




UNIVERSIDAD DE GUADALAJARA



Este material no tiene costo alguno y es proporcionado al estudiante con fines educativos, para la crítica y la investigación respetando la reglamentación en derechos de autor.

El **uso indebido** es responsabilidad del usuario.

IPng (IP de *next generation*), propone que el campo de cada dirección tenga 128 bits (16 bytes, lo que resulta en 340 sextillones de posibles combinaciones) de manera que pueda solucionarse el principal problema que hoy presentan las redes IP: *el agotamiento de las direcciones*.

Las funciones más importantes que se están implementando en la actualidad son Multicast y RSVP:

- **Multicast:** las direcciones multicast (Clase D) están habilitadas para que un usuario de la red pueda enviar sus datagramas a un conjunto de usuarios que han sido configurados como miembros de un grupo multicast de varias subredes.

Un grupo multicast puede estar formado por cualquier conjunto de máquinas, sin restricción de sus localizaciones físicas o de su número. Además, una máquina puede formar parte de uno o más grupos multicast *en cualquier momento* y no tiene que pertenecer a un grupo para enviar mensajes a miembros de un grupo.

- **RSVP:** es un protocolo de reserva de recursos para aquellas aplicaciones que requieran un ancho de banda preestablecido y asegurado como las comunicaciones de voz y la videoconferencia. Está especialmente indicado para aplicaciones multimedia.

IPv6 es una evolución lógica de IPv4, que introduce mejoras sustanciales como es el aumento de la capacidad de direccionamiento, una característica esencial para soportar el gran aumento de dispositivos —humanos y máquinas entre sí— conectados a Internet, vía un terminal fijo o móvil, y soporte para las nuevas aplicaciones multimedia en tiempo real. Con esta nueva versión se elimina la necesidad de espacios privados de direcciones ya que la disponibilidad es tan abundante que lo hace innecesario.

Además, aporta capacidad para la autenticación de las transacciones de comercio electrónico y privacidad en las comunicaciones, garantizando la integridad de los datos y su confidencialidad. Permite ofrecer calidad de servicio, lo que es equivalente a velocidad y servicios diferenciados, ya que el formato del paquete IPv6 tiene un nuevo campo de identificación de flujo que se puede utilizar para este fin.

Otro aspecto muy importante, cuando se accede desde un terminal móvil, es que IPv6 permite las comunicaciones entre redes fijas y móviles, permitiendo la movilidad del terminal que mantiene su dirección original y genera una secundaria basada en su posición, con lo que la dirección original no se ve afectada.

• El foro IPv6

Como sucede con otras tantas tecnologías, se ha creado el Foro IPv6, un consorcio mundial formado por más de 100 empresas, cuyo objetivo es fomentar el uso de esta nueva versión, nuevas aplicaciones y soluciones globales, resolviendo problemas y compartiendo conocimientos y experiencias entre sus miembros. Su *home page* se encuentra en www.ipv6forum.com.

En conclusión, se puede decir que poco a poco la nueva versión del protocolo IP se irá imponiendo y es sólo una cuestión de tiempo decir cuándo la versión actual IPv4 se verá totalmente reemplazada por la nueva. No es un capricho, es una necesidad incuestionable que se va haciendo más urgente conforme el rango de direcciones IP actuales se va consumiendo.

5.3 Protocolos de transporte

La capa de Transporte es una capa intermedia entre los niveles orientados a la red (subred) y los orientados a las aplicaciones. Su misión es recoger los datos que provienen de la capa de Red, fraccionarlos adecuadamente (segmentación) y asegurarse de que lleguen correctamente a la dirección destino, esté o no en la misma subred que la fuente de datos.

Los dos protocolos principales de la capa de transporte son el UDP y el TCP. El primero ofrece una transferencia de mensajes no fiable y no orientada a conexión y el segundo, una transferencia fiable y orientada a conexión.

IPv6 extiende el rango de direcciones hasta 2^{128} , lo que aproximadamente es 3×10^{38}

5.3.1. Protocolo UDP

Un protocolo, no orientado a conexión, utilizado comúnmente con IP es el **UDP (Protocolo de Datagrama de Usuario)**. Más simple que el TCP ya que confía en un servicio de red seguro, por lo que las funciones de recuperación frente a errores y desorden no las posee. El UDP incluye toda la información en cada mensaje y se emplea para la gestión remota de la red y para servicios de acceso por nemónico. Soporta multicast.

El protocolo UDP proporciona una comunicación muy sencilla entre las aplicaciones de dos ordenadores. Al igual que el protocolo IP, UDP es:

- **No orientado a conexión.** No se establece una conexión previa con el otro extremo para transmitir un mensaje UDP. Los mensajes se envían sin más y éstos pueden duplicarse o llegar desordenados al destino.
- **No fiable.** Los mensajes UDP se pueden perder o llegar dañados.

El protocolo UDP proporciona una forma para que las aplicaciones envíen datagramas IP sin tener que establecer una conexión. Se describe en la RFC 768.

UDP utiliza el protocolo IP para transportar sus mensajes. Como vemos, no añade ninguna mejora en la calidad de la transferencia; aunque sí incorpora los puertos origen y destino en su formato de mensaje. Las aplicaciones (y no el protocolo UDP) deberán programarse teniendo en cuenta que la información puede no llegar de forma correcta.



Figura 5.3. Funcionamiento de UDP.

• Formato del mensaje UDP

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32																
Puerto UDP origen																Puerto UDP destino																															
Longitud mensaje UDP																Suma verificación UDP																															
Datos																																															
...																																															

Figura 5.4. Formato y campos de un mensaje UDP.

- **Puerto UDP de origen** (16 bits, opcional). Número de puerto de la máquina origen.
- **Puerto UDP de destino** (16 bits). Número de puerto de la máquina destino.
- **Longitud del mensaje UDP** (16 bits). Especifica la longitud medida en bytes del mensaje UDP incluyendo la cabecera. La longitud mínima es de 8 bytes.
- **Suma de verificación UDP** (16 bits, opcional). Suma de comprobación de errores del mensaje. Para su cálculo se utiliza una *pseudo-cabecera* que también incluye las direcciones IP origen y destino. Para conocer estos datos, el protocolo UDP debe interactuar con el protocolo IP.
- **Datos.** Aquí viajan los datos que se envían las aplicaciones. Los mismos datos que envía la aplicación origen son recibidos por la aplicación destino después de atravesar toda la Red de redes.

5.3.2. Protocolo TCP

El protocolo **TCP** (*Transmission Control Protocol*, protocolo de control de transmisión), está basado en IP que es no fiable y no orientado a conexión y, sin embargo, es:

- **Orientado a conexión.** Es necesario establecer una conexión previa entre las dos máquinas antes de poder transmitir ningún dato. A través de esta conexión, los datos llegarán siempre a la aplicación destino de forma ordenada y sin duplicados. Finalmente, es necesario cerrar la conexión.
- **Fiable.** La información que envía el emisor llega de forma correcta al destino.

El protocolo TCP permite una comunicación fiable entre dos aplicaciones, con independencia del hardware y el software. De esta manera, las aplicaciones que lo utilicen no tienen que preocuparse de la integridad de la información: *dan por hecho que todo lo que reciben es correcto.*

La primera definición del protocolo TCP/IP la realizaron Kerf y Kahn en 1974. IP cubre la capa de red y TCP la de transporte.

El flujo de datos entre una aplicación y otra viajan por un *circuito virtual*. Sabemos que los datagramas IP pueden seguir rutas distintas, dependiendo del estado de los enrutadores intermedios, para llegar a un mismo sitio. Esto significa que los datagramas IP que transportan los mensajes siguen rutas diferentes aunque el protocolo TCP logre la ilusión de que existe un único circuito por el que viajan todos los bytes uno detrás de otro (algo así como una tubería entre el origen y el destino). Para que esta comunicación pueda ser posible es necesario abrir previamente una conexión. Esta conexión garantiza que todos los datos lleguen correctamente de forma ordenada y sin duplicados. La unidad de datos del protocolo es el *byte*, de tal forma que la aplicación origen envía bytes y la aplicación destino recibe estos bytes.

Sin embargo, cada byte no se envía inmediatamente después de ser generado por la aplicación, sino que se espera a que haya una cierta cantidad de bytes, se agrupan en un *segmento* y se envía el segmento completo. Para ello son necesarias unas *memorias intermedias* o *buffers*. Cada uno de estos segmentos viaja en el campo de datos de un datagrama IP. Si el segmento es muy grande, será necesario fragmentar el datagrama, con la consiguiente pérdida de rendimiento; y si es muy pequeño, se estarán enviando más cabeceras que datos. Por consiguiente, es importante elegir el mayor tamaño de segmento posible que no provoque fragmentación.

El protocolo TCP envía un *flujo de información no estructurado*. Esto significa que los datos no tienen ningún formato, son únicamente los bytes que una aplicación envía a otra. Ambas aplicaciones deberán ponerse de acuerdo para comprender la información que se están enviando.

Cada vez que se abre una conexión, se crea un canal de comunicación bidireccional en el que ambas aplicaciones pueden enviar y recibir información, es decir, una conexión es *full-duplex*.

• Fiabilidad

¿Cómo es posible enviar información fiable basándose en un protocolo no fiable, como es el IP? Es decir, si los datagramas que transportan los segmentos TCP se pueden perder, ¿cómo pueden llegar los datos de las aplicaciones de forma correcta al destino?

La respuesta a esta pregunta es sencilla: cada vez que llega un mensaje se devuelve una confirmación (*acknowledgement*) para que el emisor sepa que ha llegado correctamente. Si no le llega esta confirmación pasado un cierto tiempo, el emisor reenvía el mensaje.

5.3.3. Formato del segmento TCP

Ya se ha comentado que el flujo de bytes que produce una determinada aplicación se divide en uno o más segmentos TCP para su transmisión. Cada uno de estos segmentos viaja en el campo de datos de un datagrama IP. Para facilitar el control de flujo de la

información los bytes de la aplicación se numeran. De esta manera, cada segmento indica en su cabecera el primer byte que transporta. Las confirmaciones o acuses de recibo (ACK) representan el siguiente byte que se espera recibir (y no el número de segmento recibido, ya que éste no existe).

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Puerto TCP origen																Puerto TCP destino															
Número de secuencia																															
Número de acuse de recibo																															
HLEN				Reservado						Bits código						Ventana															
Suma de verificación																Puntero de urgencia															
Opciones (si las hay)																								Relleno							
Datos																															
...																															

Figura 5.5. Formato y campos de un mensaje TCP.

- **Puerto fuente** (16 bits). Puerto de la máquina origen. Al igual que el puerto destino es necesario para identificar la conexión actual.
- **Puerto destino** (16 bits). Puerto de la máquina destino.
- **Número de secuencia** (32 bits). Indica el número de secuencia del primer byte que transporta el segmento.
- **Número de acuse de recibo** (32 bits). Indica el número de secuencia del siguiente byte que se espera recibir. Con este campo se indica al otro extremo que los bytes anteriores se han recibido correctamente.
- **HLEN** (4 bits). Longitud de la cabecera medida en múltiplos de 32 bits (4 bytes). El valor mínimo de este campo es 5, que corresponde a un segmento sin datos (20 bytes).
- **Reservado** (6 bits). Bits reservados para un posible uso futuro.
- **Bits de código** o indicadores (6 bits). Los bits de código determinan el propósito y contenido del segmento. A continuación se explica el significado de cada uno de estos bits (mostrados de izquierda a derecha) si está a 1.
 - **URG**. El campo *Puntero de urgencia* contiene información válida.
 - **ACK**. El campo *Número de acuse de recibo* contiene información válida, es decir, el segmento actual lleva un ACK. Obsérvese que un mismo segmento puede transportar los datos de un sentido y las confirmaciones del otro sentido de la comunicación.
 - **PSH**. La aplicación ha solicitado una operación *push* (enviar los datos existentes en la memoria temporal sin esperar a completar el segmento).
 - **RST**. Interrupción de la conexión actual. Apagar y volver a empezar.
 - **SYN**. Sincronización de los números de secuencia. Se utiliza al crear una conexión para indicar al otro extremo cuál va a ser el primer número de secuencia con el que va a comenzar a transmitir (veremos que no tiene por qué ser el cero).
 - **FIN**. Indica al otro extremo que la aplicación ya no tiene más datos para enviar. Se utiliza para solicitar el cierre de la conexión actual.
- **Ventana** (16 bits). Número de bytes que el emisor del segmento está dispuesto a aceptar por parte del destino. Sirve para el control de flujo y conforme se vayan asintiendo (ACK) éstas, se pueden enviar más.
- **Suma de verificación** (16 bits). Suma de comprobación de errores del segmento actual. Para su cálculo se utiliza una *pseudo-cabecera* que también incluye las direcciones IP origen y destino.
- **Puntero de urgencia** (16 bits). Se utiliza cuando se están enviando datos urgentes que tienen preferencia sobre todos los demás e indica el siguiente byte del campo

Datos que sigue a los datos urgentes. Esto le permite al destino identificar dónde terminan los datos urgentes. Obsérvese que un mismo segmento puede contener tanto datos urgentes (al principio) como normales (después de los urgentes).

- **Opciones** (variable). Si está presente, únicamente se define una opción: el tamaño máximo de segmento que será aceptado.
- **Relleno**. Se utiliza para que la longitud de la cabecera sea múltiplo de 32 bits.
- **Datos**. Información que envía la aplicación.

5.3.4. Funcionamiento de TCP

TCP/IP es una familia de protocolos que proporcionan una comunicación entre nodos extremo-a-extremo. TCP proporciona los servicios a nivel de transporte e IP a nivel de red. TCP utiliza al IP para establecer comunicaciones fiables entre subredes de datos.

El protocolo IP es no orientado a conexión y no asegura la entrega de todos los datagramas de un mensaje. El protocolo TCP, que utiliza los servicios del IP, incluye los procedimientos necesarios para asegurar la transferencia de datos de forma correcta y ordenada (orientado a conexión), con lo que, en conjunto, resultan adecuados para la transmisión segura de datos.

ENRUTAMIENTO EN REDES TCP/IP

- Las prestaciones vienen dadas por el mecanismo de conmutación a Nivel 3

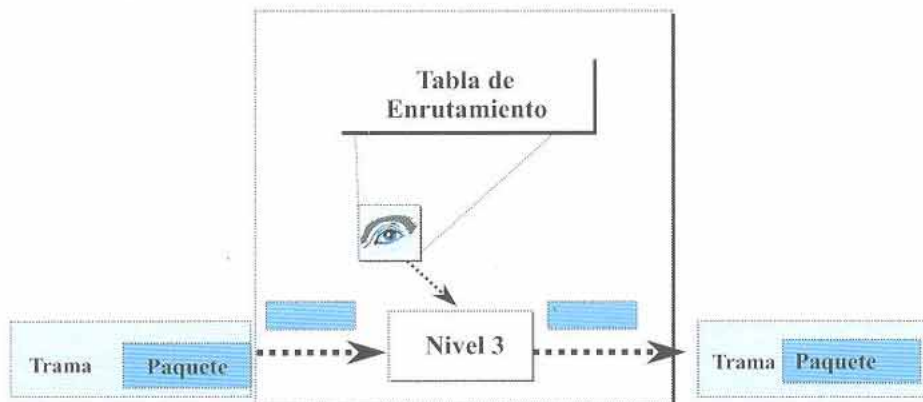


Figura 5.6. El protocolo TCP, en combinación con IP, se utiliza para el enrutamiento de los paquetes de datos.

El protocolo TCP posee funciones tales como: fragmentación de mensajes, retransmisión de segmentos, reordenamiento, establecimiento de prioridades; también define los formatos de los datos, asentimientos, procedimientos de establecimiento y finalización de conexiones. Todo ello con la finalidad de lograr un servicio orientado a conexión y extremadamente fiable para la transferencia de datos.

Debido a que permite ser utilizado por varios usuarios en forma simultánea, hace uso de los Puntos de Acceso al Servicio (SAP) –puntos en los cuales los niveles superiores a TCP reciben los servicios del TCP– para direccionar los diversos usuarios.

Las unidades de transferencia de datos del TCP (TPDU) tienen un formato de 20 bytes, para todos los intercambios. Vale resaltar el campo del número de secuencia, que es utilizado para asignar a cada uno de los segmentos un número que asegure su entrega ordenada.

A pesar de que los términos datagrama y paquete son muy a menudo utilizados como sinónimos, en realidad existe una diferencia. Mientras el datagrama es específico de los protocolos TCP/IP y representa la mínima unidad lógica utilizable por los diversos protocolos, el paquete es una entidad física bien presente para quien administra una red de tipo Ethernet (figura 5.7). En el caso, por lo demás muy frecuente, que en un paquete viaje un solo datagrama, la diferencia es sólo teórica pero existen también específicas

configuraciones hardware de red que utilizan paquetes de dimensión menor respecto a la del datagrama individual. Entonces sucede que un datagrama se descompone en más paquetes durante el envío a la red específica y que sea recompuesto a su llegada, de forma absolutamente transparente respecto al mismo datagrama que “no se da cuenta” de haber sido descompuesto y luego recompuesto. Es evidente cómo en dicha situación los términos paquete y datagrama no coinciden.

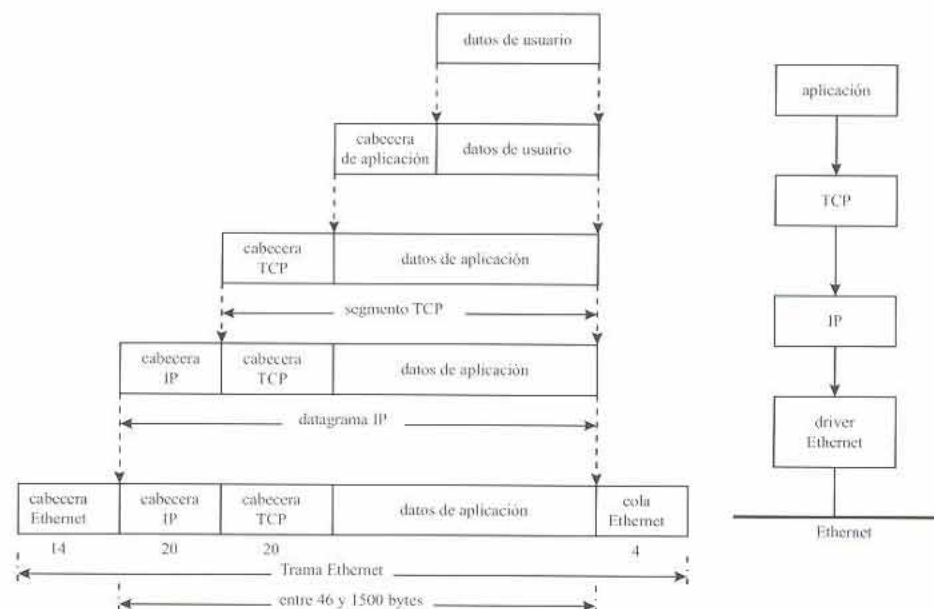


Figura 5.7. Diferencia entre segmentos, datagramas y tramas, en una transmisión en red local Ethernet, haciendo uso del protocolo TCP/IP.

5.4 Conexiones

Una conexión son dos pares *dirección IP:puerto*. No puede haber dos conexiones iguales en un mismo instante en toda la Red. Aunque bien es posible que un mismo ordenador tenga dos conexiones distintas y simultáneas utilizando un mismo puerto. El protocolo TCP utiliza el concepto de conexión para identificar las transmisiones.

En el siguiente ejemplo (figura 5.8) se han creado tres conexiones. Las dos primeras son al mismo servidor Web (puerto 80) y la tercera a un servidor de FTP (puerto 21).

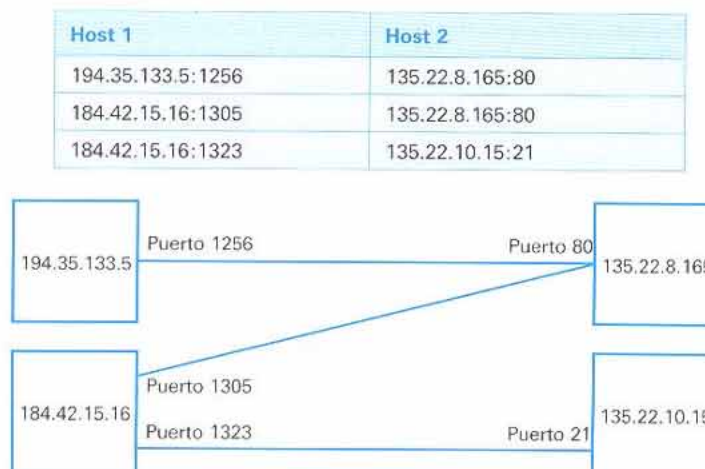


Figura 5.8. Conexiones entre dispositivos, utilizando diferentes puertos.