2005 se publicó el primer borrador del estándar que definió un rango de velocidades obligatoria y opcionales en ambos canales, de 20 y 40 MHz, en 2.4 GHz y 5 GHz. En la Tabla 8.13 se presenta una lista de los Esquemas de Codificación y Modulación MCS obligatorios, con sus correspondientes velocidades. Aparte de estas velocidades, existen otras opcionales.

IEEE 802.11n utiliza símbolos OFDM de 4 μseg, como en 802.11a y 802.11g, aunque aumenta el número de sub-portadoras de 48 a 52. Con esto se logra aumentar la velocidad a un máximo de 65 $Mbps$, para un sistema de transmisión simple. Con dos transmisores se alcanza un máximo de

130 $Mbps$. Existen modos opcionales de tres transmisores que permiten alcanzar 195 $Mbps$ y con cuatro se llega a 260 $Mbps$.

La utilización de canales de 40 MHz, aumenta el número de sub-portadoras a 108, lo que provee velocidades de 135 $Mbps$, 270 $Mbps$, 405 $Mbps$ y 540 $Mbps$, para uno hasta cuatro transmisores, respectivamente.

Una opción consiste en reducir el intervalo de guarda a 400 ns, incrementando de este modo la velocidad máxima a 300 $Mbps$, con dos flujos espaciales en un canal de 40 MHz. Otras opciones utilizan hasta 3 o aún 4 flujos espaciales. La velocidad opcional máxima es de 600 $Mbps$, que se logra con 4 flujos espaciales en un canal de 40 MHz e intervalos de guarda de 400 ns.

Para mejorar la ineficiencia generada por la sobrecarga propia del método de acceso, el estándar introdujo algunos cambios a nivel MAC.

Previamente, hemos calculado cómo se afecta la velocidad efectiva de transmisión debido a los preámbulos de capa física. Esta situación se empeora en el caso de tramas *unicast* que, obligatoriamente, llevan asociadas tramas de ACK. Para resolver este problema, 802.11n introduce la posibilidad de agregar tramas. Básicamente, se permite colocar dos o más tramas juntas en una única transmisión, aumentando el tamaño de los datos transmitidos, al mismo tiempo que se minimiza la cantidad de ACK asociados.

Existen dos métodos: agregado de Unidades de Datos de Servicio MAC (MSDU) y agregado de Unidades de Datos de Protocolo MAC (*MPDU*). Ambos permiten un único preámbulo de capa física para una transmisión agregada.

Al permitir la transmisión de varias tramas juntas, se reduce la cantidad potencial de colisiones y, por ende, el tiempo de *backoff*. Por este motivo se modifica el tamaño de trama máxima, incrementándoselo de 4 $KBytes$ a 64 $KBytes$.

El agregado de MSDU se utiliza para juntar varias tramas *Ethernet* en una única trama 802.11. Este tipo de agregado es el más eficiente debido a la longitud más corta de los encabezados *Ethernet* respecto de los encabezados 802.11. De esta manera, una única trama o encabezado 802.11 transporta múltiples tramas 802.3. La trama agregada puede cargar hasta 7935 $bytes$, unas 5 tramas de tamaño máximo y más de 100 tramas de tamaño mínimo. La trama completa se protege con un único CRC. Si en la recepción se verifican errores, se debe retransmitir la trama completa agregada.

Tabla 8.13

Índice MCS	Tipo	Relación de Código	Flujos Espaciales	Velocidad (Mbps) Canales de 20 MHz		Velocidad (Mbps) Canales de 40 MHz	
				800ns	400ns	800ns	400ns
0	BPSK	1/2	1	6.50	7.20	13.50	15.00
1	QPSK	1/2	1	13.00	14.40	27.00	30.00
2	QPSK	3/4	1	19.50	21.70	40.50	45.00
3	16QAM	1/2	1	26.00	28.90	54.00	60.00
4	16QAM	3/4	1	39.00	43.30	81.00	90.00

5	64QAM	2/3	1	52.00	57.80	108.00	120.00
6	64QAM	3/4	1	58.50	65.00	121.50	135.00
7	64QAM	5/6	1	65.00	72.20	135.00	150.00
8	BPSK	1/2	2	13.00	14.40	27.00	30.00
9	QPSK	1/2	2	26.00	28.90	54.00	60.00
10	QPSK	3/4	2	39.00	43.30	81.00	90.00
11	16QAM	1/2	2	52.00	57.80	108.00	120.00
12	16QAM	3/4	2	78.00	86.70	162.00	180.00
13	64QAM	2/3	2	104.00	115.60	216.00	240.00
14	64QAM	3/4	2	117.00	130.00	243.00	270.00
15	64QAM	5/6	2	130.00	144.40	270.00	300.00

Para una transmisión desde un dispositivo móvil, dentro de un BSS del tipo *Infraestructura*, el encabezado contendrá la dirección fuente y la del AP. En los encabezados internos aparecerían las direcciones destino, que pueden ser múltiples. En cambio, en una transmisión desde un AP, se pueden combinar tramas provenientes desde diferentes dispositivos pero con el mismo destino final.

En el caso de agregado MPDU, se pueden enviar hasta 64 tramas 802.11 juntas, cada una con su propio encabezado MAC, su propio CRC y su propia carga de datos, hasta $4095\ bytes$. Cada trama agregada puede transportar hasta $64\ Kbytes$ de datos. De este modo, en el caso de recepción con errores, la retransmisión puede realizarse de manera selectiva. Esta mejora se complementa con otra facilidad incorporada a la MAC, que se conoce como ACK de Bloque (*BA*), que permite indicar cuáles sub-tramas del agregado se han recibido correctamente.

Cuando no es posible realizar agregado, 802.11n provee un mecanismo de reducción de sobrecarga asociado a la transmisión de un flujo de tramas a diferentes destinos. La extensión 802.11e para Calidad de Servicio (QoS, Quality of Service) agrega una funcionalidad, a un único transmisor, para transmitir una ráfaga de tramas durante una única oportunidad de transmisión. En esta ocasión, el transmisor no precisa realizar un *back off* aleatorio entre transmisiones, separando sus tramas por el menor de todos los tiempos permitidos, SIFS. 802.11n mejora este mecanismo, definiendo un nuevo tiempo, aún más pequeño, conocido como RIFS (*Reduced IFS*). La única desventaja es que el uso del mecanismo queda restringido a redes tipo *greenfield*, es decir aquellas donde no hay dispositivos 802.11a, b, o g.

La compatibilidad con dispositivos más viejos es crítica, aunque 802.11n presenta varios mecanismos para proveer compatibilidad hacia atrás. Por ejemplo, se ofrece una forma de operación protegida de modo mezclado similar a la de 802.11g. En este modo de operación, 802.11n transmite un preámbulo y un campo de Señal que pueden ser decodificados por dispositivos 802.11a y g. De este modo, se anuncia la presencia de señal y el tiempo de ocupación del medio. A continuación de estos campos, la información se transmite a la velocidad de 802.11n y sobre múltiples flujos espaciales.

En cuanto a la migración, no sólo se trata de la adquisición de nuevos AP, sino que se deberán tener en cuenta algunos aspectos importantes:

- IEEE 802.11n opera tanto en la banda de 2.4 GHz, como en la banda de 5 GHz. Las restricciones de operación serán muy diferentes en cada caso. El uso de canales de 40 MHz en la banda de 2.4 GHz no es recomendable, ya que el resto de la banda se vería interferida por esta transmisión. Es preferible en este caso el uso de canales de 20 MHz, como es el caso de los dispositivos 802.11b y g. En todo caso, de extenderse sobre un segundo canal de 20 MHz, éste deberá estar libre de dispositivos 802.11b y g, situación bastante improbable al momento de una migración. Por otra parte, aún cuando la migración fuese posible, no hay suficiente ancho de banda disponible en 2.4 GHz como para cubrir el despliegue de una WLAN mediana.

- En cuanto al canal de 5 GHz, en los últimos años se han liberado algunas condiciones regulatorias, aumentando el número de canales. Esto permite un despliegue cómodo de redes *802.11n*, aún cuando se estén utilizando canales de 40 MHz.

- La carga que ofrecerá este tipo de redes a la conexión *Ethernet* también es un detalle a tener en cuenta. Por este motivo, seguramente una migración a 802.11n se deberá acompañar de una actualización a 1 $Gbps$ de la *Ethernet*, para evitar cuellos de botella.

8.8 Comparación de Estándares

Tabla 7.14 - Comparación de los estándares 802.11

Parámetro	**802.11**	**802.11**	**802.11b**	**802.11g**	**802.11a**	**802.11n**
Codificación	FHDSS 2GFSK / 4GFSK	DSSS DBPSK/ DQPSK	HR/DSSS CCK/ DQPSK	BPSK, QPSK, 16-QAM, 64-QAM OFDM	BPSK, QPSK, 16-QAM, 64-QAM OFDM	BPSK, QPSK, 16-QAM, 64-QAM OFDM-MIMO
Banda	2.4 GHz	2.4 GHz	2.4 GHz	2.4 GHz	5 GHz	2.4 y 5 GHz

Canales	20 MHz	20 MHz	20 MHz	20 MHz	20 MHz : 48 $datos$ - 4 Piloto	$52 + 4$ / 20 MHz $108 +$ 6 / 20 MHz
Velocidad	1 , 2 $Mbps$	1 , 2 $Mbps$	Agrega 5.5 y 11 Mbps	Hasta 54 $Mbps$	54 $Mbps$	130 $Mbps$ Obligatorio 600 $Mbps$ Opcional
SIFS	28 µseg	10 µseg	10 µseg	10 µseg	16 µseg	16 µseg Posibilidad *RIFS*
Ranura	50 µseg	20 µseg	20 µseg	20 µseg 9 µseg	9 µseg	9 µseg
Encabezados capa física	128 µseg	192 µseg	192 µseg / 96 µseg	192 μseg / 96 μseg / 20 μseg	20 µseg	20 µseg
Intervalo de Guarda					800 ns	400 ns/800 ns

Bibliografía

1. Official IEEE Standards Association web site http://standards.ieee.org/about/get/802/802.11.html
2. Official industry association web site http://www.wi-fi.org/
3. Proakis, John, Salehi, Masoud, "Communication Systems Engineering". Prentice Hall, 1994.
4. "Complementary Code Keying Made Simple". Application Note, Intersil, May 2000. http://www.eetasia.com/ARTICLES/2001MAY/2001MAY25_NTEK_DSP_AN.PDF
5. Gast, Matthew, "802.11 Wireless Networks: The Definitive Guide". O'Reilly, 2002.
6. Gast, Matthew, "When Is 54 Not Equal to 54? A Look at 802.11a, b, and g Throughput". Updated:8/14/2003 http://www.oreillynet.com/pub/a/wireless/2003/08/08/wireless_throughput.html
7. Vassis, Dimitris, Kormentzas, George, Rouskas, Angelos and Maglogiannis, Ilias "The IEEE 802.11g Standard for High Data Rate WLANs". IEEE Network, May/June 2005.
8. Yasser Ahmed Abbady, Mohab Mangoud, "OFDM Basics". Wireless communication course (fall 2006). http://www.angelfire.com/planet/wiresem/OFDM%20Basics.pdf
9. "802.11n Primer". Whitepaper, AirMagnet, 2008. http://airmagnet.flukenetworks.com/assets/whitepaper/WP-802.11nPrimer.pdf

10. Paul,Thomas, Ogunfunmi, Tokunbo, "Wireless LAN Comes of Age: Understanding the IEEE 802.11n Amendment". IEEE Circuits and Systems Magazine, 2008. http://www.gao.ece.ufl.edu/EEL6502/ieee_circuits_systems.pdf.
11. Lozano, Angel, Jindal, Nihar, " Transmit Diversity vs. Spatial Multiplexing in Modern MIMO Systems". IEEE Transactions on Wireless Communications, VOL. 9, NO. 1, JANUARY 2010.
12. "Meraki White Paper: 802.11n Technology". 2011. https://meraki.cisco.com/lib/pdf/meraki_whitepaper_802_11n.pdf
13. "802.11n Primer". Aerohive Networks, Inc, 2011. http://www.aerohive.com/pdfs/Aerohive-Whitepaper-0211n_Technology_Primer.pdf
14. Kolap, Jyoti, Krishnan, Shoba, Shaha, Ninad, "Frame Aggregation Mechanism for High Throughput 802.11N WLANS". International Journal of Wireless & Mobile Networks (IJWMN) Vol. 4, No. 3, June 2012. http://airccse.org/journal/jwmn/0612wmn09.pdf
15. Schwengler, Thomas, "Wireless & Cellular Communications, Class Notes for TLEN-5510, Fall 2013". http://morse.colorado.edu/~tlen5510/text/classweb.html#classwebli1.html
16. Sklar, Bernard, "The Characterization of Fading Channels". http://faraday.ee.emu.edu.tr/ee569/art_sklar5_fading.pdf

Problemas

1. Explique las diferencias entre desvanecimiento de pequeña y gran escala.

2. Explique conceptualmente los parámetros que caracterizan el efecto de desvanecimiento en el canal inalámbrico, tanto en el dominio del tiempo como en el dominio de la frecuencia. Haga estimaciones de los valores de los mismos para la banda de $5\ GHz$.

3. ¿Porqué un canal de IEEE 802.11 ocupa $22\ MHz$ de ancho de banda?

4. Investigue la separación y cantidad de canales disponibles en las bandas de 2.4 y $5\ GHz$.

5. ¿Porqué usar OFDM para aumentar la velocidad en lugar de una técnica de modulación convencional?

6. ¿Cuál es la relación entre el tiempo de guarda de un símbolo OFDM y el desparramo del retardo en un canal de *fading*?

7. ¿Qué es un prefijo cíclico y porqué se elige esta técnica en OFDM?

8. Explique cómo se modulan los datos para alcanzar la velocidad de $54\ Mbps$ en 802.11a?

9. ¿Se puede transmitir en OFDM en 802.11g en presencia de dispositivos 802.11b?

10. ¿Qué diferencias técnicas existen entre 802.11n y 802.11g?

11. Investigue la aparición de nuevos protocolos y compare con los presentados en este capítulo.

Parte III

TCP / IP

CAPÍTULO IX

Protocolo IPv4

La interconexión de redes exige la presencia de elementos de cooperación denominados dispositivos de encaminamiento, también conocidos como enrutadores (routers). Se trata de dispositivos especialmente dedicados al procesamiento de paquetes, que se convierten en piezas fundamentales en las redes WAN pues en ellos se efectúan las decisiones necesarias para transportar los paquetes a las redes destino.

En pos de este objetivo, en los propios elementos de decisión, se almacena información en forma de una Tabla de Enrutamiento. La existencia de esta Tabla, que es consultada por el router al momento de tomar la decisión de encaminamiento, implica también la existencia de algún mecanismo de carga de la información en la propia tabla y de mantenimiento y modificación de la misma ante posibles cambios de la situación operativa de la red.

Por su parte, los propios paquetes deberán transportar algún tipo de información que permita a los routers efectuar la decisión de encaminamiento. La existencia de una gran red exige la especificación de direcciones con significado global, cuyo propósito es la identificación unívoca de los protagonistas de la comunicación. A su vez, los extremos transmisor y receptor generalmente se encuentran instalados en sistemas más pequeños, también llamados redes locales, donde se necesita una identificación, aunque de significado físico y acotado.

En este escenario, una de las primeras inquietudes que surge se refiere a la relación entre las direcciones globales y las direcciones de significado local. Otra inquietud, determinada por la propia tecnología de interconexión, se relaciona con el manejo de la diferencia de velocidades entre distinto tipo de redes. A su vez, en el nivel de las aplicaciones, se deberán solucionar ciertos desafíos para lograr que varias aplicaciones trabajen al mismo tiempo comunicándose con distintos destinos.

En cualquier caso, el objetivo final es que dos estaciones diferentes, con distinto hardware y/o Sistema Operativo, probablemente ubicadas en redes de diferente tecnología o filosofía de funcionamiento, puedan tener las capacidades apropiadas para que poder comunicarse entre sí, a través de una red de redes. En este tipo de comunicación existen varios protagonistas: Internet, routers, dispositivos de usuario y protocolos.

Este capítulo introduce los aspectos más importantes del Protocolo de Internet, conocido como IP, con el que se ofreció una respuesta al problema del traslado de paquetes, desde un nodo fuente a un nodo destino. La versión 4 de este protocolo, IPv4, fue capaz de sobrellevar el crecimiento explosivo de Internet y aún se usa en nuestros días, aunque el problema de vaciamiento de direcciones hizo necesario diseñar un reemplazo, más acorde a las necesidades actuales de las redes de datos.

9.1 Funcionalidad Asociada al Protocolo de Red de Internet

El Protocolo de Internet (IP, Internet Protocol) fue diseñado para que ofreciera un servicio de entrega de paquetes sin conexión, sobre una base de paquete por paquete, siguiendo una modalidad no confiable. Se denominó a este tipo de entrega ***best effort***, interpretándose como entrega del mejor esfuerzo, sin garantías. Con este mecanismo, cualquier paquete arribado con errores, simplemente se descarta, quedando en manos de capas superiores la recuperación por la pérdida de paquetes en la red. Esta característica no confiable del protocolo IP, llevó posteriormente al desarrollo de mecanismos de alerta y mecanismos de prueba, adicionales al protocolo propiamente dicho, pero que completan su funcionalidad.

La función principal del protocolo es el encaminamiento de paquetes. Esta característica propia llevó a considerar la necesidad de un esquema de **direccionamiento global** para la identificación de los protagonistas de la comunicación. A medida que la red de Internet fue creciendo, esta elección derivó en un problema de configuración. Al principio, la definición de una dirección de red para un dispositivo se realizaba de forma manual, aunque con el correr de los tiempos se impuso la configuración automática.

La configuración de los ***routers*** también resultó en diferentes estrategias, teniendo en cuenta que la carga de información de las Tablas de Enrutamiento sólo puede ser del tipo estático, o carga manual, para redes pequeñas o de tráfico predecible. En general, con el correr de los años, se impuso la carga dinámica, adaptable de manera automática a la situación de una red. Esta derivación, dio lugar a la aparición de nuevos protocolos, denominados protocolos de enrutamiento, diseñados para la comunicación de aquella información relevante que permite a los *routers* acomodar sus Tablas de Enrutamiento a diferentes estados de congestión de la red.

La forma de direccionar los paquetes a destino final, conocida como enrutamiento **Red Destino-Próximo Salto,** impuso un formato a la información cargada en las tablas, así como a la manera de consultar dicha información. De este modo, la consulta a la Tabla de Encaminamiento se realiza tomando como información de entrada sólo la dirección de la red a la que pertenece el destino final del paquete, en tanto que la tabla ofrece como resultado de la consulta, la dirección del siguiente *router* en el camino a ese destino. Obviamente, para que este esquema de enrutamiento funcione correctamente, las tablas en los diferentes *routers* deben cargarse con información coordinada y consistente.

Esta filosofía de encaminamiento puede derivar en ciertos problemas de circulación de paquetes, ante la presencia de errores en las tablas. Por ejemplo, un error que generase un lazo en las propias rutas, con dos *routers* apuntándose entre sí como dirección de próximo salto, provocaría la circulación infinita de paquetes entre ambos *routers* intervinientes en ese lazo. La posibilidad de la existencia de este tipo de situaciones, motivó el agregado de información para que los paquetes tuvieran un **Tiempo de Vida** (TTL, Time to Live), útil como herramienta de descarte.

También, la necesidad de atravesar distintas redes, llevó a considerar que los paquetes puedan encontrar diferentes Unidades de Transferencia Máxima (MTU, Maximum Transfer Units), definiéndose este parámetro como la carga de datos máxima soportable por el nivel inmediatamente por debajo de IP. Si la **MTU** de la red sobre la que se va a re-enviar un paquete, es menor que la MTU de la red de procedencia, se impone la necesidad de un **mecanismo de fragmentación**. El protocolo IP permite que un paquete se fragmente en cualquier punto a lo largo de su camino, ya sea en origen o toda vez que un *router* deba entregarlo sobre un enlace de MTU menor. El procedimiento de re-ensamble, en cambio, sólo se realiza en el destino final. Esta última restricción se debe a una decisión de diseño referida a evitar la carga extra, ya sea de procesamiento o de asignación de recursos, en los dispositivos de encaminamiento intermedios.

Por los motivos expuestos, el conjunto de funcionalidades mencionado se repartió, en realidad, entre tres protocolos. El Protocolo de Internet (IP, Internet Protocol) se definió en la RFC 791 de 1981. IP reúne las funcionalidades de enrutamiento, tiempo de vida, direccionamiento, fragmentación y mecanismos de pruebas. El Protocolo de Mensajes de Control de Internet (ICMP, Internet Control Message Potocol), definido en la RFC 792 de 1981, provee los mecanismos de alerta mencionados. Por su parte, el Protocolo de Resolución de Direcciones (ARP, Address Resolution Protocol), definido en la RFC 826 de 1982, ofrece el mecanismo de relación entre direcciones locales y direcciones globales para redes de tipo difusión.

9.2 Protocolo IP

Teniendo en cuenta las funcionalidades mencionadas en el apartado previo, es más sencillo entender los distintos campos que integran el encabezado del Protocolo IP, presentados en la Fig. 9.1.

El campo **Versión** ($4\ bits$) presenta el número de versión del protocolo. IP versión 4 es la versión actual de IP, con probable migración a IPv6.

La **Longitud de la Cabecera IP** (IHL, IP Header Length) ($4\ bits$) es un número que expresa en decimal la cantidad de palabras de $32\ bits$ que conforman el encabezado. Este campo puede presentar un valor mínimo 5, significando una longitud de cabecera de 5 palabras de $32\ bits$, o su equivalente de $20\ bytes$. Se trata del tamaño mínimo para la cabecera fija, que es obligatoria. El valor máximo de este campo es 15, interpretable como 15 palabras de

32 $bits$, es decir 480 $bits$ o 60 $bytes$. El campo se debe a la existencia de opciones. El límite establece una cantidad máxima de opciones, que no puede superar los 40 $bytes$.

El campo **Tipo de Servicio** (**TOS,** Type of Service) (8 $bits$) sirve para indicar las necesidades particulares de Calidad de Servicio (QoS, Quality of Service) requeridas a la red por una aplicación particular. En este campo se intentó definir el modo en que los *routers* debían ordenar los paquetes en sus propias colas de re-envío. Para ello, se desarrollaron tres criterios, cada uno con su propia bandera de señalización, tal como se observa en la Fig. 9.2. Se podía requerir una ruta para minimizar el retardo poniendo en alto el bit D (*Delay*), maximizar el ancho de banda levantando el bit T (*Throughput*), o maximizar la confiabilidad de los caminos atravesados con el bit R (*Reliability*). A su vez, los tres bits al inicio del campo, marcan un nivel de precedencia para el paquete. En la práctica, el campo TOS nunca se utilizó para tomar decisiones de enrutamiento.

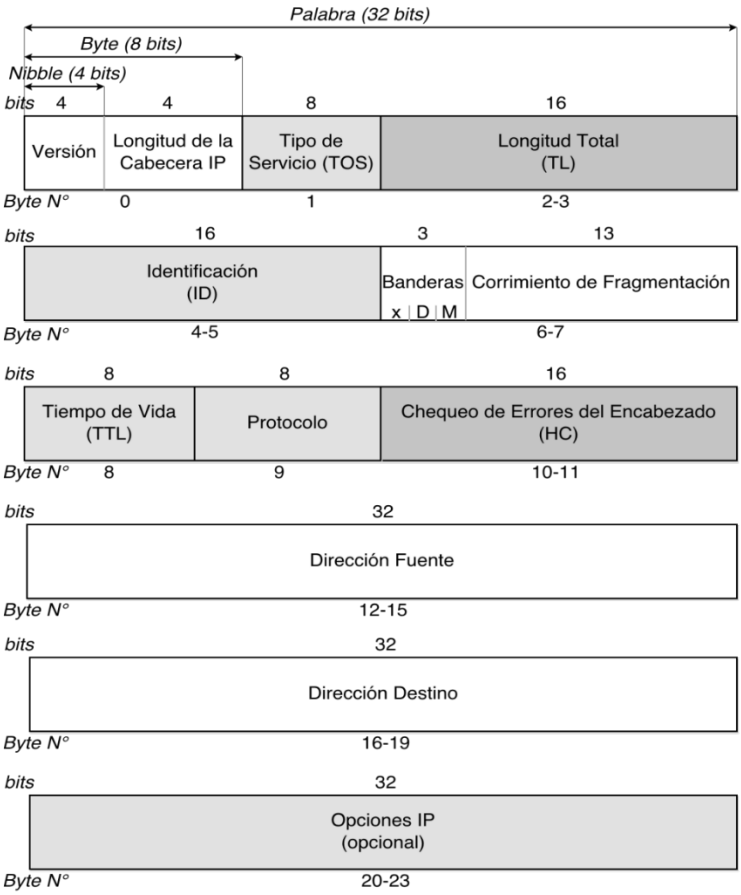


Figura 9.1 - Encabezado del Protocolo IP.

En la actualidad, la definición de servicios diferenciados ha cambiado esta manera tan precaria de dar prioridad al traslado de paquetes. Justamente con este nombre, Servicio Diferenciado (*DiffServ*), se conocen actualmente los

primero 6 $bits$ del campo TOS. Los últimos dos bits se denominan Notificación de Congestión Explícita (ECN, Explicit Congestion Notification), utilizándose para colaborar en situaciones de congestión. El código con que se llena el campo *DiffServ* indica a un *router* el tratamiento que debería dar al paquete en el proceso de re-envío. Existen varios patrones definidos, refiriéndose a situaciones que van desde tratamiento de rutina hasta re-envío expeditivo, sobre una base por salto. Por su parte, el par de bits ECN sirve para marcar un paquete que pasa por un *router* cuyos puertos se encuentren excesivamente cargados de tráfico. La consecuencia de esta marca sobre el datagrama por parte de un *router* se genera en destino, al ser apreciada por protocolos de nivel superior, tales como TCP, que inician un mecanismo de aviso al extremo origen, para que baje la velocidad y así evitar la pérdida de paquetes en el *router* congestionado.

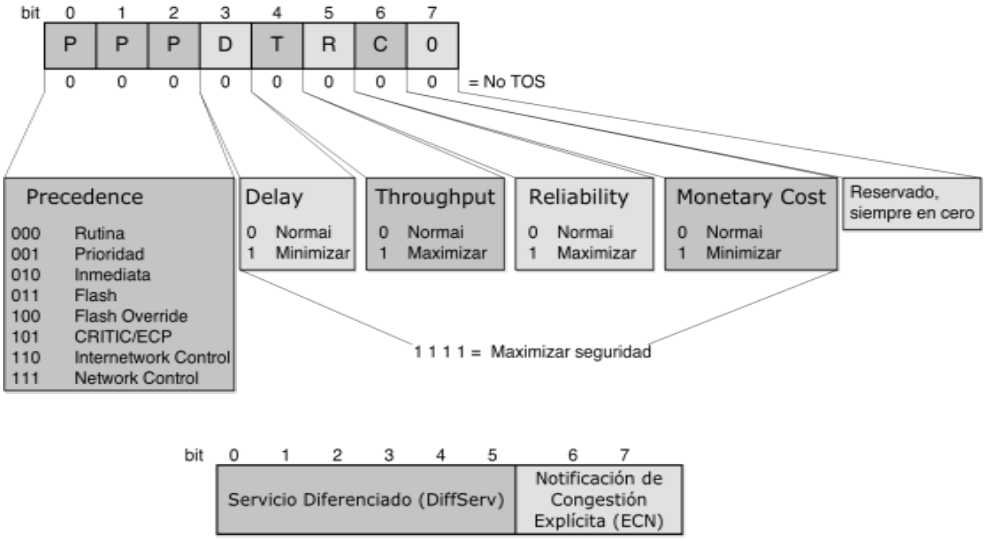


Figura 9.2- Campo de Tipo de Servicio.

El campo **Longitud Total** (**TL,** Total Length) (16 $bits$) es un número cuyo significado decimal es la cantidad de bytes total del paquete: encabezado y datos. Al tratarse de un campo de 16 $bits$, la longitud total máxima resultante es de $2^{16} - 1 = 65535\ bytes$. Una de las razones de la presencia de este campo es la existencia de una unidad mínima de transferencia en redes LAN CSMA/CD, que obliga a rellenar a nivel MAC las tramas de tamaño inferior a este valor. Es pertinente aclarar que, en caso de fragmentación, el valor que carga este campo es el correspondiente a la longitud total del fragmento.

La **Identificación ID** (16 $bits$) es número de carácter aleatorio que, junto con la dirección fuente, identifica unívocamente el paquete. Generalmente, para evitar confusiones, se incrementa en uno por cada paquete que se envía por una interfaz. Es un campo fundamental en el proceso de re-ensamblado de los

paquetes fragmentados, ya que todos los fragmentos, obligatoriamente, tienen en común el mismo ID.

El campo **Banderas** (**Flags)** (3 $bits$) se trata de una secuencia de 3 bits. El primero no tiene uso y se ajusta en "0". El segundo se utiliza para indicar a los *routers* si se permite o no fragmentar un paquete determinado. Se trata del bit DF (Don't Fragment) que, cuando se ajusta en "1" significa No Fragmentar. Si un paquete posee este bit en alto y debe re-enviarse sobre un enlace de menor MTU, se lo descartará, enviando al origen un mensaje de error ICMP para informar esta situación. El tercer bit es particularmente útil en el proceso de re-ensamble de un datagrama fragmentado. Se lo conoce como bit MF (More Fragments) y se usa para indicar si un paquete es el último de una secuencia fragmentada o se trata de un fragmento intermedio. Cuando el bit MF se encuentra en "0" significa que es el último fragmento de un paquete.

El campo **Corrimiento de Fragmentación** (**FO,** Fragment Offset) (13 $bits$) es un número cuyo significado decimal representa la posición del fragmento actual, medida en unidades de 8 $bytes$, dentro del datagrama original. La razón de esta interpretación tan extraña es más simple de lo que parece. Con 13 $bits$, como posee este campo, se pueden contar hasta $2^{13} - 1 = 8191\ bytes$. Este valor, interpretado como un número de 8 $bytes$ logra cubrir la longitud mayor posible del datagrama, ya que $2^{13} x\ 2^8 - 1 = 2^{16} - 1 = 65535\ bytes$.

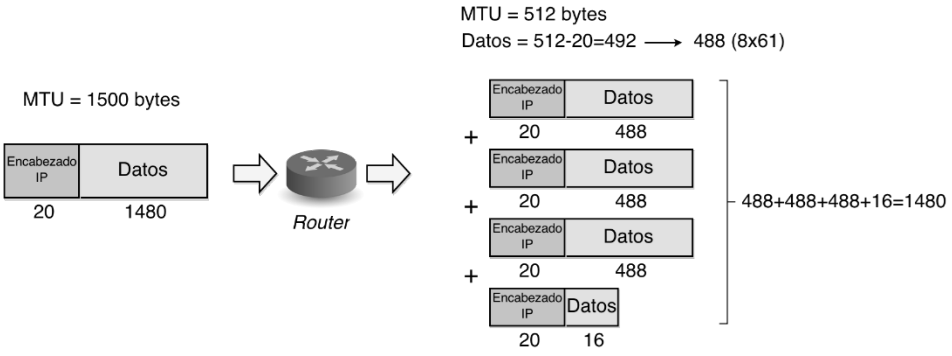


Figura 9.3 - Ejemplo de Fragmentación.

A modo de ejemplo, en la Fig. 9.3, se presenta el caso en que un paquete debe atravesar un *router* que conecta una red LAN, cuya MTU es de 1500 $bytes$ (1480 $bytes$ de datos + 20 $bytes$ de encabezado IP), con una red de menor MTU, apenas 512 $bytes$. Al volcar el paquete original sobre la interfaz de re-envío, se debe descontar de la nueva MTU los bytes correspondientes a la cabecera *IP*. De este modo se calcula el lugar que resta para los datos, en este caso 512 $bytes$ – 20 $bytes$ = 492 $bytes$. Se debe verificar que el número resultante sea múltiplo de 8 para poder usar correctamente el campo de FO. Si no lo es, se debe redondear el valor al número entero más cercano que lo sea, en este caso 488. Esto significa que cada paquete será capaz de cargar 488 bytes de

datos, resultando los datos originales repartidos en 4 fragmentos, según muestra la figura. Los campos relevantes de los fragmentos resultantes se presentan en la Tabla 9.1.

Tabla 9.1 – Ejemplo de Fragmentación.

Fragmento N°	**TL**	**ID**	**MF**	**FO**
1	508	XXXXXXXXXX	1	0
2	508	XXXXXXXXXX	1	61
3	508	XXXXXXXXXX	1	122
4	36	XXXXXXXXXX	0	183

El campo **Tiempo de Vida** (**TTL,** Time To Live) (8 $bits$) originalmente fue pensado como una cantidad en segundos que representaba el tiempo de vida de un paquete, debiendo descontarse en cada *router*, considerando el caso que, cuando llegara a "0" , el paquete fuera descartado. En la práctica se usa para medir la cantidad de saltos o *routers* que un paquete puede atravesar. Se inicializa en el origen a un valor *default* establecido por el Sistema Operativo. El valor 64 es el actual recomendado por el IANA. Este valor, cargado por el origen del paquete, se descuenta en una unidad en cada *router* atravesado. Al llegar a "0" , se descarta el paquete y se envía un mensaje ICMP de error al origen del mismo. Se usa para enfrentar situaciones de error en las Tablas de Enrutamiento, cuando existen lazos de enrutamiento. De no ser por este campo, estos lazos se asociarían a datagramas atrapados sin posibilidad de ser descartados.

Un campo **Protocolo** (8 $bits$) indica el tipo de datos que el paquete encapsula. Por ejemplo, para TCP el valor establecido es 6, en tanto que para UDP es 17 y para ICMP se asocia el valor 1. Sirve para demultiplexado en la recepción.

El campo **Chequeo de Errores del Encabezado** (**HC,** Header Checksum) (16 $bits$) cubre sólo el encabezado y se usa para proteger el paquete en el procesamiento en los *routers*. Tomando el encabezado como un conjunto de palabras de 16 $bits$, el transmisor coloca en este campo el complemento a 1 de la suma módulo 2 (EXOR) de las palabras resultantes. El receptor del paquete, ya sea el destino final o *routers* intermedios, calcula el complemento a 1 de la suma de todos los campos. Si no hubo errores, la cuenta debe dar todos "0". Si hubo errores se descarta el paquete sin ningún aviso. Es preciso destacar que, al modificarse el campo TTL en el procesamiento de un *router*, es necesario re-calcular el HC antes del re-envío.

9.2.1 Direcciones IP

La estructura original de direcciones IP fue desarrollada a comienzos de los años 80, siendo utilizada desde entonces por la comunidad de Internet. A pesar de tratarse de un espacio finito, de tipo jerárquico, el esquema ha resistido el

crecimiento extraordinario de la propia red global, manteniéndose vigente por varias décadas.

Por tratarse de números de 32 $bits$, representarían un conjunto de unos 4.500 millones de direcciones, útiles para ser asignadas a dispositivos de acceso a la red Internet. En realidad, de todo este espacio, un conjunto menor fue reservado para comunicación de grupos, posibilidad conocida *multicasting IP*, y otro conjunto, de la misma extensión, fue separado para pruebas. Parecen números descomunales pero no han resultado suficientes para la expansión desarrollada por la Internet actual.

Las direcciones IP son números asignados a interfaces de computadoras. Los usuarios de Internet utilizan nombres para acceder a los diversos sitios de la gran red, tales como páginas web o correo electrónico, puesto que les resulta más sencillo recordar nombres que recordar números, sobre todo si estos son tan largos como las direcciones IP. Un servicio especial, transparente a los usuarios traduce esos nombres a direcciones IP. Los mensajes de cualquier comunicación, se encapsulan en paquetes IP que llevan en su encabezado las identificaciones unívocas o direcciones IP de las máquinas fuente y destino que resulten protagonistas de una comunicación. Dichas direcciones también se utilizan para facilitar el encaminamiento de paquetes IP.

Todos los dispositivos conectados a la red tendrán al menos una dirección IP por cada interfaz de acceso, tal como se presenta en la Fig. 9.4.

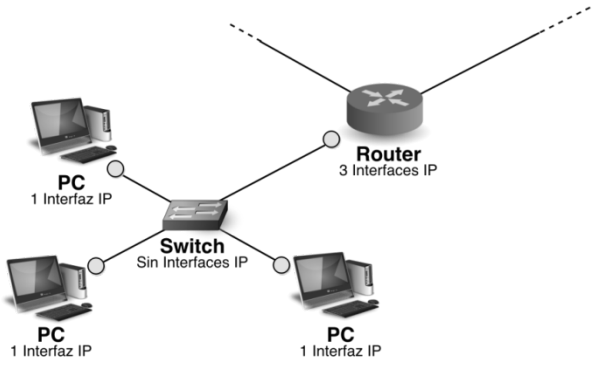


Figura 9.4 - Interfaces y Direcciones IP.

Al representar interfaces de red y colaborar con la funcionalidad de enrutamiento, las dirección IP de cada interfaz se relaciona específicamente con la red sobre la cual está conectada. De este modo, si un dispositivo, configurado con una dirección IP determinada, se mueve a otra red, se deberá reconfigurar con la dirección apropiada.

Generalmente, las computadoras personales tendrán una única dirección IP mientras que otros dispositivos, tales como *routers*, poseerán más de una dirección, ya que sus interfaces se encuentran sobre diferentes redes. Por su parte, un puente o un *switch* no poseen direcciones IP por que conectan dispositivos a

nivel de enlace y, por lo tanto, se caracterizan por sus direcciones MAC *Ethernet*. Esta diferencia en los esquemas específicos de direccionamiento, hace necesaria la existencia de alguna funcionalidad que mapee direcciones entre ambos niveles. Esa es la tarea que realiza el protocolo ARP.

La dirección MAC es necesaria para intercambiar datos entre dispositivos de una misma red, en tanto que la dirección IP sirve para el transporte a través de diferentes redes, ya que involucra el destino final de un paquete. Algunas de las cuestiones más importantes en cuanto a la conexión de un dispositivo a Internet, se relacionan con la manera en que se asignan estas direcciones. En una red privada es la propia administración quien controla la asignación de todos los dispositivos. En una red pública, accesible desde cualquier dispositivo conectado a Internet, se precisa un mecanismo que asegure que diferentes organizaciones posean distintas direcciones, ya que la asignación debe ser unívoca y no pueden existir solapamientos. Para lograr este último objetivo, se creó un organismo de registro y administración de direcciones IP. En la actualidad, han surgido técnicas que permiten que una red privada se administre libremente por detrás de un *router* con capacidad de Traducción de Direcciones de Red (NAT, Network Address Translation), pero siempre la conexión a Internet exige un requerimiento a organismos o representantes con autoridad en el reparto de direcciones.

Otra cuestión importante se relaciona con la configuración de la dirección IP. Al principio este trabajo se realizaba de forma manual, significando una gran carga sobre los hombros del administrador a medida que las redes crecían de tamaño. Este tipo de asignación era estática, en el sentido de que no variaba a menos que el administrador manualmente lo hiciera. Con el tiempo, surgieron protocolos, tales como el Protocolo de Configuración Dinámica (DHCP, Dinamic Host Configuration Protocol), que permiten la configuración dinámica de direcciones, facilitando enormemente la tarea del administrador.

9.2.2 Formato y Clases de Direcciones IP

El formato de las direcciones IPv4 se basa en un número de $32\ bits$. Dada la magnitud de un número de estas características, dichas direcciones se escriben como un conjunto de 4 números de $8\ bits$, y se leen en formato decimal separados por puntos, para mayor facilidad de interpretación. De este modo, una dirección IPv4 es un conjunto de 4 números variables entre 0 y 255. Por ejemplo, una dirección típica es 192.168.0.1.

Como se ha explicado anteriormente, un formato de direcciones de $32\ bits$ provee un espacio de 2^{32} direcciones, aunque no todas están disponibles para ser asignadas. En su momento parecía una cantidad más que suficiente, pero la expansión de Internet pero la propia estructura interna elegida para las direcciones, generó una serie de dificultades conocidas como problema de vaciamiento de direcciones IP. En respuesta a este problema, surgieron nuevas técnicas de compensación para evitar cambiar el esquema de direcciones original IPv4. NAT y el Enrutamiento Interdominio sin Clases (CIDR, Classless Inter-Domain Routing) fueron algunas de las soluciones temporales, pero finalmente

se pensó en el diseño de un nuevo protocolo en reemplazo de IPv4, dando nacimiento a una nueva versión: IPv6.

El formato original de direcciones elegido por los diseñadores del protocolo IPv4 es del tipo jerárquico de clases, por ello se denominó *classful*. La idea detrás de este diseño tenía como objetivo facilitar la tarea de procesamiento en la decisión de enrutamiento que pesa sobre los *routers*. Por este motivo, el número de 32 $bits$ se consideró dividido en dos partes: un Identificador de Red (NetID, Network ID) y un Identificador de Dispositivo (HostID) dentro de esa red. Este formato tuvo su origen en las direcciones de la red más conocida en aquella época: la red de telefonía. Si se piensa en cualquier número de TE fijo, distinguiremos un primer conjunto de dígitos que identifican un país (+54 en Argentina) y luego otro que representa una localidad (0223 para Mar del Plata). Finalmente, el número de TE en sí se refiere a la localización del aparato dentro de la red local. De la misma manera, el NetID identifica una red en Internet, mientras que el HostID se refiere al dispositivo dentro de esa red.

En la dirección de ejemplo de la Fig. 9.5, se han asignado los primeros 8 $bits$ al NetID y los últimos 24 $bits$ al HostID. Esta separación facilita la decisión de enrutamiento, ya que los *routers* sólo utilizan el NetID en el procesamiento de re-envío de los paquetes. Una consecuencia directa de la elección de este formato, es que el movimiento de máquinas entre distintas redes implica un cambio de dirección IP.

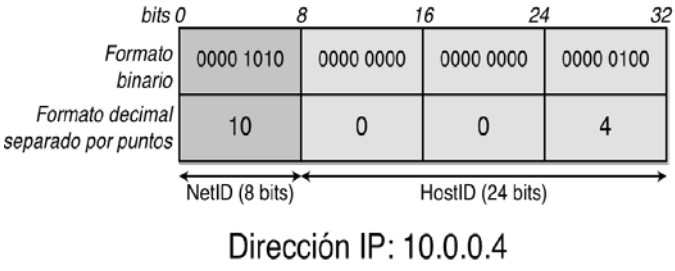


Figura 9.5 – NetID y HostID de Direcciones IP.

En este esquema original, la separación entre identificadores no es fija, sino que se establece en el límite de algún número entero de bytes, según el concepto de clases de direcciones IP. Los diseñadores de IPv4, dividieron el espacio en tres clases, diferenciadas por la cantidad de bytes asignados al NetID. Denominaron **clase A**, a aquellas direcciones cuyo NetID se fijó en 1 $byte$. Para la **clase B** destinaron 2 $bytes$ para el NetID y definieron una tercera clase, la **clase C**, con los 3 $bytes$ iniciales de la dirección IP con ese propósito.

Además, idearon un esquema en el que los primeros bits de cada clase sirvieran para que los *routers* pudiesen distinguir rápidamente a cuál clase pertenece la dirección IP destino del paquete procesado. De este modo, pueden

separar la información de NetID que necesitan para el enrutamiento sólo por inspección de los primeros bits de la dirección IP.

Como consecuencia de esta elección de diseño, más tarde surgió el concepto de **máscara de red**, un número de 32 $bits$ desarrollado como un conjunto de bits en "1" en coincidencia con la longitud del NetID, seguido por un conjunto de bits en "0" en correspondencia con la parte de la dirección destinada al HostID. Si se realiza una AND lógica entre la máscara de red y la dirección IP destino, se puede separar el NetID que se precisa para realizar la decisión de Enrutamiento. Por ejemplo, para el caso de la Fig. 9.5, la máscara de red sería 255.0.0.0.

La idea detrás de la jerarquía de clases surgió también del reconocimiento de la existencia futura de diversos tipos de organizaciones, con diferentes necesidades de conexión. En su concepción original, el protocolo previó además la existencia de algún tipo de autoridad central encargada de la distribución de direcciones IP. Para facilitar la tarea de esta autoridad, se eligió dividir el espacio total de direcciones en porciones de diferente tamaño, de tal manera que fuera más fácil adaptar la entrega a las estructuras de distintas organizaciones. Con estas premisas se desarrolló un esquema de cinco clases que cubrieron el total del espacio de direcciones IP, tal como se observa en la Fig. 9.6.

La **clase A**, con 1 $byte$ reservado para el NetID, tiene una máscara de red que se puede escribir como 255.0.0.0 y se la puede referir, considerando un léxico más moderno, como una red /8. Esta clase posee la mitad del espacio de direcciones y reserva un conjunto de 24 $bits$ para poder identificar dispositivos dentro de una misma red. Por este motivo, se la consideró para ser entregada a aquellas pocas organizaciones con una cantidad enorme de estaciones de trabajo.

Para la **clase B** se reservó un espacio equivalente a un cuarto del total, con una máscara de red de 16 $bits$, expresada como 255.255.0.0 ó /16, que resulta de la misma longitud que el HostID. Esta clase se ideó para su utilización por parte de muchas organizaciones con una cantidad de hosts considerable.

Por su parte, la **clase C** ocupa un octavo del espacio total y fue ideada para organizaciones con pocas estaciones de trabajo. Su máscara de red es de 24 $bits$, 255.255.255.0 ó /24, quedando escasos 8 $bits$ para identificación de hosts.

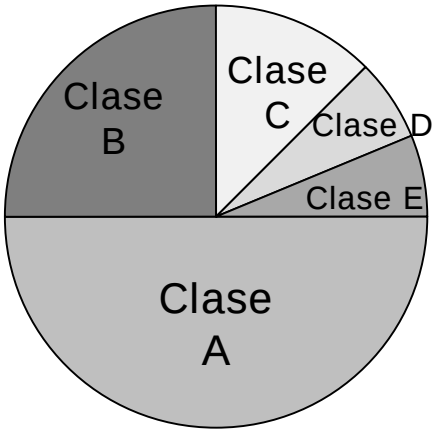


Figura 9.6 – Espacio Total de Clases de Direcciones IP.

Entre las tres clases mencionadas se repartieron las 7/8 partes del espacio total, considerándoselas exclusivamente para direccionamiento *unicast*. Se definieron dos clases más, una para considerar la comunicación tipo *multicast*, conocida como **clase D**, y otra para efectuar experimentos en la red global, denominada **clase E**.

La Fig. 9.7 presenta una visión más detallada de la división en las clases mencionadas. La **clase A** comienza con el primer bit en "0", reservándose de este modo la mitad del espacio de direcciones. Cuando el primer bit es "1" y el segundo bit es "0", se trata de una dirección **clase B**. Si los dos primeros bits son "1" y el tercero es "0", se trata de una dirección **clase C**. Por su parte, la **clase D** comienza con el patrón "1110" y la ***clase E*** con "1111". Observar que con esta asignación, sólo por inspección de los primeros bits de una dirección IP, resulta muy sencillo para un *router* interpretar la máscara de red apropiada para aplicar.

En la Tabla 9.2 se presentan las máscaras, rangos, cantidad de redes y de *hosts* que se corresponden con cada clase. La información de la Tabla es fácilmente deducible con lo explicado hasta ahora, aunque en la última columna hay un detalle en la cantidad de hosts/red que parece llamativo. En cada clase *unicast*, existen dos direcciones no utilizables para la configuración de un dispositivo. Este detalle se entenderá más claramente cuando se presenten los grupos de direcciones especiales.

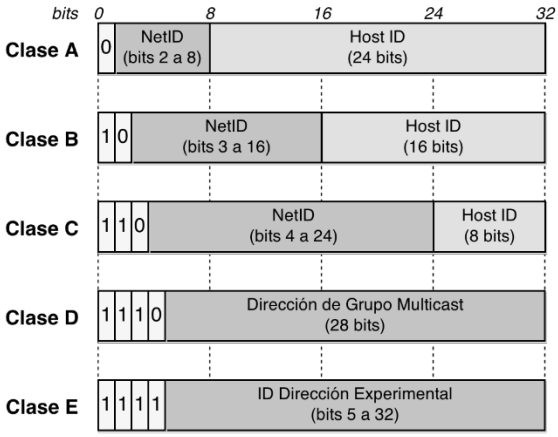


Figura 9.7 – Clases de Direcciones IP.

Tabla 8.2 - Clases de Direcciones IP.

Clase	Máscara	Rango de Direcciones	Cantidad de Redes	Cantidad de hosts/red
Clase A	/8 ó 255.0.0.0	1.0.0.0 a 127.255.255.255	2^7	2^{24}-2
Clase B	/16 ó 255.255.0.0	128.0.0.0 a 191.255.255.255	2^{14}	2^{16}-2
Clase C	/24 ó 255.255.255.0	192.0.0.0 a 223.255.255.255	2^{21}	2^8-2
Clase D	-	224.0.0.0 a 239.255.255.255	-	-
Clase E	-	240.0.0.0 a 255.255.255.255	-	-

Por inspección de la Tabla, también se hace apreciable uno de los principales problemas del direccionamiento por clases. Si se supone que una organización precisa direcciones IP para 1500 dispositivos, surge una pregunta clave: ¿Qué tipo de clase le otorgaría la autoridad de distribución? Otorgarle una dirección clase A o una clase B, resultaría excesivo. Una clase C no alcanzaría, aunque si se le asignasen al menos 6 numeraciones o redes clase C, con capacidad para 1524 dispositivos, cubriría la necesidad de la organización y sobrarían 24 direcciones. Esta última asignación parece la más apropiada, pero no se debe olvidar que el enrutamiento se decide en base a direcciones de red, así es que la Tabla de Enrutamiento alojada en el *router* que apuntara a dicha organización, precisaría guardar información sobre estos 6 prefijos de red. Cuanto mayor es la cantidad de redes destino que se agregan a una Tabla de Enrutamiento, mayor es la cantidad de sus líneas y aumenta en proporción el tiempo de recorrido de la misma toda vez que haya que realizar un procesamiento de re-envío de paquetes. En contraposición, si la autoridad de distribución asignara a la organización una

dirección clase B, se lograría salvar el problema de extensión de la Tabla de Enrutamiento, pero la asignación sería ineficiente puesto que sobrarían 64034 direcciones.

Este ejemplo demuestra uno de los problemas presentados por el esquema de direcciones original de IPv4, que luego devino en el vaciamiento de dichas direcciones.

9.2.3 Direcciones IP especiales y reservadas

Existen direcciones IP fijas con significado especial:

- **Todos** "**0**" ó "**0.0.0.0**": es la dirección especial con significado *este dispositivo*, que suele verse como dirección fuente en mensajes de protocolos de configuración automática de direcciones IP. Cuando una máquina en proceso de arranque (*booteo*) no tiene asignada una dirección de red, si soporta un protocolo de configuración automática tal como DHCP, comenzará a enviar mensajes en búsqueda de un servidor que le permita obtener dicha dirección. Estos mensajes DHCP van encapsulados en IP, pero como el dispositivo aún no tiene asignada una dirección, tiene permitido colocar la dirección "0.0.0.0" en el campo dirección fuente. Esta dirección de arranque de una máquina con configuración automática, puede tener una variante: el espacio de NetID cargado con todos "0" y un número particular en el espacio HostID, significando un host particular en *esta red*. Por ejemplo, 0.0.0.120 será el host 120 de determinada red *clase C*.
  Nunca se podrá escribir esta dirección especial como dirección destino. Sólo se podrá utilizar como dirección fuente en los casos mencionados.

- **Todos** "**1**" ó "**255.255.255.255**" : es la dirección especial con significado *todos*, que suele verse como dirección destino en mensajes que precisan ser comunicados a todos los dispositivos conectados a una red. Se denomina dirección de *broadcast*. Cualquier *router* que procesara un paquete *IP* con esta dirección destino, interpretará que no debe re-enviarlo, ya que se trata de un *broadcast* para la red local. Como en el caso previo, puede presentarse una variante. Por ejemplo, puede escribirse el espacio de NetID con un número particular y en el espacio HostID colocar todos "1", significando que se trata de un mensaje destinado a *todos los hosts* de la red referenciada. Por ejemplo, la dirección 10.255.255.255 se refiere a una comunicación dirigida a todos los *hosts* de la red 10. A diferencia del caso anterior, un *router* que procese un paquete con este tipo de dirección destino, lo re-enviará a la red que corresponda. Nunca se verá este tipo de direcciones como dirección fuente, pues sólo pueden usarse como dirección destino.
  La existencia de estas direcciones especiales es la explicación de la resta de 2 a la cantidad de dispositivos por red de la Tabla 9.2.

Supongamos una dirección clase C, con 3 *bytes* designados al NetID, por ejemplo 192.168.1. Dentro de esta red, la dirección todos "0" en el HostID, es decir 192.168.1.0, se refiere a la propia red, como una forma de referenciarla. Esta referencia se podría observar en una línea de la Tabla de Enrutamiento o en un mensaje de arranque para cualquier máquina que solicite a un servidor un número dentro de esa red. De este modo, la dirección todos "0" en el HostID no es utilizable para configuración *unicast*.

Por otra parte, considerando en el mismo ejemplo la dirección todos "1" en el espacio de HostID, es decir 192.168.1.255, se utiliza como dirección destino, para enviar mensajes a todos los hosts dentro de esa red. Es decir que las numeraciones todos "1" en el HostID poseen un significado especial y no pueden asignarse a ningún host en particular.

De ahí que, de todo el espacio de direcciones de una red particular disponible para numerar dispositivos, dos no pueden usarse, un HostID por que se utiliza para referenciar la propia red y el otro porque se utiliza para dirigir un mensaje por *broadcast*.

Además de las direcciones especiales, existen rangos de direcciones apartados para usos particulares:

- **Dirección de Loopback**: se trata de cualquier dirección *unicast* dentro del rango clase A 127.0.0.0 - 127.255.255.255. Generalmente se asigna la dirección 127.0.0.1 . Cualquier Sistema Operativo que posea capacidad de comunicación en red por TCP/IP, incluye una interfaz de red virtual que se suele referenciar como *lo*, por *loopback*. Esta interfaz permite que aplicaciones para red de tipo cliente se comuniquen con sus pares de tipo servidor dentro de una misma máquina, sin necesidad de bajar los mensajes a la red física pero recorriendo toda la pila TCP/IP. Los mensajes que se pasan a la interfaz de *loopback* no se entregan al controlador de red sino que recorren la pila en sentido inverso como si provinieran de la red. Es una herramienta muy útil para prueba durante el proceso de programación de aplicaciones de red.

- **Direcciones Reservadas**: se ha comentado previamente sobre la asignación de direcciones en redes privadas, en contraposición con las redes públicas. Existen bloques separados de direcciones que se destinan a la configuración de redes privadas. Esto significa que se pueden configurar máquinas dentro de un dominio privado con estas direcciones sin necesidad de obtener un permiso de la autoridad de asignación. La ventaja que ofrece esta posibilidad es la libertad otorgada al administrador en la asignación, pero la desventaja se presenta cuando dicha red se debe conectar a Internet. En ese caso se precisa al menos una dirección de dominio público, no reservado, otorgado por la autoridad de asignación, y un *router* NAT, con capacidad de traducción entre las direcciones de dominio privado y las de dominio público. Por este

motivo, a estos bloques reservados se los suele denominar también como *direcciones no ruteables*. Nunca se podrán observar este tipo de direcciones en Internet, ni como dirección fuente ni como dirección destino.

Entre los bloques separados, se encuentran la red clase A 10 (10.0.0.0 – 10.255.255.255), los bloques clase B 169.254 (169.254.0.0 – 169.254.255.255) y 172.16 (172.16.0.0 – 172.16.255.255) y el conjunto de 256 redes consecutivas clase C 192.168 (192.168.0.0 – 192.168.255.255). Un administrador puede usar cualquiera de estos rangos de direcciones para configurar los dispositivos internos, para que la comunicación entre ellos sea posible. Si la red se desea conectar a Internet, entonces una autoridad de administración le otorgará al administrador al menos una dirección de carácter público, o sea que no pertenezca a estos rangos. La configuración de un *router* con capacidad NAT permitirá la comunicación global.

9.2.4 Multicast IP

Se denomina *multicast* al tipo de comunicación en grupo, en la que un dispositivo envía un mensaje a un conjunto de receptores. Se trata de una funcionalidad relativamente fácil de realizar a nivel de enlace, pero que encuentra cierto grado de dificultad cuando se pretende desarrollar de manera global, involucrando protagonistas en diferentes redes. Entonces, es necesario cierto tipo de funcionalidad adicional relacionada con el procesamiento de paquetes en los *routers*, con la administración de grupos de comunicación y con el mapeo de direcciones de hardware a direcciones de red.

Para poder realizar la comunicación *multicast*, se separa un esquema de direccionamiento especial que sirve para identificar al grupo de dispositivos involucrados en la comunicación. En IPv4 se destinó el conjunto de direcciones clase D del esquema original con este propósito. Las direcciones de la clase D comienzan con el patrón "1110", correspondiéndose al rango numérico de 224 a 239 en el primer byte, que comprende el espacio de direcciones IP en el rango 224.0.0.0 a 239.255.255.255. Se trata de direcciones que siempre aparecerán en el campo de dirección destino de los paquetes IP, ya que se corresponden a la dirección de un grupo.

Cualquier dispositivo posee la habilidad para juntarse o abandonar un grupo. Los mensajes que un dispositivo envía a un grupo llevan como dirección fuente la dirección *unicast* con que se lo ha configurado, y como dirección destino la del grupo al cual se ha unido. Cualquier dispositivo perteneciente al grupo recibirá los mensajes. En general, el dispositivo que envía mensajes a un grupo, no sabe cuántos ni cuáles dispositivos están recibiendo sus datagramas.

Los últimos 28 $bits$ de una dirección *multicast* identifican al grupo en particular. IANA define el rango 224.0.0.0 – 224.0.0.255 como reservado para direcciones *multicast* bien conocidas, destinadas para protocolos de enrutamiento, protocolos de descubrimiento de topología de bajo nivel o de

mantenimiento, y reporte de miembros de grupos. Los *routers* con capacidad *multicast* no re-enviarán paquetes con dirección destino dentro de este rango.

Algunos ejemplos de grupos bien conocidos: 224.0.0.0 (reservado, no usado), 224.0.0.1 (todos los dispositivos de una subred), 224.0.0.2 (todos los *routers* de una subred), 224.0.0.4 (todos los *routers* que usan el protocolo de Enrutamiento DVMRP), 224.0.0.5 (todos los *routers* que usan el protocolo de enrutamiento OSPF), 224.0.0.6 (*routers* designados que usan el protocolo de enrutamiento OSPF) y 224.0.0.9 (*routers* designados que usan el protocolo de enrutamiento RIP-2).

En cuanto a la administración de grupos, se requiere un mecanismo que permita a diferentes dispositivos juntarse o abandonar de manera dinámica aquellos grupos con los cuales desean interactuar. Existe un protocolo, denominado Protocolo de Administración de Grupos de Internet (IGMP, Internet Group Management Protocol), que sirve a estos fines. IGMP define el formato de ciertos mensajes para intercambio entre dispositivos y *routers*, que transportan información relativa a los grupos y los miembros de los mismos.

El enrutamiento de datagramas *multicast* debe tener en cuenta que, en este tipo de comunicación grupal, existirá la necesidad de crear múltiples copias de un mismo paquete. Este caso especial de re-envío genera una funcionalidad extra de hardware en los *routers*, precisando de protocolos adicionales, con algoritmos diferentes a los de enrutamiento *unicast*, que deben ser eficientes para evitar la generación de tráfico innecesario que ocuparía mayor ancho de banda, sobrecargando enlaces y los propios dispositivos de enrutamiento.

9.2.5 Opciones IP

El último campo del encabezado IP es el campo de opciones, que puede estar presente o no. Se trata de un campo incluido por los diseñadores para ofrecer alternativas de flexibilidad en el tratamiento de paquetes en los *routers*. Existe una cantidad definida de opciones y, aunque en un paquete puede incluirse más de una opción, la longitud total del campo de opciones debe ser un múltiplo de 32 $bits$ pues el campo HL mide la longitud total del encabezado en palabras de 32 $bits$.

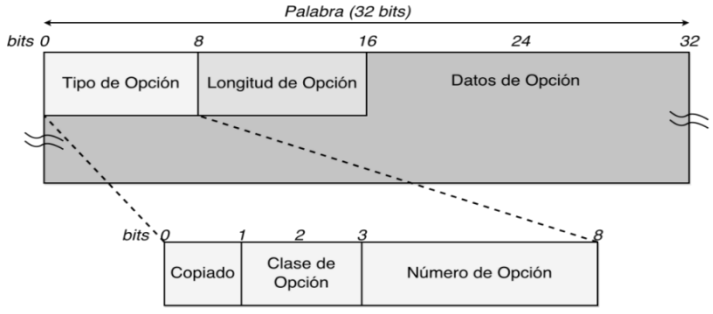


Figura 9.8 - Formato de opciones IP.

Por su parte, cada opción IP tiene su propio formato, estructurado como se presenta en la Fig. 9.8. Se conoce esta disposición como formato TLV (Tipo, Longitud, Valor). Dentro del sub-campo Tipo existe una bandera de copiado para indicar si la opción debe ser o no copiada en los fragmentos, en el caso de tener que re-enviar el datagrama sobre un enlace con MTU menor. Los sub-campos Clase y Número sirven para identificar la opción.

En la Tabla 9.3 se presentan las opciones definidas más conocidas. La mayoría de las opciones se definieron en la RFC original de IPv4. En aquel tiempo, Internet era una red pequeña y no existían los peligros de seguridad conocidos en nuestros días. Estos motivos, sumados a la limitación en la longitud del encabezado, convirtieron en impráctico al campo de opciones.

Observando la especificación de algunas opciones se puede entender porqué prácticamente han quedado obsoletas. Por ejemplo, aquellas opciones que proponen un listado de direcciones IP o sellos de tiempo, ambos conformados por una sucesión de bloques de 4 $bytes$, carecen de lugar para registrar más de nueve valores en el encabezado. En cualquiera de estos casos, teniendo en cuenta la longitud máxima del campo de opciones, a la que se deben restar 2 $bytes$ del sub-campo Tipo y del sub-campo Longitud de la propia opción y, considerando que el campo completo debe ser múltiplo de 32 $bits$, sólo queda lugar para 36 $bytes$ de datos opcionales.

Tabla 8.3 - Opciones IP.

Clase	**Número**	**Longitud (bytes)**	**Nombre**	**Descripción**
0	0	1	Fin de Opciones	Se trata del byte $hx00$, utilizado para indicar fin de lista de opciones.
0	1	1	No Operación	Utilizada para alinear internamente las opciones a múltiplos de 32 $bits$.
0	2	11	Seguridad	Obsoleta, se pensó para marcar paquetes pertenecientes a tráfico del tipo confidencial.
0	3	Variable	Enrutamiento de Fuente Irrestricto (*Loose Source Routing*)	Listado de direcciones IP, escrito en origen. Se corresponden con direcciones de *routers* por los que el paquete debe transitar, aunque se permite el procesamiento en otros intermedios.
0	7	Variable	Registro de Ruta	Esta opción obliga a los *routers* que procesan el datagrama a registrar en el campo de opciones la

Clase	Número	Longitud (bytes)	Nombre	Descripción
				dirección *IP* de la interfaz por donde lo re-envían.
0	9	Variable	Enrutamiento de Fuente Estricto (*Strict Source Routing*)	Listado de direcciones IP, escrito en origen, que se corresponde a las direcciones de *routers* por los que el datagrama con esta opción debe transitar obligatoriamente, con prohibición de procesamiento en otro que no se encuentre en el listado.
2	4	Variable	Sello de Tiempo (*Timestamp*)	Esta opción obliga a los *routers* que procesan el datagrama a escribir un sello de tiempo al momento del re-envío.
2	18	12	*Traceroute*	Para apoyo de la herramienta del mismo nombre.

9.3 Protocolo ARP (Address Resolution Protocol)

La comunicación entre clientes y servidores en Internet se facilita por el uso de nombres en lugar de direcciones IP. El usuario cliente conoce el nombre del sitio al que desea acceder y genera un mensaje de petición a la aplicación con la que desea comunicarse. Este mensaje debe ser encapsulado en un segmento TCP o un datagrama UDP, y luego en un paquete IP. Es decir que ese nombre debe traducirse de alguna manera a una dirección IP destino. Esta tarea la realiza un servicio denominado Servicio de Nombres de Dominio (DNS, Domain Name Service) de modo transparente para el usuario. Una vez obtenida la dirección IP del destino, se puede completar el armado del paquete IP, debiendo éste ser encapsulado en el formato que corresponda al nivel de enlace.

Se presenta entonces la necesidad algún mecanismo que ofrezca la dirección destino correspondiente a este nivel, por ejemplo una dirección MAC. Existe una gran diferencia en el significado de estas direcciones MAC, que sólo posee un alcance local, con respecto a las direcciones IP, cuyo alcance es global.

A modo de ejemplo, supongamos que un alumno, desde su casa, desea consultar material de estudio de Redes de Datos. En el navegador de su PC escribirá el nombre del servidor Web de la Facultad de Ingeniería: http://www.fi.mdp.edu.ar/electronica. Este servidor se encuentra conectado mediante una placa *Ethernet* a la red interna de la Facultad de Ingeniería. Aunque la cátedra le informara a cada alumno la dirección MAC del servidor, este dato no aportaría nada a una comunicación realizada desde fuera de la red de la Facultad de Ingeniería. Cuando el alumno, desde su computadora en casa, realice

el pedido de consulta, el servicio DNS de su propio proveedor le permitirá obtener la dirección IP del servidor Web destino pero, al momento de encapsular el mensaje a nivel de enlace, lo que interesa es dirigirlo al servidor de su propio proveedor de Internet, para que este realice el enrutamiento apropiado del pedido.

Se ha mencionado que el enrutamiento IP es del tipo salto a salto (*hop-by-hop*). Esto significa que, si el vínculo con el ISP funciona correctamente, será el *router* de éste el que conozca mediante la información almacenada en su Tabla de Enrutamiento, la dirección del siguiente *router* en el camino sobre la red desde la casa del alumno a la Facultad de Ingeniería. Es decir que el camino entre fuente y destino es una serie de saltos entre *routers* hasta llegar a destino final. En cada uno de estos saltos, hay una comunicación real a nivel de enlace, en la que importa conocer la dirección a nivel de enlace del *router* del siguiente salto. La dirección MAC del servidor Web sólo importa en la entrega final del mensaje, desde el *router* de entrada a la Facultad de Ingeniería a la ubicación del servidor dentro de esa red.

Dicho esto, se presentará primero un caso simplificado, en el que el alumno desea realizar la consulta pero ahora sentado frente a una máquina dentro de la Facultad, es decir, en la misma red que el servidor.

En cualquiera de los casos mencionados, se precisa un mecanismo que permita la traducción o relación entre direcciones IP y direcciones MAC. Este trabajo lo realiza el Protocolo de Resolución de Direcciones ARP, cuya aplicación es posible sólo en el caso de redes de difusión.

La Fig. 9.9 plantea la situación a resolver cuando ambos actores de la comunicación se encuentran en una misma red LAN, cuya numeración IP es 192.168.0.0. Supongamos que la máquina 1 desea comunicarse con la máquina 3 de la misma red. Ya se ha explicado cómo se puede obtener una dirección IP a partir de un nombre, ahora interesa entender el mecanismo que permite obtener una dirección MAC a partir de esa dirección IP.

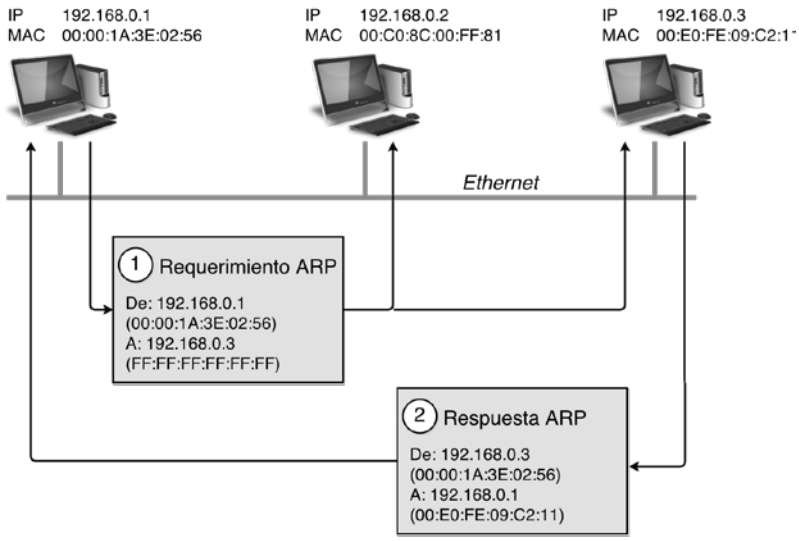


Figura 9.9 - Protocolo ARP.

La máquina 1 puede emitir un mensaje por *broadcast* a nivel de enlace que contenga la dirección IP destino, en este caso 192.168.0.3. Todos los dispositivos de la LAN pueden ver este mensaje, pero sólo la máquina 3 reconocerá su propia dirección en este mensaje y enviará su dirección MAC al interesado, en este caso 192.168.0.1 . Es decir que, a diferencia del requerimiento, la respuesta es por *unicast*. Complementariamente, la máquina 1, al recibir la respuesta, puede guardarla en memoria para futuras comunicaciones con la máquina 3. Esto es lo que se conoce como mecanismo de *caching*. Los datos que se guardan en la memoria se conocen como Cache ARP o Tabla ARP.

El cache ARP tiene la forma de una tabla, con pares de direcciones IP y MAC que se corresponden entre sí. Existe un cache de este tipo por cada interfaz de tipo *Ethernet* que posea cualquier dispositivo. Si una máquina posee más de una placa *Ethernet*, mantendrá una tabla por cada placa de red. Los datos no permanecen en la tabla por tiempo indefinido, sino que tienen un tiempo de vida, vencido el cual, se remueve la información. Esta remoción se hace para evitar que las tablas crezcan de manera indefinida y para dar mayor flexibilidad al esquema de resolución en caso de reemplazo de placas o cambios de direcciones IP. En general, las implementaciones ARP fijan un tiempo de vida por línea de entre 10 y 20 $minutos$.

El caché se carga no sólo en el dispositivo que inicia la resolución, sino también en el que es destinatario de la misma. Algunas implementaciones también cargan el caché con el mensaje de *broadcast*, aunque no sean destinatarios del requerimiento. Así pueden anexar la línea que caracteriza al dispositivo que inicia el requerimiento. Esto es posible debido a la información que cargan los mensajes del protocolo ARP.

Los mensajes ARP poseen un formato como el que se presenta en la Fig. 9.10. El primer campo, el de Tipo de Hardware (16 $bits$) lleva la codificación correspondiente a las redes *Ethernet*. El campo Tipo de Protocolo (16 $bits$) es complementario del anterior y se refiere al nivel de red, siendo el valor $0x800$ correspondiente a IPv4. El campo Longitud de la Dirección de Hardware (8 $bits$) llevará el valor 6 para *Ethernet*, en tanto que el campo Longitud de Dirección de Protocolo (8 $bits$) será 4 para IPv4. El campo Código de Operación (16 $bits$) sirve para distinguir entre preguntas (valor 1) y respuestas (valor 2).

Los dos campos que siguen llevan escrita la Dirección de Hardware (48 $bits$) y la Dirección IP (32 $bits$) del transmisor del mensaje ARP. Los últimos dos campos se refieren al destinatario del mensaje y, en el caso de un requerimiento ARP, la parte de Dirección de Hardware del Target se rellena con "0", ya que se trata de la dirección que se pretende averiguar. En el campo Dirección de Protocolo del Target se carga la dirección IP por la que se pregunta. La respuesta también tiene cuatro direcciones que se escriben en orden inverso al del requerimiento, ya que se intercambian los roles entre transmisor y receptor del mensaje. En total, el mensaje tiene una longitud de 28 $bytes$ y viaja encapsulado en una trama *Ethernet*, cuya longitud mínima es de 64 $bytes$. Por este motivo, los mensajes ARP se rellenan con una serie de "0", hasta alcanzar la longitud mínima requerida.

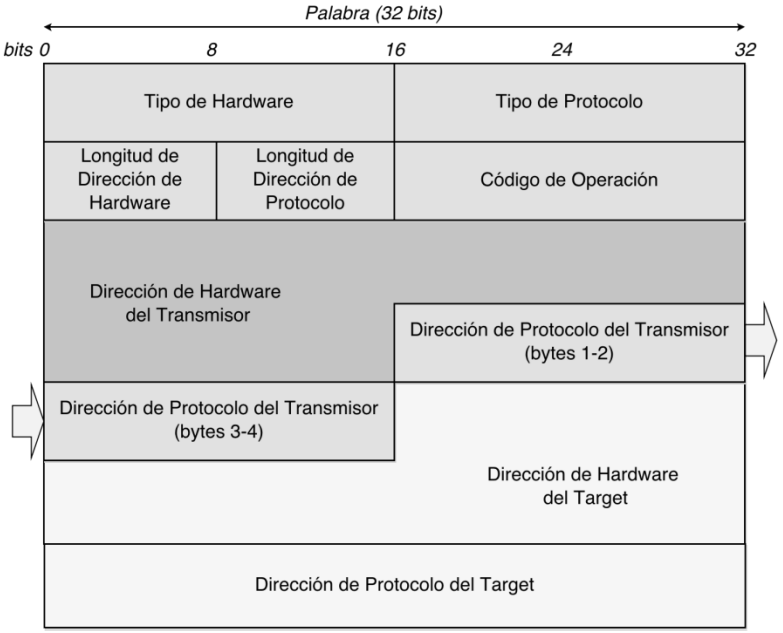


Figura 9.10 - Mensaje ARP.

Existen programas que permiten analizar los paquetes que circulan en una red. Estos programas, conocidos como husmeadores o *sniffers*, colocan a la placa de red en un modo de funcionamiento denominado promiscuo, en el cual la placa levanta todas las tramas que pasan por la red, aún las que no van dirigidas a ella.

A continuación se presenta un ejemplo de mensaje ARP encapsulado en *Ethernet*, levantado desde una red cableada con un *sniffer*:

Ethernet II	Src: 00:1f:d0:aa:bb:cc Dst: ff:ff:ff:ff:ff:ff Type: ARP (0x0806) Trailer: 000000000000000000000000000000000000
Address Resolution Protocol	Hardware type: Ethernet (0x0001) - Protocol type: IP (0x0800) - Hardware size: 6 - Protocol size: 4 - Opcode: request (0x0001) - Sender MAC address: 00:1f:d0:aa:bb:cc - Sender IP address: 192.168.0.1 - Target MAC address: 00:00:00:00:00:00 - Target IP address: 192.168.0.3

En este caso, la máquina con dirección IP $192.168.0.1$ y dirección MAC $00{:}\,1f{:}\,d0{:}\,aa{:}\,bb{:}\,cc$, pregunta a todos los dispositivos de la red LAN cuál es el que posee la dirección IP $192.168.0.3$. En la pregunta, la dirección MAC desconocida se rellena con ceros. En la respuesta este campo se completa y los dos pares de direcciones se presentan al revés que en la pregunta.

Restaría relacionar lo anteriormente explicado con una situación donde los actores se encontraran en distintas redes, como era el caso inicial de nuestro

alumno. Evidentemente, la funcionalidad ARP se desplegará por separado en cada enlace atravesado por el mensaje. De esta manera, en la red donde se encuentra la PC del alumno, su propio ARP generará un mensaje tratando de averiguar la MAC del *router* de salida. La información sobre la IP de dicho router la tomará desde su propia Tabla de Enrutamiento, ya que ese *router* es la puerta de salida o *gateway* a cualquier red diferente de la propia.

Una vez obtenida la MAC del *router* de salida, la PC del alumno podrá generar un mensaje de requerimiento de la página Web en cuestión que tendrá la particularidad de llevar como Dirección Destino IP la del servidor que aloja la página buscada, y como dirección MAC Destino la del *router* de salida de la red donde se encuentra el alumno cliente. Al reconocer el *router* su propia MAC en la trama, la levantará. Al leer la Dirección Destino IP, consultará en su propia Tabla de Enrutamiento y elegirá la interfaz apropiada de salida para la entrega del mensaje al siguiente *router* en el camino hacia el destino.

De compartir estos dos *routers* un enlace tipo LAN de difusión, se editará el requerimiento ARP para la dirección IP del *router* del próximo salto. Esta situación se repetirá en cada enlace de difusión que exista en el camino hacia el servidor Web. En cualquiera de estos enlaces, las direcciones fuente y destino IP del mensaje de requerimiento de la página Web se corresponderán con la fuente original del mensaje (la PC del alumno) y el destino final del mismo (la dirección IP de la máquina que aloja la página Web). El detalle a nivel de enlace, es que las direcciones irán variando entre fuente y destino, dentro de cada enlace particular. Recién cuando el mensaje llega al último *router*, en el ejemplo el *gateway* de la Facultad de Ingeniería, habrá un requerimiento ARP preguntando por la dirección IP del servidor Web. Sólo en la entrega final, la dirección IP y la dirección MAC destino del mensaje serán las del servidor. En este último tramo la dirección IP fuente es la de la PC del alumno, en tanto que la MAC fuente se corresponderá con la del *gateway* de la Facultad.

Como conclusión, cualquier mensaje que viaja a través de Internet, puede ir variando sus direcciones a nivel de enlace, pero nunca se cambian las direcciones IP origen y destino.

En la Fig. 9.11 se presenta un ejemplo de dos máquinas separadas por un *router*. La idea sería investigar los requerimientos ARP que serán necesarios para poder realizar un *ping* entre las dos máquinas A y B, separadas por el *router* R.

El comando *ping* se utiliza como herramienta para realizar pruebas de conexión. Sus mensajes se generan a partir del protocolo ICMP y se encapsulan en IP. En este ejemplo, no se presentarán los detalles de dichos mensajes, sólo se pretende hacer hincapié en los campos correspondientes a las direcciones, tanto a nivel MAC como a nivel IP. Es de destacar que el *router* de este ejemplo tiene interfaces sobre dos redes, por lo que posee dos direcciones MAC y debe ser configurado con una dirección IP diferente en cada interfaz.

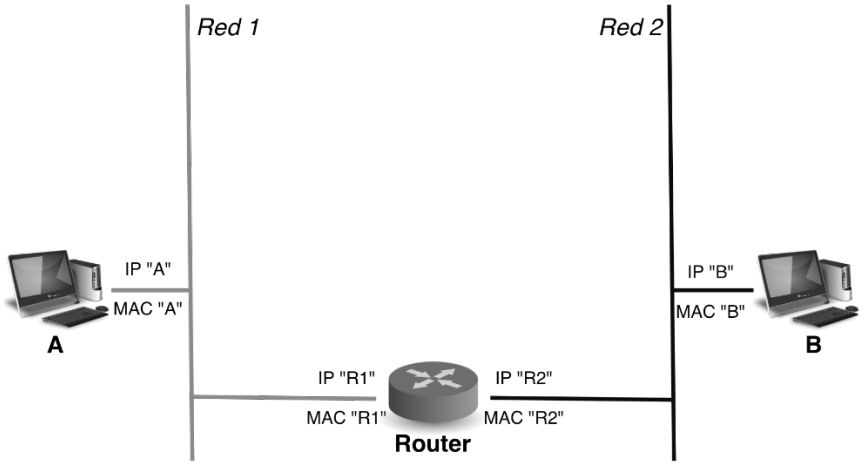


Figura 9.11 - Ping entre máquinas en distintas redes.

La Tabla 9.4 presenta un resumen de los mensajes intercambiados.

Tabla 9.4 – Mensajes intercambiados en un ping entre máquinas en distintas redes.

Mensaje Número	**Tipo**	**Contenido**
1	ARP Req.	***ETHERNET HEADER*:** SRCMAC: MACA. DSTMAC: FF:FF:FF:FF:FF:FF. TYPE: ARP. ***ARP HEADER*:** ARP REQUEST. SENDERMAC: MACA. SENDERIP: IPA. TARGETMAC: 00:00:00:00:00:00. TARGETIP: IPR1
2	ARP Rta.	***ETHERNET HEADER*:** SRCMAC: MACR1. DSTMAC: MACA. TYPE: ARP. ***ARP HEADER*:** ARP REPLY. SENDERMAC: MACR1. SENDERIP: IPR1. TARGETMAC: MACA. TARGETIP: IPA
3	Ping Req.	***ETHERNET HEADER*:** SRCMAC: MACA. DSTMAC: MACR1. TYPE: IP. ***IP HEADER*:** SOURCE IP: IPA. DESTIP: IPB. DATA: ECHO REQUEST
4	ARP Req.	***ETHERNET HEADER*:** SRCMAC: MACR2. DSTMAC: FF:FF:FF:FF:FF:FF. TYPE: ARP. ***ARP HEADER*:** ARP REQUEST. SENDERMAC: MACR2. SENDERIP: IPR2. TARGETMAC: 00:00:00:00:00:00. TARGETIP: IPB
5	ARP Rta.	***ETHERNET HEADER*:** SRCMAC: MACB. DSTMAC: MACR2. TYPE: ARP. ***ARP HEADER*:** ARP REPLY. SENDERMAC: MACB. SENDERIP: IPB. TARGETMAC: MACR2. TARGETIP: IPR2

Mensaje Número	Tipo	Contenido
6	Ping Req.	***ETHERNET HEADER*:** SRCMAC: MACR2. DSTMAC: MACB. TYPE: IP. ***IP HEADER*:** SOURCE IP: IPA. DESTIP: IPB. DATA: ECHO REQUEST.
7	Ping Rta.	***ETHERNET HEADER*:** SRCMAC: MACB. DSTMAC: MACR2. TYPE: IP. ***IP HEADER*:** SOURCE IP: IPB. DESTIP: IPA. DATA: ECHO REPLY
8	Ping Rta.	***ETHERNET HEADER*:** SRCMAC: MACR1. DSTMAC: MACA. TYPE: IP. ***IP HEADER*:** SOURCE IP: IPB. DESTIP: IPA. DATA: ECHO REPLY.

9.4 Protocolo ICMP (Internet Control Message Protocol)

ICMP, el Protocolo de Mensajes de Control de Internet, fue ideado para reportar situaciones de error a los dispositivos de transmisión y proveer mecanismos de testeo de presencia de equipos receptores, funcionalidades con las que el protocolo original IP no fue dotado. Se trata de un protocolo muy básico, implementado mediante un conjunto de mensajes que comparten un mismo formato.

Como se ha explicado, IP es un protocolo de red cuyo mecanismo de entrega es del tipo sin conexión, no confiable y sin mensajes de reconocimiento (ACK). Esto significa que no hay seguridad en la entrega de los paquetes. Prácticamente toda la confiabilidad necesaria para algunos tipos de aplicaciones descansa en TCP. Esta ausencia de confiabilidad a nivel de red se trató de suplir por medio de ICMP, cuyos mensajes se encapsulan en paquetes IP. A pesar de este encapsulado, que conduciría a ubicar el protocolo a nivel de transporte, por encima de IP, el estándar RFC 792 lo coloca al mismo nivel que IP, pero como una entidad separada.

Existen dos versiones del protocolo: ICMPv4 descripto en la RFC 792 para IPv4 e ICMPv6 descripto en la RFC 2463 para IPv6. A su vez, existen otros protocolos que definen sus propios mensajes basados en ICMP, tales como el que desarrolla la funcionalidad de *Traceroute* o el que especifica los mensajes de descubrimiento de *routers*.

En términos generales, ICMP ofrece un mecanismo para la detección de errores, que sólo pueden ser reportados al dispositivo origen del datagrama, debido a que en el paquete IP sólo figuran las direcciones IP fuente y destino. Por su parte, el receptor del mensaje ICMP no tiene obligación de responder o de tomar alguna precaución. Sólo se lo notifica.

La Tabla 9.5 presenta un listado de los mensajes ICMP definidos para la versión 4. Todos ellos son reconocidos por el número "1" presente en el campo Protocolo del encabezado IP que los encapsula. Los mensajes se pueden generar

en *routers* o dispositivos finales, según el caso. Se dividen en dos grandes grupos: mensajes de reporte y mensajes del tipo requerimiento/repuesta.

Los mensajes de reporte se utilizan para comunicar situaciones anómalas. Los mensajes de tipo requerimiento/respuesta son generados por comandos o programas especiales con el propósito de solicitar algo a otro dispositivo o, simplemente constatar su existencia con una respuesta.

Es de destacar que, como los mensajes ICMP generan una carga extra sobre la red, existen una serie de reglas para el intercambio de los mismos. Por ejemplo, si cada dispositivo que recibiera un reporte de error lo contestase, se podría generar un efecto avalancha. Por este motivo, existen reglas que establecen que no se deben generar mensajes de error ICMP en respuesta a alguno de los siguientes mensajes:

- **Mensaje de Error ICMP**: para prevenir un intercambio infinito. Sin embargo, se puede generar un mensaje de error en respuesta a un mensaje de información ICMP.
- **Datagrama destinado a broadcast o multicast**: si uno de estos datagramas generase un error en cada destinatario, se podría generar un enorme volumen de tráfico.
- **Fragmentos IP**: excepto el primero. De no existir esta regla, se generaría un tráfico innecesario.
- **Datagrama con dirección de fuente tipo No-Unicast**: esto previene que los mensajes ICMP sean enviados por *broadcast* o a algún tipo de dirección especial no-ruteable.

Como se observó anteriormente, todos los mensajes comparten un formato que se presenta en la Fig. 9.12. Cabe aclarar que el *checksum* se calcula como en el caso del encabezado IP, pero cubre todo el mensaje ICMP.

El cuerpo del mensaje transporta información relevante a cada mensaje específico. Para los mensajes de error se incluye una parte del paquete que produjo el error: el encabezado IP completo y los primeros 8 $bytes$ de los datos, ya que allí se pueden leer los números de puertos TCP o UDP fuente y destino, que se escriben al principio de los respectivos encabezados. De este modo, se puede realizar un mejor diagnóstico y comunicar a niveles superiores lo sucedido.

Tabla 9.5 - Mensajes ICMP.

Tipo	Código	Descripción	Requerimiento	Error
0	0	Respuesta de Eco (ping)	x	
3	0	Red inalcanzable		x
	1	Host inalcanzable		x
	2	Protocolo inalcanzable		x
	3	Puerto inalcanzable		x
	4	Paquete muy grande		x
	5	Fallo de Enrutamiento de origen		x

Tipo	Código	Descripción	Requerimiento	Error
	6	Red destino desconocida		x
	7	Host destino desconocido		x
	8	Obsoleto		x
	9	Red destino administrativamente prohibida		x
	10	Host destino administrativamente prohibido		x
	11	Red inalcanzable por TOS		x
	12	Host inalcanzable por TOS		x
	13	Comunicación prohibida administrativamente por filtrado		x
	14	Violación de precedencia de host		x
	15	Corte de precedencia en efecto		x
4	0	Control de Flujo		x
5	0	Redirección para red		x
	1	Redirección para host		x
	2	Redirección para TOS y red		x
	3	Redirección para TOS y host		x
8	0	Requerimiento de eco (ping)	x	
9	0	Aviso de ROUTER	x	
10	0	Solicitud de ROUTER	x	
11	0	TTL = ”0” en tránsito		x
	1	Tiempo de vida en cero en re-ensamble.		x
12	0	Problema de parámetros, Header IP con errores		x
	1	Problema de parámetros, Opción requerida no hallada.		x
13	0	Requerimiento de sello de tiempo	x	
14	0	Respuesta de sello de tiempo	x	
15	0	Requerimiento de información	x	
16	0	Respuesta de información	x	
17	0	Requerimiento de máscara	x	
18	0	Respuesta de máscara	x	

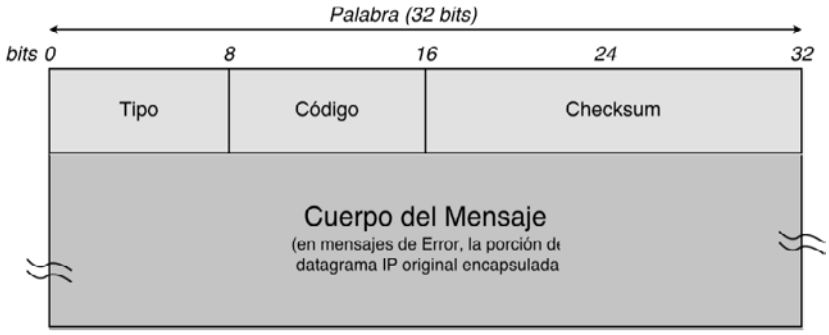


Figura 9.12 - Formato de mensaje ICMP.

9.4.1 Mensajes de Error ICMP

En la Tabla 9.5 se pueden observar un gran número de mensajes de error de Tipo 3, generalmente conocidos como de Destino Inalcanzable. Estos mensajes se incluyen para complementar la funcionalidad *best effort* asociada a IP. En los casos generales de pérdida de paquetes por fallas de entrega, TCP suplirá la falla con algún mecanismo de retransmisión, pero en otros casos, tales como errores en las Tablas de Enrutamiento o inexistencia de la dirección IP destino, la retransmisión no servirá. Los mensajes ICMP de este tipo permiten informar al extremo transmisor el problema en cuestión.

El mensaje **Destino Inalcanzable** es generado en el caso que un dispositivo intente transmitir un datagrama por entrega directa pero, por alguna razón, no pueda alcanzar el destino. Un ejemplo puede darse en el caso de un *router* del último salto tratando de enviar requerimientos ARP a un dispositivo que no esté presente o activo.

El mensaje de **Red inalcanzable** se generará cuando existan problemas en la Tabla de Enrutamiento y el paquete no pueda ser entregado a la red destino, en tanto que el de **Host inalcanzable** implica que se ha ruteado la red destino pero no se encuentra el host, probablemente también por problemas de enrutamiento.

También existen mensajes de error ICMP que indican la prohibición de entrega por problemas administrativos, típicamente por presencia de un *firewall* que descarte tráfico que no respete su propia política de seguridad. En realidad, lo más correcto desde el punto de vista de la seguridad, sería no informar estas situaciones porque enviando estos mensajes se descubren políticas de seguridad. En la actualidad se aconseja descartar silenciosamente aquellos paquetes que generarían este tipo de mensajes.

Los mensajes de **Puerto Inalcanzable** se generan ante situaciones de entrega de paquetes a una aplicación no disponible. Generalmente se trata de mensajes de aplicaciones encapsuladas en UDP, cuando el mensaje se dirige a un número de puerto no usado por un proceso servidor.

El mensaje **Protocolo inalcanzable** se refiere a un error en el número que figura en el campo Protocolo del paquete IP.

El mensaje de **Paquete muy grande** se refiere a la imposibilidad de re-enviar el paquete IP sobre una red de MTU menor, debido al bit DF levantado en el encabezado IP. Existe un mecanismo de aplicación actual, denominado descubrimiento de la MTU de un camino (*path MTU discovery*), que intencionalmente genera paquetes IP con el bit DF en alto, con el propósito de generar este tipo de mensajes de error ICMP, para encontrar la MTU menor en una ruta determinada entre fuente y destino.

El mensaje de **Fallo de Enrutamiento de Origen** se refiere a fallas en la entrega de un paquete con opción de enrutamiento de origen.

El resto de los mensajes de este tipo son obsoletos, ya sea por cuestiones de seguridad o por la adopción práctica de un enrutamiento que no tiene en cuenta el campo TOS del encabezado IP. Lo mismo sucede con el mensaje de Control de Flujo, cuyo mecanismo primitivo ha sido ampliamente superado por el método utilizado en TCP.

Los mensajes de **Redirección** se pensaron como un mecanismo de instalación de rutas de manera automática para evitar la configuración manual de la Tabla de Enrutamiento, pero dejaron de usarse porque, en principio, instalaban rutas a dispositivos, no a redes, resultando de este modo ineficientes. En la versión IPv6 se retoma su utilización, pero con un nuevo significado.

En cuanto al mensaje informativo **Tiempo Excedido** referido a que la cuenta del campo TTL ha llegado a "0" en el encabezado IP, este mensaje se genera para evitar sobrecargar la red con paquetes en el caso de errores en las Tablas de Enrutamiento que puedan generar lazos de enrutamiento (*routing loops*), donde los paquetes circulan entre *routers* sin poder entregarse al destino final. El receptor del mensaje podría optar por reaccionar aumentando el valor del TTL de los nuevos paquetes generados y probar su re-envío para chequear si el error pudo haber sido éste, o es que el valor del TTL utilizado por default es muy pequeño.

El otro mensaje de este tipo, **Tiempo de Vida en Cero en Re-ensamble**, se refiere a alguna situación de error en el destino final debido al re-ensamble de algún paquete. Recordemos que, en IPv4, la fragmentación se puede generar en cualquier punto de la red, pero el re-ensamble es un proceso que sólo se realiza en destino. El proceso de re-ensamble lleva asociado un tiempo límite para evitar la asignación de recursos infinita a una situación donde, por ejemplo, existan fragmentos perdidos.

Los mensajes referidos a **Problemas de Parámetros** se generan al procesar algún paquete IP cuyo encabezado posea algún campo con valores inesperados. El mensaje incluye un puntero que indica el campo del encabezado causante del problema.

9.4.2 Mensajes de Requerimiento/Respuesta ICMP - *ping*

Como se puede apreciar en la Tabla 9.5, aparte de los mensajes de error ICMP además incluye una serie de mensajes de tipo requerimiento-respuesta.

Un par de estos mensajes se utiliza para chequeo de la conexión, de tal manera que si dos dispositivos se quieren comunicar, uno le enviará al otro un mensaje tipo **Requerimiento de eco** y, si el segundo está presente, le responderá con un mensaje **Respuesta de eco**. De este modo, el primer dispositivo reconoce que la comunicación es posible o que el destino es alcanzable. Estos mensajes, además de los campos Tipo, Código y Checksum, llevan un campo de Identificador y otro de Número de Secuencia, ambos de 16 $bits$. El campo Identificador se utiliza para correlacionar el requerimiento con la respuesta y su valor depende de la implementación particular. El campo Número de Secuencia sirve al mismo propósito y su presencia se debe a que el comando *ping* con el que se generan estos mensajes, emite más de un mensaje por vez, generalmente tres o cuatro, por default según el Sistema Operativo. A continuación, sigue un campo de datos opcionales, cuyo valor también depende del Sistema Operativo que genere los mensajes. Su longitud es configurable mediante parámetros del propio comando *ping*.

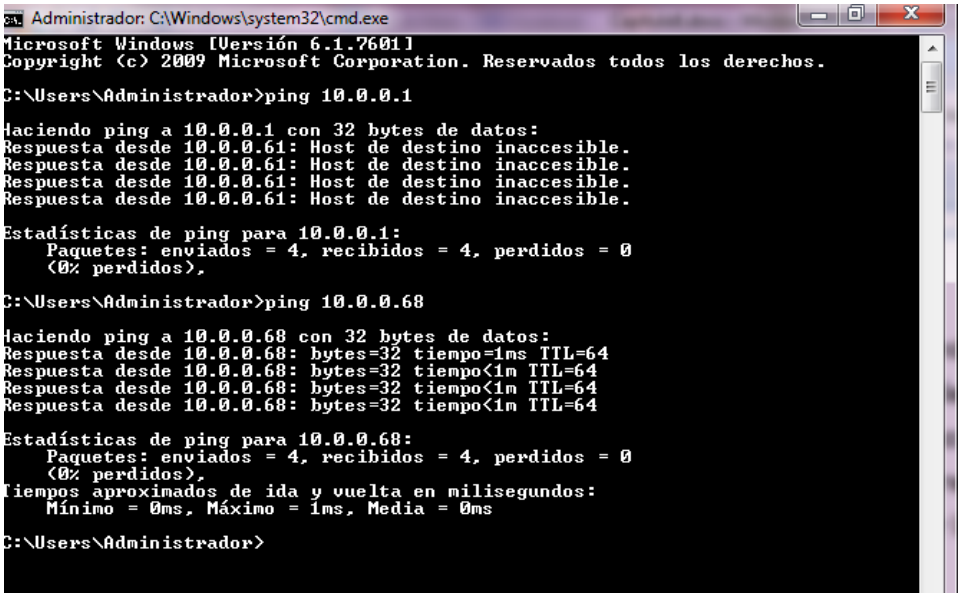


Figura 9.13 - Captura de pantalla, comando *ping*.

Para comprender mejor el funcionamiento de este comando, se presentan algunas pruebas realizadas desde la línea de comandos de una máquina con dirección IP 10.0.0.61. Primero, se realizó un ping a la máquina 10.0.0.1 con el comando "ping 10.0.0.1". Esta máquina no se encontraba disponible, así que, si observamos el tráfico de red, veremos el pasaje de tres tramas consecutivas de requerimiento ARP con dirección destino MAC de *broadcast*, preguntando por la dirección IP 10.0.0.1. Al no obtenerse una respuesta ARP, el propio comando nos devuelve un mensaje de Host de destino inaccesible, a pesar de no haberse generado ningún mensaje ICMP. Esta situación se presenta en la captura de pantalla de la Fig. 9.13.

A continuación se editó otro comando *ping*, ahora dirigido a una máquina que estaba presente, con dirección 10.0.0.68. En este caso, existe una respuesta ARP, así que se puede editar el mensaje de Requerimiento de Eco y recibir la Respuesta de Eco.

El mensaje de requerimiento se encamina hacia el destino, en este caso en la misma red. El sistema destino copia la información de eco en un mensaje nuevo de Respuesta de Eco, destruye el mensaje original ICMP de Requerimiento, convierte su propia dirección en la dirección fuente del mensaje de respuesta, llenando la de destino con la dirección IP del dispositivo originador. El mensaje es encaminado por la red de vuelta al origen, donde el comando presenta al usuario cuatro respuestas, porque se generan cuatro mensajes por default. Esta información se muestra al final junto a una estadística de tiempos de ida y vuelta, y de paquetes enviados y paquetes recibidos en la comunicación.

A continuación se presentan dos mensajes de requerimiento y respuesta de eco, obtenidos mediante un *sniffer*:

Requerimiento de Eco

Ethernet II,
Src: 90:2b:34:aa:bb:cc
Dst: 1c:6f:65:dd:ee:fe
Type: IP (0x0800)

Internet Protocol, Versión: 4
Longitud de Encabezado: 20 bytes
Servicio Diferenciado: 0x00
Longitud Total: 60
Identification: 0x19b6 (6582)
Flags: 0x00
Fragment offset: 0
TTL: 128
Protocol: ICMP (0x01)
Header checksum: 0x0c8b
Source: 10.0.0.61
Destination: 10.0.0.68

Internet Control Message Protocol, Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0x4d52
Identifier: 0x0001
Sequence number: 0x0009
Data (32 bytes): 0x 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70

(abcdefghijklmnopqrstuvwabcdefghi)

Respuesta de Eco

Ethernet II,
Src: 1c:6f:65:dd:ee:fe
Dst: 90:2b:34:aa:bb:cc
Type: IP (0x0800)

Internet Protocol, Version: 4
Longitud de Encabezado: 20 bytes
Servicio Diferenciado: 0x00
Longitud Total: 60
Identification: 0x5831
Flags: 0x00
Fragment offset: 0
TTL : 64
Protocol: ICMP (0x01)
Header checksum: 0x0e10
Source: 10.0.0.68
Destination: 10.0.0.61.

Internet Control Message Protocol, Type: 0 (Echo (ping) reply)
Code: 0
Checksum: 0x5552
Identifier: 0x0001
Sequence number: 0x0009
Data (32 bytes): 0x 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70
(abcdefghijklmnopqrstuvwabcdefghi)

Se observa que el dispositivo 10.0.0.61 crea un Requerimiento de Eco ICMP con dirección fuente propia y dirección destino 10.0.0.68. Los datos que carga dependen de la implementación, en este caso se trata de las letras del abecedario que se repiten hasta alcanzar la longitud del mensaje por default adoptada por el Sistema Operativo. Se observa que la longitud total anunciada por el encabezado IP es de 60 *bytes*, significando que el mensaje ICMP es de 40 *bytes*. Si a estos 40 *bytes* se le restan los 8 *bytes* de encabezado ICMP, los datos ocupan un espacio de 32 *bytes*, como se puede comprobar.

Existe otro tipo de mensajes de Requerimiento/Respuesta, se trata de los mensajes de **Sello de Tiempo**, que se idearon originalmente para sincronizar aplicaciones en diferentes máquinas. En Internet, las máquinas de usuario y los *routers* operan de manera independiente, cada uno con su propio sistema de reloj, pudiendo existir diferencias entre ellos en cuanto a la exactitud de la medición del tiempo, o el tiempo de inicialización. Este par de mensajes ICMP permite el intercambio de información de tiempo entre dos dispositivos. Al igual que los mensajes de eco, llevan un Identificador y un Número de Secuencia. A estos campos se agregan los siguientes: Sello de Tiempo de origen, Sello de Tiempo del receptor (llenado por el destino al momento de la recepción) y Sello de Tiempo de transmisión (llenado por el destino al momento de bajar el paquete a

la red). Estos dos últimos sirven para obtener información sobre el tiempo de procesamiento en el receptor. Los tres Sellos de Tiempo se representan en el formato de Tiempo Universal o del Meridiano de Greenwich (UT, Universal Time). Se trata del número de milisegundos transcurridos desde la medianoche que ocupa un campo de 4 $bytes$.

Por apoyarse en un intercambio de paquetes IP, esta medición de diferencias de tiempo no es confiable, por lo que han aparecido métodos más sofisticados para sincronismo de máquinas, por ejemplo el Protocolo de Tiempo de Red (NTP, Network Time Protocol).

Respecto de los mensajes de **Aviso de router** y **Solicitud de router**, se idearon para colaborar en la configuración de dispositivos conectados a Internet. Tratando de evitar la ineficiencia que significa la configuración manual del *router* de salida de una red en cada una de las máquinas de la misma, se ideó un proceso de Mensajes de Descubrimiento de Router, definido en la RFC 1256, que utiliza los mensajes ICMP mencionados. Si poseen esta capacidad, los *routers* transmiten mensajes periódicos, cada 10 $minutos$ por ejemplo, del tipo Aviso de *router*, proveyendo información importante, tal como su propia dirección IP. Los dispositivos de usuario escuchan estos mensajes y agregan la información a su propia Tabla de Enrutamiento.

Cualquier dispositivo que no haya sido configurado manualmente, al momento del arranque desconocerá su propio *router* de salida hasta no recibir el mensaje de Aviso. En lugar de esperar la llegada de un mensaje, también puede transmitir un mensaje de Solicitud, obligando al *router* a contestar mediante un mensaje extra de respuesta del tipo Aviso *unicast*. Excepto en el último caso, los mensajes de Aviso se transmiten con dirección destino *multicast* 224.0.0.1, para que todos los dispositivos asociados a esa dirección puedan escucharlos. Por su parte, los mensajes de Solicitud se transmiten sobre la dirección *multicast* a la que se asocian todos los *routers*, 224.0.0.2.

En la actualidad, la alternativa de configuración automática más utilizada en IPv4 es el protocolo DHCP, que es capaz de ofrecer mayor cantidad de información que este mecanismo ICMP.

Por su parte, los mensajes de **Requerimiento de máscara** se idearon para soporte de una técnica más moderna denominada *subnetting*. Esta técnica permite expandir el esquema de direcciones tradicional tipo *classful* de dos identificadores, NetID y HostID, a tres identificadores. A los identificadores originales, se agrega un Identificador de Subred, generándose una modificación respecto de las máscaras tradicionales de 8, 16 y 24 $bits$. Los mensajes ICMP de este tipo sirven para conseguir la máscara de subred de la red local, por requerimiento a los *routers*.

9.4.3 Herramienta *Traceroute*

Se ha mencionado la utilización de los mensajes de Requerimiento de Eco y Respuesta de Eco por parte del comando *ping* para chequeo de la conexión entre dos dispositivos. Existe otra herramienta de diagnóstico que sirve para descubrir la secuencia de *routers* presentes a lo largo de todo el camino entre fuente y destino.

Se trata del comando *traceroute* que se invoca acompañado de la dirección destino. Este comando genera un primer mensaje al destino con el campo TTL del datagrama IP ajustado a 1. Entonces, al transmitirlo sobre la red, el generador del mensaje recibe de su propio *router* de salida un mensaje ICMP de **Tiempo excedido**. Luego, se genera un mensaje con el campo TTL en 2, que logra pasar el primer *router* pero que es descartado en el segundo, aunque allí se genera nuevamente un mensaje de **Tiempo excedido** que permite obtener la dirección de este segundo *router*. A medida que se van conociendo los *routers* en el camino, se aumenta paulatinamente el valor del campo TTL, en cada mensaje, hasta alcanzar el destino final. La llegada a destino final se certifica con la generación de un mensaje ICMP de **Puerto Inalcanzable**, pues *traceroute* genera mensajes dirigidos a un puerto destino UDP probablemente no usado.

Esta forma original de *traceroute* fue modificada en la RFC 1393 para que el mecanismo fuera más eficiente y tuviera en cuenta posibles cambios de ruta en el medio de la prueba. El nuevo mecanismo usa un mensaje ICMP que únicamente sirve para este propósito. Para ello, se define una nueva **opción IP Traceroute**, cuya presencia en un paquete de Eco ICMP (u otro), llamado paquete saliente, hará que cada *router* en el camino a destino envíe el nuevo mensaje ICMP Traceroute al originador del paquete saliente. De esta forma, se generarán menos paquetes que con el esquema anterior.

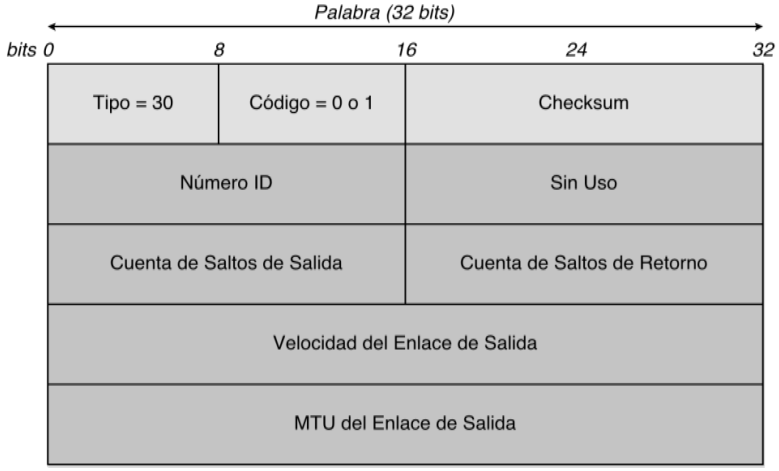


Figura 9.14 - Formato de mensaje ICMP Traceroute.

En la Fig. 9.14 se presenta el formato del nuevo mensaje ICMP. El campo Tipo define el nuevo mensaje, en tanto que el campo de Código puede tener dos valores posibles, según que el paquete saliente haya sido ruteado correctamente (“0”) o haya sido descartado (“1”). El campo de Número ID se copia de la opción

IP *Traceroute* del paquete que genera el mensaje *Traceroute*. Los campos de Cuenta de Saltos de Salida y de Cuenta de Saltos de Retorno también se copian de la mencionada opción y en iguales circunstancias. El primero se refiere al número de *routers* por los que ha pasado el mensaje original. El segundo registra el número de *routers* por los que pasa el mensaje de retorno. El campo de Velocidad del Enlace de Salida se especifica en $bytes/seg$ y el de MTU del Enlace de Salida en $bytes$.

Aunque este método posee ventajas sobre el primero, precisa el soporte de la nueva opción por parte de dispositivos y *routers*. Por este motivo, la mayoría de los Sistemas Operativos continúan utilizando el método de TTL variable.

9.5 Enrutamiento IP

9.5.1 Procedimiento

En el caso de usuarios finales, el concepto de enrutamiento es muy sencillo. Cada máquina de usuario debe tener almacenada una Tabla de Enrutamiento que contenga información sobre su propia red y la de su *router* de salida por default, también conocido como *gateway* o puerta de salida. Diferente es el caso de los *routers*, principalmente porque ellos poseen la capacidad adicional de re-enviar paquetes que toman desde alguna de sus interfaces y descargan sobre otra.

En términos generales, a nivel IP existe una Tabla de Enrutamiento en memoria, que es consultada toda vez que exista un paquete a ser transmitido.

En el caso de la recepción de un paquete, primero se chequea su dirección IP destino, para ver si se corresponde con la propia o se trata de una dirección IP de *broadcast* o de *multicast* de un grupo al que la máquina se haya asociado previamente. Si es así, el paquete se levanta y se entregará luego al protocolo especificado en el campo homónimo del encabezado IP. De no ser este el caso, si la máquina se encuentra configurada como un *router*, se consulta la Tabla de Enrutamiento y se re-envía en base a la información obtenida. De tratarse de un dispositivo sin capacidad de enrutamiento, el paquete es descartado de manera silenciosa, esto es sin generación de un mensaje ICMP.

En todos los casos, la Tabla de Enrutamiento en memoria deberá contener mínimamente información sobre la red destino, la dirección IP del próximo *router* y la especificación de la interfaz de salida para la transmisión del paquete.

Es preciso recordar que la filosofía de enrutamiento IP es salto a salto, es decir que se desconoce la ruta completa a un destino. De esta forma, se asume que el *router* del siguiente salto se encuentra realmente más cerca del destino y conectado directamente con el dispositivo transmisor del paquete.

Al recibir un datagrama, la consulta a la Tabla de Enrutamiento se realiza según la siguiente secuencia:

- ✓ Búsqueda de coincidencia de alguna entrada con la dirección destino IP completa (NetID,HostID). Si existe, enviar al *router* de siguiente salto o a la interfaz conectada de manera directa, según la información de ciertas

banderas que caracterizan cada línea de la Tabla. Esta búsqueda contempla los casos de enlaces punto a punto.

- ✓ Búsqueda de coincidencia de alguna entrada con la parte de la dirección IP que se corresponde con el NetID, previa aplicación de la máscara de red correspondiente. Si existe, enviar al *router* de siguiente salto o a la interfaz conectada de manera directa, según la información de ciertas banderas que caracterizan cada línea de la Tabla.

- ✓ Búsqueda de una entrada de tipo *default*. Si existe, enviar al *router* de siguiente salto.

De no verificarse ninguna de las anteriores coincidencias, el paquete se considera no entregable y, de haberse generado en el propio host, se suele informar a la aplicación que lo generó con algún mensaje del tipo *host* inalcanzable o red inalcanzable.

9.5.2 Ejemplo de Configuración de Tabla de Enrutamiento

Para entender mejor los conceptos expresados previamente, se desarrollará un ejemplo de configuración sobre el caso de una red completa que se presenta en la Fig. 9.15.

Se trata de una red administrada de manera privada y conformada por 3 redes internas, Red 2 a Red 4, y un vínculo serie de enlace con el proveedor que se denomina Red 1. También se puede observar que existen dos *routers* internos, Router X y Router Y, y algunos *hosts* en cada red. Se han etiquetado las interfaces con nombres *eth*, simbolizando interfaces a redes tipo *Ethernet* y, para el caso de la RED 1, *sl* es el enlace serie de la propia red con el ISP.

La primera cuestión a resolver consiste en configurar cada interfaz con una dirección IP apropiada. En los primeros tiempos de Internet, el administrador debía contactarse con la entidad administradora de las direcciones IP y solicitar tantas direcciones de red como cantidad de redes internas debía administrar. Este detalle resultó en una consecuencia indeseable que ya hemos mencionado: el problema del vaciamiento del espacio de direcciones IP.

Supongamos que la entidad proveedora, poniendo en consideración que las redes internas son de pocas máquinas, haya entregado al administrador de la red privada tres direcciones de red Clase C públicas : 202.2.2.0, 202.2.3.0 y 202.2.4.0. El administrador deberá repartir estas direcciones entre las tres redes. Por ejemplo, decide numerar la Red 2 como 202.2.2.0, la Red 3 con 202.2.3.0 y la Red 4 con 202.2.4.0. A su vez, dentro de cada red, se debe configurar una dirección IP en cada *host* y en cada interfaz de *router* que se conecte a dicha red. Una posible configuración se presenta en la Fig. 9.16.

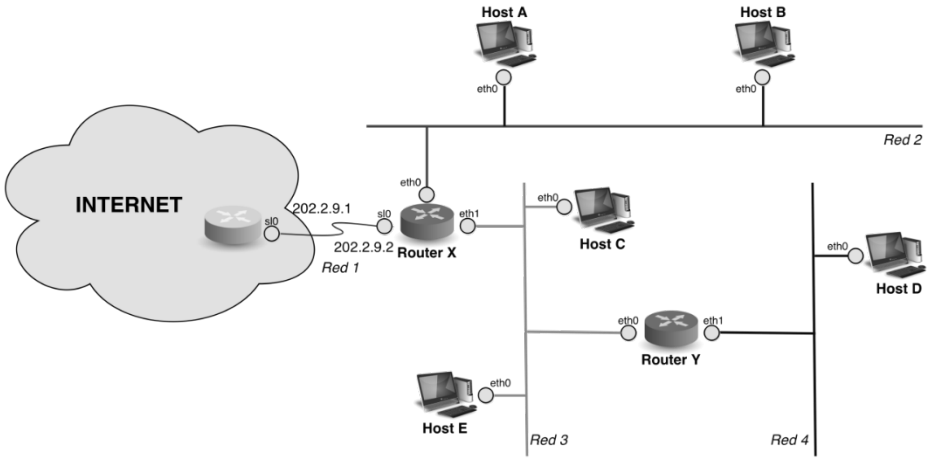


Figura 9.15 - Ejemplo de red sin configurar.

Se suele configurar las interfaces de los *routers* con la primera dirección asignada a cada red. Si la configuración se realiza de manera manual, el administrador debería ejecutar en cada máquina un comando *ifconfig* si el Sistema Operativo es *Linux*, o *ipconfig* si es Windows. Cualquiera de estos comandos posee opciones apropiadas que se pueden consultar a través de la ayuda ofrecida en cada caso. Ejecutando este comando sin opciones o el comando *netstat* con la opción *–i*, se despliega información sobre la configuración.

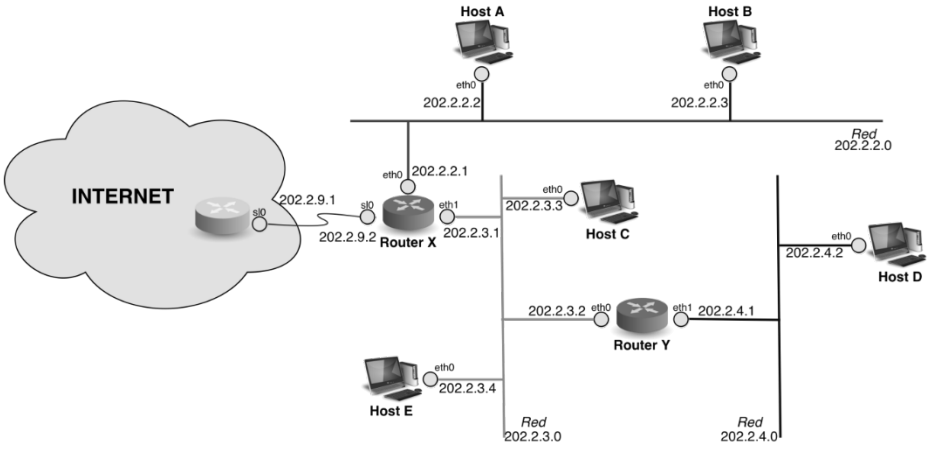


Figura 9.16 - Todas las interfaces con dirección IP.

Aparte de asignar las direcciones IP, cada máquina debe tener cargada una Tabla de Enrutamiento para poder empezar a comunicarse en red. Las Tablas se cargan manualmente con el comando *route* y se pueden comprobar con el mismo comando o con el comando *netstat* con la opción *-r*.

En general, la Tabla de Enrutamiento se despliega como una serie de líneas ordenadas por columnas. En la primera columna se consigna la Red

Destino, luego se escribe su Máscara de Red, el Gateway y la Interfaz a la que se corresponde.

La columna de la **Red Destino** contiene un campo de 32 $bits$ que se consulta luego de una operación de máscara sobre la dirección destino del paquete a transmitir. La máscara permite separar la parte de red de la dirección, es decir el NetID. También en esta columna puede aparecer como destino la red 0.0.0.0, significando *ruta default*, o una dirección IP completa, en el caso de especificarse una ruta a un dispositivo determinado.

La columna de la **Máscara** contiene otro campo de 32 $bits$ que se debe aplicar a través de una operación AND a la dirección IP destino del paquete que se intenta re-enviar. El resultado de esta operación es lo que se compara con los datos que aparecen en la columna Red Destino, en busca de alguna coincidencia.

La columna de **Gateway** contiene la dirección IP del dispositivo del siguiente salto. Esta columna marca el siguiente destino de procesamiento en el camino del paquete IP. La columna de **Interfaz** contiene el nombre con que IP referencia la interfaz de red que se usará para transmitir el paquete a la dirección de próximo salto.

Cada línea de la Tabla que represente una red destino no propia, lleva información sobre la dirección del **Gateway** o Puerta de Salida. Puede además existir información extra, tal como la **Métrica** que es una medida del costo asociado a cada ruta, y otras columnas relacionadas con estadísticas de uso y de paquetes enviados, tales como **Cuentas de Referencia** y **Uso**. También pueden asociarse **Flags** a rutas: G por ruta indirecta o a través de un Gateway, U por Up o activa y H por ruta a *host*.

Por ejemplo, la información básica que se debería almacenar en la Tabla de Enrutamiento del *host* A, sería:

Tabla 9.6 – Tabla de Enrutamiento del host A.

Host A			
Red Destino	*Máscara de Red*	*Gateway*	*Interfaz*
127.0.0.0	255.0.0.0	127.0.0.1	lo
202.2.2.0	255.255.255.0	----------	eth0
202.2.3.0	255.255.255.0	202.2.2.1	eth0
202.2.4.0	255.255.255.0	202.2.2.1	eth0
Internet	----------	202.2.2.1	eth0

La primera línea muestra una ruta a la interfaz de *loopback* y es obligatoria. El resto de las rutas salen por la única interfaz de red que posee este host, la de *Ethernet*, denominada *eth0*. Luego aparece una línea por cada red con las que el host debe comunicarse, comenzando por la red propia, luego las internas y, por último, la salida a Internet. Para la red propia no se precisa Gateway. Para las demás redes, el Gateway es la interfaz del Router X que se encuentra sobre la misma red del *host* A. Se puede observar que las últimas 3 líneas cargan la misma información, por lo que podrían reunirse en una única línea, que denomina *ruta default*, resultando la siguiente una Tabla más real:

Tabla 9.7 – Tabla de Enrutamiento del host A.

Host A			
Red Destino	*Máscara de Red*	*Gateway*	*Interfaz*
127.0.0.0	255.0.0.0	127.0.0.1	lo0
202.2.2.0	255.255.255.0	----------	eth0
Default	----------	202.2.2.1	eth0

La ruta *default* se refiere a toda red que no sea ninguna de las que está explicitada en la Tabla. En el procesamiento de cualquier paquete por parte del *host* A, si la red destino es diferente a la de las dos primeras líneas de la Tabla, la ruta seleccionada es la *default* y el paquete es enviado a la dirección MAC que se corresponde con la dirección IP 202.2.2.1, previo requerimiento ARP. En estos casos, la dirección IP destino y la MAC destino no se corresponderán entre sí, aunque la MAC destino será siempre la de la interfaz *eth0* del Router X.

Si, en cambio, el host A pretende comunicarse con otro dispositivo de su propia red, el requerimiento ARP se editará preguntando por ese dispositivo. En este caso, los mensajes se encapsularán con dirección IP destino y MAC destino correspondiente a un mismo dispositivo.

Cualquier otro *host* de la Red 2 deberá tener configurada la misma tabla.

Siguiendo este razonamiento, la Tabla de Enrutamiento del *host* C sería la misma que la del *host* E:

Tabla 9.8 – Tabla de Enrutamiento del host C.

Host C			
Red Destino	*Máscara de Red*	*Gateway*	*Interfaz*
127.0.0.0	255.0.0.0	127.0.0.1	lo0
202.2.3.0	255.255.255.0	----------	eth0
202.2.4.0	255.255.255.0	202.2.3.2	eth0
Default	----------	202.2.3.1	eth0

Observar que, como el *host* C y el E se encuentran en una red intermedia, no pueden encaminar todo por default. Esto es porque, si desean comunicarse con la red 202.2.4.0, deberán recurrir a la interfaz *eth0* del Router Y. En cambio, para poder comunicarse con cualquier otra red, incluyendo Internet, deben recurrir a la interfaz *eth1* del Router X.

La Tabla para el *host* D en la Red 4, es muy parecida a las Tablas de la Red 2:

Tabla 9.9 – Tabla de Enrutamiento del host D.

Host D			
Red Destino	*Máscara de Red*	*Gateway*	*Interfaz*
127.0.0.0	255.0.0.0	127.0.0.1	lo0
202.2.4.0	255.255.255.0	----------	eth0
Default	----------	202.2.4.1	eth0

La Tabla para el Router Y es:

Tabla 9.10 – Tabla de Enrutamiento del Router Y.

Router Y			
Red Destino	*Máscara de Red*	*Gateway*	*Interfaz*
127.0.0.0	255.0.0.0	127.0.0.1	lo0
202.2.3.0	255.255.255.0	----------	eth0
202.2.4.0	255.255.255.0	----------	eth1
Default	----------	202.2.3.1	eth0

Y para el Router X:

Tabla 9.11 – Tabla de Enrutamiento del Router X.

Router X			
Red Destino	*Máscara de Red*	*Gateway*	*Interfaz*
127.0.0.0	255.0.0.0	127.0.0.1	lo0
202.2.2.0	255.255.255.0	----------	eth0
202.2.3.0	255.255.255.0	----------	eth1
202.2.9.0	255.255.255.0	202.2.9.1	sl0
202.2.4.0	255.255.255.0	202.2.3.2	eth1
Default	----------	202.2.9.1	sl0

En estas condiciones, si desde el *host* A se hiciera un *ping* al *host* D, la primera Tabla que se consulta es la del *host* A, que determina por *ruta default* la entrega del paquete a la interfaz *eth0* del Router X, que se encuentra sobre la Red 2. Esta información sirve para realizar el requerimiento ARP apropiado, para luego encapsular el mensaje en una trama con Dirección MAC Destino correspondiente a la *eth0* del Router X, y Dirección IP Destino la del *host* D. Cuando la trama se vuelca sobre la Red 2, el Router X la asume como propia y, cuando des-encapsula a nivel IP, aplica la máscara a la dirección destino y consulta su propia Tabla de Enrutamiento, determinando que el paquete debe ser re-enviado por su interfaz *eth1* a la interfaz *eth0* del Router Y. Nuevamente se realiza un requerimiento ARP y, al recibir la respuesta, se encapsula el mensaje con Dirección destino IP al *host* D y Dirección MAC destino correspondiente a la de la interfaz sobre esta red del Router Y.

El Router Y levantará este paquete y, al aplicar la máscara a la Dirección destino IP, reconocerá la línea de la red 202.2.4.0, almacenada en su propia

Tabla. Como se trata de la red propia, ahora el requerimiento ARP se hará preguntando por la Dirección IP final, la del *host* D. Al obtener la respuesta, encapsulará un paquete en el que la Dirección MAC y la IP destino corresponden al *host* D. El Router Y re-enviará el paquete por su interfaz *eth1*, como se le indica en la Tabla.

9.5.3 Enrutamiento Dinámico

En el ejemplo previo, las Tablas de Enrutamiento eran de pocas líneas, lo que facilita la posibilidad de cargarlas de manera manual mediante el comando *route*. Las Tablas almacenadas de esta manera no se modificarán a menos que lo haga el administrador. Este tipo de configuración es apropiada en redes pequeñas, con una salida a Internet o pocos *routers* internos. Cuando estas condiciones no se cumplen, generalmente se utilizan protocolos especiales para cargar las Tablas y también para poder modificarlas en el caso de que cambien ciertas condiciones. Esta forma de configuración se relaciona con lo que se denomina enrutamiento dinámico.

La evolución de la propia red de Internet, derivó en una arquitectura descentralizada, en la que existen grupos de redes independientes denominados Sistemas Autónomos (AS, Autonomous System). Cada AS es un conjunto de redes conectadas por *routers*, administradas por una única entidad, que aplica internamente su propia política de enrutamiento y posee un número único que la identifica frente a los demás.

Esta arquitectura asegura un esquema de intercambio de información de enrutamiento más eficiente, ya que la naturaleza del intercambio de información es diferente dentro de un AS que entre AS, utilizándose en cada caso diferentes protocolos. Por ejemplo, entre *routers* de un mismo AS corren protocolos de enrutamiento interior, seleccionables por la administración. Entre Sistemas Autónomos, en cambio, corren protocolos de enrutamiento exterior, para los que hay que establecer un lenguaje común. En este sentido, también se puede distinguir entre *routers* internos al Sistema Autónomo y *routers* de borde, que conectan Sistemas Autónomos entre sí.

A su vez, cada protocolo de enrutamiento utiliza un algoritmo y se caracteriza por una métrica. El algoritmo provee un método para que el protocolo pueda determinar la mejor ruta a un destino y para determinar la forma en que comparte información de enrutamiento con otros *routers*. La métrica es una medida del costo o medida de eficiencia de una ruta particular.

Los algoritmos comúnmente usados son de dos tipos: **Vector Distancia** y **Estado de Enlace**. Los algoritmos de Vector Distancia también se conocen con el nombre de sus inventores: Bellman y Ford. La métrica que utilizan es la cuenta de saltos y el intercambio de información de enrutamiento lo realizan periódicamente y entre *routers* vecinos o directamente conectados entre sí. Los algoritmos de Estado de Enlace, también conocidos como Primero el camino más corto (SPF, Shortest Path First) seleccionan rutas de acuerdo a un seguimiento dinámico del camino más corto entre dos redes. Cada *router* mantiene

información sobre la topología de la red y la actualiza mediante pruebas de red alcanzable e intercambio de información con otros *routers*. Usan diferentes métricas para determinar el mejor camino.

Uno de los protocolos de enrutamiento interior de Vector Distancia más conocidos, debido a su simplicidad y facilidad de configuración, es el Protocolo de Información de Enrutamiento (RIP, Routing Information Protocol). Existen dos versiones para IPv4, RIPv1 y RIPv2, y una para IPv6, conocida como RIPng (next generation).

Aunque la RFC 1058, que describe el protocolo original, recién apareció en 1988, RIP comenzó a utilizarse como un estándar de facto unos años antes, debido al éxito obtenido por una adaptación de un protocolo de Xerox, realizada por un grupo de investigadores de Berkeley en 1982, para la Distribución Estándar de Berkeley (BSD, Berkeley Standard Distribution) del Sistema Operativo UNIX. Este Sistema Operativo hacía correr un proceso, conocido como demonio, denominado *routed*, que se comunicaba sólo usando RIP.

RIP utiliza un algoritmo de Vector Distancia para determinar las rutas. Cada *router* que participa del protocolo, mantiene una Tabla de Enrutamiento con entradas que reflejan las distancias a cada una de las redes almacenadas, medida en saltos. Periódicamente, cada $30\ seg$, cada *router* transmite un mensaje a todos sus vecinos, con información de la Tabla de Enrutamiento. La información del mensaje se encapsula en UDP, y se refiere a la red destino, la distancia del *router* a esa red y la dirección del próximo salto a la red destino. Los *routers* receptores aprovechan este mensaje para actualizar sus propias Tablas, agregando a cada métrica un salto. Luego de la actualización, los receptores transmiten sus propias Tablas a sus vecinos. De este modo, se logra un efecto de propagación de información.

La ventaja de RIP es su sencillez, pero entre sus limitaciones, la versión original presentaba problemas de convergencia en el caso de fallos, por los retardos en la propagación de la información. También tiene un límite máximo de 15 en la cantidad de saltos entre dos destinos, número que puede ser pequeño para Sistemas Autónomos muy grandes.

En respuesta a algunos de estos problemas, se desarrolló un nuevo protocolo de enrutamiento, denominado OSPF (Open Shortest Path First), que permite seguir el estado real de la red en todo momento y, según la información, tomar decisiones de manera dinámica.

Cada *router* dentro de un AS que corra OSPF, mantiene una copia de una estructura denominada Base de Datos de Estado de Enlace (LSDB, Link-State DataBase). Cada enlace a una red o a otro *router* representa una línea en la base de datos que, a su vez, se pesa por un costo. La métrica puede ser de varios tipos, no quedando limitada a la cantidad de saltos como en RIP. Los *routers* que corren OSPF chequean el estado de sus enlaces a cada uno de sus vecinos y envían la información que colectan al resto de sus vecinos. Estos, por su parte, propagan la información a través de todo el AS. Con la información del estado de los enlaces que recibe, cada *router* crea una Tabla de Enrutamiento.

OSPF se encapsula sobre IP. Es un protocolo de Estado de Enlace, a diferencia de RIP que se encapsula en UDP y es un protocolo Vector Distancia. La principal diferencia es que los protocolos de estado de enlace convergen más

rápido, afrontando los cambios con mejor estabilidad. Además, OSPF tiene capacidad para calcular diferentes rutas a un mismo destino, teniendo en cuenta costos relacionados con velocidad, tiempo de ida y vuelta, confiabilidad y otros. También puede realizar balance de carga en el caso que existan varias rutas con el mismo costo a un mismo destino. La desventaja es su complejidad.

Por último, entre *routers* de distintos AS corren protocolos de enrutamiento exterior, por ejemplo el Protocolo de Gateway de Borde (BGP, Border Gateway Protocol). Los *routers* que corren BGP intercambian información referida a la situación alcanzable de una red, que incluye la numeración de los AS a atravesar para llegar a la misma. Esto es apropiado para construir grafos de interconexión entre AS, evitando así lazos de enrutamiento o permitiendo aplicar políticas de enrutamiento específicas. BGP es un protocolo de vector distancia, pero carece de los problemas de RIP debido a la enumeración mencionada. Como se encapsula en TCP, la información intercambiada se puede actualizar sobre una base incremental.

9.6 Subnetting

En el ejemplo de la Sección 9.4.1 se observa el esquema de clases IP original aplicado a un caso concreto. En una gran organización, a medida que la red crece, se van agregando sub-redes. En el esquema original, para poder configurarlas siguiendo la jerarquía de clases, o se solicitaban nuevos bloques de direcciones clase C, uno por cada nueva subred incorporada, o bien se disponía de un gran bloque clase B y así el administrador distribuía internamente el espacio. En el primer caso, el aumento de la cantidad de direcciones generaba el problema de crecimiento de las Tablas de Enrutamiento por cada nueva red incorporada. Por este motivo, inicialmente las direcciones clase B fueron las más solicitadas, generándose un problema de vaciamiento de direcciones IP.

La RFC 950 describió una técnica, denominada *subnetting*, que se podía aplicar al esquema de clases para que el direccionamiento fuera más eficiente y flexible. Teniendo en cuenta que el esquema original de direcciones IP dividía la dirección en dos partes, NetID y HostID, la propuesta de *subnetting* fue agregar un nivel adicional, utilizando parte del espacio de *hosts* para tal fin.

El esquema de *subnetting* propuso que la parte de HostID pudiera dividirse en dos: una parte de subred y la parte de *host* propiamente dicha. De este modo, la jerarquía original pasó a ser de tres niveles. Así, un administrador puede organizar la red en subredes según la estructura interna de la red a administrar. Cada subred poseerá su propio SubnetID y todas compartirán el mismo NetID. Esta técnica no sólo otorga mayor flexibilidad al momento de la configuración, sino que además es invisible desde fuera de la propia organización, evitando el crecimiento de las Tablas de Enrutamiento en la propia Internet.

Este cambio generó diferencias en la estructura original de direcciones en lo que respecta a la definición de la máscara de red. Para que fuera posible encaminar paquetes dentro de una red dividida en subredes, los *routers* debían

ser capaces de identificar la subred correspondiente, mediante una máscara de subred. La Fig. 9.17 presenta un ejemplo de estos cambios.

La forma en que se realiza la división es responsabilidad del administrador de la red. La cantidad de bits que se asignan al SubnetID depende de la cantidad de subredes internas presentes y de la expectativa de crecimiento a futuro de la organización. Así, por ejemplo, si la organización originalmente contara con tres subredes pero esperara crecer a futuro, podría adoptar un SubnetID de 3 $bits$, permitiendo de este modo la configuración de ocho subredes internas. Por ejemplo, la subdivisión de la clase B de Fig. 9.17 genera espacio para 256 subredes internas. Esta información adicional es comunicada a los *routers* a través de la configuración de una máscara apropiada, que no sólo incluye las de las Clase A, B o C original, sino que se extiende para incluir los bits separados para *subnetting*.

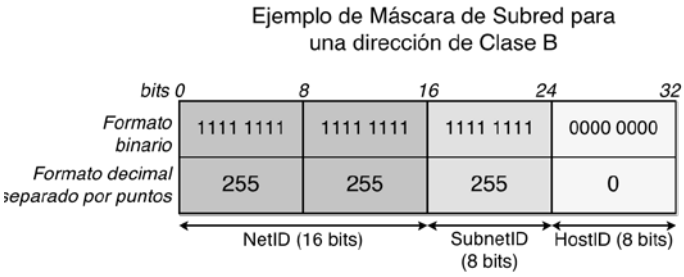


Figura 9.17 - Subnetting.

Modificado así el esquema original, se podrían encontrar direcciones Clase A con máscaras superiores a /8, direcciones Clase B con máscaras superiores a /16 y direcciones Clase C con máscaras superiores a /24. Por ejemplo, una dirección Clase A /11, representa una subdivisión interna de la dirección original en 8 subredes. La misma estructura interna correspondería al caso de una dirección Clase B /19 o una Clase C /27.

9.6.1 Ejemplo de Configuración de Subnetting

Para entender mejor los conceptos expresados previamente, se desarrollará un ejemplo de configuración mediante *subnetting* para el caso de la red de la Fig. 9.15.

Supongamos que la entidad administradora, le haya entregado al administrador de la red privada una dirección de red Clase C numerada como 202.2.2.0. El administrador deberá repartir este bloque entre las tres redes internas, para lo cual deberá elegir la porción de los 8 $bits$ de HostID de la Clase C original, que separará para SubnetID. En este ejemplo, con dos bits sería suficiente, pero considerando un posible crecimiento a futuro de la organización, el administrador decide separar 3 $bits$, dando lugar a una posible expansión interna de 8 subredes. Esta elección implica un límite en la cantidad de *hosts* para cada subred, ya que sólo quedan 5 $bits$ para identificarlos.

De este modo, el administrador puede configurar ocho subredes internas. A saber:

- Subred #0: 202.2.2.(**000**00000)$_2$ = 202.2.2.0; máscara 255.255.255.224 ó /27
- Subred #1: 202.2.2.(**001**00000)$_2$ = 202.2.2.32; máscara 255.255.255.224 ó /27
- Subred #2: 202.2.2.(**010**00000)$_2$ = 202.2.2.64; máscara 255.255.255.224 ó /27
- Subred #3: 202.2.2.(**011**00000)$_2$ = 202.2.2.96; máscara 255.255.255.224 ó /27
- Subred #4: 202.2.2.(**100**00000)$_2$ = 202.2.2.128; máscara 255.255.255.224 ó /27
- Subred #5: 202.2.2.(**101**00000)$_2$ = 202.2.2.160; máscara 255.255.255.224 ó /27
- Subred #6: 202.2.2.(**110**00000)$_2$ = 202.2.2.192; máscara 255.255.255.224 ó /27
- Subred #7: 202.2.2.(**111**00000)$_2$ = 202.2.2.224; máscara 255.255.255.224 ó /27

Se destaca la existencia de la subred #0, ya que tiene la misma numeración IP que el bloque principal asignado. La diferencia entre ambos se explicita a través de la máscara: la de la red completa es $202.2.2.0/24$, mientras que la de la subred #0 es la $202.2.2.0/27$. La diferenciación la realizará el *router* de entrada a la red, en este caso el Router X. También es posible numerar una subred identificada por todos "1", como es el caso de la subred #7.

A su vez, dentro de cada subred es posible identificar $(2^5 - 2)$ hosts. Esto es debido a que no se puede numerar el host todos "0" porque se confundiría con la propia denominación de la subred y tampoco es posible usar la numeración de todos "1" porque se corresponde a la dirección de broadcast de la subred.

Como en el anterior ejemplo, dentro de cada subred se debe configurar una dirección IP en cada *host* y en cada interfaz de *router* que se conecte a dicha red. Una posible configuración se presenta en la Fig. 9.18. Se ha elegido asignar la numeración de la subred #1 a la Red 2 (202.2.2.32), la de la subred #2 a la Red 3 (202.2.2.64)y la de la subred #3 a la Red 4 (202.2.2.96).

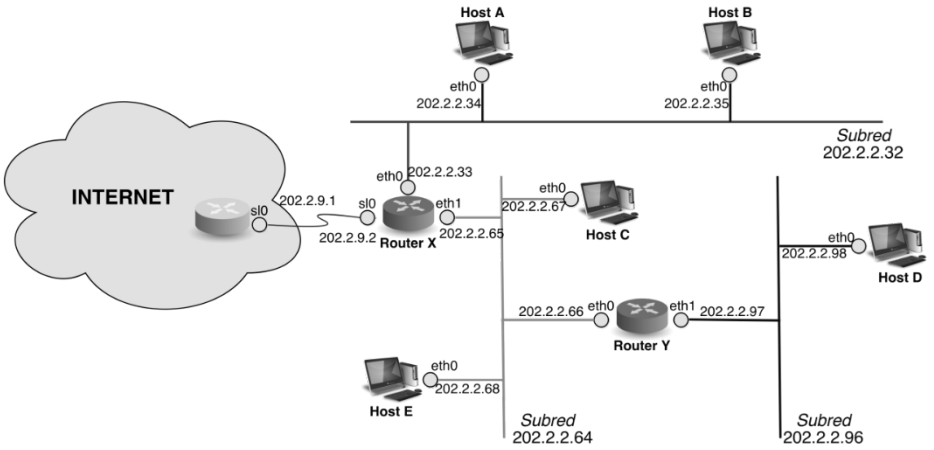


Figura 9.18 - Subnetting. Todas las interfaces con dirección IP.

Como antes, se suele configurar las interfaces de los *routers* con la primera dirección asignada a cada red. El comando *ifconfig* (Linux) o *ipconfig* (Windows) se puede usar para configurar cada interfaz, pero se debe agregar como dato la máscara de subred. Lo mismo sucede cuando se agregan rutas con el comando *route.*

Como ejemplo, desarrollaremos el conjunto de direcciones para una de las subredes. Para la subred 202.2.2.64:

- Dirección de subred #2: 202.2.2.64/27
- Dirección del primer host: 202.2.2.(**010**00001)$_2$ = 202.2.2.65/27
- Dirección del segundo host: 202.2.2.(**010**00010)$_2$ = 202.2.2.66/27
- Dirección del tercer host: 202.2.2.(**010**00011)$_2$ = 202.2.2.67/27
- Dirección del cuarto host: 202.2.2.(**010**00100)$_2$ = 202.2.2.68/27
- Dirección del quinto host: 202.2.2.(**010**00101)$_2$ = 202.2.2.69/27
- Dirección del sexto host: 202.2.2.(**010**00110)$_2$ = 202.2.2.70/27

  ……
- Dirección del host #29: 202.2.2.(**010**11101)$_2$ = 202.2.2.93/27
- Dirección del host #30: 202.2.2.(**010**11110)$_2$ = 202.2.2.94/27
- Dirección de *broadcast* de la subred #2: 202.2.2.(**010**11111)$_2$ = 202.2.2.95

Se observa que la dirección siguiente a la de *broadcast* es la dirección de red de la subred #3. Del mismo modo, la dirección previa a la de la subred #2 es la dirección de *broadcast* de la subred #1.

Cualquier *host* de la Red 2 deberá tener configurada la misma tabla que el *host* A.

Tabla 9.12 – Tabla de Enrutamiento del *host* A.

Host A			
Red Destino	*Máscara de Red*	*Gateway*	*Interfaz*

127.0.0.0	255.0.0.0	127.0.0.1	lo0
202.2.2.32	255.255.255.224	----------	eth0
default	----------	202.2.2.33	eth0

La Tabla de Enrutamiento del *host* C sería la misma que para el *host* E:

Tabla 9.13 – Tabla de Enrutamiento del *host* C.

Host C			
Red Destino	*Máscara de Red*	*Gateway*	*Interfaz*
127.0.0.0	255.0.0.0	127.0.0.1	lo0
202.2.2.64	255.255.255.224	----------	eth0
202.2.2.96	255.255.255.224	202.2.2.66	eth0
Default	----------	202.2.2.65	eth0

La Tabla para el *host D* en la Red 4, es muy parecida a las Tablas de la Red 2.

Tabla 9.14 – Tabla de Enrutamiento del *host* D.

Host D			
Red Destino	*Máscara de Red*	*Gateway*	*Interfaz*
127.0.0.0	255.0.0.0	127.0.0.1	lo0
202.2.2.96	255.255.255.224	----------	eth0
Default	----------	202.2.2.97	eth0

En cuanto a los *routers*, la Tabla para el Router Y es:

Tabla 9.15 – Tabla de Enrutamiento del ROUT Y.

ROUT Y			
Red Destino	*Máscara de Red*	*Gateway*	*Interfaz*
127.0.0.0	255.0.0.0	127.0.0.1	lo0
202.2.2.64	255.255.255.224	----------	eth0
202.2.2.96	255.255.255.224	----------	eth1
Default	----------	202.2.2.65	eth0

y para el Router X:

Tabla 9.16 – Tabla de Enrutamiento del ROUT X.

ROUT X			
Red Destino	*Máscara de Red*	*Gateway*	*Interfaz*
127.0.0.0	255.0.0.0	127.0.0.1	lo0
202.2.2.32	255.255.255.224	----------	eth0
202.2.2.64	255.255.255.224	----------	eth1

202.2.9.0	255.255.255.0	202.2.9.1	sl0
202.2.2.96	255.255.255.224	202.2.2.66	eth1
Default	----------	202.2.9.1	sl0

Observar que en el Router X se hace la separación de las subredes internas en cuanto al enrutamiento. En esta situación, el ISP apunta a esta organización con una única línea en sus propias Tablas de Enrutamiento, la que corresponde a la red $202.2.2.0/24$.

En términos generales, en todos los casos de *subnetting*, si la porción de *hosts* consta de $x\ bits$ y se separan $y\ bits$ para la numeración de subredes, podremos numerar hasta 2^y subredes, cada una con $2^{x-y} - 2$ *hosts*.

9.7 Máscara de Subred de Longitud Variable, VLSM

La técnica de *subnetting* permitió a los administradores asignar direcciones IP en base a las conexiones físicas de las redes, resultando en un esquema más flexible que el tradicional de clases, aunque con alguna desventaja, derivada del hecho de agregar sólo un nivel más en la jerarquía de direcciones. En este sentido, la primer pregunta que surge es cómo resolver casos en los que las subredes posean diferencias notables entre la cantidad de *hosts*, ya que en este caso el esquema planteado resultaría ineficiente.

Supongamos que en el ejemplo del apartado 9.5 la red original $202.2.2.0/24$ deba dividirse en seis subredes. Esta situación obligaría a tomar mínimamente 3 $bits$ para cada subred, dejando espacio en cada caso para 30 *hosts*. Supongamos además, que cuatro de las seis subredes son pequeñas, de pocos *hosts*, apenas unos 12, en tanto que otra subred precisa mantener 50 *hosts* y la restante posee una carga de 100 *hosts*. En total se precisarían direcciones para 162 *hosts*. Si contáramos la cantidad total de direcciones para *hosts* en el esquema, $6x(2^5 - 2) = 180$, pareciera que sobraran direcciones, pero en la realidad no se podrían asignar bajo estas condiciones según las exigencias propuestas. La única solución pareciera ser un pedido de un nuevo espacio Clase C, con el consiguiente gasto de muchas direcciones que quedarían sin utilizar. Esta restricción se debe a una característica propia de la técnica de *subnetting*, que consiste en dividir un espacio en porciones de igual tamaño: la Clase C original se dividió en ocho porciones, de 30 *hosts* cada una.

La técnica de Configuración de Subred de Máscara de Longitud Variable (VLSM, Variable Length Subnet Masking) otorgó una mejora al esquema básico de *subnetting*, justamente para poder proveer solución a los casos tales como el mencionado.

Detrás de VLSM subyace la idea de *subnetting* recursivo: a partir de una red, se realiza un *subnetting*, pero a partir de ese *subnetting* se pueden realizar más, resultando en un esquema de subredes a partir de otras subredes. No es necesario aplicar el mecanismo en todas las subredes presentes, sino únicamente

en las necesarias. De este modo, en el resultado del reparto de direcciones IP, algunas porciones o subredes serán más grandes que otras.

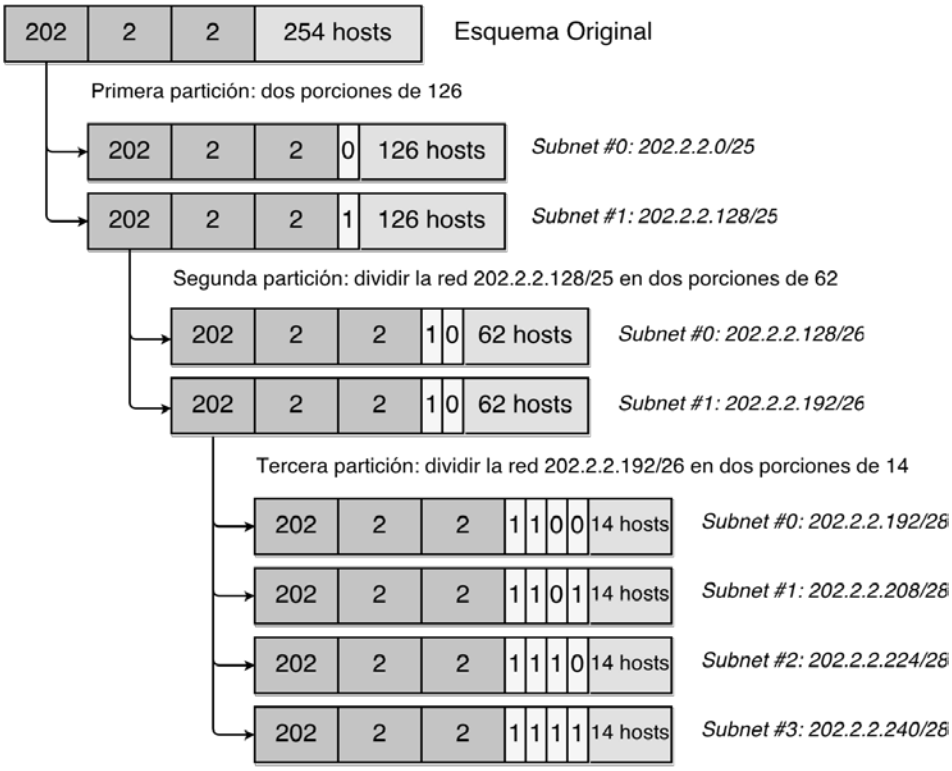


Figura 9.19 - VLSM.

Para el ejemplo mencionado, si partiéramos de la dirección original e hiciéramos un *subnetting* de 1 bit, se generarían dos subredes iguales, cada una con espacio para 126 *hosts*, tal como se presenta en la Fig. 9.19. Así se soluciona el espacio necesario para la subred de 100 *hosts*.

Si se tomara la segunda de las subredes generadas, 202.2.2.128/25, y se volviese a aplicar *subnetting* de 1 bit, se obtendrían las subredes 202.2.2.128/26 y 202.2.2.192/26, cada una con espacio para 62 *hosts*. De este modo, se cubre la necesidad de la subred de 50 hosts.

Si luego se tomara la segunda de esas sub-subredes y se la volviera a dividir, tomando 2 $bits$ adicionales para identificación de subred, obtendríamos cuatro subredes de 14 *hosts* cada una. Así se cubren las redes más pequeñas del problema propuesto. La división completa del espacio se presenta esquemáticamente en la Fig. 9.19.

Con este sencillo ejemplo, se ha pretendido demostrar que el proceso recursivo no requiere la asignación del mismo prefijo extendido de red en cada

nivel de la recursión y puede concretarse según las necesidades con las que se enfrente el administrador.

Detrás de cada *router* donde se aplica *subnetting*, se esconde más de una subred, pero todas pueden verse desde el lado de afuera de ese *router* con el mismo prefijo de subred o máscara de subred. Este concepto se conoce como **agregado de rutas**. Para el ejemplo anterior, si el último *router* tuviera que anunciar mediante un protocolo de enrutamiento las cuatro subredes que están por detrás de él, lo podría realizar en un único aviso, con dirección 202.2.2.192/26. Por encima de él, el *router* de la división recursiva previa, puede agregar las dos subredes también en un único anuncio 202.2.2.128/25. Dado que la estructura de subredes no es visible fuera de la organización, el *router* de entrada inyecta una única ruta sobre la Tabla de Enrutamiento de la Internet global. De este modo, el agregado de rutas, evita el agrandamiento innecesario de las Tablas de Enrutamiento.

Por otra parte, los protocolos de enrutamiento deben ser capaces de llevar información del prefijo de red extendido en cada anuncio de ruta. Los protocolos de enrutamiento modernos, tales como OSPF, permiten el desarrollo de VLSM pues llevan el prefijo de red extendido o máscara junto con cada anuncio de ruta. RIP-2 también permite agregar información de prefijo en los mensajes.

Es de destacar que los *routers* sobre los que se despliegan redes con *VLSM*, deben poder implementar un algoritmo de re-envío de paquetes modificado, consistente con la **máscara más larga**, también conocido como *longest match*. Cualquier ruta con un prefijo más largo, describe un conjunto más pequeño de destinos que la misma ruta con un prefijo de red más corto. Como resultado de esto, una ruta con prefijo más largo, se dice que es más específica mientras que aquella con un prefijo más corto lo es menos. Los *routers* deben seleccionar la ruta con el prefijo de red mayor, el más específico, al re-enviar el tráfico.

Por ejemplo, si la dirección destino de un paquete IP fuera 202.2.2.241 el *router* de entrada lo asumiría como propio al aplicarle la máscara de entrada, entregándolo a la subred 202.2.2.192. Este router aplica la máscara /28 para decidir que el paquete se debe entregar a la subred 202.2.2.240. Efectivamente, el dispositivo 202.2.2.241 deberá alojarse en esta subred, teniendo en cuenta que su prefijo de subred ha coincidido en parte con otros más cortos dentro de la misma administración. Si se encontrara en otra subred, el sistema de enrutamiento no re-enviará nunca el tráfico al *host*, dado que el algoritmo *longest match* supone que ese host es parte de la red con el prefijo más largo. Esta particularidad significa que debe tenerse mucha precaución al asignar direcciones a los *hosts*, simplemente para asegurarse que cada uno de ellos sea accesible.

En general, es aconsejable además, que la asignación de direcciones tenga significado topológico para que sea posible el agregado de rutas. El soporte de enrutamiento jerárquico y la reducción de las Tablas de Enrutamiento de la organización, exigen que la asignación de direcciones refleje la topología real de la red, pues se reduce la cantidad de información de enrutamiento al tomar el conjunto de direcciones asignadas a una región particular de la topología y agregarlas en un único aviso de ruta.

9.8 Enrutamiento Interdominio Sin Clases, CIDR.

La técnica de *subnetting* contribuyó en parte a mejorar el sistema original de clases de direcciones, ya que proveyó un mecanismo más flexible desde el punto de vista administrativo, para que cada organización pueda configurar su propia red de manera invisible al resto. De todas maneras, el esquema quedó sujeto al original y, como consecuencia, no pudo contribuir a resolver el problema de vaciamiento de direcciones IP, que comenzara a percibirse desde la década del 90.

El problema del vaciamiento del espacio de direcciones IP se generó en los comienzos de la gran red, cuando fueron asignadas la mayoría de las direcciones Clase A y Clase B, quedando sólo disponible direcciones Clase C. El vaciamiento de direcciones Clase B significó un verdadero problema para la asignación de espacio en el caso de organizaciones de mediano tamaño, para las que una dirección Clase C, con un máximo de 254 *hosts*, es insuficiente. Por otra parte, se ha mencionado que la asignación de varias direcciones Clase C a este tipo de organizaciones desemboca en el problema de un aumento dramático en el tamaño de las Tablas de Enrutamiento.

En este sentido, comenzaron a aparecer nuevas aproximaciones con diferentes propuestas. Entre ellas, una muy revolucionaria, propuso un esquema de direcciones sin clases o *classless*, definido en las RFC 1517 a 1520 como Enrutamiento Inter Dominio Sin Clases (CIDR, Classless InterDomain Routing).

En cierto modo, la idea que aportó CIDR fue la de adaptar *subnetting* a todo Internet, aunque en un sentido diferente. El direccionamiento sin clases permite que, en vez de tomar un bloque y dividirlo en subredes, se puedan agregar o juntar redes en conjuntos mayores, denominados super-redes, reduciendo de este modo el tamaño de las Tablas de Enrutamiento. Por este motivo, muchas veces se asocia CIDR con el concepto de *supernetting*.

La oportunidad de agregar redes se asoció a la idea de no trabajar con clases, por lo que las subredes agregadas pueden tener diferente tamaño. En este sentido, se puede relacionar CIDR con el agregado de VLSM, pero sobre un esquema en el que se elimina el sistema de clases, reemplazándolo por uno más flexible, de estructura jerárquica de múltiples niveles de diferentes tamaños. Aunque la técnica alivia el problema de crecimiento de las Tablas de Enrutamiento, su aplicación implica cambios en los protocolos de enrutamiento y en la manera de interpretar la información de enrutamiento, ya que el esquema de agregado de redes CIDR exige que las direcciones pertenezcan a bloques contiguos.

El método de asignación es por prefijos de red, previendo que se encuentre ligado a la topología subyacente para que pueda facilitar cuestiones relacionadas con el escalamiento y el sistema global de enrutamiento. Una de las consecuencias de esta estrategia es que la asignación por prefijos queda sujeta a la relación entre un ISP y sus clientes, dado que esta relación es la que determina la topología de Internet.

La asignación de direcciones por prefijos, en CIDR, es una asignación por bloques que tiene en cuenta las necesidades más estrictas del solicitante. Así, por ejemplo, si una organización precisara 6000 direcciones, se le asignará un bloque limitado a una cantidad de bits suficiente para cubrirlo. En este caso, un bloque de 13 $bits$ aportaría $2^{13} = 8190$ direcciones. En el esquema tradicional, se otorgarían 65534 direcciones en un bloque Clase B, en cuyo caso se desperdiciaría espacio para otras organizaciones de este tipo.

Con CIDR deja de existir el concepto de direcciones asignadas por clase, contribuyendo como un paliativo al problema del vaciamiento, aunque esto no implica que no se pueda seguir aplicando técnicas como la de *subnetting*. La contrapartida es que CIDR es un poco más complejo que el esquema original, incluso en el tema de configuración de los *routers*. CIDR, como en el caso de *subnetting*, utiliza una máscara de red para separar NetID de HostID, por lo que las direcciones CIDR se escriben como un par $u.v.w.x/máscara$. Por ejemplo, el bloque 185.11.142.0/22 significa que los primeros 22 $bits$ conforman el NetID, en este caso 185.11.142.0, con máscara de subred 255.255.252.0. Se trata de un bloque de $2^{10} - 2 = 1022$ $hosts$.

Es importante destacar las cuestiones detrás de la decisión de desplegar redes basadas en CIDR. Por ejemplo, muchas máquinas que reconocen el esquema de clases no permitirán una configuración con una máscara más corta que cualquiera de las tradicionales, generando problemas potenciales. Por ejemplo, si deseáramos desarrollar 200.25.16.0 como una /20 para definir una red capaz de soportar $(2^{12} - 2) = 4.094$ *hosts*, el software que se ejecuta en cada máquina podría no permitir que una Clase C tradicional, como es la de la numeración 200.25.16.0 se configure con una máscara de 20 $bits$, ya que lo natural sería asociarla a una de 24 $bits$, o un número mayor de bits si se piensa en un esquema de *subnetting*.

Sin embargo no habría problemas en desarrollar 200.25.16.0/20 para asignación como un bloque de 16 redes de máscara /24 dado que los *hosts* que no soportan CIDR interpretarán su /24 local como una red Clase C. De la misma manera, la dirección Clase B del esquema tradicional 130.14.0.0/16, se podría desplegar como un bloque de 255 redes /24, ya que se interpretarán las /24 como subredes de una /16.

La Tabla 9.17 provee información sobre los bloques de direcciones CIDR más comunes. Se puede ver que una asignación /15 puede también especificarse como 255.254.0.0 y contiene un bloque contiguo de 131.072 direcciones IP, que se puede interpretar como 2 redes Clase B o 512 redes Clase C. En la Tabla 9.17, la primera columna presenta el prefijo en notación CIDR, la segunda muestra la misma información pero en notación tradicional. La tercer y cuarta columnas traducen dichos prefijos a cantidad total de direcciones y su equivalente en clases.

Para el nuevo esquema sin clases, el IANA dividió el espacio en grandes bloques que entregó a los cuatro Registros Regionales de Internet (RIR, Regional Internet Registry) de aquel momento: ARIN para América Anglosajona, RIPE para Europa, el Oriente Medio y Asia Central, APNIC para Asia y la Región Pacífica y LACNIC para América Latina y el Caribe. A su vez, estos registros

regionales dividen sus propios bloques y los distribuyen entre los Registros Nacionales (NIR, National Internet Registry), los Registros Locales (LIR, Local Internet Registry) y los propios ISP. Estos últimos pueden subdividir sus propios bloques para sus clientes y estos, a su vez, repetir el proceso.

Tabla 9.17 - Notación CIDR.

Prefijo CIDR	***Notación Decimal***	***Número de Direcciones Individuales***	***Equivalente en Clases***
/13	255.248.0.0	$2^{19} = 512K$	8 Clases B /2048 Clases C
/14	255.252.0.0	$2^{18} = 256K$	4 Clases B /1024 Clases C
/15	255.254.0.0	$2^{17} = 128K$	2 Clases B /512 Clases C
/16	255.255.0.0	$2^{16} = 64K$	1 Clases B /256 Clases C
/17	255.255.128.0	$2^{15} = 32K$	128 Clases C
/18	255.255.192.0	$2^{14} = 16K$	64 Clases C
/19	255.255.224.0	$2^{13} = 8K$	32 Clases C
/20	255.255.240.0	$2^{12} = 4K$	16 Clases C
/21	255.255.248.0	$2^{11} = 2K$	8 Clases C
/22	255.255.252.0	$2^{10} = 1K$	4 Clases C
/23	255.255.254.0	$2^9 = 512$	2 Clases C
/24	255.255.255.0	$2^8 = 256$	1 Clase C
/25	255.255.252.128	$2^7 = 128$	1/2 Clase C
/26	255.255.252.192	$2^6 = 64$	1/4 Clase C
/27	255.255.252.224	$2^5 = 32$	1/8 Clase C

Además de utilizar unidades bloques de redes contiguas, se cambiaron las reglas de asignación para las direcciones clase C, dividiendo la entrega por regiones geográficas. Cada bloque es de $2^{25} = 33.554.432$ direcciones, con un prefijo común:

- Para Europa: se separó el rango $194.0.0.0 - 195.255.255.255$, con prefijo común /7.
- Para América del Norte: $198.0.0.0 - 199.255.255.255$, con prefijo común /7.
- Para América Central y América del Sur: $200.0.0.0 - 201.255.255.255$, con prefijo común /7.
- Para Asia y el Pacífico: $202.0.0.0 - 203.255.255.255$, con prefijo común /7.
- Reservado: $204.0.0.0 - 223.255.255.255$.

Un *router* fuera de Europa que reciba un paquete con destino dentro del rango $194.xx.yy.zz - 195.xx.yy.zz$, sólo puede enviarlo al *gateway* de Europa. Esto reduce significativamente el tamaño de las tablas ya que un bloque de 33 millones de direcciones se resume en una sola línea de entrada.

Para mayor claridad de conceptos, se recomienda atender el ejemplo de delegación de direcciones desarrollado en la RFC 4632. De todas maneras, a continuación se ofrecen algunos ejemplos.

Ejemplo 1: El bloque $208.128.0.0/11$ es un bloque CIDR de más de dos millones de direcciones que se ha asignado a una empresa X. Una empresa intermedia, designada como Y, alquiló a X una conexión a Internet, otorgándosele el bloque $208.130.28.0/22$, capaz de admitir hasta $2^{10} = 1.024$ direcciones IP. A su vez, Y utilizó parte de su bloque para alojar sus servidores públicos. Eligió para ello el bloque $208.130.29.0/24$, una de cuyas direcciones es $208.130.29.33$. Fuera de la red de X, el prefijo $208.128.0.0/11$ se usa para encaminar hacia X el tráfico dirigido, no sólo a $208.130.29.33$, sino también a cualquiera de los cerca de dos millones de direcciones IP con el mismo prefijo CIDR. En el interior de la red de X, la línea de la Tabla correspondiente a $208.130.28.0/22$ dirigirá el tráfico al bloque alquilado por Y. El prefijo $208.130.29.0/24$ se usará sólo dentro de la red de Y, para direccionar tráfico a sus servidores públicos.

Ejemplo 2: Supongamos que se desea agregar un bloque de direcciones de red, cuya numeración se encuentra entre la dirección $131.0.0.0$ y la dirección $131.7.0.0$. Se precisa encontrar una máscara de subred que permita que las redes aparezcan como pertenecientes al mismo bloque. En *subnetting* se usan bits de la porción de *hosts*, o sea nos movemos de izquierda a derecha sobre la dirección. En *supernetting*, para realizar el agregado, en vez de tomar bits de la porción de *hosts*, tomamos los bits de bajo orden de la porción de red, es decir nos movemos de derecha a izquierda sobre la dirección. En el sistema tradicional, la máscara de una red Clase B es $255.255.0.0$, también referenciada como $/16$, se toman 3 $bits$ de la porción de red, la máscara se convierte en $/13$ o $255.248.0.0$. Así, todo el rango entre $131.0.0.1$ y $131.7.255.255$, puede representarse por un prefijo común $131.0.0.0/13$. Se trata de ocho redes Clase B.

Ejemplo 3: Supongamos tres sitios de Europa, denominados A, B y C, solicitando respectivamente 2048, 1024 y 4096 direcciones. Arrancando a partir de una dirección base $194.24.0.0$, se asignan los bloques según el siguiente esquema:

- Organización A: 2048 direcciones (2^{11}) en el rango $194.24.0.0 - 194.24.7.255$, con máscara $/21$.
- Organización B: 1024 direcciones (2^{10}) en el rango $194.24.8.0 - 194.24.11.255$, con máscara $/22$.
- Organización C: 4096 direcciones (2^{12}) en el rango $194.24.16.0 - 194.24.31.255$, con máscara $/20$.

A partir de esta asignación, los *routers* europeos anexarán entradas para 194.24.0.0/21, 194.24.8.0/22 y 194.24.16.0/20. Si se debe procesar un paquete con dirección destino IP 192.24.17.4, la AND con la máscara /21 separará 192.24.16.0, la AND con la máscara /22 separará 192.24.16.0 y la AND con la máscara /20 separará 192.24.16.0. Evidentemente, la coincidencia de máscara más larga se da en el caso de la Organización C. Es importante comprender que se debe tener cuidado al configurar las direcciones de esta organización, para que no existan ambigüedades, y el destino realmente se ubique en la red de mayor coincidencia de la máscara.

9.9 Traducción de Direcciones, NAT

Se trata de una tecnología desarrollada para extender el tiempo de vida útil del espacio de direcciones de IPv4. El Traductor de Direcciones (NAT, Network Address Translation) permite que un gran número de dispositivos configurados utilizando direcciones privadas, puedan comunicarse internamente por medio de ellas, debiendo compartir un pequeño grupo de direcciones públicas si se desea que la red tenga conexión a Internet. De este modo, se limita bastante la asignación de direcciones IP públicas, con la ventaja adicional de mejorar cuestiones relacionadas con la seguridad ante posibles ataques.

Es decir que NAT permite conservar el espacio de direcciones IP, expandir las redes internamente sin necesidad de solicitar nuevas direcciones IP, cambiar de proveedor de Internet sin tener que reconfigurar cada dispositivo y proveer un nivel mayor de seguridad al operar de manera similar a un *firewall* ya que, al trabajar los clientes con direcciones privadas, no pueden ser accedidos desde Internet, excepto que por configuración así se lo permita.

La técnica de traducción de direcciones es posible debido a la manera habitual en que las distintas organizaciones utilizan Internet.

En la gran red, la mayoría de los *hosts* se comportan como clientes, por lo que no precisan ser accedidos desde la red pública. Es el cliente el que contacta al servidor. En general, las aplicaciones tipo servidor no necesitan iniciar el contacto con el cliente. Por otra parte, estadísticamente hablando, en grandes redes conectadas a Internet, a pesar de tener muchos *hosts*, no todos acceden a la red al mismo tiempo. Por ejemplo, cuando se navega Internet, las conexiones realmente suceden al momento de descargar las páginas, el resto del tiempo, el usuario generalmente está leyendo la información descargada.

Además, todas las comunicaciones hacia Internet, ya sean domésticas u organizacionales, pasan a través de un *router* de salida. Esta característica convierte al propio *router* en un punto de control del tráfico que lo atraviesa.

De este modo, para que la traducción NAT funcione se precisa: un espacio de direcciones privadas, una o más direcciones IP públicas asignadas a la red que pretende conectarse, y un *router* de salida a Internet con capacidad NAT. Es el propio *router* el que oficia de traductor entre direcciones privadas y direcciones públicas, en ambos sentidos de la comunicación, debiendo reemplazar unas y otras en los propios paquetes.

En general, la traducción es transparente, pero puede suceder que algunas aplicaciones tengan problemas en su paso a través del *router*, debido a la manipulación que necesariamente se produce en las cabeceras al realizar la traducción. Cambiar un campo en el encabezado IP implicará, como mínimo, recalcular el campo de Chequeo de Errores. Lo mismo sucederá con los campos de control de error de capas superiores. Entonces, para poder hacer su trabajo correctamente, el *router* NAT debe mantener el estado de las conexiones y esto significa operar sobre múltiples protocolos.

Por otra parte, el propio sistema de traducción agrega una dificultad adicional al acceso a servidores web internos desde la Internet, y significa una carga extra de procesamiento sobre el *router*.

A pesar de estas desventajas, NAT se ha convertido en una herramienta muy utilizada.

Para comprender mejor su funcionamiento, NAT ofrece una terminología propia en cuanto a los distintos tipos de direcciones que se manejan. Por ejemplo, en base al lugar donde se encuentra un dispositivo en la red, se definen:

- **Direcciones internas**: son las pertenecientes a los dispositivos de la red privada.
- **Direcciones externas**: son las de Internet, se trata de direcciones públicas, que se encuentran fuera de la red local.

  También, en base a la localización del datagrama:
- **Dirección Local**: se refiere a una dirección que aparece en un paquete en la red interna, ya sea que se refiera a cualquiera de los dos casos anteriores.
- **Dirección global**: se refiere a una dirección que aparece en un paquete en la red externa.

El *router* NAT oficia de interfaz entre la red interna y la red externa. Los dispositivos internos usan un esquema de direccionamiento de red local, a diferencia de los externos, que usan un direccionamiento global. Así, es posible definir cuatro tipos de direcciones específicas:

- **Direcciones internas locales**: las de los dispositivos en la red local. Por ejemplo, pueden ser una dirección de la Clase A 10.0.0.0.

- **Direcciones internas globales**: son públicas y se las llama enrutables. Se asignan a los dispositivos internos para que puedan comunicarse con Internet. La asignación se hace en el *router* NAT. Por ejemplo, en el caso en que un cliente interno, alojado en una máquina con dirección 10.0.0.27, desee comunicarse mediante un requerimiento HTTP con un servidor en Internet habilitado en la dirección 200.151.116.12 , la máquina cliente encapsula el requerimiento en un paquete con Dirección Fuente 10.0.0.27 y Dirección Destino 200.151.116.12 . Cuando el *router* tome el datagrama, traducirá la dirección interna, reemplazándola con alguna de las direcciones públicas asignadas a la organización, por

ejemplo 170.210.65.4. De otro modo, la respuesta no llegaría porque el bloque 10.0.0.0 es no enrutable, recordemos que se trata de uno de los bloques separados para redes privadas. La dirección 170.210.65.4 es la dirección interna global que se corresponde con 10.0.0.27.

- **Direcciones externas globales**: son públicas y registradas, por ejemplo la dirección del servidor 200.151.16.1.

- **Direcciones externas locales**: se trata de direcciones de dispositivos externos, tal cual como se referencian en los dispositivos en la red local. En ocasiones, pueden ser las mismas que las externas globales de los dispositivos externos.

Un ejemplo de esta clasificación se presenta en la Fig. 9.20.

Es decir que NAT traduce identidades de dispositivos internos o externos, partiendo de representaciones locales y llegando a las globales, y al revés. Por ejemplo, en NAT tradicional, los dispositivos internos se refieren a los externos usando su dirección global, coincidiendo de este modo las direcciones externas globales con las externas locales.

Para saber cómo se deben traducir las direcciones, en el *router* existe una tabla de traducción que permite realizar el mapeo, ya que guarda la relación entre las direcciones locales internas y las globales internas. De ser necesario, también guardará el mapeo de globales externas a globales internas para transacciones entrantes.

La Fig. 9.21 ofrece un ejemplo de traducciones en el caso de una solicitud cliente desde la red privada.

En la tabla de traducción, las entradas pueden ser estáticas, también llamadas permanentes. Este sería el caso si deseáramos que un dispositivo interno con dirección 10.0.0.27 siempre usara una dirección interna global fija, por ejemplo 170.210.65.4.

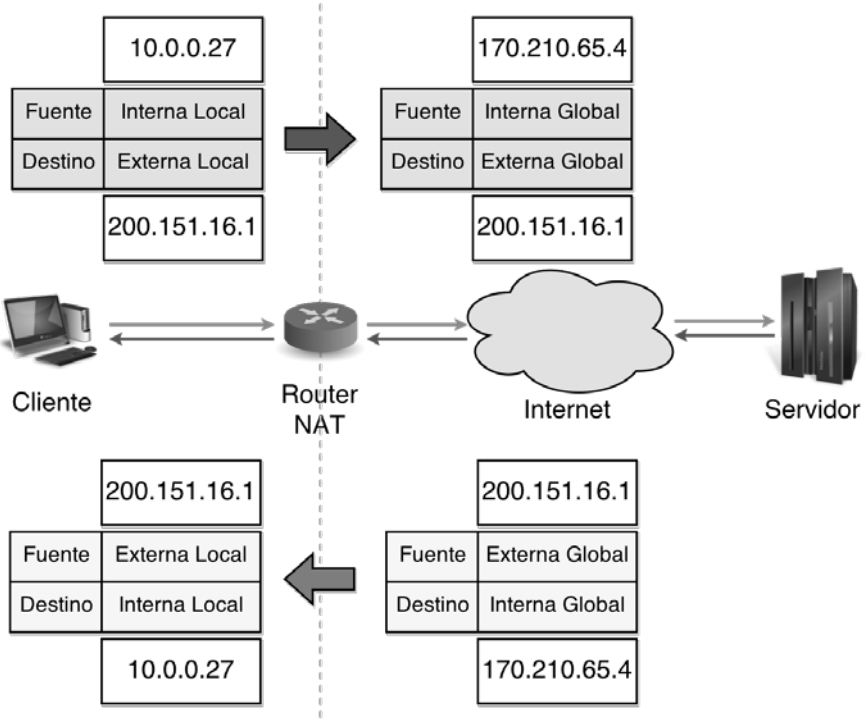


Figura 9.20 - Tipos de direcciones NAT.

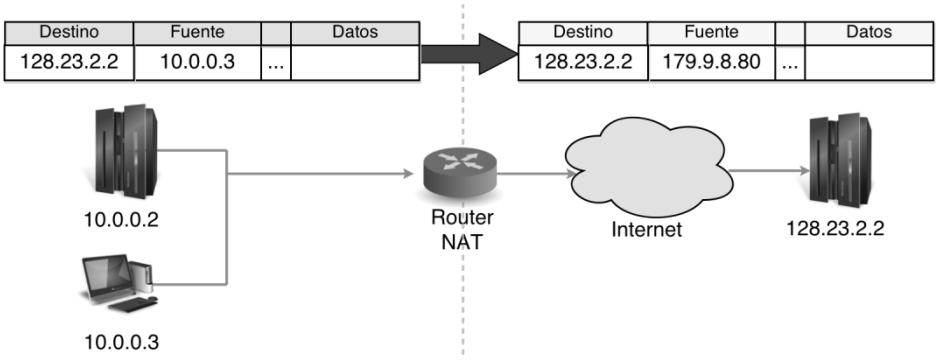


Tabla de Traducción NAT

Dirección IP Local Interna	Dirección IP Global Interna	Dirección IP Global Externa
10.0.0.3	179.9.8.80	128.23.2.2

Figura 9.21 - Traducción NAT.

Pueden existir también entradas dinámicas, generadas en los casos en que se asignan direcciones para la comunicación. En estos casos, la relación de

direcciones se genera a demanda y, luego de ser usadas, se descartan las entradas de la tabla que guardan la información. En el caso más común de este tipo de NAT, existe un grupo de direcciones internas globales a compartir entre un número mayor de dispositivos.

El mapeo estático es ideal para dispositivos que necesitan ser representados siempre con la misma IP pública. Este tipo de mapeo sirve para transacciones denominadas *inbound*, que se inician en la red pública y están dirigidas a un servidor de la red interna. La configuración, en estos casos, es manual y no se permite compartir estas direcciones IP entre dispositivos en la red interna.

A esta forma de comunicación se la conoce como el problema de la dirección escondida, ya que desde Internet se precisaría acceder a una máquina, por ejemplo un servidor, que se encuentra configurada con una dirección privada no enrutable. Por este motivo, el mapeo estático ofrece una solución que permite que la dirección global asignada sea públicamente conocida.

Por su parte, el mapeo dinámico es el que regularmente se utiliza en el caso de clientes en las máquinas de la red interna, que pueden hasta llegar a compartir una IP pública para acceder a otras máquinas externas. Su configuración es un poco más complicada, pero una vez realizada, es automática. En este caso, la direcciones externas asignadas a los dispositivos internos se realiza dinámicamente, por requerimiento y, cuando la sesión termina, el traductor NAT libera la relación establecida para que la dirección global pueda reciclarse para usos posteriores.

Es posible mezclar ambos tipos de mapeo en el mismo sistema, según el dispositivo interno en cuestión y teniendo en cuenta que no exista solapamiento entre ambos.

Una posibilidad más sofisticada de NAT, aún más amplia que el mapeo dinámico, consiste en el uso de puertos para multiplexado de las direcciones privadas sobre una única dirección IP pública.

Los números de puerto son esquemas de direccionamiento típicos a nivel TCP o UDP, para identificación de diferentes conexiones entre dos direcciones. Se trata de números de $16\ bits$ sin signo, es decir que existen $2^{16} = 65536$ puertos posibles, asignados por la aplicación emisora o receptora.

En términos generales, los números de puerto se pueden clasificar en dos categorías: bien conocidos y efímeros. Los puertos bien conocidos son asignados por el IANA y, tradicionalmente, comprenden el rango $0 - 1023$. Las aplicaciones que usan este tipo de puertos son ejecutadas como servidores y permanecen a la escucha de conexiones. Algunos ejemplos de puertos bien conocidos son los de los servidores para transferencia de archivos FTP (puerto 21), el de *login* remoto seguro SSH (puerto 22), el del servidor de correo SMTP (puerto 25) y el del servidor web HTTP (puerto 80). Los puertos efímeros, por su parte, son normalmente empleados por las aplicaciones de usuario, de forma temporal, cuando los clientes se conectan con los servidores, y comprenden el rango superior al número 1024.

El esquema NAT conocido como NAPT (Network Address Port Translation) considera el multiplexado por puertos para permitir simultaneidad de conexiones con una única dirección IP pública, sin interferencia entre ellas.

Dado que la combinación del par (dirección IP; número de puerto) fuente y destino, define unívocamente una conexión, NAT podría no solo cambiar la dirección IP de una conexión, sino también el puerto del encabezado UDP o TCP, siempre que se mantenga la información apropiada en un registro. Generalmente, el esquema se utiliza en el caso de varias direcciones internas locales que comparten una única dirección interna global, y es conocido también como NAT sobrecargado (Overloaded NAT).

En la Fig. 9.22 se presenta un ejemplo de NAPT, considerando que la dirección de salida es 170.210.65.2 puede ser compartida por varias máquinas simplemente realizando una traducción por puertos. Se observa que la solicitud interna desde el puerto 7000 de un cliente con dirección IP 10.0.0.2, se traduce a una solicitud con el mismo número de puerto, pero dirección IP pública 170.210.65.2. Cualquier otra solicitud, desde otro cliente interno, se podrá diferenciar de esta por el número de puerto, a pesar de usar la misma dirección IP de salida.

Por último, existe un problema para soporte transparente de muchas aplicaciones por parte de NAT. NAT impone restricciones en el desarrollo de aplicaciones que cargan direcciones IP, o derivadas, dentro del flujo de datos. Por este motivo, en la RFC 2993 se discuten varias implicaciones arquitecturales del uso de NAT, aceptando la utilización de traductores NAT en convivencia con Puertas a Nivel de Aplicación (ALG, Application Level Gateways) para cada aplicación afectada. También, la RFC 3235 desarrolla algunas recomendaciones de diseño para nuevos protocolos que deban atravesar NAT, de tal manera que el agregado de un ALG no sea engorroso, dificultando el despliegue de las mismas.

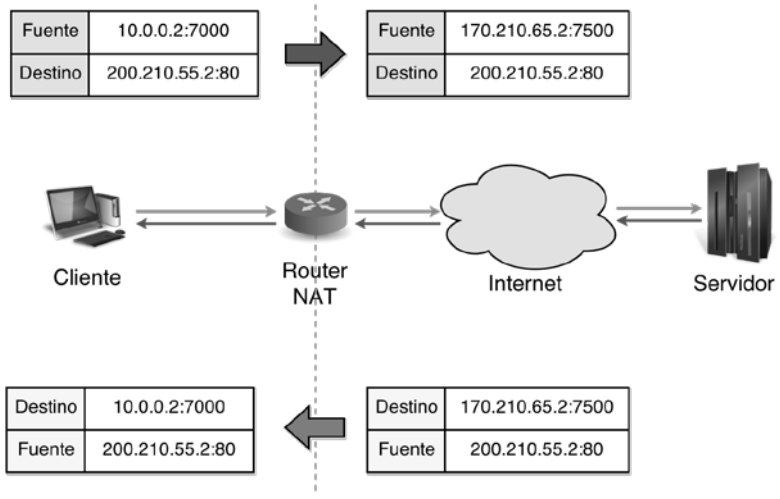


Figura 9.22 - NAPT.

9.10 Configuración Automática de Direcciones IP

Antes de comenzar a transmitir o recibir información, cada máquina en una red precisa tener asignada una dirección IP. Otros datos necesarios para la comunicación son la dirección IP del *router* de salida, la máscara de subred y la dirección del servidor de DNS.

La configuración automática de los dispositivos comenzó a presentarse como una verdadera necesidad a medida que las redes fueron creciendo en tamaño y variedad de dispositivos conectados.

Uno de los primeros problemas que enfrentó la configuración manual fue la administración de estaciones de trabajo sin disco, ya que la ausencia de hardware de almacenamiento sólo podía derivar en la presencia en la red de alguna entidad que otorgara a este tipo de máquinas los elementos necesarios para comenzar a comunicarse en el momento del arranque.

Al principio, esta funcionalidad se intentó cubrir con el protocolo inverso de ARP, denominado RARP (Reverse ARP), que funcionaba en el más bajo nivel dando respuesta al problema de la dirección IP, pero requería de un hardware capaz de realizar difusión y utilizaba una dirección MAC para identificación.

Debido a estas limitaciones surgió como alternativa el Protocolo de Arranque (BOOTP, BOOTstrap Protocol), definido en la RFC 951. El mismo permitía que una pila TCP/IP mínima, típicamente almacenada en una memoria ROM de una placa, pero sin información de configuración, obtuviera los datos necesarios para comenzar una comunicación.

BOOTP se basaba en un intercambio de mensajes del tipo cliente/servidor, encapsulados en UDP e IP, que servía para completar una primera fase del arranque de una máquina sin disco: la obtención de la dirección IP y otros parámetros de configuración. Una segunda fase, no definida en BOOTP, implicaba la descarga de información adicional necesaria para la comunicación. Generalmente se usaba el Protocolo Trivial de Transferencia de Archivos (TFTP, Trivial File Transfer Protocol) en esta segunda fase.

El primer desafío que debieron enfrentar los diseñadores de BOOTP se relacionaba con resolver la cuestión de cómo una computadora puede armar datagramas para comunicarse con un servidor si desconoce su propia dirección IP. La respuesta la encontraron en las direcciones IP especiales.

Por ejemplo, la dirección destino "255.255.255.255" de un datagrama se corresponde con un *broadcast* limitado, es decir un *router* no re-enviará este paquete fuera de su red. Esta dirección se traduce a la dirección MAC "$ff:ff:ff:ff:ff:ff$". Una aplicación cliente puede usar el *broadcast* limitado para forzar a IP a transmitir un datagrama por *broadcast* sobre la red local, antes de que el protocolo conozca su propia dirección IP en la red local o la dirección IP de la máquina servidora. En este caso, la dirección IP origen, aún desconocida, se puede rellenar con la dirección especial "0.0.0.0", siendo la dirección MAC origen la propia del cliente.

El segundo dilema que debieron resolver se refería a cómo generar una respuesta. El servidor conoce la dirección IP del solicitante, pues la tiene cargada en un archivo de configuración, con formato de pares (dirección MAC / dirección IP), pero no puede enviar la respuesta por *unicast* IP porque el cliente no se reconocería como destinatario. Además, si el dispositivo a configurar se

encuentra en una LAN, el servidor probablemente primero haría un requerimiento ARP preguntando por dicha dirección IP, pero ningún dispositivo le contestaría pues no reconocería esa dirección como propia.

De este modo, el servidor se encuentra limitado a responder según dos opciones: o bien edita un *broadcast* limitado, o puede tomar la información necesaria para cargar su caché ARP desde el contenido del propio mensaje BOOTP enviado por el cliente. La última opción implicaría cargar el caché ARP a partir de los datos proporcionados por una aplicación, pero no todas las implementaciones de los Sistemas Operativos permiten la carga de caché ARP de este modo ya que dicho caché se carga a partir de los mensajes ARP y en respuesta a requerimientos ARP. Debido a este inconveniente, la única opción válida resulta ser la primera, pero aún así no se soluciona el problema.

La respuesta se presenta en la Fig. 9.23. Curiosamente, la aplicación cliente levanta desde un puerto bien conocido, el puerto 68. El servidor, por su parte, atiende los requerimientos en el puerto 67.

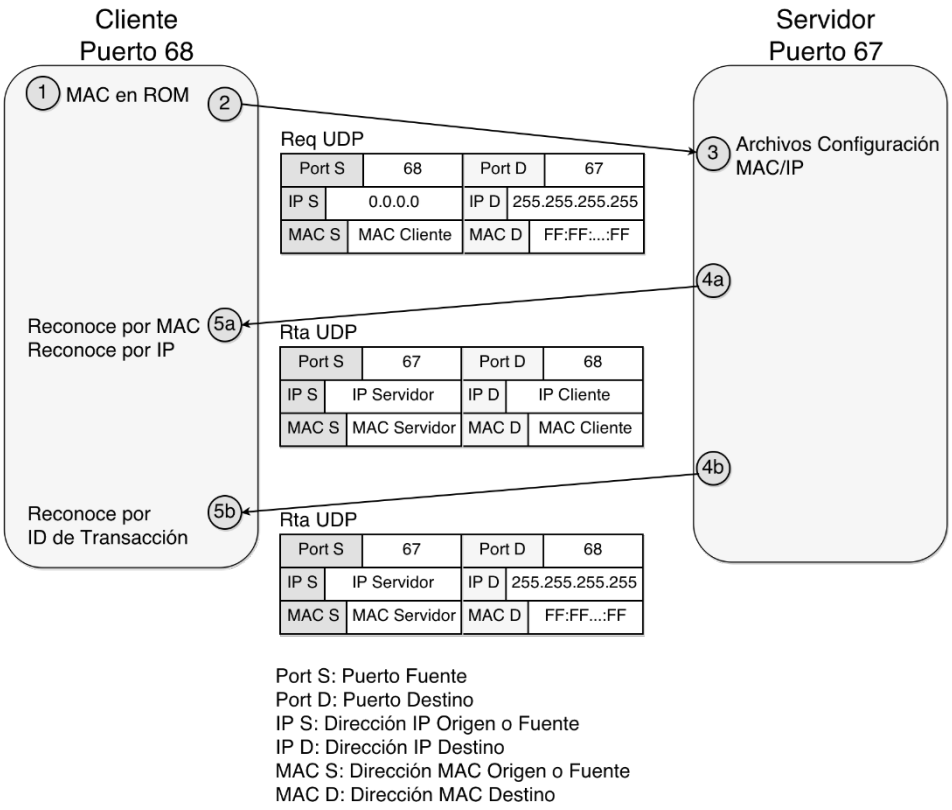


Figura 9.23 - Intercambio Cliente-Servidor BOOTP.

En referencia a la Fig. 9.23, la etiqueta 1 destaca el hecho de que el cliente conoce su propia dirección MAC. Para solicitar los datos para su configuración, edita un requerimiento BOOTP, que lleva la etiqueta 2 ,

encapsulado en UDP con puerto fuente "68" y puerto destino "67". Este datagrama UDP a su vez se encapsula en IP, con dirección fuente "0.0.0.0" y dirección destino "255.255.255.255". A nivel de enlace, la dirección MAC fuente es la del cliente, en tanto que la dirección MAC destino es la de *broadcast*.

La máquina servidora levanta este mensaje, tal como lo hace el resto de las máquinas de la red. También todas procesan este mensaje a nivel IP, pero sólo la que tiene funcionando la aplicación servidora es capaz de entregarlo al puerto 67, el resto lo descarta.

La aplicación servidora toma, del propio mensaje, el dato sobre la dirección MAC del cliente. Con esta información consulta su tabla de configuración, destacada con la etiqueta 3, que posee los pares (dirección MAC / dirección IP). Si el servidor es capaz de llenar el caché ARP a partir del mensaje, podrá editar una respuesta tal como la que se indica con la etiqueta 4a en la Fig. 9.23, del tipo *unicast*. La máquina cliente levantará la respuesta porque la dirección MAC destino es la propia. El problema es que no se puede anticipar cómo reaccionará a nivel IP puesto que desconoce su propia dirección a ese nivel.

La respuesta etiquetada como 4b en la Fig. 9.23 parecería ser la menos problemática: se edita por *broadcast* para asegurarse que la máquina cliente la levante a nivel IP. El encabezado IP llevará escrito el valor correspondiente de UDP en el campo Protocolo. El encabezado UDP le señalará el puerto "68", que el cliente deberá estar escuchando. Por último, dentro del propio mensaje BOOTP, un campo Identificador de Transacciones le permitirá reconocer al cliente que ésta es la respuesta a su pregunta.

El hecho de que la respuesta sea editada por *broadcast* impone una restricción adicional al número de puerto en el que el cliente espera la respuesta. Si el cliente estuviera escuchando en un puerto efímero, por encima de 1023, otro cliente que estuviese trabajando en el mismo puerto tomaría esta respuesta como propia, siendo recibido este mensaje por otras aplicaciones. Por este motivo, se elige que el cliente levante la comunicación desde un puerto bien conocido, diferente del puerto del servidor, ya que si el cliente usara el puerto "67", todos los servidores de la red estarían tomando las respuestas *broadcast*, verían que es una respuesta y volverían a esperar un requerimiento, resultando un modo de funcionamiento muy ineficiente.

Por otra parte, si muchos clientes arrancaran al mismo tiempo, cada uno vería las respuestas de los demás. Por este motivo es necesario agregar el Identificador de Transacción y la presencia de la dirección de hardware del cliente en el propio mensaje BOOTP.

Como se ha señalado, los requerimientos y respuestas BOOTP van encapsulados en datagramas UDP, protocolo no orientado a la conexión, no confiable. Debido a esto, la responsabilidad de una comunicación confiable queda en manos de la aplicación. Se utiliza el Control de Errores de UDP para detección de errores y se fija en el datagrama IP el bit DF en "1" para poder satisfacer requerimientos del clientes con muy poca memoria y, por tanto, sin capacidad de re-ensamble. También se permiten duplicados en las respuestas, ya que BOOTP está diseñado para aceptar y procesar la primera de todas las respuestas. Respecto de la pérdida de datagramas, se utiliza un esquema de *timeout* y retransmisión: al

editar el requerimiento, el cliente lanza un reloj, si éste expira sin haber recibido una respuesta, se retransmite el requerimiento.

Para evitar que, luego de una falla general en la que todas las máquinas vuelvan a arrancar al mismo tiempo, hagan sus requerimientos al servidor simultáneamente y éste se sobrecargue, se utiliza un retardo aleatorio para el *timeout*, cuyo valor inicial se encuentra en un rango de 0 a 4 seg, y se duplica en cada retransmisión hasta alcanzar los 60 seg, cuando se reinicializa.

La Fig. 9.24 presenta el formato de un mensaje BOOTP. El campo Código de Operación se escribe con el número "1" para el requerimiento y "2" para la respuesta. El campo Tipo de Hardware se carga con un código referente a *Ethernet* y el de Longitud de Hardware con el número "6", indicando que la dirección de hardware cuenta con seis bytes de longitud. El campo Cuenta de Saltos es ajustado a "0" en el requerimiento del cliente y aumentado cuando un servidor del tipo *proxy*, funcionando como agente de retransmisión, lo re-envía fuera de la red local, ya que BOOTP permite que un servidor se ubique fuera de la red si el *router* puede instalar un agente de retransmisión. El agente escuchará requerimientos en el puerto 67, los retransmitirá por *unicast* al servidor instalado en otra red, copiando su propia dirección IP en el campo Gateway del mensaje BOOTP, y sumando uno a la cuenta de saltos. La respuesta del verdadero servidor será dirigida al agente de retransmisión, que luego deberá entregarlo al cliente.

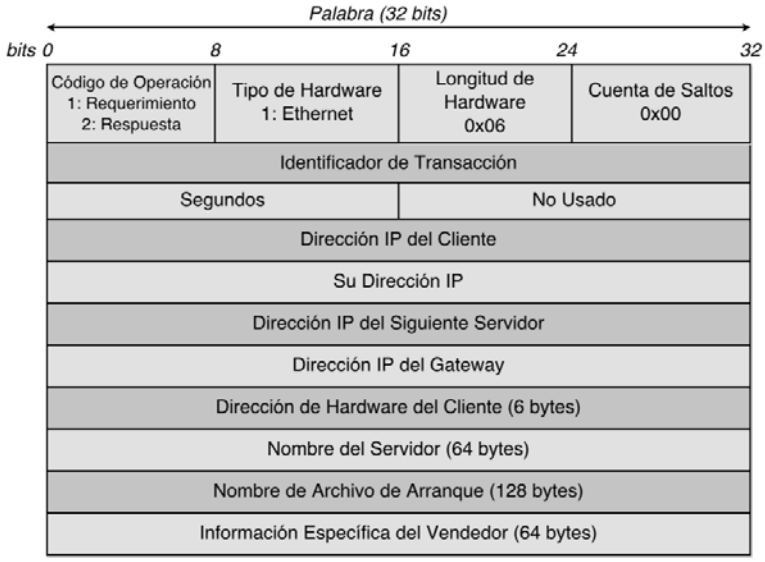


Figura 9.24 - Mensaje BOOTP.

El Identificador de Transacción es un campo de 32 $bits$ que se crea en el requerimiento como un número aleatorio y que luego se repite en la respuesta. En el campo Segundos el cliente coloca la cantidad de segundos transcurridos desde el arranque. Se supone que este campo le sirve al servidor para dar prioridad a las respuestas. En el protocolo original, el campo que sigue era Reservado, pero la

RFC 1542 incluyó en este campo una bandera de *broadcast*, para que el cliente indique al servidor que desea recibir este tipo de respuestas.

Normalmente, el campo Dirección IP del Cliente se rellena con el valor "0.0.0.0" en el requerimiento, pero puede suceder que un cliente que ya posea una dirección IP, la escriba en este campo para recibir del servidor respuestas *unicast*. En el campo Su Dirección IP el servidor escribe la dirección que le otorga al cliente. El campo Dirección IP del Siguiente Servidor indica la dirección del servidor al que se debe recurrir en la segunda fase de la configuración, por ejemplo para obtener una imagen del Sistema Operativo, en caso de máquinas sin disco rígido. En el campo Dirección IP del Gateway se incluye la dirección del servidor *proxy* cuando se re-envían los mensajes a servidores remotos. En el campo Dirección de Hardware del Cliente se escribe la dirección MAC, a modo de identificación del cliente.

En el campo Nombre del Servidor se escribe el nombre del servidor donde el cliente puede ir a buscar información adicional para completar su configuración. Esta información se complementa con el campo Nombre de Archivo de Arranque, que contiene el directorio y nombre del archivo de arranque que el cliente puede usar para cargar una imagen del Sistema Operativo. Ambos campos tienen una longitud máxima, de 64 y 128 $bytes$ respectivamente.

El último campo del mensaje se denomina Información Específica del Vendedor, tiene una longitud fija de 64 $bytes$ y, originalmente, fue agregado para que diferentes fabricantes pudieran adaptar sus necesidades para distintos tipos de hardware. El cliente puede usar este campo para solicitar cierto tipo de información en el requerimiento. Por su parte, el servidor puede incluir parámetros que le interesa que el cliente conozca. Para poder establecer un lenguaje común, el campo comienza con un número de 4 $bytes$ que indica el comienzo del campo y que se conoce como *magic cookie*: "99.130.83.99". A continuación, cada información específica se codificará en el formato TLV (Tipo/Longitud/Valor). Algunos de los datos que se pueden enviar en este campo son la dirección del *router* por default, la dirección del servidor DNS local, la MTU del enlace y la máscara de subred.

Uno de los problemas que presentaba BOOTP era su ausencia de adaptación a situaciones dinámicas. Por ejemplo, el protocolo no es capaz de solucionar aspectos relacionados con el movimiento de computadoras entre diferentes redes, o adaptarse al caso de una red con un número de máquinas superior a la cantidad de direcciones IP. La principal causa de estas falencias, se debía a que la información de configuración quedaba totalmente en manos del administrador, al igual que cualquiera de los cambios que se pudiesen producir. En este sentido, se trataba de un protocolo de mapeo estático.

La solución a estos inconvenientes la presentó el nuevo Protocolo de Configuración Dinámica de Hosts (DHCP, Dinamic Host Configuration Protocol), definido en la RFC 2131. Este protocolo permite a una máquina adquirir toda la información de configuración en un único mensaje pero, a diferencia de BOOTP, la adquisición de direcciones IP es dinámica. Por este motivo, DHCP reemplazó a BOOTP, convirtiéndose en el protocolo de mayor utilización para configuración automática actual.

A pesar de las diferencias, DHCP se diseñó en base a BOOTP, conservando muchas características del viejo protocolo. Por ejemplo, en DHCP se mantuvo la funcionalidad de agentes de retransmisión y el campo de Opciones de los mensajes DHCP es prácticamente como el campo de Información Específica de Vendedor de BOOTP. EN DHCP no se usan tablas estáticas para mapeo de direcciones de hardware a direcciones IP. En su lugar, se usan conjuntos de direcciones IP, denominados *pool*, que permiten realizar asignación dinámica. La operación de intercambio de mensajes es enteramente similar a la de BOOTP, aunque se crearon nuevos mensajes para poder cumplir la funcionalidad mencionada.

Por flexibilidad, el mecanismo de asignación de direcciones de DHCP permite tres escenarios posibles:

- **Asignación Manual**: es equivalente a BOOTP, donde el administrador asigna una dirección IP particular a un dispositivo. La asignación se comunica mediante DHCP. Tiene sentido en casos de dispositivos tales como servidores o *routers*.

- **Asignación Dinámica**: la asignación de una dirección es transitoria, seleccionándola a partir de un *pool* de direcciones. El servidor guarda un registro de las direcciones asignadas. Se dice que el cliente alquila una dirección del *pool* por un tiempo determinado cuyo valor es decisión del administrador del servidor. Cuando se vence el plazo, el cliente puede pedir permiso para seguir usando la misma dirección o solicitar una nueva.

- **Asignación Automática**: se asigna una dirección de manera permanente, seleccionándola a partir de un *pool* de direcciones. Es un caso muy especial de asignación dinámica, con tiempo de alquiler infinito. Generalmente, se prefiere la asignación manual.

La asignación dinámica es el motivo de la popularidad del protocolo. En este caso, el administrador debe configurar apropiadamente el servidor en cuanto a las licencias de alquiler, siendo el tiempo en que se otorgarán dichas licencias una de las cuestiones más importantes a resolver, ya que los tiempos de alquiler son dependientes del entorno. Por ejemplo, en un laboratorio para alumnos podría ser de 1 hora, mientras que en una red corporativa los tiempos ascenderían a 1 día. Para poder acomodar tan variados escenarios, el protocolo permite al cliente requerir un período específico y al servidor informar el período que garantiza. El máximo sería infinito, asociándose éste el caso estático de BOOTP.

Cada vez que una máquina cliente arranca, debe comunicarse con un servidor DHCP para alquilar una dirección. Si la asignación es dinámica, durante su funcionamiento normal, debe realizar otras actividades relacionadas con la administración del alquiler, ya que éste se asocia a un límite en el tiempo. Finalizado este tiempo, puede suceder que re-negocie el alquiler de la misma dirección o que el servidor le otorgue una nueva. Para comprender mejor el

funcionamiento, en la Fig. 9.25 se presenta el Diagrama de Estados del funcionamiento de un cliente DHCP.

En el estado de Inicialización, el cliente comienza el proceso de adquisición de una licencia. Lo primero que hace es editar un mensaje DHCPDISCOVER, que se envía por *broadcast* al puerto 67, con el propósito de encontrar un servidor. La transmisión de este mensaje lo coloca en el estado Selección, en el cual el cliente espera recibir mensajes DHCPOFFER por parte de algún servidor, que contenga la dirección IP ofrecida y el tiempo de alquiler de la misma. Una vez recibido los ofrecimientos, el cliente elige uno y transmite un mensaje DHCPREQUEST por *broadcast*, para avisar a los servidores cuál ha sido su elección. Esta transmisión lo coloca en el estado Requerimiento, donde espera recibir una respuesta del servidor seleccionado.

Normalmente, en el estado Requerimiento, el cliente recibirá un mensaje DHCPACK del servidor seleccionado, confirmándole el alquiler. Entonces deberá almacenar un registro del tiempo que se le ha destinado para el alquiler y podrá pasar al estado Bound, el estado normal de operación.

El cliente abandona el estado Bound cuando haya transcurrido la mitad del tiempo de alquiler, momento en el que edita otro mensaje DHCPREQUEST, por *unicast*, dirigido al servidor que le ha otorgado la dirección IP, solicitando renovarla. Esta situación lo hace mover al estado Renovación. También podría suceder que el cliente decida terminar la asociación con esa dirección, entonces editaría un mensaje DHCPRELEASE, liberando el uso de la dirección IP y retornado al estado Inicialización.

Para renovar la licencia, el cliente en el estado Renovación, debería recibir un mensaje DHCPACK, que lo regresa al estado Bound, con el tiempo de alquiler renovado. De recibir un mensaje DHCPNAK, debe volver obligatoriamente al estado de Inicialización. También podría suceder que no recibiera respuesta alguna por parte del servidor. En este caso, al expirar el 87.5% del tiempo de alquiler, debe transmitir un mensaje DHCPREQUEST por *broadcast*, pasando al estado Rebind. En este último estado, si recibe una respuesta positiva, regresa al estado Bound, sino regresa al estado de Inicial.

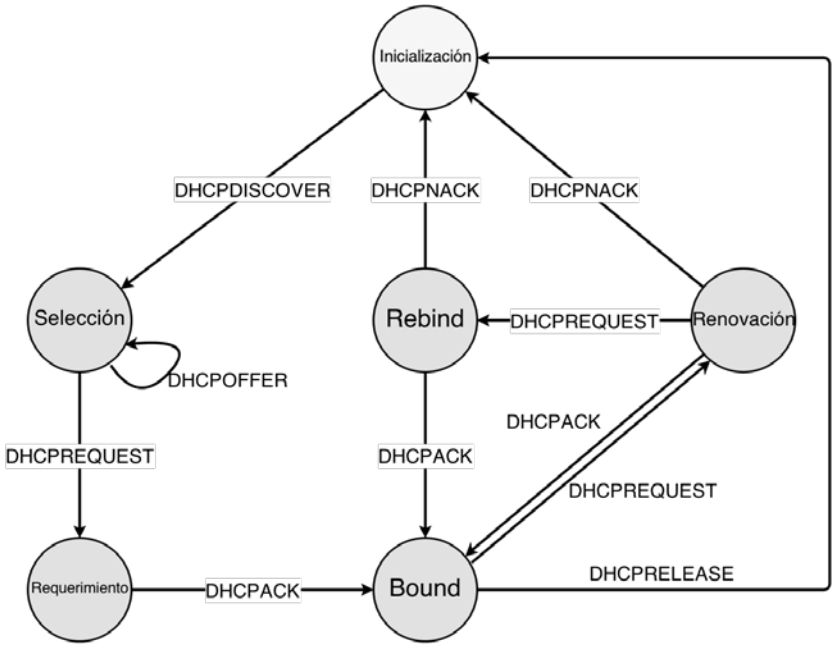


Figura 9.25 – Diagrama de Estados del cliente DHCP.

A continuación, se presenta un mensaje de ofrecimiento en el que el servidor le otorga una dirección IP 192.168.0.59 al cliente, durante 10 *minutos*. Transcurrido la mitad de este tiempo, se produce una re-negociación, que el servidor acepta, re-asignando la dirección por otros 10 *minutos*.

DHCP OFFER

Ethernet II

Destination: ff:ff:ff:ff:ff:ff // Source: 00:e0:7d:86:68:6c // Type: IP (0x0800)

Internet Protocol

Version: 4 // Header length: 20 bytes // Differentiated Services Field: 0x00 // Total Length: 284 // Identification: 0xb386 // Flags: 0x00 // Fragment offset: 0 // Time to live: 64 // Protocol: UDP (0x11) // Header checksum: 0x05a2 (correct) // Source: 192.168.0.1 // Destination Address: 255.255.255.255

User Datagram Protocol Source port: bootp (67) // Destination port: (68) // Length: 264 // Checksum: 0x3e0f (correct)

Bootstrap Protocol

Message type: Boot Reply (2) // Hardware type: Ethernet // Hardware address length: 6 // Hops: 0 // Transaction ID: 0x90174b06 // Seconds elapsed: 0 // Bootp flags: 0x0000 (Unicast) // Client IP address: 0.0.0.0 // Your (client) IP address: 192.168.0.59 // Next server IP address: 0.0.0.0 // Relay agent IP address: 0.0.0.0 // Client hardware address: 00:a0:4b:02:f4:09 // Server host name not given // Boot file name not given // Magic cookie: (OK) // Option 51: IP Address Lease Time = 10 minutes // Option 54: Server Identifier = 192.168.0.1 // Option 53: DHCP Message Type = DHCP Offer // End Option

DHCP REQUEST (RENEGOCIACIÓN)

Ethernet II

Destination: 00:e0:7d:86:68:6c (router) // Source: 00:a0:4b:02:f4:09 (TflLan_02:f4:09) // Type: IP (0x0800)

Internet Protocol
Version: 4 // Header length: 20 bytes // Differentiated Services Field: 0x00 // Total Length: 328 // Identification: 0x4100 (16640) // Flags: 0x00 // Fragment offset: 0 // Time to live: 128 // Protocol: UDP (0x11) // Header checksum: 0x7718 (correct) // Source: 192.168.0.59 // Destination: router 192.168.0.1
User Datagram Protocol
Source port: 68 (68) // Destination port: bootp (67) // Length: 308 // Checksum: 0xe888 (correct)
Bootstrap Protocol
Message type: Boot Request (1) // Hardware type: Ethernet // Hardware address length: 6 // Hops: 0 // Transaction ID: 0x8808ed11 // Seconds elapsed: 0 // Bootp flags: 0x0000 (Unicast) // Client IP address: 192.168.0.59 // Your (client) IP address: 0.0.0.0 // Next server IP address: 0.0.0.0 // Relay agent IP address: 0.0.0.0 // Client hardware address: 00:a0:4b:02:f4:09 // Server host name not given // Boot file name not given // Magic cookie: (OK) // Option 53: DHCP Message Type = DHCP Request // Option 61: Client identifier Hardware type: Ethernet Client hardware address: 00:a0:4b:02:f4:09 // Option 12: Host Name = "LABcapM13" // Option 81: Client Fully Qualified Domain Name (13 bytes) // Option 60: Vendor class identifier = "MSFT 98" // Option 55: Parameter Request List 1 = Subnet Mask 15 = Domain Name 3 = Router 6 = Domain Name Server 44 = NetBIOS over TCP/IP Name Server 46 = NetBIOS over TCP/IP Node Type 47 = NetBIOS over TCP/IP Scope 43 = Vendor-Specific Information 77 = User Class Information // End Option

DHCP ACK
Ethernet II
Destination: 00:a0:4b:02:f4:09 (TflLan_02:f4:09) // Source: 00:e0:7d:86:68:6c (router) // Type: IP (0x0800)
Internet Protocol
Version: 4 // Header length: 20 bytes // Differentiated Services Field: 0x00 // Total Length: 302 // Identification: 0xc981 (51585) // Flags: 0x00 // Fragment offset: 0 // Time to live: 64 // Protocol: UDP (0x11) // Header checksum: 0x2eb1 (correct) // Source: router (192.168.0.1) // Destination Address: 192.168.0.59
User Datagram Protocol
Source port: bootp (67) // Source or Destination Port: 68 // Length: 282 // Checksum: 0x294f (correct)
Bootstrap Protocol
Message type: Boot Reply (2) // Hardware type: Ethernet // Hardware address length: 6 // Hops: 0 // Transaction ID: 0x8808ed11 // Seconds elapsed: 0 // Bootp flags: 0x0000 (Unicast) // Client IP address: 192.168.0.59 // Your (client) IP address: 192.168.0.59 // Next server IP address: 0.0.0.0 // Relay agent IP address: 0.0.0.0 // Client hardware address: 00:a0:4b:02:f4:09 // Server host name not given // Boot file name not given // Magic cookie: (OK) // Option 1: Subnet Mask = 255.255.255.0 // Option 51: IP Address Lease Time = 10 minutes // Option 54: Server Identifier = 192.168.0.1 // Option 53: DHCP Message Type = DHCP ACK // Option 3: Router = 192.168.0.1 // Option 6: Domain Name Server = 200.63.65.4 // End Option

En la actualidad, un servidor DHCP no precisa ser instalado ya que generalmente los *routers* lo incluyen como parte del propio software. En cuanto a los clientes, casi todos los Sistemas Operativos modernos los integran.

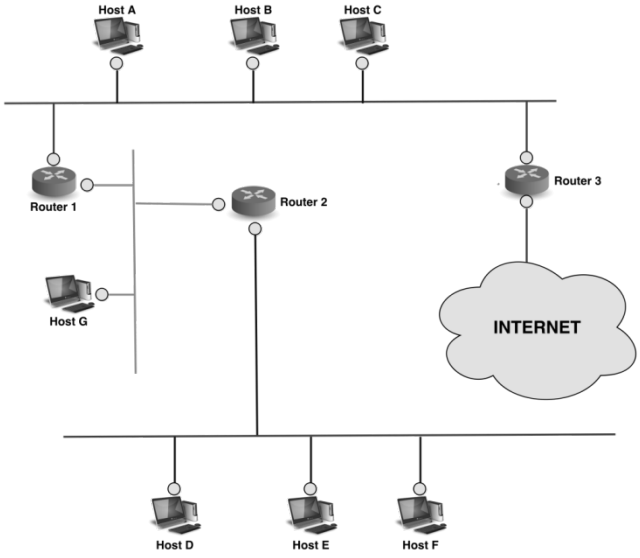
Bibliografía

1. RFC 791 “Internet Protocol”, September 1981. http://tools.ietf.org/html/rfc791
2. RFC 792 “Internet Control Message Protocol”, September 1981. https://www.ietf.org/rfc/rfc792.txt
3. RFC 826 “An Ethernet Address Resolution Protocol”, November 1982. http://tools.ietf.org/html/rfc826
4. RFC 950 “Internet Standard Subnetting Procedure”, August 1985. https://www.ietf.org/rfc/rfc950.txt
5. RFC 951 “Bootstrap Protocol”, September 1985. http://tools.ietf.org/html/rfc951
6. RFC 1058 “Routing Information Protocol”, June 1988. http://tools.ietf.org/html/rfc1058
7. RFC 1256 “ICMP Router Discovery Messages”, September 1991. https://www.ietf.org/rfc/rfc1256.txt
8. RFC 1393 “Traceroute Using an IP Option”, January 1993. http://tools.ietf.org/html/rfc1393
9. RFC 1517 “Applicability Statement for the Implementation of Classless Inter-Domain Routing (CIDR) ”, September 1993. https://www.ietf.org/rfc/rfc1517.txt
10. RFC 2131 “Dynamic Host Configuration Protocol”, March 1997. https://www.ietf.org/rfc/rfc2131.txt
11. RFC 2663 “IP Network Address Translator (NAT) Terminology and Considerations”, August 1999. http://tools.ietf.org/html/rfc2663
12. Comer, Douglas, “Internetworking with TCP/IP: Principles, Protocols and Architecture v. 1”. Pearson Education, 1995.
13. Stevens, W. Richard, “TCP/IP Illustrated, Vol. 1: The Protocols (Addison-Wesley Professional Computing Series) ”. Addison-Wesley, 1993.
14. Kozierok, Charles M., “The TCP/IP Guide”. http://www.tcpipguide.com/free/t_toc.htm

Problemas

1. Dada la dirección IP con los bits de prefijo indicados: 132.25.69.0/20, determinar si es una dirección de host o subred. En ambos casos identifíquelo. Cualquiera sea el caso determine la dirección de *broadcast* de la subred asociada.

2. Dado el siguiente esquema de red, asigne direcciones IP a cada máquina y desarrolle las tablas de enrutamiento en los *routers*. Dispone del siguiente bloque de direcciones: 170.210.62.0. Verificar que la máxima cantidad de hosts podrá llegar hasta 20 en cada subred. Utilizar como dirección de salida 170.210.63.65.

3. 

4. En el esquema del problema anterior ejemplifique el intercambio de paquetes ARP para comunicación entre dos *hosts* de una misma LAN y entre un *host* de una LAN y otro *host* de una LAN adyacente

5. La longitud del campo de datos asociados con el requerimiento de un usuario es de $1200\ bytes$. Suponiendo que los datos se enviarán a otro host de la misma red y que dicha red posee una MTU de $512\ bytes$, explique la cantidad de datagramas IP necesarios para enviar los datos si el header asociado a los mismos es de $20\ bytes$. Determine el contenido de los campos más representativos de la funcionalidad de fragmentación.

6. Explique las diferencias entre VLSM, subnetting, CIDR y NAT, aclarando la utilidad de cada caso.

CAPÍTULO X

Protocolo IPv6

IPv4 fue ideado cuando la red mundial, que hoy conocemos como Internet, era apenas una red experimental. A pesar de ello, el protocolo ha superado las expectativas originales, adaptándose a los cambios durante más de treinta años. En todo ese tiempo, tanto la aparición de nuevas aplicaciones como la conexión masiva de dispositivos, evidenciaron algunas limitaciones de la versión 4 del protocolo IP.

El problema más relevante de IPv4 se tradujo en un vaciamiento de su espacio de direcciones. A pesar de ello, se realizaron muchos esfuerzos para retrasar la fecha en que finalmente IPv4 se quedaría sin direcciones.

La técnica de subnetting ofreció una facilidad para que los administradores no tuvieran que solicitar una dirección de red diferente por cada segmento de red agregado a la Internet. Esta facilidad impuso a los routers la condición de reconocer máscaras de red diferentes a las de las clases originales.

El problema de subnetting era que dividía el espacio en rangos con la misma capacidad de dispositivos. En respuesta a esta limitación, surgió la posibilidad de usar máscaras de longitud variable. En este sentido, VLSM agregó flexibilidad a la estructura propuesta por subnetting, aunque sumó algunas dificultades a la hora de configurar las direcciones.

El inminente vaciamiento de las direcciones clase B y el crecimiento exponencial de las líneas de las Tablas de Ruteo, fue abordado a través de la propuesta CIDR. La nueva estructura propuso abandonar el sistema de clases y usar una nomenclatura de dirección y máscara. Apartarse de la estructura original permitió obtener una forma más ajustada de asignación de direcciones. Para poder enfrentar el consecuente crecimiento de las Tablas de Ruteo, se planteó que la asignación de la nueva propuesta de direcciones sin clases tuviera significado topológico, lo que permitió realizar agregado de rutas.

Por su parte, NAT introdujo un mecanismo muy exitoso para que varios dispositivos pudieran compartir una misma dirección IP.

A pesar de todas las soluciones ofrecidas, el esquema de direcciones IPv4 se agotó en febrero de 2011. Desde muchos años antes, un grupo de trabajo de la IETF venía investigando sobre un nuevo protocolo de red diseñado especialmente para la Internet moderna, que fue plasmado en diversas RFC y conocido con el nombre de IPv6. Sus diferencias con IPv4 son muy grandes, ya

que el nuevo protocolo no sólo aborda el problema de un espacio de direcciones expandido. En este sentido, el principal desafío será el de la migración, pasando por una etapa intermedia en la que dos protocolos que son incompatibles puedan coexistir, permitiendo que los servicios de IPv6 sean accesibles con clientes desde plataformas IPv4 y al revés.

10.1 Motivaciones

Considerando que el protocolo original IP fue desarrollado en el año 1981, pensado para una red de tipo experimental, sorprende que su uso haya persistido hasta la actualidad, aún ante el crecimiento exponencial de Internet de los últimos años. Hemos visto que una de las desventajas más importantes que tuvo IP en su versión cuatro fue el esquema tan rígido de direccionamiento, lo que generó un problema de vaciamiento del espacio de direcciones, con sus consecuentes intentos de solución.

Se debe tener en cuenta que los mecanismos creados para solucionar el problema del vaciamiento, se pensaron para mantener la compatibilidad con IPv4, para asegurar la continuidad de su funcionamiento.

La motivación real para el diseño de un nuevo protocolo IP se encuentra no sólo en el problema del vaciamiento de direcciones, sino también en el soporte de nuevas aplicaciones y los problemas de seguridad presentados en IPv4. El problema de vaciamiento, por sí solo, tiene graves consecuencias, en cuanto a la limitación que impone sobre el crecimiento de la propia Internet, la restricción que implica con respecto al ingreso de nuevos usuarios, los problemas surgidos de la ineficacia de los esquemas de ruteo más utilizados y la obligación casi impuesta del uso de NAT como método paliativo de uso extendido.

Por su parte, el desarrollo de nuevas aplicaciones exige tiempos de respuesta apropiados por parte de la red y su esquema de transporte de información, lo que deriva en la posibilidad de obtener mayor ancho de banda. No menos importante son los nuevos escenarios que estas aplicaciones plantean y que, a menudo, derivan en problemas de seguridad importantes.

En respuesta a este último desafío, se produjo la aparición de nuevos protocolos para mejora de la seguridad en las comunicaciones, como por ejemplo el Nivel de Protección Seguro (SSL, Secure Socket Layer) y el Protocolo Seguro de Transferencia de Hipertexto (HTTPS, Hypertext Transfer Protocol Secure). A pesar de este avance, se debe destacar que no se ha estandarizado ninguna solución.

Por los motivos expuestos, se propuso el diseño de un nuevo protocolo de red para Internet, que derivó en una versión conocida como IPv6 o IPng (IP New Generation). IPv6 representa el mayor de los cambios estructurales al protocolo de Internet desde 1981, año en que se publicó la RFC descriptora de IPv4.

El nuevo protocolo fue definido en la RFC 2460, publicada en 1998, aunque casi paralelamente se definieron nuevas versiones de otros protocolos para soporte de IPv6. Por ejemplo, la RFC 2461 describe el Protocolo de

Descubrimiento de Vecinos IPv6 (NDP, Neighbor Discovery Protocol) y la RFC 2463 define ICMP versión 6 (ICMPv6). El esquema de direcciones fue mejor explicado en la RFC 2373 (IP Version 6 Addressing Architecture) y en la RFC 2374 (An IPv6 Aggregatable Global Unicast Address Format), ambas del año 1998. Posteriormente, el esquema propuesto inicialmente fue muy discutido con respecto a su significado conceptual, dando lugar a la aparición de nuevas definiciones, que se publicaron en la RFC 3513 (Internet Protocol Version 6 (IPv6) Addressing Architecture) y la RFC 3587 (IPv6 Global Unicast Address Format), en el año 2003.

Algunos de los principales objetivos de diseño del nuevo protocolo no tuvieron que ver con la expansión del esquema de direcciones, sino que actualizaron la concepción del viejo protocolo en muchos aspectos.

10.2 Despliegue IPv6

Si bien es cierto que la expansión del espacio de direcciones fue muy importante, pasando de utilizar un espacio de $32\ bits$ (10^9 direcciones) a otro de $128\ bits$ ($3.4x10^{38}$ direcciones), también se dio importancia a la forma de dividir dicho espacio para el mejor aprovechamiento y gestión del mismo. En este sentido, IPv6 anula la necesidad de usar NAT, facilita las tareas de configuración y re-numeración de direcciones, mejora la eficiencia del ruteo actual haciéndolo más flexible ante futuras posibilidades, incorpora la comunicación *multicasting*, que antes era opcional, provee mejor soporte en cuanto la seguridad y calidad de servicio y considera la posibilidad de movilidad desde el punto de vista de IP.

A pesar de tantos cambios, la filosofía de diseño de IPv6 todavía mantiene mucho de su antecesor IPv4, ya que la idea que lo originó era más cercana a la de una actualización que a la de un reemplazo.

La implementación de IPv6 comenzó con el desarrollo de redes experimentales para pruebas de la operación del protocolo. Luego, en 1996, estas redes se conectaron en una inter-red, también de carácter experimental, conocida como 6BONE. El gran problema de la migración es que los esquemas de direccionamiento de IPv4 e IPv6 no son compatibles, como tampoco lo es el formato de los paquetes. En este sentido, la transición exige, de todas maneras, alguna forma de inter-operatividad.

La IETF ha trabajado mucho en cuestiones específicas para asegurar una transición no traumática entre versiones del protocolo. Los métodos más importantes que consideran una compatibilidad hacia atrás, permitiendo la convivencia de ambos protocolos son:

- **Dispositivos Doble Pila o Dual Stack**: se trata de *routers* que se pueden programar para funcionar tanto en IPv4 como en IPv6, permitiendo de este modo la comunicación con ambos tipos de *hosts*.
- **Traductores IPv4/IPv6**: se trata de dispositivos *dual stack* que pueden aceptar requerimientos de *hosts* IPv6, convertirlos a paquetes tipo IPv4,

transmitirlos a destinos IPv4 y luego manejar de manera inversa las respuestas.

- **Túnel IPv4 de IPv6**: en el caso de dispositivos IPv6 que están separados por *routers* IPv4, se propone una solución que consiste en el encapsulado de los datagramas IPv6 dentro de paquetes IPv4, para que estos últimos puedan ser interpretados por *routers* convencionales.

En la actualidad, muchos ISP, fabricantes de equipamiento de red y compañías web han adherido al nuevo protocolo, habilitando IPv6 en sus productos y servicios. El lanzamiento formal de IPv6, en junio de 2012, duplicó el uso global del protocolo, cuestión que se repitió al año siguiente. De continuar esta tendencia, se espera que, hacia el año 2020, más de la mitad de los usuarios de Internet se conecten a la gran red mediante el protocolo IPv6.

10.3 Direcciones IPv6

El cambio en la estructura de direcciones obedeció no sólo al problema de vaciamiento, sino también al objetivo de flexibilizar el uso de las mismas, para su adaptación a las redes actuales y su ajuste con respecto a futuras expansiones.

A pesar de los cambios, se conservaron algunos aspectos del esquema original de IPv4. Por ejemplo, se sostuvo la identificación de red y el esquema de ruteo, manteniendo la existencia de direcciones públicas y privadas, aunque con ciertas diferencias. También se mantuvo el significado global de las direcciones de red, que se continúan distinguiendo de las direcciones físicas, asignándose una por cada interfaz de acceso a una red.

Una diferencia importante es que en IPv6 no existen las direcciones de *broadcast*, ya que se incorpora un mayor protagonismo de las direcciones *multicast*. Así, la comunicación tipo *broadcast* se implementa mediante *multicast*, que deja de ser una funcionalidad opcional.

También se define un nuevo tipo de direcciones, conocidas como *anycast*. Se trata de direcciones asignadas a partir del espacio global direcciones *unicast* para que varios dispositivos compartan una misma dirección IP. La utilidad de las direcciones *anycast* se presenta en circunstancias de comunicación en grupo, cuando se desea transmitir algo, pero no necesariamente a todos los miembros. En *anycast*, la asociación de una dirección destino a varias máquinas, permite que se seleccione una de estas máquinas para ser la destinataria de la información. La información se encamina al mejor destino desde el punto de vista de la topología de la red y del protocolo de ruteo. Es decir que un paquete enviado a una dirección *anycast*, es entregado a la máquina más próxima. Uno de los primeros usos que recibió este tipo de direccionamiento en Internet fue para permitir la réplica de servidores DNS sobre las mismas direcciones IP, en respuesta a ataques distribuidos de denegación de servicio ocurridos sobre servidores *root* DNS.

Elegir la longitud apropiada para el nuevo espacio de direcciones fue motivo de muchas discusiones. Algunos pensaban que 64 $bits$ era el tamaño ideal, por una cuestión de eficiencia de procesamiento, para minimización de la sobrecarga que significa una longitud mayor del encabezado. Otros consideraron que era mejor elegir direcciones de longitud variable hasta 160 $bits$, lo suficientemente grandes como para permitir autoconfiguración cuando se usan direcciones IEEE 802. Finalmente, los diseñadores acordaron un espacio de direcciones fijas de 128 $bits$, ya que de este modo se permite la división y asignación sin temor al vaciamiento. La desventaja de manejar números tan grandes se compensó con la posibilidad de definir ciertos rangos sin el temor a quedarse sin direcciones.

Tal como en IPv4, para poder manejar más cómodamente números tan largos, se tuvo que fijar un sistema de representación. La división por $bytes$ no resultaba cómoda para manejar un número de 128 $bits$, por lo que se optó por una representación hexadecimal de palabras de 16 $bits$. Cada palabra se separa de la siguiente con el signo ":".

Así, una dirección IPv6 podría ser "$2031{:}\,0000{:}\,130f{:}\,0000{:}\,0000{:}\,09c0{:}\,876a{:}\,130b$", pero existen algunas reglas que permiten acortar la representación. Por ejemplo, los ceros al comienzo de un campo son opcionales y pueden omitirse, de tal manera que "$2031{:}\,0{:}\,130f{:}\,0{:}\,0{:}\,9c0{:}\,876a{:}\,130b$" puede ser otra representación de la misma dirección. También los ceros en campos sucesivos se pueden representar como "::" aunque sólo una vez en la dirección. Es decir que la dirección anterior también se podría escribir como "$2031{:}\,0{:}\,130f{::}\,9c0{:}\,876A{:}\,130b$".

Siguiendo las mismas reglas, una representación válida para la dirección "$fedc{:}\,ba98{:}\,7654{:}\,3210{:}\,0000{:}\,0000{:}\,0000{:}\,0089$" sería "$fedc{:}\,ba98{:}\,7654{:}\,3210{::}\,89$". Se trata de una representación con compresión de formato, muy apropiada en el caso de ciertas direcciones IPv6 especiales, tales como la de *loopback* "$0{:}\,0{:}\,0{:}\,0{:}\,0{:}\,0{:}\,0{:}\,1$", que se puede escribir como "::1", o la dirección IP no especificada "$0{:}\,0{:}\,0{:}\,0{:}\,0{:}\,0{:}\,0{:}\,0$", que se puede denotar simplemente por "::".

También existe una notación alternativa para entornos mixtos de dispositivos IPv4 e IPv6. En estos casos, se puede escribir la dirección en un formato "$x{:}\,x{:}\,x{:}\,x{:}\,x{:}\,x{:}\,d.d.d.d$", donde las letras "$x$" representan los valores en formato hexadecimal de las seis palabras de 16 $bits$ de mayor orden de la dirección, y las letras "d" representan los valores decimales de los cuatro bytes de menor orden. Es decir que, en estos casos, los últimos 32 $bits$ son los de la dirección IPv4, por lo que se permiten direcciones del tipo "::212.200.31.251" para representar la dirección "212.200.31.251".

Como en el caso de direcciones IPv4 sin clases, las direcciones IPv6 se dividen, fundamentalmente, en un número que representa el NetID, seguido de otro que sería el HostID. Al primero, se lo denomina prefijo, y la cantidad de bits que lo compone se llama longitud del prefijo, adoptándose para su representación la misma notación que en CIDR. Por ejemplo, la notación "$805b{:}\,2d9d{:}\,dc28{::}\,fc57{:}\,d4c8{:}\,1fff/48$" se refiere a una dirección con prefijo de 48 $bits$.

En cuanto a la asignación de direcciones IPv6, los diseñadores pensaron en un esquema que facilitara lo más posible la entrega a ISP, organizaciones e individuos. Al principio, imaginaron una estructura similar a la de clases de IPv4, donde el tipo de dirección se indica mediante una sucesión predeterminada de bits al comienzo, tal como se presenta en la RFC 2373. Como esta forma de división del espacio recordaba demasiado el esquema IPv4 y el principal objetivo de los desarrolladores era mejorar el mecanismo de decisión de ruteo, pero bajo un esquema de asignación que podría estar sujeto a cambios, surgieron modificaciones que se publicaron en la RFC 3513.

La RFC 3513 define tipos de direcciones IPv6, identificados por sus bits de mayor orden, tal como se presenta en la Tabla 10.1.

Tabla 10.1 - Tipos de Direcciones IPv6

Tipo de Dirección	Prefijo Binario	Notación
No especificado	00…0 (128 bits)	::/128
Loopback	00…1 (128 bits)	::1/128
Muticast	1111 1111	FF00::/ 8
Link-local unicast	1111111010	FE80::/10
Site-local unicast	1111111011	FEC0::/10
Global unicast	2000::/3	

10.3.1 Direcciones Unicast IPv6

La inspección de la Tabla 10.1 permite observar que, según el prefijo, existen tres tipos de direcciones *unicast*. Los prefijos indican el alcance de la dirección, refiriéndose este término al entorno de una red donde la dirección puede ser usada y a las posibilidades de comunicación de un nodo con ese tipo de dirección. Por ejemplo, *link-local* (enlace local) se refiere al uso entre nodos de la misma red, mientras *site-local* (sitio local) implica la comunicación dentro de un mismo sitio o administración y *global* se asocia al uso en Internet. A su vez, dentro de las direcciones *unicast* del tipo *global* pueden existir subtipos especiales de direcciones, tales como direcciones IPv6 con direcciones IPv4 embebidas. También se deja abierta la posibilidad a futuro de definir nuevos subtipos.

Según el papel que juegue un dispositivo, distinguiéndose fundamentalmente éste por su actuación como *host* o como *router*, puede tener mucho o poco conocimiento de la estructura interna de la dirección IPv6.

Como mínimo, un nodo puede considerar una dirección *unicast*, incluyendo la propia, como carente de estructura interna, pero si se tratara de un dispositivo un poco más sofisticado debería considerar la existencia de un prefijo de subred de $n\ bits$, seguido de un identificador de interfaz de $(128 - n)\ bits$. En general, los *routers* reconocerán una estructura interna jerárquica, necesaria al momento del ruteo, pero este conocimiento variará según la posición del propio *router* en la jerarquía de ruteo.

Para el traslado de tráfico de paquetes **IPv6 *unicast* global** se separó un espacio que comienza con los primeros tres bits de la dirección en "001" y que permite cualquier comunicación del tipo global. En principio, la forma de asignar estas direcciones fue un dilema, pero la experiencia acumulada con el mismo problema en IPv4, permitió considerar que sería beneficioso que la propia estructura interna de la dirección reflejara la topología de la red.

Una estructura interna de direcciones que refleja la topología facilita la tarea de asignación en diversos niveles de la jerarquía, permitiendo a los ISP la entrega flexible de espacios de direcciones, al igual que a las organizaciones finales el manejo más libre de sus propios bloques. También, una dirección con estructura interna puede dar indicios de ubicación o de jerarquía de ruteo, facilitando el agregado de rutas para un enrutamiento más eficiente.

Por este motivo, se planteó una forma genérica de dividir el espacio de $128\ bits$ de una dirección *unicast* global, tal como se presenta en la Fig. 10.1. Como se puede observar, la dirección se divide en tres espacios. El Prefijo de Ruteo Global (GRP, Global Routing Prefix) de $48\ bits$ junto con el Identificador de Subred (SubnetID) de $16\ bits$, representa los dos niveles básicos que las direcciones precisan contar dentro de una estructura jerárquica: el nivel global y el específico del sitio.

El GRP comienza con el prefijo "001" por pertenecer al espacio global asignado y se trata de los identificadores de red designados a la mayoría de los sitios, conocidos genéricamente como "/48".

El campo de SubnetID, de $16\ bits$, otorga suficiente flexibilidad a cada sitio como para dividirse internamente en subredes que puedan reflejar su propia estructura. Por ejemplo, una organización muy pequeña podrá fijar todos los bits de este campo en "0" porque no precisa subdividir su estructura interna. Una organización mediana, en cambio, puede usar este campo para dividir su espacio en subredes, exactamente como lo permite IPv4 con el concepto de *subnetting*. Por otra parte, una organización grande puede usar este espacio como en el caso VLSM, para crear múltiples niveles de jerarquía, según las necesidades internas que aparezcan.

Los últimos $64\ bits$ de la dirección quedan disponibles para los Identificadores de Interfaz, debiendo ser únicos dentro de un GPR y un SunetID.

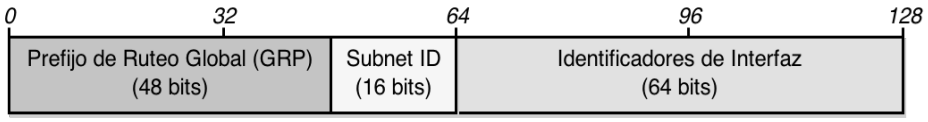


Figura 10.1- Estructura Interna de dirección IPv6 globales unicast.

Es interesante destacar que, al principio, la RFC 2374 había dividido el prefijo global en una jerarquía de dos niveles, que denominaba Top-Level Aggregators (TLA, Agregadores de Nivel Superior) de $13\ bits$ y Next-Level Aggregators (NLA, Agregadores de Siguiente Nivel) de $24\ bits$. Los primeros se consideraron para organizaciones muy grandes, con muchos bloques IPv6 asignados por las autoridades. Los segundos serían los que tomarían bloques a

partir de los anteriores y, a su vez, entregarían espacios a organizaciones de usuarios finales o sitios de Internet. Entre ambos campos, se reservaban 8 $bits$ por si se presentaban necesidades de expansión a futuro. Con el paso del tiempo y la experiencia acumulada en Internet. esta estructura resultó poco flexible, permitiéndose posteriormente a los Registros Regionales de Internet (APNIC, ARIN, LACNIC y RIPE) disponer de estos primeros bits como se considerara necesario, cuestión que se oficializó en la RFC 3587 de 2003. Por este motivo, en la actualidad no existe una única estructura que fije cómo se divide el prefijo de ruteo de 48 $bits$ y esto otorga mucha mayor flexibilidad a la asignación.

En cuanto al Identificador de Interfaz de las direcciones *unicast*, se destina para identificar interfaces en un enlace, por lo que se requiere que sean únicos dentro de un prefijo de SubnetID. Para todas las direcciones *unicast*, excepto para aquellas que comienzan con el valor binario "000", se exige que el identificador se construya en un formato especial, denominado EUI-64 Modificado.

Justamente, la posibilidad de contar con 64 $bits$ para el espacio de Identificador de Interfaz, llevó a pensar en incluir alguna forma de mapeo de las direcciones físicas en las direcciones de red, apartándose así del planteo de IPv4. La consecuencia directa de esta idea se tradujo en una mayor facilidad de administración, ya que la dirección IP se podría derivar de una dirección IEEE 802 MAC.

Recordemos que las direcciones MAC se dividen en dos bloques de 24 $bits$. El primer bloque, conocido como Identificador Organizacionalmente Único OUI es un número que proviene de fabricación, en tanto que el segundo es específico del dispositivo.

IEEE definió un formato denominado Identificador Único Extendido de 64 bits (EUI-64, Extended Unique Identifier-64) similar al formato MAC de 48 $bits$, pero con un identificador de dispositivo de 40 $bits$, en lugar de 24 $bits$. Esta modificación aumenta 65.535 veces las direcciones de dispositivos dentro de un OUI. Como se ha mencionado, para el Identificador de Interfaz de la dirección IPv6 se adoptó una forma modificada de EUI-64.

La Fig. 10.2 representa este cambio de formato. Su aparición se debe a que la mayoría de los dispositivos actuales todavía utiliza el formato de 48 $bits$, por lo que se debía definir la manera de convertirlo a EUI-64, y luego a su versión modificada adoptada para el Identificador de Interfaz de IPv6.

Para traducir una dirección MAC tradicional al nuevo esquema, en el espacio de orden más bajo del nuevo identificador se colocan los 24 $bits$ de menor orden de identificación del dispositivo y los 16 $bits$ que siguen se rellenan con la palabra hexadecimal "$fffe$". En el espacio de 24 $bits$ de mayor orden se copian los correspondientes de la vieja dirección. Por último, se cambia el bit 7, segundo menos significativo del primer byte.

Por ejemplo, si un dispositivo tiene una dirección MAC "00: 30: 48: 2A: 19: 89", primero se la debe transformar al formato EUI-64, por copiado, en "00: 30: 48: ff: fe: 2a: 19: 89". Luego hay que complementar el bit

mencionado previamente, derivando en el nuevo número "$02:30:48:ff:fe:2a:19:89$", tal como se muestra en la Fig. 10.2.

Para armar la dirección IPV6 de alcance *link-local*, se agrega el prefijo reservado $fe80::/10$, quedando finalmente la dirección $fe80::230:48ff:fe2a:19:89$. Sin embargo, si se conoce el prefijo global de $64\ bits$ se puede escribir el mismo armando una dirección *global*.

De este modo, se facilita la administración ya que no es necesario guardar relaciones entre direcciones MAC y direcciones IP, aunque la desventaja es que, si cambia la dirección física, también cambia la dirección de red.

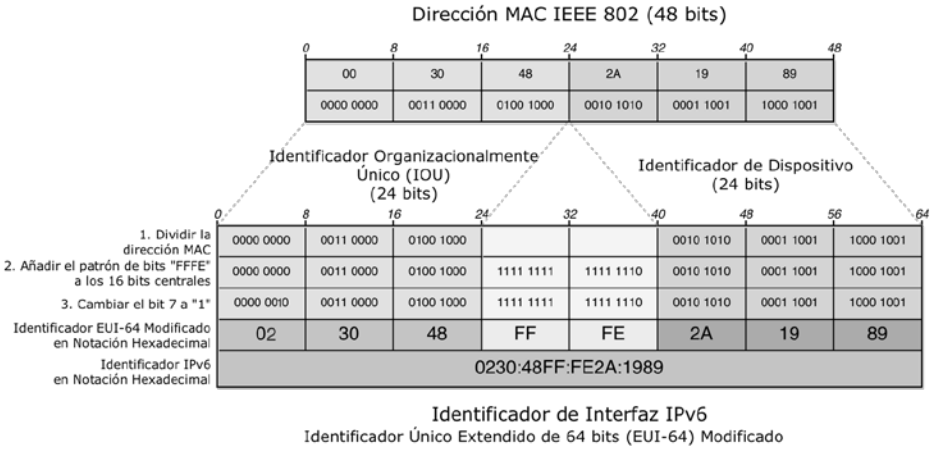


Figura 10.2- Identificador de Interfaz de 64 bits generado a partir de una MAC

Dentro del espacio *unicast* también se definen direcciones con significado especial.

La dirección IPv6 todos ceros es una **dirección no especificada**, que se refiere al propio host y se usa como dirección fuente de un paquete cuando el dispositivo no conoce su propia dirección IP, como sucede en IPv4. Se trata de la dirección "$0:0:0:0:0:0:0:0$", que se puede expresar como "$::$" . Esta dirección nunca debe usarse como dirección destino.

La dirección *unicast* "$0:0:0:0:0:0:0:1$", también expresable como "$::1$", es la dirección para la funcionalidad de ***loopback***. Como en IPv4, puede ser utilizada por un dispositivo para enviarse paquetes IPv6 a sí mismo, tratándose como una dirección de alcance local de una interfaz virtual. Nunca debe usarse como dirección fuente en un paquete que se envíe sobre una red y cualquier paquete que lleve esta dirección como dirección destino no debe bajarse a la red.

Por otra parte, los mecanismos de transición a IPv6 incluyen una técnica para que, tanto *hosts* como *routers* puedan crear túneles de paquetes IPv6 sobre infraestructura de enrutamiento IPv4. Aquellos nodos IPv6 que usan esta técnica tienen asignadas direcciones *unicast* IPv6 especiales que cargan direcciones IPv4

en los últimos 32 $bits$, conocidas como **direcciones IPv6 compatibles con IPv4**. Su formato es del tipo "$::a.b.c.d$", siendo los primeros 96 $bits$ nulos y $a.b.c.d$ la notación tradicional IPv4.

Otro tipo de direcciones IPv6 que tienen una dirección IPv4 embebida permite representar las direcciones de nodos IPv4 como direcciones IPv6 en un formato conocido como **dirección IPv4 mapeada a IPv6**. Estas direcciones siguen la forma: "$::ff:ff:a.b.c.d$".

En ambos casos, los primeros 80 $bits$ son nulos y la diferencia es el valor otorgado a los 16 $bits$ siguientes. El formato compatible distingue a aquellos dispositivos que entienden IPv6, en tanto que el formato mapeado se usa en los casos de dispositivos IPv4 convencionales.

También existen dos tipos de direcciones *unicast* definidas para uso local: las de enlace local y las de sitio local. Se trata de un espacio de direcciones para uso privado, no enrutable, de significado sólo local a un sitio o enlace particular, que comienza con el prefijo "$0xfe$", llevando el primer bit del siguiente byte en "1". Es decir que el siguiente byte puede ser cualquier número entre "$0x8$" y "$0xF$". A su vez, este espacio se divide en dos tipos de direcciones, según su alcance.

Las **direcciones de alcance de enlace local**, conocidas como ***link-local***, sólo tienen significado dentro de un enlace interpretado como red física. No se re-envían ni siquiera dentro de la organización. Pueden usarse para direccionamiento en un enlace, ya sea para configuración automática de direcciones, descubrimiento de vecinos o cuando no hay *routers* presentes. Comienzan con el prefijo "fe" y llevan el décimo bit en "0", o sea que se trata de los prefijos hexadecimales "$fe8$", "$fe9$", "fea" y "feb".

Las **direcciones de alcance de sitio local**, conocidas como ***site-local***, permiten un tipo de direccionamiento interno sin necesidad de prefijo global público. Los *routers* reenviarán los datagramas con estas direcciones dentro del sitio, nunca hacia Internet. Se caracterizan por tener el bit décimo en "1". Es decir que comienzan con "$0xfe$" y luego sigue algún número en el rango hexadecimal "c" - "f". Se trata de los prefijos hexadecimales "fec", "fed", "fee" y "fef".

10.3.2 Direcciones Multicast IPv6

Como se ha mencionado, uno de los mayores cambios sufridos por el esquema de direcciones IPv6 fue la eliminación de la dirección de *broadcast*. En su lugar, los diseñadores del nuevo protocolo eligieron expandir el alcance del significado del direccionamiento *multicast*.

En el esquema IPv6, las direcciones *multicast* comienzan con el prefijo hexadecimal "$0xff$". El formato de los restantes 120 $bits$ se presenta en la Fig. 10.3, encontrándose dividido de la siguiente manera, en el orden que se precisa a continuación:

- **Banderas o Flags**: se trata de 4 $bits$ reservados para indicar la naturaleza de la dirección. Los primeros tres bits se fijan en "0". El cuarto bits se denomina bandera de transitorio. Cuando este bit se ajusta en "0"

significa que la dirección *multicast* tiene asignado un significado permanente, tratándose de las direcciones *multicast* bien conocidas. Cuando se ajusta en "1" se refiere a un grupo de carácter transitorio.

- **ID de alcance**: se trata de 4 $bits$ utilizados para indicar el alcance de la dirección. De los 16 valores posibles, sólo se encuentran definidos algunos: "0" queda reservado, "1" es para alcance de nodo local o *loopback*, "2" para alcance de enlace, "5" para alcance de sitio, "8" para alcance de organización local, "14" es para alcance global, y el valor "15" queda reservado.

- **ID de grupo**: es el número que conforman los 112 $bits$ restantes.

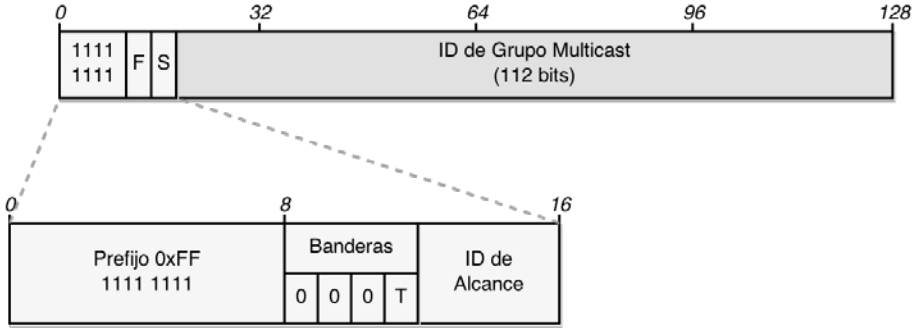


Figura 10.3- Direcciones Multicast IPv6.

Las direcciones de alcance global deben ser únicas en todo el ámbito de Internet, en tanto que las de alcance local son únicas dentro de una organización. Este significado del alcance sirve para que los *routers* tomen decisiones de re-envío de los paquetes.

Entre las **direcciones *multicast* bien conocidas** se puede mencionar la de todos los nodos, en sus dos versiones, "$ff:01:0:0:0:0:0:1$" para nodo local y "$ff:02:0:0:0:0:0:1$" para enlace local. También existe una dirección multicast para todos los routers, "$ff0x:0:0:0:0:0:2$", dentro del alcance especificado por el ID de alcance.

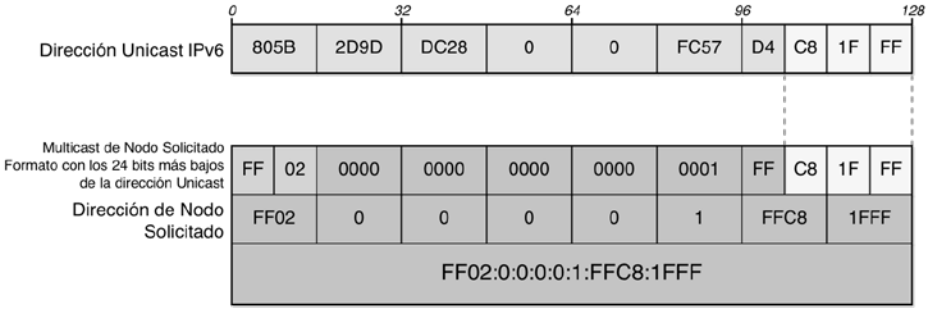


Figura 10.4- Dirección Nodo Solicitado IPv6.

Por otra parte, cada dirección *unicast* se asocia a una dirección *multicast* especial, denominada **dirección de nodo solicitado**, que se crea a partir de la propia dirección *unicast* con el propósito de que otros nodos la usen para comunicarse dentro de la red local. Por ejemplo, esta dirección se usa en el protocolo de Descubrimiento de Vecinos de IPv6 (ND, Neighbor Discovery) para resolución de direcciones de manera más eficiente que ARP en IPv4. La dirección lleva la bandera T en "0" y el ID de alcance en "2", indicando dirección permanente con alcance de enlace, mientras que los 112 $bits$ de ID de grupo se arman con los primeros 9 $bytes$ en "0" seguidos de 1 $byte$ fijo "$0x01$". Los siguientes 2 $bytes$ se fijan en alto "$0xff$", y los últimos 3 $bytes$ se mapean a partir de sus pares de la dirección *unicast* IPv6, tal como se presenta en la Fig. 10.4. Este mapeo permite que múltiples prefijos correspondientes a diferentes estratos, mapeen a la misma dirección de Nodo Solicitado, reduciendo el número de direcciones *multicast* a las que un nodo debe asociarse.

10.3.3 Direcciones Anycast IPv6

Se trata de un concepto nuevo que se desarrolla en la RFC 1546. El significado de una dirección destino *unicast* es el de enviar un mensaje a un único dispositivo, en tanto que una dirección destino *multicast* pretende alcanzar a un grupo de dispositivos. Las direcciones destino *anycast*, por su parte, se usan para llegar a cualquier miembro de un grupo, generalmente al más cercano, por cuestiones de eficiencia.

Este nuevo esquema se pensó para mejorar las comunicaciones provistas por el protocolo previo en aquellos casos en que un servicio es desplegado por cierta cantidad de servidores o *routers*, sin importar cuál de ellos lo provea. En términos de ruteo, significa que el paquete se re-enviará a cualquiera de los *routers* del grupo que esté más cerca. Esto permite compartir carga entre *routers* y mejorar la flexibilidad en caso de fallas.

Lo interesante es que no se trata de un grupo de direcciones especiales. Simplemente son direcciones *unicast*, que pasan a ser *anycast* cuando una dirección *unicast* se asigna a más de una interfaz.

10.4 Encabezado IPv6.

Aparte de la expansión del espacio de direccionamiento, IPv6 presenta un encabezado más simplificado, con soporte mejorado de extensiones y opciones, y campos para incorporar capacidad de etiquetado de flujos y de autenticación y encriptado.

Con fines de comparación, en la Fig. 10.5 se presentan los encabezados de ambos protocolos. En dicha figura se han marcado en gris claro aquellos campos de IPv4 que se mantuvieron en IPv6 y en negro aquellos campos de IPV4 que se quitaron. El color intermedio simboliza los campos nuevos definidos en IPv6 y el blanco se marca una redefinición de campos ya existentes.

El encabezado IPv6 se extiende a 40 *bytes* fijos y posee 8 campos, mientras que en IPv4 se cuenta con un encabezado de 20 *bytes* fijos y hasta 40 *bytes* de opciones, repartidos en 12 campos. El nuevo protocolo quita el campo de opciones de IPv4 y define cabeceras de extensión, aparte de la cabecera principal, para aquellos casos en los que se precise información adicional.

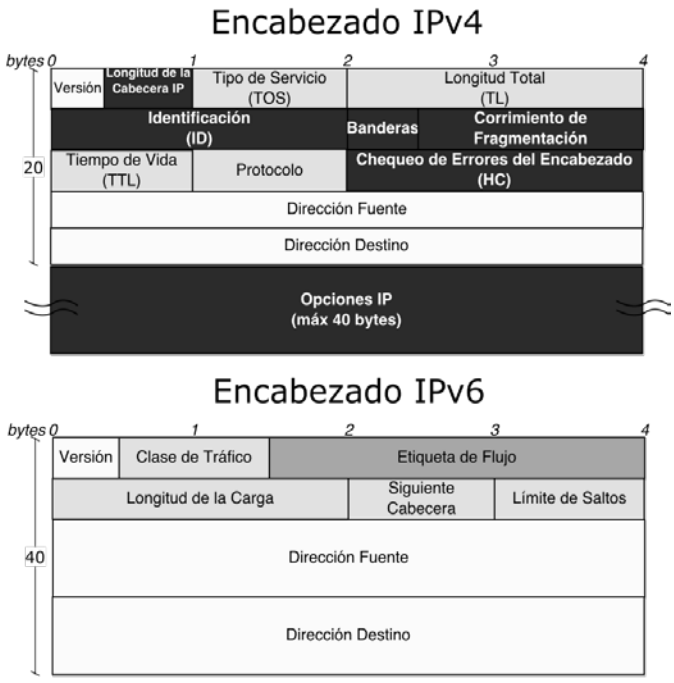


Figura 10.5- Comparación de encabezados IPv4 vs IPv6.

Existen campos de IPv4 que se eliminan en el encabezado IPv6, como es el caso del campo de control de error o *checksum*. La decisión responde al hecho de que el chequeo de errores ya se realiza a nivel de la capa de enlace que encapsula el paquete IP. De este modo, se quita algo de carga de procesamiento a los *routers*.

Por otra parte, desaparecen los campos IPv4 que cumplen funcionalidad de fragmentación. Este mecanismo se modifica conceptualmente en IPv6, ya que se descarga esta funcionalidad de los *routers*. En IPv6 los *routers* no fragmentan y, para que la interconexión de redes siga siendo posible, el nuevo protocolo adiciona un Mecanismo de Descubrimiento de la MTU obligatorio. De ser necesario, la fragmentación se realiza en origen y el re-ensamble en destino.

El campo de longitud total de IPv4, se transforma en un campo de 16 *bits* denominado Longitud de la Carga, en referencia a los datos transportados por IPv6. La carga de datos de un paquete IPv6 puede llegar a ser de 65.535 *bytes*,

aunque una extensión particular del encabezado permite una carga mayor que se conoce con el nombre de *jumbo payload*.

Como se ha explicado, el campo TOS de IPv4 se había anexado originalmente para influir en las decisiones de ruteo, pero en la práctica nunca llegó a aplicarse. En IPv6, se incluye un campo Clase de Tráfico con el que se pretende realizar la función original de TOS, pero con un formato mejorado. Por otra parte, un campo de Etiquetado de Flujo permite marcar tráfico como prioritario, agregando funcionalidad en cuanto a QoS.

En IPv6, el campo Límite de Saltos de $8\ bits$ se corresponde con el campo TTL de IPv4, que se ha renombrado para mayor claridad.

El campo protocolo de IPv4 señalaba el encabezado encapsulado por el paquete, generalmente TCP o UDP. En IPv6, este campo cambia de nombre y de significado. Se trata de un campo denominado Encabezado Siguiente, de $8\ bits$. En IPv6, en lugar de usar una cabecera de longitud variable, se utilizan una serie de cabeceras encadenadas que son parte del mismo protocolo. Por este motivo, desaparece el campo de opciones de IPv4, ya que los diseñadores prefirieron mantener en el encabezado aquellos campos verdaderamente necesarios y agregar cabeceras de extensión para ser usadas según las circunstancias.

Por este motivo, la estructura integral de un datagrama IPV6, es diferente respecto de un paquete IPv4, siguiendo el formato que se presenta en la Fig. 10.6. Se trata de un encabezado fijo de $40\ bytes$, cuyos campos se describirán posteriormente, seguido de una serie de cabeceras de extensión que cargan información extra según sea necesario. Se dice que se encuentran encadenadas, ya que cada una apunta a la siguiente, para indicar su presencia. Por último aparecen los datos que forman parte del nivel superior a IP.

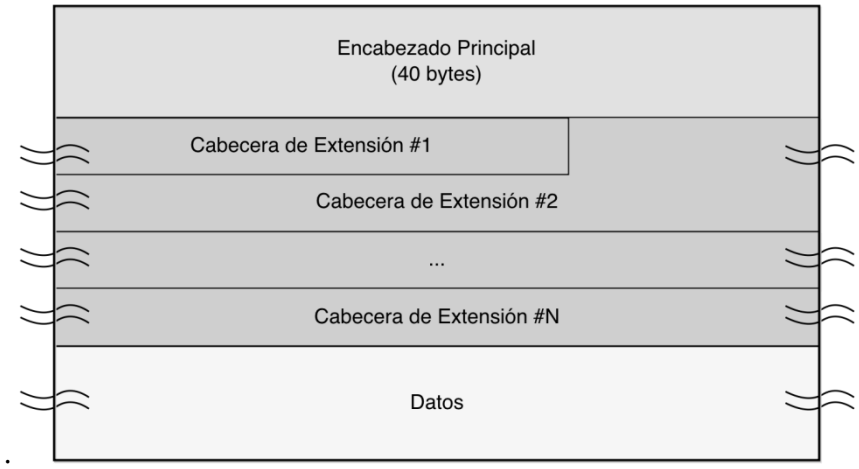


Figura 10.6- Estructura integral de un paquete IPv6

A continuación, se describe brevemente la funcionalidad de cada campo del encabezado IPv6:

- **Versión** (4 $bits$): mantiene el significado de IPV4, excepto que en este campo, los datagramas IPv6 llevarán el número "6".

- **Clase de Tráfico** (8 bits): da vigencia al campo TOS pero por medio de la utilización de Servicios Diferenciados (DS, Differentiated Services), método definido en la RFC 2474, que especifica técnicas de QoS tanto para IPv4 como para IPv6. En esta RFC se propuso el reemplazo del campo TOS por un campo de 6 $bits$, denominado DS, en IPv4. Los Servicios Diferenciados se sostienen sobre un mecanismo de clasificación y marcado de paquetes que pertenecen a cierta clase de tráfico. Estos paquetes, al ser reconocidos por *routers* que manejan este servicio, son re-enviados de acuerdo a las prioridades con las que están marcados. Las prioridades pueden ser de varios tipos, por ejemplo baja pérdida o baja latencia, en un tratamiento *hop-by-hop*. Dentro de un nodo, IPv6 deberá proveer la facilidad para que los protocolos de niveles superiores puedan pasarle los valores del campo Clase de Tráfico a los paquetes que los mismos originan.

- **Etiqueta de Flujo** (20 $bits$): este campo puede usarse para etiquetar secuencias de paquetes que requieren tratamiento especial por parte de los *routers*, tales como servicios de *QoS* no default y servicios de tiempo real. El concepto de flujo como una secuencia de paquetes enviados desde una fuente a uno o más destinos, se define en la RFC 2460. Por ejemplo, de este modo, un *stream* de video puede identificar sus datagramas de la misma manera para que reciban el mismo tratamiento en los *routers* intermedios entre fuente y destino y evitar de este modo problemas de latencia. La naturaleza de este tratamiento especial debería llegar a los *routers* a través de algún protocolo de control, tal como el Protocolo de Reserva de Recursos (RSVP, Resource Reservation Protocol), o por información transportada por los mismos paquetes, por ejemplo en alguna opción *hop-by-hop*. Cualquier flujo se identifica por medio de su dirección de origen, su dirección destino y la Etiqueta de Flujo, cuando la misma es distinta de "0". La etiqueta la asigna la fuente del paquete y se trata de un número aleatorio que oficia de entrada a una función *hash* para el flujo particular, tal como se sugiere en el Apéndice A de la RFC 2460.

- **Longitud de la carga** (16 $bits$): reemplaza el campo de Longitud Total de IPv4 pero, en este caso, se incluye sólo la longitud de de los datos y las cabeceras de extensión, es decir la longitud de todo lo que no es cabecera fija.

- **Siguiente Cabecera** (8 $bits$): cuando el paquete lleva cabeceras de extensión, este campo carga el ID de la siguiente cabecera. En caso contrario, cumple la misma función que el campo Protocolo de IPv4, conservando los valores que allí aparecían.

- **Límite de Saltos** ($8\ bits$): cumple la misma función que el campo TTL de IPv4.

- **Direcciones Fuente y Destino** ($128\ bits$): cuyos detalles ya se han abordado.

10.5 Cabeceras de Extensión IPv6.

El encabezado de $40\ bytes$ descripto es de carácter obligatorio, pero puede estar seguido por una o varias Cabeceras de Extensión, situadas antes de los datos. Estas cabeceras dan mayor flexibilidad al procesamiento ya que incluyen campos sólo necesarios en determinadas circunstancias. La Tabla 10.2 presenta las Cabeceras de Extensión más conocidas.

Las cabeceras forman una cadena, ya que todas poseen un campo Siguiente Cabecera que lleva el ID de la cabecera que sigue en la cadena, tal como se presenta en la Fig. 10.7. El encadenado indica el orden en que dichas cabeceras han de ser procesadas. Por ejemplo, si un paquete IPv6 transportara la cabecera de Extensión de Opciones Hop-By-Hop y la de Fragmentación, según la Tabla 10.2 el valor del campo Siguiente Cabecera en el encabezado principal sería "0", en el de Opciones sería "44" y en el de Fragmentación sería "6", si los datos encapsulados fueran los de un segmento TCP. También en ese orden sería su procesamiento.

Cuando aparece más de una cabecera de extensión, IPv6 recomienda el siguiente orden: Salto a Salto, Opciones de Destino (para opciones a ser procesadas por el primer destino que aparece en el campo de Dirección Destino IPV6 y los destinos listados en la cabecera de Enrutamiento), Enrutamiento, Fragmentación, Autenticación, ESP y Opciones de Destino (para ser procesadas por el destino final).

Tabla 10.2 – Cabeceras de Extensión más conocidas

Valor	Nombre de la Cabecera	Descripción
0	Extensión de Opciones Salto a Salto o Hop-by-Hop	De longitud variable. Define un conjunto de opciones en formato TLV para ser procesadas en los *routers* que encuentra el paquete en su camino de fuente a destino. Es la única cabecera con procesamiento intermedio.
1	ICMPv4	
6	TCP	
17	UDP	
43	Extensión de Enrutamiento	De longitud variable. Con esta extensión, se permite que el dispositivo fuente del

		datagrama especifique una ruta para el paquete.
44	Extensión de Fragmentación	De 8 $bytes$. Se incluye cuando el datagrama contiene un fragmento, para indicar el orden del mismo dentro del datagrama original, permitiendo además distinguir si se trata del último fragmento.
50	Encapsulado de Carga Seguro (ESP, Encapsulating Security Payload)	De longitud variable. Carga datos cifrados.
51	Extensión de Autenticación	De longitud variable. Carga información para verificación de autenticación.
58	ICMPv6	
59	No hay Cabecera de Extensión	
60	Extensión de Opciones en Destino	De longitud variable. Define un conjunto de opciones en formato TLV para su procesamiento en destino.

Todas las cabeceras de extensión tienen un tamaño que es múltiplo de 8 $bytes$, aunque algunas cargan un relleno para cumplir esta premisa. También, todas las cabeceras son para examen y procesamiento en el destino del paquete, excepto el caso de opciones Hop-by-Hop. Por definición, todas las cabeceras son opcionales y deberían aparecer una sola vez en la cadena, excepto la cabecera de Opciones de Destino que puede aparecer dos veces. Cuando un nodo no reconoce una cabecera, simplemente descarta el paquete, enviando a destino un mensaje ICMPv6 del tipo Problema de Parámetros.

Tal como se presenta en la Fig. 10.8, todas las cabeceras comienza con dos campos fijos, denominados Siguiente Cabecera y Longitud de Cabecera de Extensión, ambos de 1 $byte$. Cuando las cabeceras cargan opciones, como en el caso de las Opciones Salto a Salto y Opciones de Destino, estas se describen mediante una estructura interna que es un conjunto de tres valores (*Tipo* (1 $byte$)/*Longitud* (1 $byte$)/*Valor* ($variable$)), es decir en formato TLV.

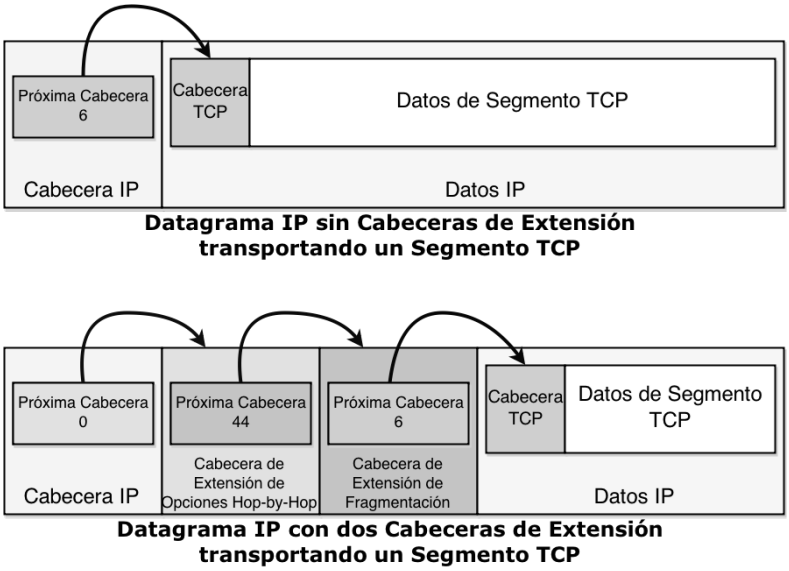


Figura 10.7 - Cabeceras de Extensión, campo Siguiente Cabecera.

La secuencia de opciones debe procesarse en el orden estricto en que se presentan. El campo Tipo, además de identificar la opción, especifica la acción a tomar cuando el dispositivo que procesa el datagrama no reconoce la opción. Las acciones en estos casos pueden ser: saltar la opción, descartar silenciosamente el datagrama o descartar el datagrama editando un mensaje ICMP del tipo Problema de Parámetros. También el campo Tipo posee una bandera de cambio que, cuando está levantada, permite modificar las opciones mientras el datagrama es procesado.

La cabecera de extensión de **Opciones Salto a Salto**, es de longitud variable y define un conjunto arbitrario de opciones para que sean examinadas por todos los dispositivos en el camino de fuente a destino. Cuando se usa esta extensión, se identifica en la cabecera IPv6 con el campo Siguiente Cabecera en "0", colocándose al comienzo de todas las cabeceras de extensión ya que se debe procesar en cada nodo del camino. La opción definida para esta cabecera en la RFC 2675, permite la transmisión de paquetes con cargas mayores a $65535\ bytes$, conocidos como *jumbogramas*. En el campo Valor de la opción se coloca un número de $32\ bits$ que indica la longitud total en $bytes$ del *jumbograma* menos $40\ bytes$ de la longitud de la cabecera fija IPv6. En este caso, el campo Longitud de la Carga en el encabezado principal, se fija en "0", permitiéndose cargar hasta $4\ Gbytes$ de datos en un mismo datagrama. Esta posibilidad exige algunas precauciones a nivel UDP y TCP también descriptas en la RFC 2675, debido a la existencia de un campo longitud en UDP y a la fijación de una longitud máxima de segmento en TCP.

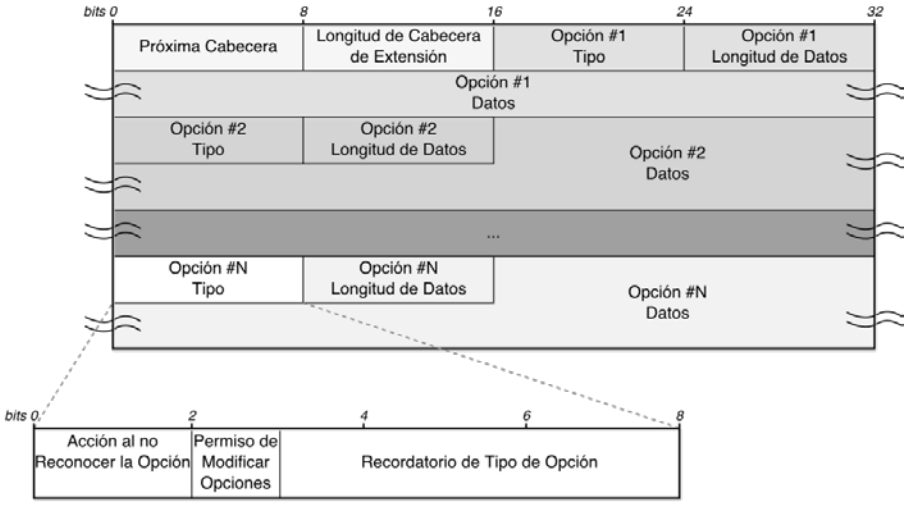


Figura 10.8 - Estructura interna de las Cabeceras de Extensión.

La cabecera de extensión de **Opciones de Destino** se coloca al final de las cabeceras cuando debe ser procesada sólo en el destino final. Se usa para transportar información adicional sólo necesaria en destino y se carga con opciones en el formato TLV.

La **Cabecera de Ruteo** sirve para el mismo propósito que la opción de ruteo de fuente de IPv4: listar uno o más nodos intermedios a ser visitados en el camino a destino. Luego del campo Siguiente Cabecera y Longitud de la Cabecera de Extensión, esta cabecera lleva un campo de 1 $byte$ denominado Tipo de Ruteo, considerado para distinguir a futuro ruteos de diferentes premisas. Luego sigue el campo Segmentos que Faltan, también de 1 $byte$, que indica el número de *routers* que falta atravesar hasta llegar a destino. A continuación se presenta el listado de direcciones de 128 $bits$ IPv6 que especifican la ruta a seguir por el paquete.

La **Cabecera de Fragmentación** se incluye en paquetes fragmentados en la dirección fuente del datagrama para que puedan ajustarse al valor de MTU en destino. En la parte de datos incluye un campo de FO, una bandera de MF y un campo de ID, todos con mismo significado que en IPv4. El datagrama a ser fragmentado se considera dividido en dos partes. La parte que no se puede fragmentar consiste del encabezado IPv6 más encabezados de extensión que deben ser procesados en nodos en la ruta al destino. El resto es la parte que se puede fragmentar y se divide en fragmentos cuya longitud debe ser un múltiplo de 8 $bytes$, como en IPv4.

En IPv6 se impusieron algunos cambios con respecto a la funcionalidad de fragmentación, con el propósito de mejorar la eficiencia de procesamiento en el ruteo y adaptarse a la nueva tecnología vigente. En este sentido, se modificó el valor de la MTU mínima que se exigía poder manejar en IPv4, de 576 $bytes$, aumentándose a 1280 $bytes$. Esto mejora la eficiencia entre la relación de la

longitud de datos a la longitud del encabezado, intentando a la vez reducir la probabilidad de fragmentación.

También, IPv6 impuso la condición de que sólo el nodo fuente del datagrama puede ser capaz de fragmentar, descargando esta funcionalidad de los *routers*. Para poder cumplir con esta nueva condición de fragmentación, la fuente debe conocer la MTU mínima en el camino a destino. Una de las consecuencias de este nuevo criterio es que, si un *router* tuviera que entregar un datagrama a un enlace con MTU menor, sólo podría descartarlo, pudiendo advertir a la fuente de esta situación por medio de un mensaje de error ICMPv6. Otra consecuencia es que adquiere significado un viejo mecanismo ideado para IPv4, conocido como Descubrimiento de la MTU de un Camino, definido en la RFC 1981. El re-ensamble se sigue realizando en destino, como en IPV4.

El mecanismo de descubrimiento de la MTU aprovecha la generación, por parte de los *routers*, de mensajes ICMPv6 dirigidos a la fuente, en el caso de que no puedan fragmentar un datagrama. Generalmente, el dispositivo fuente comienza a transmitir datagramas de acuerdo al tamaño de la MTU de su propia red y, si recibe uno de estos mensajes ICMPv6, adapta la longitud de los siguientes datagramas al tamaño de MTU anunciado en el propio mensaje de error. Una desventaja de este mecanismo es su lentitud de adaptación a situaciones dinámicas, ya que el mismo se sostiene sobre una única ruta, pero ésta podría variar en el tiempo.

La **Cabecera de Autenticación** tiene un formato definido en la RFC2402. Esta cabecera fue ideada para proveer integridad y autenticación de origen a los datos de datagramas IP. Adicionalmente, de forma opcional, provee protección contra ataques *replay*, según la asociación de seguridad establecida con el destino de los datos. La cabecera puede usarse sola, aplicarse en combinación con la cabecera ESP, o desplegarse en modo túnel de manera anidada.

La cabecera contiene un campo de $32\ bits$ que carga un valor arbitrario conocido como Índice de Parámetros de Seguridad (SPI, Security Parameter Index). Este valor, en combinación con la dirección destino y un protocolo de seguridad, identifica la Asociación de Seguridad elegida para el datagrama. Otro campo de $32\ bits$, conocido como Número de Secuencia, contiene un valor que crece de manera monótona, para evitar que datagramas viejos se utilicen para realizar ataques *replay*.

Un campo, de longitud variable múltiplo de $32\ bits$, contiene el Valor de Chequeo de Integridad (ICV, Integrity Check Value) para ese datagrama. El algoritmo de autenticación empleado para el cálculo del ICV se selecciona en el establecimiento de la Asociación de Seguridad. Pueden elegirse códigos de autenticación basados en algoritmos de cifrado simétricos o funciones *hash*, muchas veces combinadas con algoritmos de firma digital. El nodo receptor es el encargado de calcular el ICV sobre los campos especificados del datagrama, con el algoritmo acordado en principio, para verificar que coincida con el campo del encabezado de autenticación. Si coincide, el datagrama es legítimo y se debe aceptar para su procesamiento. Si el chequeo falla, se lo debe descartar, siendo aconsejable registrar el evento para posteriores auditorías. El registro debería

incluir el valor del SPI, la fecha y hora de recepción, la dirección fuente, la dirección destino y el Identificador de Flujo IPv6.

La **Cabecera de Encapsulado de Carga Seguro ESP** es similar a la definida en la RFC 2406. Esta cabecera puede ser aplicada sola, en combinación con la de autenticación o en forma anidada en el modo túnel (antes de un encabezado IP encapsulado). ESP se usa para proveer confidencialidad, autenticación de origen de los datos, integridad y servicio anti- *replay*, según lo establecido en la Asociación de Seguridad preliminar. Toda la información a continuación de esta cabecera se encuentra cifrada y, por lo tanto, no es accesible a dispositivos intermedios.

Los campos incluidos en la cabecera ESP son: Índice de Parámetros de Seguridad SPI ($4\ bytes$), Número de Secuencia ($4\ bytes$), Datos de Carga (de longitud variable), Relleno ($0 - 255\ bytes$), Longitud de Relleno ($1\ byte$), Siguiente Cabecera ($1\ byte$) y Datos de Autenticación (de longitud variable). Los primeros dos campos son similares a los del mismo nombre de la cabecera de autenticación y cumplen la misma función, ya explicada.

El campo de Datos de Carga contiene los datos descriptos en el campo Siguiente Cabecera. A pesar de tratarse de un número entero de bytes, muchas veces se precisa el campo de Relleno por las propias características de los algoritmos de cifrado utilizados. El campo Datos de Autenticación contiene un ICV calculado sobre el ESP completo, cuya longitud queda especificada por la función de autenticación elegida.

10.6 Configuración de Direcciones IPv6

Como se ha explicado, la configuración manual de dispositivos IPv4 comenzó a convertirse en un verdadero problema cuando las redes comenzaron a expandirse e Internet parecía crecer de manera exponencial. El protocolo DHCP se ideó para salvar esta situación, aliviando así el trabajo de los administradores.

Los diseñadores de IPv6 pensaron avanzar un paso más, dotando a los dispositivos que trabajaran con este protocolo de la capacidad de configurarse de manera automática e independiente.

La RFC 2462 especifica los pasos a seguir para que un dispositivo pueda configurar sus interfaces IPv6. El proceso incluye la creación de una dirección de alcance de enlace local y la verificación de la unicidad de dicha dirección en ese enlace. En cualquier caso, el mecanismo puede realizarse de dos maneras, denominadas sin estado y con estado.

El **mecanismo de auto-configuración sin estado** no requiere configuración manual de *hosts* ni servidores adicionales. En este caso, un *host* puede generar sus propias direcciones por medio de una combinación de la información disponible localmente e información recibida de *routers*. Los *routers* anuncian por medio de mensajes los prefijos que identifican la subred asociada con un enlace y los *hosts* generan identificadores de interfaz únicos en dicha subred. La dirección es una combinación de ambos. De no existir *routers*, un host

sólo puede generar direcciones del enlace local pero, a partir de ellas, puede comunicarse con otros nodos en el mismo enlace.

En el **mecanismo de auto-configuración con estado**, los *hosts* obtienen sus direcciones de interfaz y toda la información necesaria para su configuración, a partir de servidores. Un ejemplo de este caso lo constituye la utilización de servidores DHCPv6 en la propia red. Esta aproximación se suele usar cuando se requiere un control más rígido sobre las direcciones asignadas.

Ambos mecanismos se complementan. Por ejemplo, un *host* puede usar el mecanismo sin estado para la configuración de sus propias direcciones y el mecanismo con estado para obtener otro tipo de información. Es el administrador quien especifica el mecanismo a utilizar a través del ajuste de ciertos campos en los mensajes ICMPv6 de Aviso de Routers.

Ya sea para el caso de un *host* o de un *router*, cuando un sistema arranca, comienza el proceso de autoconfiguración generando una dirección de enlace local, conformada por el prefijo de enlace local bien conocido y el identificador de la interfaz. Este tipo de direcciones de enlace local comienza con el prefijo "$0xfe$" seguido de los bits "10". A continuación la dirección presenta $54\ bits$ en "0" y, por último, el identificador de interfaz de $64\ bits$, tal como se ha mencionado en el apartado 10.3.1.

Se trata de una dirección tentativa, ya que el dispositivo debe verificar que otro dispositivo en el mismo enlace no se encuentre usando la misma dirección. La verificación se realiza transmitiendo un mensaje de Solicitud de Vecino con la dirección en cuestión, de tal modo que, si otro dispositivo la reconoce como propia, contesta con un mensaje Aviso de Vecino. De suceder esto, se detiene el proceso de autoconfiguración, requiriéndose una configuración manual sobre la interfaz en cuestión. Por otra parte, si el nodo comprueba que su dirección de enlace local tentativa es única, se la puede asignar a su interfaz. En ese momento el nodo posee conectividad IP con sus nodos vecinos.

Sólo en el caso de un *host*, el proceso continúa como se explicará a continuación. El siguiente paso consiste en recibir un Aviso de Router o determinar que no hay *routers* presentes. La recepción de mensaje de Aviso de Router especifica la clase de autoconfiguración que el *host* debería realizar, pero si no se reciben estos mensajes, se determina la no existencia de *routers* y la autoconfiguración debería realizarse mediante el mecanismo con estado.

Los mensajes de Aviso de Router son periódicos, pero si el host no desea esperar, edita mensajes de Solicitud de Router a la dirección *multicast* de todos los *routers*. Aparte de señalar el mecanismo de autoconfiguración a utilizar, los mensajes de Aviso de Router pueden contener opciones de información de prefijo para el caso de autoconfiguración sin estado, para poder generar direcciones de alcance global o para el sitio. Además, pueden llevar información del prefijo de subred y valores de tiempo de vida para las direcciones creadas con el prefijo anunciado. La generación periódica de Avisos de Router permite que los *hosts* actualicen la información recibida también de manera periódica.

En definitiva, el proceso de autoconfiguración sin estado se apoya en la definición de direcciones IPv6 de alcance de enlace local, la posibilidad de comunicación *multicast*, la definición de un protocolo conocido con el nombre de Descubrimiento de Vecino (ND, Neighbor Discovery) y la posibilidad de

generar un identificador de interfaz a partir de la dirección física existente. Con estos elementos se puede generar una dirección de carácter temporal, para luego crear una dirección permanente, cuando se puedan determinar los prefijos de la red donde se encuentra el dispositivo en cuestión.

La ventaja más importante de esta posibilidad de autoconfiguración es el soporte de movilidad IP: los dispositivos pueden cambiar de red, generando direcciones válidas sin necesidad de conocer de antemano la existencia de servidores o los prefijos de red.

10.7 ICMPv6

Esta nueva versión del protocolo ICMP quedó especificada en la RFC 2463. Allí, como en el caso de IPv4, se definen dos tipos de mensajes: mensajes de error y mensajes de información. Todos los mensajes comparten un formato común que consiste en tres campos:

- **Tipo** ($1\ byte$): indica el tipo de mensaje, determinando su valor el formato de los datos. Los mensajes de error llevan el bit de mayor orden del campo Tipo fijado en "0". Es decir que los mensajes de información llevan números entre "128" y "255" en dicho campo. En realidad, la RFC 2463 define cuatro mensajes de error y sólo dos mensajes de información. Los mensajes de error son Destino Inalcanzable, Paquete Demasiado Grande, Tiempo Excedido y Problema de Parámetros. Los mensajes de información son Requerimiento de Eco y Respuesta de Eco.

- **Código** ($1\ byte$): depende del tipo de mensaje, sirve para dar mayor granularidad a la definición.

- **Chequeo** ($2\ bytes$): protección de error para el mensaje ICMPv6.

Cualquier mensaje ICMPv6 se encapsula en un datagrama IPv6 que puede llevar cabeceras de extensión. El valor de Siguiente Cabecera con que se reconoce estos mensajes es el número "58". Un nodo que envíe un mensaje ICMPv6 debe determinar las direcciones fuente y destino del encabezado IPv6 mediante una serie de reglas enunciadas en la RFC 2463. También se incluyen reglas para la generación y procesamiento de los mensajes ICMPv6.

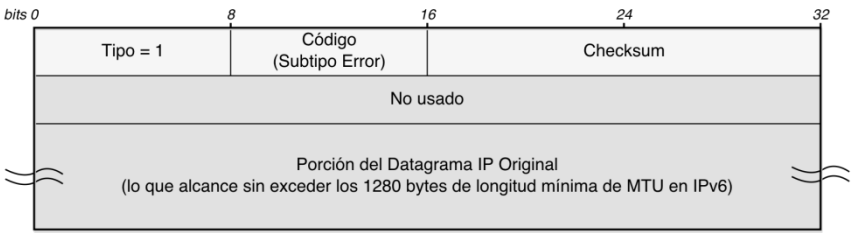


Figura 10.9 - Mensaje ICMPv6 Destino Inalcanzable.

En la Fig. 10.9, se presenta el formato del mensaje **Destino Inalcanzable**. Estos mensajes deberían ser generados por los *routers* como consecuencia de no haber podido entregar un paquete a destino por razones no debidas a una situación de congestión. El campo Código permite distinguir entre los casos de ausencia de ruta a destino, imposibilidad de entrega por filtrado de mensajes, dirección inalcanzable o puerto inalcanzable, pero lo más importante de este mensaje es la carga, en la porción de datos, del datagrama original que produjo el error. El procesamiento en el receptor de estos mensajes es similar al caso de IPv4.

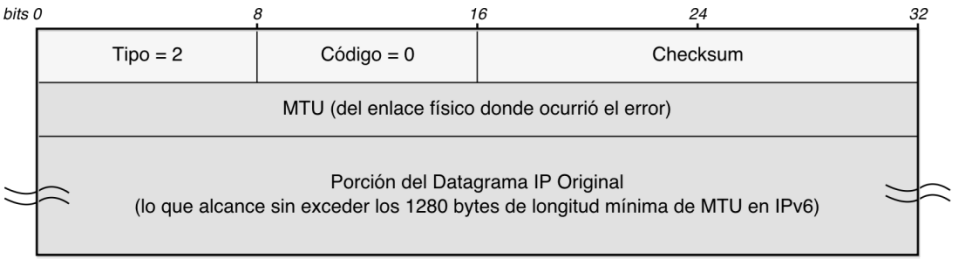


Figura 10.10- Mensaje ICMPv6 Paquete Demasiado Grande.

La Fig. 10.10 presenta el formato del mensaje **Paquete Demasiado Grande** que debe ser enviado por un *router* en respuesta a un paquete que no pudo ser re-enviado porque su tamaño es superior a la MTU del enlace saliente. Recordemos que IPv6 no permite la fragmentación en nodos intermedios. Por este motivo, se precisa un mecanismo adicional para evitar la fragmentación, descripto en la RFC 1981: el descubrimiento de la MTU mínima de un camino entre fuente y destino. La propia información que carga este mensaje se usa como parte del mecanismo de descubrimiento de la MTU.

En la Fig. 10.11 se presenta el formato del mensaje **Tiempo Excedido**. Recordemos que el encabezado IPv6 lleva un campo de Límite de Saltos, similar al de TTL de IPv4. Cuando, para cualquier datagrama en tránsito, la cuenta de saltos llega a cero, el mismo se descarta y se debe enviar un mensaje Tiempo Excedido al dispositivo de la dirección fuente del datagrama, es decir el nodo que lo generó. Como en el caso de IPv4, también se pueden generar este tipo de mensajes en los casos en que hubo fragmentación y el destino no pudo terminar de re-ensamblar el paquete original por pérdida de uno o más fragmentos.

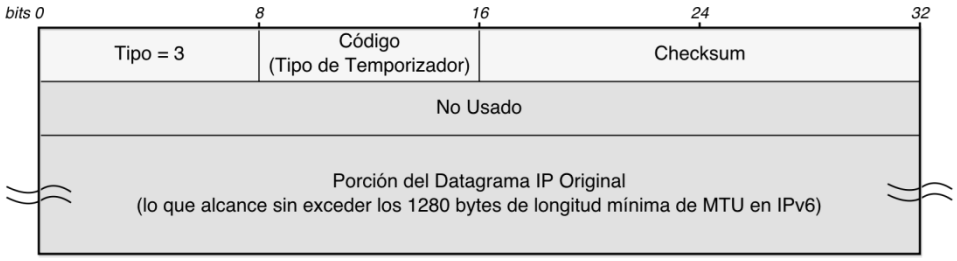


Figura 10.11- Mensaje ICMPv6 Tiempo Excedido

El formato del mensaje **ICMPv6 Problema de Parámetros** se presenta en la Fig. 10.12. Este mensaje se ideó para cubrir cualquier clase de error que los mensajes descriptos previamente no fueran capaces de considerar. El campo de Código sirve para distinguir tres situaciones particulares: campo de encabezado incorrecto, siguiente cabecera no reconocida y opción IPv6 no reconocida. El propio mensaje lleva un Puntero que indica al campo del encabezado del datagrama que generó el problema.

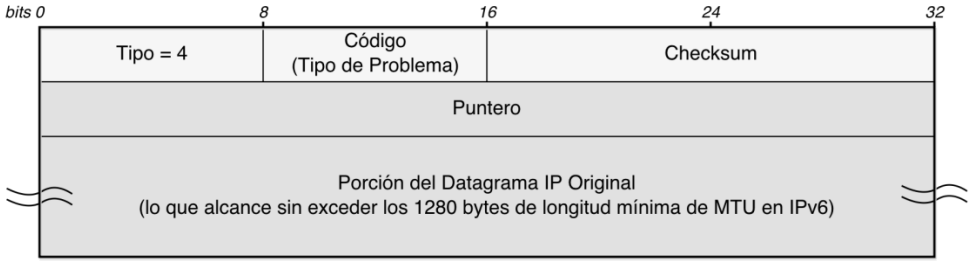


Figura 10.12- Mensaje ICMPv6 Problema de Parámetros

La Fig. 10.13 presenta el formato del mensaje de **Eco**. Su propósito es el mismo que en IPv4: realizar chequeos para determinar si un nodo es alcanzable. Como en ICMPv4, para adaptar la respuesta al requerimiento, se incluyen los campos Identificador y Número de Secuencia. Ambos mensajes se utilizan en la versión de *ping* para IPv6, a veces llamada *ping6*.

Además de los mensajes de error mencionados, posteriormente se definieron seis tipos más de mensajes de información: Aviso de Router y Solicitud de Router, Aviso de Vecino y Solicitud de Vecino, Re-direccionar y Re-numeración de Router. La mayoría de estos mensajes forman parte del protocolo de Descubrimiento de Vecino ND, definido en la RFC 2461. A continuación se presentarán los mensajes mencionados y, en el próximo apartado, se desarrollarán algunos conceptos importantes del protocolo ND.

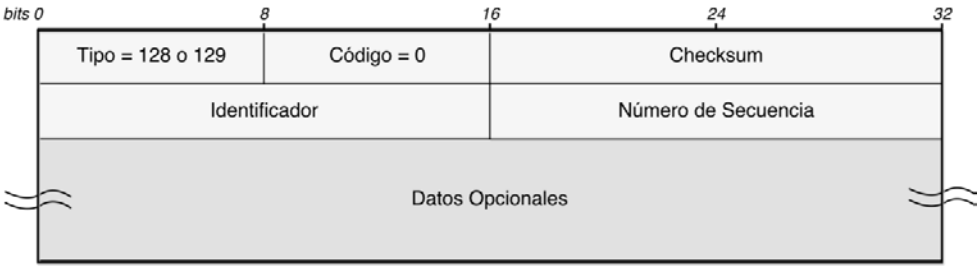


Figura 10.13- Mensaje ICMPv6 Requerimiento/Respuesta de Eco.

Ningún dispositivo puede usar Internet sin conocimiento del *router* que le oficiará de *gateway*. Para tal efecto, se ideó una técnica de descubrimiento para que cualquier dispositivo pueda localizar el *router* de salida a Internet y conocer los parámetros relacionados con la operación en la red local. En este sentido, los *routers* envían periódicamente mensajes de Aviso de Router y también deben estar atentos a los pedidos de Solicitud de Router enviados por los *hosts*.

En la Fig. 10.14 se presenta el mensaje de **Aviso de Router**. Los mensajes de Aviso los transmiten los *routers*, de manera periódica, para informar a los *hosts* de su propia presencia y características y enviarles parámetros que precisan para su funcionamiento. Como todos los mensajes anteriores, comparte un formato inicial de campos Tipo, Código y Chequeo.

A continuación se presenta un campo de 1 *byte* donde se carga el valor del Límite Actual de Salto, que es el valor default que el *router* aconseja a los dispositivos para su carga en el campo Límite de Saltos de las cabeceras IPv6. Un valor "0" en este campo implicaría que no hay valor recomendado.

Luego, el mensaje de Aviso presenta otro campo de 1 *byte* denominado Banderas. En dicho campo, los 6 *bits* últimos quedan reservados y los 2 *bits* en el inicio permiten que el *router* avise a los *hosts* el método de autoconfiguración utilizado en la red local. El bit M en alto indica el uso de un método con estado, tal como DHCP, para la configuración de direcciones. El bit O en alto indica el uso de un método con estado para información adicional a la propia dirección.

A continuación, un campo de 2 *bytes* denominado Tiempo de Vida del Router establece el tiempo, en segundos, durante el cual este *router* se puede tomar como default. El valor "0" es un indicador de que el *router* no debería usarse como *router* default.

El campo de 4 *bytes* que sigue, de Tiempo Alcanzable, anuncia a los dispositivos por cuánto tiempo, medido en milisegundos, deberían considerar a un vecino alcanzable, luego de haber recibido información al respecto.

El campo siguiente, denominado Tiempo de Retransmisión, también de 4 *bytes*, es el tiempo en milisegundos que el dispositivo debería esperar antes de hacer una retransmisión.

Por último, aparece un campo de Opciones, que puede presentar tres tipos de información: la dirección de capa de enlace del *router*, la MTU de la red local e información de prefijo para la red local, análoga al prefijo de subred de IPv4.

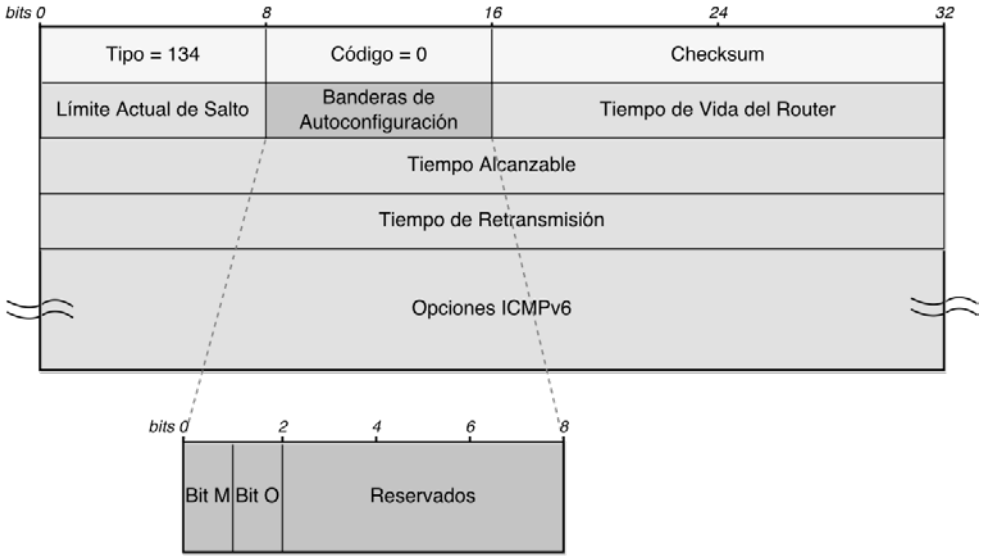


Figura 10.14 - Mensaje ICMPv6 de Aviso de Router.

La Fig. 10.15 presenta el mensaje de **Solicitud de Router**, mucho más sencillo que el anterior. Cuando un *host* desea saber, de manera inmediata, si hay *routers* presentes, pueden transmitir este mensaje, obligando a los *routers* presentes a contestar con un mensaje de Aviso de Router. El mensaje de solicitud sólo incluye un campo de Opciones que podría servir para indicar la dirección a nivel de capa de enlace del dispositivo que envía la solicitud. Estos mensajes se suelen enviar a la dirección *multicast* de todos los *routers*, mientras que los mensajes periódicos de Aviso se envían a la dirección *multicast* de todos los nodos. La única excepción se presenta en el caso en que se transmitan en respuesta a una Solicitud, en cuyo caso se dirigen a la dirección *unicast* del *host* que envió el requerimiento.

Además de los mensajes mencionados, el protocolo ND utiliza los mensajes ICMPv6 de Solicitud de Vecino y Aviso de Vecino para el descubrimiento de *hosts*. El mensaje de Solicitud de Vecino permite corroborar que un vecino existe y es posible de alcanzar, para iniciar la resolución de direcciones. El mensaje de Aviso de Vecino sirve para confirmar la presencia de un *host* o de un *router*, transportando información de nivel de enlace cuando sea necesario.

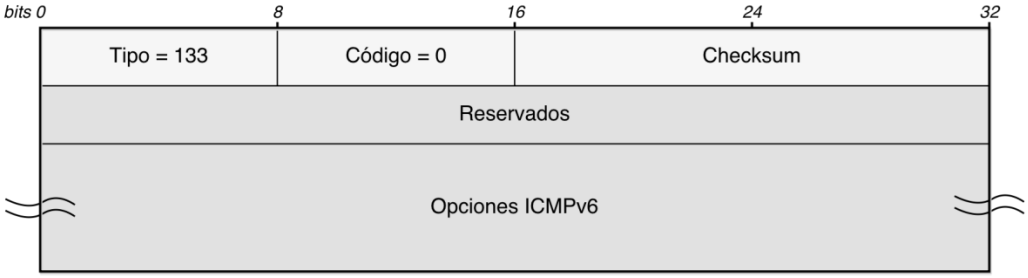


Figura 10.15- Mensaje ICMPv6 de Solicitud de Router.

La Fig. 10.16 presenta el mensaje de **Aviso de Vecino**. Este mensaje lleva un campo de 4 $bytes$ con sólo tres banderas definidas. La bandera R se enciende cuando el mensaje lo transmite un *router* y se ajusta en "0" si lo hace otro dispositivo. La bandera S se enciende en los mensajes de respuesta a los de Solicitud de Vecino. La bandera O, cuando está encendida, significa que se debe sobre-escribir cualquier información almacenada en caché, referida al nivel de enlace de un dispositivo, con la información transportada por este mensaje. Generalmente va encendida en mensajes de Aviso de Vecino no solicitados, que se transmiten para forzar cambios.

El campo de 16 $bytes$ denominado Dirección de Blanco se corresponde al campo con el mismo nombre en el mensaje de Solicitud, cuando el mensaje de Aviso de Vecino es en respuesta al mismo. Generalmente, se carga con la dirección IPv6 del dispositivo que transmite el Aviso de Vecino.

Un mensaje de Aviso de Vecino editado como respuesta a una Solicitud *multicast*, lleva como opción la dirección de nivel de enlace del que lo transmite. Si es en respuesta a una Solicitud *unicast*, normalmente se incluye la opción para refresco del caché del solicitante, aunque esto no sería necesario por la forma en que se generó la solicitud.

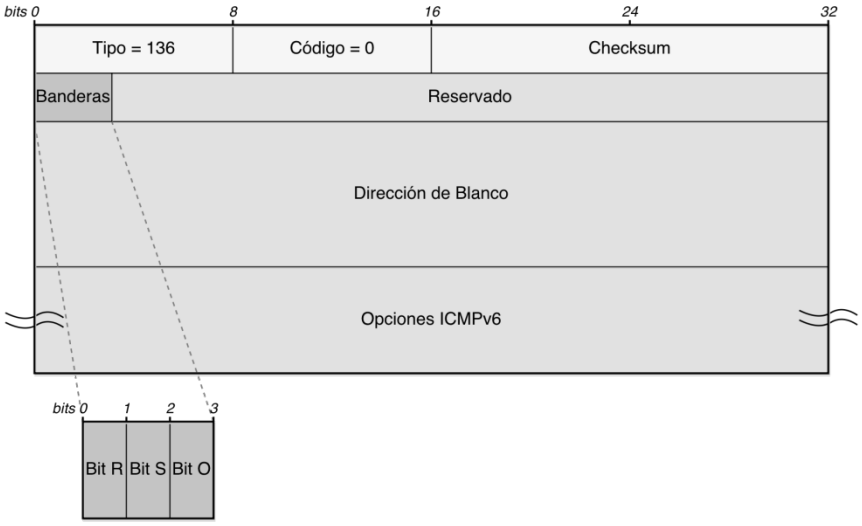


Figura 10.16- Mensaje ICMPv6 de Aviso de Vecino.

La Fig. 10.17 presenta un mensaje de **Solicitud de Vecino**, que carga como Dirección de Blanco, la dirección *unicast* IPv6 del dispositivo cuya dirección de nivel de enlace se está tratando de resolver. Entre las opciones, se pueden incluir las dos direcciones del solicitante: la dirección IPv6 y la dirección a de nivel de enlace. Estos mensajes se envían a la dirección *unicast* del dispositivo que se desea resolver, pero también se pueden enviar a la dirección *multicast* de nodo solicitado.

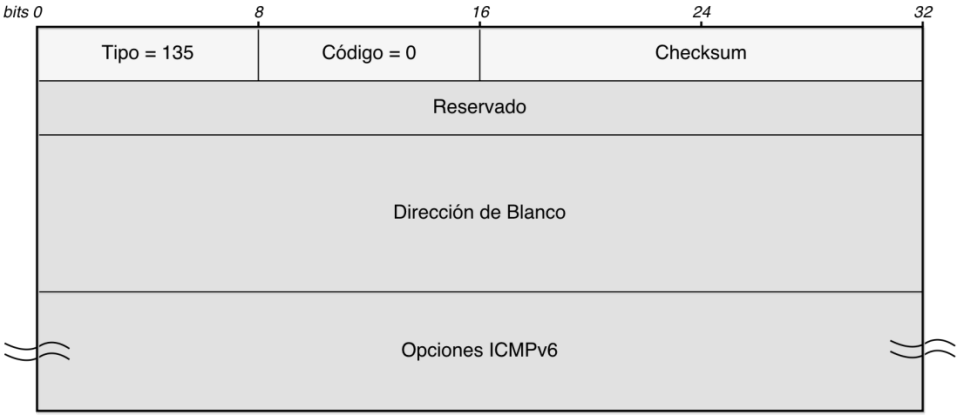


Figura 10.17- Mensaje ICMPv6 de Solicitud de Vecino.

Esta última dirección es una dirección especial que se usa para que un dispositivo envíe por *multicast* un datagrama que será recibido por el nodo cuya dirección está tratando de resolver.

En el caso de un mensaje de Aviso de Vecino generado en respuesta a una solicitud, el mismo se envía por *unicast* al nodo que generó el pedido, a menos que el mensaje se haya enviado desde una dirección no especificada, en cuyo caso se transmite a la dirección *multicast* de todos los nodos.

Si el mensaje de Aviso de Vecino no es en respuesta a una solicitud, se transmite a la dirección *multicast* de todos los nodos.

En la Fig. 10.18 se presenta un mensaje ICMPv6 denominado **Re-Direccionar**. Estos mensajes también existían en la versión cuatro del protocolo, pues fueron ideados para cargar líneas en las Tablas de Ruteo de aquellos dispositivos que arrancaban sin información de ruteo. ICMPv6 rescata la utilización de estos mensajes dentro del contexto del protocolo ND.

Un *router* puede enviar un mensaje Re-Direccionar por *unicast* a un *host*, cuando es capaz de interpretar que dicho dispositivo tiene una ruta mal cargada hacia un destino. El sentido de mal cargada en este contexto, es la certeza de que existe una ruta más eficiente para ese destino. En estos mensajes, un campo denominado Dirección de Blanco presenta la dirección del nuevo *router* para la ruta. Otro campo, Dirección Destino, se carga con la dirección del dispositivo destino que disparó el mensaje, es decir la línea de la Tabla a modificar. En el campo de Opciones, se puede cargar la dirección del nivel de enlace del nuevo *router*, para evitar pasar por un proceso de resolución de direcciones, y el encabezado IPv6 del datagrama que generó este mensaje.

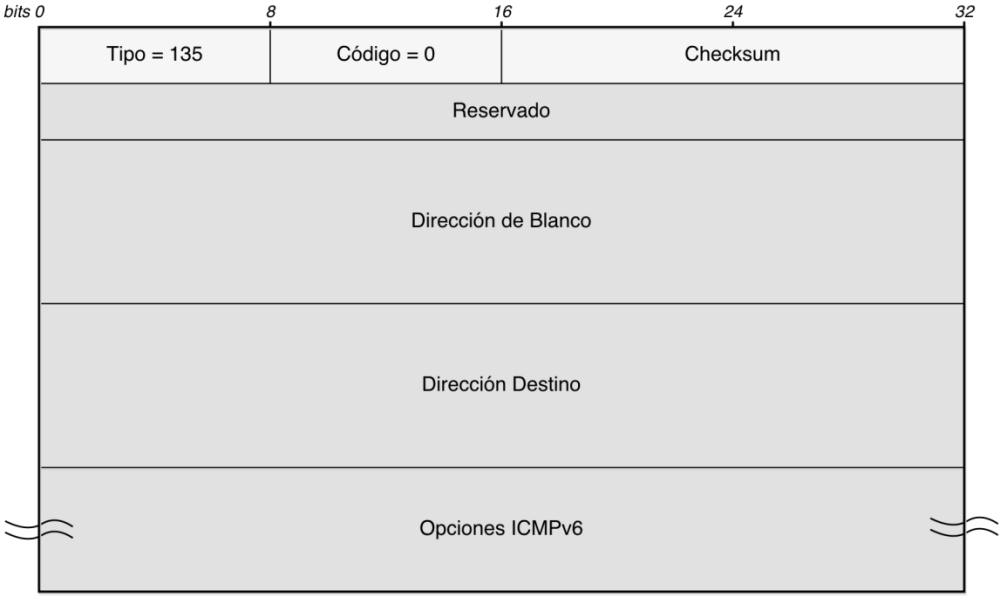


Figura 10.18 - Mensaje ICMPv6 de Re-Direccionar.

Por último, la Fig. 10.19 presenta el mensaje **Re-numeración de Router**. Este mensaje se creó para facilitar tareas de re-numeración de *routers* en el caso de migración entre proveedores, enviándose a la dirección *multicast* de todos los routers. La RFC 2894 describe una técnica que permite la re-numeración de los *routers* dentro de un Sistema Autónomo, mediante el reparto de nuevos identificadores de red.

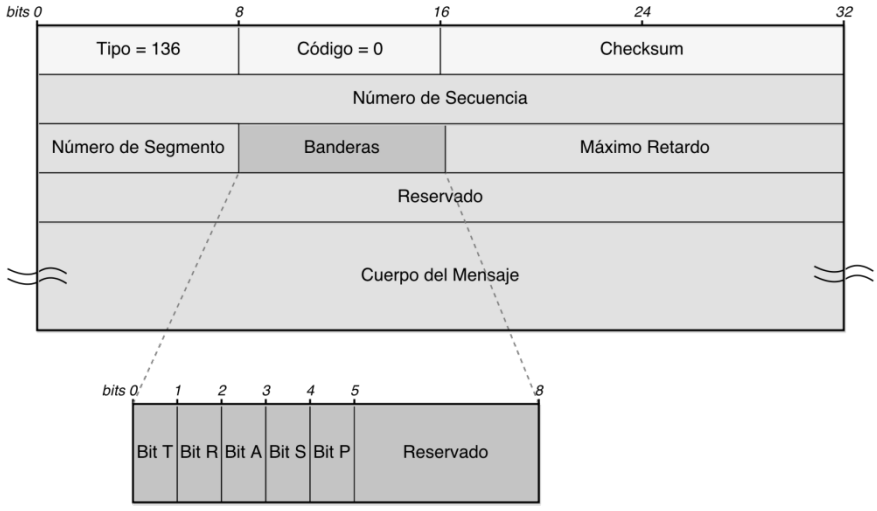


Figura 10.19- Mensaje ICMPv6 de Re-numeración de Router.

El administrador de la red designa un dispositivo para generar uno o más mensajes, denominados Comandos de Re-numeración, que proveen un listado de prefijos de aquellos *routers* que serán re-numerados. Todos los *routers* revisan estos mensajes, buscando encontrar prefijos de sus propias interfaces en el listado. De ser así, deben modificarlos por los nuevos, que también se especifican en el Comando. A veces los comandos cargan información adicional referida al momento y la forma en que se procederá a la re-numeración. Otras veces pueden solicitar respuestas, para verificar los resultados.

El mensaje de Re-numeración cuenta con un campo de 32 $bits$ en el que se carga un Número de Secuencia por cuestiones de seguridad, para evitar ataques de *replay*. De no existir esta protección, un atacante podría inyectar mensajes falsos de re-numeración, mediante duplicados o mensajes fuera de orden. A su vez, un campo Número de Segmento, de 8 $bits$, sirve para diferenciar mensajes de re-numeración con el mismo Número de Secuencia, pero que son válidos.

La bandera T indica que la re-numeración se realizará en un entorno de prueba, como si fuera una simulación de la misma. La bandera R sirve para exigir una respuesta como resultado de la re-numeración. La bandera A levantada indica que el proceso debe hacerse en todas las interfaces. La bandera S, cuando está encendida, implica que se trata de un requerimiento de re-numeración sólo para aquellas interfaces que se encuentren en el mismo sitio que la que recibió el

pedido. La bandera P sirve para indicar que ya se ha procesado un pedido previo de la misma naturaleza y, entonces el presente, no se procesará.

A continuación, un campo de $2\,bytes$ presenta el tiempo en milisegundos del Máximo Retardo que se autoriza a permanecer al dispositivo bajo proceso de re-numeración sin emitir una respuesta.

En el campo denominado Cuerpo del Mensaje figuran pares de valores, indicando el prefijo a re-numerar y el nuevo prefijo. Esta información es la que usan los *routers* para contrastar con sus propios prefijos y cambiar, de acuerdo a si figuran en el listado o no. En el cuerpo del mensaje de resultado, cada *router* explicita las interfaces que se han visto modificadas con resultado exitoso.

Los mensajes de re-numeración se suelen enviar a la dirección *multicast* de todos los *routers*, ya sea de entorno de enlace local o de sitio local.

10.8 Protocolo de Descubrimiento de Vecinos ND

El Protocolo de Descubrimiento de Vecinos integra muchas de las funciones referidas a la comunicación local, que se realizaban a través de varios protocolos en entornos IPv4. Definido en la RFC 2461, este protocolo utiliza varios de los mensajes ya vistos en ICMPv6 y especifica el concepto de vecino como el de dispositivos pertenecientes a un mismo enlace local, que se pueden enviar datos de manera directa.

El protocolo ND no solo ayuda a encontrar los dispositivos vecinos en una red, sino también colabora con la instalación de la funcionalidad necesaria para la conectividad local, el enrutamiento de datagramas y la configuración de los nodos, ya sean estos *routers* o *hosts*. El término vecino puede referirse a cualquiera de estos dispositivos, aunque el protocolo presenta un trato diferencial entre ellos, distinguiendo diferentes funcionalidades para los casos de descubrimiento entre *host* y *routers* y para los casos de comunicación entre *hosts* solamente.

Entre las funciones de descubrimiento, se define métodos para que los *hosts* localicen *routers* en su propia red, descubran los prefijos de red que les permitan entregar paquetes de manera directa o indirecta, se enteren de parámetros importantes de la red local, tales como la MTU, y se puedan auto-configurar de manera automática.

Entre las funciones de comunicación, se definen métodos para poder resolver direcciones, descubrir direcciones de próximo salto en cuanto al envío de datagramas, detectar si un vecino se encuentra directamente conectado y poder determinar si la dirección elegida para configuración se encuentra duplicada dentro del entorno local.

Realmente, ND implementa su funcionalidad a través de los siguientes mensajes ICMPv6:

- **Aviso de Router**: transmitidos regularmente por *routers* para que los *hosts* sepan de su existencia, y para proveerles información de prefijo de red y otros parámetros.

- **Solicitud de Router**: transmitidos por *hosts* para solicitar a cualquier *router* local que le envíe un mensaje de Aviso de Router, y no tener que esperar el siguiente de estos mensajes periódicos.
- **Aviso de Vecino**: transmitidos por *hosts* para indicar su existencia y proveer al resto sobre información propia.

- **Solicitud de Vecino**: transmitidos para verificar la existencia de otro *host* y solicitarle el envío de un Aviso de Vecino.

- **Mensajes de Re-direccionamiento**: transmitidos por un *router* para dar a conocer a un host sobre una mejor manera de encaminar datos hacia un destino.

El protocolo ND propone que, antes de que un *host* pueda comunicarse en red, encuentre algún *router* local que le permita conocer más información sobre su entorno. Este proceso de descubrimiento se realiza mediante los mensajes de Aviso y Solicitud de Router. Estos mensajes, por su parte, pueden incluir opcionalmente direcciones de capa física del nodo que los transmite. La presencia de esta información extra colabora con la aceleración del proceso de resolución de direcciones.

Aunque la entrega directa de paquetes entre dispositivos de la misma red no requiere el uso de *routers*, el protocolo ND separa una funcionalidad para determinar la entrega de próximo salto en el entorno local. Esta funcionalidad se refiere al procesamiento que realiza un *host* cuando examina la dirección destino de un datagrama y decide si debe realizarse una entrega directa o indirecta para el mismo. En IPv6 esta decisión se toma comparando la información de prefijo obtenida de los *routers* locales con el propio prefijo de red del datagrama. De coincidir, se decide por la entrega directa. Si no coinciden, se elige la dirección de próximo salto a partir de una lista de *routers* locales, que puede haber sido configurada por un administrador u obtenida a partir de la fase de descubrimiento del propio protocolo. Cada vez que el host realiza una decisión de este tenor, guarda la información pertinente en una memoria caché, para que las siguientes decisiones se realicen de manera más eficiente.

La entrega directa de datagramas, por su parte, requiere el conocimiento de la dirección física de un vecino. El protocolo ND provee un mecanismo de resolución de direcciones mediante la transmisión de mensajes de Solicitud de Vecino que llevan la dirección IPv6 del dispositivo cuya dirección de nivel de enlace se desea conocer. Este mensaje es respondido por el nodo en cuestión mediante el mensaje de Aviso de Vecino, que se dirige a una dirección *multicast* especial del dispositivo, conocida como la dirección de nodo solicitado. Este intercambio de mensajes puede presentarse entre *hosts* o entre un *host* y un *router*. Además, el protocolo permite que un dispositivo transmita un Aviso de Vecino,

sin necesidad de una solicitud, en el caso de cambios en las direcciones de hardware.

Otra de las funcionalidades agregadas a los mensajes de Aviso y Solicitud de Vecino se relaciona con la detección de dispositivo alcanzable. Todos los dispositivos mantienen información actualizada referida a sus vecinos. El conocimiento de que un vecino no es alcanzable es importante en el caso que afecte el comportamiento de otros, como sería el caso de un *router* utilizado como dispositivo de próximo salto. Por este motivo, el caché de información de vecinos lleva asociado un tiempo de vida, que se anuncia en los mensajes de Aviso de Router. También se puede actualizar el caché al recibir respuestas a mensajes de Solicitud de Vecino.

Cabe aclarar que los mensajes de Aviso y Solicitud de Vecino también se ocupan de la funcionalidad de detección de direcciones duplicadas, en la fase del proceso de autoconfiguración.

Algunas de las mejoras del protocolo ND respecto de ideas ya presentes en IPv4 son: la formalización de un mecanismo para descubrir un *router*, el reemplazo del viejo protocolo ARP por un proceso integrador que utiliza *multicast* en vez de *broadcast* y la mejora del mecanismo de re-direccionamiento, pudiendo incluirse en todos los casos autenticación y criptografía a nivel de red. Además, la posibilidad de realizar el proceso de auto-configuración con ND hace innecesaria la presencia de servidores DHCP. No menos importante es la posibilidad de detectar continuamente la presencia o ausencia de vecinos, permitiendo seleccionar rutas de modo dinámico.

10.9 Transición IPv4 a IPv6

Uno de los mayores desafíos de las redes configuradas con IPv4 es la planificación de una migración a IPv6. La clave para una transición exitosa es la compatibilidad con la gran base instalada de dispositivos IPv4, al mismo tiempo que se despliega IPv6. La RFC 4213, de Octubre de 2005, especifica dos mecanismos de compatibilidad IPv4 que pueden ser usados por dispositivos IPv6: doble pila y túnel. Otro mecanismo, denominado traducción, intenta convertir direcciones entre los distintos espacios.

10.9.1 Configuración de doble pila

La manera más sencilla para que un nodo IPv6 sea compatible con otro que sólo reconoce IPv4, es proveer al primero de una implementación IPv4. Los nodos IPv6 que además contienen una implementación IPv4 se conocen como nodos IPv4/IPv6 o nodos de doble pila, tal como se representa en la Fig. 10.20. Estos nodos pueden enviar y recibir datagramas de ambos protocolos, operar con

nodos IPV4 mediante paquetes IPv4 y operar con nodos IPv6 usando paquetes IPv6.

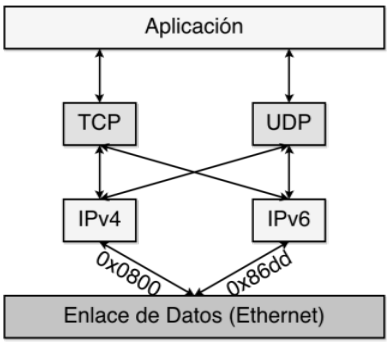


Figura 10.20- Configuración Dual IP.

Existen tres maneras de operar con estos nodos: con la pila IPv4 habilitada y la pila IPv6 deshabilitada, al revés, o con ambas pilas habilitadas. En el último caso, cuando el nodo se encuentra en una red sólo IPv4 o sólo IPv6, debe utilizarse algún mecanismo de túnel para manejar tráfico sobre la otra versión de IP.

Un cliente doble pila puede conectarse con un servidor sólo IPv4, con un servidor sólo IPv6 o con servidores doble pila. Un servidor doble pila puede aceptar conexiones de cualquier tipo de cliente. Se trata de un mecanismo muy flexible, pero con cierta complejidad a nivel de la implementación y los requerimientos de memoria, por lo que podría no ser una opción en el caso de ciertos dispositivos pequeños.

Una cuestión muy importante es que la aproximación de doble pila requiere la inclusión de librerías de resolución de nombres capaces de reconocer registros de resolución de nombre DNS tipo A y AAAA y sus inversos. Los registros tipo A retornan direcciones IPv4, en tanto que los registros tipo AAAA retornan direcciones IPv6. Cuando se solicita a DNS por el nombre de un nodo dual, el orden de las respuestas es un indicador del protocolo a usar, pero son las aplicaciones que reciban las respuestas las que determinarán el uso de IPv4 o de IPv6.

Dado que la mayoría de los servidores y clientes actuales tienen soporte para operación doble pila, este mecanismo de transición parece apropiado como herramienta a largo plazo, aunque presenta algunas reservas. Para que sea efectivo, es importante que existan suficiente cantidad de direcciones IPv4 para los ISP y sus clientes, pero también es preciso tener disponibilidad de direcciones IPv6. Esto es debido a que los dispositivos doble pila precisan ambos tipos de direcciones.

En caso de que el ISP no cuente con suficiente cantidad de direcciones IPv4, se deberá usar NAT. La situación de redes doble pila con dispositivos con direcciones IPv6 globales e IPv4 privadas traducidas se soluciona con Carrier-Grade NAT (CGN), NAT44(4), NAT64, NAT464, y Dual-Stack Lite.

10.9.2 Configuración de Túnel

Mientras no exista una estructura de enrutamiento IPv6, se deben aportar soluciones para transportar datagramas IPv6 sobre la infraestructura IPv4 desplegada. *Hosts* y *routers* con capacidades duales IPv4/IPv6 pueden generar túneles para el paso de datagramas IPv6 sobre topologías de enrutamiento IPv4, por encapsulado de dichos paquetes en IPv4, tal como se presenta en la Fig. 10.21 y se describe en la RFC 4213. En estos casos, el campo Protocolo del encabezado IPv4 se rellena con el número 41, conociéndose el método como encapsulado de Protocolo 41 y también 6in4. Este tipo de túnel requiere que ambos extremos tengan direcciones IPv4 globalmente enrutables, es decir que no pueden encontrarse detrás de un NAT.

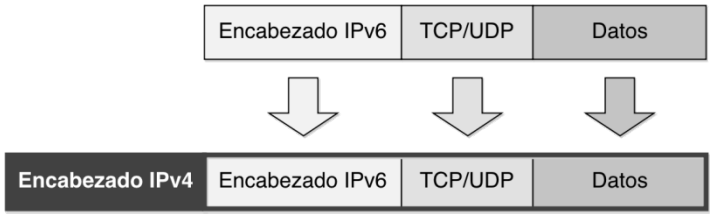


Figura 10.21 – Concepto de Túnel

Existen muchas formas de túneles, según los extremos que los conforman.

Un túnel **router a router** se establece entre *routers* duales IPv6/IPv4, interconectados por una estructura IPv4. El propósito es pasar paquetes encapsulados entre ellos. El túnel es sólo un segmento del camino extremo a extremo que el paquete transita.

Los túneles **host a router** son establecidos por *hosts* duales IPv6/IPv4 que pasan paquetes IPv6 encapsulados a *routers* intermedios IPv6/IPv4, alcanzables por una infraestructura IPv4. El túnel es el primer segmento del camino extremo a extremo que el paquete transita. El caso contrario involucra un túnel **router a host** que se establece en el caso de *routers* IPv6/IPv4 que transmiten paquetes IPv6 hacia un destino final, representado por un *host* IPv6/IPv4. En este caso el túnel es el último segmento del camino del paquete.

Un túnel **host a host** se establece entre *hosts* IPv6/IPv4 interconectados por una infraestructura IPv4, para que puedan transmitir paquetes IPv6 entre ellos. En estos casos el túnel es el camino extremo a extremo que transita el paquete. La utilidad de este tipo de túnel es importante en los casos de dispositivos duales aislados.

Los **túneles configurados** se basan en la configuración manual del otro extremo del túnel: la dirección IPv4 del extremo al final del túnel. Pueden usarse dentro de la red de una organización, entre redes IPv6 pequeñas que se encuentran aisladas, aunque generalmente son usados por servicios de *brokers*. Entre los *brokers* más conocidos, se puede mencionar Hurricane Electric IPv6 Tunnel

Broker, que ofrece un servicio gratuito para conexiones desde hosts o *routers* IPv6 con conectividad IPv4, a los propios *routers* IPv6 de la empresa. Las denominaciones más comunes para los túneles configurados son: túneles manuales, túneles estáticos, túneles de protocolo 41 o 6in4. El propósito de estos túneles es ofrecer una conexión del tipo punto a punto, fija, entre dos sistemas, usando el encapsulado mencionado.

Los **túneles automáticos** pueden ser usados por dispositivos que utilicen direcciones IPv6 compatibles con IPv4, del tipo $::a.b.c.d$. Cuando se transmiten datagramas con dirección destino compatible, la dirección IPv4 destino, en el encabezado IPv4 que encapsula, se toma de la dirección compatible. La limitación en este caso es que ambos extremos con funcionalidad IPv6 deben soportar el túnel automático.

Un túnel automático no requiere pre-configuración. Algunos de los túneles de configuración automática más conocidos son:

- **ISATAP**: túnel automático entre *hosts* o entre *host* y *router*, basado en un formato particular de dirección IPv6 con inclusión de dirección IPv4 embebida, que sirve para conectar dispositivos IPv6 a través de una intranet IPv4. Se describe en la RFC 4214.

- **6to4**: se trata de un túnel automático *router* a *router* o *host* a *router*, basado en un prefijo global particular y el uso de direcciones IPv4 embebidas para permitir comunicación entre *hosts* IPv6 separados por redes Internet IPv4. Se describe en la RFC 3056.

- **Teredo**: túnel automático entre hosts IPv6 a través de redes Internet IPv4. Los *hosts* pueden encontrarse detrás de un NAT y, para atravesarlo, Teredo envía los paquetes IPv6 en datagramas UDP.

- **6over4**: túnel *host* a *host* automático, para conectar dispositivos IPv6 aislados usando *multicast* IPv4. Descripto en RFC 2529.

- **6rd**: permite el despliegue de IPv6 a través de redes de ISP con infraestructura IPv4. Deriva de 6to4 pero soluciona los problemas de arquitectura tan importantes en ese mecanismo.

El **Protocolo de Direccionamiento tipo Túnel Automático Intra-Sitio (ISATAP, Intra-Site Automatic Tunnel Addressing Protocol)** es un método de configuración automática para túneles entre *hosts* y *routers* o sólo entre *hosts*. ISATAP usa un prefijo de red de $64\ bits$ a partir del cual se arman las direcciones ISATAP en dispositivos de doble pila. El identificador de interfaz se genera concatenando $::5efe$ y las direcciones IPv4 de los nodos. Así, un nodo con dirección 200.0.182.38 posee un identificador ISATAP del tipo $::5efe:200.0.182.38$, al que se le pueden agregar prefijos para completar la dirección IPv6. Por ejemplo $fe80::5efe:200.0.182.38$ sería la dirección de enlace local IPv6 del nodo ISATAP indicado.

Los *hosts* con soporte ISATAP mantienen una lista de *routers* con soporte ISATAP, que obtienen mediante mensajes IPv4 de Solicitud de Router. Cuando desean comunicarse con un *host* IPv6, arman un paquete IPv6 cuya dirección destino obtienen a partir del servicio DNS y cuya dirección fuente es la derivada a partir del identificador ISATAP. Luego encapsulan el paquete con un encabezado IPv4 cuya dirección fuente es la IPv4 utilizada para derivar el identificador ISATAP y la dirección destino es la IPv4 del *router* ISATAP. Este paquete también lleva el campo Protocolo 41. El *router* ISATAP procederá al des-encapsulado y ruteo IPV6 del paquete original. La respuesta IPv6 se reconocerá por el prefijo global de la dirección ISATAP destino, y será entregado al *router* ISATAP para que proceda a su encapsulado para la entrega final.

ISATAP se encuentra implementado en varios Sistemas Operativos, pero presenta problemas de seguridad ya que no ofrece mecanismo de autenticación, permitiendo que cualquier *host* doble pila que conozca la existencia del *router* pueda obtener su servicio.

6to4 es una técnica de túnel de paquetes IPv6 encapsulados en paquetes IPv4, definidos en la RFC 3056, que se apoya sobre un formato particular de dirección IPv6 para identificar los paquetes que pasarán por el túnel 6to4. La dirección comienza con el prefijo global 6to4, $2002::/16$, seguido de la dirección IPv4 pública para el sitio destino final. Por ejemplo, si el sitio tiene un *router* con interfaz de salida IPv4 $200.0.182.38$, el prefijo será $2002:c800:b708/48$. La dirección IPv4 debe ser pública, para que el bloque IPv6 derivado de ella sea globalmente único. De este modo, un *router* con salida a Internet y dirección IP pública, crea un prefijo $/48$ y lo comunica a los hosts de su red mediante los mensajes de Aviso de Router, para que los hosts con capacidad IPv6 configuren una dirección IPv6 basada en este prefijo. En los hosts 6to4 se agregará una ruta a este prefijo y una default al router 6to4 de su propia red.

Cuando un *host* 6to4 requiera comunicarse con otro *host* 6to4 fuera de su red, ambas direcciones, fuente y destino, comenzarán con el prefijo $2002::/16$, seguido de la dirección IPv4 del *router* de salida escrita en formato IPv6. Los bits restantes de la dirección IPv6 se corresponderán con el SubnetID interno de cada sitio y el identificador de interfaz propio. El *host* 6to4 origen editará un paquete IPv6 con dirección destino correspondiente a la del *host* 6to4 con el que desee comunicarse. El *router* de salida reconocerá en la dirección destino el otro extremo del túnel y encapsulará los paquetes IPv6 en paquetes IPv4, que destinará a la interfaz IPv4 del *router* del otro extremo. En el extremo final del túnel, el *router* 6to4 des-encapsulará el paquete y re-enviará el datagrama IPv6 original al *host* 6to4 destino.

De este modo, se permite que dispositivos aislados IPv6 se comuniquen a través de túneles automáticos sobre una red IPv4 que no soporta IPv6, con la única condición que el *router* de borde posea una dirección IPv4 globalmente única en la interfaz que conecta con la red IPv4.

El método 6to4 también reconoce componentes denominados *routers relay* 6to4, que permiten la comunicación entre redes Internet IPv4 y redes del mismo tipo IPV6. Estos dispositivos conforman túneles con *routers* 6to4, que los

apuntan en sus tablas de ruteo para enrutamiento de paquetes IPv6 con prefijo diferente a 6to4. El *relay* 6to4 es generalmente accesible en la dirección *anycast* 192.88.99.1, no pudiendo el usuario tener control sobre el *relay* elegido para la comunicación

Debido al hecho de que no es posible dirigirse a un *relay* específico, 6to4 presenta problemas potenciales de seguridad. Al desconocerse la localización y el operador del *relay*, la manipulación de los paquetes no puede ser monitorizada. También, se han observado muchas fallas de conectividad en este tipo de esquemas por problemas de enrutamiento a todos los *relays* instalados, quedando abierto el interrogante referido a posibles mejoras.

Teredo es una tecnología, desarrollada por Microsoft, que permite conectividad IPv6 a *hosts* con capacidad IPv6 a través de Internet IPv4, aún cuando se encuentren detrás de un NAT.

6to4 provee un servicio desde el *router* de borde, conectado a Internet, pero no trabaja bien en caso de múltiples NAT entre un sitio y la red Internet IPv4. Teredo resuelve el problema por medio de túneles entre *hosts*. Esto representa un desafío para el propio NAT que generalmente sólo realiza traducciones para tráfico TCP o UDP, no para Protocolo 41. Por este motivo, los paquetes IPv6 se encapsulan en mensajes UDP que, a su vez se encapsulan en IPv4. Así, los mensajes UDP pueden atravesar más de un NAT.

La infraestructura de Teredo consiste de clientes, servidores, *relays* y *relays* específicos de *host*. Un cliente Teredo es un nodo IPv6/IPv4 que soporta una interfaz Teredo a través de la cual se transmiten paquetes hacia otro cliente o nodo Teredo sobre IPv6. Se encuentra conectado a Internet IPv4 detrás de un NAT y se lo configura con una dirección IPv6 con prefijo Teredo 2001:0::/32, con ayuda de un servidor Teredo.

Un servidor Teredo es un nodo IPv6/IPv4 que está conectado a Internet IPv4 que asiste en la configuración de direcciones de los clientes y en facilitar la comunicación inicial entre los clientes. Como no re-envían paquetes pueden ser capaces de soportar muchos clientes. Un *relay* Teredo es un *router* IPv6/IPv4 que puede re-enviar paquetes entre clientes Teredo sobre Internet IPv4 y hosts en redes IPv6. Un *relay* específico de *host* es un nodo IPv6/IPv4 que se conecta tanto a Internet IPv4 como a Internet IPv6. Se puede comunicar con clientes Teredo mediante su acceso a la Internet IPv4, sin necesidad de un *relay*.

La RFC 5389 desarrolla un protocolo que sirve como herramienta para otros cuando se debe atravesar un NAT. Puede usarse para determinar la dirección IP y puerto asignado por el NAT a un dispositivo, chequear la conectividad entre dispositivos finales y mantener las vinculaciones del NAT. Teredo usa un procedimiento parecido, pero no es compatible con todos los dispositivos NAT, razón por la cual no funciona en todos los casos. Por este motivo, Teredo define procedimientos para que un cliente pueda obtener una dirección Teredo y crear y mantener un mapeo en el NAT, para poder iniciar una comunicación con otro cliente Teredo, un *relay* específico de *host* y nodos sólo IPv6. El procedimiento inicial depende de si el cliente Teredo se encuentra detrás de un NAT estático o de uno restringido (dinámico).

6over4 es una técnica de túnel que se basa en el uso de direcciones *multicast* IPv4 y permite a un dispositivo IPv6 aislado, no directamente conectado

a un *router*, trabajar con toda la funcionalidad del protocolo IPv6. 6over4 define además un método para generar una dirección IPv6 local a partir de una dirección IPv4, y un mecanismo para realizar ND sobre IPv4. Los túneles pueden ser sólo entre *hosts* o entre *hosts* y *routers*.

La generación de la dirección de enlace local se determina colocando en los $32\ bits$ más bajos la dirección IPv4. Luego se adiciona un prefijo $fe80::/16$ seguido de una cadena de de "0". Por ejemplo, la dirección IPv4 200.0.182.38 se convierte en un identificador de interfaz $::c800:b626$ y luego en la dirección 6over4 $fe80::c800:b626$.

Los paquetes IPv6 se encapsulan en paquetes IPv4 con dirección *multicast*, es decir que se dirigen a un grupo. Por este motivo, se conoce a este método como Ethernet virtual. Cualquier *router* IPv6 que corra 6over4 y se encuentre en el grupo *multicast* sirve como punto final del túnel, para posterior enrutamiento de los paquetes mediante IPv6.

El descubrimiento de los *routers* IPv6 se realiza mediante ND. Para poder utilizar este protocolo, se deben encapsular los paquetes IPv6 *multicast* en paquetes *multicast* IPv4 con dirección destino $239.192.x.y$, donde x e y son los últimos dos bytes de la dirección *multicast* IPv6. Por ejemplo, un mensaje a todos los routers del enlace local, se dirigiría a la dirección $ff02::2$ y se encapsularía con dirección destino 239.192.0.2.

6over4 no se encuentra implementado en muchos Sistemas Operativos, principalmente porque se apoya en el soporte IPv4 de *multicast*, con sus problemas propios de administración y escalabilidad.

La RFC 5569 describe un mecanismo conocido como **6rd**, basado en 6to4, para permitir a un proveedor desplegar rápidamente servicios IPv6 *unicast* a sitios IPv4 a los que el propio proveedor entrega equipamiento para conexión a Internet, por ejemplo por ADSL o por cable. La sigla que acompaña al número significa justamente despliegue rápido o *rapid deployment*. La solución implica una actualización en los equipos *router* domésticos y la operación de *gateways* entre la infraestructura IPv4 operada por el ISP y la Internet global IPv6, para soporte de encapsulado IPv6 en IPv4.

Mientras que un ISP que desarrolla 6to4 puede garantizar la salida de paquetes IPv6 provenientes de sus clientes, así como la entrega de paquetes desde otros sitios 6to4, no puede garantizar la entrega de paquetes provenientes de sitios IPv6, pues estos precisan llegar a un *relay* 6to4 para que éste los encapsule. El problema es que no hay garantías de la existencia en todas partes de estos *relays*, ni tampoco hay garantías de que estos *relays* puedan enviar paquetes a cualquier punto de la Internet IPv4.

Otro problema es que, si el ISP opera varios *relays* 6to4 y los convierte en alcanzables por la red Internet IPv6 para el prefijo 6to4 $2002::/16$, puede llegar a recibir paquetes destinados a otros ISP, que también estén usando 6to4. Si el ISP no re-envía estos paquetes, esto se traduciría en pérdidas de conectividad. Si, por el contrario, los re-envía, resultaría en una pérdida de eficiencia en el ofrecimiento del servicio para sus propios clientes.

Por eso, 6rd hace algunas modificaciones a 6to4, para que los paquetes que provengan de Internet entren a sus *gateways* 6rd sólo si se destinan a sus

clientes. Para ello reemplaza el prefijo estándar 6to4 por un prefijo IPv6 propio, reemplazando además la dirección *anycast* 6to4 por otra elegida por el propio ISP. Por su parte, un cliente 6rd se configurará de manera automática con un prefijo 6rd asignado por el ISP (con la dirección IPv4 mapeada), la especificación de la longitud de ese prefijo y la dirección del servidor *relay* 6rd.

10.9.3 Traducción

Es un mecanismo utilizado para comunicación entre redes IPv4 e IPV6 a través de un traductor de direcciones. El *host* iniciador de la comunicación debe conocer de antemano la dirección traducida que el del otro extremo posee. Una vez enviado el paquete, el *router* con capacidad de traducción lo pasará al otro formato de direcciones, válido en la red destino. También deberá traducir la dirección fuente a este nuevo formato. Es similar al mecanismo de NAT pero presenta mayor cantidad de desafíos a gran escala, debido a la asimetría entre ambos espacios de direcciones.

La forma de conversión puede ser sin estados o con estados. En el caso sin estados, se supone que cada *host* de la red IPV6 genera su dirección IPv6 a partir de una dirección IPv4, por el agregado de un prefijo pre-establecido, que puede ser el del propio proveedor. En el caso de los *hosts* IPv4, también generan direcciones IPv6 por agregado del mismo tipo de prefijo. La traducción se hace por extracción o agregado del prefijo. Para los hosts IPv6, el conocimiento de las direcciones de los hosts IPv4 mapeadas del otro lado del *router* se puede hacer por requerimiento a un DNS64: un servidor que traduce registros A de hosts IPv4 en registros AAAA para IPv6 siguiendo la regla de mapeo. También, los *hosts* IPv6 registran sus direcciones IPv4 en el servidor, con registros tipo A, que resuelve las preguntas provenientes del lado IPv4. El problema de estos traductores es el consumo de direcciones IPv4.

En el caso de traductores con estado, se mantiene un *pool* de direcciones y las conexiones se resuelven por números de puerto. El problema en este caso es la iniciación de la conexión desde el lado de la red IPv4, cuando el mapeo de la dirección destino IPv6 en IPv4 todavía no ha sido instalado. Para resolver el desafío de comunicar clientes sólo IPv6 a servidores sólo IPv4, la respuesta más aceptada es el traductor detallado en la RFC 6146, conocido como **NAT 64**. Un host sólo IPv6, con un prefijo IPv6 único, se puede conectar a demanda a una red IPv4 a través de este tipo de mecanismo, ya que el dispositivo NAT64 lo mapeará a una dirección IPv4. Dado que la dirección IPv4 ocupa un espacio de $32\ bits$, el ISP precisará un prefijo $/96$ para realizar el mapeo de IPv4. Este prefijo es el prefijo bien conocido $64{:}\ ff9b{::}/96$ que debe ser anunciado sobre la red IPv6, de la misma manera que el prefijo de direcciones del *pool* IPv4 debe anunciarse del lado de la red IPv4.

Un componente adicional, denominado **DNS64**, se puede usar en conjunto con NAT64, para devolver una dirección IPv6 construida a partir de la IPv4, mapeada con el prefijo mencionado previamente. Esta es la respuesta IPv6

esperada por el cliente, que sólo entiende de direcciones IPv6. De este modo, el traductor puede traducir la dirección destino IPv4 del paquete a enviar.

El principal problema de estos traductores es el mantenimiento de la información para cada flujo de datos, por lo que la tabla de búsqueda puede convertirse en el cuello de botella del sistema. Por este motivo, su aplicación masiva es difícil de lograr.

Bibliografía

1. RFC 1546 "Host Anycasting Service", November 1993. http://tools.ietf.org/html/rfc1546
2. RFC 1981 "Path MTU Discovery for IP version 6", August 1996. https://www.ietf.org/rfc/rfc1981.txt
3. RFC 2373 "IP Version 6 Addressing Architecture", July 1998. http://www.ietf.org/rfc/rfc2373.txt
4. RFC 2374 "An IPv6 Aggregatable Global Unicast Address Format ", July 1998. http://tools.ietf.org/html/rfc2374
5. RFC 2402 "IP Authentication Header", November 1998. https://www.ietf.org/rfc/rfc2402.txt
6. RFC 2406 "IP Encapsulating Security Payload (ESP) ", November 1998. http://tools.ietf.org/html/rfc2406
7. RFC 2460 "Internet Protocol, Version 6 (IPv6) Specification", December 1998. http://tools.ietf.org/html/rfc2460
8. RFC 2461 "Neighbor Discovery for IP Version 6 (IPv6) ", December 1998. https://www.ietf.org/rfc/rfc2461.txt
9. RFC 2462 "IPv6 Stateless Address Autoconfiguration", December 1998. http://tools.ietf.org/html/rfc2462
10. RFC 2463 "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", December 1998. https://www.ietf.org/rfc/rfc2463.txt
11. RFC 2474 "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", December 1998. https://www.ietf.org/rfc/rfc2474.txt
12. RFC 2675 "IPv6 Jumbograms", August 1999. http://tools.ietf.org/html/rfc2675
13. RFC 2894 "Router Renumbering for IPv6", August 2000. http://tools.ietf.org/html/rfc2894
14. RFC 3056 "Connection of IPv6 Domains via IPv4 Clouds", February 2001. https://www.ietf.org/rfc/rfc3056.txt
15. RFC 3068 "An Anycast Prefix for 6to4 Relay Routers", June 2001. https://www.ietf.org/rfc/rfc3068.txt
16. RFC 3513 "Internet Protocol Version 6 (IPv6) Addressing Architecture", April 2003. https://www.ietf.org/rfc/rfc3513.txt
17. RFC 3587 "IPv6 Global Unicast Address Format", August 2003. https://tools.ietf.org/rfc/rfc3587.txt

18. RFC 4213 “Basic Transition Mechanisms for IPv6 Hosts and Routers”, October 2005. http://tools.ietf.org/html/rfc4213
19. RFC 5389 “Session Traversal Utilities for NAT (STUN”, October 2008. http://tools.ietf.org/html/rfc5389
20. RFC 5569 “IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)”, January 2010. http://tools.ietf.org/html/rfc5569
21. RFC 6146 “Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers”, April 2011. http://tools.ietf.org/html/rfc6146
22. Kozierok, Charles M., “The TCP/IP Guide”. http://www.tcpipguide.com/free/t_toc.htm
23. Rooney, Tim, “IPv4-to-IPv6 Transition and Co-existence Strategies”, 2011. http://www.ipv6.bt.com/Downloads/bt_wp_IPv6_Transition_Strategies_2011.pdf
24. Koršič, Luka, Straus Istenič, Matjaž, “IPv4/IPv6 Transition Mechanisms”, 2011 https://www.ripe.net/ripe/meetings/regional-meetings/dubrovnik-2011/presentations/IPv6_Transition_Mechanisms_Ripe_07092011_v1.2.pdf
25. RIPE Network Coordinate Center “IPv6 Transition Mechanisms”, Last updated: 08 Apr 2014 http://www.ripe.net/lir-services/training/e-learning/ipv6/transition-mechanisms
26. H3C Technology Introduction “Tunneling Introduction”, H3C Technologies Co., Limited Copyright 2003-2014. http://www.h3c.com/portal/Products___Solutions/Technology/IPv4___IPv6_Services/Technology_Introduction/200702/201180_57_0.htm

Problemas

1. Realice un cuadro comparativo entre las cabeceras IPv4 e IPv6, destacando los campos semejantes, los que han sido modificados y los que han sido quitados del viejo protocolo. Destaque los motivos en cada caso.

2. Utilice las reglas de compresión para re-escribir la siguiente dirección IPv6 $2001{:}\,0db8{:}\,0000{:}\,0000{:}\,b450{:}\,0000{:}\,0000{:}\,00b4$.

3. Explique las mejoras que incorpora IPv6 en cuanto al proceso de configuración.

4. ¿Qué funcionalidad tienen las cabeceras de extensión IPv6?

5. ¿Cómo se organiza el nuevo espacio de direcciones IPv6? ¿Qué consideraron sus diseñadores al elegir esta estructura?

6. ¿Qué es el protocolo ND y para qué se utiliza?

7. Realice un cuadro comparativo entre los diferentes mecanismos de migración presentados. Considere aspectos relacionados con sus ventajas, desventajas y escenarios posibles.

CAPÍTULO XI

Protocolo TCP – Aspectos Generales

Se ha abordado en capítulos previos la funcionalidad necesaria para que la interconexión de redes sea posible, en particular el protocolo IP en sus dos versiones en uso hoy en día. Esta funcionalidad no alcanza para muchas aplicaciones, principalmente debido a que IP es un protocolo no confiable y no orientado a la conexión, cuya filosofía de entrega de datos es la del ¨mejor esfuerzo¨. Esta filosofía no parece apropiada en el caso de aplicaciones que precisan una entrega confiable, sin pérdidas y ordenada de sus mensajes. Otras aplicaciones no funcionan bien sin algún tipo de prevención frente a las situaciones de congestión en la red o de diferencia de capacidades de procesamiento entre transmisor y receptor.

Existen dos protocolos en la Arquitectura TCP/IP capaces de proveer cierta distancia entre el nivel de red y el de las aplicaciones: el Protocolo de Control de la Transmisión (TCP, Transmission Control Protocol) y el Protocolo de Datagrama de Usuario (UDP, User Datagram Protocol). Ambos protocolos funcionan en el mismo nivel, equivalente a la capa de transporte en el modelo OSI, y poseen un esquema de direcciones común, aunque ofrecen diferentes servicios. También comparten ciertos aspectos de comunicación con el Sistema Operativo en cuanto a la transferencia de datos.

UDP es un protocolo más sencillo, parecido a IP en el sentido de no ofrecer confiabilidad, ya que su filosofía es no orientada a la conexión. Existen muchas aplicaciones tolerantes a las pérdidas que pueden transportar sus mensajes en protocolos de estas características. Aplicaciones relacionadas con el streaming de video o las del tipo multimedia pueden funcionar bien en estas condiciones. Otras aplicaciones, del tipo transaccional, asumen que el modelo requerimiento/respuesta es apropiado en sí mismo para transportar pequeños mensajes de datos. Para este tipo aplicaciones, la ausencia de respuesta es un alerta suficiente para disparar una retransmisión y se programan internamente con mecanismos propios para solucionar estas circunstancias.

Por el contrario, TCP es un protocolo más complejo, con capacidad de brindar confiabilidad a las aplicaciones cuyos mensajes encapsula, ya que su filosofía de funcionamiento es orientada a la conexión, tiene capacidad de comunicación full dúplex y soporta algoritmos de control de congestión y de

control de flujo. Muchas aplicaciones que transportan mensajes de gran volumen de datos precisan de estas características. Como ejemplo se puede mencionar el protocolo de comunicación usado en la web, HTTP.

11.1 Establecimiento de una Comunicación - Sockets

En capítulos previos se han presentado diferentes esquemas de direccionamiento en diferentes niveles de la Arquitectura TCP/IP. Por ejemplo, a nivel MAC existen direcciones de 48 $bits$ que tienen significado únicamente local, dentro de una red LAN. El protocolo IP también presenta su propio esquema de direccionamiento, pero su significado es global, para hacer posible el enrutamiento de mensajes entre redes. La cabecera IP posee además el campo Protocolo, que permite el de-multiplexado en recepción entre aquellos protocolos que se encapsulan en IP, por ejemplo TCP o UDP.

De la misma manera, existe un esquema de direccionamiento a nivel de transporte, ya que debe ser posible diferenciar entre aplicaciones corriendo al mismo tiempo dentro de un mismo dispositivo. Tanto para TCP como para UDP, estas direcciones se denominan puertos o *ports*. Para identificar el proceso que origina una transmisión, ambos protocolos incluyen un **Puerto Fuente**. Con la misma concepción, un **Puerto Destino** identifica el proceso en destino con el que se establece la comunicación.

Según la RFC 6335, se trata de números de 16 $bits$, en el rango entre 0 y 65535, separados en tres grandes grupos:

- **Puertos bien conocidos** o ***well-known***: identifican a los servidores estándar, reservando un grupo de números para dichas aplicaciones que se establece en el rango de 1 a 1023. Cada servidor tiene un número específico de puerto asignado para que el proceso servidor escuche los requerimientos que le llegan desde la red, destinados a ese número de puerto. El *software* que implementa la aplicación, por convención, usa el mismo puerto reservado en todos los dispositivos donde corre como proceso. La asignación fija permite a los clientes encontrar a los servidores sin necesidad de información de configuración. Durante casi 20 años, el IANA periódicamente publicaba tablas de estas asignaciones numéricas, en formato RFC. La última tuvo carácter de estándar, conocido con el nombre STD 2 (RFC 1700). Actualmente, existe una base de datos en línea, accesible a través de la página web www.iana.org. Algunos ejemplos de puertos bien conocidos son: el servicio de correo que usa el puerto 25, el servicio web que escucha en el puerto 80 y el servicio de nombres en el puerto 53. La mayoría utiliza sólo un puerto bien conocido del lado servidor, aunque una excepción es el caso de BOOTP ya visto, que usa dos puertos: el puerto 68 para el servidor y el puerto 67 para el cliente.

- **Puertos efímeros*:*** los clientes no necesitan utilizar puertos bien conocidos porque lo que interesa es que inicien la comunicación con los servidores conociendo su número de puerto de antemano. Este número, como el propio del cliente, se transporta en los campos puerto destino y puerto fuente de los datagramas UDP o de los segmentos TCP. El dispositivo donde se aloja el cliente proporciona un puerto a cada proceso cliente, por el tiempo que éste lo necesite, de ahí su denominación de efímero. Los números de puertos efímeros tienen valores mayores de 1023, en el rango de 1024 a 49151, y se escriben en el campo puerto fuente de datagramas UDP o segmentos TCP provenientes de un cliente. Se debe asignar un número diferente para cada proceso cliente que se levanta en el dispositivo y este número no puede volver a usarse inmediatamente una vez finalizada la comunicación.

- **Puertos dinámicos**: se trata de los puertos en el rango $49152 - 65535$, también conocidos como puertos privados. No se encuentran asignados.

Algunas aplicaciones del tipo servidor pueden funcionar sobre UDP o TCP utilizando el mismo número de puerto. Esto es posible debido a que existe de-multiplexado en la entrega, a nivel IP.

Por otra parte, a lo largo de los años, con la aparición de nuevas aplicaciones, se han permitido puertos bien conocidos, designados por el IANA, por encima de 1023, en el rango de los puertos efímeros. Por ejemplo, el puerto 8080, HTTP alternativo, queda reservado como otro posible puerto para un servidor web.

Vale la pena aclarar que, muchas veces, un ataque a la seguridad de un sistema comienza por el acopio de información referida al estado de los puertos de diferentes dispositivos. Para lograr este objetivo, existen programas que permiten determinar cuáles puertos en una determinada máquina se encuentran abiertos o en estado LISTEN, o sea preparados para recibir requerimientos de clientes. El sentido de obtener un listado de puertos en el estado referido es buscar vulnerabilidades, para atacar la máquina objetivo. Estas pruebas se pueden realizar con dispositivos desde redes externas a la que está siendo probada.

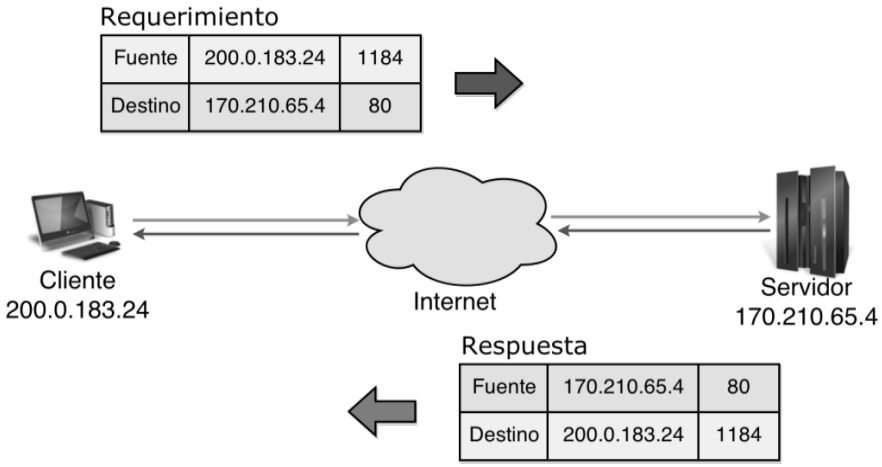


Figura 11.1 – Comunicación Cliente- Servidor.

En una fase de funcionamiento normal entre cliente y servidor se intercambian mensajes, encapsulados en TCP/UDP y también en IP, correspondiéndose a cada nivel un esquema de direccionamiento. Por ejemplo, un requerimiento típico de una de una página web desde un dispositivo que cuenta con un navegador, implicará el envió de un mensaje con dirección IP destino la correspondiente al servidor y puerto destino bien conocido 80 u 8080. Del lado cliente, el navegador corre como aplicación cliente en el dispositivo con dirección IP cargada en la dirección fuente del mensaje y número de puerto obtenido dentro del rango de los efímeros, por ejemplo 1184 tal como se aprecia en la Fig. 11.1. Se observa allí también que en el requerimiento y la respuesta se intercambian tanto la dirección IP como los puertos origen y destino. La combinación $(Dirección\ IP, Número\ de\ Puerto)$ identifica unívocamente un extremo de la comunicación: un dispositivo en la red y la aplicación corriendo en el mismo. Esta combinación se conoce con el nombre de *socket*. Se trata de una definición muy importante, pues conociendo el par de *sockets*, se puede identificar unívocamente cualquier comunicación en la red:

$$(Dirección\ IP\ Fuente, Número\ de\ Puerto\ Fuente)$$
$$(Dirección\ IP\ Destino, Puerto\ Destino) \tag{11.1}$$

El concepto de *socket* es la base para la programación de la Interfaz de Programas de Aplicación (API, Application Program Interface) que lleva su mismo nombre. Existen varios tipos de API, todos intentan ofrecer a las aplicaciones una forma sencilla de usar la Arquitectura TCP/IP para comunicarse.

En entornos de aplicación se crean *sockets* para permitir la comunicación entre dos procesos diferentes en la misma o en diferentes máquinas. Se trata de una forma de comunicación entre computadoras que utiliza el concepto de descriptores de archivos (fd, file descriptors) derivado de sistemas operativos tipo

Unix. En Unix, toda acción del tipo I/O se realiza leyendo o escribiendo un descriptor de archivos. El propio descriptor es, en realidad, un número entero asociado con un archivo abierto, que puede ser una conexión de red, un archivo de texto o un terminal. Para un programador se trata de un concepto de bajo nivel ya que los comandos del tipo *read()* y *write()* funcionan con *sockets* de la misma manera que lo hacen con archivos. La diferencia entre *sockets* y *fd* se establece en la creación del *socket* y en la incorporación de operaciones especiales para control del *socket*.

Un *socket* de Unix se usa en aplicaciones programadas bajo el paradigma cliente/servidor para establecer conexión entre ambos. Existen cuatro tipos de *sockets*, aunque sólo dos son los más usados, asumiéndose que los procesos se comunican sólo entre *sockets* del mismo tipo:

- **Stream Sockets*:*** este tipo de comunicación entre procesos se traduce en una entrega en la red garantizada y en orden. Se usa TCP para la transmisión de datos. Si la entrega no es posible, el transmisor recibe una indicación de error.

- **Datagram Sockets*:*** en este caso, la entrega en la red no se encuentra garantizada. Se trata de comunicaciones del tipo sin conexión, en las que simplemente se construye un paquete y se lo transmite. Se usa UDP para la transmisión de datos.

Por su parte, en lo que respecta al tipo de servicio ofrecido, es importante distinguir entre dos clases de servidores:

- **Servidores Iterativos:** se trata de los servidores más sencillos, cuya capacidad se limita a la atención de un cliente por vez. Es decir que, recién después de completar un requerimiento, atienden otra solicitud cliente. El cliente cuya solicitud sigue a continuación, debe esperar que el servidor se desocupe para poder ser atendido.

- **Servidores Concurrentes:** estos servidores permiten correr múltiples procesos y servir varios requerimientos a la vez. La forma más sencilla de hacerlo, en *Unix*, es por medio de una llamada al sistema, conocida como *fork*, que genera un proceso copia de sí mismo, denominado *hijo* para poder manejar cada cliente por separado.

Por otro lado, las llamadas al sistema para establecer una conexión son diferentes para cliente y servidor, aunque ambas incluyen la construcción básica de un *socket*, ya que ambos procesos establecen sus propios *sockets*.

Del lado cliente, cuando se establece una comunicación por red, se crea un *socket* con la llamada al sistema *socket()*. Se conecta el *socket* a la dirección del servidor con la llamada al sistema *connect()*. Se transmiten y reciben datos con las llamadas al sistema *read()* y *write()*.

Del lado servidor, se crea un *socket* con la llamada al sistema *socket()*. Se asocia el *socket* a una dirección con la llamada al sistema *bind()*. Se escuchan

conexiones con la llamada al sistema *listen()*. Se aceptan conexiones con *accept()*. Esta llamada típicamente se bloquea hasta que un cliente se conecta con el servidor. Se transmiten y reciben datos con las llamadas al sistema *read()* y *write()*.

En la Fig. 11.2 se puede observar la sucesión de llamadas mencionadas, tanto del lado de un cliente TCP, como del lado del servidor concurrente. Cuando se aborde el estudio del protocolo TCP, se comprenderá mejor la relación existente entre los parámetros de algunas llamadas al sistema y el propio protocolo.

La primitiva *socket()* crea el socket, llevando asociados parámetros para determinar el tipo de servicio, el protocolo y el formato de direcciones. Una combinación de parámetros para un servidor del tipo concurrente podría ser la especificación de una familia de protocolos (TCP/IP), la forma de transmitir los datos (flujo de bytes/datagramas) y el tipo de protocolo utilizado (TCP/UDP).

Una vez creado el *socket* y asignados *buffers* de memoria para transmisión y recepción, la primitiva devuelve a la aplicación un descriptor de socket (sd, socket descriptor) para que se utilice luego en las siguientes llamadas a primitivas. A su vez, ese descriptor identifica un registro con una tabla que describe el *socket*.

A continuación, la aplicación servidora debe ligar el *socket* a una dirección. Para lograrlo, se llama a la primitiva *bind()* que lleva como parámetros el *sd* y la dirección del *socket*, es decir la combinación $(dirección\ IP\ , Nº\ de\ puerto\ bien\ conocido)$. La llamada *bind()* devuelve un código, indicando error o éxito en la ejecución.

Para instruir al *socket* del lado servidor que atienda las llamadas entrantes se utiliza la llamada al sistema *listen()*, cuyos parámetros son el descriptor del socket y la longitud máxima de la cola de atención de clientes. Esta llamada también puede devolver un código de éxito o de error.

Para aceptar llamadas entrantes, el servidor utiliza la primitiva *accept()*, colocando a la aplicación servidora en modo bloqueado en espera de solicitudes provenientes de clientes TCP. La llamada tiene como parámetros el descriptor de sockets y la dirección del *socket*, devolviéndose un código de error o éxito luego de su ejecución.

La secuencia de llamada *socket()*, *bind()*, *listen()* y *accept()* del lado servidor, se conoce con el nombre de **apertura pasiva**.

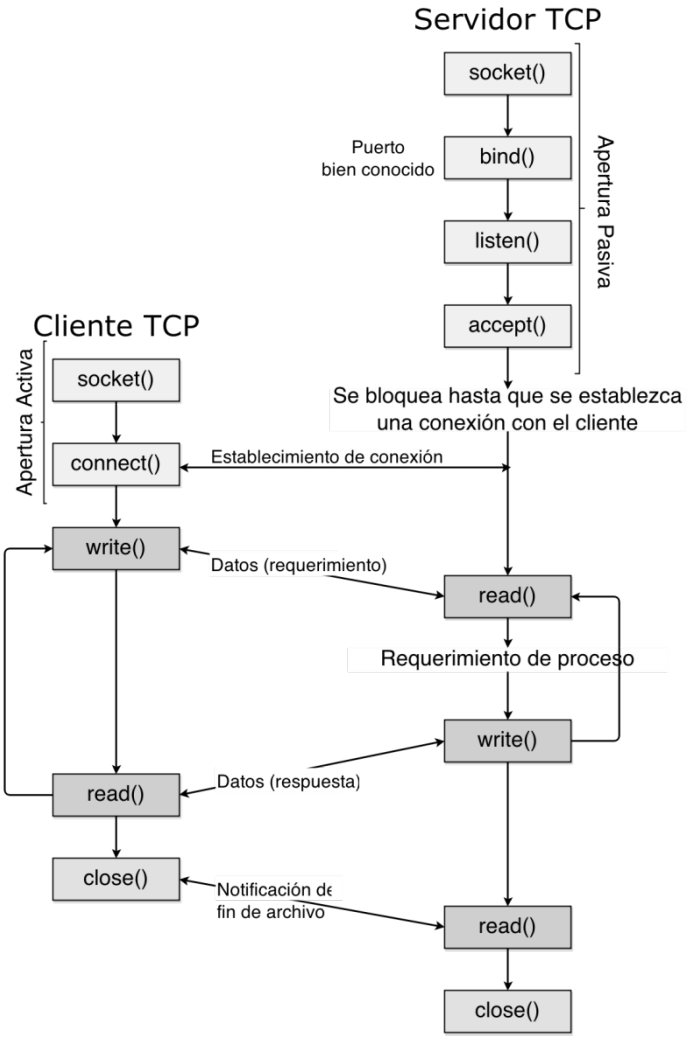


Fig. 11.2 – Llamadas al sistema para la comunicación en red.

Por su parte, la aplicación cliente también edita una primitiva *socket()* para crear un nuevo *socket* para establecer una conexión. La creación del socket le devuelve un descriptor, para referenciarlo en llamadas posteriores, y crea una tabla en memoria con los datos asociados, que incluyen el par de sockets $(Dirección\ IP, Número\ de\ Puerto)$ que identifica unívocamente la conexión.

A continuación, el cliente editará un *connect()*, cuyos parámetros incluyen el descriptor de *socket*, el número de puerto local, el número de puerto destino, la dirección IP destino, la precedencia de la comunicación y datos opcionales. La precedencia se refiere a la forma en que se debe completar el campo TOS del encabezado IP. Los datos opcionales pueden referirse, por ejemplo, a nombres de usuario y contraseñas.

Luego de esta llamada la aplicación cliente pasa a estado bloqueado mientras que el protocolo TCP local inicia el establecimiento de una conexión con el servidor. Las primitivas *socket()* y *connect()* constituyen una **apertura activa**, ya que es el cliente el que inicia la conexión enviando segmentos TCP sobre la red.

Cuando la aplicación servidora recibe, a través de la red, la solicitud de una nueva conexión, se desbloquea y crea una nueva instancia de sí misma, por si tuviera que atender otras conexiones. Esto se realiza con la primitiva *fork()*, que crea un nuevo *socket* entre la nueva instancia de aplicación y el protocolo TCP local. El proceso padre regresa al estado bloqueado a la espera de nuevas solicitudes.

Recién en este punto las aplicaciones cliente y servidor pueden comenzar a intercambiar datos en ambos sentidos con las primitivas *send()* y *receive()*.

Tanto el protocolo TCP del lado cliente, como el del servidor, pueden soportar múltiples conexiones al mismo tiempo para distintos clientes. Existe un registro de conexión que permite distinguir entre las conexiones simultáneas. El registro de conexión incluye un identificador, que no es otra cosa que el par de *sockets*, un tamaño de segmento máximo (MSS, Maximum Segment Size), un número de secuencia inicial (ISN, Initial Sequence Number) que es diferente para cada extremo de la conexión, un valor de precedencia, el tamaño de la ventana para control de flujo y variables de estado asociadas al funcionamiento del propio protocolo.

Por otra parte, cada *socket* se haya asociado a un buffer de transmisión y otro de recepción. Generalmente el buffer de recepción se utiliza para reordenar los datos antes de entregarlos a la aplicación correspondiente. La llamada *send()* coloca datos en el buffer de transmisión del *socket*, para que la entidad local TCP los lea. Los parámetros de esta llamada incluyen el descriptor de *socket*, un puntero al buffer que contiene el bloque de datos y su longitud en *bytes*.

Existe un parámetro asociado a la primitiva *send()*, denominado bit de empuje o bit de *push*, mediante el cual la aplicación local puede solicitar al otro extremo la entrega inmediata del bloque de datos. Existe también otra bandera, denominada bit de urgente, que se utiliza en aplicaciones interactivas, cuando se envía un comando urgente. Entonces se activa el bit correspondiente del encabezado TCP, como una manera de solicitar una respuesta rápida al envío de estos datos al otro extremo de la conexión.

Las primitivas *close()* o *shutdown()* se utilizan para liberar cada extremo de la conexión. La comunicación finaliza borrando los registros de conexión y la instancia creada por la llamada al sistema *fork()*.

De lo desarrollado hasta aquí, se puede deducir que existe un tipo de *socket*, que se podría identificar como *socket* de red, que se utiliza para identificar el extremo final de una comunicación entre procesos a través de una red de datos. Este tipo de *sockets* también se suelen denominar ***sockets de Internet***, dado que la comunicación en red se apoya en el protocolo IP.

Por otra parte, un ***socket API*** es una interfaz provista por el sistema operativo para que las aplicaciones utilicen una red de datos a través de los *socket*s de Internet.

11.2 Protocolo UDP

El protocolo UDP es utilizado por aquellas aplicaciones que precisan un transporte rápido sobre la arquitectura TCP/IP, ya que la propia programación de los mismos incluye, de ser necesario, la forma de tratar problemas relacionados con la retransmisión de mensajes o confiabilidad de entrega.

Tanto un cliente como un servidor comunicándose a través de una red con este protocolo, utilizarán las llamadas al sistema *socket()* y *bind()*, con los parámetros apropiados, utilizando luego las llamadas *send()* y *receive()* para el intercambio de datos.

El protocolo UDP se describe en la RFC 768, donde se presentan las funcionalidades que debería permitir la interfaz de usuario: la creación de puertos nuevos de recepción, las operaciones sobre dichos puertos que permitan obtener los bytes de datos con una indicación de los números de puerto fuente y destino y las operaciones necesarias para transmitir datagramas con especificación de datos, puertos y direcciones de ambos extremos. La forma en que se implementa queda a elección de cada distribuidor.

Al igual que IP, UDP no proporcionan una entrega garantizada ni control de flujo ni recuperación de errores. De ser necesario brindar un mínimo de confiabilidad a la comunicación, la funcionalidad específica deberá ser implementada por el software de la propia aplicación. Por este motivo, el encabezado UDP es muy sencillo, de apenas 8 $bytes$, como se puede apreciar en la Fig. 11.3. Se suele denominar datagrama UDP al conjunto de encabezado y datos encapsulados.

Los primeros dos campos del encabezado se refieren a los puertos de origen y destino, cada uno de 16 $bits$. El puerto de origen es el número asignado al proceso que origina el datagrama. Generalmente lleva un puerto efímero cuando se trata de un requerimiento de cliente, y un puerto bien conocido en las respuestas del servidor. El puerto destino es el del proceso en el extremo receptor.

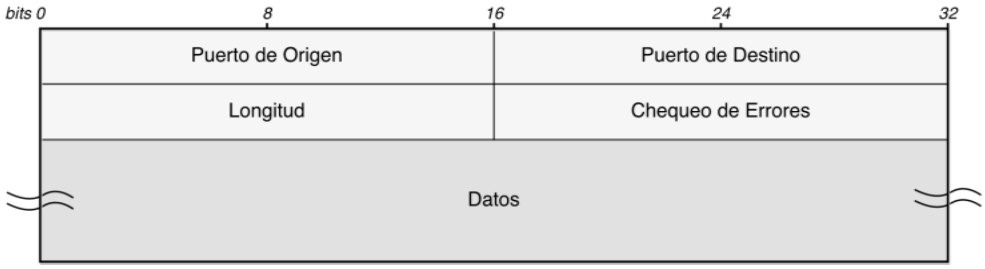


Figura 11.3 – Encabezado UDP

El campo Longitud, también de 16 $bits$, se refiere a la longitud total del datagrama UDP, encabezado más datos, medida en $bytes$.

El campo de Chequeo de Errores es opcional. Se trata de un control de errores que se calcula de una manera muy particular, para cubrir el datagrama completo e información cargada en un encabezado falso o *pseudoheader*. El *pseudoheader* tiene una longitud de $12\ bytes$ e incluye los siguientes campos: Dirección IP fuente ($4\ bytes$), Dirección IP destino ($4\ bytes$), campo Protocolo del encabezado IP ($1\ byte$), campo Longitud del encabezado UDP ($2\ bytes$) y un campo de Relleno en "0" ($1\ byte$). El chequeo de errores se realiza teniendo en cuenta esta cabecera falsa, la propia cabecera UDP y los datos cargados por el datagrama, que deben rellenarse hasta completar una cantidad de bits múltiplo de 16.

En el campo de Chequeo de Errores se coloca el resultado de la operación complemento a uno de la suma complemento a uno de los elementos mencionados, operándose sobre palabras de $16\ bits$. Si la suma de dos palabras genera un bit de acarreo en el bit más significativo, se lo debe sumar al menos significativo. Una vez que se sumaron todas las palabras, se debe invertir el resultado binario. Si se incluye este cálculo, el campo se cargará con un número distinto de cero. Una vez que el datagrama llegue a destino, se re-calculará y, si se detectan errores, se descartará el datagrama de manera silenciosa.

La sencillez de este protocolo lo hace apropiado para aquellas aplicaciones que no precisan de toda la confiabilidad ofrecida por TCP. Se trata de aplicaciones en las que se valora la rapidez en la entrega, debido a que una fase de establecimiento de la conexión, como la que ofrece TCP, significaría una pérdida de tiempo no tolerable. También podría tratarse de aplicaciones que no toleran el retardo generado por re-transmisiones, otra funcionalidad que TCP desarrolla.

Un ejemplo clásico lo constituye aplicaciones multimedia: al transmitir un *streaming* de video, la rapidez y continuidad del flujo de datos importa más que la pérdida de algunos $bytes$, pues el usuario no apreciaría sino disrupciones significativas del servicio y las retransmisiones se realizarían sobre partes pasadas del propio flujo, retrasando la presentación apropiada en tiempo real. Otro ejemplo se encuentra en las aplicaciones transaccionales. Este tipo de aplicaciones funcionan bajo el modo requerimiento/respuesta, siendo apropiadas para la funcionalidad ofrecida por UDP, debido a que la propia respuesta es un indicador de la llegada del requerimiento. De no producirse esta situación, las aplicaciones pueden ser programadas con funcionalidades adaptadas para la retransmisión.

11.3 Aspectos más significativos del protocolo TCP

En distintos tipos de redes existen diferentes modos de funcionamiento. Hemos presentado las redes de conmutación de circuitos y las de conmutación de paquetes. Dentro de estas últimas hemos distinguido entre servicio orientado a la conexión y servicio sin conexión. Cada tipo de red lleva asociado una serie de protocolos, pero por encima de la capa de red existen los llamados protocolos de

aplicación y otros, del nivel de transporte, que también ofrecen soporte a las aplicaciones.

Los protocolos de capa de transporte ocultan a las aplicaciones los servicios proporcionados por las capas de red, proporcionando servicios propios, independientes de los servicios de red. Se trata de protocolos extremo a extremo (e*nd-to-end*), denominados así puesto que permiten la comunicación, en su propio nivel, entre los extremos finales transmisor y receptor. Se ha mencionado que UDP proporciona un servicio sin conexión, del tipo datagramas, en tanto que TCP ofrece un servicio orientado a la conexión a nivel de la capa de transporte. Utilizar uno u otro servicio depende de los requisitos de la aplicación en cuestión.

El protocolo TCP fue definido en la RFC 793, con la finalidad de ofrecer toda la funcionalidad de **transporte confiable de datos** que precisan muchas aplicaciones de Internet. Tiene una característica común con UDP, en el sentido de que ambos comparten el mismo esquema de direccionamiento, aunque TCP utiliza estos números de puertos para establecer conexiones seguras entre origen y destino de la comunicación.

Al tratarse de un **protocolo orientado a la conexión** a nivel de transporte, los dispositivos extremos deben establecer una conexión antes de comenzar a intercambiar datos. Esto significa que, al principio de la comunicación, existe una fase especial de establecimiento de la conexión, en la que se pueden negocian parámetros para acordar la transferencia de datos confiable. Luego de cumplirse de manera exitosa la fase de establecimiento, se permite comenzar a intercambiar datos. El servicio de transferencia de información es bidireccional y confiable. Cuando finaliza la comunicación, se debe realizar otro procedimiento especial para finalizarla.

Cada conexión establecida por TCP se identifica por un **par de *socktes***. El par de valores $(Dirección\ IP, Número\ de\ puerto)$ es el *socket* que identifica un extremo de la conexión. El par de *sockets*, correspondiente a ambos extremos, puede identificar unívocamente la conexión. Esta forma de identificación es una de las razones por las que se permite la simultaneidad de conexiones sin conflictos, ya sea desde un mismo dispositivo IP, a través de diferentes números de puerto, o con distintos dispositivos IP, sobre el mismo número de puerto. El primer caso se puede asociar a una perspectiva del lado cliente, en tanto que el segundo se puede pensar como la función realizada por un servidor.

En TCP, los datos se transmiten en unidades denominadas **segmentos**, cuyo tamaño máximo depende de la Unidad de Transmisión Máxima (MTU, Maximum Transmission Unit) de la red subyacente. Esta limitación permite definir un parámetro conocido como **Tamaño Máximo de Segmento** (MSS, Maximum Segment Size), que resulta de restar al valor de la MTU la longitud de la cabecera IP y la propia longitud de la cabecera TCP.

Una vez establecida una conexión TCP, el **vínculo es bidireccional**, ya que ambos extremos pueden enviar y recibir datos, sin importar cuál de ellos haya iniciado la conexión. En cualquier caso, cada vez que TCP transmite un segmento, inicia una cuenta de tiempo, mediante un reloj apropiado, a la espera de la recepción de un segmento especial de reconocimiento ACK, proveniente del otro extremo de la comunicación. Si no se recibe un reconocimiento sobre los datos transmitidos durante el tiempo precisado, se inicia una retransmisión. Esta

estrategia, agregada al protocolo para ofrecer confiablidad, se conoce con el nombre de **estrategia de expiración (*timeout*) y retransmisión**.

Las aplicaciones bajan sus mensajes para que el protocolo los encapsule en segmentos. En realidad, TCP trata la transferencia de datos como un **flujo de bytes**. Este concepto significa que TCP es capaz de distinguir los datos transportados con esta granularidad. Así, el contenido de cada segmento deberá ser reconocido mediante un segmento ACK, pero se ofrece la ventaja de poder especificar el reconocimiento de manera acumulativa sobre varios segmentos transmitidos y recibidos correctamente, apuntando a un número de byte particular. Esto es posible debido a la adopción de un mecanismo de control de errores que trabaja en conjunto con un mecanismo de control de flujo del tipo **ventana deslizante**. Además, el mecanismo de control de flujo permite acomodar diferencias entre tamaños de buffers y velocidades entre los extremos en comunicación.

Una de las consecuencias de la transmisión orientada al flujo de bytes es que el protocolo no puede distinguir entre mensajes enviados por la aplicación, quedando en manos de la programación de las aplicaciones la distinción entre diferentes mensajes.

Debido a que el mecanismo de entrega subyacente del tipo *best effort* de IP puede provocar la recepción de segmentos fuera de orden, TCP debe contar con alguna información extra para un reordenamiento, previo a la entrega de datos a la aplicación. La decisión de diseño de TCP de tratar los mensajes de las aplicaciones como un flujo de bytes, condujo a la definición de un campo de **número de secuencia** en el encabezado del protocolo. En este sentido, cada byte de datos tiene asignado un número de secuencia para poder registrar el estado de la transmisión, aunque la misma se realiza por bloques cuyo tamaño máximo depende del valor del MSS resultante para la red subyacente.

También, la pérdida de un segmento ACK que reconozca la llegada de cierta cantidad de datos, puede generar retransmisiones innecesarias, por lo cual el protocolo debe contar con la posibilidad de manejar la recepción de **segmentos duplicados**.

Toda la confiabilidad ofrecida por TCP puede no ser suficiente cuando la situación del conjunto de redes subyacente, entendido como el conjunto de redes y *routers* que deben atravesar los segmentos, no es la óptima. En ciertos casos, cuando la red se encuentra sobrecargada, pueden suceder situaciones de demoras en el manejo de los paquetes en las colas de los *routers*, o aún peor, el descarte de los mismos. Es lo que se define como situación de congestión de una red. A nivel de TCP se percibirá la situación por la ausencia de llegada de segmentos ACK para los datos transmitidos. Esta situación disparará retransmisiones que pueden agravar aún más la condición de congestión. A pesar de tratarse de un problema primordialmente de responsabilidad del nivel de red, miles de conexiones establecidas al mismo tiempo podrían generar una situación de colapso. Por este motivo, el protocolo TCP incluye algoritmos para implementar un **mecanismo de control de congestión**.

11.4 Mecanismo de Control de Flujo por Ventana Deslizante

Se ha advertido que TCP brinda un servicio orientado al flujo de bytes, ya que puede aceptar datos de cualquier tamaño o estructura debido a que trata estos datos, provenientes de las aplicaciones, como un flujo de bytes. Serán las aplicaciones las encargadas de imponer reglas de entendimiento para poder separar los mensajes de dicho flujo de manera apropiada.

Los bytes de datos que bajan las aplicaciones son acumulados por TCP, que los empaqueta regularmente en segmentos, para luego ser encapsulados por el protocolo IP. La única limitación, como se ha expresado, es la longitud, condicionada al valor del MSS. Este parámetro es intercambiado entre ambos extremos de la comunicación, durante el establecimiento de la conexión a nivel TCP.

Para poder ofrecer un servicio confiable, TCP debe llevar un registro de los datos bajados por las aplicaciones, ya transmitidos, para saber si fueron recibidos correctamente. En este sentido, es necesario contar alguna forma de identificación, que además sirva para subirlos en el orden correcto en el extremo receptor.

Al ser orientado al flujo de bytes, la forma de identificación más apropiada no podría ser la de asociar un número diferente por cada mensaje, sino la de llevar una cuenta de cada byte. Para ello, se anexa el campo **Número de Secuencia** en el encabezado TCP. Al asignar un número a cada byte, el Número de Secuencia del encabezado se llenará con el número correspondiente al primer byte de cada segmento.

El seguimiento de un flujo de bytes confiable, exige una numeración similar con respecto a la recepción. Por este motivo, la devolución de un segmento ACK también se acompaña con un **Número de ACK**. Se trata de otro campo del encabezado, donde el receptor anuncia al transmisor el próximo byte que espera recibir de manera correcta y en orden.

Ambos números se utilizan en la estrategia de control de flujo de ventana deslizante. Con este tipo de control de flujo se permite la transmisión de varios segmentos a la vez, cuyo volumen queda sujeto a un aviso del tamaño del buffer de recepción, proveniente del otro extremo de la comunicación. Este dato da lugar a otro campo en el encabezado TCP: **el tamaño de la ventana de recepción**, interpretado como el espacio libre que queda en el buffer de recepción.

De esta manera, basado en el Número de Secuencia, en el Número de ACK y en el tamaño de la ventana, cada extremo puede mantener el estado de la comunicación mediante el seguimiento del estado de dos buffers: el de los segmentos transmitidos y el de los segmentos recibidos. El aviso del tamaño del buffer de recepción limita la cantidad de bytes que un extremo le puede enviar al otro. Como en este buffer se almacenarán los segmentos recibidos hasta el momento que puedan ser entregados a las aplicaciones, el anuncio de su tamaño es un número variable, pudiendo llegar a anunciarse un valor nulo, como clara indicación de detener la transmisión de datos por falta de espacio. Por su parte, el transmisor lleva un registro de los bytes enviados, distinguiendo entre los que ya han sido reconocidos y los que aún no han recibido la confirmación de recepción mediante segmentos ACK.

Tal como se presenta en la Fig. 11.4, se puede asociar el registro de bytes del buffer de transmisión a cuatro categorías, según su estado:

- **Bytes ya enviados y que han recibido ACK**: no es necesario mantener estos datos en el buffer de transmisión puesto que ya se han sido enviados y recibidos correctamente.

- **Bytes ya enviados y que no han recibido ACK**: estos datos deben permanecer en el buffer de transmisión, dispuestos para posibles retransmisiones.

- **Bytes todavía no enviados pero que el receptor estaría dispuesto a recibir**: se refiere a aquellos datos que podrían ser enviados, puesto que el receptor tiene lugar en el buffer de recepción para recibirlos. El motivo por el cual no han sido enviados y, por lo tanto no se encuentran almacenados en el buffer de transmisión, es muy variable y está ligado al devenir de la comunicación. Por ejemplo, podría suceder que no haya datos para transmitir.

- **Bytes todavía no enviados y que el receptor no se encuentra dispuesto a recibir**: el anuncio del tamaño del buffer de recepción prohíbe al transmisor avanzar más allá de un límite en el flujo de bytes transmitidos.

La Fig. 11.4 presenta las cuatro categorías mencionadas presentadas en un momento particular de una comunicación. En este caso, se ha recibido ACK para todos los bytes hasta el número 31, por lo tanto podrían desecharse. Los bytes 32 a 45 ya se han enviado, pero se encuentran en espera de la recepción de un segmento ACK que los reconozca. Los bytes 46 a 51 se podrían enviar, puesto que hay lugar en el buffer de recepción en el otro extremo, según lo anunciado en el tamaño de la ventana. Más allá del byte 52 no es posible enviar datos.

Conceptualmente, en el caso de la Fig. 11.4, si llegara un ACK que incluyera reconocimiento para el byte 32, este pasaría a formar parte de la categoría 1. Adicionalmente, si el receptor no variara su anuncio de tamaño de buffer de recepción, en ese momento se podrían llegar a enviar los bytes 46 a 52, avanzando la categoría 3 un lugar hacia la derecha. Efectivamente, los bordes de la categoría 1 y la categoría 4 se moverían sincronizadamente hacia la derecha, deslizándose un lugar. De ahí el nombre de ventana deslizante para este mecanismo de control de flujo.

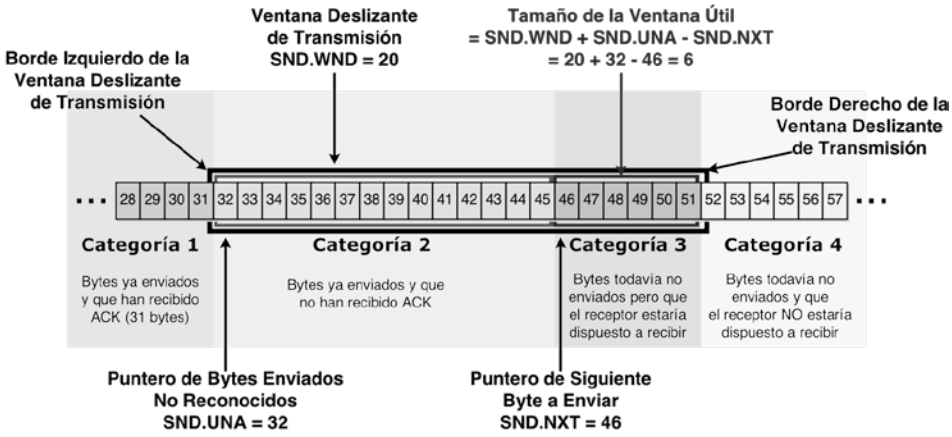


Figura 11.4 – Ventana conceptual de transmisión

El transmisor lleva el registro de los datos del flujo de bytes transmitido a través de una serie de variables internas, conocidas como punteros de la ventana de transmisión. Tal como se aprecia en la Fig. 11.4, tres punteros dividen las cuatro categorías de bytes transmitidos:

- **SND.UNA**: es el puntero que almacena el número de secuencia del primer byte enviado que todavía no ha sido reconocido. Se trata del primer byte de la Categoría 2, todos los previos pertenecen a la Categoría 1. El nombre del puntero se refiere a bytes enviados (*send*) no reconocidos (*unacknowledge*)

- **SND.NXT**: este puntero almacena el número de secuencia del siguiente byte de datos a ser transmitido. Marca el inicio de la Categoría 3. El nombre del puntero se refiere al siguiente byte (*next*) a enviar (*send*)

- **SND.WND**: este puntero guarda el tamaño de la ventana (*window*) de transmisión (*send*), especificado como la cantidad de bytes transmitidos que todavía no han sido reconocidos. Si se suma el valor de SND.UNA con el de SND.WND, se obtiene el número de secuencia del primer byte de la Categoría 4.

La cantidad de bytes que conforman la Categoría 3 se puede obtener mediante la operación $SND.UNA + SND.WND - SND.NXT$. Este valor es lo que se conoce como ventana útil, ya que se corresponde con la cantidad de bytes extra que el transmisor podría llegar a enviar. También observando la variación de los valores de los punteros se puede apreciar el carácter deslizante de la ventana, ya que toda vez que se reciban segmentos ACK, aumentará el valor del puntero SND.UNA, moviendo bytes hacia la Categoría 1 y, por ende deslizando el borde izquierdo de la ventana. Si el valor de SEND.WND permaneciera

constate, este efecto producirá el correspondiente deslizamiento del borde derecho. La consecuencia inmediata es que aumentará el tamaño de la ventana útil.

Las Categorías y punteros asociados a la ventana deslizante de transmisión tienen una relación directa con Categorías y punteros asociados a la ventana deslizante de recepción, en el otro extremo de la comunicación. Este concepto se desarrolla en la Fig. 11.5, donde se distinguen las siguientes tres Categorías:

- **Bytes ya recibidos y que han sido reconocidos con ACK**: no es necesario mantener estos datos en el buffer de recepción, puesto que ya han sido recibidos correctamente y subidos a la aplicación.

- **Bytes todavía no recibidos, con espacio en el buffer del receptor para ser aceptados**.

- **Bytes todavía no recibidos pero que el receptor no estaría dispuesto a recibir por falta de espacio en el buffer**

Por ejemplo, en el caso de la Fig. 11.5, el receptor todavía no ha recibido bytes con numeración posterior al número de secuencia 32 y cuenta con espacio para recibir 20 $bytes$.

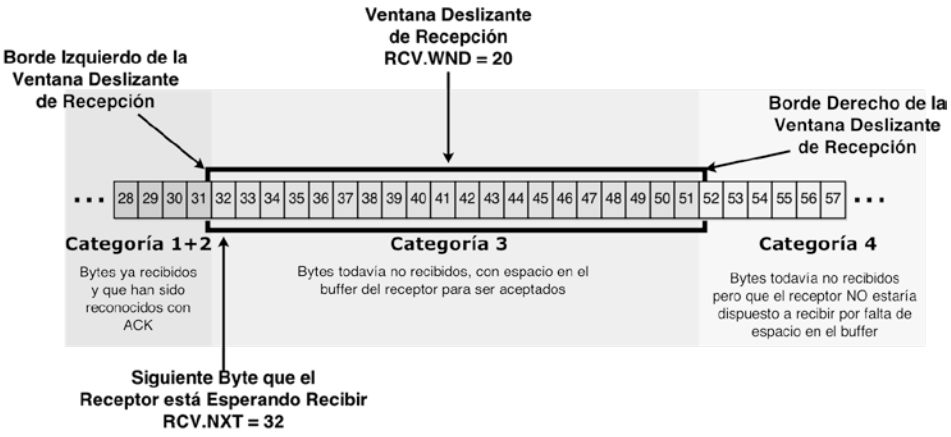


Figura 11.5 – Ventana conceptual de recepción.

Las tres Categorías de recepción se distinguen mediante el uso de los siguientes punteros:

- **RCV.NXT**: se trata del número de secuencia del siguiente (*next*) byte que el receptor está esperando recibir (*receive*). Este valor se corresponde

con el primer byte de la Categoría 3. Todos los números previos se han recibido y reconocido.

- **RCV.WND**: es el tamaño del buffer de recepción que se anuncia al otro extremo.

El aviso de recepción correcta debe indicar el número del siguiente byte que se espera recibir en el campo de Número de ACK del encabezado TCP. Dicho de otro modo, un segmento ACK avisa hasta cuál byte se ha recibido correctamente y en orden. Cuando se pierden bytes en el camino, si se reciben segmentos fuera de orden, el protocolo original sólo puede repetir el valor del Número de ACK del último byte recibido correctamente y en orden. La definición de nuevas opciones para el protocolo TCP permite la posibilidad de reconocer segmentos por bloques de bytes. Este mecanismo se denomina reconocimiento selectivo y mejora sobremanera la eficiencia del protocolo.

11.5 Encabezado TCP

Los segmentos TCP poseen un encabezado de 20 $bytes$ fijo, aunque se permite el agregado de opciones. En general, los segmentos transportan información de control y datos simultáneamente.

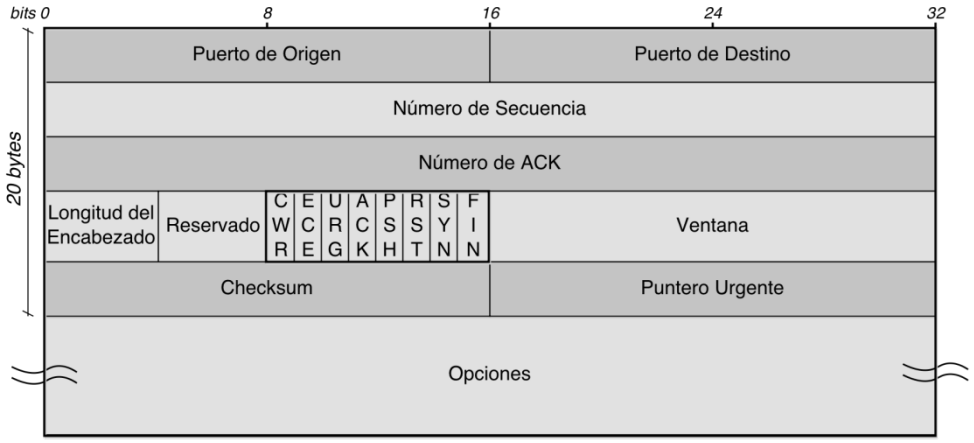


Figura 11.6 – Encabezado TCP

El formato del encabezado, que se presenta en la Fig. 11.6, consta de los siguientes campos:

- **Puerto Fuente** ($16\ bits$): se trata de un número que identifica al proceso que originó el segmento. Generalmente será un puerto efímero en requerimientos de cliente a servidor y un puerto bien conocido en las respuestas al cliente.

- **Puerto Destino** ($16\ bits$): de la misma longitud que el puerto fuente, identifica el proceso destino del segmento que, junto con la dirección IP destino, conforma el socket destino. El par de sockets fuente y destino identifican una conexión.

- **Número de Secuencia** ($32\ bits$): identifica el número del primer byte de datos que transporta el segmento. Los bytes de un flujo de datos en un único sentido se numeran consecutivamente a partir de un Número de Secuencia Inicial (ISN, Initial Sequence Number) que se enuncia en el primer segmento de una conexión TCP. El Número de Secuencia es un campo de $32\ bits$ que vuelve a "0" si se alcanza el valor máximo posible $2^{32} - 1 = 4.294.967.295$. Es decir que se deberán transmitir más de $4\ Gbytes$, desde un extremo al otro de una conexión TCP, para que la numeración vuelva al punto inicial.

- **Número de ACK** ($32\ bits$): contiene el siguiente Número de Secuencia que el transmisor del segmento de ACK espera recibir. Se trata del Número de Secuencia del último byte de datos recibido correctamente y en orden, más 1. Este campo es válido si la Bandera de ACK está en "1", situación que se repite en todos los segmentos intercambiados en una conexión, excepto en el primero. Como se ha mencionado, la forma de reconocimiento de datos de TCP es acumulativa. Por ejemplo, suponiendo que un extremo de la conexión transmitiera tres segmentos cuyos Números de Secuencia fueran 1024 , 2024 y 3024 respectivamente, es decir que cada segmento transportara $1000\ bytes$ de datos, si en el receptor se recibiera el primer segmento correctamente, se editaría un segmento ACK con Número de ACK 2024. Pero si luego llegara el tercer segmento, correctamente aunque fuera de orden, se re-editaría el segmento ACK previo, recibiéndose un ACK duplicado en el extremo transmisor. De este modo, la recepción de ACK duplicados indica la posible existencia de algún tipo de congestión en la red. En el ejemplo mencionado, recién cuando se recibiera el segundo segmento, se podría editar un segmento con Número de ACK 4024, apuntando al byte que se espera recibir y asegurando que todos los previos llegaron correctamente y en orden. Los Sistemas Operativos más modernos incluyen nuevas opciones TCP que permiten realizar ACK Selectivo (SACK, Selective ACK) si ambos extremos lo soportan.

- **Longitud del Encabezado** ($4\ bits$): número de palabras de $32\ bits$ que conforman el encabezado TCP. También se conoce este campo como *Data Offset* ya que indica dónde comienzan los datos. El campo es

necesario porque existe la posibilidad de agregar 40 $bytes$ de opciones como máximo.

- **Bandera CWR**: es la bandera de Ventana de Congestión Reducida (CWR, Congestion Window Reduced) que es levantada por el transmisor del segmento para indicar que recibió un segmento TCP con la bandera ECE en alto y ha respondido con algún mecanismo de control de congestión. Fue agregado al encabezado original por la RFC 3168.

- **Bandera ECE**: es la bandera de Eco de Notificación de Congestión Explícita (ECE, Explicit Congestion Notificación Eco) para que el receptor pueda informar al transmisor cuando ha recibido un paquete con Congestión Experimentada (CE, Congestion Experienced). ECN es una extensión opcional de IP y TCP, definida en la RFC 3168, que permite notificación de congestión entre extremos finales sin descarte de paquetes. Tradicionalmente, las situaciones de congestión se detectan por pérdida de paquetes. Sin embargo, si se negocia ECN entre los extremos TCP, los *routers* que poseen la capacidad pueden marcar los encabezados IP para avisar situaciones de congestión inminente, antes de comenzar a descartar los paquetes. El receptor del paquete marcado, levanta la bandera ECE para avisar al transmisor, que reduce el flujo de transmisión, tal como si hubiese detectado paquetes perdidos.

- **Bandera URG**: esta bandera se levanta para indicar prioridad en la entrega de datos de un segmento. En este caso, el campo Puntero Urgente es válido.

- **Bandera ACK**: esta bandera, en estado levantado, indica que el campo Número de ACK es válido.

- **Bandera PSH**: bandera ideada para indicar al receptor que se solicita el pronto pasaje de datos recibidos a la aplicación.

- **Bandera RST**: el transmisor levanta la bandera cuando enfrenta algún tipo específico de problemas o errores. Su efecto es abortar la conexión.

- **Bandera SYN**: en el inicio de la conexión, el primer segmento lleva esta bandera levantada, para que los extremos se apresten a sincronizar sus respectivos ISN y establecer la conexión. El ISN es el número escrito en el campo Número de Secuencia del segmento con el bit de SYN en alto.

- **Bandera FIN**: cuando uno de los extremos levanta esta bandera, indica que requiere terminar la conexión pues ya no tiene datos para transmitir.
- **Ventana** (16 $bits$): con este campo, cada extremo de una conexión TCP avisa, al otro extremo, el tamaño disponible en su buffer de recepción. Dentro del esquema de control de flujo por ventana deslizante, el campo se debe interpretar como la cantidad de bytes, contados a partir del

número especificado en el campo Número de ACK, que el receptor estaría dispuesto a recibir.

- **Checksum** (16 $bits$): para protección de integridad. Se calcula sobre el segmento completo más un *pseudoheader* especial, similar al utilizado en *UDP*.

- **Puntero Urgente** (16 $bits$): campo válido cuando el bit URG se encuentra en "1". Se trata de un número, que debe sumarse al Número de Secuencia transportado en este segmento, para obtener el Número de Secuencia del último byte de datos considerados urgentes. Es un mecanismo utilizado por el transmisor para marcar ciertos datos de manera especial, para tratamiento urgente por parte del otro extremo.

Opciones: campo de longitud variable, de 40 $bytes$ como máximo. La Tabla 11.1 presenta las opciones definidas. Todas se especifican mediante un campo Tipo de 1 $byte$. Las opciones de más de un byte, llevan un campo Longitud de 8 $bits$. La opción más conocida es la del tamaño máximo de segmento MSS, anunciado en los segmentos SYN durante el establecimiento de una conexión.

Tabla 11.1 – Opciones TCP

Tipo	Longitud	Valor	Descripción
0	-	-	**Fin de lista de Opciones**: un byte único al final de las opciones. Su único propósito es alinear las opciones en palabras de 32 $bits$, cuando el fin de una opción no coincide con el marco esperado del encabezado.
1	-	-	**No Operación**: un byte que se usa como separador entre opciones, para contribuir a la alineación en palabras de 32 $bits$.
2	4	MSS	**Tamaño Máximo de Segmento**: opción sólo usada en un segmento con el bit SYN levantado, para anunciar al otro extremo el tamaño máximo del segmento TCP. El valor depende de la MTU.
3	3	Factor de Escalamiento de Ventana	**Escalamiento de Ventana**: Se implementa esta opción cuando el tamaño de la ventana anunciada supere el valor máximo que se puede anunciar en el campo Ventana

Tipo	Longitud	Valor	Descripción
			(65.535 $bytes$). El valor es una potencia de dos, anunciado al inicio, y usado durante la conexión para multiplicar el valor anunciado en el campo Ventana y así obtener el verdadero valor que se está utilizando.
4	2	-	**SACK permitido**: opción para indicar que el extremo soporta ACK Selectivo.
5	variable	Block ACK	**SACK***:* si se permite SACK, esta opción despliega los bloques de datos no contiguos recibidos, para que no sean retransmitidos
8	10	Sellos de Tiempo	**Sellos de Tiempo**: opción utilizada para medir tiempos de ida y vuelta en una comunicación. También se puede usar como protección cuando el campo Número de Secuencia se consume y vuelve a empezar dentro de una misma conexión.
14	3	Algoritmo de Checksum Alternativo	**Requerimiento de Checksum Alternativo***:* Opción que permite solicitar un algoritmo de generación de *checksum* diferente. Debe haber un acuerdo entre ambas partes.
15	variable	Checksum Alternativo	**Checksum Alternativo**: Si el valor del *checksum* alternativo no cabe en el campo de 16 $bits$ del encabezado, se coloca en esta opción.
28	4	UTO	**UTO**: nueva opción en la que un extremo puede anunciar el valor de expiración (User Timeout) de datos pendientes.
29	variable	TCP-AO	**TCP-AO**: nueva opción que permite el uso de algoritmos criptográficos para llevar adelante la autenticación de los segmentos de una conexión.

Es interesante destacar que la carga de datos es también opcional, en el sentido de que no tienen que estar obligatoriamente presentes. Por ejemplo, en el inicio de la conexión, los segmentos no cargan datos. También, en el transcurso de una conexión, se pueden enviar segmentos con la bandera ACK levantada, sin

datos, con fines exclusivos de reconocimiento de datos recibidos o sólo para el anuncio de una modificación en el tamaño de la ventana.

A medida que nos vayamos adentrando en la funcionalidad del protocolo, se desarrollarán con mayor detalle aspectos de varias de las opciones mencionadas en la Tabla 11.1.

11.6 Inicio y Terminación de una Conexión TCP

Al haber sido definido como un protocolo orientado a la conexión, ambos extremos de una comunicación TCP deben establecer una conexión antes de empezar a intercambiar datos. También se define una forma especial de finalizar la conexión.

11.6.1 Inicio de una Conexión TCP

El proceso de inicio de una conexión TCP se realiza de manera independiente para cada conexión que, a su vez, queda identificada por un par de *sockets* y asociada con su propia estructura de datos intercambiados, a modo de indicación del estado de la misma. Por ejemplo, el protocolo de ventana deslizante exige llevar el registro de los números de bytes recibidos y ya reconocidos, transmitidos y aún no reconocidos y aquellos que se podrían llegar a transmitir puesto que el receptor estaría dispuesto a recibirlos. Este registro obliga, antes de pasar por la fase de inicio, a preparar un Bloque de Control de la Transmisión (TCB, Transmission Control Block) que el protocolo utilizará para guardar toda la información necesaria para esa conexión.

Dentro del paradigma cliente/servidor, el inicio de una conexión TCP depende del rol que se asigna a cada extremo de la comunicación. Tanto cliente como servidor realizarán una operación conocida como apertura u OPEN. Según el rol que asuma cada extremo, esta operación se conoce con distinto nombre:

- **OPEN Activo**: se llama así a la operación realizada por el que inicia la conexión, enviando un segmento con la bandera SYN en alto al otro extremo. Este papel generalmente lo asume el cliente, pues es quien precisará algún recurso alojado en el servidor. Se denomina activo porque supone la transmisión de un segmento sobre la red.

- **OPEN Pasivo**: es la operación que realiza el otro extremo de la comunicación. En general, el servidor es un proceso en espera de solicitudes de conexión, escuchando en el número de puerto que se le haya asignado. El motivo por el que se lo llama pasivo se debe a esta espera.

Sea cual sea el extremo que inicie la conexión, al momento del OPEN, ambos crean una estructura TCB asociada a la misma. Dicha estructura contendrá información sobre el estado de la conexión y será mantenida hasta la terminación.

Al momento de editar el segmento SYN, el cliente cuenta con información sobre el par de *sockets* asociado a la conexión. Sin embargo, el servidor sólo puede identificar la conexión por el par de sockets al momento de recibir dicho segmento. El proceso de intercambio de segmentos TCP al momento del inicio, se suele denominar Apretón de Manos de Tres Vías (3- way handshake), en referencia al intercambio de tres mensajes. Una de los objetivos más importantes del inicio es sincronizar los ISN e intercambiar parámetros que influirán en la operación posterior del protocolo TCP, para la conexión particular.

Tal como se presenta en la Fig. 11.7, para establecer la conexión, cada extremo transmite un segmento SYN y recibe un segmento ACK del otro extremo. En la figura, se ha denominado cliente al extremo que realiza el OPEN Activo, y servidor al otro extremo. El cliente transmite un segmento con la bandera de SYN levantada y un número de secuencia inicial ISN, propio del lado cliente, que en la figura se designa como "M". A recibir este segmento, el extremo servidor contesta, enviando un segmento SYN con su propio ISN, que en la figura se representa con la letra "N". Este segmento tiene además levantada la bandera de ACK, para reconocer el primer segmento enviado por el cliente, con un Número de ACK asociado que apunta al siguiente byte que el servidor espera recibir: "$M + 1$" . Finalmente, el cliente debe reconocer el segmento recibido desde el servidor, transmitiendo un segmento con la bandera de ACK en alto y Número de ACK "$N + 1$”. Como se puede observar, en total se intercambian tres segmentos, de ahí el nombre de Tres Vías.

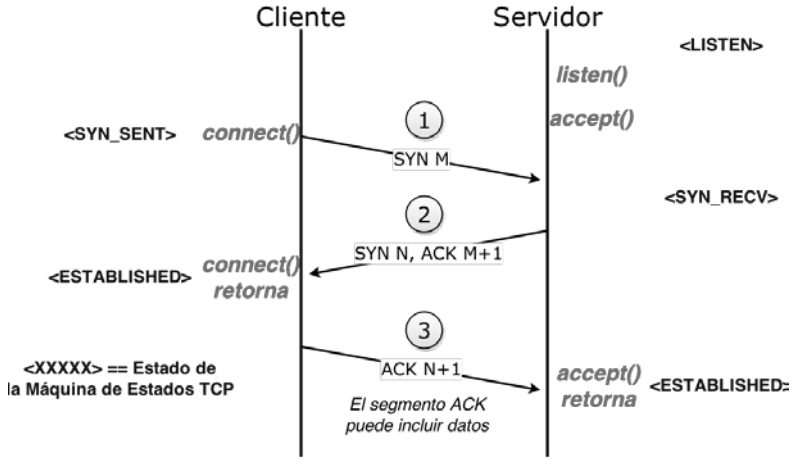


Figura 11.7 – Inicio de una conexión TCP

En la Fig. 11.7, se destacan las llamadas al sistema que realiza cada extremo al editar los segmentos y los diferentes estados por los que pasan, tanto cliente como servidor. Las llamadas al sistema se han abordado en el apartado

referido a *Sockets*, en este mismo Capítulo. Profundizaremos sobre detalles de los estados más adelante, cuando se exponga el funcionamiento de la Máquina de Estados de TCP.

En un caso real, levantado por un *sniffer*, se puede observar un primer segmento que viaja desde cliente a servidor, que podría ser del siguiente tipo:

Transmission Control Protocol
Src Port: 49265 // Dst Port: 80 // Sequence number: 28358 // Header |length: 32 bytes //
Flags: 0x0002 (SYN)
0... .... = Congestion Window Reduced (CWR): Not set
.0.. .... = ECN-Echo: Not set
..0. .... = Urgent: Not set
...0 .... = Acknowledgment: Not set
.... 0... = Push: Not set
.... .0.. = Reset: Not set
.... ..1. = Syn: Set
....0 = Fin: Not set
Window size: 2097152 (scaled)
Checksum: ---
Options: (12 bytes)
TCP MSS Option: True
Maximum segment size: 1460 bytes
NOP
Window scale: 8 (multiply by 256)
NOP
NOP
SACK permitted

En este segmento se aprecia que la Longitud del Encabezado es de $32\ bytes$, indicando la existencia de opciones. Muchos *sniffers* presentan el valor ISN como "0" porque interesa la numeración relativa, pero en este segmento se presenta la numeración inicial ISN del lado cliente, expresada en numeración decimal: "28358". Se observa que el segmento posee una única bandera levantada, la de SYN.

En el campo de opciones se anuncia el valor MSS de la máquina origen, $1460\ bytes$. Evidentemente se corresponde al caso de una LAN cableada, cuya MTU es de $1500\ bytes$, valor al que hay que restarle la longitud de las cabeceras convencionales de IP y TCP, para hallar el valor MSS. También, en este caso, en el campo de opciones, se observan algunas cabeceras de alineamiento (NOP), y el anuncio de que el extremo iniciador de la conexión tiene capacidad para realizar escalamiento de la ventana y trabajar con ACK selectivo. Junto con el anuncio de escalamiento se pasa el valor con el que se debe escalar el anuncio, en este caso 8, significando en realidad 2^8. Si se suman los campos de las opciones 4 bytes (MSS) $+$ 3 bytes (NOP) $+$ 2 bytes (SACK permitido) $+$ 3 bytes (Escalamiento) $= 12$ bytes, se puede confirmar que el valor de longitud del encabezado, de $32\ bytes$, es correcto.

También con un *sniffer*, se ha obtenido la respuesta del servidor:

Transmission Control Protocol
Src Port: 80 // Dst Port: 49265 // Sequence number: 38353 // Ack Number: 28359 // Header length: 32 bytes
Flags: 0x0012 (SYN, ACK)
0... .... = Congestion Window Reduced (CWR): Not set
.0.. .... = ECN-Echo: Not set
..0. .... = Urgent: Not set
...1 .... = Acknowledgment: Set
.... 0... = Push: Not set
.... .0.. = Reset: Not set
.... ..1. = Syn: Set
....0 = Fin: Not set
Window size: 373760 (scaled)
Checksum: ----
Options: (12 bytes)
TCP MSS Option: True
Maximum segment size: 1460 bytes
NOP
NOP
SACK permitted
NOP
Window scale: 6 (multiply by 64)

Este segundo segmento del protocolo de Tres Vías, lleva levantadas dos banderas: SYN y ACK. Se puede verificar que el Número de ACK es el correspondiente al ISN del cliente más uno. El servidor, por su parte, levanta con su propio ISN, en este caso "38353" , y anuncia sus propias opciones de operación que, casualmente, coinciden con las capacidades del cliente, aunque se anuncia otro factor para el escalamiento de la ventana.

Por último, se completa el inicio de la conexión TCP con un tercer segmento, transmitido de cliente a servidor:

Transmission Control Protocol
Src Port: 49265 // Dst Port: 80 // Seq: 28359 // Ack: 38354 // Header length: 20 bytes //
Flags: 0x0010 (ACK)
0... .... = Congestion Window Reduced (CWR): Not set
.0.. .... = ECN-Echo: Not set
..0. .... = Urgent: Not set
...1 .... = Acknowledgment: Set
.... 0... = Push: Not set
.... .0.. = Reset: Not set
.... ..0. = Syn: Not set
....0 = Fin: Not set
Window size: 65536 (scaled)
Checksum: ----

Este segmento lleva sólo la bandera de ACK levantada, su Número de Secuencia es el que espera recibir el servidor y su Número de ACK es el ISN del servidor más uno, indicando que el previo se recibió correctamente. Con este último segmento del inicio, ambos extremos logran sincronizar sus números de secuencia y almacenar los parámetros necesarios para la comunicación.

A partir de este momento, la conexión queda establecida, permitiéndose que comience el intercambio de datos. En la comunicación, se aceptarán todos aquellos segmentos con el mismo par de *sockets*, cuyo *checksum* se verifique como correcto y cuyo Número de Secuencia sea válido.

11.6.2 Número de Secuencia Inicial ISN

La selección del ISN no es una cuestión menor, ya que se realiza de modo tal de evitar confusiones entre segmentos pertenecientes a distintas conexiones. La RFC 793 especifica que el valor de ISN debe provenir de un contador de $32\ bits$ que se incrementa cada $4\ \mu seg$, es decir que se trata de un parámetro variable con el tiempo. El propósito es que los números de secuencia de segmentos intercambiados entre el mismo par de sockets, pero en conexiones diferentes, realizadas en distinto tiempo, no colisionen entre sí. Este tipo de conexiones se conocen como **encarnaciones,** definiéndose como diferentes instancias de una conexión entre el mismo par de *sockets*. El problema de las encarnaciones se presenta si un segmento retrasado de una conexión ya terminada, aparece dentro de una nueva conexión, generando riesgos potenciales que deben ser minimizados a través de una selección apropiada del ISN.

El hecho de que un segmento sea aceptado en una conexión, mientras su número de secuencia sea válido, representa una vulnerabilidad del protocolo. Si un atacante conoce el par de sockets y estima los números de secuencia en uso, a partir de valores ISN utilizados previamente, puede falsificar un segmento TCP con el propósito de interrumpir una conexión en curso. Por este motivo, los sistemas operativos modernos tienden a seleccionar el ISN de manera casi aleatoria, a través de un algoritmo, tal como se contempla en la RFC 6528.

11.6.3 Tiempo de Expiración en el Inicio de una Conexión TCP

Pueden existir circunstancias que no permitan el establecimiento de una conexión. Por ejemplo, si se pretende contactar un servidor fuera de la red propia y el *router* de salida de la red se encuentra caído, no existirán respuestas a los requerimientos ARP y se informará esta situación al cliente con algún tipo de mensaje de error.

Otra situación posible podría referirse a la propia caída de la máquina servidora, considerando la misma fuera de la propia red del cliente. En este caso, el problema no se manifiesta por la ausencia de respuesta ARP, sino porque no se completa el protocolo de tres vías. En este caso, TCP intentará establecer la conexión transmitiendo segmentos SYN de manera repetida. Lo interesante sería

observar los tiempos en los que el protocolo envía cada nuevo segmento SYN y cuántas veces lo intenta. En realidad, *TCP* aumenta exponencialmente los tiempos entre cada envío, realizando un retroceso o *back off* exponencial no aleatorio. El máximo número de intentos es configurable en algunos sistemas operativos, aunque cinco veces es un valor normal. Finalizado este tiempo, se avisa a la aplicación cliente con algún mensaje de error.

11.6.4 Fin de la Conexión TCP

Una vez establecida la conexión, los extremos finales pueden enviar datos en ambas direcciones. Eventualmente, alguno de los procesos no tendrá más datos para enviar e iniciará el cierre, ya sea por sí mismo o por orden del usuario de la aplicación.

Como sucede en el inicio, el cierre de una conexión TCP también implica un intercambio especial de segmentos, pasando el protocolo por una serie de estados.

Debido a que el fin de la conexión debe realizarse de manera segura, ambos extremos deben cerrar por su cuenta, asegurándose de que el otro también lo haya hecho y evitando perder datos en el proceso. Por este motivo, iniciar el fin de una conexión es avisar al otro extremo que no se enviarán más datos pero, a pesar de ello, se estaría dispuesto a recibirlos, de ser necesario.

Para poder cerrar una conexión se levanta la bandera de FIN en un segmento que puede transportar datos pero, para el que primero lo envía, se trata de los últimos datos que transmitirá. El extremo que recibe el segmento FIN, debe responder con un segmento ACK. Aún así, el proceso continúa hasta que el que extremo que recibió el FIN transmita su propio segmento FIN y reciba un ACK. Evidentemente, se deben intercambiar cuatro segmentos para que esto sea posible.

La Fig. 11.8 presenta el fin de una conexión TCP con los estados asociados a cada transmisión o recepción de segmentos. El dispositivo que inicia el cierre transmite un segmento FIN cuyo número de secuencia dependerá de la cantidad de datos que el mismo haya transmitido. En la figura, genéricamente, se ha denominado "M" al Número de Secuencia que se asocia al segmento FIN que transmite el cliente, que en este caso se dice que realiza el cierre o CLOSE activo. El servidor contesta con un segmento cuya bandera de ACK se encuentra levantada y cuyo Número de ACK se corresponde con el número "$M + 1$". Por su parte, el servidor transmite luego su propio segmento FIN, con el número de secuencia que corresponda, genéricamente "N", por lo que el Número de ACK de la respuesta deberá ser "$N + 1$".

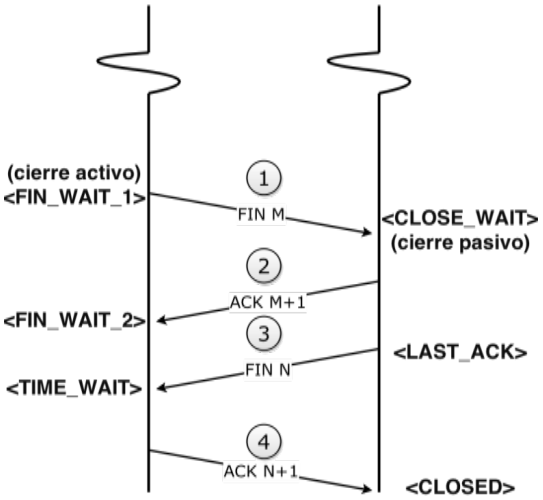


Figura 11.8 - Fin de una conexión TCP

Continuando con el caso real del apartado previo sobre el inicio de una conexión, a continuación se presentan los segmentos intercambiados en el cierre de la misma, levantados por el *sniffer*.

El primer segmento es transmitido por el cliente, con un Número de Secuencia "28872" y un Número de ACK "38721". Si restáramos a este número de secuencia el valor del ISN del cliente, el valor obtenido como resultado nos permite saber la cantidad de bytes transmitidos por el cliente: $NS(FIN)_C - ISN_C = 28872 - 28358 = 514\ bytes$. En realidad el cliente ha transmitido $513\ bytes$, un número menos debido a que el segmento SYN consume un número de secuencia aunque no transporta datos. A su vez, si restáramos al Número de ACK recibido, el valor de ISN del servidor, $NACK_C - ISN_S = 38721 - 38353 = 368\ bytes$, podríamos saber el número de bytes que el servidor le ha enviado al cliente, en este caso $367\ bytes$.

Transmission Control Protocol
Src Port: 49265 // Dst Port: 80 // Seq: 28872 // Ack: 38721 // Header length: 20 bytes
Flags: 0x0011 (FIN, ACK)
0... .... = Congestion Window Reduced (CWR): Not set
.0.. .... = ECN-Echo: Not set
..0. .... = Urgent: Not set
...1 .... = Acknowledgment: Set
.... 0... = Push: Not set
.... .0.. = Reset: Not set
.... ..0. = Syn: Not set
....1 = Fin: Set
Window size: 65280 (scaled)
Checksum: ----

En este ejemplo particular, como el servidor ya no tiene más datos que enviar, contesta con un único segmento, con ambas banderas levantadas. El número de secuencia es el que el cliente está esperando recibir: "38721". El número de ACK es "28873", indicando que se ha recibido correctamente hasta el byte numerado "28872".

Transmission Control Protocol
Src Port: 80 // Dst Port: 49265 // Seq: 38721 // Ack: 28873 // Header length: 20 bytes
Flags: 0x0011 (FIN, ACK)
0... .... = Congestion Window Reduced (CWR): Not set
.0.. .... = ECN-Echo: Not set
..0. .... = Urgent: Not set
...1 .... = Acknowledgment: Set
.... 0... = Push: Not set
.... .0.. = Reset: Not set
.... ..0. = Syn: Not set
....1 = Fin: Set
Window size: 6912 (scaled)
Checksum: ----

El último segmento debe ser el ACK de contestación del cliente al segmento FIN transmitido por el servidor:

Transmission Control Protocol
Src Port: 49265 // Dst Port: 80 // Seq: 28873 // Ack: 38722 // Header length: 20 bytes
Flags: 0x0010 (ACK)
0... .... = Congestion Window Reduced (CWR): Not set
.0.. .... = ECN-Echo: Not set
..0. .... = Urgent: Not set
...1 .... = Acknowledgment: Set
.... 0... = Push: Not set
.... .0.. = Reset: Not set
.... ..0. = Syn: Not set
....0 = Fin: Not set
Window size: 65280 (scaled)
Checksum: ----

Es muy común que el extremo cliente cierre la conexión, pero hay servidores Web que realizan el CLOSE activo, una vez que han entregado los datos solicitados por el cliente. Generalmente, una terminación normal de la conexión se realiza invocando la llamada al sistema *close()*.

11.6.5 Fin de la Conexión TCP – Estado Medio Cerrado (Half Close)

En el cierre de la conexión TCP bidireccional, cada extremo debe enviar un segmento FIN cuando ya no tiene datos para enviar al otro. En el caso de que

TCP reciba un segmento FIN debe notificar a la aplicación que el otro extremo ya no tiene datos para transmitir.

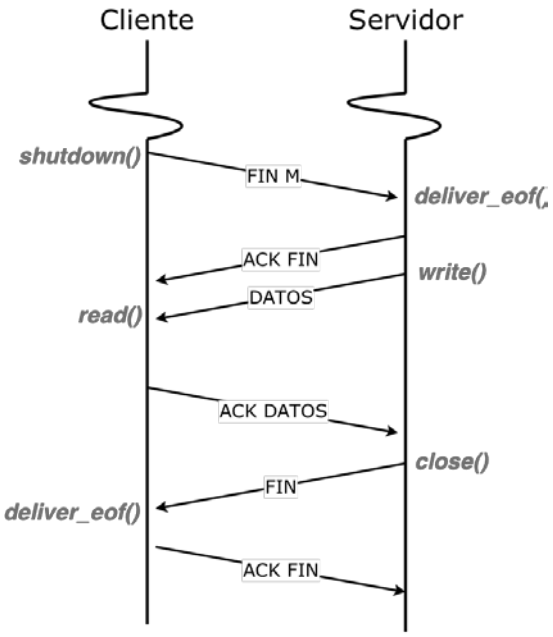


Figura 11.9 – Operación Half Close

Ciertos *sockets* API soportan una operación conocida como Half Close, que se presenta en la Fig. 11.9. En este caso, la aplicación cierra la conexión con la llamada al sistema *shutdown()*, indicando que ya no tiene más datos para enviar pero puede recibirlos. Como en el caso normal, se editará un segmento FIN, recibiendo un segmento ACK proveniente del otro extremo. La diferencia es que, antes de que el otro extremo realice el cierre, puede transmitir datos, que serán reconocidos por el extremo en estado Half Close. Una vez finalizada la transmisión, el otro extremo cerrará por su cuenta normalmente: editando un segmento FIN y esperando la contestación ACK, tal como se aprecia en la figura.

11.6.6 Apertura y Terminación Simultáneas

Es posible que dos aplicaciones realicen aperturas activas entre sí al mismo tiempo, en un escenario conocido como apertura u OPEN simultáneo. En este caso, cada extremo debe transmitir un segmento SYN al puerto bien conocido del otro extremo, cruzándose los segmentos en la red. Por este motivo, se dice que es posible, aunque sea bastante improbable.

Por ejemplo, una aplicación en el host A, desde el puerto local X, envía un segmento SYN a otra aplicación, en el puerto Y del host B. Al mismo tiempo,

la aplicación en el host B, desde el puerto local Y, envía un segmento SYN a la aplicación en el puerto X del host A. Para que esto sea posible, cada aplicación debe conocer el puerto donde la otra escucha

TCP impone reglas para el manejo de esta situación: sólo una de las conexiones podrá ser establecida. Cuando se recibe el SYN proveniente del otro extremo, se lo debe reconocer y retransmitir el SYN enviado antes. De este modo, la apertura se realiza mediante el intercambio de cuatro segmentos y no se puede distinguir entre cliente y servidor, ya que ambos extremos actúan de las dos maneras. La Fig. 11.10 presenta esta situación.

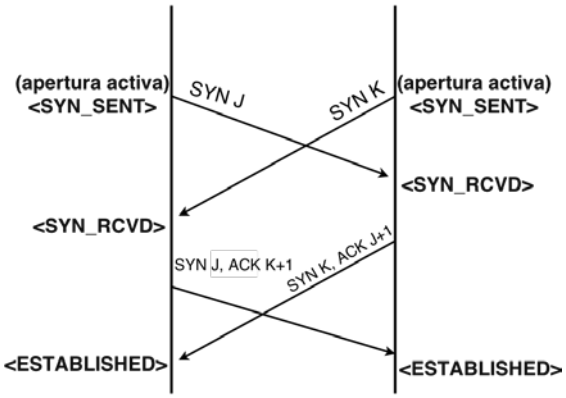


Figura 11.10 - Open simultáneo.

Una utilidad actual de la posibilidad descripta se denomina ***hole punching*** y sirve para establecer comunicaciones entre dos partes, en distintas organizaciones, que se encuentran detrás de un NAT restrictivo. En este caso, para que la comunicación sea posible, primero debe haberse realizado una comunicación saliente para crear la correspondiente línea de mapeo en el NAT. Pero si el otro extremo también se encuentra detrás de un NAT, no hay forma de atravesarlo pues también se precisa que se haya establecido el mapeo, para poder saber la dirección IP y el número de puerto del otro. La técnica de *hole punching* se usa para juegos *online*, comunicación entre pares P2P y VoIP. Se refiere al caso de dos dispositivos, detrás de un NAT, que están tratando de conectarse entre sí por medio de conexiones TCP salientes. El comportamiento no se encuentra estandarizado, por lo que no funciona con cualquier NAT.

Uno de los motivos por el que sucede esta imposibilidad de la comunicación es para evitar ataques. Por protección, no se permite la entrada de segmentos SYN, excepto sobre las máquinas servidoras (que poseen un mapeo estático en el NAT), o en respuesta a un segmento SYN saliente. Por otra parte, cada máquina detrás de un NAT desconoce, tanto su propio mapeo de salida como el mapeo de salida de la otra máquina, detrás de otro NAT. Para que sea posible la comunicación entre ellas, se debe resolver un problema que se conoce como

predicción de los puertos: adivinar el punto remoto público del par. Si alguno de los NAT, detrás de los cuales se encuentran los protagonistas de la comunicación, utiliza un algoritmo predecible de asignación de puertos, la comunicación es posible. En ciertos casos, se necesita la ayuda de un tercer protagonista, un servidor en Internet, entre ambas redes.

Por ejemplo, si el NAT asigna puertos externos de manera secuencial a puertos internos secuenciales, primero los pares que desean comunicarse se conectarán a una tercera parte y conocerán el mapeo, luego podrán adivinar las asignaciones subsiguientes. Si, en cambio, el NAT utiliza un esquema de preservación de puertos, en el que el puerto fuente del dispositivo interno siempre se mapea al mismo puerto público, la predicción es trivial y los pares simplemente se comunican los números de puerto a través de otro canal de comunicación, antes de realizar el *OPEN* simultáneo en TCP.

Una vez realizada la predicción de los puertos, ambos extremos envían un segmento SYN al otro. Cuando el NAT de cada una de las redes recibe el segmento SYN saliente que proviene del dispositivo interno, crea un mapeo. Luego, los segmentos se cruzan en algún punto de la red y, según cuál llegue primero, eventualmente uno de los NAT permitirá el paso del segmento SYN, mapeándolo al dispositivo interno. Este contestará con un segmento SYN con la bandera de ACK encendida. Así se podrá establecer una conexión entre dispositivos ubicados detrás de un NAT.

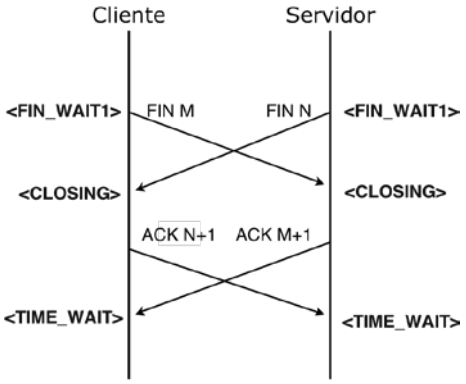


Figura 11.11 - Close simultáneo.

Así como el protocolo permite situaciones de apertura simultánea, también considera el caso en que los segmentos FIN de dos extremos de una comunicación TCP se entrecrucen en la red, en un escenario conocido como terminación o CLOSE simultáneo, tal como se presenta en la Fig. 11.11. Este sería el caso si el cliente y el servidor realizaran un CLOSE activo al mismo tiempo. TCP impone una regla para esta situación, obligando a cada extremo a reconocer el FIN recibido del otro lado. El resultado es que se intercambia el mismo número de segmentos que en una terminación normal. La diferencia es

que los segmentos no se presentan de la manera secuencial habitual, sino intercalados.

11.7 Diagrama de Estados TCP

Las reglas que manejan, tanto el inicio como el fin de una conexión TCP, pueden resumirse en un Diagrama de Estados como el que se presenta en la Fig. 11.12. Mientras dure una conexión TCP, los extremos finales pasarán por una serie de estados que se pueden asociar al comportamiento de una Máquina de Estados Finita (FSM, Finite State Machine), descripta por una serie de situaciones entre las que pueden existir transiciones disparadas por determinados eventos que también pueden generar ciertas acciones.

Los nombres de los estados, once en total, se corresponden con los resultados obtenidos al invocar el comando *netstat*. El mismo muestra un listado de las conexiones activas de una computadora, tanto las de carácter entrante como las salientes.

En la Fig. 11.12, las transiciones típicas de un cliente se presentan en líneas llenas y las de un servidor, en líneas punteadas. Las transiciones que conducen al estado ESTABLISHED (establecido) se corresponden con el inicio de la conexión TCP, mientras aquellas que lo abandonan se refieren al cierre.

El punto de partida se señala como un estado **CLOSED** ficticio, desde el que parten y en el que terminan todas las conexiones TCP. A partir de este estado, un servidor realiza un ***open pasivo*** sobre un puerto TCP, preparando los parámetros TCB necesarios para aceptar conexiones entrantes. Esta acción coloca al servidor en el estado **LISTEN**, en espera de pedidos de conexión.

El cliente, por su parte, realiza un ***open activo*** mediante el envío de un segmento SYN, acción que prepara un TCB para esa conexión y lo coloca en el estado **SYN-SENT** (SYN enviado). Cuando el servidor recibe el segmento SYN, transmite un segmento con dos banderas encendidas, SYN y ACK, moviéndose hacia el estado **SYN-RCVD** (SYN recibido).

La transición normal al estado **ESTABLISHED** (establecido) es, para el cliente, cuando estando en SYN-SENT, recibe un segmento SYN con la bandera de ACK levantada, entonces transmite un segmento ACK. En el caso del servidor, la transición el estado ESTABLISHED es desde SYN-RECEIVED, cuando recibe un ACK.

El caso de apertura simultánea se contempla en la transición de SYN_SENT a SYN_RCVD, ya que habiendo enviado un segmento SYN, se recibe otro segmento de similares características, que se debe reconocer re-enviando el segmento SYN, pasando al estado SYN-RECEIVED, a la espera de un ACK que finalice el inicio de la conexión.

El estado **ESTABLISHED** se corresponde con la fase de intercambio de datos de una conexión TCP.

Para el cierre, la figura presenta dos casos, encerrados en rectángulos punteados: *close activo* y *close pasivo*. Si un dispositivo inicia el cierre, realizando el close activo, transmite un segmento FIN, situación que lo hace pasar

al estado **FIN-WAIT-1** (fin-espera-1), a la espera de un segmento ACK. Por su parte, el dispositivo que recibe el FIN, debe enviar un ACK, pasando al estado **CLOSE-WAIT** (cierre-espera). Este último realiza un *close pasivo*.

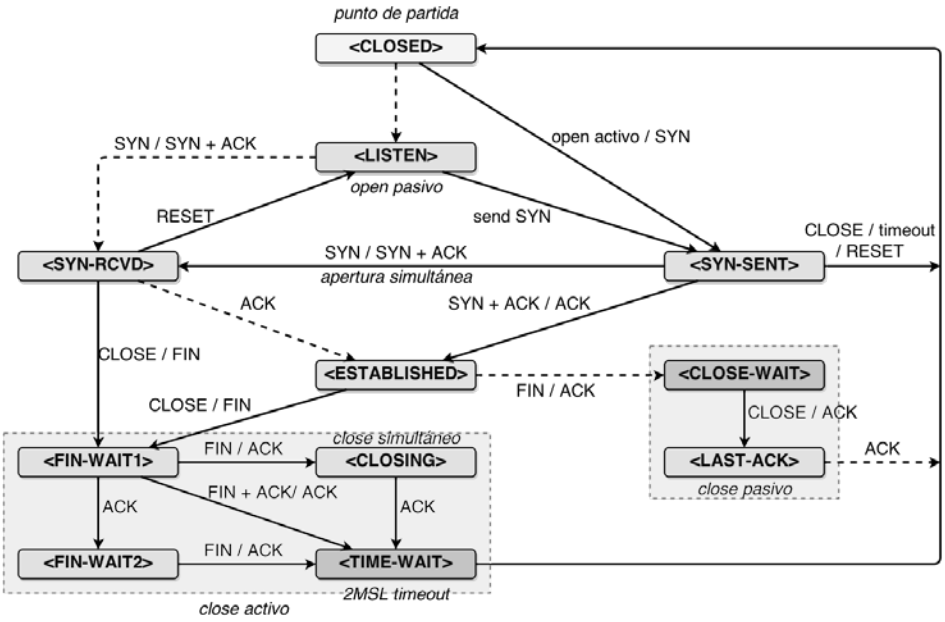


Figura 11.12 – Diagrama de Estados TCP

Una vez iniciado el cierre, el dispositivo cuyo protocolo TCP se encuentra en el estado **CLOSE-WAIT**, debe avisar a la aplicación que corre sobre TCP que el otro proceso desea finalizar la conexión. Cuando la aplicación lo requiera, TCP transmitirá un segmento FIN y la máquina pasará al estado **LAST-ACK** (último-ACK), a la espera de que el otro extremo reconozca este segmento.

Un dispositivo cuyo TCP se encuentre en el estado **FIN-WAIT-1**, puede moverse hacia dos estados, según los segmentos entrantes. Puede pasar al estado **FIN-WAIT-2**, por recepción de un segmento ACK, o puede moverse al estado **TIME-WAIT** (tiempo-espera), por que recibió un segmento FIN con la bandera de ACK levantada, y además transmitió un ACK.

La transición desde **FIN-WAIT-2** a **TIME-WAIT** se realiza cuando se recibe un segmento FIN y se transmite un ACK. El estado **FIN-WAIT-2** tiene asociado un reloj cuyo propósito es evitar que la conexión quede en este estado para siempre, a la espera de un segmento FIN. Esto podría llegar a suceder en el caso en que el otro extremo cayera. Si la conexión ha permanecido inactiva, cuando el reloj expira, TCP vuelve al estado **CLOSED**.

En el estado **TIME_WAIT** se enciende otro reloj, denominado 2MSL. La sigla significa Tiempo de Vida Máximo del Segmento (MSL, Maximum

Segment Lifetime), y se refiere a la cantidad de tiempo máxima que puede circular un segmento en la red. Ya se ha visto que el encabezado IP posee un campo TTL, asociado a una cuenta de saltos. Este valor efectivamente limita el tiempo de vida de un segmento. De todos modos, la RFC 793 especifica un valor de 2 *minutos* para reloj MSL, aunque implementaciones más modernas definen valores menores.

La conexión debe quedarse en ese estado por el doble de tiempo de MSL, para asegurarse que el ACK enviado verdaderamente llegue al otro extremo. De no ser así, se recibirá un nuevo segmento FIN, ya que el otro extremo tiene asignado un *timeout* a la transmisión de dicho segmento. Si no se recibe el ACK, se retransmitirá el segmento FIN. El propósito es asegurarse que la conexión se haya cerrado en ambos extremos.

Como se ha visto, el protocolo también permite un *close simultáneo*, representado por el caso en que ambos extremos pasan del estado **ESTABLISHED** al estado **FIN-WAIT-1**. Cada extremo transmite un segmento FIN y, cuando recibe el segmento FIN del otro extremo, pasa al estado **CLOSING** (cerrando), enviando un ACK. Cuando reciben el ACK, la transición es hacia el estado **TIME_WAIT**, donde se inicia el reloj 2MSL.

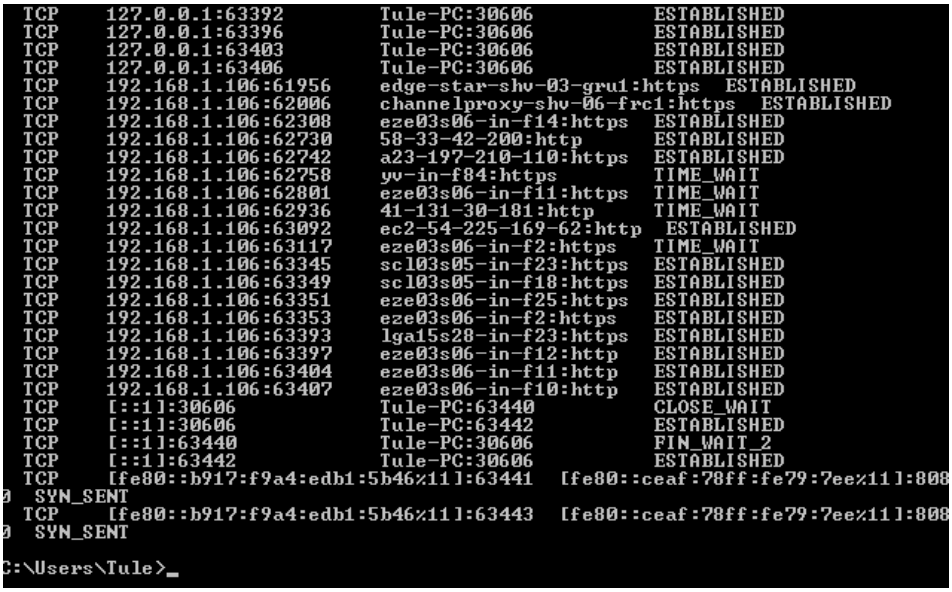


Figura 11.13 – Comando netstat

.

En la Fig. 11.13, se presenta una captura de la pantalla de una PC, resultante de un comando *netstat*. Se puede observar el par de sockets involucrado en cada conexión, así como el estado de la misma, coincidiendo su denominación con los estados del diagrama visto.

11.7.1 Espera 2MSL - Segmentos retrasados - Tiempo de Silencio (Quiet Time)

Mientras una conexión está establecida, el par de *sockets* que la caracteriza, no puede ser usado nuevamente, para evitar errores en futuras encarnaciones. No pueden existir nuevas instanciaciones de esta conexión hasta que no se haya agotado el reloj asociado al final, 2MSL. De suceder luego, la nueva conexión deberá comenzar con un valor ISN mayor. Además, se agrega una restricción: cualquier conexión en estado **TIME-WAIT** que reciba algún segmento retrasado que no sea un segmento FIN, debe descartarlo. Es decir que el reloj 2MSL provee protección contra segmentos retrasados de una instanciación previa de la conexión.

Existe otra situación particular que representa un problema. Podría suceder que un extremo de la conexión en estado **TIME-WAIT**, caiga y luego arranque nuevamente e inmediatamente intente establecer una nueva conexión sobre el mismo par de *sockets*, dentro del período 2MSL. A pesar de tomarse el recaudo de usar otro valor de ISN, podrían existir problemas con segmentos retrasados de la conexión previa a la caída. La protección en este caso, la dicta la propia RFC 793, que establece que TCP debe esperar un tiempo MSL antes de crear nuevas conexiones, luego de un re-arranque o una caída. Este tiempo se suele denominar tiempo de silencio o *quiet time*.

11.7.2 Estado FIN_WAIT_2

En el estado **FIN_WAIT_2** el protocolo TCP se encuentra en la situación de haber transmitido un segmento FIN y haber recibido un reconocimiento ACK por dicho segmento. Para poder salir de este estado debe recibir un segmento FIN, indicador del cierre de la aplicación del otro extremo. Hasta que la aplicación del otro extremo no realice su propio cierre, la situación puede prolongarse indefinidamente en el tiempo.

Para prevenir una espera infinita en el estado **FIN_WAIT_2**, muchas implementaciones asocian un reloj a este estado, cuando el cierre es realizado con la llamada al sistema *close()*. Si el reloj expira y no se recibió el segmento FIN esperado, la máquina de estados TCP se mueve al estado **CLOSED**.

11.8 Problemas en las conexiones TCP

Una vez establecida la conexión, pueden suceder situaciones equívocas o terminaciones anormales de la misma. Muchas de estas situaciones se manejan

con segmentos que presentan el bit RST en alto. Otras, menos evidentes, deben manejarse con ciertos recaudos.

11.8.1 Conexión a un Puerto Inexistente

Si se revisa el Diagrama de Estados de la Fig. 1112, se observa la posible existencia de segmentos RST en el inicio de la conexión. Podría suceder que se intentase iniciar una conexión a un puerto donde no existe un proceso escuchando.

Para probar la respuesta de una implementación en este caso, se puede ingresar, en cualquier navegador, la dirección de un servidor web seguido de ":" y el número de un puerto que se sospeche que probablemente se encuentre cerrado, en lugar del puerto 80. De verificarse esta situación, el usuario recibirá un aviso del tipo "no se puede conectar" o "conexión rechazada". Mediante un *sniffer*, se puede observar el intercambio de segmentos en la red:

***Transmission Control Protocol* (Requerimiento)**
Src Port: 50172 // Dst Port: 25200 // Seq: 61244 // Header length: 32 bytes //
Flags: 0x0002 (SYN) // Window size: 2097152 // Checksum: ---- //
Options: (12 bytes): TCP MSS Option: True; Maximum segment size: 1460 bytes; NOP; TCP Window Scale Option: True; Window scale: 8 (multiply by 256); NOP; NOP; SACK permitted
***Transmission Control Protocol* (Respuesta)**
Src Port: 25200 // Dst Port: 50172 // Seq: 0 // Ack: 61245 // Header length: 20 bytes //
Flags: 0x0014 (RST, ACK) //
Window size: 0 // Checksum: ---

En respuesta a un segmento SYN dirigido a un puerto cerrado, se recibe un segmento con la bandera RST levantada. Los valores de los campos de Número de secuencia y Número de ACK, en el segmento RST, tienen sus particularidades. El Número de Secuencia se ajusta a "0", pero el Número de ACK es un número más que el ISN del otro extremo: "61244 + 1". Por este motivo, el segmento RST debe ser aceptado por el iniciador de la comunicación, puesto que tiene levantada la bandera de ACK y transporta un número de ACK válido. Esta condición es importante en cuanto a la prevención de ataques, para prevenir casos de segmentos espurios con los que se intenta cerrar conexiones legítimas.

Una cuestión no menos importante que el lector debería relacionar es que, si se realizara una prueba de este tipo con una aplicación que se encapsule en UDP, entonces se recibiría un mensaje ICMP de puerto inalcanzable como respuesta.

El manejo del segmento RST es diferente según el estado en el que se encuentre el protocolo TCP que lo recibe. En el caso de un cliente como el del ejemplo, la conexión es abortada y el receptor del RST vuelve al estado CLOSED, avisando a la aplicación de esta situación. En el caso de un servidor, si se

encuentra en el estado SYN-RECEIVED, vuelve al estado LISTEN, transición apreciable en la Fig. 11.12.

11.8.2 Terminación Abortiva

Otra aparición de segmentos RST en la red puede deberse a una situación de liberación abortiva. Se denomina así al cierre de una conexión TCP que no finaliza de manera normal, con el intercambio de segmentos FIN descripto. En una liberación abortiva, la conexión finaliza por alguna acción del usuario o del proceso, que se traduce en el envío de un segmento RST al otro extremo.

En una liberación abortiva, la transmisión de un segmento RST tiene prioridad respecto de otros datos y la aplicación debe ser la que provea la forma de realizarla. Por ejemplo, si el usuario utilizara algún comando de interrupción inmediata de la comunicación en curso, esta acción se traducirá en la transmisión de un segmento TCP con la bandera RST en alto. En este caso, el segmento llevará el Número de Secuencia y el Número de ACK que correspondan al estado de la conexión. El receptor de este segmento, efectivamente abortará la conexión, no enviará un segmento ACK en reconocimiento del segmento RST, y avisará a la aplicación con algún mensaje del tipo *"Conexión reseteada por el par"*.

11.8.3 Asesinato TIME_WAIT

Algunos problemas generados por la llegada de segmentos viejos duplicados se describen en la RFC 1337, junto con mecanismos de prevención para evitarlos.

El único estado, durante el desarrollo de una terminación normal, en el cual la recepción de un segmento RST puede generar problemas, es el estado TIME-WAIT. Si un extremo en este estado recibe un segmento RST, se vería obligado a transitar prematuramente al estado CLOSED. Se conoce este caso como Asesinato de TIME-WAIT (TIME-WAIT Assassination), ya que se cerraría la conexión antes de cumplir el tiempo 2MSL. El problema, en este caso, es que se pueden generar conflictos con nuevas instanciaciones de la conexión ante la llegada de viejos segmentos y falta de sincronismo entre los estados de las máquinas de estado de ambos extremos, por solapamiento de números de secuencia.

La situación no sería tan extraña de suceder si el extremo que ha realizado el open activo, supongamos el lado cliente, se encuentra en el estado TIME_WAIT y recibe un segmento viejo duplicado. Este segmento tendrá un Número de Secuencia anterior a todos los reconocidos y un Número de ACK también menor. El cliente sólo puede atinar a contestar con un segmento ACK duplicado. El problema se genera cuando este segmento llega al servidor: si éste ya se encuentra en el estado CLOSED, no será capaz de reconocerlo y editará un segmento RST. La llegada del segmento RST al extremo cliente produce el asesinato TIME_WAIT, cerrando la conexión de manera prematura.

La mayoría de las implementaciones solucionan este conflicto, sencillamente rechazando cualquier segmento RST que se recibiera en estas circunstancias.

11.8.4 Situación de Conexión Mitad Abierta - Half Open

Se dice que una conexión TCP se encuentra en estado *Half Open* (mitad abierto) cuando uno de los extremos de la misma ha caído, sin que el otro se haya enterado. Un ejemplo de esta situación podría ser una caída abrupta provocada por un corte de luz. En este caso, el extremo caído no podrá realizar ni un cierre ordenado ni uno abortivo, dejando simplemente de transmitir datos. El otro extremo no podrá detectar el problema, conservando recursos para una conexión inexistente. TCP ofrece una forma de detectar este estado a través de un método opcional, conocido como *keepalive* (mantener vivo), que permite chequear si el otro extremo de una conexión establecida se encuentra presente luego de cierto tiempo durante el cual no se han intercambiado datos.

11.9 Servidores TCP - Ataques y Mecanismos de Defensa

Hemos visto que, si la aplicación servidora es del tipo concurrente, cuando recibe la solicitud de una nueva conexión, se desbloquea y crea una nueva instancia de sí misma, por si tuviera que atender otras conexiones. La primitiva *fork()* crea un nuevo socket entre la nueva instancia de aplicación y el protocolo TCP local, provocando que el proceso padre regrese al estado bloqueado, a la espera de nuevas solicitudes.

Los servidores concurrentes son capaces de atender múltiples pedidos de conexión, pero es a nivel del protocolo TCP donde primero se establecen las conexiones, pudiendo derivar en algunas limitaciones. Por ejemplo, si el Sistema Operativo estuviera ocupado y no pudiera ser interrumpido, o la propia aplicación servidora se encontrara creando un nuevo proceso justo cuando llegan pedidos de conexión, es el protocolo TCP el que debe manejar por sí mismo estas situaciones.

Antes de subir los pedidos a la aplicación servidora, la conexión puede estar en uno de dos estados: SYN-RCVD o ESTABLISHED. En el primer caso, todavía no se completó el intercambio de tres vías, en el segundo ya finalizó la fase de inicio. Se trata de conexiones sobre las que el servidor no tiene conocimiento, por lo que el Sistema Operativo debe colocarlas en colas de espera.

Cuando las aplicaciones colocan un *socket* en estado LISTEN, precisan especificar un cola de conexiones pendientes o *backlog* para ese *socket*. Muchas veces se describe el *backlog* como un valor límite para las conexiones entrantes.

El Sistema Operativo puede utilizar una única cola para manejar los estados SYN-RCVD y ESTABLISHED, cuyo tamaño se determina en el argumento *backlog* de la llamada al sistema *listen*(). Cuando se recibe un segmento SYN, se contesta con un segmento SYN+ACK y se agrega la conexión

a la cola. En el momento en que se reciba el segmento ACK proveniente del otro extremo, la máquina evoluciona al estado ESTABLISHED y la conexión se coloca en estado de ser entregada a la aplicación. Es decir que, en esta única cola, existirán conexiones en estado SYN- RCVD y en estado ESTABLISHED, pero sólo estas últimas podrán ser retornadas a la aplicación con la llamada al sistema *accept*().

Otra opción es que el Sistema Operativo utilice dos colas, una para SYN-RCVD y otra para ESTABLISHED. Puede suceder que conexiones que forman parte de la primera cola, pasen a la segunda cuando terminen de completarse. En este caso, la llamada al sistema *accept*() se implementa como una forma de consumir conexiones de la cola de ESTABLISHED y el argumento *backlog* de la llamada *listen*() determina el tamaño de esta cola.

Las implementaciones TCP derivadas de BSD usan la primera opción. De esta forma, cuando se llena la cola limitada por el valor del *backlog*, TCP no transmitirá más segmentos SYN+ACK en respuesta a segmentos SYN entrantes. Generalmente, a partir de este punto, TCP descartará los segmentos SYN, en vez de responder con un segmento RST. De esta forma logrará que el cliente vuelva a intentar iniciar la conexión, en vez de asumir que el servicio no está disponible.

En Sistemas Operativos tipo Linux, a partir de la versión 2.2, la llamada al sistema *listen*() especifica la longitud de la cola para conexiones ya establecidas que están a la espera de ser aceptadas por la aplicación. Esto se corresponde con la segunda de las opciones mencionadas: dos colas, una para SIN-RCVD, cuya longitud es especificada por el sistema y almacenada en la variable *net.ipv4.tcp_max_syn_backlog* (valor 1000 por default), y otra de conexiones aceptadas, cuya longitud especifica la aplicación y se almacena en la variable *net.core.somaxconn* (valor 128 por default).

Por lo tanto, si una implementación TCP en Linux recibe el segmento ACK que finaliza el protocolo de tres vías y la cola de conexiones aceptadas está llena, debe tomar una decisión práctica en cuanto a la aceptación o el rechazo.

Existe un reloj asociado con el estado SYN-RCVD, de tal manera que, si no se recibe el segmento ACK, se debe re-enviar el segmento SYN+ACK. En el caso en que la cola se encuentre llena, el protocolo TCP del lado servidor procede ignorando el segmento ACK, como si no lo hubiera recibido. Por su parte, el lado cliente recibirá múltiples segmentos SYN+ACK, asumiéndose los respectivos segmentos ACK como perdidos, debiendo entonces re-enviarlos. Si, del lado servidor, se reduce la cola de *backlog* antes de alcanzar el máximo de retransmisiones de SYN+ACK, eventualmente se procesará uno de los ACK duplicados, el estado pasará a ESTABLISHED y se agregará la conexión a la cola de conexiones aceptadas.

Esta forma de manejar las conexiones tiene sus consecuencias desde el punto de vista de la seguridad de un sistema.

Un ataque por inundación de segmentos SYN (*SYN flood*) es un ataque de Denegación de Servicio (DoS) en el que uno o más clientes maliciosos generan una serie de intentos de conexión sobre un servidor, transmitiendo segmentos SYN, que colocan al extremo servidor en estado SYN-RCVD, pero nunca terminan con el inicio de la conexión. Muchos de estos ataques se combinan con falseamiento de direcciones IP de los clientes, método conocido como *IP*

spoofing. Ante este tipo de ataques, el servidor asigna recursos a cada conexión no completada y, eventualmente, puede quedar sin memoria debido a la cantidad de conexiones *half open* soportadas, denegando el servicio a usuarios legítimos. El ataque se explica en la RFC 4987, junto con algunos mecanismos de defensa.

La inundación de segmentos SYN tiene como objetivo vaciar el *backlog* de conexiones *half open* asociadas a cierto número de puerto. De este modo, los usuarios legítimos serán rechazados. Los parámetros del ataque son el tamaño, la frecuencia y los medios disponibles para seleccionar direcciones IP falsas. El tamaño debe alcanzar los valores de *backlog* para que el ataque sea efectivo. En máquinas servidoras para grandes volúmenes de tráfico, las variables de *backlog* se configuran en valores altos.

Por otro lado, aunque el protocolo no lo exige, ciertos Sistemas Operativos configuran tiempos límite para conexiones *half open*, luego del envío del segmento SYN con ACK, de tal manera de cortar las retransmisiones SYN ACK y liberar el TCB si se excede el límite. En algunas implementaciones, se establece un tiempo de $75\ seg$, en otras se aplica un mecanismo de *back off* y se libera luego de $511\ seg$. Estos valores marcan la frecuencia del ataque, de tal manera que se deben generar nuevos segmentos SYN falsos ni bien se liberen los TCB de los previos.

Una solución bastante obvia sería aumentar el tamaño del *backlog*, pero se ha demostrado que las implementaciones presentan problemas en cuanto a las estructuras de datos que utilizan y los algoritmos de búsqueda que manejan, no pudiendo escalar correctamente ante tamaños que superan unos pocos cientos de conexiones.

Otro mecanismo rápido de defensa consiste en acortar el período de *timeout* entre SYN recibidos y la liberación del bloque TCB. Esta elección puede frenar el ataque, liberando espacio para conexiones legítimas, pero también podría evitar que algunas conexiones legítimas se pudiesen establecer. Además, si el atacante aumentara la frecuencia para contrarrestar la disminución del *timeout*, esta táctica resultaría inefectiva. Por este motivo, algunas implementaciones la utilizan sólo cuando la ocupación del *backlog* o o la velocidad de recepción de segmentos SYN superan algún umbral.

La RFC 4987 presenta un mecanismo de defensa frente a un ataque de Denegación de Servicio, denominado *SYN cookies*, que permite codificar dentro del número de secuencia del segmento que combina SYN y ACK, la mayoría de la información que debería almacenarse para una conexión ante el arribo de un segmento SYN.

Usando este método, el servidor no precisa asignar memoria para los pedidos de conexión entrantes hasta que no se cumpla el protocolo de tres vías, o sea cuando llega el último ACK. Entonces, se recuperan los parámetros necesarios y la conexión pasa al estado ESTABLISHED.

Producir *SYN cookies* implica una elección cuidadosa del ISN del lado servidor, ya que se debe codificar cualquier información esencial en el campo de número de secuencia del segmento SYN + ACK, que luego volverá como número de ACK desde el cliente legítimo.

Linux adopta una técnica que construye el valor ISN de $32\ bits$ en un número dividido en tres partes, cada una con su propio significado. Los primeros

5 $bits$ representan el valor de un contador de 32 $bits$ que se incrementa en 1 cada 64 seg, expresado en $módulo$ 32. Los 3 $bits$ siguientes codifican el valor de MSS de lado servidor. Los 24 $bits$ restantes son un *hash*, seleccionado por el servidor, de los 4 valores del par de socket, junto con el número del primer campo. Cuando el servidor recibe de vuelta este número en el ACK, verifica el *hash* y puede reconstruir sus propias colas de SYN-RCVD.

El método tiene algunos inconvenientes, por incompatibilidades con algunas opciones TCP. Por ejemplo, codificar el valor MSS implica una representación binaria de sólo ocho valores pre-definidos. Algo similar sería necesario para algunas otras opciones, presentándose problemas para poder negociarlas en el inicio, ya que no almacenan el primer segmento SYN del cliente. Además, el contador representa un tiempo límite de 64 seg, transcurridos los cuales, el arrancaría nuevamente. Por estos motivos, la funcionalidad no está habilitada por default, sino que sólo se habilita bajo ciertas condiciones de ataque.

Bibliografía

1. RFC 793 “Transmission Control Protocol ”, September 1981. http://tools.ietf.org/html/rfc793
2. RFC 1122 “Requirements for Internet Hosts -- Communication Layers”, October 1989. http://tools.ietf.org/html/rfc1122
3. RFC 1323 “TCP Extensions for High Performance”, May 1992. https://www.ietf.org/rfc/rfc1323.txt
4. RFC 1337 “TIME-WAIT Assassination Hazards in TCP”, May 1992. http://tools.ietf.org/html/rfc1337
5. RFC 2018 “TCP Selective Acknowledgment Options”, October 1996. http://tools.ietf.org/html/rfc2018
6. RFC 4987 “TCP SYN Flooding Attacks and Common Mitigations”, August 2007. http://tools.ietf.org/html/rfc4987
7. RFC 5681 “TCP Congestion Control”, September 2009. http://tools.ietf.org/html/rfc5681
8. RFC 6056 “Recommendations for Transport-Protocol Port Randomization ”, January 2011. http://tools.ietf.org/html/rfc6056
9. RFC 6691 “TCP Options and Maximum Segment Size (MSS)”, July 2012. http://tools.ietf.org/html/rfc6691
10. Service Name and Transport Protocol Port Number Registry http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml, Last Updated 2014-08-15.
11. Kozierok, Charles M., “The TCP/IP Guide”. http://www.tcpipguide.com/free/t_toc.htm
12. Comer, Douglas, “Internetworking with TCP/IP: Principles, Protocols and Architecture v. 1”. Pearson Education, 1995.

13. Stevens, W. Richard, "TCP/IP Illustrated, Vol. 1: The Protocols (Addison-Wesley Professional Computing Series) ". Addison-Wesley, 1993.
14. IBM Knowledge Center, "How sockets work", http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_71/rzab6/howdosockets.htm
15. Berbstein, D.J., "SYN Cookies", http://cr.yp.to/syncookies.html

Problemas

1. ¿Cuáles son las características de un modelo Cliente-Servidor? ¿Cómo se determina cuál es el cliente y cuál es el servidor en una comunicación de transporte? ¿Cómo se asignan los puertos a la comunicación? ¿Qué diferencias existen en la construcción de sockets entre ambos extremos?

2. Explique la forma en que el protocolo TCP realiza control de flujo. ¿Por qué deben existir dos ventanas en cada extremo? ¿A qué unidad de datos se aplica un ACK? ¿Por qué pueden existir ACK duplicados?

3. ¿Qué diferencia hay entre una conexión *half open* y otra *half close*?

4. ¿Qué sucede con una conexión en el estado TIME_WAIT que recibe un duplicado del FIN que la colocó en ese estado? ¿Y si recibe un RST?

5. Cuando se pierde un segmento ACK, no necesariamente se genera una retransmisión. ¿Por qué?

6. Con un *sniffer*, en un intercambio cliente-servidor, se capturó el siguiente segmento TCP:

   IP 181.30.240.109: 80 > 192.168.1.105:51543 : SYN 2545627512:2545627512(0) ack 2933496280 win 65535 <mss 1460,nop, nop, wscale 6, nop,sack OK>.

   En el flujo de la conexión que se observa a continuación, los segmentos de datos enviados de cliente a servidor fueron de $640\ bytes$, en tanto que, de Servidor a Cliente, los segmentos eran de $1460\ bytes$.

   a) Describa el par de *sockets* que define la conexión.

   b) ¿A cuál segmento del protocolo de tres vías del inicio se corresponde este segmento?

   c) Complete el inicio de la conexión, con los Números de Secuencia Iniciales de cada lado, sabiendo que el cliente se encuentra en una red LAN y acepta las mismas opciones que el Servidor, pero su factor de escalamiento de ventana es la mitad que el de éste.

   d) Si que el cliente envía al servidor un único segmento de datos y el servidor transfiere al cliente 3 segmentos de datos, complete la transferencia de datos y el ciclo de cierre de la conexión con los números de secuencia que correspondan.

CAPÍTULO XII

Transferencia de datos TCP

En el capítulo previo se han presentado los aspectos más significativos del protocolo TCP relacionados con el inicio y fin de una conexión. Muchos de ellos se plasman en los campos del encabezado, pero otros comportamientos surgen de las transiciones explicadas a partir del Diagrama de Estados.

En este capítulo se presentará el protocolo en acción, una vez que la conexión se ha establecido. Se desarrollará en detalle la estrategia de expiración (timeout) y retransmisión y el ajuste de los parámetros que la manejan.

Se explicará cómo afronta TCP situaciones de pérdida de segmentos, aparición de segmentos duplicados y casos de desorden en los segmentos recibidos, debiendo asegurar, en cualquier caso, una comunicación confiable.

El capítulo abordará la dinámica que el protocolo imprime sobre la transferencia de datos, distinguiendo entre los casos de aplicaciones interactivas, donde la transmisión es de pocos bytes por vez, y la transferencia de grandes volúmenes de datos.

12.1 Estrategia de expiración y retransmisión

Cada segmento enviado por TCP, lleva asociado un reloj de descuento o expiración (*timeout*) que, si se agota sin que se reciba un segmento cuyo Número de ACK cubra los datos transportados en el segmento transmitido, dispara la retransmisión. El tiempo se asocia a una variable conocida como Expiración de Retransmisión (RTO, Retransmission Timeout). La estrategia significa que el reloj debe ajustarse de manera de dar tiempo a bajar el segmento a la red, que ésta lo transporte encapsulado en IP hacia destino, que el destino lo levante de la red y verifique sus errores y, de ser correcto, que edite un segmento ACK para que viaje de regreso hacia la fuente. Todo este tiempo se conoce como Tiempo de Viaje de Ida y Vuelta (RTT, Round Trip Time).

En el capítulo previo, se ha mencionado que existe un tiempo de expiración asociado al inicio de una conexión. Su naturaleza es del tipo retroceso exponencial, manejado por dos variables: la cantidad de reintentos y el tiempo

total antes de proceder a un aviso de imposibilidad de comunicación. Estos valores podrían fijarse de antemano, en términos generales, suficientes para cubrir cualquier tipo de conexión.

Lo interesante, durante la fase de transmisión de datos para una conexión particular, y aún dentro de una misma conexión, es cómo fijar el valor del RTO, ya que el propio RTT puede llegar a variar por diversos motivos, muchos de los cuales se encuentran asociados directamente a la situación de carga de tráfico en la red.

La estrategia más inteligente en todos los casos sería fijar el valor del RTO en base a una medición del RTT. Esto es lógico cuando se piensa que, si se fijara el valor del RTO en un tiempo menor que el RTT promedio para la conexión particular, probablemente se producirían retransmisiones innecesarias. De la misma manera, si se fija un valor superior, se podría perder capacidad de reacción, trabajándose de manera ineficiente en el caso de una red congestionada. Por este motivo, el valor fijado para el RTO debe seguir las propias fluctuaciones del RTT.

Para cada conexión, el protocolo TCP toma muestras del valor del RTT, midiendo el tiempo que transcurre entre la transmisión de un segmento de datos con un Número de Secuencia dado y la recepción de un segmento de ACK, cuyo Número de ACK lo reconozca como recibido correctamente. Dentro de una misma conexión, se pueden obtener muchas muestras que pueden servir para estimar las variaciones del RTT, para el ajuste apropiado del RTO.

La estimación debe ser lo más cercana posible a la realidad para no generar retransmisiones disparadas prematuramente, innecesarias, o tardías, por falta de capacidad de reacción a las variaciones.

12.1.1 Cálculo del Estimador Suavizado en el Método Clásico

Originalmente, en la RFC 793, se ofrecía un ejemplo de cálculo de un Estimador Suavizado del RTT (SRTT, Smoothed RTT), según la siguiente relación:

$$SRTT = \alpha \; x \; SRTT + (1 - \alpha) x \; RTT \tag{12.1}$$

En la Ec. (12.1), RTT es la medición actual del tiempo de ida y vuelta y $SRTT$ del lado derecho de la igualdad es la estimación previa. Con ambos valores, pesados de manera apropiada, se obtiene el nuevo valor de $SRTT$. El factor de suavizado α determinará el peso que la última medición, o el valor previo estimado, tendrán sobre la nueva estimación. Un valor común recomendado para α era 0.8 ó 0.9, obteniendo el 80% ó 90% de la nueva estimación, a partir de la estimación previa, y sólo el 20% ó 10% a partir de la medición actual. Una vez calculado el estimador, la RFC 793 establecía el valor de RTO como:

$$RTO = \min(ubound, \max(lbound, (SRTT)x\beta)) \tag{12.2}$$

En esta expresión, $ubound$ se refiere a un límite superior recomendado en 1 min y $lbound$ a un límite inferior recomendado en 1 seg. El parámetro β se define como la varianza del retardo, cuyo valor recomendado se encontraba en el rango 1.3 a 2. En términos prácticos, el valor de RTO, terminaba fijado en 1 seg o el doble del estimador. Esta medición funcionaba correctamente en conexiones estables, sin grandes variaciones de RTO.

12.1.2 Cálculo del Estimador Estándar de Jacobson

En 1988, Van Jacobson publicó un artículo en el que observaba que la estimación clásica resultaba apropiada para cierto nivel de carga, a partir del cual la conexión respondería al incremento con retransmisiones de paquetes que sólo habrían sido retrasados en tránsito. En esas situaciones de congestión, los paquetes duplicados cargaban aún más la red. Por este motivo, propuso un nuevo método de estimación, en el que calculaba el error entre la estimación y la medición, luego el valor medio de la estimación y, por último, el valor medio de la desviación. En su explicación, Van Jacobson comenzó reordenando la ecuación original, definiendo $(1 - \alpha) = g$, de tal manera que la expresión (12.1) se puede volver a escribir como:

$$SRTT = SRTT + g(RTT - SRTT) \tag{12.3}$$

En esta ecuación, el error es la diferencia $(RTT - SRTT)$, por lo que la nueva predicción de la estimación queda determinada por la predicción previa y una fracción del error. Van Jacobson consideró el error compuesto por dos componentes, un componente aleatorio, propio de las fluctuaciones de tráfico E_r, y otro debido a una mala elección del estimador E_e. De este modo la Ec. (12.3) anterior se convierte en:

$$SRTT = SRTT + gE_r + g\,E_e \tag{12.4}$$

Una de las cuestiones principales es la elección del factor g. Según Jacobson, el término aleatorio E_e mueve el valor de $SRTT$ hacia el valor medio real, sin importar el valor de g usado, pero un valor pequeño de g mimimiza la influencia de esta componente aleatoria. Por este motivo, aconsejó un valor en el rango 0.1 a 0.2 .

El valor estimado $SRTT$ oscilará aleatoriamente alrededor del promedio verdadero y la desviación estándar será la del valor medido RTT , convergiendo con una contante de tiempo $1/g$. Para medir la desviación estándar de RTT,

Jacobson desechó la varianza, puesto que eso implicaba el cálculo de un cuadrado, $(RTT - SRTT)^2$, y luego de una raíz, complicando la implementación. Por este motivo, en lugar de la desviación estándar, propuso calcular la desviación media, por provenir de un cálculo más conservador y mantener una relación mayor, pero cercana a uno, con la desviación estándar, siempre que los errores de predicción tengan una distribución normal. Basado en estas premisas, propuso el siguiente cálculo:

$$\begin{gathered} Err = (M - A) \\ A \leftarrow A + g\,Err \\ D \leftarrow D + h(|Err| - D) \\ RTO = A + 4D \end{gathered}$$

(12.5)

En este conjunto de ecuaciones, M es el valor medido, A es el estimado y D la desviación media. Para que el cálculo sea rápido debería ser realizado en aritmética de números enteros, pero como $g < 1$ es un número no entero, impuso un escalamiento para las expresiones previas. Jacobson propuso $g = {}^1/_8$ y $h = {}^1/_4$, para que fuera más sencillo escalar mediante corrimientos binarios, y porque la mayor ganancia para la desviación permite al RTO seguir mejor las variaciones del RTT.

La RFC 6298 recomienda ajustar los valores iniciales para RTO en $1\,seg$, cuando la conexión recién comienza y no se han podido realizar mediciones. Este valor se eleva a 3 seg en el caso que el primer segmento SYN no logre llegar al otro extremo o se pierda el ACK que lo cubre. Una vez conseguida una primera medición del RTT, se recomienda inicializar los estimadores a $(A \leftarrow M)$ y $(D \leftarrow {}^M/_2)$.

El método de Jacobson es la base sobre la que muchas implementaciones TCP calculan el RTO hoy en día, logrando adaptarse de manera apropiada a los vaivenes de la conexión.

12.1.3 Problema de la medición en las retransmisiones - Algoritmo de Karn

La medición del RTT presenta un problema en el caso de dispararse una retransmisión. Cuando se transmite un segmento, si no se recibe el ACK correspondiente, dentro del RTO establecido, se procede a su retransmisión. Si posteriormente llegase un ACK para dicha retransmisión, no sería posible determinar con certeza si se corresponde a la transmisión o a la retransmisión. Surge entonces la pregunta sobre a cuál de ambos se asocia la medición.

Este problema, que se conoce como ambigüedad del ACK, fue estudiado por Phil Karn, quien determinó que no era preciso asociar una medición a esta situación para luego calcular el estimador. La RFC 6298 también estableció que

no se deben tomar muestras del RTT sobre segmentos retransmitidos, excepto que los segmentos carguen la nueva opción de sello de tiempo, con la que se puede remover la ambigüedad.

De todas maneras, el hecho de haber tenido que retransmitir, permite obtener información sobre el estado de la red, ya que TCP interpreta una situación de *timeout* como un indicador de congestión. No tendría sentido entonces seguir usando el mismo valor de RTO, ya que probablemente disparará nuevas retransmisiones. Lo más sensato sería aumentar el valor de RTO para reducir la carga sobre la red.

A instancias de lo observado por Karn, TCP incorpora para estos casos un factor de *backoff* para el RTO, que se duplica con cada nueva retransmisión. Esta situación se mantiene mientras no se pueda realizar una medición correcta, sobre un segmento no retransmitido. Entonces, el valor del factor de *backoff* vuelve a 1 y el valor de RTO se calcula a partir de los estimadores.

12.1.4 Opción Sello de Tiempo para medición de RTT

La RFC 1323 define una opción TCP, conocida como Sello de Tiempo, que puede usarse para lograr una medición más precisa del RTT. En caso de usar esta opción, para tomar una muestra del valor de RTT se agrega al encabezado un número de 32 $bits$, que luego es copiado por el otro extremo en el segmento ACK. Se puede medir el RTT registrando el tiempo de transmisión del segmento y la llegada del ACK con el valor de Sello de Tiempo correspondiente, y restando ambos valores entre sí. Aunque se pueden lograr mediciones más precisas con este mecanismo, es necesario acompañarlo de reglas para diferentes situaciones de medición, tal como se explicará en el siguiente Capítulo.

12.1.5 Algoritmo de Manejo del RTO

La RFC 6298 indica un algoritmo recomendado para el manejo del reloj de retransmisión:

1. Cada vez que se envía un segmento con datos, aunque se trate de una retransmisión, hay que encender el reloj que expirará luego de $RTO\ seg$ (para el valor actual del RTO).
2. Cuando todos los datos pendientes se hayan reconocido, apagar el reloj.
3. Cuando se reciba un ACK para datos nuevos, re-arrancar el reloj para que expire luego de $RTO\ seg$.
   Cuando el reloj expire, se debe:
4. Retransmitir el segmento más viejo que no haya sido reconocido.
5. Ajustar el valor $RTO = RTOx2$ (reloj de *backoff*). El valor máximo de RTO debe ser, por lo menos, de $60seg$.
6. Arrancar el reloj de retransmisión para que expire luego de $RTO\ seg$.

7. Si el reloj expira mientras espera la llegada de un ACK para un segmento SYN y la implementación está usando un valor de $RTO < 3\ seg$, se debe re-inicializar el RTO a 3 seg cuando comience la transferencia de datos.

Luego de una retransmisión, cuando se hayan enviado nuevos datos y recibido un ACK para ellos, se puede obtener una nueva medición para RTT, y un cálculo para RTO, que puede provocar una disminución en el valor fijado por el *backoff*.

12.1.6 Retransmisiones y re-empaquetado

Una cuestión a tener en cuenta es que, al momento de que se produce una expiración del RTO, se dispara una retransmisión, aunque no necesariamente del mismo segmento. TCP puede re-empaquetar datos, transmitiendo un segmento de mayor tamaño por cuestiones de eficiencia. Esta funcionalidad es posible por la forma de numerar los datos que adopta el protocolo: por byte y no por segmento.

12.2 Aplicaciones interactivas

Una aplicación interactiva que se apoya en TCP precisa que mensajes de poco volumen de datos se transmitan entre cliente y servidor: un carácter de teclado, un movimiento del mouse o de un joystick. Estos pequeños mensajes provocan acciones del lado servidor. Se trata de paquetes denominados t*ynigrams*, por diminutos o *tiny*, ya que son de muy pocos bytes, razón por la que su transmisión resulta ineficiente, debido al gran gasto u *overhead* que representa el encabezado frente a la carga de datos. Por otra parte, si se eligiera transmitir varios de estos paquetes en uno solo, se podrían generar situaciones molestas para el usuario, debido a que las aplicaciones interactivas son sensibles al retardo.

Se suele distinguir este tipo de transporte respecto del tráfico de grandes volúmenes de datos. La mayoría del tráfico TCP es de este último tipo pero, a pesar de ello, el protocolo mantiene reglas especiales para el transporte de tráfico interactivo.

Un ejemplo de viejas aplicaciones que generaban este tipo de tráfico son *rlogin* y *telnet*. Ambas permitían el *login* remoto, aunque padecían de muchos inconvenientes en cuanto a seguridad. Una aplicación más nueva, denominada *SSH*, por *Secure Shell*, trabaja de forma similar a *telnet*, pero usa técnicas de cifrado, logrando que el tráfico sea mucho más seguro frente a la instalación de *sniffers*.

Un flujo de datos típico de este tipo de aplicaciones interactivas se presenta en la Fig. 12.1.

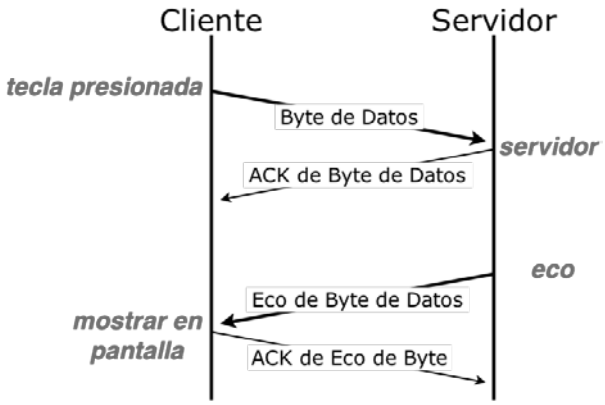


Figura 12.1 – Flujo de Datos Interactivo

En este caso, cada tecla presionada por el usuario genera un segmento que viaja por la red hasta el servidor. Éste debe hacer eco de lo que el cliente escribe, para que el usuario lo pueda apreciar en su pantalla. Con esta forma de operación, la transmisión de un único carácter, pueden generar hasta cuatro segmentos: el de datos y su correspondiente ACK, y el de eco y su propio ACK. En la Fig. 12.1 se puede observar que, luego de la llegada del primer dato al lado servidor, se retrasa el segmento de ACK que lo reconoce, probablemente en espera de poder cargar el eco, para hacer la transmisión más eficiente.

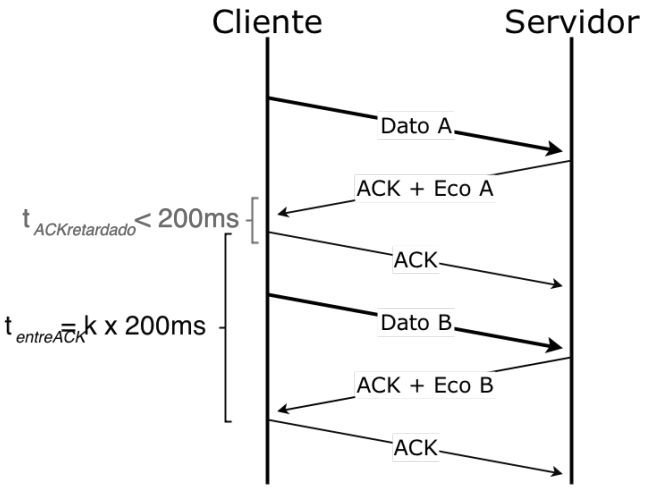


Figura 12.2 – Flujo de Datos Interactivo

Normalmente, el lado servidor es lo suficientemente rápido como para combinar ambos segmentos, tal como se observa en la Fig. 12.2. Esta técnica de combinación de ACK con datos se conoce con el nombre de *piggybacking*, que

significa montado a caballito. Del lado cliente, sin embargo, no se puede juntar el ACK con el siguiente dato, pues esto depende de la rapidez con que el usuario maneje el teclado.

La técnica de **ACK retardado** es una técnica muy usada por el protocolo TCP. Lo cierto es que no se puede retrasar el envío del segmento ACK sin que se ponga en juego cierta probabilidad de retransmisión. Por este motivo, la RFC 1122 establecía un máximo de $500\,mseg$ al tiempo de retraso, pero muchas implementaciones lo ajustan en $200\,mseg$.

Hemos mencionado ya que TCP utiliza una técnica conocida como **ACK acumulativo**, con la cual no es preciso generar un ACK por cada segmento recibido. Siguiendo este criterio, el protocolo puede demorar la generación de un ACK, a la espera de poder combinarlo con el envío de datos. La Fig. 12.2 presenta el caso en que la implementación posee un límite de $200\,mseg$, resaltando el hecho de que las transmisiones de segmentos ACK se disparan a partir de un reloj interno del mismo período.

12.2.1 Algoritmo de Nagle

Para poder manejar de manera eficiente situaciones de paquetes pequeños inyectados en la red, TCP incluye el Algoritmo de Nagle, que se basa en la existencia de ACK retardados y se encuentra descripto en la RFC 896.

El algoritmo establece que, cuando una conexión tiene datos en el buffer de transmisión a la espera de ACK, no puede transmitir nuevos datos bajados por el usuario en forma de segmentos pequeños, hasta que los datos no reciban el correspondiente ACK. De este modo, no sólo se busca juntar cierta cantidad de datos en un segmento, transmitiéndolo cuando se reciba el ACK, sino además se agrega características de auto-sincronismo a la propia conexión, ya que cuanto más rápido se reciban segmentos ACK, más pronto se transmitirán los datos.

Un detalle importante es que la aplicación del algoritmo es diferente según el entorno.

Por ejemplo, si la conexión se ha establecido entre dos máquinas en una misma red LAN, cuyo retardo ida y vuelta es mucho más pequeño que $1\,mseg$, se precisaría que el usuario fuera capaz de manejar el teclado a una velocidad superior a $1000\,caracteres/seg$ para poder observar el algoritmo trabajando. En este entorno, el Algortimo de Nagle no se dispara, puesto que el segmento de ACK siempre llega antes que el siguiente carácter en el buffer. En un entorno WAN, sin embargo, se modifican bastante los tiempos, pudiendo llegarse a retardos en el orden de los segundos. En este caso, un usuario generando caracteres a razón de $1\,carácter/seg$, aumenta la probabilidad de que el Algoritmo de Nagle se dispare, colectando datos antes de que llegue un ACK. De esta manera, se reducirá el número de paquetes pequeños en la red aunque la conexión se verá afectada, probablemente aumentado la duración total de la misma, debido a la característica de auto-sincronismo impuesta.

Generalmente, el Algoritmo de Nagle se encuentra habilitado por default, pero hay ocasiones, en entornos altamente interactivos, donde los efectos de

performance que genera pueden ser indeseables con respecto a la latencia. Por este motivo, se lo puede deshabilitar sobre una base por conexión, a través de una opción de *socket()*.

12.3 Tratamiento de datos - Bandera PSH

Mientras que las aplicaciones manejan la velocidad y el momento con que envían datos a TCP, no pueden controlar ni la velocidad ni el tiempo con que TCP los envía a la red. En el caso de transmisiones de grandes archivos esto no sería un problema mientras TCP acumulara los datos en buffer y los fuera transmitiendo a medida que la conexión lo permitiese.

En el caso de una aplicación interactiva, no se desea que TCP acumule los datos, sino que los transmita lo más rápidamente posible, para que no se perciban demoras en el proceso interactivo. Para manejar esas situaciones se incluyó la bandera de PUSH en el encabezado TCP. Cuando se invoca esta funcionalidad, TCP creará un segmento o varios que contengan los datos en espera y los transmitirá con la bandera de PUSH en alto. Los límites de los mensajes dependen de las aplicaciones.

En la práctica se suele encender esta bandera cuando se vacía el buffer de transmisión, toda vez que se transmite un segmento.

12.4 Tratamiento de datos - Bandera URG

El protocolo TCP permite que las aplicaciones marquen datos que precisan tratamiento urgente. Cuando se envían datos de este tenor para ser transmitidos, levantan la bandera URG del encabezado, indicando que el campo Puntero URG es válido.

El mecanismo urgente de TCP permite marcar un punto en el flujo de bytes como el final de la información que precisa tratamiento urgente. Siempre que dicho número sea mayor que el extremo izquierdo de la ventana de recepción (RCV.NXT), el protocolo receptor debe avisar a la aplicación para que ésta entre en modo urgente. Cuando el borde izquierdo de la ventana de recepción haya alcanzado el número de byte que marca el puntero urgente, TCP debe avisar a la aplicación que puede volver al modo normal de operación. Esto significa que ciertos datos que fueron recibidos como normales, podrían convertirse en datos urgentes si se recibe una indicación URG en un segmento posterior, antes que dichos datos hayan sido subidos a la aplicación.

El último byte de datos urgentes se encuentra sumando el campo Número de Secuencia con el valor del campo Puntero Urgente, cargados en el segmento con la bandera URG levantada.

Este mecanismo presentó varias ambigüedades desde el momento de su definición. No se trata de un mecanismo de entrega de datos fuera de banda, pero muchas implementaciones procesan los datos urgentes de ese modo, no entregándolos como parte del flujo de datos normal, como lo especifica la RFC

original, sino separando el último byte de datos urgente. También, durante muchos años existió una ambigüedad referida a si el puntero urgente apuntaba al primero o al último byte de datos urgentes, hasta que la RFC 1011 aclaró que se trata del último byte.

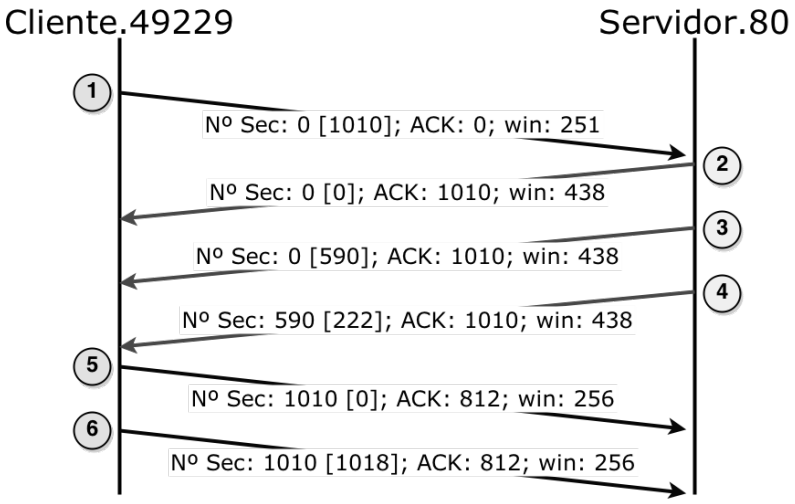
Como consecuencia de estas cuestiones, la RFC 6093 aconsejó que las nuevas aplicaciones no usaran el mecanismo urgente de TCP, aunque estableció que todas las implementaciones TCP lo deben seguir incorporándolo para que pueda ser utilizado por aquellas aplicaciones ya existentes que lo invocan.

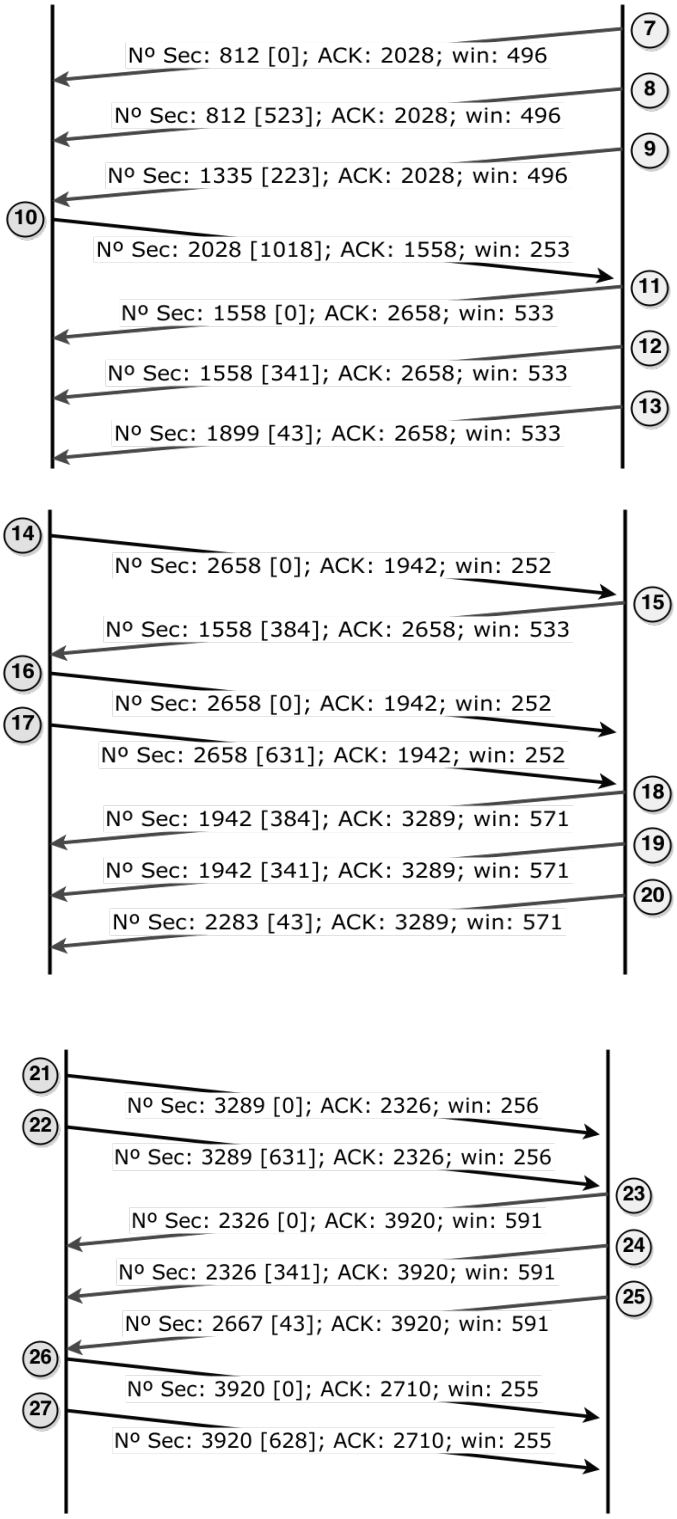
12.5 Flujo de grandes volúmenes de datos

En contraposición con el tráfico de aplicaciones interactivas, el 90% de las transmisiones TCP se refiere a intercambio de grandes volúmenes de datos. A continuación, se presentan varios ejemplos de transferencias de datos entre un cliente y un servidor web. En algunos casos, se ha omitido el inicio de la conexión y se ha numerado los segmentos con un número de secuencia relativo, a partir del número "0", reservado para el primer byte de datos de cada extremo. Los números entre corchetes "[ ]" representan la carga de datos de cada segmento. Resaltado junto al instante de transmisión de cada segmento aparece un número indicador del orden del mismo en el flujo de datos TCP.

12.5.1 Intercambio de segmentos – Situación de timeout

En la Fig. 12.3, se presenta parte de una transferencia de datos entre un cliente y un servidor web. La intención de este ejemplo es ver al protocolo en acción en una transferencia de segmentos con gran cantidad de datos y situaciones de *timeout*. Se aconseja al lector seguir el flujo de Números de Secuencia y Números de ACK para entender mejor cómo trabaja TCP.




7
Nº Sec: 812 [0]; ACK: 2028; win: 496
8
Nº Sec: 812 [523]; ACK: 2028; win: 496
9
Nº Sec: 1335 [223]; ACK: 2028; win: 496
10
Nº Sec: 2028 [1018]; ACK: 1558; win: 253
11
Nº Sec: 1558 [0]; ACK: 2658; win: 533
12
Nº Sec: 1558 [341]; ACK: 2658; win: 533
13
Nº Sec: 1899 [43]; ACK: 2658; win: 533
14
Nº Sec: 2658 [0]; ACK: 1942; win: 252
15
Nº Sec: 1558 [384]; ACK: 2658; win: 533
16
Nº Sec: 2658 [0]; ACK: 1942; win: 252
17
Nº Sec: 2658 [631]; ACK: 1942; win: 252
18
Nº Sec: 1942 [384]; ACK: 3289; win: 571
19
Nº Sec: 1942 [341]; ACK: 3289; win: 571
20
Nº Sec: 2283 [43]; ACK: 3289; win: 571
21
Nº Sec: 3289 [0]; ACK: 2326; win: 256
22
Nº Sec: 3289 [631]; ACK: 2326; win: 256
23
Nº Sec: 2326 [0]; ACK: 3920; win: 591
24
Nº Sec: 2326 [341]; ACK: 3920; win: 591
25
Nº Sec: 2667 [43]; ACK: 3920; win: 591
26
Nº Sec: 3920 [0]; ACK: 2710; win: 255
27
Nº Sec: 3920 [628]; ACK: 2710; win: 255

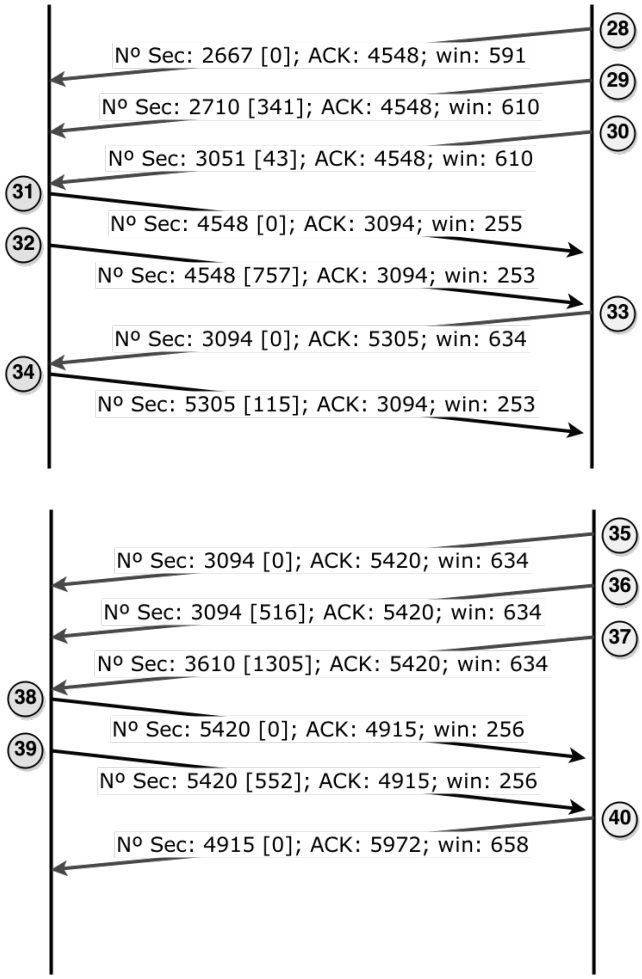


Figura 12.3 – Flujo de grandes volúmenes de datos

El segmento 1 es un pedido GET del cliente HTTP, que carga 1010 *bytes* de datos y anuncia un tamaño de ventana de 251 *bytes*. Es dable aclarar que ste valor de ventana, debería escalarse, puesto que ambos extremos negociaron en el inicio la utilización de la opción Escalamiento de Ventana, que acá se omite. Por lo tanto, lo mismo debería interpretarse a partir de los valores de ventana anunciados por el servidor.

En el segmento 2, el servidor reconoce todos los datos provenientes del cliente, mediante un ACK retardado, o sea un segmento sin datos, únicamente transmitido para reconocimiento de los datos recibidos. Es preciso hacer notar que, un segmento ACK sin datos, lleva el número de secuencia que el otro extremo está esperando recibir, que también será el número de secuencia del siguiente segmento de datos que el cliente transmita.

El servidor comienza a transmitir datos en los segmentos 3 y 4. Primero transmite 590 *bytes* y luego 222 *bytes*, que son reconocidos en conjunto con

un ACK retardado proveniente del cliente, cuyo número de secuencia es 1010. El número de ACK es la suma $590 + 222 = 812$.

En el segmento 6, el cliente realiza un nuevo pedido, de 1018 *bytes* de datos. Su número de secuencia es 1010, porque esa fue la cantidad de datos transmitida en el primer segmento, cuyo número de secuencia se fijó en "0", para numeración relativa. El servidor reconoce la llegada correcta de estos nuevos datos con el ACK retardado del segmento 7. Los segmentos 8 y 9 son la respuesta al requerimiento, transmitida en dos paquetes de 523 y 223 *bytes* de datos, respectivamente. Obsérvese en ambos extremos las leves variaciones de la ventana anunciada, sin olvidar que está escalada.

En el segmento 10 se puede observar un caso de *piggybacking*: el cliente reconoce todos los bytes recibidos y aprovecha la transmisión del segmento de ACK para enviar 1018 *bytes* de datos. Con el segmento 11, el servidor sólo reconoce una parte de los datos recibidos. En los segmentos 12 y 13, el servidor envía en total 384 *bytes* de datos, pero no reconoce nuevos datos recibidos. La llegada correcta de estos datos al lado cliente se reconoce en el segmento 14, otro ACK retardado. Evidentemente, este segmento ACK se pierde y el extremo servidor se ve obligado a retransmitir estos datos en el segmento 15. Esto es evidente en el Número de Secuencia del segmento 15.

Vale la pena destacar que la situación de *timeout* es una indicación fuerte de congestión para TCP. Por otra parte, el segmento 15 es una muestra de lo que en TCP se conoce como re-empaquetado o *repacketization*: en lugar retransmitir los datos en dos segmentos, como se habían transmitido originalmente, el protocolo realiza la retransmisión juntando ambos segmentos en uno.

El cliente reconoce la retransmisión en el segmento 16 y, posteriormente, en el segmento 17, transmite más datos. Luego, el lado servidor transmite un segmento con 384 *bytes* de nuevos datos, tal como lo indica su Número de Secuencia. Los segmentos 19 y 20 representan una *repacketization* de estos datos para su retransmisión. El segmento 21, proveniente del cliente, reconoce los datos previos y el segmento 22 carga 631 *bytes* de un pedido del cliente, que se reconocen en el segmento 23.

Los segmentos 24 y 25 son dos nuevas transmisiones del servidor. El segmento 26 es un ACK retardado del lado cliente. En el segmento 27, el cliente envía datos, que el servidor reconoce en el segmento 28. Nuevamente el servidor transmite dos segmentos, que el cliente reconoce con un ACK retardado.

En el segmento 32, el cliente transmite 757 *bytes* de datos, que el servidor reconoce en el segmento 33. La situación es similar para los segmentos 34 y 35. Los segmentos 36 y 37 provenientes del servidor, son reconocidos de manera acumulativa con el segmento 38 del cliente. En el segmento 39, el cliente vuelve a enviar datos y el servidor los reconoce con el ACK retrasado del segmento 40.

Si repitiéramos la misma conexión, enviando al servidor los mismos requerimientos, el flujo no tendría por qué ser igual que el presentado, ya que la forma en que se transfieren los datos depende de la situación particular, de ambos extremos y de la red, en un momento dado.

12.5.2 Intercambio de segmentos – Recepción de ACK duplicados

En la Fig. 12.4 se presenta otro intercambio de datos entre otro cliente y el mismo servidor, donde se puede observar una situación de congestión, menos grave que el *timeout*, que se percibe por la recepción de ACK duplicados.

En el segmento 1, el servidor transmite $53\ bytes$, luego 505 y $47\ bytes$, en los segmentos 2 y 3 respectivamente. En estos segmentos se observa el mismo número de ACK. Mediante el número de secuencia de estos segmentos, se puede inferir que el servidor ha transmitido $(56836 - 1)bytes$, ya que el número de secuencia inicial ISN también cuenta, aunque en el segmento SYN no se envían datos. También, por el Número de ACK, se interpreta que el servidor ha recibido segmentos correctos y en orden hasta el byte 10987.

En el segmento 4, el cliente reconoce todas las transmisiones previas, pero su número de secuencia es 2730 números mayor que lo que el servidor le ha reconocido. Esto se confirma en el segmento 5, cuando el servidor transmite $41\ bytes$, pero insiste con el mismo número de ACK que en los segmentos previos. En el segmento 6, el cliente tiene datos para transmitir y aprovecha para reconocer los datos del segmento previo transmitido por el servidor.

La llegada del segmento 6 de $41\ bytes$ de datos del cliente, dispara del lado servidor un ACK sin datos, que repite el Número de ACK 10988. En su forma tradicional, cuando el protocolo TCP recibe datos correctamente pero fuera de orden, lo único que puede hacer es repetir el número de ACK correspondiente al último segmento recibido en orden y sin errores. Es lo que se denomina generación de ACK duplicados. No es una situación tan grave como un *timeout*, pero es un indicador de que un segmento pudo haberse perdido, aunque hay otros que siguen llegando correctamente pero fuera de orden.

El aviso parece no ser suficiente para disparar una retransmisión, ya que el lado cliente sigue enviando nuevos datos, en los segmentos 8 y 9, primero $1460\ bytes$ y luego $24\ bytes$. La llegada de estos datos al servidor dispara dos nuevos ACK duplicados, se trata de los segmentos 9 y 10. Al recibirlos, el TCP del lado cliente considera necesaria la retransmisión del segmento con el número de secuencia esperado por el servidor, en el que carga $1460\ bytes$ de datos. Evidentemente, esta transmisión llena el hueco que existía en el buffer de recepción del protocolo TCP del lado servidor, pues este avanza el borde izquierdo de la ventana hasta el Número de ACK 15243 en el segmento 13.

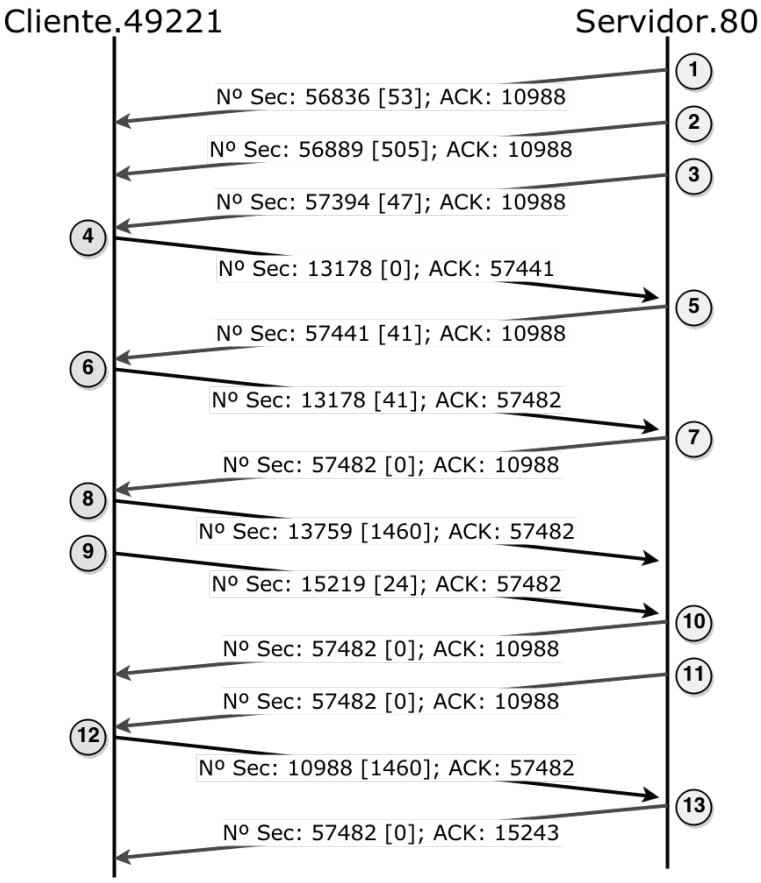


Figura 12.4 – Situación de congestión. ACK duplicados.

12.5.3 Control de Flujo - Manejo de las ventanas de transmisión y recepción

En la Fig. 12.5 se presenta otro intercambio cliente-servidor con el propósito de seguir la variación de las ventanas deslizantes de transmisión y recepción en entorno de flujo de grandes volúmenes de datos.

En este ejemplo, el intercambio TCP comienza con el inicio de la conexión. El cliente envía un segmento SYN, con el valor de la opción MSS que indica que se encuentra en una LAN, un anuncio de factor de escalamiento de ventana de 8 y el aviso de que puede hacer reconocimiento de datos por bloque, SACK. El extremo TCP servidor puede manejar las mismas opciones, aunque el factor de escalamiento que usará es 6, según lo anuncia en el segmento 2. Como

ambos entienden la opción de escalamiento de ventana, a partir de este momento cada segmento del lado cliente anunciará en el campo Ventana (win) un valor que deberá ser multiplicado por $2^8 = 256$ para hallar el verdadero tamaño de la ventana. Del lado servidor, cada anuncio de ventana deberá ser multiplicado por 64 del lado cliente, para hallar el verdadero valor de la ventana de recepción del servidor. Se ha elegido presentar los campos de *win* de los segmentos subsiguientes con el valor ya escalado, pero en realidad, por ejemplo en el segmento 4, el valor escrito en el campo *win* es 256, y en el segmento 5 es 108.

En el segmento 4, el cliente envía al servidor 513 $bytes$ de datos con Número de Secuencia 1. La transmisión de este segmento se traduce en una ventana de transmisión (recuadro de línea punteado) con 513 $bytes$ almacenados a la espera de un *ACK*. La ventana de recepción escalada anunciada por el cliente en este segmento es 65536 $bytes$.

La transmisión del segmento 5 obedece al hecho de que el servidor ha recibido los datos correctamente, quedando la ventana de recepción (recuadro de línea llena) con el puntero de la izquierda apuntando al número 514, el siguiente byte que espera recibir. El anuncio de la ventana escalada del servidor es 6912 $bytes$.

Luego el servidor transmite el segmento 6, con 352 $bytes$ de datos. Esto no modifica el valor de la ventana anunciada al otro extremo, pero sí modifica la situación de su propia ventana de transmisión, en este caso con 352 $bytes$ de datos a la espera de un ACK. La llegada de este segmento, coloca el estado de la ventana de recepción del cliente como se ha supuesto en la figura. El segmento 7 es otra transmisión del servidor, de 15 $bytes$, que modifica el estado de la ventana de transmisión tal como se presenta en el recuadro punteado. Del lado cliente, si los datos llegan correctamente, se almacenan en la ventana de recepción, a la espera de ser entregados a la aplicación. La transmisión del ACK en el segmento 8, confirma la llegada correcta y vuelve a dejar libre el buffer de recepción del lado cliente. La recepción de este segmento permite al protocolo TCP del lado servidor, retirar los datos pendientes de su ventana de transmisión.

A partir de este segmento la situación del par de ventanas de cada lado se mantiene estática porque ya no se intercambian más datos.

En este ejemplo, se ha incluido el cierre de la conexión, para resaltar que tanto el segmento SYN, como el segmento FIN, utilizan un número de secuencia, a pesar de no transportar datos.

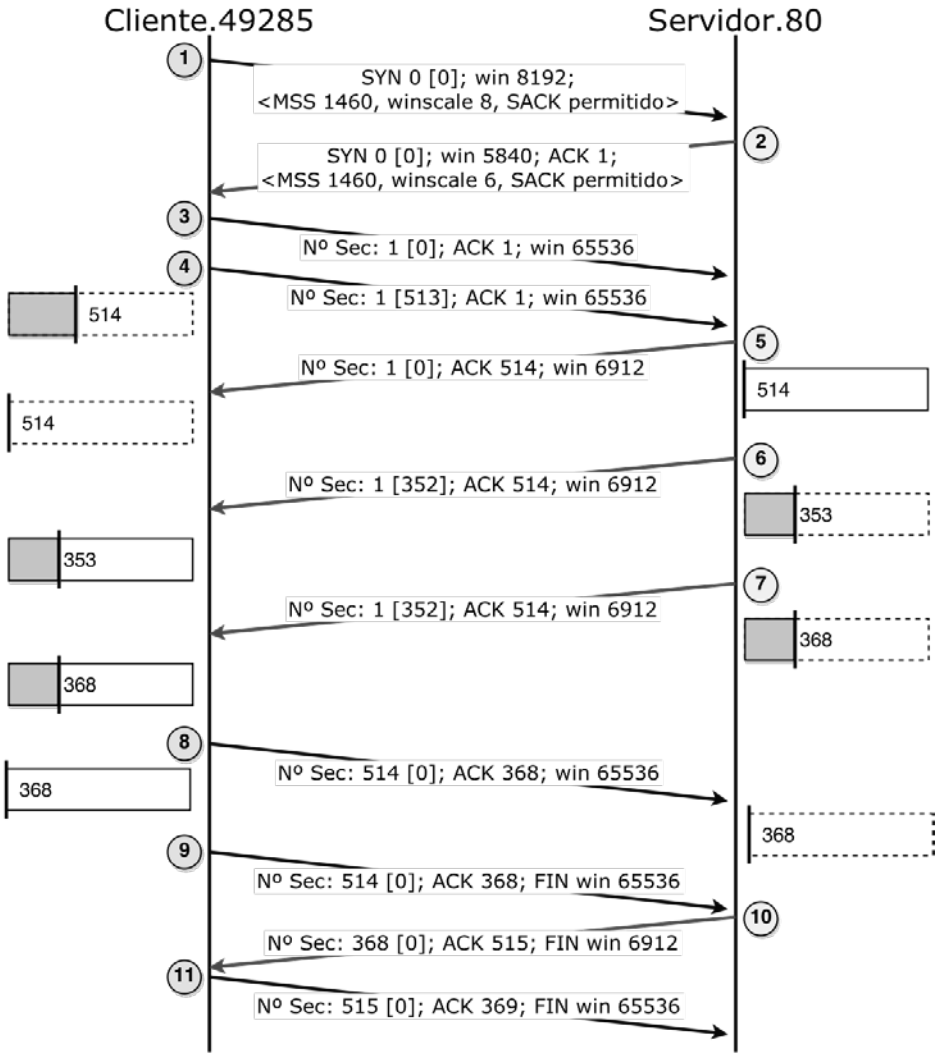


Figura 12.5 – Control de Flujo por Ventanas Deslizantes.

12.6 Aviso de ventana nula - Síndrome de la ventana tonta

El anuncio de la ventana es lo que permite realizar el control de flujo por parte del receptor. Dado que TCP almacena los segmentos en el buffer de recepción, para subir a las aplicaciones aquellos que llegaron correctamente y en orden, si las aplicaciones no están ocupadas, inmediatamente leerán los datos del buffer, manteniéndose el anuncio de la ventana en un valor casi estático, constante. Otras veces, cuando las aplicaciones tienen tareas que cumplir, no pueden vaciar inmediatamente el buffer de recepción. En estos casos, se observarán anuncios de ventana más pequeños, pudiéndose llegar a una situación de aviso de ventana nulo. Un aviso de este tipo detendrá el flujo de

datos transmitidos desde el otro extremo. La forma de reanudar la comunicación es enviando un segmento de actualización del estado de la ventana, una vez que se haya liberado espacio en el buffer de recepción.

En general, luego de un aviso $win = 0$, la actualización del tamaño de la ventana o *window update*, se realiza por medio de un segmento sin datos, o sea un segmento ACK, cuya transmisión no es confiable. El peligro de que dicho segmento se pierda podría conducir a una situación de punto muerto o *deadlock*, donde cada extremo espera que el otro realice alguna acción: el transmisor espera la actualización y el receptor espera que le envíen datos.

Para evitar estas situaciones, TCP incluye un reloj persistente, conocido como *timer persist*, que dispara la transmisión de segmentos de sondeo de ventana, también conocidos como *window probes*, para obligar al extremo que anunció la ventana nula a realizar una actualización. La RFC 1122 aconseja disparar el mecanismo luego del primer RTO transcurrido desde el aviso $win = 0$ y, a partir de allí, manejarlo con un mecanismo de *backoff* exponencial. No se define un límite máximo de segmentos de sondeo transmitidos, simplemente el mecanismo deja de funcionar al recibir una respuesta $win \neq 0$.

Los segmentos de sondeo se cargan con un byte de datos, para que su transmisión sea confiable, obligando al receptor a generar un aviso de ventana por medio de un segmento ACK. El dato se aceptará o no del lado receptor, dependiendo del estado del buffer de recepción. Por ejemplo, si el último ACK recibido con aviso de ventana $win = 0$ transporta un Número de ACK 9034 , entonces el segmento de sondeo llevará un Número de Secuencia 9034, que es el que espera recibir el otro extremo. Puede cargar un byte de datos por que el protocolo permite que la numeración sobrepase el borde de la ventana cerrada. La respuesta del otro extremo, si la ventana continuara siendo nula, es un segmento ACK de número 9034, es decir que no se reconoce el byte de datos. Por este motivo el segmento *window probe* se sigue retransmitiendo.

El aviso de ventana nula puede causar problemas si no se imponen ciertas condiciones. Si el receptor, ni bien sale de la condición de ventana cerrada, empieza a anunciar tamaños de ventana pequeños, y el transmisor aprovecha estos anuncios para transmitir datos inmediatamente, vuelve a llenar la ventana, provocando nuevos avisos de $win = 0$. Esta situación se conoce como Síndrome de la Ventana Tonta (SWS, Silly Window Syndrome). Para evitarlo, basta con cumplir una serie de reglas, presentadas en la RFC 1122:

- **Regla 1*:*** Los receptores no deben anunciar tamaños de ventana pequeños, aunque sean mayores que los advertidos previamente, hasta que la ventana pueda alcanzar el un valor que se fija en el mínimo entre el tamaño de un segmento completo, o sea el valor MSS, o la mitad del espacio del buffer de recepción. Esta regla se aplica cuando las aplicaciones liberan espacio del buffer de recepción, disparando actualizaciones, o en respuesta a *window probes*.

- **Regla 2***:* Para evitar el síndrome de la ventana tonta, los transmisores no deben enviar segmentos, al menos que se cumplan alguna de las siguientes condiciones:

    1. Se puede enviar un segmento de tamaño MSS. Esta condición se incluye para evitar la ineficiencia debida al *overhead*, presente al transmitir segmentos pequeños.
    2. Se puede enviar al menos la mitad del tamaño de ventana máximo anunciado por el receptor durante la conexión. Esta condición es para manejar la situación de aquellos dispositivos que anuncian tamaños de ventana de pocos bytes, tal vez menores que MSS.
    3. Se puede transmitir todo lo que queda pendiente mientras no se está esperando ACK por ningún dato o el algoritmo de Nagle está deshabilitado para esta conexión. Esta última condición es aplicable al caso en que la aplicación del lado transmisor realice pequeñas escrituras.

12.7 Conexiones ociosas - Mecanismo Keepalive

Se ha explicado hasta aquí la forma en que se establece una conexión, con su correspondiente reserva de recursos, algunas cuestiones relativas al intercambio de datos y los aspectos relevantes del cierre de una conexión, ya sea que suceda de forma ordenada o desordenada, liberando los recursos pre-asignados.

En este apartado se abordará la situación de aquellas conexiones que, una vez establecidas, permanecen ociosas, sin transmitir ni recibir datos. Esta situación podría deberse a una serie de razones no tan inusuales.

Un cliente podría establecer una conexión TCP con una aplicación servidora, y luego el propio usuario retirarse de la máquina y dejar la conexión establecida sin intercambio de datos por un tiempo indeterminado. También, podría suceder que cayeran *routers* intermedios entre ambos extremos de la comunicación, condenando las conexiones a la inactividad, aunque se encuentren establecidas. Dependiendo de las circunstancias, podría ser útil que los extremos permanezcan conectados aunque tengan muy pocos datos para intercambiar. En otros casos, sería mejor terminar la conexión, para no desperdiciar recursos.

Muchas implementaciones TCP proveen un método, denominado mantener vivo o *keepalive*, para probar si el otro extremo de la conexión se encuentra presente. El método lleva un reloj asociado, que se conoce con el mismo nombre, y que permite marcar tiempos que disparan el envío segmentos de sondeo al otro extremo, para comprobar si se encuentra presente.

El problema con este esquema es que se pueden confundir situaciones de ociosidad legítimas con fallas transitorias de la red, terminando conexiones por motivos ajenos a las mismas. La controversia se extiende cuando se menciona que las aplicaciones deberían hacerse cargo de esta funcionalidad, aunque la

contrapartida afirma que, si hubiese muchas aplicaciones que lo precisasen, lo más lógico sería que TCP la proveyera.

Como la funcionalidad es opcional, existen algunos parámetros de configuración que se ajustan para controlar la ausencia de actividad de una conexión por algún período de tiempo, para disparar el mecanismo de sondeo. Un parámetro ajusta el período de repetición de los segmentos de sondeo, cuando no se recibe respuesta del otro extremo. También es posible configurar un número máximo de estos segmentos. Cuando se alcanza el límite sin respuestas, se considera el otro extremo inalcanzable y se finaliza la conexión.

A diferencia de los segmentos de sondeo del reloj persistente, el número de secuencia de un sondeo *keepalive* es un número menos que el mayor número de ACK recibido desde el otro extremo. Al igual que los segmentos de sondeo del reloj persistente, estos segmentos pueden cargar un byte de datos. El dato en sí no es relevante para la conexión, ni queda sujeto al mecanismo tradicional de retransmisión, pero, de encontrarse presente y activo, el otro extremo contestará con un ACK.

En cualquier caso, existen cuatro escenarios posibles:

1. El dispositivo en el otro extremo se encuentra activo y alcanzable. El protocolo TCP responderá con un ACK, reiniciando el reloj de *keepalive* en el extremo TCP que realiza el sondeo.

2. El dispositivo en el otro extremo ha caído y se encuentra en proceso de re-arranque. No se recibirá respuesta por parte del protocolo TCP del otro extremo, por lo que se continuará con el sondeo *keepalive*, según los parámetros de expiración y número máximo de segmentos de prueba especificados. Si se alcanza este número y aún no se recibe respuesta, se da por terminada la conexión. La aplicación que realiza el sondeo puede recibir entonces un mensaje de "error por expiración". Vale la pena aclarar que, cuando un dispositivo cae, no puede realizar el cierre, ni siquiera emitiendo un segmento RST.

3. El dispositivo en el otro extremo ha caído y re-arrancado. La respuesta al sondeo es un segmento con la bandera RST levantada, ya que no se reconoce la conexión que había estado establecida. Esta respuesta termina la conexión y se comunica a la aplicación a través de un mensaje que suele ser del tipo "conexión reiniciada por el par".

4. El dispositivo en el otro extremo se encuentra activo pero es inalcanzable por problemas en la red. La situación no se podrá distinguir del caso 2, aunque podría llegar a recibirse un mensaje de error ICMP.

La confusión entre el segundo y el cuarto escenario es lo que genera tanta controversia con respecto al uso de esta herramienta.

En sistemas tipo Linux, los valores default del reloj de *keepalive* suelen ser de $2\ hs$, ajustándose en $75\ seg$ el intervalo de sondeo y 9 segmentos como

valor máximo de pruebas de sondeo. En Windows, en cambio, el intervalo se suele fijar en 1 seg y el valor máximo en 10 segmentos.

El mecanismo de *keepalive* no se encuentra definido en la especificación de TCP pero, como algunos Sistemas Operativos lo incorporan, la RFC 1122 aclara que utilizarlo puede llevar a cerrar una conexión perfectamente establecida por falla en la Internet. También se critica en dicha RFC el consumo de ancho de banda de los mensajes en una conexión naturalmente ociosa y el costo en aquellos casos donde el cargo es por paquete transmitido.

Bibliografía

1. RFC 793 "Transmission Control Protocol ", September 1981. http://tools.ietf.org/html/rfc793
2. RFC 896 "Congestion Control in IP/TCP Internetworks", January 1984. http://tools.ietf.org/html/rfc896
3. RFC 1122 "Requirements for Internet Hosts -- Communication Layers", October 1989. http://tools.ietf.org/html/rfc1122
4. RFC 1323 "TCP Extensions for High Performance", May 1992. https://www.ietf.org/rfc/rfc1323.txt
5. RFC 6093 "On the Implementation of the TCP Urgent Mechanism", January 2011. http://tools.ietf.org/html/rfc6093
6. RFC 6247 "Moving the Undeployed TCP Extensions RFC 1072, RFC 1106, RFC 1110, RFC 1145, RFC 1146, RFC 1379, RFC 1644, and RFC 1693 to Historic Status ", May 2011. http://tools.ietf.org/html/rfc6247
7. RFC 6298 "Computing TCP´s Retransmission Timer", June 2011. http://tools.ietf.org/html/rfc6298
8. Jacobson, Van, "Congestion Avoidance and Control", CM SIGCOMM Computer Communication Review. ACM, 1988. p. 314-329.
9. Karn, P., C. Partridge, "Improving Round-Trip Time Estimates in Reliable Transport Protocols", SIGCOMM 87. http://ccr.sigcomm.org/archive/1995/jan95/ccr-9501-partridge87.pdf
10. Kozierok, Charles M., "The TCP/IP Guide". http://www.tcpipguide.com/free/t_toc.htm
11. Comer, Douglas, "Internetworking with TCP/IP: Principles, Protocols and Architecture v. 1". Pearson Education, 1995.
12. Stevens, W. Richard, "TCP/IP Illustrated, Vol. 1: The Protocols (Addison-Wesley Professional Computing Series) ". Addison-Wesley, 1993.

Problemas

1. ¿Cómo se establece el *RTO* para una conexión TC? ¿Qué importancia tiene el Algoritmo de Karn?

2. Explique el concepto de re-empaquetado en TCP.

3. ¿Cuál es la característica del flujo de datos interactivos? ¿Para qué se utiliza el Algoritmo de Nagle?

4. ¿Cuál es el sentido del *timer persist*? ¿Qué relación existe entre el *timer persist* y el síndrome de la ventana tonta?

5. Mencione ventajas y desventajas asociadas con la implementación de un *timer keepalive*.

6. Sobre la siguiente captura de un segmento TCP, sabiendo que el cliente únicamente ha transferido 393 bytes al Servidor desde que comenzara la conexión, contestar:
   a) Nº de Secuencia del SYN inicial de Cliente a Servidor y Nº de Secuencia del SYN del Servidor al Cliente.
   c) Nº de Secuencia y Nº de ACK del segmento con los 393 bytes de Cliente a Servidor.
   d) Sabiendo que en el SYN inicial de Cliente a Servidor el campo de opciones era: < Maximum segment size: 1460 bytes, NOP, Window scale: 1, NOP, NOP, SACK permitted >, determine el verdadero valor de la ventana anunciada en el segmento capturado.

Transmission Control Protocol,
Source port: http (80)
Destination port: 1543 (1543)
Sequence number: 2976738361
Acknowledgement number: 1544685068
Header length: 20 bytes
Flags: 0x0010 (ACK)
0... .... = Congestion Window Reduced (CWR): Not set
.0.. .... = ECN-Echo: Not set
..0. .... = Urgent: Not set
...1 .... = Acknowledgment: Set
.... 0... = Push: Not set
.... .0.. = Reset: Not set
.... ..0. = Syn: Not set
....0 = Fin: Not set
Window size: 6432
Checksum: 0xfc01 (correct).

CAPÍTULO XIII

Control de Congestión y Nuevas Opciones TCP

En este capítulo se explicará una serie de reacciones que el protocolo TCP presenta frente a situaciones de congestión. El comportamiento se implementa por medio de algoritmos, sobre todo en el caso de intercambio de grandes volúmenes de datos.

En capítulos previos, se ha abordado el tema de control de flujo, contexto en el que se trata de frenar la inyección de datos por pedido de la entidad receptora. El transmisor adapta su velocidad de envío de datos en base al anuncio de ventana que realiza el receptor.

En el caso de control de congestión, se debe frenar la inyección de datos por percepción de una situación propia de la red, al borde o en estado de congestión. En términos de los protagonistas de la comunicación, esto significa que hay descarte de paquetes en los routers, porque se les exige manejar más tráfico del que pueden. El desafío consiste en proveer al protocolo TCP de indicadores de la situación, para que sea capaz de desarrollar una reacción adecuada. También se pretende que pueda percibir la evolución, por si la congestión es transitoria y existe alguna forma de recuperación.

En este capítulo, además de los puntos más importantes relacionados con el control de congestión TCP, se desarrollarán los conceptos referidos a las nuevas opciones del protocolo, incorporadas posteriormente para mejorar su comportamiento, adaptándolo a las tecnologías más modernas.

13.1 Percepción de la congestión

La situación de congestión de una red se traduce en la pérdida o retraso en el procesamiento de los paquetes, por saturación de las colas de entrada de los *routers*. A nivel TCP, una situación grave de congestión se percibe por el vencimiento de los tiempos asociados al RTO, generándose retransmisiones que, por su parte, inyectan nuevamente paquetes a una red ya congestionada. Esta situación podría estar vinculada a miles de conexiones TCP, cuyos segmentos se

encuentran atravesando los *routers* desbordados, empeorando de este modo la situación existente. Para evitar estas complicaciones, el protocolo TCP incorpora mecanismos para percibir y contrarrestar la congestión. De manera implícita, la propia filosofía con la que trabaja TCP puede ayudar a percibir la situación, por observación de la relación entre segmentos transmitidos y reconocimientos recibidos desde el otro extremo. Por este motivo, no es necesario incluir mecanismos explícitos de sondeo como los expuestos en el capítulo anterior, válidos para otro tipo de situaciones.

Otra situación de congestión, aunque menos grave que la mencionada, se asocia a la recepción de segmentos ACK duplicados, dado que el protocolo genera este tipo de segmentos cuando recibe datos correctos pero fuera de orden. La llegada de segmentos desordenados puede ser indicador de retraso o de pérdida de otros segmentos, decidiéndose por alguno de ambos casos según la cantidad de segmentos duplicados recibidos.

La RFC 793, definición original de TCP, no incluyó una explicación precisa de este tipo de previsiones pero, a medida que Internet se desarrollaba de manera explosiva, se hizo necesaria la definición de mecanismos de percepción y control. La RFC 2001 documentó varios trabajos previos que no estaban desarrollados como estándares. Su autor, W. Stevens, pensó que era necesario desarrollar definiciones claras de cuatro algoritmos propuestos por Von Jacobson: Arranque Lento (*Slow Start*), Evitar la Congestión (*Congestion Avoidance*), Retransmisión Rápida (*Fast Retransmit*), y Recuperación Rápida (*Fast Recovery*). Posteriormente surgieron actualizaciones, plasmadas en la RFC 2581 que luego fueron actualizadas por la RFC 5681.

Otros métodos de detección de congestión, complementarios de estos algoritmos tradicionales, basados en el mecanismo de Notificación Explícita de Congestión (ECN, Explicit Congestion Notification), permiten informar la situación antes de que comience una situación de pérdida de paquetes en los *routers*, para permitir que los extremos de la conexión tomen los recaudos necesarios. Este mecanismo no se ha desarrollado ampliamente pues se apoya en la implementación de protocolos adicionales y capacidades especiales en los propios dispositivos de enrutamiento.

13.2 Algoritmos Clásicos para Control de Congestión

El protocolo TCP asume que la pérdida de paquetes es consecuencia de una situación de congestión. La reacción propuesta por los diseñadores de diversos algoritmos consistió en bajar la velocidad de transmisión.

Se ha visto que el mecanismo de control de flujo es capaz de bajar la velocidad de transmisión mediante el aviso de la ventana de recepción win. Se trata de una herramienta impuesta por el receptor para control de la velocidad de intercambio establecida entre extremos de la comunicación. En el caso de una red congestionada, lo apropiado sería encontrar otro mecanismo que permitiera al transmisor, una vez detectada la situación de congestión propiamente dicha, bajar la velocidad para no agravar el problema.

Con este propósito, TCP incorpora una variable extra para el control de la ventana de transmisión W. Se trata de la ventana de congestión *cwnd* (*congestion window*), un parámetro variable durante la transmisión con el cual se trata de estimar la situación en la que se encuentra la red en un momento determinado.

El protocolo establece que no se podrá transmitir mayor cantidad de bytes que lo establecido por el valor alcanzado por $cwnd$, siempre que esta variable permanezca por debajo del valor anunciado win. Dicho en otras palabras:

$$W = \min(cwnd, win) \tag{13.1}$$

La variable W representa la cantidad máxima de datos transmitidos que el protocolo TCP puede tener en espera de recepción de ACK. La RFC también denomina tamaño en vuelo o *flight size* a la cantidad de datos transmitidos pendientes de ACK, cantidad que siempre es menor que W.

La idea detrás del algoritmo de control de congestión es establecer dinámicamente el valor de W de acuerdo a las variaciones percibidas en el estado de la red y el estado del buffer de recepción. Lo ideal sería que se estableciera en un valor óptimo, de tal manera que quedara sincronizado el envío de segmentos con la recepción de ACK, en un esquema de auto-sincronismo dinámico. Esta condición depende del tiempo de ida y vuelta de la conexión RTT y de la velocidad posible de transmisión, limitada por el enlace más comprometido en todo el camino entre transmisor y receptor r_{bmin}. El producto de estos dos parámetros ofrece una estimación del valor de la ventana óptima en una determinada situación:

$$W_{\acute{o}pt} = RTT \; x \; r_{bmín}. \tag{13.2}$$

Lograr un esquema que mantenga dinámicamente el valor de W cercano al valor $W_{\acute{o}pt}$ es uno de los objetivos del trabajo conjunto de varios algoritmos para el control de congestión.

13.2.1 Arranque Lento - Slow Start

Al comenzar la transferencia de datos, el único parámetro cierto que conoce el transmisor es el tamaño de ventana anunciado por el receptor win. Arrancar una conexión TCP inyectando segmentos, de acuerdo al límite impuesto por el tamaño de ventana informado por el receptor, podría funcionar sin inconvenientes si ambos extremos se encontraran en la misma red LAN, pero esta forma de transferencia sería problemática para dispositivos ubicados en distintas redes, si es que existen enlaces lentos y *routers* sobrecargados en el camino entre transmisor y receptor.

El algoritmo de Arranque Lento (Slow Start) con que se inicia una conexión TCP toma la precaución de tener en cuenta estos detalles, interpretando

que la cantidad de segmentos enviados sobre la red tiene directa relación con los segmentos ACK devueltos desde el lado receptor. Por este motivo, TCP inicia una conexión de manera conservadora, lenta, tratando de probar la capacidad disponible, evitando generar demasiada carga, con el propósito de alcanzar una situación estable de auto-sincronismo con respecto a los arribos de segmentos ACK. También se recurre a este algoritmo cuando se genera una re-transmisión por *timeout*, como una forma de re-sincronizar la propia conexión.

Luego de completar el protocolo de tres vías del inicio de la conexión, *Slow Start* establece que el valor inicial de la ventana de congestión debe ser de 1 segmento SMSS, interpretándose SMSS como el valor MSS del lado transmisor:

$$cwnd = 1 \text{ SMSS}. \tag{13.3}$$

De este modo, en el inicio, según la Ec. (13.1), también se fija el valor para la ventana de transmisión $W = 1$, eligiendo inicialmente transmitir un único segmento. Sin embargo, el algoritmo establece que se debe incrementar el valor de la ventana de congestión al ritmo de recepción de segmentos ACK. Es decir, en el inicio la ventana de congestión se abre en un segmento cada vez que se recibe un ACK que cubra nuevos datos:

$$cwnd = cwnd + 1SMSS. \tag{13.4}$$

En realidad, la RFC 5681 acepta una pequeña modificación propuesta en la RFC 3465, para variar el tamaño de $cwnd$. En vez de aumentar el valor en una cantidad fija por cada arribo de ACK, se sugiere basar el incremento en el número de bytes N, previamente no reconocidos, que cada ACK cubre:

$$cwnd = cwnd + \text{mín}(N, SMSS). \tag{12.5}$$

Así, el transmisor comienza la conexión transmitiendo un segmento de datos y, la consecuencia práctica será que la ventana de congestión irá aumentando gradualmente su valor, como máximo a dos segmentos luego del primer RTT, luego a cuatro segmentos en el segundo RTT y así sucesivamente.

En cualquier caso, la cantidad de datos transmitidos deberá respetar el cumplimiento de la Ec. (13.1) pero, si se considera un buffer de recepción ilimitado del lado receptor, el ritmo de inyección de segmentos en la red quedará fijado por el ritmo de incremento de $cwnd$.

De este modo, mientras la red lo permita, el crecimiento inicial no será para nada lento, sino que en realidad crecerá casi de manera exponencial.

A modo de ejemplo, la Fig. 13.1 presenta un caso más realista del aumento de $cwnd$ en el arranque de una conexión, suponiendo que la ventana anunciada no genere limitación alguna. Al principio, el transmisor inyecta un

único segmento en la red y luego del primer RTT recibe el ACK, por lo que puede fijar $cwnd = 2\ segmentos$ y transmitir dos segmentos. Cuando recibe el ACK del primero de ellos puede fijar $cwnd = 3\ segmentos$ y, recién cuando recibe el ACK para el segundo, puede ajustar $cwnd = 4\ segmentos$ y transmitir en consecuencia. Es decir que la ventana de congestión podrá avanzar a $cwnd = 8\ segmentos$, luego de haber recibido los respectivos ACK, tal como se grafica en la figura.

El límite superior de este crecimiento inicial lo impondrá la propia red con algún indicador de congestión, ya sea por pérdida de segmentos (*timeout* y retransmisión) o por la recepción de determinada cantidad de ACK duplicados. De este modo, en el inicio de la conexión, se intenta llegar hasta el valor de máxima capacidad de transmisión permitido por la red subyacente. Llegado al valor máximo posible, algún indicador de congestión obligará a disminuir el valor de $cwnd$, de acuerdo a lo establecido por el algoritmo a aplicar en cada caso.

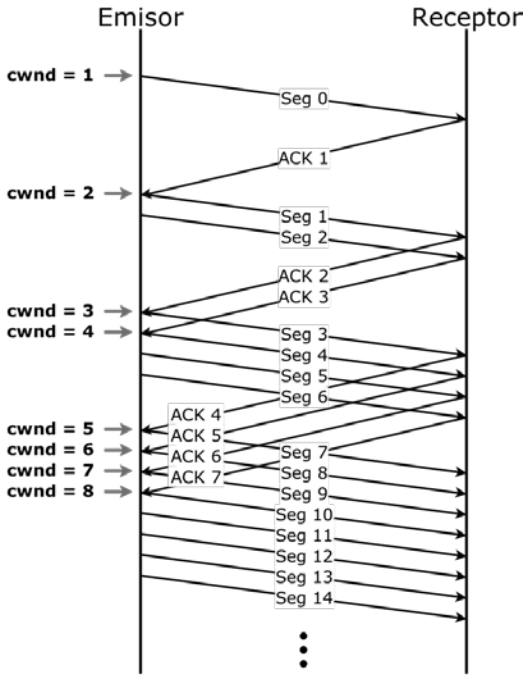


Figura 13.1 – Slow Start

Otra variable de estado aparece en juego cuando se intenta controlar situaciones de congestión. Se trata del umbral de arranque lento (*slow start threshold)* $ssthresh$.

Esta variable determina si el control de la transmisión de datos se realiza por el algoritmo *Slow Start* o por el algoritmo *Congestion Avoidance*. El valor inicial de $ssthresh$ debería ser arbitrariamente alto para que la transferencia inicial verdaderamente fuese realizada según las condiciones de la red. Se establece que $ssthresh$ debe disminuirse en respuesta a una situación de

congestión, pues determinará un límite a la forma de comportamiento, en cuanto a la inyección de paquetes, luego de detectarse este tipo de eventos.

13.2.2 Algoritmo para Evitar la Congestión - Congestion Avoidance

El algoritmo para Evitar la Congestión (Congestion Avoidance) asume que la pérdida de paquetes por errores es despreciable, por lo que supone que este indicador es debido exclusivamente a la saturación de las colas de los *routers* en algún punto entre fuente y destino de la conexión TCP. Como ya se ha mencionado en el apartado previo, existen dos indicadores de pérdida de paquetes: *timeout* y recepción de ACK duplicados.

Ya que el algoritmo para evitar la congestión trabaja en conjunto con el de arranque lento, se utiliza el valor umbral para establecer cuándo opera uno y cuando el otro, de tal manera que:

$$cwnd < ssthresh \Rightarrow Slow\ Start$$
$$cwnd > ssthresh \Rightarrow Congestion\ Avoidance \quad (13.6)$$

Básicamente, la diferencia entre ambos es la velocidad a la que aumenta posteriormente el valor de la ventana de congestión $cwnd$. En el primer caso el crecimiento es exponencial, en el segundo caso es lineal.

El algoritmo combinado se puede resumir de la siguiente manera:

1. **Inicialización**: en el arranque de la conexión se establece $cwnd = 1\ segmento$ y $ssthresh = 65535\ bytes$.
2. **Transmisión**: siempre se puede transmitir una ventana respetando la condición $W = (cwnd, win)_{mín}$.
3. **Percepción de congestión**: si se percibe congestión, ya sea por *timeout* o por recepción de ACK duplicados, se ajusta el valor del umbral $ssthresh = W/2$ (imponiendo un valor mínimo dos segmentos). El valor es la mitad del valor alcanzado por la ventana de transmisión en el momento de producirse la situación de congestión.
   Además, si la situación de congestión es por *timeout* se ajusta el valor de la ventana de congestión a $cwnd = 1\ segmento$, es decir que se aplica *Slow Start*.
4. **Luego de la situación de congestión**: a medida que regresen segmentos ACK para la transmisión de nuevos datos, se incrementa $cwnd$, aunque la velocidad de incremento depende de si TCP se encuentra en *Slow Start* o en *Congestion Avoidance*.

De este modo, luego de una situación de congestión por *timeout*, se aplica *Slow Start* hasta que el valor de la ventana de congestión $cwnd$ alcance la mitad del valor de la ventana de transmisión que generó la congestión, que queda

almacenado en la variable $ssthresh$. Es decir que el umbral funciona como una barrera de advertencia para el protocolo. Cuando $cwnd$ alcanza este valor, el algoritmo *Congestion Avoidance* se hace cargo de la situación. Este algoritmo propone un crecimiento lineal de la ventana de congestión $cwnd$, ya que autoriza un aumento de, como máximo 1 segmento por cada RTT transcurrido, no importa cuántos ACK se hayan recibido. Esto parece bastante apropiado, ya que el protocolo está inyectando segmentos en una cantidad cercana a la que produjo la situación de congestión, más de la mitad de la ventana. Al superar el valor, se debe actuar con cautela, esperando percibir cuál es la nueva situación de la conexión a medida que ésta avanza.

Más precisamente, la RFC 5681 propone que, durante la aplicación de *Congestion Avoidance*, el incremento de la ventana de congestión se rija por la siguiente expresión:

$$cwnd_{+} = SMSS \; x \; SMSS/cwnd \tag{13.7}$$

El ajuste se debe ejecutar para cada ACK nuevo, que no sea un duplicado. Los diseñadores consideraron que esta forma de actualización provee una aproximación aceptable al incremento lineal de un segmento completo por cada RTT, debido a que el propio valor de $cwnd$ aumenta muy poco y se encuentra en el denominador de la Ec. (13.7).

Por ejemplo, si se supone una ventana $cwnd = 4 \; xSMSS$, la llegada del primer ACK generará un incremento de $0.25x \; SMSS$, colocando el valor de la ventana en $cwnd = 4.25 \; SMSS$. El segundo ACK coloca el nuevo valor en $cwnd = 4.48 \; x \; SMSS$ y la llegada del tercer ACK lo lleva a $cwnd = 4.70x \; SMSS$. De este modo, el cuarto ACK, recibido luego de un tiempo equivalente a RTT genera $cwnd = 4.91x \; SMSS$. Prácticamente, la ventana se ha agrandado en un segmento más durante RTT. En el caso de recepción de ACK retardados el crecimiento también es lineal, pero un poco más lento.

En la Fig. 13.1 se presenta la evolución del valor de la ventana de congestión $cwnd$ normalizado al tamaño de un segmento $SMSS$, en función del tiempo, medido en unidades de RTT.

Para el caso particular de la figura, la conexión arranca normalmente, recibiendo un segmento ACK para cada segmento transmitido hasta $RTT = 5$, motivo por el cual la ventana de congestión puede alcanzar el valor $cwnd = 32$. Entonces se produce un *timeout*. En $RTT = 6$, el protocolo reacciona arrancando nuevamente con el algoritmo *Slow Start*, fijando $cwnd = 1$ y $ssthresh = 16$, la mitad del valor de la ventana W que produjo la congestión.

Nuevamente, se comienza inyectando un único segmento en la red y, cuando regresa su ACK, se prueba inyectando dos segmentos. Si llega el reconocimiento para ambos, se aumenta la ventana de congestión a $cwnd = 4$. En el siguiente RTT se logran transmitir 8 segmentos con éxito, por lo que el valor de $cwnd$ puede llegar a 16. Habiendo alcanzado el valor umbral, el crecimiento ya no puede ser exponencial. A partir de aquí, de no percibirse congestión, el crecimiento de $cwnd$ podrá ser de 1 segmento por cada RTT,

aplicando el algoritmo de *Congestion Avoidance*. En la figura se puede observar este comportamiento de apertura lineal hasta el instante 14, donde se presenta otro síntoma diferente de congestión: la recepción de ACK duplicados. Por ahora, sólo se invita al lector a notar la diferencia en la reacción con respecto al caso de *timeout*.

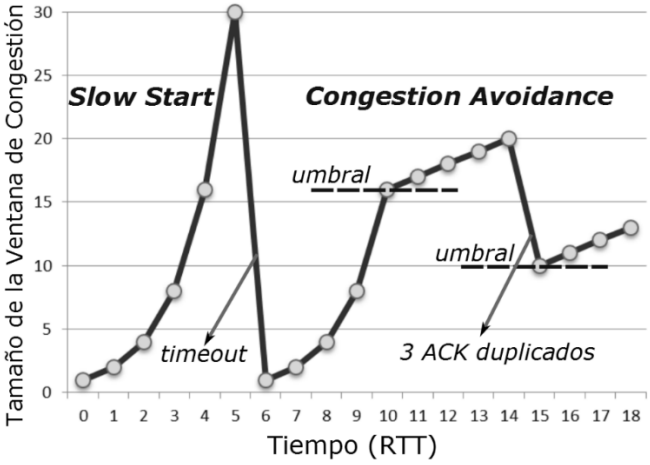


Figura 13. 2 – Slow Start y Congestion Avoidance.

13.2.3 Retransmisión Rápida - Fast Retransmit y Recuperación Rápida - Fast Recovery

En 1990, Von Jacobson propuso una modificación al algoritmo descripto en el apartado previo. Bajo la premisa de que TCP puede generar un segmento ACK duplicado por cada segmento recibido correctamente fuera de orden, el protocolo puede interpretar que, si sólo se trata de un retraso o un reordenamiento, el número de segmentos ACK duplicados puede ser uno o dos, para luego recibir el nuevo segmento de ACK que corresponda al reconocimiento de todos los datos enviados. En cambio, la recepción de tres o más duplicados, se interpreta como un indicador fuerte de pérdida de un segmento.

Por estos motivos, cuando se reciben tres o más ACK duplicados, TCP procede al re-envío inmediato del segmento supuestamente perdido, aún antes de que se venza el RTO. Esta forma de actuar se conoce como retransmisión rápida (*fast retransmit*), justamente porque no se espera el vencimiento del reloj de retransmisión.

A pesar de no ser tan grave como la situación de *timeout*, ya que la conexión sigue entregando datos, el protocolo toma recaudos para evitar la congestión, no siendo necesario arrancar nuevamente con *Slow Start*, ya que esto implicaría una decisión demasiado drástica. En este caso, el proceder se conoce con el nombre de algoritmo de recuperación rápida (*fast recovery*). Se trata de un

algoritmo que permite mejorar la eficiencia en condiciones de congestión moderada, especialmente cuando se han alcanzado grandes valores de ventana.

Ambos algoritmos se implementan juntos, bajo las siguientes premisas:

1. **Recepción de 3 ACK duplicados seguidos**: se fija $ssthresh = {}^{W}/_{2}$, la mitad del valor de la ventana que generó la congestión. Se debe retransmitir el segmento perdido (*fast retransmit*) y ajustar el valor de la ventana de congestión a $cwnd = {}^{W}/_{2} + 3$. De este modo, $cwnd$ aumenta en la misma proporción que el número de segmentos que han llegado al otro extremo.
2. **Llegada posterior de ACK duplicado**: cada vez que llega un segmento ACK duplicado, $cwnd$ aumenta en el tamaño de un segmento. Este proceder aumenta la ventana de congestión en proporción al segmento adicional que abandonó la red. Se puede transmitir un segmento si el nuevo valor de la ventana de congestión lo permite.
3. **Llegada posterior de un nuevo ACK**: cuando se recibe un ACK que reconoce nuevos datos, se ajusta el valor de la ventana de congestión al valor del umbral del paso 1, es decir $cwnd = ssthresh = {}^{W}/_{2}$. Supuestamente, este nuevo ACK debería ser el del segmento retransmitido en el paso 1, un RTT luego de su retransmisión. También, este nuevo ACK debería reconocer todos los segmentos intermedios entre el segmento perdido y la recepción del primer ACK duplicado. Este paso es la aplicación del algoritmo *Congestion Avoidance*.

En este punto, se invita al lector a observar la última parte de la evolución de la ventana de congestión de la Fig. 13.2.

Para mayor claridad, en la Fig. 13.3, se presenta el intercambio de segmentos en una conexión que padece este tipo de congestión.

La conexión arranca inicializando las variables $cwnd = 1\ segmento$ y $ssthresh = 65535\ bytes$, como exige *Slow Start*. Se supone que no hay limitación por el anuncio de ventana receptora, por tanto, la evolución de la conexión seguirá el valor alcanzado por la ventana de congestión.

Se transmite el primer segmento y, al recibir el ACK, se establece $cwnd = 2\ segmentos$. Se transmiten dos segmentos, cuya aviso de recepción correcta permite continuar aumentando el valor de la ventana a $cwnd = 4\ segmentos$.

La situación evoluciona siguiendo los lineamientos del algoritmo *Slow Start* hasta que la ventana alcanza el valor $cwnd = 8\ segmentos$. Entonces es posible transmitir los segmentos designados con Número de Secuencia 7 a 14. Cada uno de ellos, a excepción del último, dispara un nuevo ACK, permitiendo que la ventana de congestión siga aumentando hasta $cwnd = 15\ segmentos$.

El segmento con Número de Secuencia $NS = 14$ se pierde, por lo que el arribo del segmento $NS = 15$ dispara el segundo ACK duplicado. A partir de allí,

cada segmento fuera de orden que llegue al otro extremo, disparará un ACK duplicado.

La llegada del tercer ACK duplicado genera una rápida reacción. Se modifica el valor del umbral a la mitad del valor de la ventana (parte entera) que generó la situación de congestión $ssthresh = \lfloor 15/2 \rfloor = 7$. Se retransmite (*fast retransmit*) el segmento perdido $NS = 14$, y se ajusta el valor de la ventana de congestión al del umbral más tres: $cwnd = 10\ segmentos$. Se trata del algoritmo *fast recovery*.

A partir de allí, la llegada de cada ACK duplicado permitirá abrir en un segmento el valor de la ventana de congestión. Por ejemplo, el cuarto ACK duplicado fija $cwnd = 11\ segmentos$, pero como existen más segmentos pendientes, no se pueden transmitir nuevos. Recién cuando llega el noveno ACK duplicado, la ventana de congestión alcanza el valor $cwnd = 16\ segmentos$ pero, como todavía existen 15 pendientes, sólo se puede transmitir un segmento. La situación persiste hasta el catorceavo ACK duplicado, lográndose transmitir hasta 6 segmentos.

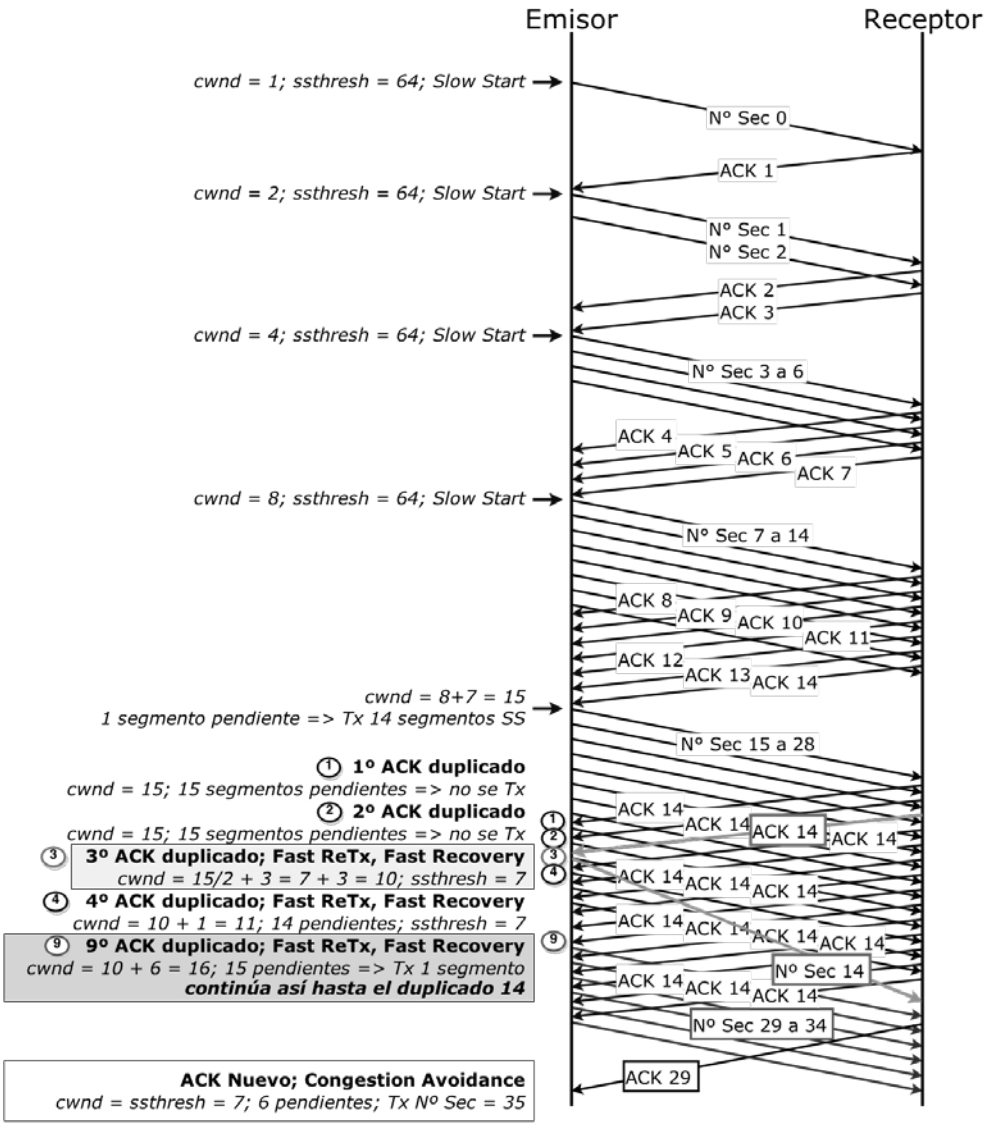


Figura 13.3 – Recepción de ACK duplicados.

En la Fig. 13.4, se presenta la evolución de la ventana de congestión para el ejemplo planteado.

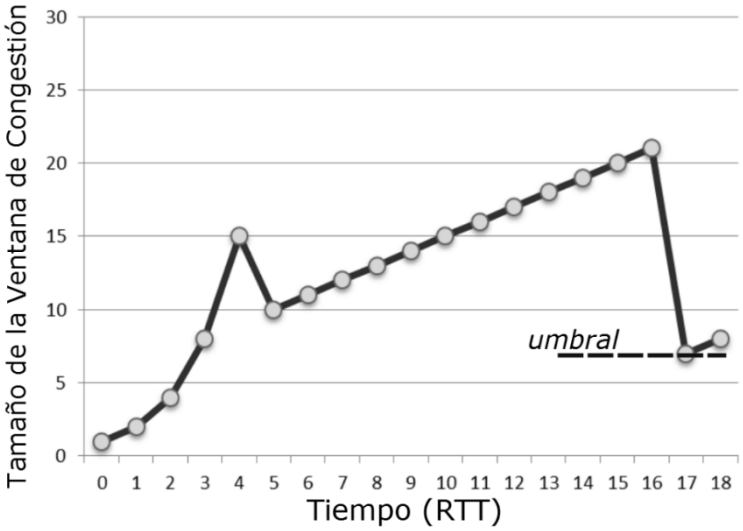


Figura 12.4 – Evolución de ventana de congestión con recepción de ACK duplicados.

En la Fig. 13.5 se completa el intercambio de segmentos para el ejemplo presentado, considerando la evolución de la ventana de congestión, a partir de la llegada de un ACK nuevo.

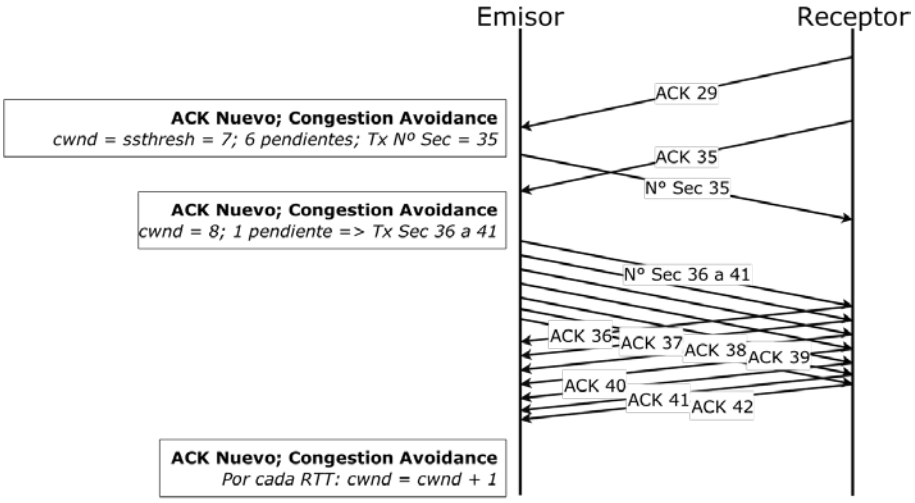


Figura 13.5 - Luego de la recepción de ACK duplicados

Es importante aclarar que el protocolo considera un segmento como ACK duplicado cuando el receptor tiene datos pendientes de reconocimiento, el segmento ACK no carga datos ni lleva las banderas de SYN y FIN en alto. Además, el Número de ACK del duplicado debe ser igual al mayor que se haya recibido en esa conexión y la ventana anunciada por el mismo debe ser la misma que la del último ACK recibido.

Complementariamente a lo explicado en este apartado, si TCP utiliza la opción SACK, puede tomar ventaja de información extra ante la recepción de un ACK duplicado, como se verá más adelante.

13.3 Modificación New Reno al Algoritmo Fast Recovery

La primera implementación de un Sistema Operativo que incluyó los algoritmos para arranque lento y evitar la congestión fue la versión 4.2 de Unix, también conocida como BSD UNIX o Tahoe. Posteriormente, el mecanismo *fast recovery* fue incluido en la versión 4.3 BSD Reno, sentando así las bases para la implementación estándar de TCP.

El reconocimiento de potenciales problemas en el caso de pérdida de múltiples paquetes cuando se invocan *fast retransmit* y *fast recovery*, llevó a considerar la utilización de la opción SACK para poder realizar retransmisiones de manera más inteligente. Para aquellas implementaciones que no usan la opción o deben comunicarse con otras que no la soportan, la RFC 5681 reconoce la existencia de un mecanismo conocido como ACK parcial, descripto en la RFC 3782, que, aunque no se encuentra estandarizado, implícitamente está permitido, ya que sigue las reglas generales de los cuatro algoritmos descriptos.

El problema de *fast recovery* es que, cuando se pierden varios paquetes de una ventana, una vez que el paquete retransmitido por *fast retransmit* es exitosamente recibido y reconocido por un nuevo ACK, el valor de *cwnd* vuelve a bajar, antes de haber podido retransmitir todos los paquetes perdidos. A este segmento ACK se lo conoce como ACK parcial. La consecuencia es que, al reducir la ventana de congestión, TCP puede quedar ocioso, causando el disparo de mecanismos de retransmisión por RTO, reduciéndose drásticamente la velocidad, al invocar el mecanismo *Slow Start*.

La modificación conocida como Algoritmo New Reno se ocupa de estos casos. New Reno introduce una nueva variable $recover$, que almacena el Número de Secuencia más alto de la última ventana transmitida. Este valor se contrasta con el Número de ACK recibido, de tal manera que se abandona el crecimiento de *cwnd*, propio de *fast recovery*, sólo cuando este último valor es mayor que el de la variable. Se permite de este modo una respuesta de retransmisión a los denominados segmentos ACK parciales, mejorando notablemente la eficiencia del protocolo.

13.4 Control de Congestión con ACK Selectivo – SACK

La eficiencia del reconocimiento acumulativo de paquetes conduce a situaciones de pobre performance cuando TCP sufre la pérdida de varios paquetes por ventana de transmisión. El transmisor se ve forzado a esperar un RTT para acertar cuáles segmentos se han perdido, pudiendo llegar a retransmitir segmentos recibidos correctamente. Cuando se pierden varios segmentos, la situación se puede complicar, a tal punto de perder el sincronismo que ofrece la propia llegada de segmentos ACK.

La RFC 2018 describe un mecanismo denominado ACK Selectivo (SACK, Selective ACK) que se integra al protocolo original para ayudar a mejorar las limitaciones mencionadas. Con esta nueva opción, el receptor puede especificar al transmisor los segmentos que ha recibido fuera de orden, para que el transmisor sólo proceda a re-enviar aquellos segmentos que se han perdido.

Los extremos de una conexión TCP informan que son capaces de trabajar con ACK Selectivo por anuncio de la opción SACK permitido en el segmento SYN de inicio o en su respuesta, que lleva las banderas SYN y ACK levantadas. A partir de allí, si ambos extremos son capaces, se pueden anunciar los bloques de datos llegados fuera de orden como una opción TCP.

Cada bloque se anuncia por medio de dos Números de Secuencia de $32\ bits$, que representan el bloque de datos recibidos correctamente, pero fuera de orden. El primer Número de Secuencia se refiere al borde izquierdo del bloque (LE, Left Edge), es decir el primer byte recibido correctamente. El segundo Número de Secuencia es el borde derecho (RE, Right Edge), que se refiere al siguiente número de byte que se espera recibir, interpretándose que se recibió hasta el previo correctamente.

Tal como se presenta en la Fig. 13.6, cada bloque ocupa un espacio de $8\ bytes$. A su vez la opción incluye un campo Tipo y otro de Longitud, ambos de $1\ byte$. Como el campo de opciones TCP no puede ser mayor de $40\ bytes$, la opción no puede cargar más de 4 bloques. Si esta opción se utilizara en conjunto con otra, por ejemplo la opción de Sello de Tiempo, sólo podrían cargarse tres bloques.

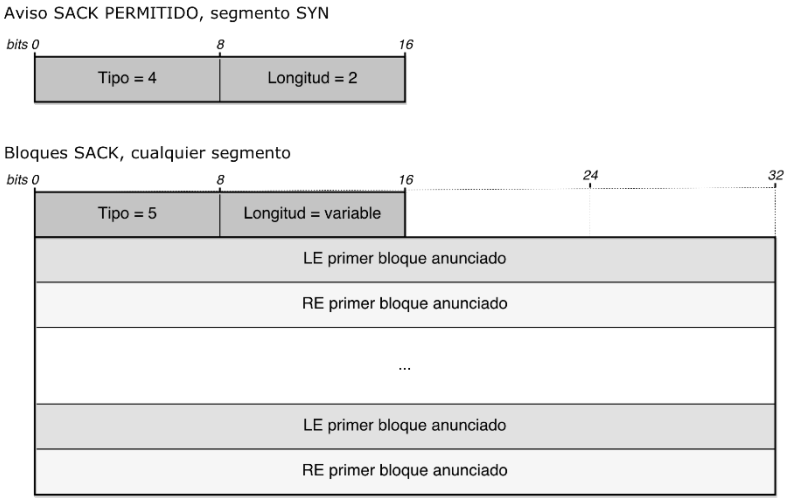


Figura 13.6 - Opción SACK. Anuncio y Utilización

A continuación se detalla el intercambio de opciones en el inicio de la conexión entre un cliente y un servidor web. Ambos trabajan con las opciones SACK y escalamiento de ventana.

Transmission Control Protocol, Source port: 49265, Destination port: 80

Sequence number: 0 (relative sequence number)
Header length: 32 bytes
Flags: 0x0002 (SYN)
0... .... = Congestion Window Reduced (CWR): Not set
.0.. .... = ECN-Echo: Not set
..0. .... = Urgent: Not set
...0 .... = Acknowledgment: Not set
.... 0... = Push: Not set
.... .0.. = Reset: Not set
.... ..1. = Syn: Set
....0 = Fin: Not set
Window size: 2097152 (scaled)
Checksum: 0x1563
Options:
TCP MSS Option: True
Maximum segment size: 1460 bytes
NOP
TCP Window Scale Option: True
Window scale: 8 (multiply by 256)
NOP
NOP
SACK permitted

Transmission Control Protocol, Source port: 80, Destination port: 49265

Sequence number: 0 (relative sequence number)
Acknowledgement number: 1 (relative ack number)
Header length: 32 bytes
Flags: 0x0012 (SYN, ACK)
0... .... = Congestion Window Reduced (CWR): Not set
.0.. .... = ECN-Echo: Not set
..0. .... = Urgent: Not set
...1 .... = Acknowledgment: Set
.... 0... = Push: Not set
.... .0.. = Reset: Not set
.... ..1. = Syn: Set
....0 = Fin: Not set
Window size: 373760 (scaled)
Checksum: 0x5035
Options: (12 bytes)
TCP MSS Option: True
Maximum segment size: 1460 bytes
NOP
NOP
SACK permitted
NOP
TCP Window Scale Option: True
Window scale: 6 (multiply by 64)

El receptor utilizará la opción SACK toda vez que almacene segmentos correctos fuera de orden en el buffer de recepción. Colocará como primer bloque los bordes izquierdo LE y derecho RE del segmento más recientemente recibido que no pueda generar un nuevo número de ACK por llegar fuera de orden. Así se sigue el criterio de informar con prioridad el último evento sucedido, debido al escaso espacio disponible en el campo de opciones. También se deberían incluir los últimos bloques reportados que no sean internos al primero anunciado, para resguardarse de posibles pérdidas de segmentos con la opción SACK.

El transmisor podría marcar los segmentos en la cola de retransmisión que se encuentren dentro de los límites de un bloque anunciado, para no sujetarlos a retransmisiones posteriores. Así, cualquier segmento que no hubiese sido marcado y se encuentre por debajo, en número de secuencia, con respecto al máximo segmento marcado, quedaría sujeto a retransmisión, generalmente por recepción de ACK duplicado.

Luego de un *timeout*, se deberían desmarcar todos los bloques en la cola de retransmisión, debiéndose retransmitir el segmento sobre el borde izquierdo de la ventana. El resto permanece en la cola, al menos hasta la llegada de un ACK que los cubra. El protocolo trabaja de esta forma porque considera el mecanismo de ACK selectivo como consultivo, pudiendo suceder que el extremo receptor haya anunciado un bloque, para luego arrepentirse. Esta consideración resulta en bloques almacenados en el buffer de retransmisión, a pesar de haber

sido reconocidos por el mecanismo selectivo, aunque formalmente sólo se pueden retirar cuando un número de ACK normal los cubra.

La Fig. 13.7 presenta un ejemplo de intercambio de segmentos con la opción ya habilitada. Se trata del envío de 8 segmentos de 500 $bytes$, pero llegan sólo los primeros cuatro. Esta situación no dispara la opción SACK.

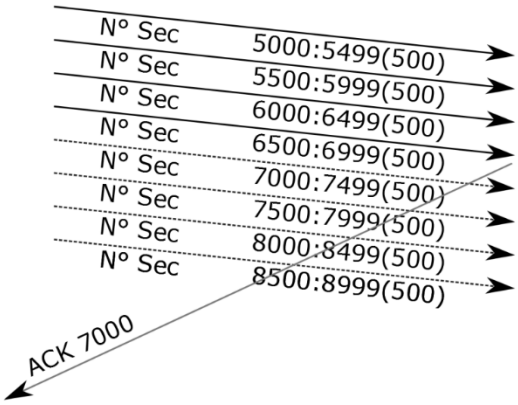


Figura 13.7 – Opción SACK no se dispara.

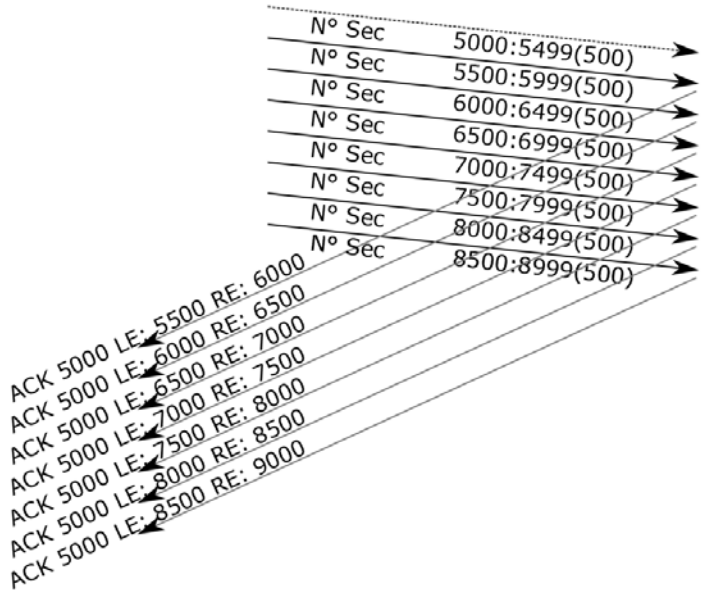


Figura 13.8 – Opción SACK, actualización de bloques consecutivos.

En el ejemplo de la Fig. 13.8, se transmiten los mismos 8 $segmentos$ del ejemplo previo, pero se pierde el primero. En este caso, la llegada del segundo segmento dispara una notificación *SACK* para los 500 $bytes$ recibidos

correctamente. La recepción de los segmentos sucesivos, genera nuevas notificaciones en las que se modifica el borde derecho, de acuerdo al segmento recibido, anunciándose un único bloque cada vez. Es de destacar la generación de un ACK duplicado por cada recepción correcta, aunque fuera de orden.

En el caso de la Fig. 13.9, el primer segmento llega bien, disparando un número de ACK 5500, que apunta al segundo segmento. Este se pierde, pero llega el tercer segmento, que genera un ACK duplicado que transporta el anuncio de la recepción del bloque comprendido entre los bytes 6000 y 6499, de ahí que el borde derecho se numere como 6500. Luego, se pierde el cuarto segmento, pero arriba el quinto, disparando otro ACK duplicado, que carga el anuncio de los dos bloques recibidos en el orden explicado previamente. Primero se notifica el último que llegó correctamente ($LE = 7000,\ RE = 7500$) y luego el de la anterior opción SACK ($LE = 6000,\ RE = 6500$). El sexto segmento también se pierde, pero el séptimo llega correctamente, disparando otro ACK duplicado y ordenando los bloques como se presenta en la figura: ($LE = 8000,\ RE = 8500$), ($LE = 7000,\ RE = 7500$) y ($LE = 6000,\ RE = 6500$).

Evidentemente, el tercer ACK duplicado dispara una retransmisión, pero antes llega el cuarto segmento retrasado. Este último dispara otro ACK duplicado, con la opción SACK que anuncia los siguientes bloques: primero el último en actualizarse ($LE = 6000,\ RE = 7500$) y luego el primero del SACK previo ($LE = 8000,\ RE = 8500$). Cuando llega la retransmisión del segundo segmento, avanza el número de *ACK* hasta 7500, debiendo anunciarse aún el bloque ($LE = 8000,\ RE = 8500$), pues todavía no ha sido ordenado.

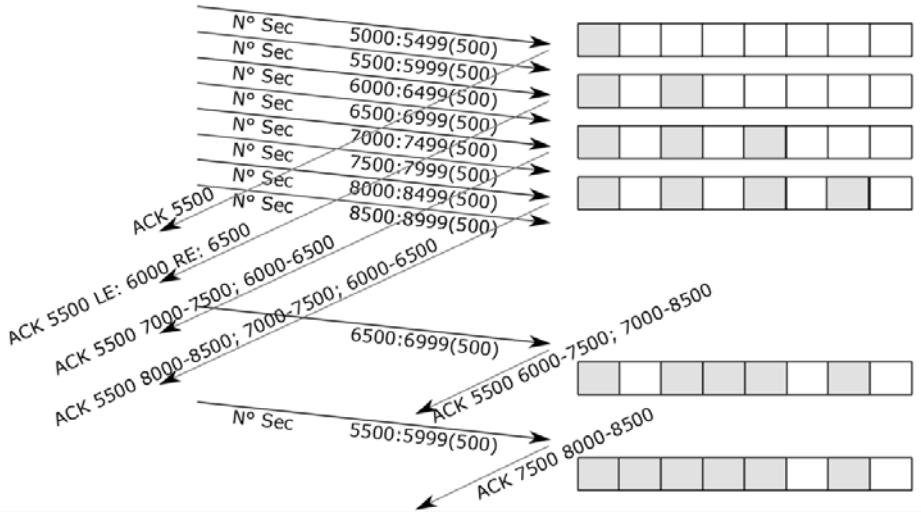


Figura 13.9 – Opción SACK, actualización de bloques no consecutivos.

En la RFC 2018 se define la opción SACK, presentando ejemplos de su utilización, pero no se especifica su uso cuando se reciben segmentos duplicados. La RFC 2883 se ocupa de estos casos, usando la nomenclatura D-SACK para los bloques SACK que reportan un segmento duplicado. Se trata de una extensión

compatible con la anterior, es decir que, para implementarla, sólo es necesario haber negociado la opción SACK. Cuando se usa D-SACK, el primer bloque de la opción debería especificar los números de secuencia del segmento duplicado que dispara el ACK. Si el duplicado forma parte de un bloque mayor de datos no contiguos que se encuentran en el buffer de recepción, el bloque SACK siguiente debería especificarlo. Se pueden anunciar luego otros bloques SACK.

Por otra parte, la RFC 3517 presenta un algoritmo de recuperación de pérdidas para TCP que se basa en el uso de la opción SACK. Durante la fase de *fast recovery*, SACK mantiene una nueva variable, denominada $pipe$, que almacena el número estimado de paquetes pendientes en tránsito. El transmisor sólo envía datos nuevos o retransmitidos cuando se verifica $pipe < cwnd$, incrementando la nueva variable en 1 cada vez que se transmite o retransmite un segmento. La variable desciende en 1 cada vez que arriba un ACK duplicado con la opción SACK anunciando nuevos datos recibidos en el otro extremo.

Al utilizar la nueva variable $pipe$ se puede separar la decisión de cuándo retransmitir un segmento respecto de cuál segmento retransmitir. Este pequeño cambio permite responder mejor a la situación de pérdidas de más de un segmento de la ventana.

13.5 Extensiones TCP para redes de alta velocidad

A medida que las redes fueron mejorando las velocidades de transmisión, por encima de los 10 Mbps, se empezaron a notar ciertos problemas de falta de adaptación de TCP. La performance del protocolo depende de la velocidad de transferencia y del retardo ida y vuelta asociado a la conexión, tomando su producto ($r_b \; x \; RTT$) como medida de la cantidad óptima de datos que se pueden almacenar en el buffer de transmisión para obtener el mejor rendimiento sobre la conexión TCP. Cuando este producto aumenta, algunos de los campos del encabezado TCP empiezan a resultar insuficientes para la labor para la que fueron concebidos.

Un ejemplo lo constituye un enlace transmitiendo datos a $100\ Mbps$ hacia otro país, con un retardo en el orden de los $10\ mseg$, generando un producto de $r_b \; x \; RTT = 100\ Mbps \; x \; 10\ mseg = 1.000.000\ bits = 125.000\ bytes$, bastante superior al tamaño máximo de ventana win que es posible anunciar, con $16\ bits$ en el encabezado destinado a ese efecto. Para lidiar con esta limitación, se define la nueva opción de escalamiento de ventana. Se trata de un factor que se anuncia en el inicio, que se debe multiplicar con el valor anunciado por el otro extremo, para obtener el valor escalado de la ventana.

Otro problema que padecen las conexiones TCP sobre redes de alta velocidad es la recuperación frente a las pérdidas de segmentos. Cuando sucede un *timeout*, el protocolo debe comenzar desde el principio, con *Slow Start*. Este mecanismo de recuperación se mejoró bastante con la implementación de *fast retransmit* y *fast recovery*, que permite la recuperación de un segmento por ventana sin tener que vaciar la conexión. Lo cierto es que, al aumentar el tamaño de la ventana, también aumenta la posibilidad de perder más de un paquete,

disminuyendo la probabilidad de una recuperación rápida y bajando, de este modo, la eficiencia de la conexión. La mejora introducida por el reconocimiento de bloques SACK se orienta a la posibilidad de manejar los casos de más de un paquete perdido por ventana de transmisión.

Por último, la estrategia de *timeout* y retransmisión se apoya sobre mediciones dinámicas lo más exactas posibles del RTT. El protocolo introduce una nueva opción para asegurar mediciones más precisas. Se trata de la opción Sello de Tiempo, que también se utiliza para apoyar el mecanismo de Numeración de Secuencia, cuando las redes son tan rápidas que es posible utilizar toda la numeración y dar la vuelta (volver a empezar) sobre una misma conexión.

Excepto por la opción SACK, todas las nuevas opciones mencionadas, se encuentran definidas en la RFC 1323.

13.5.1 Opción Escalamiento de Ventana

El escalamiento de la ventana anunciada en el campo win de $16\ bits$, se realiza a través del anuncio de un factor de escala en el campo de opciones del segmento SYN, que permite escalar el valor anunciado a un número de $32\ bits$. De este modo, se fija en el inicio el valor del factor en cada dirección de la conexión y permanece fijo durante toda la duración de la misma. El valor máximo sigue limitado por el tamaño máximo del buffer de recepción.

En los segmentos SYN, la opción ocupa $3\ bytes$, sirviendo para informar que la implementación puede anunciar y recibir valores escalados de la ventana y comunicar el factor de escala que el otro extremo debe aplicar al campo win. El factor de escala se limita a una potencia de 2 y se codifica de manera logarítmica, para poder ser implementado por operaciones binarias de corrimiento. La Fig. 13.10 presenta la opción en el segmento SYN. El valor $shift.cnt$ es el del factor de escala.

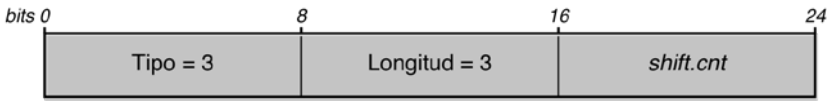


Figura 13.10 – Opción Escalamiento de Ventana.

Para poder aplicar posteriormente el escalamiento, ambos extremos deben enviar la opción en el segmento *SYN*. Una vez intercambiados los factores, el protocolo TCP que envía la opción toma el verdadero valor de la ventana y lo corre hacia la derecha (divide) tantos bits como indica $shift.count$, copiando el resultado en el campo win de la cabecera TCP.

La Fig. 13.11 presenta la utilización de esta nueva opción en el inicio y durante el transcurso de una conexión TCP. Como se puede apreciar, en el segmento SYN, el cliente le anuncia al servidor su capacidad para escalar la

ventana, comunicándole el factor que usará para tal efecto, en este caso R. Lo propio hace el Servidor con su factor de escalamiento S, en el segmento con las banderas SYN y ACK levantadas.

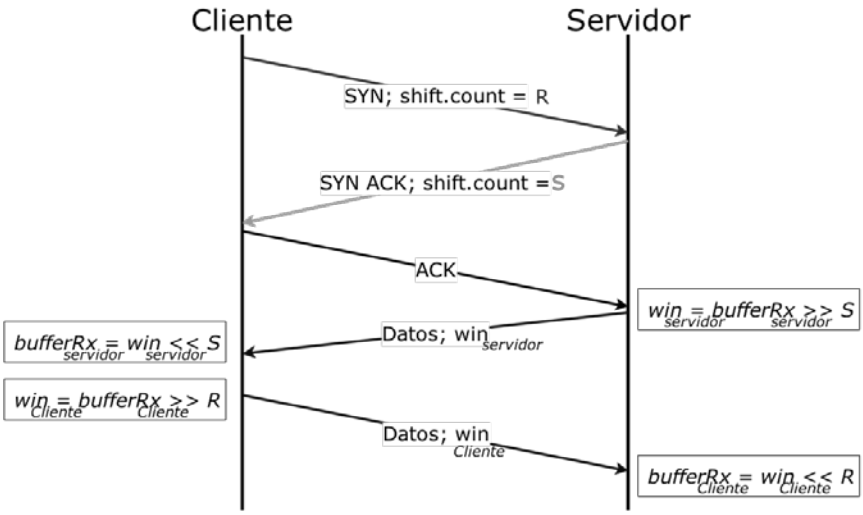


Figura 13.11 – Escalamiento de ventanas.

A partir de ese momento, cada vez que un extremo anuncie su propia ventana win, tomará el verdadero tamaño del buffer de recepción, $buffer_{Rx}$, lo dividirá por el factor que haya anunciado y copiará el resultado en el campo win de la cabecera TCP del segmento transmitido. El otro extremo, al recibir este valor, debe multiplicarlo por el factor que le fuera anunciado en el inicio, para obtener el valor verdadero del tamaño del buffer de recepción.

Por otra parte, TCP determina si un segmento es nuevo o viejo testeando si su número de secuencia se encuentra dentro de los 2^{31} $bytes$ del borde izquierdo de la ventana de recepción. Si no lo está, lo descarta por viejo. Para asegurarse que los datos nuevos no se puedan confundir con segmentos viejos, el borde izquierdo de la ventana de transmisión debe encontrarse, al menos, 2^{31} $bytes$ separado del borde derecho de la ventana de recepción. La misma condición debe sostenerse entre el borde derecho de la ventana del transmisor y el borde izquierdo de la ventana del receptor. Dado que los bordes derecho e izquierdo, ya sea de la ventana transmisora o receptora, difieren en el tamaño de la ventana y, dado que ambas ventanas, de transmisor y receptor, pueden estar fuera de fase en, a lo sumo, el tamaño de la ventana, la restricción se traduce en la expresión $2\ x\ win_{máx} < 2^{31}$, que también se puede escribir como $win_{máx} < 2^{30}$.

Cuando se usa la opción de escalamiento, con factor de escalamiento S, el tamaño máximo de la ventana será $win_{máx} < (2^{16} - 1)\ x\ 2^S < 2^{30}$. Esta condición impone una restricción sobre el factor de escalamiento: $S \leq 14$. Si el

factor anunciado fuera mayor de 14, se registra como un error y se adopta 14 como factor.

Es apropiado aclarar que, la ventana de congestión $cwnd$ no es afectada por factor de escalamiento.

13.5.2 Medición de RTT con opción Sello de Tiempo

La estimación precisa y dinámica del RTT de una conexión es necesaria para adaptarse a las condiciones de un tráfico cambiante, y así evitar situaciones de ausencia de estabilidad, conocidas como colapso por congestión.

La medición del tiempo de ida y vuelta sobre un segmento por ventana de transmisión, como trabajan muchas implementaciones antiguas, resulta adecuada para tamaños de ventana pequeños, pero inaceptable en caso de redes de alta velocidad. Por ejemplo, una ventana de 8 segmentos, con una frecuencia de muestreo de un segmento por ventana, generaría una estimación tolerable en términos de la propia conexión. Sin embargo, cuando el tamaño de la ventana crece, conteniendo decenas o cerca de una centena de segmentos, la estimación no es precisa y puede generar muchos errores, traducidos en retransmisiones espurias. Por otra parte, Karn demostró que no es posible medir estimaciones válidas en el caso de retransmisiones.

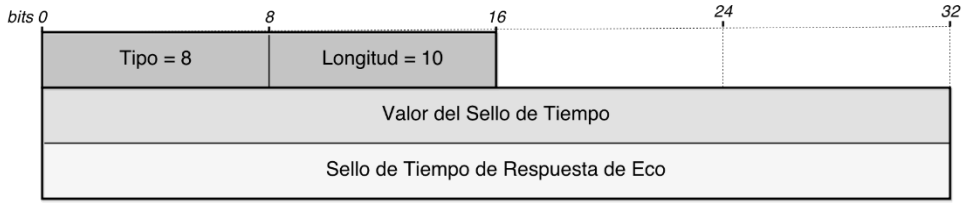


Figura 13.12 – Opción Sello de Tiempo

La solución propuesta para todos estos problemas usa el campo de opciones, cargándolo con un Sello de Tiempo. El transmisor de un segmento carga su propio sello de tiempo en el campo de opciones y el receptor lo copia, como si fuera un eco, en el segmento de ACK que cubre su número de secuencia. Al recibir el ACK, para obtener la medición de RTT, el transmisor simplemente resta entre sí los tiempos de salida del sello y de llegada del eco.

La Fig. 13.12 presenta el formato de la opción, que ocupa 10 $bytes$ ya que cada sello de tiempo se expresa como un valor de 32 $bits$.

El campo Valor del Sello de Tiempo $TSval$ contiene el valor del reloj del protocolo que transmite la opción. El campo Sello de Tiempo de Respuesta de Eco $TSecr$, que es válido sólo si la bandera de ACK se encuentra en alto, carga el valor de $TSval$ transmitido por el otro extremo. La utilización de la opción es anunciada por ambos extremos en el inicio de la conexión. Para que funcione

correctamente, el valor de sello de tiempo que se escribe en el campo $TSval$ se obtiene de un reloj virtual cuyo valor debe ser al menos proporcional al tiempo real del sistema.

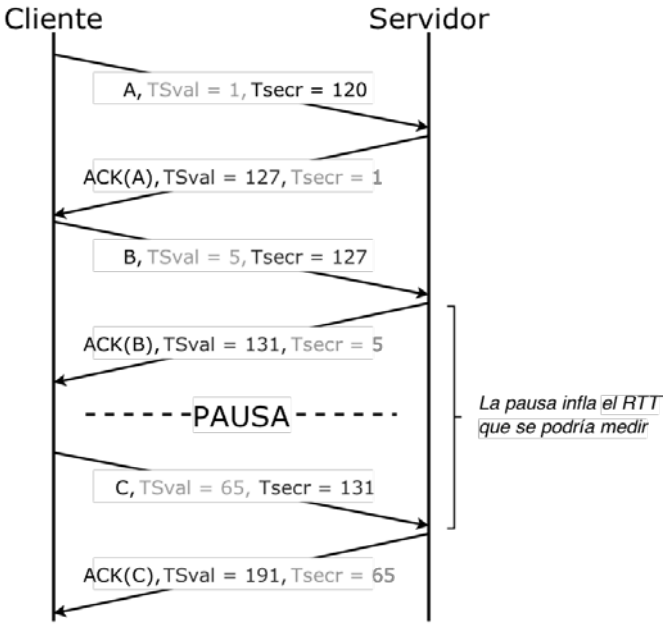


Figura 13.13 - Intercambio de Sello de Tiempo.

La Fig. 13. 13 ilustra el intercambio de sellos de tiempo en una conexión donde sólo un extremo se encuentra transmitiendo datos, no hay pérdidas y los bloques enviados sucesivamente se numeran simbólicamente como A, B y C. Los valores que acompañan el anuncio de ACK representan los correspondientes reconocimientos acumulativos.

Como se puede apreciar, el extremo transmisor envía segmentos en los instantes de tiempo 1, 5 y 65, obtenidos a partir del reloj virtual propio, que se copian en los respectivos campos $TSval$ de los segmentos transmitidos. El primer bloque A, es reconocido con un segmento ACK que carga un sello de tiempo $TSval$ propio del extremo receptor, y otro sello de tiempo $TSecr$, que resulta ser la copia del sello de tiempo del segmento recibido. Una situación similar sucede con los reconocimientos de los segmentos B y C.

El intercambio presenta una pausa de 60 unidades de tiempo, por ausencia de segmentos a transmitir. Si el extremo receptor midiera el tiempo transcurrido entre $TSval = 131$ y $TSecr = 131$, no obtendría una medición justa del RTT. Esta situación sucede en los casos de flujo de datos en un único sentido de la conexión. Se corrige imponiendo una regla muy sencilla: cuando se recibe un valor $TSecr$ sólo se utiliza para actualizar la medición de RTT si el segmento que lo carga transporta nuevos datos, avanzando de este modo el borde

izquierdo de la ventana transmisora. Este no es el caso en el ejemplo presentado, ya que el extremo de la derecha no está transmitiendo datos.

Existen casos especiales, para los cuales también se imponen reglas. Por ejemplo, si se recibiera más de un sello de tiempo antes de enviar un ACK, el receptor debería elegir a cuál de los sellos de tiempo hacer eco, ignorando el resto para minimizar la cantidad de información de estado almacenada sobre la conexión.

En este sentido, existen tres situaciones a ser consideradas:

- **ACK retrasados**: cuando TCP edita un ACK acumulativo sobre varios segmentos, debe decidir cuál valor $TSval$ colocar en el campo $TSecr$. Para que la medición en el otro extremo sea lo más cercana a la realidad posible, se debería elegir el valor $TSval$ del primer segmento recibido que todavía no se haya reconocido. De este modo se permite al extremo medidor considerar el tiempo de procesamiento adicional debido al retraso del ACK.

- **Segmento perdido**: en este caso el receptor realiza el reconocimiento de los segmentos fuera de orden, colaborando con el algoritmo de retransmisión rápida que se disparará en el transmisor. Esta situación es típica de un estado de congestión, en la que del lado transmisor es preferible sobrestimar el valor de RTT. Por este motivo, el ACK de un segmento fuera de orden debería copiar el sello de tiempo del último segmento recibido que haya avanzado el borde izquierdo de la ventana. La misma regla se aplica en el caso de segmentos re-ordenados.

- **Llegada de segmentos que llenan un hueco**: la llegada de estos segmentos representa la medición más actual del estado de la red. Por otro lado, un valor de RTT calculado en base al sello de tiempo de un segmento previo, muy probablemente incluiría el *timeout* asociado a la retransmisión del extremo transmisor, sesgando de manera incorrecta la estimación. Por este motivo, se realiza el eco del sello de tiempo del último segmento recibido, el que llena el hueco.

La Fig. 13.14 presenta el primero de los casos especiales abordados. Se reciben tres segmentos al mismo tiempo, pero el segmento ACK carga en el campo de eco el valor de sello de tiempo correspondiente al de menor número de secuencia.

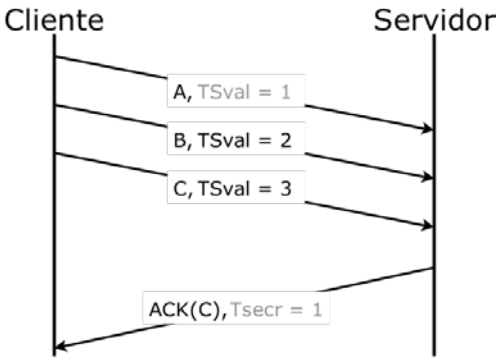


Figura 13.14 – Sello de tiempo para ACK retrasado

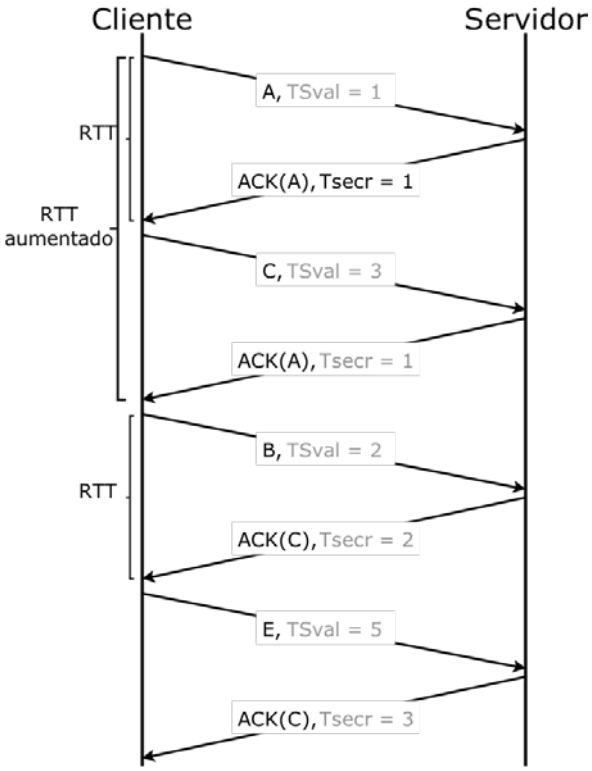


Figura 13.15 – Sello de tiempo para segmentos fuera de orden.

La Fig. 13.15 presenta un ejemplo de los dos últimos casos especiales. La recepción del primer bloque dispara un ACK que transporta el eco correspondiente. Posteriormente se recibe un bloque desordenado, aplicándose la regla de referirse al $TSval$ del último segmento recibido correctamente y en orden. La aplicación de este criterio genera un valor aumentado del RTT, pero

esta corrección es justa, puesto que la red se encuentra entregando segmentos fuera de orden, signo de un estado de congestión. La situación se compensa con la llegada del segmento faltante, como se aprecia en la figura, aunque no se ha dibujado la misma en escala respecto al tiempo.

13.5.3 Protección contra la repetición del Número de Secuencia - PAWS

El uso accidental los mismos números de secuencia en una conexión puede conducir a errores serios. Un ejemplo podría referirse al caso de un segmento de datos viejo duplicado. Se trataría de un segmento viejo (de la misma conexión) que ha sido demorado por la red, siendo entregado luego, de tal manera que su Número de Secuencia cae dentro de la ventana actual de datos. No habría forma de detectar este error, generándose, en consecuencia, una posible corrupción inadvertida de los datos. De modo parecido, la recepción de un segmento ACK viejo duplicado en el transmisor podría también generar errores, muy probablemente bloqueando la conexión, forzando un RST en la misma.

Parte de la confiabilidad asociada a TCP depende de la existencia de un límite en la vida de los segmentos. El parámetro MSL pone un límite a este tiempo, ya que cualquier número de secuencia es finito y, eventualmente, podría ser reutilizado. En la práctica, MSL se refuerza a nivel IP con el campo TTL.

En general, la duplicación de Números de Secuencia se puede presentar en dos escenarios:

- La numeración se agota y comienza nuevamente (da la vuelta), volviendo a empezar sobre la misma conexión. Un problema podría suceder si un segmento se demorara en las colas de los *routers* por un tiempo comparable a lo que se tarda en utilizar los $32\ bits$ del número de secuencia. Poner un límite a esta situación exige un mecanismo aparte del de MSL, en el caso de grandes velocidades de transferencia.

- La conexión termina y se restablece inmediatamente sobre el mismo par de sockets. Un segmento retrasado de la vieja conexión puede caer dentro de la ventana actual de la re-encarnación y ser aceptado como válido. Este caso se maneja con MSL, evitando el uso de un mismo par de sockets durante un tiempo equivalente al doble de MSL.

Asociado al consumo completo del campo Número de Secuencia, se puede definir un tiempo. Para transmitir toda la numeración de secuencia, se precisa un tiempo, presentado en la siguiente ecuación:

$$T'_{wrap} = {2^{32} x\ 8}/{r_b} > 2MSL \tag{13.8}$$

La desigualdad es la condición que se debe verificar para una operación libre de errores. Se denomina tiempo de dar la vuelta o tiempo de agotamiento del número de secuencia a la expresión:

$$T_{wrap} = {2^{31} x\ 8}/{r_b} \tag{12.9}$$

Es decir que T_{wrap} debería ser superior a los 2 $minutos$ que el protocolo establece para MSL. Por ejemplo, en una red Ethernet rápida, el tiempo de agotamiento sería levemente superior a MSL:

$${2^{31} x\ 8}/{r_b} = {2^{31} x\ 8}/{100\ Mbps} = \ 171.8\ seg \tag{12.10}$$

En una red Gigabit, el tiempo sería sólo de $17,18\ seg$, inferior a MSL.

Es claro que el problema se presenta en enlaces de alta velocidad, pero también es cierto que existe una velocidad efectiva de transferencia que queda establecida por el tamaño de la ventana anunciada. En este sentido, se podría expresar una velocidad efectiva de transferencia limitada por el valor máximo de la ventana y el tiempo de ida y vuelta para la conexión:

$$r_{b(efectiva)} = \ {2^{16}}/{RTT} \tag{12.11}$$

Con esta limitación, si el enlace presenta un valor alto de RTT, T_{wrap} no llega a ser menor que MSL. Por ejemplo, una conexión cuyo RTT se encuentre en el orden de los $10\ mseg$, limita la velocidad efectiva de transferencia a:

$$r_{b(efectiva)} = \ {2^{16}}/{10\ mseg} = 6.55\ Mbps \tag{12.12}$$

Asociándose a un valor más real de T_{wrap}, de $43.7\ min$.

Diferente resulta el caso de utilización de una ventana escalada, por ejemplo en 8, pues la velocidad efectiva se multiplica por este número, $r_{b(efectiva)} = 52.4\ Mbps$, resultando $T_{wrap} = 5.46\ min$, un valor más cercano a MSL.

Una manera de solucionar el problema, sería aumentar el número de bits del campo Número de Secuencia en el encabezado TCP. Esta solución se puede realizar a través de una opción que incluye un Sello de Tiempo, que permite agregar un valor de tiempo $TSval$, de $4\ bytes$, a cada segmento transmitido, inclusive los segmentos ACK. La idea es poder descartar un segmento que se

interpreta como un duplicado viejo si éste se recibe con un Sello de Tiempo $TSval$ menor que el último recientemente recibido en esa conexión.

El mecanismo se conoce con la sigla PAWS (Protection Against Wrapped Sequences numbers). PAWS exige que el valor del Sello sea monótonamente creciente y que la elección del Sello almacenado para la comparación lo garantice. Por ejemplo, se puede optar por almacenar en una variable $TS.Recent$ el Sello del último segmento que haya avanzado el borde izquierdo de la ventana de recepción.

Así, el sello de tiempo se usa como una extensión de 32 $bits$ del número de secuencia y, debido a que crece monótonamente, permite distinguir entre segmentos viejos y nuevos con el mismo número de secuencia.

Bibliografía

1. RFC 793 “Transmission Control Protocol ”, September 1981. http://tools.ietf.org/html/rfc793
2. RFC 896 “Congestion Control in IP/TCP Internetworks”, January 1984. http://tools.ietf.org/html/rfc896
3. RFC 1122 “Requirements for Internet Hosts -- Communication Layers”, October 1989. http://tools.ietf.org/html/rfc1122
4. RFC 1323 “TCP Extensions for High Performance”, May 1992. https://www.ietf.org/rfc/rfc1323.txt
5. RFC 2001 “TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms ”, January 1997. http://tools.ietf.org/html/rfc2001
6. RFC 2018 “TCP Selective Acknowledgment Options ”, October 1996. http://tools.ietf.org/html/rfc2018
7. RFC 2581 “TCP Congestion Control”, April 1999. http://tools.ietf.org/html/rfc2581
8. RFC 2883 “An Extension to the Selective Acknowledgement (SACK) Option for TCP ”, July 2000. http://tools.ietf.org/html/rfc2883
9. RFC 3465 “TCP Congestion Control with Appropriate Byte Counting (ABC) ”, February 2003. https://www.ietf.org/rfc/rfc3465.txt
10. RFC 3517 “A Conservative Selective Acknowledgment (SACK)-based Loss Recovery Algorithm for TCP”, April 2003. http://tools.ietf.org/html/rfc3517
11. RFC 3782 “The NewReno Modification to TCP's Fast Recovery Algorithm”, April 2004. http://tools.ietf.org/html/rfc3782
12. RFC 5681 “TCP Congestion Control”, September 2009. http://tools.ietf.org/html/rfc5681
13. RFC 6298 “Computing TCP´s Retransmission Timer”, June 2011. http://tools.ietf.org/html/rfc6298
14. Jacobson, Van, “Congestion Avoidance and Control”, CM SIGCOMM Computer Communication Review. ACM, 1988. p. 314-329.

15. Fall, Kevin, Floyd, Sally. "Simulation-based Comparisons of Tahoe, Reno and SACK TCP". Computer Communication. Review, July 1996. http://ee.lbl.gov/papers/sacks.pdf
16. Kozierok, Charles M., "The TCP/IP Guide". http://www.tcpipguide.com/free/t_toc.htm
17. Comer, Douglas, "Internetworking with TCP/IP: Principles, Protocols and Architecture v. 1". Pearson Education, 1995.
18. Stevens, W. Richard, "TCP/IP Illustrated, Vol. 1: The Protocols (Addison-Wesley Professional Computing Series) ". Addison-Wesley, 1993.

Problemas

1. Explique la evolución en el tiempo de las variables involucradas (ventana de congestión cwnd y el valor de umbral ssthresh) y grafique (hasta 25 RTT luego de haberse iniciado la conexión) el caso de una conexión TCP que arranca correctamente, para luego pasar por:
    a) En el 5º RTT sufre una situación de recepción de tres AKC`s duplicados.
2. b En el 15º RTT vuelve a sufrir una situación de congestión como la previa.
3. c) En el 21º RTT sufre un timeout.
4. Medir los valores de cwnd y ssthresh en unidades de 1Kbyte.

5. ¿Qué campos de la cabecera TCP están relacionados con el Control de Flujo y por qué? ¿Cuáles se relacionan con el Control de Congestión y por qué?

6. ¿Con qué campo de las opciones se puede realizar una medición eficiente de RTT y cómo se hace? De un ejemplo.

7. ¿Para qué sirve la nueva opción PAWS?

8. Con la nueva opción SACK ¿Dejaron de existir los ACK duplicados?


eudem