Las capas de la red de computadores

Xavier Vilajosana Guillén

PID_00147723

UOC
Universitat Oberta de Catalunya
www.uoc.edu

Ninguna parte de esta publicación, incluido el diseño general y la cubierta, puede ser copiada, reproducida, almacenada o transmitida de ninguna forma, ni por ningún medio, sea éste eléctrico, químico, mecánico, óptico, grabación, fotocopia, o cualquier otro, sin la previa autorización escrita de los titulares del copyright.

Índice

Introducción 5

Objetivos 6

1. El nivel de transporte 7
1.1. Objetivos 7
1.2. Servicios ofrecidos por la capa de transporte 8
1.3. Relación entre la capa de transporte y la capa de red 9
1.4. Transporte no orientado a la conexión: UDP 12
1.4.1. Cabecera UDP 12
1.4.2. Cabecera UDP 13
1.5. Transporte orientado a la conexión: TCP 14
1.5.1. Funcionamiento básico de TCP 15
1.5.2. Cabecera TCP 16
1.5.3. Establecimiento de la conexión 19

2. El nivel de red 21
2.1. Funcionalidades básicas: direccionamiento 21
2.2. Servicios de red 24
2.2.1. Modelo de red en modo de circuitos virtuales 25
2.2.2. Modelo de red en modo datagrama 26
2.2.3. Servicio de red orientado y no orientado a la conexión 27
2.3. Direccionamiento en Internet: el protocolo IP 27
2.3.1. IPv4 28
2.3.2. IPv6 42
2.4. Protocolos de soporte a IP 49
2.4.1. Internet Control Message Protocol 49
2.4.2. Address Resolution Protocol 51
2.4.3. Network Discovery Protocol 52
2.4.4. Dynamic Host Configuration Protocol 53

3. El enlace de datos y el control de acceso al medio 54
3.1. Terminología y definiciones 55
3.2. Tipo de enlaces 56
3.3. Tipo de servicios suministrado en la capa de red 56
3.4. Servicios proporcionados por la capa de enlace 57
3.5. Adaptadores y dispositivos de red 58

4. El nivel físico 60
4.1. Medios de transmisión 60

4.1.1. Par trenzado 60
4.1.2. Cable coaxial de banda base 62
4.1.3. Fibra óptica 62

Resumen..... 64

Bibliografía..... 65

Introducción

En este módulo veremos en detalle las características y funcionamiento de los diferentes niveles de la red. El objetivo del módulo es dar una visión general del funcionamiento interno de una red de computadores, ya sea de área local, ya a gran escala, como es Internet. La aproximación a los niveles de la red será desde los niveles más próximos a las aplicaciones hasta los niveles más específicos del hardware.

Como hemos visto en el módulo "Conceptos de redes de computadores", las redes de computadores han sido estructuradas en diferentes niveles que abstraen en los niveles superiores las complejidades de los niveles inferiores. El funcionamiento del nivel aplicación, sus servicios y protocolos los veremos en detalle en el módulo "En el nivel de aplicación". Antes, y para entender algunos de los conceptos del nivel aplicación, pasaremos por el nivel de transporte, que ofrece fiabilidad a la red permitiendo el acceso de diferentes aplicaciones a un único medio de transmisión, la red. Seguidamente profundizaremos en el conocimiento de la capa de red. El nivel de red es primordial para el funcionamiento de Internet pues nos permite dirigir e identificar los nodos de una red. En este módulo conoceremos el funcionamiento actual de la red, así como las nuevas tecnologías de red que se convertirán en estándares en un futuro. La capa de enlace abstrae los nodos de la red del medio físico sobre el que transmiten la información. Esta capa se encarga de dirigir la información entre nodos físicos adyacentes y garantizar una transmisión fiable entre ellos. Hay que destacar la tarea de la subcapa de control de acceso al medio que permite transmitir información entre dos nodos independientemente de la tecnología de red utilizada, ya sea inalámbrica, ya cableada. Finalmente, el nivel físico se encarga de los aspectos de modulación y codificación de las señales físicas que son dependientes directamente de la tecnología y el medio de transmisión.

El módulo, en definitiva, os permitirá haceros una idea general y amplia del funcionamiento de una red de computadores.

Objetivos

El estudio de este módulo os tiene que permitir alcanzar los objetivos siguientes:

1. Profundizar en el conocimiento de cada una de las capas de las redes de computadores.

2. Conocer las funciones principales de la capa de transporte.

3. Conocer el funcionamiento de los protocolos TCP y UDP.

4. Conocer los servicios ofrecidos por la capa de red.

5. Entender el direccionamiento en las redes de computadores.

6. Conocer los estándares de red actuales.

7. Tener una visión global del funcionamiento de la capa de enlace y física.

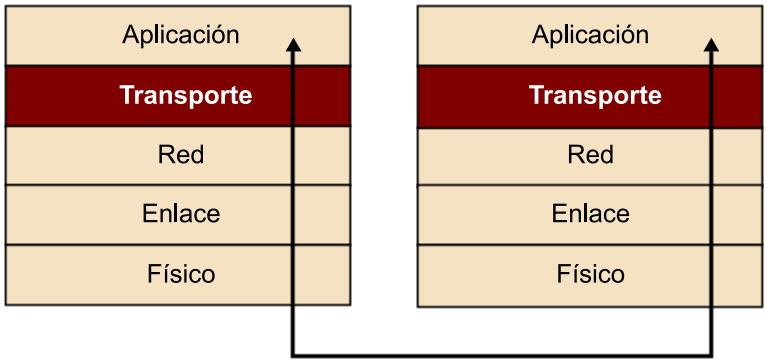
1. El nivel de transporte

El nivel de transporte abstrae la complejidad de la red a las aplicaciones. En este módulo, veremos su funcionamiento con detalle.

1.1. Objetivos

La capa de transporte se encarga de proveer comunicación extremo a extremo entre procesos de aplicaciones ubicadas en diferentes *hosts*. Desde el punto de vista de la aplicación, es como si los *hosts* estuvieran directamente conectados, pero en realidad podrían estar en lugares opuestos del planeta, conectados mediante múltiples *routers* y tipo de enlaces diferentes. La capa de transporte ofrece las funcionalidades básicas para alcanzar este nivel de comunicación lógica, sin que las aplicaciones se tengan que preocupar de la infraestructura de red subyacente.

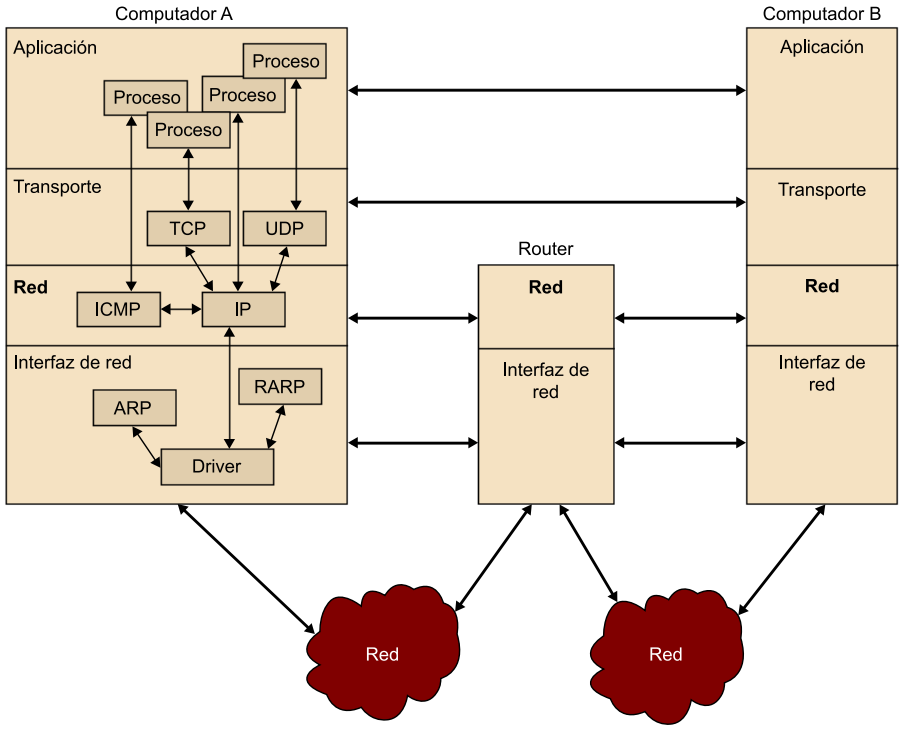
Mensajes extremo a extremo



Los protocolos de la capa de transporte se implementan en los dispositivos extremos de la red y no en los *routers* ni en los dispositivos intermedios. La información transmitida por las aplicaciones es convertida en paquetes de la capa de transporte, conocidos como **segmentos de la capa de transporte**. Los segmentos de la capa de transporte se construyen dividiendo la información que quiere transmitir la aplicación en segmentos de un tamaño determinado, a los que se añade una cabecera. Cada segmento se envía a la capa de red donde será incluido en los paquetes del nivel de red conocidos como **datagramas**. A partir de la capa de red, los datagramas son enviados a los destinatarios pasando por diferentes dispositivos que únicamente examinarán la información correspondiente a la capa de red. Sólo los receptores, al recibir el datagrama en la capa de red, extraerán el segmento de transporte y lo pasarán a la capa de transporte del receptor. La capa de transporte procesará el segmento y lo pasará a la aplicación correspondiente.

En este módulo estudiaremos con detalle los protocolos y funcionalidades de la capa de transporte de datos, haciendo énfasis en los protocolos de transporte de datos de Internet, User Datagram Protocol (UDP) y Transmission Control Protocol (TCP), que ofrecen servicios diferentes a las aplicaciones que los invocan.

La jerarquía TCP/IP



1.2. Servicios ofrecidos por la capa de transporte

La capa de transporte ofrece las siguientes funcionalidades:

- Garantiza la transmisión sin errores, extremo a extremo, independientemente del tipo de red.
- Controla la transmisión extremo a extremo.
- Es responsable del control de flujo de los datos y del control de congestión de la red.
- Es responsable de establecer, mantener y finalizar las conexiones entre dos *hosts* o un *host* y un servidor en una red.
- Asegura que los datos lleguen sin pérdidas, sin errores y sin ser duplicadas.
- Ordena los paquetes que llegan.

- Se encarga de fragmentar los mensajes y recomponerlos en destino cuando es necesario.

- Permite la multiplexación de diversas conexiones de transporte sobre una misma conexión de red.

1.3. Relación entre la capa de transporte y la capa de red

La capa de transporte se ubica justo por encima de la capa de red en la pila de protocolos. Mientras que la capa de transporte se encarga de proveer comunicación lógica entre procesos que se ejecutan en *hosts* diferentes, la capa de red provee comunicación lógica entre *hosts*. Esta diferencia es sustancial, pues la capa de transporte tiene que permitir que múltiples procesos se comuniquen de forma lógica haciendo uso de una única conexión lógica provista por la capa de red. Este concepto se llama **multiplexación** y **demultiplexación** de la capa de transporte.

Además, la capa de red no da garantías de entrega de la información, no garantiza la entrega de los segmentos, ni tampoco garantiza la integridad de la información contenida en el segmento. Por esta razón, la capa de red **no es confiable**. La misión, pues, de la capa de transporte consiste por una parte en proveer la transmisión de datos fiable y permitir la comunicación proceso a proceso en una red creada comunicando *host* a *host*.

Para introducir los protocolos del nivel de transporte, nos centramos en la jerarquía de protocolos Transmission Control Protocol/Internet Protocol (TCP/IP). En esta jerarquía se definen dos protocolos de transporte: el UDP y el TCP.

El UDP no está orientado a la conexión, lo que quiere decir que no implementa fases de establecimiento de la conexión, envío de datos y fin de la conexión, mientras que el TCP está orientado a la conexión.

En el caso de la jerarquía TCP/IP, se definen dos direcciones que relacionan el nivel de transporte con los niveles superior e inferior:

- La **dirección IP** es la dirección que identifica un subsistema dentro de una red.

- El **puerto** identifica la aplicación que requiere la comunicación.

Para identificar las diferentes aplicaciones, los protocolos TCP/IP marcan cada paquete (o unidad de información) con un identificador de 16 bits llamado **puerto**.

- **Puertos conocidos**. Están regulados por la Internet Assigned Numbers Authority (IANA). Ocupan el rango inferior a 1.024 y son utilizados para acceder a servicios ofrecidos por servidores.

- **Puertos efímeros**. Son asignados de forma dinámica por los clientes dentro de un rango específico por encima de 1.023. Identifican el proceso del cliente sólo mientras dura la conexión.

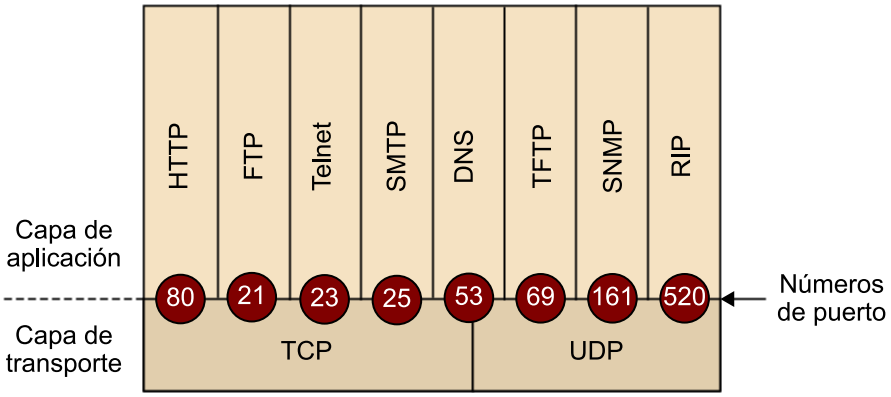
0-255 (IANA)	Aplicaciones públicas.
255-1.023(IANA)	Asignados a empresas con aplicaciones comerciales.
> 1.023	No están registrados (efímeros).

Puerto de aplicaciones públicas establecidas por IANA

Decimal	Palabra clave	Descripción
0		Reservado
1-4		No asignado
5	RJE	Entrada remota de tareas
7	ECHO	Eco
9	DISCARD	Descartar
11	USERS	Usuarios activos
13	DAYTIME	De día
15	NETSTAT	Quién está conectado o NETSTAT
17	QUOTE	Cita del día
19	CHARGEN	Generador de caracteres
20	FTP-DATA	Protocolo de transferencia de archivos (datos)
21	FTP	Protocolo de transferencia de archivos
23	TELNET	Conexión de terminal
25	SMTP	Protocolo SMTP (*simple mail transfer protocol*)
37	TIME	Hora
39	RLP	Protocolo de ubicación de recursos
42	NAMESERVER	Servidor de nombre de anfitrión
43	NICNAME	Quién es
53	DOMAIN	Servidor de denominación
67	BOOTPS	Servidor de protocolo de arranque
68	BOOTPC	Cliente del protocolo de arranque
69	TFTP	Protocolo trivial de transferencia de archivos

Decimal	Palabra clave	Descripción
75		Cualquier servicio privado de conexión telefónica
77		Cualquier servicio RJE privado
79	FINGER	*Finger*
80	HTTP	Protocolo de transferencia de hipertexto
95	SUPDUP	Protocolo SUPDUP
101	HOSTNAME	Servidor de nombre de anfitrión NIC
102	ISO-TSAP	ISO-TSAP
113	AUTH	Servicio de autenticación
117	UUCP-PATH	Servicio de ruta UUCP
123	NTP	Protocolo de tiempo de red
137	NetBIOS	Servicio de nombres
139	NetBIOS	Servicio de datagramas
143	IMAP	*Interim mail access protocol*
150	NetBIOS	Servicio de sesión
156	SQL	Servidor SQL
161	SNMP	*Simple network management protocol*
179	BGP	*Border gateway protocol*
190	GACP	*Gateway access control protocol*
194	IRC	*Internet relay chat*
197	DLS	Servicio de localización de directorios
224-241		No asignado
242-255		No asignado

Puertos usados por algunos de los protocolos de nivel aplicación



Actividad

¿Conocéis aplicaciones comerciales o de software libre que utilicen un puerto determinado? ¿Qué puertos utilizan Skype, Emule, BitTorrent, MSN?

1.4. Transporte no orientado a la conexión: UDP

El User Datagram Protocol (UDP) es un protocolo no orientado a la conexión, de manera que no proporciona ningún tipo de control de errores ni de flujo, aunque utiliza mecanismos de detección de errores. En caso de detectar un error, el UDP no entrega el datagrama a la aplicación, sino que lo descarta.

Hay que recordar que, por debajo, el UDP está utilizando el IP, que también es un protocolo no orientado a la conexión, y que, básicamente, lo que ofrece UDP, en comparación con IP, es la posibilidad de multiplexar conexiones de procesos sobre una misma conexión de red.

La simplicidad del UDP hace que sea ideal para aplicaciones que requieren pocos retardos (por ejemplo, aplicaciones en tiempo real o el DNS). El UDP también es ideal para los sistemas que no pueden implementar un sistema tan complejo como el TCP. UDP es útil en aplicaciones donde no importe la pérdida de paquetes, como telefonía o videoconferencia sobre TCP/IP, monitorización de señales, etc. Estas aplicaciones no toleran retardos muy variables. Si un datagrama llega más tarde del instante en que tendría que ser recibido, entonces se descarta porque es inservible para la aplicación. Eso se traduce en un ruido en el sonido o la imagen, que no impide la comunicación. La tabla siguiente muestra algunas de las aplicaciones más comunes que hacen uso de UDP.

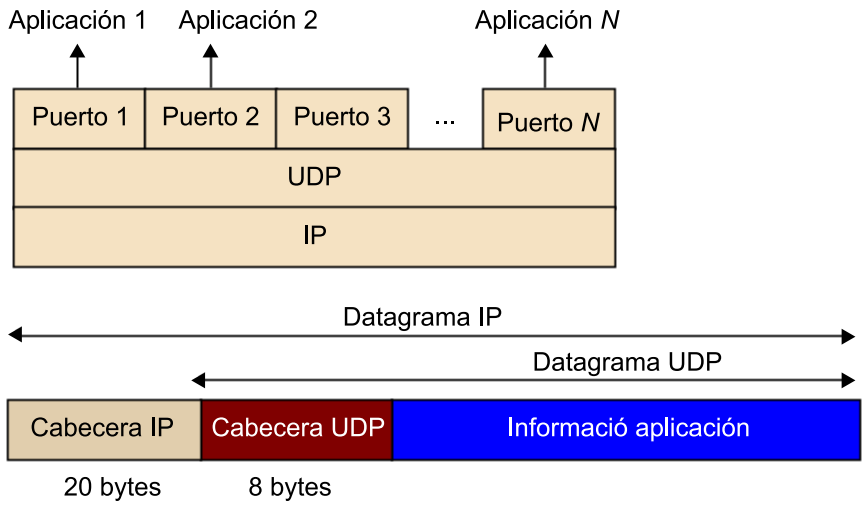
Aplicaciones	Puertos
TFTP	69
SNMP	161
DHCP - BOOTP	67/68
DNS	53

1.4.1. Cabecera UDP

La unidad de cabecera de UDP es el datagrama UDP.

- Cada escritura por parte de la aplicación provoca la creación de un datagrama UDP. No hay segmentación.

- Cada datagrama UDP creado es encapsulado dentro de un datagrama IP (nivel 3) para su transmisión.



1.4.2. Cabecera UDP

La cabecera del datagrama UDP está formada por 8 bytes. Tiene cuatro campos.

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Source port number        |     Destination port number   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        UDP length             |        UDP checksum           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Data  (if any)                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- **Puertos (fuente y destino)**: permiten identificar los procesos que se comunican.

- **UDP *length***: longitud total del datagrama UDP (payload UDP + 8). Es un campo redundante, ya que IP lleva la longitud también. Como la longitud máxima de un datagrama IP es de 65.335 bytes (16 bits de longitud de datagrama):
  - Longitud máxima de un datagrama UDP: 65.335 - 20 bytes = 65.315 bytes.
  - Longitud mínima de un datagrama UDP: 8 bytes.
  - Longitud de datos datagrama UDP: 65.315 - 8 bytes = 65.307 bytes.
  - El tamaño de un datagrama UDP depende del sistema operativo. Casi todas las API[1] limitan la longitud de los datagramas UDP (lo mismo para TCP) a la longitud máxima de los *buffers* de lectura y escritura definidos por el sistema operativo (llamadas al sistema "read" y "write").

(1)Abreviatura de *application programming interface*.

Se suele usar 8.192 bytes como tamaño máximo del datagrama UDP (FreeBSD).

- **UDP *checksum***: detector de errores, cuyo objetivo consiste en comprobar que nadie ha modificado el datagrama UDP y que éste llega a su destino correctamente. Si el *checksum* UDP es erróneo, se descarta el datagrama UDP y no se genera ningún tipo de mensaje hacia el origen. El *checksum* se aplica conjuntamente a una pseudocabecera más la cabecera UDP más el campo de datos. Esta pseudocabecera tiene 3 campos de la cabecera del paquete IP que se envía. Cabe mencionar que el *checksum* del datagrama IP sólo cubre la cabecera IP.

Checksum

El *checksum* se calcula haciendo el complemento a 1 de la suma de todas las palabras de 16 bits que conforman el datagrama UDP.

Ejemplo

Supongamos que tenemos el datagrama (descompuesto en 3 palabras de 16 bits):

0110011001100000
0101010101010101
1000111100001100

La suma de las dos primeras palabras de 16 bits es:

```
 0110011001100000
 0101010101010101
 ----------------
 1011101110110101
```

Añadiendo la tercera palabra obtenemos:

```
 1011101110110101
 1000111100001100
 ----------------
 0100101011000010
```

Ahora hacemos el complemento a 1 (cambiando 0 por 1) y obtenemos:

1011010100111101

Esta última palabra se añade también al datagrama UDP. En la recepción se hace la suma de todas las palabras de 16 bits, que tiene que dar 1111111111111111.

1.5. Transporte orientado a la conexión: TCP

El Transmission Control Protocol (TCP) es un protocolo de nivel de transporte que se usa en Internet para la transmisión fiable de información. Quizás sea el protocolo más complejo e importante de la pila de protocolos de Internet.

Como hemos visto, el UDP no garantiza la entrega de la información que le proporciona una aplicación. Tampoco reordena la información en el caso de que llegue en un orden diferente del utilizado para transmitirla. Hay aplicaciones que no pueden tolerar estas limitaciones. Para superarlas, el nivel de transporte proporciona TCP.

El TCP da fiabilidad a la aplicación, es decir, garantiza la entrega de toda la información en el mismo orden en que ha sido transmitida por la aplicación de origen. Para conseguir esta fiabilidad, el TCP proporciona un servicio orientado a la conexión con un control de flujo y errores.

TCP es un protocolo ARQ[2], extremo a extremo, orientado a la conexión (fases de establecimiento de la conexión, envío de datos y cierre de la conexión) y bidireccional (full duplex). La unidad de datos TCP es el "segmento TCP". TCP, a diferencia de UDP, intenta generar segmentos de un tamaño óptimo que se denomina Maximum Segment Size (MSS), generalmente el mayor posible para minimizar el sobrecoste (overhead en inglés) de las cabeceras, sin que produzca fragmentación en el nivel IP. Al igual que UDP, TCP utiliza multiplexación mediante el uso de puertos, tal como hemos visto con anterioridad.

(2)ARQ es la abreviatura de Automatic Repeat-reQuest Protocol; en castellano, protocolo de repetición de petición automática.

Proporciona fiabilidad mediante los siguientes controles:

- **Control de errores**. TCP es parecido al Go-Back-N pero no descarta segmentos posteriores cuando llegan fuera de secuencia. Permite las retransmisiones selectivas.

- **Control de flujo** (ventana advertida). Sirve para adaptar la velocidad entre el emisor y el receptor. Vigila que el emisor no envíe los segmentos a más velocidad de la que puede emplear el receptor para procesarlos, de manera que se puedan perder paquetes por saturación de la memoria intermedia de recepción.

- **Control de la congestión** (ventana de congestión). Para adaptar la velocidad del emisor a los encaminadores intermedios de la red y evitar así que se colapsen sus memorias intermedias y se puedan perder paquetes.

En transmisión, TCP se encarga de fragmentar los datos del nivel aplicación, y les asigna un número de secuencia antes de enviar los fragmentos al nivel IP.

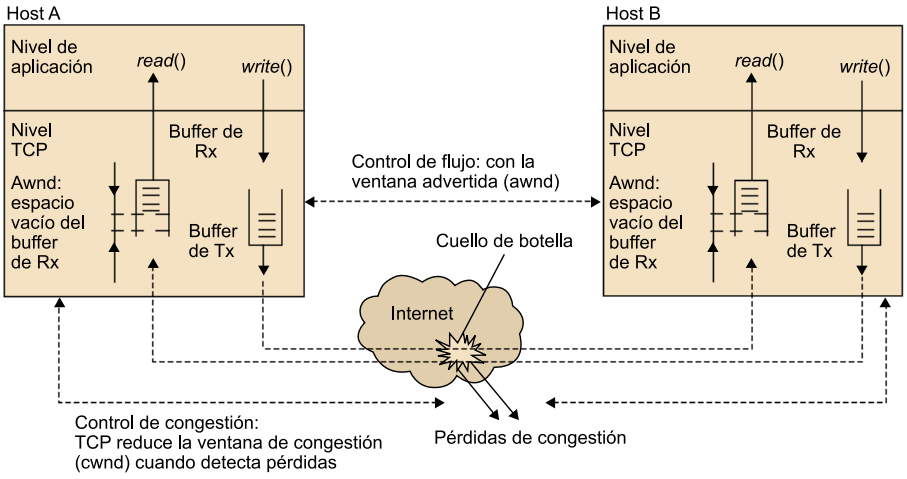
En recepción, como los segmentos pueden llegar fuera de orden, TCP los tiene que reordenar antes de pasarlos a los niveles superiores.

1.5.1. Funcionamiento básico de TCP

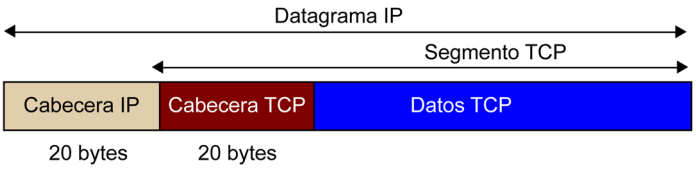
En cada extremo, TCP mantiene una memoria intermedia de transmisión y una de recepción. La aplicación utiliza los llamamientos al sistema operativo *read()* y *write()* para leer y escribir en estas memorias intermedias. Cuando la aplicación escribe la información que se tiene que enviar al emisor, TCP la

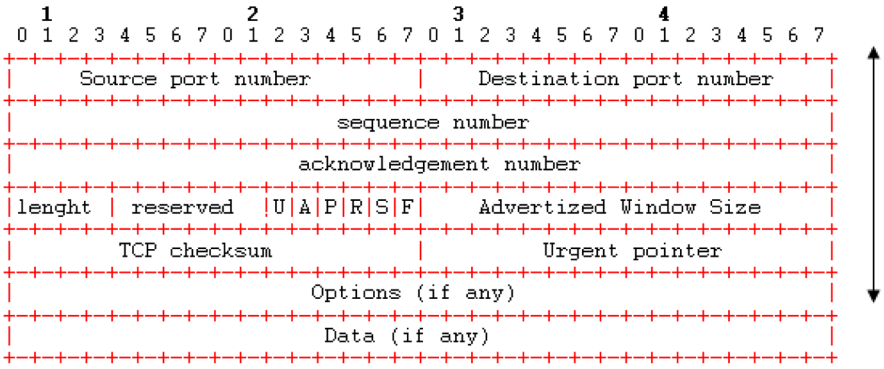
guarda en una memoria intermedia de transmisión. Cuando la memoria intermedia de Tx está llena, el llamamiento *write()* queda bloqueado hasta que vuelva a haber espacio.

Cada vez que llegan segmentos de datos al receptor, el API *read*() los pasa al nivel superior y se envían sus correspondientes confirmaciones. A medida que los datos van siendo confirmados por el receptor, se van borrando de la memoria intermedia de transmisión y queda espacio libre para que la aplicación siga escribiendo.



1.5.2. Cabecera TCP





La cabecera TCP está formada por los siguientes elementos:

- **Source Port / Destination Port**: puertos origen y destino que identifican las aplicaciones.

- **Sequence Number**: número de secuencia del segmento. Identifica el primer byte dentro de este segmento de la secuencia de bytes enviados hasta aquel momento.

- **Initial Seq. Number (ISN)**: primer número de secuencia escogido por el protocolo TCP.

- **Seq Number:** será un número a partir de ISN. Por lo tanto, para saber cuántos bytes llevamos enviados hay que hacer Seq. Number - ISN".

- **Acknowledgment Number**: contiene el próximo número de secuencia que el transmisor del ACK espera recibir, es decir, es "Seq. Number + 1" del último byte recibido correctamente.
  - TCP es FULL-DUPLEX: cada extremo mantiene un Seq. Number y un Ack Number.

- **Header length**: longitud total de la cabecera del segmento TCP en words de 32 bits (igual que el campo header length de la cabecera IP).
  - Tamaño mínimo de la cabecera TCP = 20 bytes (*header length* = 5).
  - Tamaño máximo de la cabecera TCP = 60 bytes (*header length* = 15).

- **Reserved**: bits reservados para posibles ampliaciones del protocolo. Se ponen 0.

- **Flags**: hay 6 *flags* (bits) en la cabecera. Son válidos si están en 1.

- **Urgente (URG)**: indica que se utiliza el campo Urgent Pointer de la cabecera TCP.

- **Acknowledgement (ACK)**: indica que se utiliza el campo Acknowledgement Number. Válido para todos menos para el SYN inicial.

- **Push (PSH)**: indica que el receptor tiene que pasar los datos de la memoria intermedia de recepción a los niveles superiores tan rápidamente como sea posible. Operación más rápida sin llenar la memoria intermedia. La activación de este *flag* depende de la implementación. Las implementaciones derivadas de BSD lo activan cuando la memoria intermedia de Tx se queda vacía.

- **Reset (RST)**: se activa cuando se quiere abortar la conexión. Un ejemplo es cuando se recibe un segmento de un cliente dirigido a un puerto donde no hay ningún servidor escuchando. En este caso, TCP contesta con un segmento con el *flag* de reset activado.

- **Synchronize (SYN)**: se utiliza en el establecimiento de la conexión, durante la sincronización de los números de secuencia al inicio de la conexión.

- **Finalize (FINAL)**: se utiliza en la finalización de la conexión.

- **Advertised Window Size**: tamaño de la ventana advertida por el receptor al transmisor (Sliding Window) para el control de flujo. La ventana máxima es de 65.535 bytes.

- **TCP Checksum**: se utiliza para detectar errores en el segmento TCP, y para tener una mayor certeza de que el segmento no ha llegado a un destino equivocado.

- Igual que en UDP, el *checksum* TCP se aplica conjuntamente a una pseudocabecera + la cabecera TCP + el campo de datos TCP. Esta pseudocabecera tiene 3 campos de la cabecera IP y el tamaño del segmento TCP (cabecera + payload).
  - La pseudocabecera sólo es para calcular el *checksum*: no se transmite en el segmento TCP, ni se incluye en la longitud del paquete IP.
  - El tamaño del segmento TCP no se pone en la cabecera TCP, sólo se tiene en cuenta en el cálculo del *checksum*.
  - A diferencia de UDP, en TCP el cálculo del *checksum* es obligatorio.

```
            1                   2                   3                   4
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Source IP Address                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Destination IP Address                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Zero     |Protocol(6)tcp |       Total TCP Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- **Urgent Pointer**: Campo válido si *flag* URG = 1. Implementa un mecanismo para indicar datos urgentes (es decir, que se tienen que atender lo más pronto posible). Es un puntero en el Seq Number que indica la parte de datos urgentes dentro del campo de datos. Los datos urgentes irán desde el primer byte del segmento hasta el byte indicado por el Urgent Pointer. Este *flag* se utiliza rara vez. Un ejemplo es cuando se teclea un control-C (interrupción) desde la aplicación telnet.

- **Options**: TCP permite añadir opciones a la cabecera, pero, a diferencia de IP, las opciones de TCP suelen utilizarse. Podemos destacar:
  - **Maximum Segment Size**: se utiliza durante el establecimiento de la conexión para sugerir el valor del MSS en el otro extremo MSS = MTU red directamente conectada - IP Header - TCP Header (sin opciones). Si la red es ethernet (MTU 1500), entonces MSS = 1460.
  - **Window Scale Factor**: se utiliza durante el establecimiento de la conexión para indicar que el valor de la ventana advertida se tiene que multiplicar por este factor de escala. Eso permite advertir ventanas mayores de 216.
  - **Timestamp**: se utiliza en el cálculo del RTT.

– **SACK**: permite que TCP haga retransmisión selectiva (*selective ack*). TCP utiliza el campo ACK para indicar hasta dónde se ha recibido correctamente. Con la opción SACK, el receptor puede indicar bloques de segmentos que se han recibido correctamente más allá del segmento confirmado por el ACK. De esta manera, el emisor puede escoger mejor los segmentos que se tienen que retransmitir.

- **Padding**: bytes de relleno añadidos para que la cabecera tenga un múltiplo de 32 bits.

Actividad

Asumimos que un extremo cliente TCP (emisor) ha escogido el 28.325 como número de secuencia inicial (ISN), mientras que el extremo receptor TCP (servidor) ha escogido como ISN el 12.555. ¿Qué indica un segmento cliente (emisor) TCP con número de secuencia 29.201, número ACK 12.655 y ventana 1.024?

Solución

El número de secuencia indica que el cliente ya ha transmitido desde el byte 28.325 hasta el byte 29.200 (875 bytes en total) y que en este segmento transmitirá a partir del byte 29.201. El número ACK indicará al receptor que el emisor ha recibido correctamente hasta el byte 12.654 y que espera recibir a partir del 12.655. La ventana indica al receptor que el cliente sólo puede aceptar 1.024 bytes antes de confirmarlos. Por lo tanto, el servidor TCP actualizará su ventana de transmisión en 1.024.

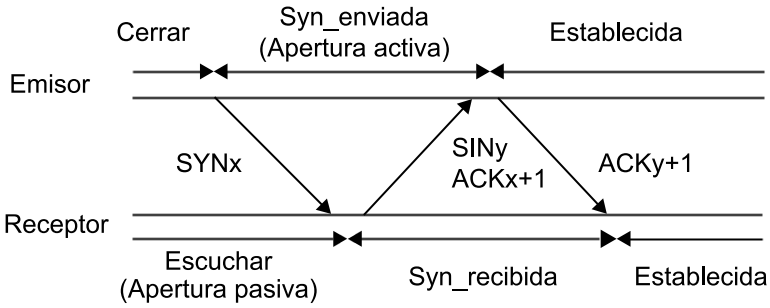
1.5.3. Establecimiento de la conexión

TCP es un protocolo orientado a la conexión tal como hemos dicho con anterioridad. Eso implica que en todo proceso de comunicación habrá una fase de establecimiento de la conexión donde emisor y receptor se sincronizan con el fin de poder intercambiar datos. EN TCP se usa el algoritmo 3-Way Handshake, que consiste en el intercambio de tres segmentos que no llevan datos (sólo es la cabecera TCP):

1) El emisor envía un segmento SYN, con petición de conexión.

2) El receptor (servidor) devuelve un segmento SYN + ACK como respuesta.

3) El cliente responde con un segmento ACK que reconoce el SYN + ACK. Una vez establecida la conexión se pasa a la fase de envío de datos.



De la misma manera que el establecimiento de la conexión, se hace necesario un proceso de finalización de la conexión. El cierre de la conexión puede deberse a diversas causas:

- El cliente o el servidor cierra la conexión (LLS *close*()).
- Por alguna razón se envía un reset de la conexión (*flag* activo *RST*).
- Cierre a causa de una interrupción, un control∧D o un control∧C, etc.

La finalización también se hace por medio del envío de segmentos TCP. El primer segmento de final puede enviarlo tanto el cliente como el servidor. El cierre normal se produce a causa de un *close()* del cliente, lo que provoca el intercambio de 3 o 4 segmentos TCP.

2. El nivel de red

La capa de red se encarga de proporcionar conectividad y ofrecer mecanismos para la selección del mejor camino entre dos puntos separados de la red, permitiendo la interconexión de equipos que pueden estar ubicados en redes separadas geográficamente unas de otras, garantizando la conectividad extremo a extremo, independientemente de la tecnología de enlace de datos utilizada y del camino que siga la información en los puntos intermedios.

Las principales ventajas que nos proporciona esta capa son, por una parte, independencia de la tecnología de red (hacia capas inferiores), y, por otra, un sistema de abstracción que permite utilizar una gran diversidad de aplicaciones y protocolos de transporte (hacia capas superiores) como, por ejemplo, TCP o UDP.

Básicamente, la capa de red, especialmente en Internet, está compuesta por tres grandes bloques:

1) el protocolo que describe la manera de enviar información,

2) el protocolo de direccionamiento, que decide por dónde tienen que ir los datagramas para llegar a su destino,

3) la capa de red también identifica el mecanismo para informar de cualquier error que se haya producido en el envío de la información.

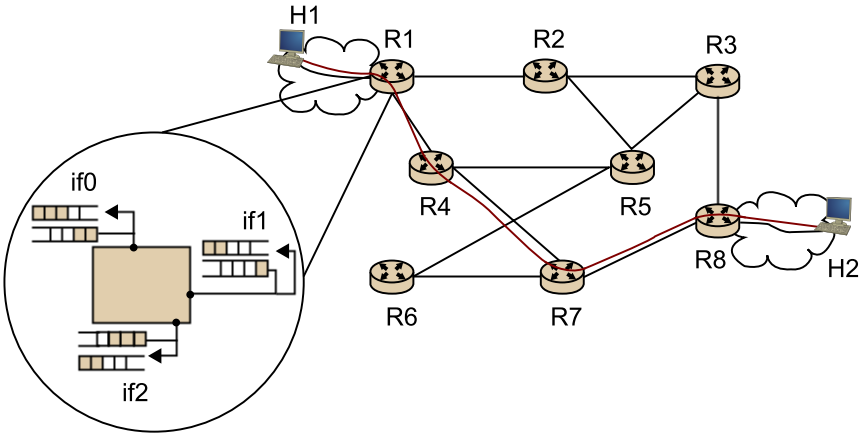
2.1. Funcionalidades básicas: direccionamiento

Una red está compuesta básicamente por dos tipos de entidades, los clientes (también conocidos por el nombre de *hosts* o el de equipos finales) y los encaminadores (o *routers*). Los clientes son los equipos de red encargados de la comunicación, son el origen y el final de la misma. Normalmente son servidores de información o equipos de usuarios finales que acceden a los servidores. Por su parte, los encaminadores, a pesar de que en determinados casos también pueden ser equipos finales, se limitan a enviar la información que reciben por una interfaz de entrada a la correspondiente de salida que lleve los datagramas hacia su destino. Para poder saber hacia dónde va la información, los encaminadores se sirven de lo que se conoce como tablas de direccionamiento.

La capa de red necesita que tanto los encaminadores como los equipos finales tengan un identificador único. Este identificador permite que cualquier otro equipo de la red lo pueda localizar y enviarle información. En particular, en una red como Internet estos identificadores se conocen como direcciones (direcciones IP).

Ejemplo de red con encaminadores y equipos finales

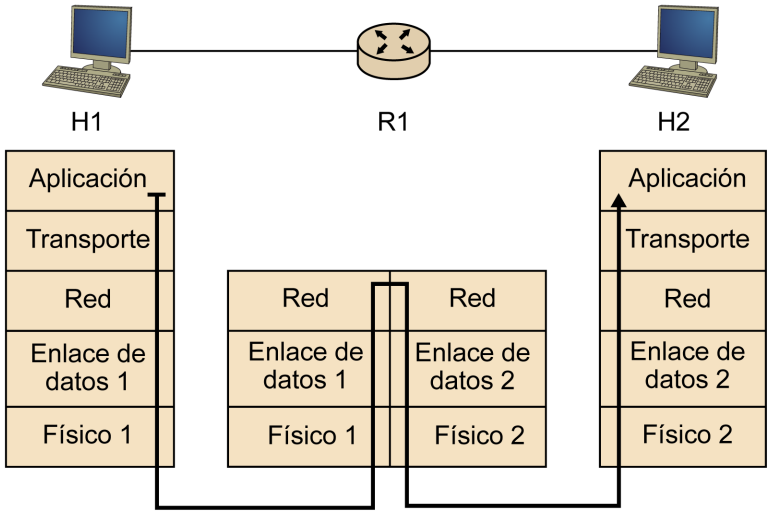
La figura siguiente muestra una red con ocho encaminadores y dos equipos finales. En la figura también se puede observar una simplificación de cómo funciona un encaminador internamente. Para simplificar, en vez de indicar las direcciones de los diversos equipos hemos identificado, por una parte, los diferentes encaminadores con R (de routers) y un número que los identifica, y por otra, los diferentes equipos finales con H (de *hosts*) y un número para identificarlos.



Los encaminadores están compuestos por una serie de interfaces de entrada y salida, que son las encargadas de recibir los datagramas de los equipos vecinos; estas interfaces están controladas por unas colas (o *buffers*) que almacenan los paquetes (de entrada o de salida) para poder enviarlos cuando sea posible, o lo que es lo mismo, cuando el encaminador tenga recursos para atender las colas de entrada, o bien cuando la red tenga recursos (ancho de banda disponible) para las colas de salida. Internamente, el encaminador dispone de una lógica para decidir qué hacer con los datagramas que llegan. Esta decisión normalmente implica enviar el datagrama por otra interfaz que lo llevará más cerca de su destino.

Así, el datagrama va saltando por los encaminadores hasta llegar al destino. Cada equipo de red por el que pasa el datagrama se conoce como *salto* o *hop*. Hay que señalar que los encaminadores trabajan a nivel de red, lo que quiere decir que no interpretan los campos presentes en los niveles superiores, tal como muestra la figura siguiente.

Capas usadas por el direccionamiento en protocolos de red



Cuando un equipo envía un datagrama hacia un destino, dicho datagrama va dirigido inicialmente al encaminador asociado a la red del equipo. Este encaminador mirará el destino del datagrama, y lo enviará por la interfaz que lo lleve hacia su destino dependiendo de una tabla de direccionamiento. El siguiente encaminador hará lo mismo hasta que el datagrama llegue a su destino final. La lista de encaminadores que sigue un datagrama se conoce como el *camino* o *path* del datagrama. Hay que señalar que este *path* será diferente dependiendo del origen y el destino del datagrama.

Como muestra, se puede ver en la figura del ejemplo anterior que el camino que siguen los datagramas para ir desde H1 hasta H2 es H1-R1-R4-R7-R8-H2, haciendo un total de 5 saltos para llegar al destino.

Hemos dicho que el equipo envía el datagrama al encaminador de su red, lo cual implica que ha de conocer a priori cómo llegar a ese encaminador con el fin de poder enviarle el datagrama. La secuencia específica de acciones que realiza el equipo se puede ver más detalladamente en la figura siguiente:

1) Se crea el datagrama con las direcciones de red origen y destino apropiadas.

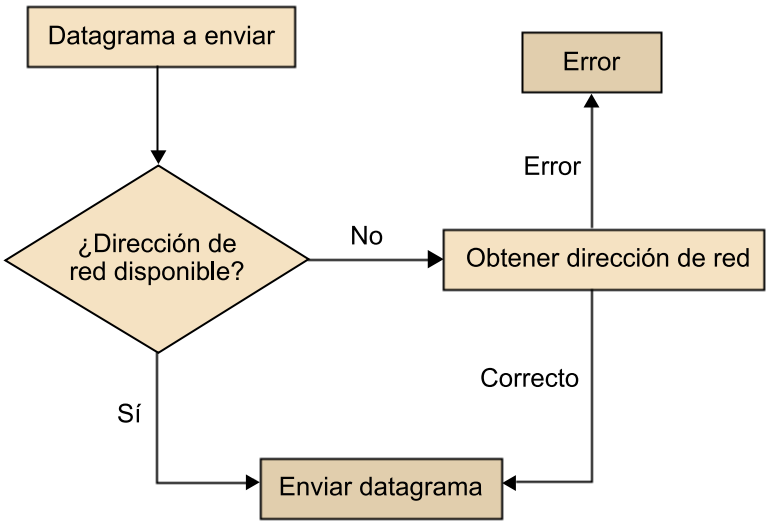
2) Se busca la dirección de red del encaminador - *next hop* (o del equipo final si está directamente conectado al encaminador - *last hop*).

3) Si no se dispone de la dirección de red, salimos con error ya que no sabemos cuál es el siguiente salto para enviar el datagrama.

4) Si hemos podido conseguir la dirección, enviamos el datagrama (con las direcciones de red origen y destino originales) al siguiente salto del *path* o al equipo final si estamos en el *last hop*.

5) Repetir desde el paso 2 hasta que el datagrama llegue a su destino.

Diagrama de bloques simplificado del envío de un datagrama a un equipo de red



El datagrama no sigue cualquier *path*, sino que los encaminadores disponen de una tabla de direccionamiento (*forwarding table* o *routing table*) que indica por qué interfaz se tienen que enviar los datagramas en función de su destino. Para llenar estas tablas es necesario utilizar unos algoritmos de direccionamiento.

2.2. Servicios de red

El servicio de red define las características que tiene que tener el transporte punto a punto de los datos en la capa de red. Así, se definen características como la fiabilidad en el envío de la información, orden de llegada de los paquetes, umbrales de retardo en hacer llegar la información a destino, información de congestión en la red, etc., entre los diferentes emisores y receptores dentro de la red.

Actualmente existen dos modelos de servicios de red claramente diferenciados, el **modelo de circuito virtual** y el **modelo de datagrama**. A continuación se describen los dos, aunque haremos más énfasis en el modelo de datagrama, dado que es el utilizado por el nivel de red propuesto por Internet y, por lo tanto, el más relevante en la actualidad.

2.2.1. Modelo de red en modo de circuitos virtuales

Un circuito virtual es un camino que se preconfigura entre dos puntos de la red, de forma que los nodos intermedios sepan a priori la dirección en la que se tiene que enviar la información perteneciente a cada circuito. Este paradigma permite acelerar enormemente el envío de paquetes entre dos puntos, ya que el procesamiento intermedio es mínimo; esta prerreserva encima permite garantizar una serie de recursos de red para el tráfico que pasa por el circuito. Por eso, este modelo de red se pensó para servicios en tiempo real (multimedia).

En cualquier circuito virtual se pueden distinguir tres fases claramente separadas:

1) Establecimiento del circuito virtual: esta fase se inicia en la capa de red del emisor, utilizando la dirección del receptor. El emisor envía un datagrama de creación de circuito que provoca que cada nodo intermedio reserve los recursos pedidos de forma iterativa hasta llegar al destino. Cada uno de los nodos intermedios tendrá que actualizar su estado para acomodar el nuevo circuito, o denegar la creación en caso de que no queden más recursos disponibles (normalmente, ancho de banda). Si el establecimiento del circuito puede llegar hasta el destinatario se avisa al emisor indicando que la conexión ha sido satisfactoria y se puede empezar a enviar información.

2) Transferencia de datos: en el caso de que se haya podido establecer el circuito virtual se puede empezar a enviar datos entre los dos puntos.

3) Desconexión del circuito virtual: esta desconexión puede iniciarla tanto el emisor como el receptor, y se avisa secuencialmente a través de la capa de red a todos los nodos intermedios hasta llegar al otro extremo. Esta desconexión permite liberar los recursos ocupados por el circuito.

Actividad

¿Qué diferencias creéis que pueden existir entre el inicio de un circuito virtual en la capa de red y el establecimiento de una conexión en la capa de transporte? (por ejemplo, el *three-way-handshaking*).

Solución

El establecimiento de la conexión de la capa de transporte involucra únicamente a dos sistemas finales. Los dos extremos acuerdan la comunicación y determinan los parámetros de conexión, mientras que los nodos intermedios de la red no intervienen. Por el contrario, el establecimiento de un circuito virtual en la capa de red obliga a involucrar a todos los nodos intermedios.

El principal inconveniente que tiene la utilización de circuitos virtuales es que los nodos intermedios tienen que mantener las reservas de recursos pedidas independientemente de que se estén utilizando, con el potencial problema de infrautilizar la red.

2.2.2. Modelo de red en modo datagrama

Si enviar información a través de un circuito virtual implica previamente establecer un camino y reservar recursos, en una red en modo datagrama (también llamado conmutación de paquetes), el paquete se envía directamente a la red con una dirección origen y una dirección destino. Entonces, es trabajo de la red (a través de las tablas de direccionamiento de cada encaminador) el hacer llegar el paquete a su destino.

Como se puede comprobar, en este tipo de comunicación no hay reserva de recursos ni camino preestablecido entre los extremos de la comunicación. Por lo tanto, a un encaminador le pueden llegar datagramas de diferentes destinos a la vez, y los datagramas pueden seguir caminos diferentes para llegar al destino (dependiendo de los algoritmos de direccionamiento), lo que provoca el efecto lateral de que los paquetes pueden llegar fuera de orden (el paquete número 2 llega antes que el número 1).

Las redes en modo datagrama son las más usadas actualmente, debido principalmente a que el protocolo de red de Internet (IP) lo utiliza. A pesar de que hemos visto que el modelo de datagrama hace una utilización de los recursos más eficiente, eso viene con un coste asociado. Con este tipo de redes se complica muchísimo la priorización del tráfico, ya que nunca se sabe a priori cuánto tráfico se recibirá, y lo que es más grave, no se sabe qué prioridad se tiene que dar a cada uno de los flujos de datos presentes en la red, tanto es así que Internet se basa en el paradigma conocido como *best effort*, que implica que la red no nos da ninguna garantía de calidad y que "lo hará lo mejor que pueda" para hacer llegar el datagrama en su destino.

Actividad

¿Cuál de los dos modelos de red vistos consideráis que hace un uso de los recursos más eficiente?

Solución

El hecho de que un circuito virtual obligue a hacer una prerreserva de recursos implica que se tiene que conocer previamente el modelo y el patrón de tráfico que sigue la aplicación, pero como eso, a priori, no es posible muchas veces, se acostumbra a hacer lo que se conoce como *overprovisioning* (reservar más recursos de los que se consideran necesarios), lo que inequívocamente lleva a un sistema menos eficiente en términos de recursos.

Por su parte, utilizar el modo datagrama no implica ninguna prerreserva, con lo que la red siempre enviará tan rápidamente como pueda la información, siempre y cuando haya recursos disponibles.

2.2.3. Servicio de red orientado y no orientado a la conexión

Análogamente a los protocolos de transporte que hemos visto anteriormente, en el nivel de red también podemos tener protocolos orientados a la conexión y otros que no lo sean. La principal diferencia entre las dos alternativas es que en el servicio orientado a conexión se guarda el estado de la conexión, o lo que es lo mismo, se tiene conocimiento de todas las conexiones establecidas, mientras que en el caso del servicio no orientado a conexión no se tiene constancia de las conexiones existentes. Un ejemplo claro de servicio de red orientado a la conexión es el modelo de circuitos virtuales visto anteriormente.

Hay que señalar que el diseño de un protocolo de red no orientado a conexión no excluye que a niveles superiores (transporte) se pueda definir un protocolo orientado a conexión. El ejemplo más indicativo de eso es la pila de protocolos TCP/IP, donde TCP es orientado a la conexión mientras que IP no lo es. Tanto es así que la arquitectura actual de Internet sólo proporciona el modelo de servicio de datagrama, lo que no garantiza el orden de los paquetes, el retardo en el envío, ni la llegada del datagrama.

2.3. Direccionamiento en Internet: el protocolo IP

El protocolo de capa de red por excelencia es Internet Protocol (IP). IP es un protocolo que basa el intercambio de información en el modelo no orientado a conexión. IP es el protocolo utilizado en Internet para identificar los nodos de la red, así como también para enviar la información de una forma estándar e independiente de la tecnología de red utilizada. Otra característica muy importante de IP es que no implementa mecanismos que garanticen la integridad de los datos que se envían por la red (eso se hace en la capa de transporte), sólo se comprueba que no haya errores de transmisión en la cabecera.

Todos los protocolos de red requieren algún mecanismo con el fin de identificar los nodos de la red; esta identificación en el protocolo IP se realiza a través de lo que se conoce como dirección IP; actualmente existen dos versiones diferentes del protocolo IP: IPv4 e IPv6. IPv4 es el protocolo más utilizado en la actualidad en Internet, pero, dado el gran crecimiento que ha sufrido la red, se ha propuesto una extensión, IPv6, más actual y que algún día se prevé que sustituya al IPv4.

Ved también

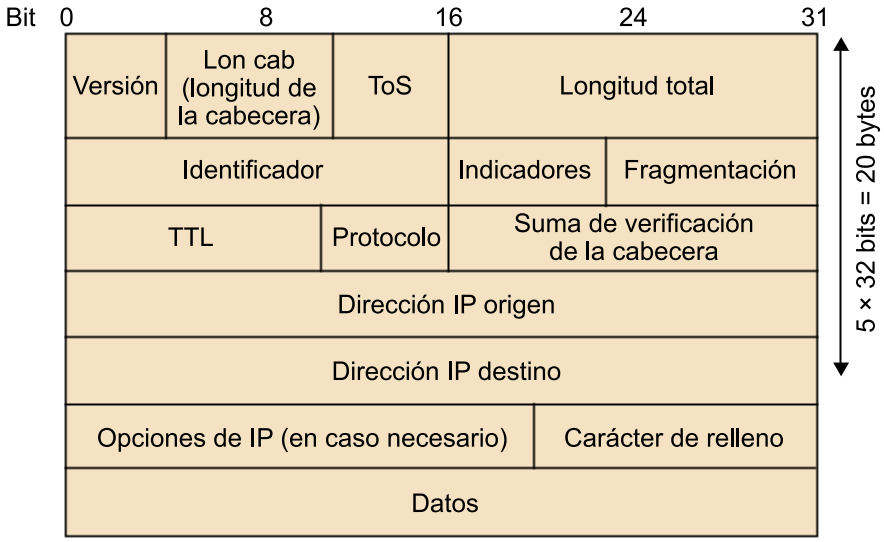
En el subapartado "Direccionamiento en Internet: el protocolo IP" se detalla cómo funcionan los protocolos IPv4 e IPv6 y qué ventajas e inconvenientes tienen.

2.3.1. IPv4

IPv4 fue propuesto en 1981 (documento RFC-791) y en la actualidad todavía es el protocolo de red por excelencia. IPv4 define el formato que se tiene que utilizar para enviar información entre dos puntos distantes de la red; el protocolo proporciona mecanismos que determinan cómo se divide el direccionamiento de una forma escalable en una red tan grande como Internet.

La cabecera IP

IPv4 define qué información de control y qué formato tienen que tener los paquetes que se envían a la red. Por eso, y al igual que con los protocolos de transporte vistos anteriormente, es necesario definir una cabecera que sirva para poder identificar los paquetes. La cabecera de Ipv4 se puede ver en la figura siguiente.



En esta figura podemos ver los siguientes elementos:

Versión (4 bits): indica qué protocolo de red utiliza este datagrama. Para IPv4 está fijado en `0x04`.

- **Hdr. Len (4 bits)**: la cabecera IP puede tener un tamaño variable a causa del campo de opciones. Ese tamaño indica en qué punto empiezan los datos del protocolo de transporte. En particular, dicho campo indica el valor en función de la cantidad de palabras de 4 octetos que tiene la cabecera; así, un valor de `0x05` quiere decir una cabecera de 20 octetos, el valor usado en la mayoría de los casos por ser el tamaño por defecto cuando no hay opciones.

- **Type of Service (ToS) (8 bits)**: este campo permite distinguir entre diferentes tipos de datagramas IP; inicialmente se definieron parámetros en función de: retardo bajo, tasa de transferencia alta o fiabilidad. Así, dependiendo del tipo de tráfico que contenga el paquete, por ejemplo tráfico

interactivo, puede quererse un retardo bajo, o en el caso de que el tráfico sea de baja prioridad, se puede querer un coste mínimo. La lista de los diferentes tipos de servicio se puede encontrar en el documento RFC-1349. En realidad, los encaminadores normalmente ignoran este campo y utilizan la técnica *best effort* para encaminar los paquetes.

- **Total Length (16 bits)**: indica el tamaño total del datagrama en octetos, lo que incluye la cabecera así como el campo de datos. Los 16 bits indican un tamaño máximo del datagrama de 65.535 octetos. Aunque en general el tamaño máximo utilizado es de 1.500 octetos.

- **Identifier (16 bits), flags (3 bits) y fragmentation (13 bits)**: estos campos hacen referencia a lo que se conoce como fragmentación IP.

Ved también

La fragmentación será explicada en el subapartado "Fragmentación IP".

- **TTL (8 bits)**: inicialmente, este campo hacía referencia al tiempo de vida del datagrama en milisegundos. Pero en la práctica contiene el máximo número de routers que puede atravesar el paquete hasta que llegue al destino. A cada salto, un encaminador decrementa en 1 el valor de este campo, cuando el TTL llega a 0, el paquete es descartado. Con esta técnica se permite descartar datagramas en el caso de que haya algún bucle provocado por algún problema con el sistema de direccionamiento, y así evitar tener paquetes en la red más tiempo del necesario. De este campo se puede derivar que el "diámetro" máximo posible de Internet es de 255 saltos. Aunque en la actualidad no acostumbra a superar los 30.

- **Protocol (8 bits)**: este campo indica el protocolo presente en la capa de transporte, que será capaz de interpretarlo. Normalmente, este campo puede ser `0x06` para TCP o `0x11` para UDP. La lista completa se puede encontrar en los documentos RFC-1700 y RFC-3232. Con este enlace entre la capa de red y la de transporte, se puede tener diversos protocolos de transporte y distinguirlos fácilmente, pasando el control al que corresponda de forma eficiente.

- **Header Checksum (16 bits)**: permite detectar algunos tipos de error de transmisión a la cabecera. Es importante señalar que no se comprueba la integridad de la capa de transporte y superiores. Recordemos que IP no garantiza la recepción de los datos. El *checksum* se calcula tratando cada dos octetos de la cabecera como enteros y sumándolos utilizando aritmética de complemento a 1, ignorando para la suma el mismo campo que contiene el *checksum*. La integridad se comprueba comparando la suma con la almacenada en la cabecera. En caso de error, el paquete se descarta. Un pequeño inconveniente de este *checksum* es que cada encaminador tiene que recalcularlo para cada paquete, dado que el campo TTL (y quizás algunas opciones) cambian a cada salto.

- **Dirección Origen (32 bits)**: indica la dirección origen del paquete.

- **Dirección Destino (32 bits)**: adonde va dirigido el paquete.

- **IP Options**: este campo es el que hace que la cabecera IP pueda ser variable en tamaño. Las opciones, que normalmente no se utilizan, permiten ampliar las funcionalidades de la cabecera IP. A pesar de no hacerse servir casi nunca, el hecho de comprobar su existencia en cada encaminador reduce mucho el rendimiento del protocolo IPv4. Por eso, durante el diseño de la versión 6 del protocolo se cambió la forma de implementar estas opciones.

- **Padding**: por motivos de eficiencia, los datos tienen que empezar en una posición múltiple de 4 octetos, por lo tanto, en el caso de que algunas opciones introduzcan una desalineación, el *padding*, que normalmente es todo ceros, alinea a la palabra del siguiente campo.

- **Data (*payload*)**: los datos del datagrama que se pasarán al nivel de transporte, o sea, la información que realmente se quiere transmitir.

Ved también

Se puede encontrar más información sobre el direccionamiento en el subapartado "Direccionamiento IPv4".

Fragmentación IP

Uno de los puntos más críticos a la hora de diseñar el protocolo IP fue la necesidad de introducir la fragmentación. La fragmentación IP es necesaria porque no todas las redes, ni todos los protocolos de enlace de datos, pueden transportar paquetes de tamaño arbitrario. En general, el tamaño máximo vendrá delimitado en función de la tecnología de red utilizada. Por lo tanto, a causa de la diversidad de tecnologías que actualmente coexisten en Internet, podemos encontrar casos en los que el tamaño máximo de trama permitido[3] sea menor en algún encaminador dentro del camino a seguir por los datagramas, forzando a IP a dividir la trama en fragmentos más pequeños que se puedan transmitir. Un ejemplo puede ser Ethernet, que permite tramas de un tamaño máximo de 1.500 bytes, mientras que una tecnología como Asynchronous Transfer Mode (ATM) en general tiene el máximo en 9.180 bytes. Hace falta señalar que cuando se fragmenta un datagrama IP, cada fragmento tiene que ser autocontenido, y tiene que poder ser ensamblado en el destino final (hacerlo en los encaminadores intermedios supondría una pérdida de rendimiento considerable), por lo que sólo dividir el datagrama no es suficiente, es necesario hacer algún tipo de proceso.

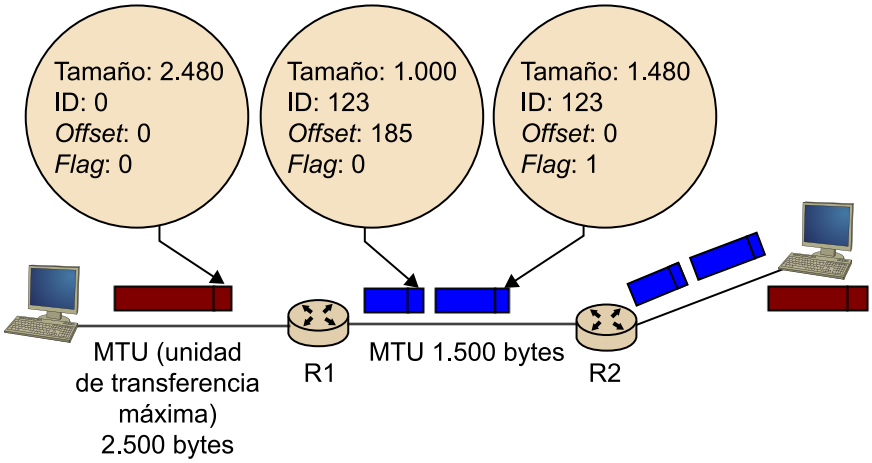
(3)En inglés, Maximum Transfer Unit (MTU).

Cuando se tiene que dividir un datagrama IP, primero se replica la cabecera IP para cada fragmento, y acto seguido se actualizan los campos de la cabecera: *identification*, *flag* y *fragmentation offset*. Así, todos los fragmentos pertenecientes al mismo datagrama tendrán el mismo identificador, cada fragmento contendrá el desplazamiento y finalmente el flag, que será 1 si hay más paquetes, o 0 si es el último. Por las restricciones en la implementación, y para reducir el número de bits que se utilizan para almacenar este desplazamiento, se de-

cidió hacerlo con múltiplos de 8 bytes; así, un desplazamiento de 64 bytes –o sea, que el fragmento IP contenga desde el byte 65 del datagrama original– se representará con un 8 en el campo *fragmentation offset* (ya que 8 x 8 = 64).

Ejemplo de fragmentación

En la figura siguiente se puede ver el caso en que un equipo envía un paquete de tamaño MTU = 2.500 bytes. Lo que quiere decir que el paquete tendrá 2.480 bytes de información útil y 20 de cabecera. A la hora de fragmentar se generan dos paquetes diferentes, uno de 1.480 + 20 y otro de 1.000 + 20. Como se puede ver, el tamaño útil no cambia, pero, por el hecho de tener dos paquetes diferentes, estamos replicando la cabecera. El valor del identificador viene dado por un contador interno en el encaminador que fragmenta, el desplazamiento (*offset*) para el primer fragmento es 0, y el flag 1, lo que indica que todavía hay más fragmentos, para el segundo fragmento el *offset* contiene un 185, ya que se especifica con grupos de 8 bytes, y un 0 en el *flag*, que indica que se trata del último fragmento del datagrama original. Cuando el datagrama llegue a su destino final será reensamblado y pasado a los niveles superiores de forma transparente.



Direccionamiento IPv4

Los protocolos de red necesitan disponer de una dirección única que permita identificar todos los nodos de la red. En el caso de Ipv4, tal como se puede deducir de la cabecera IPv4, la máxima cantidad de direcciones disponibles es muy grande: 2^{32} (4.294.967.296). Para simplificar su escritura, se dividen los 32 bit en 4 bloques de 8 bits cada uno, y además, en vez de utilizar la representación binaria, que es poco legible, en la práctica se escribe la dirección IP en notación decimal separada por puntos; la dirección estará formada por 4 bloques de números entre 0 y 2^8 - 1 (255).

Representación binaria y decimal de una dirección IP

10001111		00101101		00000001		00010111
143	·	45	·	1	·	23

Además, tener un número tan grande de direcciones supone un enorme problema de gestión, por lo que se propuso un sistema de asignación de direcciones jerárquico. En Internet, las direcciones IP están compuestas por dos partes, la parte de red y la parte del equipo, lo que se utiliza para poder estructurar las direcciones y organizarlas por zonas administrativas.

La parte de red está formada por los bits superiores de la dirección IP e indica a qué red pertenece un conjunto de equipos, o lo que es lo mismo, cuál es el encaminador de salida del conjunto de equipos. Por el contrario, la parte del equipo son los bits inferiores de la dirección IP e identifican el equipo dentro de su red.

Inicialmente, esta división con redes se hizo a través de clases, concretamente se definieron 5 clases diferentes (A, B, C, D y E) tal como muestra la tabla siguiente:

- Las **direcciones de clase A** son las destinadas a empresas muy grandes, como por ejemplo IBM, o grandes operadoras americanas, como AT&T WorldNet Services, y proporcionan acceso a 2^{24} (16.777.216) equipos por red, donde 8 bits están destinados a identificar la red y el resto hasta los 32 se utilizan para los equipos finales. De direcciones de clase A hay un total de 2^{7} (255).

Carencia de direcciones IP

El reparto de clases A es uno de los causantes de la fuerte carencia de direcciones IP en la actualidad.

- Las **direcciones de clase B** son las que se dan a grandes entidades, universidades y determinados proveedores de Internet. Permiten repartir 2^{16} (65.536) equipos por red, y hay un total de 2^{14} (16.384) direcciones de clase B.

- En el caso de las **direcciones de clase C**, son las destinadas a medianas empresas con fuerte presencia en Internet; en este caso se dispone de 2^{8} (256) direcciones, con un total de 2^{21} (2.097.152) direcciones de tipo C a repartir.

- Con respecto a las **direcciones de clase D**, se consideran un tipo de clase especial, denominadas clases multicast, que sirven para enviar el tráfico denominado punto multipunto.

- Finalmente, las **direcciones de clase E** están reservadas para un uso futuro.

División de redes por clases

Clase	Bits iniciales	Bits Red	Rango Red
A	0	7 (+1)	1.0.0.0-127.0.0.0
B	1 0	14 (+2)	128.0.0.0-191.255.0.0
C	1 1 0	21 (+3)	192.0.0.0-223.255.255.0
D	1 1 1 0	-	224.0.0.0-239.0.0.0
E	1 1 1 1 0	-	240.0.0.0-255.0.0.0

Direcciones de propósito específico

Aparte de la división en redes también se destinaron una serie de direcciones de propósito específico para casos especiales:

- **Direcciones de *host***. Indican un equipo dentro de la red actual y tienen la forma 0.**host**, donde *host* es la parte del equipo de la red actual, o sea, que la parte de la dirección de red es todo 0.

- **Direcciones de red**. Hacen referencia a la red, pero no a los equipos dentro de ella; las direcciones de red son de la forma **red**.0 para direcciones de clase C, **red**.0.0 para la clase B y **red**.0.0.0 para la clase A, o, lo que es lo mismo, que la dirección del equipo de red está toda en 0. Hay un caso especial, que es la dirección 0.0.0.0, que indica "este *host*" de "esta red", aunque no siempre se implementa en los sistemas operativos actuales.

- **Direcciones de *broadcast***. Indican todos los equipos de una red concreta. La dirección se representa con **red**.255 para direcciones de clase C. Análogamente al caso de las direcciones de red, las de clase B serán **red**.255.255, y las de clase A, **red**.255.255.255, o sea, que la dirección del equipo de red es todo 1. Siempre que se reciba un datagrama en la dirección de *broadcast* todos los equipos tienen que responder.

Las direcciones de *broadcast*, a su vez, tienen una dirección especial, que es la 255.255.255.255, que hace referencia a toda la red (Internet). Entonces, si alguien enviara un datagrama a la dirección 255.255.255.255 toda la Internet tendría que responder. Como eso provocaría graves problemas de escalabilidad y exceso de tráfico, no hay ningún encaminador que reenvíe tráfico *broadcast* por sus interfaces. El tráfico de *broadcast* siempre se quedará en la red que lo ha emitido.

Actividad

Dada la dirección IP 120.1.32.54, indicad cuál es la dirección de red, la dirección del *host* y la dirección de *broadcast* de la red.

Solución

La dirección 120.1.32.54 forma parte de las direcciones de clase A, por lo tanto, la dirección de red será la 120.0.0.0, la del host la 0.1.32.54 y la de *broadcast* sería la 120.255.255.255.

- **Direcciones de *loopback***. Son desde la 127.0.0.0 hasta la 127.0.0.255 y son las que se utilizan internamente por los equipos. Cuando un equipo arranca, automáticamente crea una interfaz virtual (interfaz de *loopback*) para uso interno del sistema operativo, normalmente sólo se utiliza la 127.0.0.1.

- **Direcciones privadas**. Son las utilizadas por redes locales internas que no salen a Internet. La lista con todos los rangos privados se puede encontrar

en la tabla siguiente. Como se observa en dicha tabla, se pueden configurar internamente diversos rangos de direcciones privadas, cuya utilidad es evitar colisiones en asignaciones de direcciones en configuraciones internas con nodos de otras redes del exterior. Otra funcionalidad es permitir asignar más direcciones a nuestras redes que IP públicas asignadas por los operadores.

Lista de rangos de direcciones privadas

Clase	Rango Red	Número de subredes
A	10.0.0.0-10.255.255.255	1
B	172.16.0.0-172.31.255.255	16
C	192.168.0.0-192.168.255.255	255

Network Address Translation

El problema principal de las direcciones privadas es que no pueden acceder a Internet directamente, los encaminadores nunca enviarán a Internet el tráfico originado o con destino a direcciones privadas, ya que no saben cómo encaminar el salto siguiente. Con el fin de evitar esta limitación y permitir la transferencia de datos entre direcciones privadas y públicas, los encaminadores incluyen una técnica denominada Network Address Translation (NAT).

Network Address Translation (NAT)

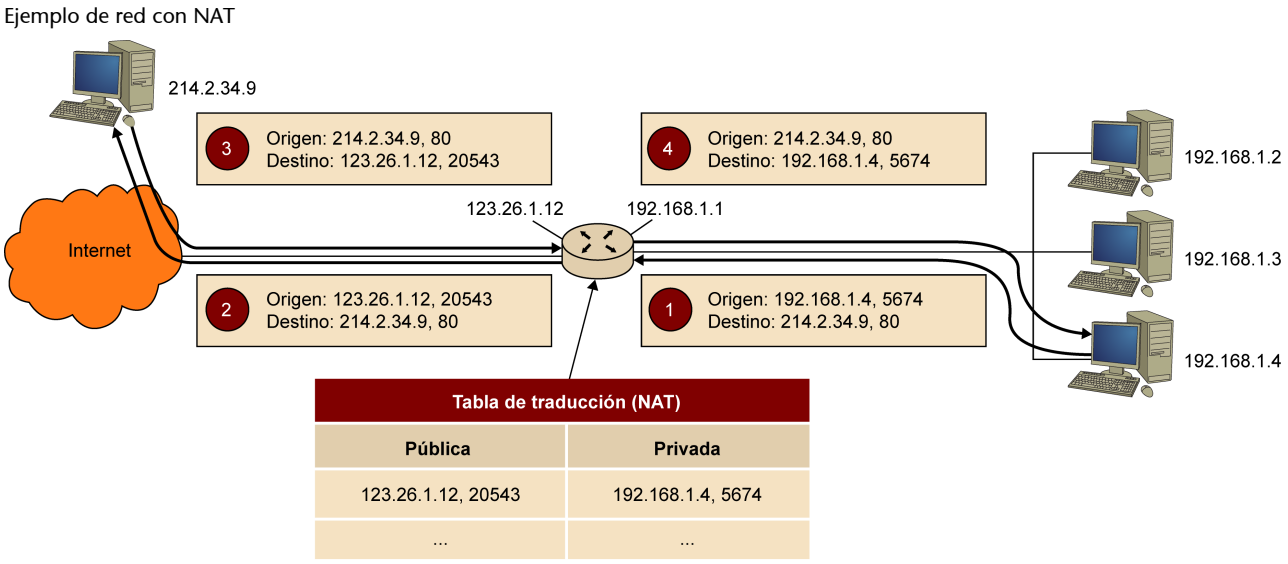
Por lo que se puede deducir de lo que se ha visto hasta ahora, cada equipo de una red IPv4 tiene que disponer de una IP pública para poder acceder a la red. Uno de los problemas principales que se encuentran cuando se piden IP a las operadoras es, generalmente, que el usuario (o la empresa) tiene más equipos que IP asignadas. Un ejemplo de eso es el del usuario con conexión ADSL, que recibe una sola IP por parte de la compañía telefónica, mientras que muchas veces el usuario dispone de diversos equipos, como el PC de sobremesa, el portátil, la PDA, etc. Con el fin de permitir que todos los equipos se puedan conectar a la red al mismo tiempo hay dos opciones, pedir más IP (solución difícil y cara) o utilizar direcciones IP privadas y configurar el encaminador para que haga la conversión desde la dirección IP privada a la IP pública disponible. Eso se puede conseguir mediante lo que se conoce como NAT.

Referencia bibliográfica

La NAT está definida en detalle en los documentos RFC-2663 y RFC-3022.

NAT es una tabla de traducción que se utiliza de la siguiente forma. Si, por ejemplo, un cliente con dirección privada quiere establecer una conexión con un equipo que tiene una dirección IP pública (punto 1 de la figura siguiente) –por ejemplo, un servidor–, el cliente enviará el paquete hacia el encaminador de su red. El mencionado encaminador tendrá configurada una tabla de traducción, donde transformará la IP origen del datagrama en una IP pública que tenga reservada a tal efecto. Para completar la traducción, el encaminador mapeará el puerto origen (de la capa de transporte) en un nuevo puerto origen asignado por el encaminador. El punto 2 de la figura muestra un ejemplo, en el que el encaminador transforma la IP origen (192.168.1.4) y el puerto origen (5.674) del equipo en la IP pública del encaminador (123.26.1.12) y un puerto asignado dinámicamente (20.543 en el ejemplo). La estación destino ve un

datagrama como si hubiera sido enviado por el encaminador, al que responde de la forma habitual usando TCP/IP. Finalmente, el encaminador, al recibir la respuesta, mira la tabla de traducción y deshace el cambio para enviar el paquete final a la estación origen. Si la entrada no hubiera estado en la tabla, el encaminador hubiera asumido que el paquete iba realmente dirigido a él.



Este mecanismo es muy útil cuando se quiere evitar el uso de direcciones públicas, a pesar de que tiene una serie de inconvenientes que no permiten usarlo en determinados entornos. Primero, hay protocolos de aplicación (por ejemplo, FTP) que incrustan la IP del cliente dentro del datagrama, esta IP es usada por el servidor para establecer una nueva conexión (por ejemplo, el caso del FTP activo), y como el cliente incrusta la IP privada, eso impide que se pueda establecer la conexión.

Otro problema importante es que todas las conexiones se tienen que iniciar desde el equipo con IP privada, ya que el encaminador tiene que establecer la entrada en la tabla de traducción antes de poder enviar información hacia el equipo con IP privada, lo que normalmente implica que no se puedan tener servidores con IP privadas. Hay que decir, sin embargo, que eso se puede solucionar con una técnica denominada Port Address Translation (PAT), en la que el encaminador tiene configurado de forma estática un mapeado, por lo que cuando llega un datagrama a un puerto concreto, automáticamente reenvía el paquete hacia el equipo que esté configurado con IP privada. En según qué entornos, el PAT se conoce también como Destination NAT (DNAT) o incluso como puerto Forwarding, pero la idea de fondo es la misma.

Classless inter Domain Routing

Una vez definidas las diferentes clases de redes, se vio que esta solución era claramente insuficiente, ya que forzaba igualmente a las grandes y medianas operadoras (con clases A y B) a gestionar desde un solo equipo un número de direcciones demasiado grande, por lo que se propuso el Classless inter Domain Routing (CIDR).

CIDR propone un mecanismo más flexible para poder subdividir nuestras redes. CIDR no sustituye la división por clases, que continúan siendo las unidades básicas de asignación de direcciones; por el contrario, CIDR nos permite dividir las direcciones asignadas en subredes más pequeñas y manejables.

Así, con CIDR, la separación entre el equipo y la red se consigue gracias a una máscara. Esta máscara tiene la forma de una dirección IP, que enmascara los bits de una dirección normal para poder distinguir el equipo y la red de forma sencilla.

Ejemplo de máscara

Una máscara de 255.255.255.0 permite separar la dirección de red de la del equipo final haciendo AND con la dirección IP, así:

143	. 45	. 1	. 23
255	. 255	. 255	. 0
143	. 45	. 1	. 0

De donde se puede extraer la dirección de la subred (los unos de la máscara - 143.45.1) y la dirección del equipo (los ceros de la máscara - 23). En este caso, la red constará de 2^8 IP válidas como una clase C, de las que 2^8 - 2 serán asignables a equipos; hay que recordar que las direcciones especiales de red y de *broadcast* (143.45.1.0 y 143.45.1.255, respectivamente) no son asignables. La representación de esta subred se hace con la siguiente nomenclatura: 143.45.1.23/255.255.255.0.

Como se puede observar, eso nos da un nivel más fino de división que simplificará mucho la gestión interna de redes; si una entidad dispone de una clase B (146.43.0.0), internamente la entidad puede decidir subdividir las 65.536 direcciones en varias subredes, por ejemplo, con 256 subredes de 256 IP cada una: desde la 146.43.0.0/255.255.255.0 a la 146.43.255.0/255.255.255.0. Observad que los valores de 255 y 0 para la dirección de red son correctos, sin que representen direcciones de *broadcast* y de red, respectivamente. Eso lo podemos saber gracias a la máscara.

Una restricción no escrita, pero generalmente adoptada a la hora de definir las máscaras, es que todos los unos de la máscara tienen que ser consecutivos.

Máscaras correctas e incorrectas

Máscaras como 255.145.0.0 se consideran inválidas, ya que, traducidas a formato binario, daría:

11111111 10010001 00000000 00000000

Mientras que otras, como 255.255.128.0, son totalmente correctas ya que, en formato binario, resulta:

11111111 11111111 11111110 00000000

Donde todos los unos son consecutivos, a pesar de no estar alineados en el octeto.

Esta restricción de los unos consecutivos nos permite simplificar la representación de la máscara en un formato más compacto; así, otra forma de indicar la separación entre la red y el equipo es a través de un formato que indica cuántos bits representan la red, por ejemplo, 143.45.1.23/24 indica que la máquina 143.45.1.23 pertenece a la red 143.45.1.0/255.255.255.0, o, lo que es lo mismo, que tiene 24 bits para la dirección de red y 8 para la de los equipos.

Por otra parte, gracias a la clasificación por subredes, los encaminadores tienen el trabajo más fácil, ya que para poder decidir la ruta que tiene que tomar cualquier datagrama, es suficiente mirar la red de destino, y no es necesario comprobar toda la dirección IP. La dirección IP entera, idealmente, sólo la mirará el último router de la cadena, o sea, el que esté dentro de la misma subred a la que pertenezca aquel destino. Para implementar este mecanismo, los encaminadores basan la decisión de direccionamiento en una política llamada Longest Prefix Match, lo que significa que, de todas las rutas posibles, siempre se coge la que tiene más bits coincidentes con el destino del paquete.

Actividad

Dentro de las siguientes subredes e IP, indicad cuántas IP asignables puede contener la red y explicad el significado.

Dirección	IP asignables	Explicación
147.83.32.0/24		
1.23.167.23/32		
1.23.167.0/32		
147.83.32.0/16		

Solución

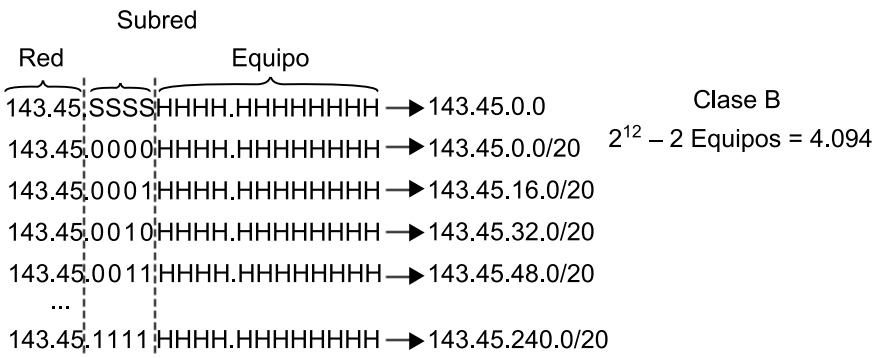
Dirección	IP asignables	Explicación
147.83.32.0/24	254	Es una dirección de red en la que tenemos 8 bits para los equipos; sabiendo que la dirección de *broadcast* y la de red no son asignables, acabamos con el total de 2^8 - 2 direcciones para asignar.
1.23.167.23/32	1	Dirección con una subred con sólo un equipo. No es útil en un caso real, pero es correcta.

Dirección	IP asignables	Explicación
1.23.167.0/32	1	Como no hay parte de equipo, todo es de red, el hecho de que el último octeto sea 0 hace que la IP no sea una dirección de red genérica sino una específica, como en el caso anterior.
147.83.32.0/16	1	Hace referencia al equipo 32.0 de la red de clase B 147.83.0.0. Hay que señalar que no es una dirección de red, ya que no todos los bits de fuera de la máscara son 0; así, se trata como una dirección de equipo.

Actividad

En la dirección de clase B 143.45.0.0/16, indicad qué subredes /20 se pueden crear y cuántos equipos contienen cada una.

Solución

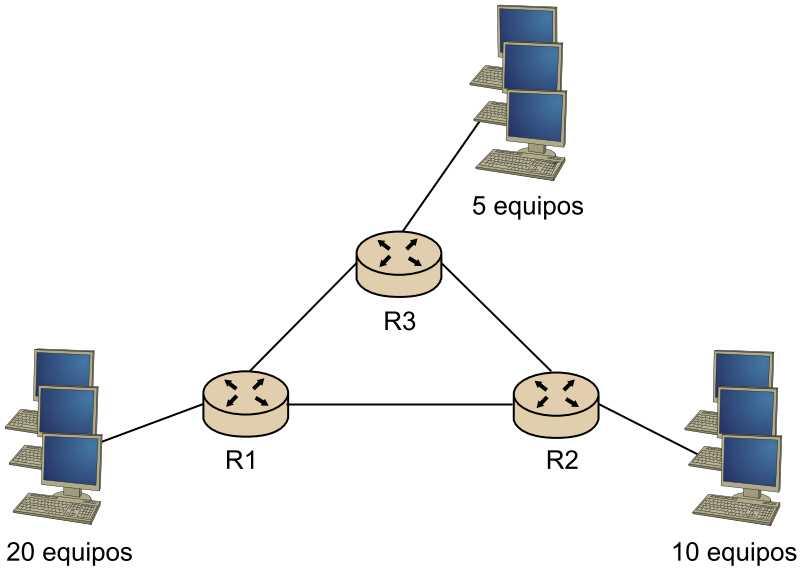


El CIDR se utiliza generalmente en conjunción con la máscara de subred de tamaño variable[4], técnica que tiene por objetivo optimizar la utilización de las direcciones IP a través de una asignación inteligente de las máscaras de red. Esta asignación se hará ahora teniendo en cuenta el número de máquinas de cada subred y se asignarán máscaras de tamaño ajustado a las necesidades particulares de cada una. VLSM se puede ver como la creación de subredes de las subredes.

(4)En inglés, Variable Length Subnet Mask (VLSM).

Actividad

1) Dada la red de la figura, se nos proporciona el rango de direcciones 147.83.85.0/24. Se pide que se asignen rangos de direcciones a todas las subredes y a los enlaces entre los encaminadores.



2) ¿Qué problema tiene la asignación de direcciones hecha en el ejercicio anterior?

Solución

1) La asignación de direcciones se puede hacer siguiendo la siguiente política:

- Los 20 equipos más el router necesitan un total de 5 bits ($2^5 = 32$).
- Los 10 + 1 equipos tienen suficiente con 4 bits ($2^4 = 16$).
- Los 5 + 1 equipos necesitan 3 ($2^3 = 8$), con lo que esta subred no podrá crecer más.
- Finalmente, los enlaces punto a punto necesitarán 2 bits, ya que necesitamos espacio para las direcciones de *broadcast* y de red.

En resumidas cuentas, nos harán falta un prefijo /27, un /28, un /29 y tres prefijos /30 respectivamente. De esta manera, una posible asignación sería:

- Los 20 equipos pueden usar la 147.83.85.0/27, donde el último byte sería 000XXXXX. Con dirección de red 147.83.85.0 y dirección de *broadcast* 147.83.85.31.
- Los 10 equipos dispondrán de la 147.83.85.32/28 donde el último byte será 0010XXXX. Con dirección de red 147.83.85.32 y dirección de *broadcast*, 147.83.85.47.
- En el caso de los 5 equipos utilizaremos la subred 147.83.85.48/29 donde el último byte será 00110XXX. Con dirección de red 147.83.85.48 y dirección de *broadcast*, 147.83.85.55.
- Finalmente, los tres prefijos /30 (de los enlaces entre los encaminadores) se pueden dividir con el último byte 001110XX, 001111XX y 010000XX, respectivamente. O sea, 147.83.85.56/30, 147.83.85.60/30 y 147.83.85.64/30. Con direcciones de red 147.83.85.56, 147.83.85.60 y 147.83.85.64. Con direcciones de *broadcast* 147.83.85.59, 147.83.85.63 y 147.83.85.67.

2) El problema de esta asignación viene dado por el hecho de que se ha ajustado demasiado el número de bits para cada subred, y en el caso de que crezca el número de equipos (especialmente en la subred de cinco equipos), nos tocaría redimensionar la red de nuevo con el coste que eso supone.

Tipo de datagramas IP

IPv4 especifica tres tipos de tráfico claramente diferenciados dentro de la red: *unicast, broadcast* y *multicast*.

El **tráfico unicast** es el más común, la comunicación está formada por dos interlocutores que se intercambian información, a menudo estas conexiones son desde un cliente hacia un servidor, que a la vez puede tener conexiones unicast hacia otros clientes.

El **tráfico broadcast** se basa en enviar la información a todos los equipos presentes en una subred. Como ya hemos visto anteriormente, eso se puede conseguir enviando un paquete a una dirección que sea la dirección de red y todo 1 a la dirección del equipo final, por ejemplo, para la red 126.76.31.0/24, la dirección de broadcast sería 126.76.31.255.

Requisitos del tráfico broadcast

Hay que señalar, sin embargo, que enviar tráfico broadcast normalmente requiere algún privilegio en la red (ser administrador), además, los encaminadores en general no propagan este tipo de tráfico con el fin de evitar problemas de seguridad, como, por ejemplo, ataques del tipo Denial of Service (DoS).

Finalmente, el caso del **tráfico multicast** se basa en el paradigma de enviar información desde un solo origen hacia muchos destinos a la vez; la base del tráfico multicast es que el emisor no tiene por qué tener conocimiento de quiénes serán sus receptores (al contrario de la política de unicast, que requiere conocer a los interlocutores). Eso se consigue a través de lo que se conoce como grupos de multicast. Como se ha visto anteriormente, la Internet Assigned Numbers Authority (IANA) ha reservado las direcciones de tipo D a multicast, éstas son las que van del rango 224.0.0.0 hasta el 239.0.0.0. Dentro de este grupo de direcciones hay unas cuantas reservadas a grupos multicast conocidos como permanentes.

Así, si una estación concreta está interesada en recibir un contenido multicast, lo que hará será suscribirse al servicio mediante el protocolo Internet Group Management Protocol (IGMP), que especifica el formato del paquete que se tiene que generar con el fin de poder registrarse en un grupo y poder recibir el contenido. IGMP soporta dos tipos de paquetes, los de pregunta y los de respuesta. Normalmente, los de pregunta son paquetes dirigidos a todos los equipos donde los que tienen sesiones multicast activas responden, así los encaminadores (que tienen que tener soporte para multicast) pueden construir lo que se conoce como el árbol multicast, para encaminar los paquetes hacia sus destinos.

La ventaja principal de multicast es que la información que se envía, en vez de replicarse desde el origen una vez por cada destino forma un árbol de manera que se minimiza el número de copias.

Actividad

Un servidor de chat tiene en un momento dado un total de 80 clientes conectados por todo el mundo. Indicad qué número y de qué tipo son las conexiones que tiene abiertas este servidor.

Solución

Dado que el chat es un protocolo que utiliza TCP/IP, y que los clientes, a pesar de hablar entre ellos, pasan siempre por el servidor, se trata del típico escenario con 80 conexiones unicast entre los 80 clientes y el servidor.

Actividad

Un administrador de la red 147.83.0.0/16 quiere enviar un paquete de broadcast a la subred 147.83.20.0/24. Indicad qué dirección de destino tendría el paquete, cuántos paquetes se generarían y a cuántas máquinas como máximo podría llegar.

Solución

Dado que la subred a la que se quiere enviar el broadcast tiene 8 bits, eso implica que se generaría un solo paquete con dirección destino 147.83.20.255 y que lo recibirían como máximo 255 - 2 = 253 estaciones. Ya que la dirección 147.83.20.0 y la 147.83.20.255 están reservadas para la dirección de red y la de broadcast, respectivamente.

El futuro de Ipv4

Cuando se diseñó IPv4 se creía que su gran número de direcciones IP (2^{32}) sería suficiente para poder aguantar el gran crecimiento que se esperaba de una red como Internet. Hay que recordar que Internet entró en funcionamiento en 1969 con el nombre de ARPANet, un proyecto subvencionado por el Departamento de Defensa de Estados Unidos. Eso provocó que, cuando Internet se desplegó al cabo de unos años en la red comercial, el reparto de direcciones no se hiciera de forma equitativa, y las grandes empresas estadounidenses pudieron adjudicarse una gran cantidad de direcciones de clase A, dejando a países, como China y otros que se han desarrollo posteriormente, con muchas menos direcciones de las necesarias. Como referencia, Estados Unidos tiene sobre 1.500 millones de direcciones asignadas, mientras que China, con una población mucho más numerosa, sólo dispone de 200 millones, aproximadamente. Para hacerse una idea, España tiene asignadas en la actualidad en torno a 22 millones.

Con este paradigma, muy pronto se ve que con la actual política para el reparto de direcciones, dentro de muy poco tiempo ya no quedarán direcciones IPv4 disponibles para asignar, lo que implicará inevitablemente que Internet no podrá crecer más. Con el fin de minimizar este problema, se diseñó la NAT, como ya hemos visto, que permite utilizar direcciones privadas para acceder a la red con una sola IP pública. En la actualidad, países como China o la India están haciendo un uso intensivo de la NAT por la falta de direcciones disponibles.

Como esta solución no es escalable y comporta una serie muy importante de problemas a los proveedores de servicios se llegó a la conclusión de que los 32 bits de direccionamiento del protocolo IPv4 eran insuficientes, por eso se diseñó el protocolo IPv6, como veremos a continuación.

Implantación del protocolo IPv6

Por motivos económicos no está implantado todavía IPv6, y si la demanda de direcciones IPv4 sigue al ritmo actual, se prevé que la IANA asignará el último rango de direcciones IPv4 a mediados del 2011 y que las autoridades regionales agotarán las que tienen pendientes de asignar en el 2012, lo que seguramente forzará a muchos países a adoptar prematuramente IPv6. En este sentido, países como Japón, China, la India y algunos de América del Sur, ya han adoptado el protocolo y utilizan algunas técnicas que permiten la interoperabilidad de los dos protocolos.

2.3.2. IPv6

La carencia de direcciones IPv4 incentivó el diseño de un nuevo protocolo de red, IPv6. En la actualidad, IPv6 está totalmente desarrollado, aunque todavía no es posible utilizarlo dentro de la red comercial, ya que los operadores todavía no han preparado sus equipos ni tampoco han hecho el reparto de direcciones a sus usuarios. Eso, y la dificultad de implantar progresivamente esta nueva versión y sustituir la anterior es lo que está retrasando su incorporación en el ámbito comercial.

Este subapartado describe brevemente este protocolo y destaca sus diferencias con la versión anterior, así como las novedades que incorpora. Para acabar el subapartado, se describen los principales problemas que hay para la migración de Ipv4 a IPv6.

Motivación

Inicialmente, a la hora de diseñar el protocolo, se pensó que no era necesario crear un protocolo entero y que sería suficiente con hacer una adaptación de IPv. Pero enseguida se vio que, para poder disfrutar de buenas optimizaciones en comparación con la versión anterior, harían falta bastantes más cambios. Así se optó por un diseño que tiene poco en común con la versión anterior.

El motivo principal que llevó a plantearse una nueva versión del protocolo era el limitado rango de direcciones que permite IPv4, que, aunque pueda parecer muy elevado, se vio que sería claramente insuficiente para cubrir la demanda del mercado en un futuro. Sobre todo, la aparición en los últimos años de una gran cantidad de dispositivos móviles que quieren formar parte de la gran red que es Internet ha provocado rápidamente que los 32 bits de direccionamiento IPv4 sea insuficiente, tanto es así que si todos estos dispositivos se quisieran conectar de forma simultánea en la red, los operadores probablemente tendrían problemas por la falta de direcciones IPv4. Para ver este problema sólo hay que pensar en cuántos teléfonos móviles hay en la actualidad; sólo en el Estado español hay alrededor de 44 millones, mientras que el número de IP que hay asignadas actualmente en el país es de unos 22 millones. Eso, sin contar los usuarios que se conectan desde sus hogares. Cabría pensar que el problema se podría minimizar con la utilización de NAT, pero, a la larga, eso puede suponer un grave problema de rendimiento en los encaminadores por tener que mantener las tablas de traducción de direcciones de millones de conexiones a la vez. Encima, cada vez hay más pequeñas y medianas empresas que quieren ofrecer a sus clientes una serie de servicios que precisan una

conexión permanente a la red, con el consecuente gasto de direcciones y la imposibilidad de utilizar la NAT masivamente. IPv6 soluciona este problema proponiendo un campo de direcciones de 128 bits.

Cabecera IPv6

La cabecera IPv6 tiene una longitud fija de 40 octetos (tal como se puede ver en la figura siguiente), y consta de los campos siguientes:

- **Version (4 bits)**: indica la versión del protocolo que contiene el paquete. Este campo tiene el mismo significado que el de la versión IPv4, pero ahora con el valor`0x06`.

- **Traffic Class (8 bits)**: este campo clasifica un paquete dentro de un tipo de tráfico determinado, conceptualmente es el equivalente del Type Of Service (TOS) de IPv4.

- **Flow Label (20 bits)**: sirve para etiquetar un conjunto de paquetes que tengan las mismas características; servirá para poder ofrecer calidad de servicio.

- **Payload Length (16 bits)**: longitud del *payload* del paquete, o sea, el paquete sin la cabecera IP. El tamaño viene representado en octetos.

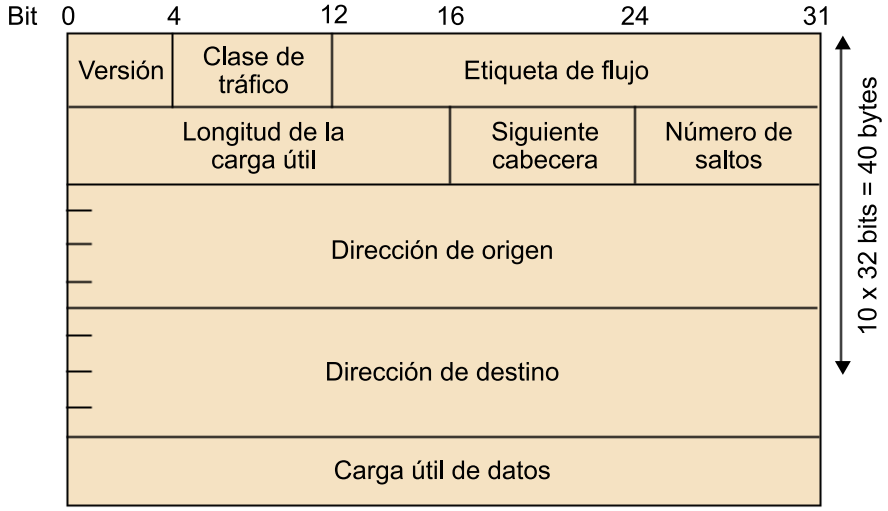
- **Next Header (8 bits)**: este campo es una gran innovación de IPv6 respecto de IPv4, dado que permite tener una cabecera básica de tamaño fijo. Este campo indica la posición en que se puede encontrar la siguiente cabecera, ahorrando así tiempo de proceso a los encaminadores intermedios al no haber opciones.

- **Hop Limit (8 bits)**: este campo es equivalente del TTL de IPv4, pero cuenta directamente *hops* y no tiempo.

- **Source Address (128 bits)**: dirección del *host* que ha originado el paquete.

- **Destination Address (128 bits)**: dirección del *host* al que va destinado el paquete.

Cabecera IPv6



A partir de los datos de la cabecera se puede observar que la diferencia más directa que hay entre ambos protocolos es la longitud de las direcciones IP, donde IPv4 tiene 32 bits, IPv6 pasa a tener 128. Este aumento en el espacio de direccionamiento permite que el rango de direcciones de la red pase de 2^{32} a 2^{128} direcciones posibles.

Como ahora tenemos muchos más bits para la dirección, la forma de especificar direcciones IPv6 se hace con la **notación de los dos puntos**. Donde una dirección IPv6 se representa con bloques de 16 bits representados en hexadecimal y separados por el símbolo ":". Por ejemplo, 2001:0DB8:0000:0000:0319:8A2E:0370:7348. Una simplificación de esta notación se puede aplicar en el caso de que una dirección tenga muchos 0 consecutivos; la forma abreviada de representarla es utilizando "::"; así, la forma compacta de representar la dirección anterior sería: 2001:0DB8::0319:8A2E:0370:7348.

Otra diferencia notable entre IPv4 e IPv6 es la jerarquización de las direcciones. La asignación de direcciones IPv4 se hizo, en su tiempo, de una forma muy anárquica, dado que no se esperaba que el crecimiento de Internet fuera tan espectacular. Actualmente, cada corporación u operadora de telefonía tiene rangos de direcciones muy dispersos y mal dimensionados, lo que hace extremadamente difícil la gestión de las direcciones disponibles, la asignación de las nuevas y el direccionamiento global. Por eso, lo que ha hecho IPv6 ha sido jerarquizar de una forma más inteligente el reparto de sus direcciones, de forma que cada país, operador o ISP, dispone de un rango concreto con un número de direcciones proporcional a su posible utilización de la red. Independientemente de la mejora de esta jerarquía en cuanto a localización geográfica, el hecho de separar de esta manera las direcciones permite asignar nuevas de una forma mucho más sencilla que hasta ahora.

De forma parecida a IPv4, se puede identificar qué tipo de dirección es sólo con el prefijo de la dirección IPv6, como indica la siguiente tabla.

Asignación de direcciones

Prefijo	Espacio de asignación
0000::/8	Reservado. Las direcciones de *loopback* y las direcciones con integración de IPv4 salen de este prefijo
0100::/8	Reservado
0200::/7	Reservado
0400::/6	Reservado
0800::/5	Reservado
1000::/4	Reservado
2000::/3	Dirección unicast global. De aquí sale el rango de direcciones que se repartirán a los usuarios. Hay 2^{125} direcciones disponibles.
4000::/3	Reservado
6000::/3	Reservado
8000::/3	Reservado
A000::/3	Reservado
C000::/3	Reservado
E000::/4	Reservado
F000::/5	Reservado
F800::/6	Reservado
FC00::/7	Dirección unicast local única
FE00::/9	Reservado
FE80::/10	Dirección de enlace local unicast (link-local)
FEC0::/10	Reservado
FF00::/8	Direcciones multicast

Un hecho muy interesante que se consideró para hacer esta asignación de direcciones es que da la posibilidad de representar direcciones de diversas tecnologías incrustadas dentro de la nueva versión del protocolo. De esta manera, se pueden representar direcciones IPv4, e incluso direcciones hardware del enlace de datos (como, por ejemplo, Ethernet).

La gran ventaja de insertar otros tipos de direcciones directamente en IPv6 es que tienen un prefijo asignado; así, por ejemplo, para tener una dirección Ethernet de un equipo dentro de un IPv6 el prefijo LAN es FE80::/10. Por tanto, si la dirección de la tarjeta Ethernet es 00:90:F5:0C:0F:ED, entonces, la dirección IPv6 queda así: FE80::0090:F50C:0FED. Como se puede observar, el pro-

ceso también se puede realizar a la inversa, cuando llega un paquete con el prefijo de red FE80 ya se puede suponer que se trata de una dirección local (*link local*) y se puede extraer la dirección hardware fácilmente. Además, con este mecanismo, cualquier interfaz de red puede configurarse de forma automática y autónoma.

Hay que señalar que IPv6, aparte de tener direcciones unicast, broadcast y multicast, como IPv4, añade soporte para un cuarto tipo, que son las direcciones anycast.

Las direcciones anycast son una gran innovación de IPv6, sobre todo porque aprovechan las direcciones unicast ya existentes. Así, una dirección unicast se vuelve anycast desde el momento en que se asigna una misma IPv6 a más de una interfaz (incluyendo equipos diferentes). La idea que hay detrás de esta implementación es que responda a las peticiones de un servicio concreto la estación más próxima.

Imaginemos dos servidores web con la misma IPv6, por ejemplo, 2001:0DB8::0319:8A2E:0370:7348, cuando la red reciba un paquete dirigido hacia esta IPv6, lo enviará a las dos estaciones, la primera que responda será la que esté más cerca del equipo que hace la petición. Actualmente, por complejidades en la implementación de este tipo de direcciones, sólo se utilizan para encaminadores. Así, una subred puede tener más de un encaminador para salir a Internet usando el mismo prefijo, y cada equipo utiliza el que esté más próximo a la estación, que consigue de forma sencilla un sistema de balanceo de carga.

Otra innovación relevante que incorpora IPv6 es la utilización mucho más intensiva del tráfico multicast dentro de las redes locales, tanto es así que los equipos, por defecto, escuchan a direcciones multicast con el prefijo FF02::1:FF00:0000/104, con el fin de evitar la generación de tráfico broadcast que afecta a todos los equipos de la subred y no siempre es deseable.

Otras mejoras menores que introduce este protocolo son:

- **Mecanismo de opciones ampliado**: las opciones forman parte de una cabecera colocado entre la cabecera IP propiamente dicho y la cabecera de la capa de transporte. Esta forma de poner las opciones permite una gestión más simple de las cabeceras por parte de los dispositivos que tienen que tratar el paquete hasta que llega a su destino, permitiendo un sistema más simple y flexible.

- **Direcciones de autoconfiguración**: la asignación dinámica de direcciones ha sido sustancialmente mejorada con respecto a su predecesor. Uno de los motivos principales de eso es el hecho de que se puede añadir la dirección hardware dentro de la dirección IPv6; así, sólo con un prefijo dado por el dispositivo de direccionamiento más próximo y con la dirección hardware se garantiza una dirección única a escala mundial. Siempre

que se utilice el prefijo asignado por el operador a la hora de generar la dirección.

- **Facilidad para la asignación de recursos**: lo que con IPv4 era el Type of Service ahora se llama Traffic Class, aunque también se tiene la posibilidad de marcar flujos individuales, lo que da mucha más flexibilidad a la hora de marcar tráfico prioritario.

- **Capacidades de seguridad**: como la seguridad, hoy día, es un tema tan importante, IPv6 incluye características de autenticación y privacidad. Por defecto, IPv6 incluye funcionalidades nativas para la creación de redes privadas virtuales (VPN, en la abreviatura en inglés) a través de IPsec (RFC-4301), protocolo de cifrado de los datos en tiempo real que con IPv4 era opcional.

Actividad

Un PC con una tarjeta Ethernet con MAC 34:27:A4:6F:AE:53. El operador le proporciona el prefijo 2001:0A54:0039::/48. Indicad la dirección link-local y la dirección de autoconfiguración de este equipo.

Solución

La dirección link-local vendrá dada por el prefijo link-local, así la dirección será: FE80::3427:A46F:AE53. Mientras que la dirección de autoconfiguración sale del prefijo y del MAC, por lo tanto: 2001:0A54:0039::3427:A46F:AE53.

Problemas de la migración en IPv6

Uno de los motivos principales por los que todavía se trabaja con IPv4 es la dificultad que comporta la migración al nuevo protocolo. La incompatibilidad de las direcciones y de las cabeceras de ambos protocolos hace que la actualización a la nueva versión no sea fácil. También hay que tener en cuenta que las aplicaciones existentes sólo soportan el sistema de direcciones de IPv4, que para aceptar las nuevas direcciones se tiene que cambiar el código de la aplicación, y también todos los llamamientos al sistema de acceso a la red.

Aparte del nivel de aplicación, hay otro problema muy grave. Dada la gran diversidad de redes que forman Internet, hay funcionando equipos muy diversos, y no todos esos equipos de comunicaciones tienen soporte para el nuevo protocolo, por lo que se tiene que actualizar el sistema operativo de los encaminadores de la red, con el consecuente gasto económico y de tiempo que eso supone, cosa que muchas de las empresas no están dispuestas a aceptar (especialmente las grandes corporaciones estadounidenses, que son las que tienen suficientes direcciones).

Por último, a causa de la gran utilización que se hace actualmente de Internet, es un gran problema tener que parar todas las redes para hacer la migración. Así, el problema que se encuentra es el enorme gasto económico para las em-

presas que controlan todas sus transacciones a través de la red, lo que fuerza a hacer la migración de forma progresiva, transparente para los usuarios y sin dejar de ofrecer los servicios disponibles en ningún momento.

Mecanismos para asistir la transición

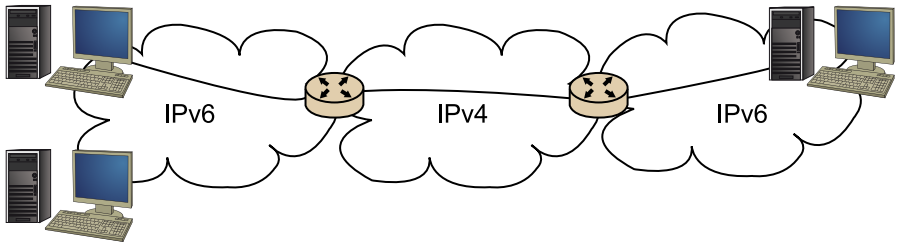
La migración entre dos protocolos cuando uno se está utilizando masivamente es extremadamente compleja. Como hemos visto, las razones, normalmente, son económicas, ya que la gran mayoría de las aplicaciones actualmente sólo soportan IPv4, y migrarlas a la nueva versión no siempre es sencillo (por ejemplo, aplicaciones bancarias). Por otra parte, todo el equipamiento hardware que forma el *backbone* de la red, si bien está preparado para soportar IPv6, no siempre tiene la configuración correcta, ni la asignación de direcciones hecha. En cualquier caso, se espera que haya una fase de coexistencia de los dos protocolos. De todas formas, a pesar de la coexistencia, hay escenarios que obligan a diseñar un plan de migración controlado. Así, los diferentes mecanismos de transición se pueden dividir en dos grandes grupos: mecanismos básicos y mecanismos para la interconexión de islas.

En cuanto a los mecanismos básicos, se pueden distinguir dos, el conocido como *Dual-Stack*, en el que los equipos utilizan simultáneamente los dos protocolos y se conectan con el que más se ajuste a las necesidades del momento, y el de Tunneling, en el que dos equipos con *Dual-Stack* crean un túnel IPv4 entre ellos, y por dentro del túnel se comunican con IPv6.

Hay que señalar que un túnel es el mecanismo por el que se encapsulan dos protocolos de red dentro de un mismo datagrama, así hay dos cabeceras de red consecutivas del nivel de red. IPv4 soporta el mecanismo de túnel a través de un valor especial en el campo *Protocolo* que se encuentra en la cabecera.

Por lo que respecta a los mecanismos para conectar islas, pretenden resolver el caso en el que diversas máquinas interconectadas a través de IPv6 (isla IPv6) se quieren conectar con otra isla IPv6, pero por el camino hay una isla IPv4, tal como muestra la figura siguiente.

Redes IPv4 y redes IPv6



Estos mecanismos también están basados principalmente en túneles. Se pueden distinguir los siguientes tipos:

- **Túneles configurados**: los extremos de los túneles entre las islas se configuran de forma manual entre las dos redes. Los extremos tienen que ser *dual-stack*.

- **Túneles automáticos**: se utilizan direcciones IPv4 mapeadas dentro de IPv6 con el prefijo reservado, que es el ::/96, por ejemplo, ::195.123.57.93, que el encaminador convierte a la dirección IPv4, y en el otro extremo se vuelve a convertir a la IPv6. Los extremos no se dan cuenta del cambio y se pueden comunicar con IPv6 sin problemas.

- **Túnel Broker**: se utiliza un *broker* (gestor), que indica al cliente un script con el fin de hacer el túnel de forma automática. El cliente tiene que ser *dual-stack*, ya que la petición en el broker va con IPv4, que contesta con un script que permite conectar a un Tunnel-Server (que también es *dual-stack*) al que permite conectarse a la red IPv6.

- **6to4**: se asigna una dirección IPv4 compatible con el prefijo IPv6 en los encaminadores, los cuales hacen un túnel. Por ejemplo, para la red 2001:d002:0507::/48 el encaminador tendría la dirección 208.2.5.7 (que sale de d002:0507). En el otro extremo se haría la operación análoga y se establecería el túnel.

2.4. Protocolos de soporte a IP

Tanto IPv4 como IPv6 son protocolos de red, pero ambos necesitan soporte de otros protocolos de la misma capa para poder llevar a cabo ciertas funcionalidades que sería complejo conseguir de otra manera. A pesar de que, dadas las diferencias estructurales entre los dos protocolos, muchos de los servicios son diferentes, todos se engloban en este subapartado para simplificar la comprensión, ya que, si bien los protocolos divergen, su funcionalidad a menudo es muy similar.

2.4.1. Internet Control Message Protocol

IP es un protocolo no orientado a conexión que tiene por objetivo el envío de información independientemente de la tecnología de niveles inferiores utilizada. Eso proporciona un entorno ideal para poder comprobar el estado de la red, o enviar información de control en el caso de que haya problemas en la red. Por ejemplo, cuando un datagrama no puede llegar a su destino, o bien hay congestión en un enlace, o bien a un paquete se le ha expirado el TTL. Todas estas situaciones requieren de algún protocolo, que trabajando en la misma capa que IP permita avisar de forma automática de cualquiera de estos eventos. Por eso se diseñó el Internet Control Message Protocol (ICMP).

El ICMP es el encargado de enviar mensajes de control (y de error) entre los diferentes equipos que forman la red.

La tabla siguiente muestra todos los tipos de mensajes existentes con ICMP.

Bibliografía

Para una descripción más completa de todos los tipos de mensajes existentes con ICMP se puede consultar el documento RFC-792.

Descripción de los diferentes mensajes ICMP

Mensaje	Descripción
Destination unreachable	Indica que no se puede llegar al destino. Este mensaje tiene un campo de código que indica si es culpa de la red, del equipo en particular o bien del puerto. También distingue si no se puede llegar al destino o bien si el destino es desconocido.
Echo request/ Echo reply	Estos dos mensajes son el de petición y el de respuesta, cuando una estación recibe un *echo request*, tiene que responder con un *echo reply* si está activa. En general es aconsejable que todos los equipos respondan estas peticiones, aunque por seguridad muchas veces se filtran los mensajes. La aplicación por excelencia que utiliza los *echo request/reply* es el `ping`.
Source quench	Este paquete sirve para regular, indica que aquel enlace está sufriendo congestión. Actualmente este tipo de paquete no se utiliza porque el control se hace mayoritariamente en la capa de transporte.
Router advertisement	Este paquete de ICMP se envía a una dirección multicast donde todos los encaminadores escuchan por defecto. Con este mensaje es posible descubrir automáticamente la existencia de nuevos encaminadores en la red.
Router discovery	Es un paquete complementario del de *router advertisement*. Pero en este caso es un encaminador que acaba de entrar en una red que pregunta qué otros encaminadores hay en la red.
TTL expired	Cuando el TTL de un paquete llega a 0 éste se descarta, y el encaminador que lo ha descartado genera un paquete TTL expired en el origen del paquete descartado.
IP header bad	En el caso de que se detecte un error de *checksum* en la cabecera de un paquete IP, éste se descarta y se avisa al origen con este paquete ICMP.

Los mensajes ICMP tienen varias utilidades, desde reportar errores hasta depurar el estado de la red. Una de las herramientas más utilizadas para poder ver si un equipo está conectado a la red es el `ping`. Otra funcionalidad es posible gracias al campo TTL de la cabecera IP, que ayuda a realizar otra tarea de depuración mediante una herramienta llamada `traceroute`; esta herramienta nos permite descubrir los encaminadores intermedios entre el origen y el destino de los datagramas. Para conseguir saber qué encaminadores cruza un datagrama, el `traceroute` envía paquetes IP consecutivos con un TTL de 1, otro de 2, otro de 3, y así sucesivamente hasta llegar al destino. El efecto de eso es que el primer encaminador, al recibir un paquete con TTL = 1, lo decrementa, y al ser 0, lo descarta y envía de vuelta un paquete ICMP de TTL *expired*. Ahora sólo hace falta que la aplicación mire a quien ha enviado este

paquete (IP origen) para saber de qué router se trata. Por descontado, con TTL = 2 sucederá lo mismo con el segundo encaminador, y después con el tercero, y así sucesivamente hasta el destino.

2.4.2. Address Resolution Protocol

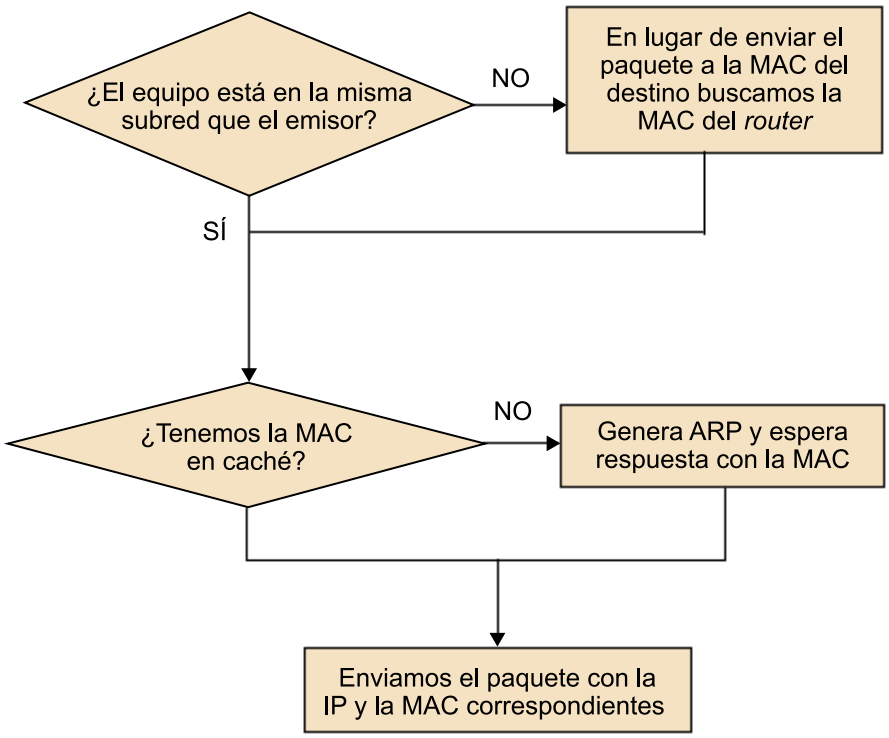
Para enviar un datagrama IP a una estación de la misma red que el emisor es necesario descubrir qué dirección del nivel del enlace de datos tiene esta estación, ya que las tecnologías de capas inferiores no entienden qué es una dirección IP. Así pues, para que la IP funcione necesita interactuar con las capas inferiores y descubrir de forma automática cuál es la dirección de enlace de datos a la que responde un equipo para poder intercambiarse información. El protocolo Address Resolution Protocol (ARP) fue diseñado específicamente para IPv4, y como veremos más adelante, para IPv6 existe el Network Discovery Protocol.

Dirección de nivel de enlace

La dirección de nivel de enlace es conocida como dirección MAC y viene dada por los dispositivos de red. Cada dispositivo de red tiene una dirección MAC única.

Para conseguir el descubrimiento de la dirección hardware de un equipo a partir de su IP, ARP se sirve de la funcionalidad que nos proporciona la capa de enlace de datos para enviar paquetes de broadcast. Entonces la procesa con el fin de conseguir la dirección hardware (llamada MAC) y poder enviar el paquete, tal como se puede ver en el diagrama de la figura siguiente.

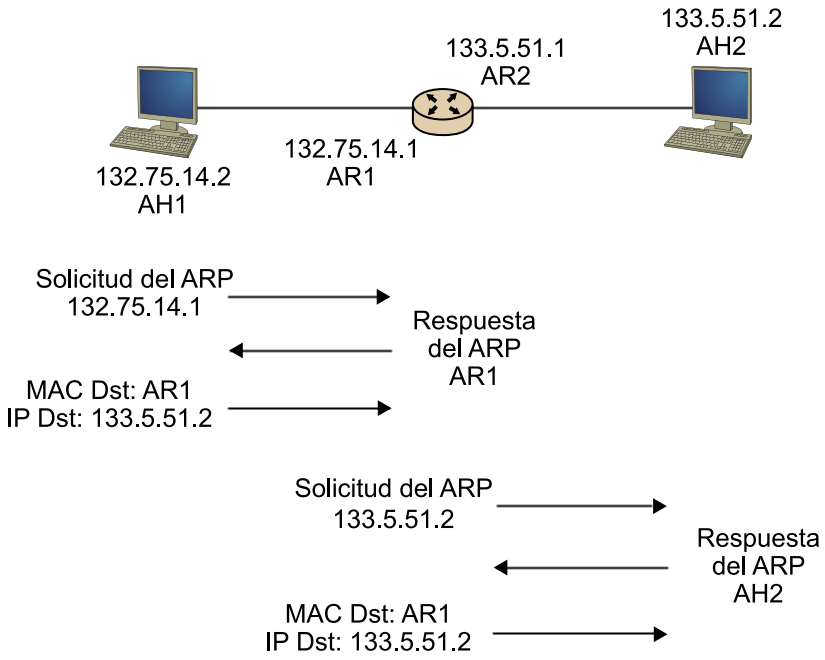
Diagrama de secuencia para el envío de un paquete IP



El proceso de ARP, ilustrado en la siguiente figura con un ejemplo, sigue una serie de pasos con el fin de descubrir la dirección. Lo primero que se tiene que conseguir es la dirección MAC del siguiente salto, por eso se envía un paquete ARP a la dirección de broadcast de nuestra red local. Este ARP buscará la IP del destino o bien la del encaminador dependiendo de si la estación forma

parte de la subred o no. Una vez se obtiene la dirección MAC, se construye un paquete que tiene como dirección MAC destino la obtenida por ARP y como IP destino la original (o sea, en el caso de que se envíe el paquete al encaminador, esta IP será la del destino final, no la del encaminador).

Ejemplo de petición por ARP



RARP

Hemos visto que ARP nos sirve para poder descubrir qué dirección MAC corresponde a una IP, también hemos visto la importancia de esta operación. Por compleción, ARP tiene una variante denominada RARP, que nos permite realizar la operación inversa, o sea, desde una dirección MAC poder averiguar a qué IP corresponde. RARP no es exactamente un protocolo de la capa de red, ya que incluye muchas funcionalidades de la capa de enlace de datos, pero, dada la estrecha relación con ARP, se acostumbran a considerar conjuntamente. RARP ya no se utiliza, ya que hay protocolos como BOOTP y DHCP que nos ofrecen esta y más funcionalidades como veremos a continuación.

2.4.3. Network Discovery Protocol

ARP es un protocolo que fue diseñado específicamente para IPv4; con los avances de las redes hasta hoy ha probado que es insuficiente para según qué servicios. Por eso, con la aparición de IPv6, se decidió que hacía falta un protocolo más completo. Así apareció el Network Discovery Protocol (NDP).

NDP es un protocolo que permite descubrir a los vecinos existentes en una red local. El modo de operar es muy similar al que utilizábamos con IPv4, ahora se envían *neighbour solicitations* y se reciben *neighbour advertisements*. Sin embargo, la gran diferencia con ARP es que se utiliza tráfico multicast en vez de broadcast. Y que NDP forma parte de un protocolo mayor llamado ICMPv6, que es la extensión en IPv6 del protocolo ICMP.

Cuando un nodo IPv6 se da de alta se pone a escuchar a un conjunto de direcciones multicast, una de ellas es la de Solicited-Node. Para automatizar este procedimiento, la dirección multicast Solicited-Node de una estación se construye de la siguiente manera: se cogen los últimos tres octetos de la dirección unicast y se añade al principio el prefijo multicast FF02::1:FF00:0000/104. Por ejemplo, la dirección multicast Solicited-Node para 2001:630:1310:FFE1:02C0:4EA5:2161:AB39 sería la FF02::1:FF61:AB39. Entonces, la estación se pone a escuchar al grupo multicast para responder con la dirección hardware del equipo al que va dirigida la petición.

Eso nos da la versatilidad que no todos los nodos reciben los anuncios, así en el caso de que no vayan dirigidos hacia ellos ya ni los llegan a ver, con la reducción en el uso de recursos que eso supone.

2.4.4. Dynamic Host Configuration Protocol

El siguiente protocolo de red que veremos no es realmente un protocolo de red, sino un protocolo de aplicación. De todas formas, dado que se utiliza para configurar la red, se explica en este subapartado. El Dynamic Host Configuration Protocol (DHCP) fue inicialmente definido en el documento RFC-1531. Es un protocolo que utilizan los dispositivos con el fin de obtener información de la configuración de los parámetros de red por un equipo IPv4 de forma automática.

El administrador de la red configura un prefijo de red, conjuntamente con un subrango de direcciones destinadas a autoconfiguración (este rango de direcciones se llama *pool*). Cuando se recibe una petición, el protocolo comprueba si el cliente está autorizado; si lo está se le asigna una IP preconfigurada, que se obtiene de una base de datos a partir de la dirección hardware del equipo, o bien una al azar del *pool* si la dirección hardware no se encuentra. Esta cesión de IP va controlada por un temporizador; cuando este temporizador expira y no se ha recibido ninguna noticia del cliente se devuelve la IP al *pool* de direcciones libres. Para evitar eso, el protocolo implementa un sistema de Keep-Alive, que va enviando renovaciones de uso de la IP al servidor para evitar que caduquen. Para hacer todas estas tareas DHCP utiliza UDP para enviar la información.

Entrando en un poco más de detalle, un cliente, cuando dé de alta una interfaz, enviará un DHCP discovery, que es un paquete broadcast con el fin de descubrir servidores DHCP. El servidor, cuando ve el paquete, comprueba la validez del cliente (base de datos de MAC) y envía un DHCP Offer con su IP. Al que el cliente responde directamente con un DHCP request, que finalmente el servidor acepta con un DHCP acknowledgement, que contiene la duración de la IP y la configuración específica que el cliente haya pedido al DHCP request. Por ejemplo, encaminador por defecto, servidor de DNS, etc.

3. El enlace de datos y el control de acceso al medio

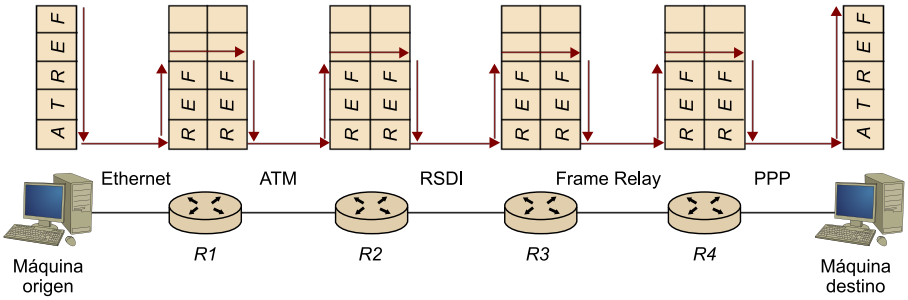
Hemos visto que la capa de red proporciona un servicio de comunicación entre dos máquinas, estableciendo diferentes rutas o caminos entre ellas. Cada ruta de comunicación está formada por una serie de enlaces, que conectan la máquina origen con la de destino utilizando unos dispositivos encaminadores intermedios. Cuando un datagrama del nivel de red sale de la máquina origen hacia la máquina destino, va atravesando cada uno de estos enlaces individuales que conforman el recorrido extremo a extremo.

Se hace necesaria una capa lógica adicional situada inmediatamente bajo la capa de red, que se ocupe de suministrar a ésta un transporte de información fiable entre los diferentes enlaces que atraviesa a lo largo de un recorrido. Esta capa recibe el nombre de nivel de enlace, y se sitúa por encima de la capa física. La capa física no es capaz de aportar ninguno de los elementos necesarios para la transmisión efectiva de información en un enlace.

El **nivel de enlace** consiste en dos programas o procesos que se ejecutan a ambos lados de un enlace y se comunican entre sí. Para que estos dos procesos se puedan comunicar es necesario establecer:

- un **formato** para la información que se intercambian, y
- un conjunto de reglas de comportamiento o protocolos necesarios para la transmisión de datos.

El principal cometido de la capa de enlace es conseguir que la comunicación de datos en un enlace se realice correctamente a través de un medio físico de transmisión. De una forma gráfica, podemos decir que el nivel de enlace se encarga de establecer y mantener un puente de comunicación entre dos nodos vecinos, para que por encima puedan circular los datagramas de nivel superior.



Ruta de comunicación creada entre dos máquinas finales, formada por 5 enlaces: 2 enlaces comunican las máquinas finales con los routers de la red y 3 enlaces internos intercomunican sólo routers de la red.

De la observación de la figura anterior, podemos destacar dos características muy importantes del nivel de enlace:

- Los enlaces, a lo largo de un recorrido de comunicación, pueden utilizar diferentes protocolos de la capa de enlace y estar constituidos por tecnologías de base totalmente diferentes. Un encaminador puede disponer de diferentes enlaces y cada uno de ellos puede utilizar un protocolo de nivel de enlace diferente. En la figura podemos observar cómo un datagrama enviado desde la máquina origen es manejado por Ethernet en el primer enlace, por el protocolo ATM en el segundo enlace, y va cambiando de tecnología sucesivamente en cada nuevo enlace.

- Una de las funcionalidades básicas del nivel de enlace consiste en encapsular/desencapsular los datagramas de la capa de red en PDU[5] de información de la capa de enlace, también llamados tramas. Observemos las flechas de la figura que indican el flujo que sigue la información a lo largo del recorrido. Cuando una trama llega al encaminador desde un enlace entrante, la capa de enlace desencapsula/extrae el datagrama de la trama recibida y lo entrega a la capa de red. Una vez la capa de red ha determinado el enlace de salida por donde debe encaminar el datagrama, lo envía al enlace. Aquí el datagrama es encapsulado según las normas del protocolo de este enlace y preparado para ser enviado.

(5)PDU son las siglas de Protocol Data Unit.

3.1. Terminología y definiciones

El **nodo** es la máquina o encaminador.

El **enlace** es el canal que conecta dos nodos adyacentes al recorrido de la comunicación.

El protocolo **de la capa de enlace** es la forma de comunicarse entre los nodos, para mover un datagrama sobre un enlace individual. Define el formato de los paquetes (PDU) intercambiados entre los nodos en los extremos del enlace, así como las acciones adoptadas por estos nodos cuando envían y reciben estos paquetes.

La **trama** son las unidades de datos intercambiadas por un protocolo de la capa de enlace. El nodo transmisor encapsula el datagrama de la capa en una trama de la capa de enlace y transmite la trama al enlace, y un nodo receptor recibe la trama y extrae el datagrama.

3.2. Tipo de enlaces

Básicamente podemos destacar dos tipos de enlaces:

- **Enlaces de comunicación punto a punto**. Sólo participan dos entidades o puntos. Son enlaces 1 a 1: compuestos por un único nodo emisor en un extremo del enlace y un único nodo receptor en el otro. Ambos nodos utilizan en exclusiva el enlace, sin compartir el canal.

  Ejemplos de enlaces punto a punto

  Son enlaces punto a punto:

  - Bucle de abonado local, cable de dos hilos telefónico para acceso a Internet.
  - Las redes de área local FastEthernet.
  - Las redes de área local GigabitEthernet.
  - PPP, HDLC (a nivel de enlace), X.25 a nivel de red y TCP a nivel de transporte (en este caso es además extremo-a-extremo).

- **Enlaces broadcast o canales de multidifusión**. Son enlaces 1 a N, donde una serie de nodos están conectados al mismo canal de comunicación. La transmisión realizada por un nodo la reciben todos los nodos conectados al enlace. Se hacen necesarias unas políticas de coordinación (o protocolos de acceso al medio) que permitan la compartición del único medio de forma eficiente, tratando de evitar al máximo las colisiones entre tramas.

  Ejemplos de enlaces broadcast

  Son enlaces broadcast:

  - Las redes de área local Ethernet (half duplex).
  - Las redes de área local sin cable Wifi.
  - Los enlaces con satélites.
  - Las redes de acceso híbrido fibra-cable (HFC).
  - Las redes de área local Token Ring.
  - Las redes de área local FDDI.

3.3. Tipo de servicios suministrado en la capa de red

El tipo de servicio que la capa de enlace suministra a la capa de red suele ser alguno de los siguientes:

- **Servicio no orientado a conexión y sin acuse de recibo**. El envío se hace sin esperar ninguna indicación del receptor sobre el éxito o fracaso de la operación. Tampoco se establece o libera una conexión. Este tipo de servicio es apropiado cuando la tasa de error es muy baja (redes locales o fibra óptica) y se deja la misión de comprobar la corrección de la transmisión a las capas superiores (nivel de red o de transporte). También se usa el servicio no confirmado cuando se quiere transmitir información en tiempo real (típicamente voz o datos) y no se quiere sufrir el retardo que impondría un servicio más sofisticado en la capa de enlace (se supone que este tipo de información puede sufrir una pequeña tasa de error sin efecto apreciable).

- **Servicio no orientado a conexión con acuse de recibo**. Se produce un acuse de recibo para cada trama enviada, aunque todavía no hay establecimiento dé conexión. De esta manera el emisor puede estar seguro de que ha llegado.

- **Servicio orientado a conexión con acuse de recibo**. Es el más seguro y sofisticado. El emisor y el receptor establecen una conexión explícita de antemano, se enumeran las tramas a enviar y se asegura que todas sean recibidas correctamente en el destino, y se transmiten seguidamente a la capa de red. En el servicio orientado a conexión se pueden distinguir tres fases: establecimiento de la conexión, envío de los datos, y finalización de la conexión. En la primera se disponen los contadores y memorias temporales necesarios para la transmisión, en la segunda se envían los datos, y en la tercera se libera la memoria ocupada con datos temporales y variables.

3.4. Servicios proporcionados por la capa de enlace

El servicio básico del nivel de enlace consiste en mover correctamente un datagrama de nivel de red, desde un nodo hasta otro adyacente sobre un enlace de comunicación fijando en el recorrido. Los posibles servicios que puede ofrecer un protocolo de la capa de enlace son:

- **Gestión de las tramas**. El nivel de enlace se encarga de la organización y gestión de las tramas.
  - Entramado o composición de la trama.
  - Sincronización a nivel de trama.
  - Transparencia de trama.
  - Numeración y secuenciación (números de secuencia).
  - Direccionamiento: si existe más de un posible destino de un mensaje es necesario identificarlo perfectamente.

- **Gestión del enlace**. Todo el proceso de inicio, mantenimiento y finalización de la transmisión requiere un considerable esfuerzo de gestión y coordinación.

- **Control de flujo**. La estación emisora y la receptora se tienen que poner de acuerdo sobre el ritmo de transmisión de datos. Si la estación receptora recibe las tramas más rápidamente de lo que es capaz de procesarlas, el nivel de enlace remoto ha de "frenarlas" para evitar que se sature la memoria intermedia o memoria temporal que almacena las tramas pendientes.

- **Control de errores**. Se trata de una de las funciones básicas del nivel de enlace. Se asume que el medio de transmisión subyacente no es perfecto e introduce errores de transmisión. Es necesario destinar una parte de los bits que se intercambian a la detección y la posterior gestión de los errores,

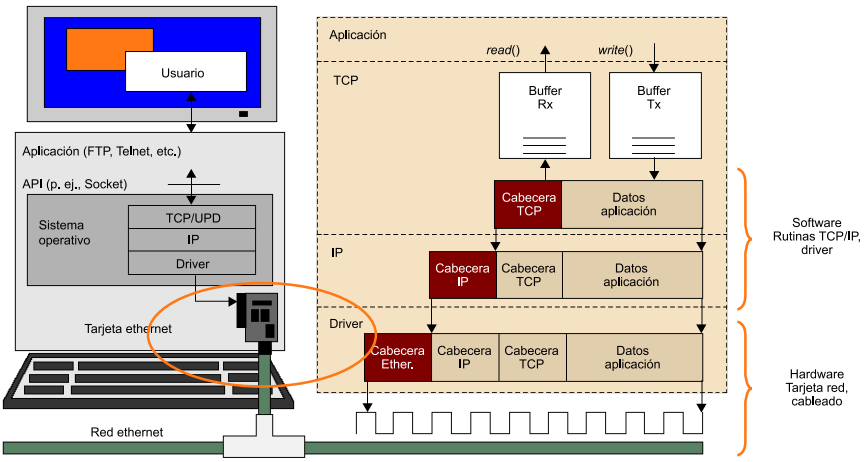
para controlar que no se produzcan errores de transmisión. En el control de errores se distinguen tres conjuntos de técnicas:

– Detección de errores. Utilización de códigos detectores de errores eficientes.
– Corrección de errores, si se utilizan códigos correctores adecuados.
– Retransmisión de tramas erróneas (entrega fiable).

- **Control de acceso al medio**. Esta funcionalidad cobra relevancia en los enlaces de acceso múltiple o broadcast, donde un número determinado de nodos comparten el mismo medio físico. El modelo OSI divide la capa de enlace en dos subcapas: la LLC (Logical Link Layer) y la MAC (Medium Access Control). La subcapa MAC es la encargada de especificar las reglas con que se transmite una trama sobre el enlace. Los protocolos MAC coordinan la transmisión de las tramas de los nodos, con el objetivo de evitar las colisiones de tramas. En los enlaces punto a punto los protocolos de acceso al medio dejan de tener sentido.

3.5. Adaptadores y dispositivos de red

Los nodos o encaminadores se conectan a los enlaces a través de un adaptador, conocido como tarjeta de interfaz de red o Network Interface Card (NIC). Físicamente, un adaptador es una placa hardware (una tarjeta PCMCIA o un dispositivo conectado al puerto USB) que contiene todos los elementos de un pequeño computador: memoria RAM, chip DSP, una interfaz de bus con la máquina, y una interfaz de enlace. Normalmente se encuentra alojado en la misma capa física que el resto del nodo, comparte la alimentación y los buses con el resto del nodo, y está en el fondo bajo el control del nodo.



Un adaptador tiene un cierto grado de autonomía:

- **En recepción**: cuando recibe una trama, determina si ésta tiene errores. Si es así, la rechaza sin notificarlo a su nodo padre. Si es correcta, desencap-

sulará el datagrama de la capa de red e interrumpirá a su nodo padre para pasarlo hacia arriba de la pila de protocolos.

- **En transmisión**: cuando un nodo pasa un datagrama hacia abajo en la pila de protocolos a un adaptador, delega totalmente en éste la tarea de transmitir el datagrama sobre el enlace. El adaptador encapsula el datagrama en una trama y transmite la trama en el enlace de comunicación.

Los componentes principales de un adaptador son la interfaz del bus y la del enlace. La interfaz del bus es responsable de comunicar con el nodo padre del adaptador. Transfiere datos e información de control entre el adaptador y el nodo padre. La interfaz del enlace es responsable de implementar el protocolo de la capa de enlace. También incluye el hardware de transmisión y recepción.

El protocolo de la capa de enlace está, mayoritariamente, implementado en este adaptador. Si el protocolo de la capa de enlace proporciona detección de errores, entrega fiable (numeración y reconocimientos) o acceso aleatorio (además de enmarcar y quitar el marco de datagramas), entonces estas funcionalidades están implementadas completamente en los adaptadores.

4. El nivel físico

Para acabar este módulo, estudiaremos la última capa del modelo OSI[6]. El nivel físico caracteriza la señal y su modulación, así como describe los medios de transmisión físicos.

(6)OSI es la abreviatura de Open Systems Interconnection; en castellano, interconexión de sistemas abiertos.

En este módulo no entraremos en los fundamentos matemáticos de la modulación y de la señal. Simplemente tenemos que saber que la información binaria se puede transmitir por un medio a través de las variaciones de alguna propiedad física, habitualmente, el voltaje o la intensidad. Podemos representar el valor de esta magnitud física como una función dependiente del tiempo. Dada esta función en forma de onda, podemos codificarla para transmitir la información que queremos. Este es el fundamento de la transmisión de información a través de un medio y sobre el que se basa toda la teoría de la comunicación.

4.1. Medios de transmisión

Existen varios medios de transmisión en función de su ancho de banda, retardo, coste, facilidad de uso, instalación y mantenimiento. Los medios se pueden clasificar en medios guiados (par de hilos, fibra óptica) o no-guiados (ondas de radio, láser a través del aire). Los siguientes factores de un medio de transmisión determinan la velocidad máxima de transmisión y la distancia máxima del medio:

- **Ancho de banda**. Al aumentar el ancho de banda se puede aumentar la velocidad de transmisión.

- **Atenuación**. En orden decreciente van el par trenzado, el cable coaxial y la fibra óptica.

- **Interferencias**. En orden decreciente van el par trenzado y el cable coaxial.

- **Número de receptores**. Atenúan y distorsionan la señal que supone una menor distancia.

4.1.1. Par trenzado

Se trata del sistema más antiguo y todavía es muy utilizado en la actualidad. Consiste en dos pares de hilos de cobre o de acero cubierto con cobre, aproximadamente de un milímetro de diámetro cada uno, que están envueltos entre sí en forma de hélice, como una molécula de ADN.

Se puedem utilizar tanto en las transmisiones digitales como en las analógicas. El ancho de banda que ofrecen depende del grueso del cable y de la distancia.

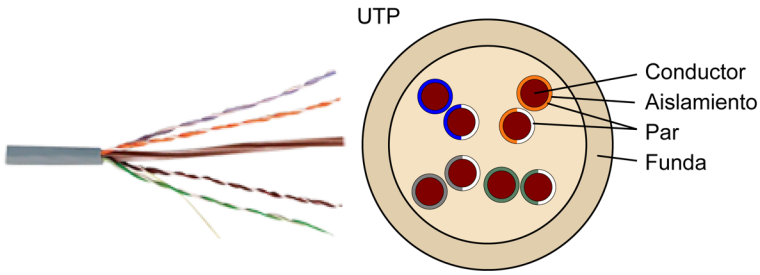
Coste del par trenzado

Debido a su bajo coste y su rendimiento, es un sistema muy popular. El coste es el más económico (más que el coaxial o la fibra óptica) por metro, pero tiene unos costes de conectividad parecidos a los de otros medios.

Es un sistema muy utilizado en los sistemas de telefonía. Generalmente, los teléfonos de las viviendas están conectadas con la centralita telefónica a través de pares trenzados (su ancho de banda es de 4 KHz). También se utiliza en las LAN a velocidades de 10, 100 y 1.000 Mbps. Es muy utilizado en las conexiones punto a punto, y su ámbito geográfico suele ser de unos 100 metros (en redes Ethernet).

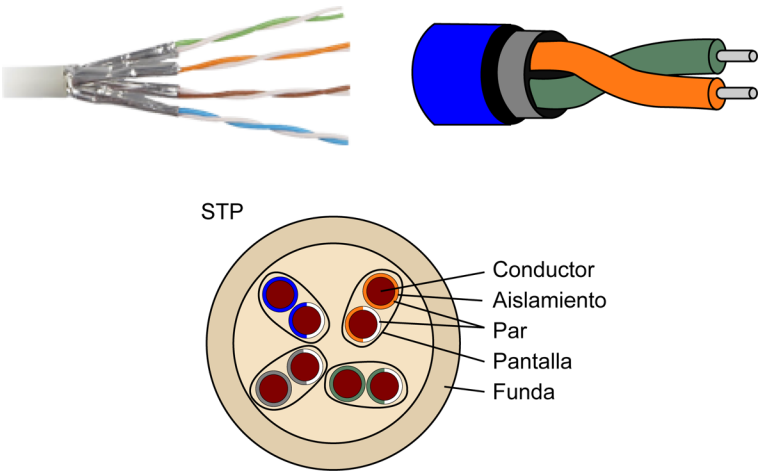
Los primeros pares trenzados no apantallados se llamaban Unshielded Twisted Pair (UTP). El par trenzado habitualmente se agrupaba en cuatro pares trenzados más dentro de una protección de plástico. Este tipo de cable se llamaba de categoría 3. Después se introdujeron los cables de categoría 5, agrupamientos con más densidad por centímetro de vueltas en el cable, que ofrecían unas características de mayor calidad sobre largas distancias.

Par trenzado no apantallado (UTP)



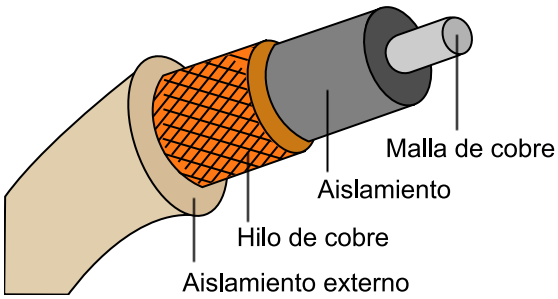
Después vino la evolución al tipo de cableado Shielded Twisted Pair (STP), llamado par trenzado apantallado, donde cada par de hilos tiene una protección individual.

Par trenzado apantallado (STP)



4.1.2. Cable coaxial de banda base

El cable coaxial tiene un recubrimiento superior a los pares trenzados, y funciona a largas distancias y altas velocidades. Este cable consiste en un hilo de cobre, rodeado por un material aislante. Para cables de 1 km de longitud, soporta velocidades de 1 a 2 Gbps. Se pueden utilizar distancias más largas pero con velocidades de transmisión más bajas. Este tipo de cable se ha utilizado para interconectar los equipos de las centrales telefónicas, para alguna red de área local o para la televisión por cable. Es más económico que la fibra óptica y más caro que el par trenzado.

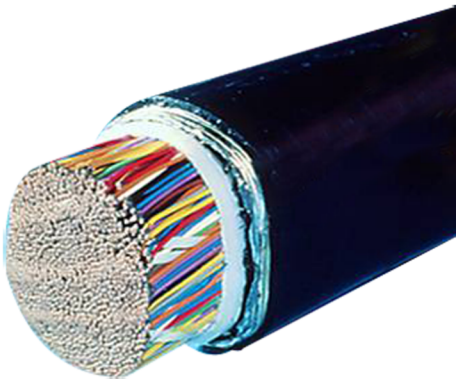


4.1.3. Fibra óptica

Un sistema de comunicaciones ópticas está formado por tres componentes: una fuente de luz, el medio de transmisión y el detector de luz. Convencionalmente, un impulso de luz indica un bit con valor 1, y la ausencia de luz indica un bit de valor 0. El medio de transmisión es una fibra óptica muy fina. El detector genera un impulso eléctrico cuando la luz incide sobre él. Así pues, instalando una fuente de luz al inicio de la fibra óptica y un detector de luz al final de la fibra, tenemos un sistema de transmisión unidireccional que acepta señales eléctricas, las convierte en impulsos de luz que se transmiten y se reconvierten en una señal eléctrica al final de la fibra.

La materia prima de la fibra óptica es el vidrio, un material no excesivamente caro y existente en grandes cantidades en nuestro planeta. El vidrio utilizado en las fibras ópticas es un vidrio transparente. La velocidad de transmisión es muy elevada, llegando a varios Gbps en 30 km de distancia.

Fibra óptica



Resumen

En este módulo se han visto en detalle las características y el funcionamiento de los diferentes niveles de la red. El objetivo del módulo ha sido dar una visión general del funcionamiento interno de una red de computadores, ya de área local, ya a gran escala, como es Internet. La aproximación a los niveles de la red se ha hecho desde los niveles más próximos a las aplicaciones hasta los niveles más específicos del hardware. En primer lugar se ha introducido el nivel de transporte que ofrece fiabilidad a la red permitiendo el acceso de diferentes aplicaciones a un único medio de transmisión, la red. Se ha visto que el funcionamiento de la red se rige en gran medida por el protocolo TCP, que es orientado a conexión y ofrece garantías de fiabilidad en la red. Para aplicaciones que no requieren garantías existe el protocolo UDP que es usado mayoritariamente por aplicaciones en tiempo real. Seguidamente se ha presentado la capa de red. El nivel de red es primordial para el funcionamiento de Internet pues nos permite dirigir e identificar los nodos de una red. En este apartado hemos visto IPv4 e IPv6. Se ha presentado el modelo de direccionamiento y la problemática actual con el número de direcciones en la red.

Se ha presentado la capa de enlace que abstrae los nodos de la red del medio físico sobre el que transmiten la información. Se ha visto que esta capa se encarga de dirigir la información entre nodos físicos adyacentes y garantizar transmisión fiable entre ellos. Hay que destacar la tarea de la subcapa de control de acceso al medio que permite transmitir información entre dos nodos independientemente de la tecnología de red utilizada, ya sea inalámbrica o cableada. Finalmente, se ha hecho una breve introducción al nivel físico que se encarga de los aspectos de modulación y codificación de las señales físicas que son dependientes directamente de la tecnología y el medio de transmisión. Se han visto los principales medios de transmisión.

Bibliografía

Almquist, P. (1992, julio). "RFC-1349: Type of Service in the Internet Protocol Suite". Consultant.

Bing, B. (2000). *High-Speed Wireless ATM and LAN*. Londres: Artech House. ISBN 1-58053-092-3.

Bing, B. (2000). *Broadband Wireless Access*. Dordrecht: Kluwer Academic Publishers. ISBN 0-7923-7955-1.

Croft, B.; Gilmore, J. (1985, septiembre). "RFC-951: Bootstrap Protocol (BOOTP)". Stanford University & Sun Microsystems,

Dijkstra, E. W. (1959). "A note on two problems in connexion with graphs". *Numerische Mathematik* (núm. 1, pág. 269-271).

Droms, R. (1993, octubre). "RFC-1531: Dynamic Host Configuration Protocol". Lewisburg: Bucknell University.

Gartner, F. C. (2003). *A Survey of Self-Stabilizing Spanning-Tree Construction Algorithms*. Lausana: Swiss Federal Institute of Technology Tech. Rep. IC/2003/38, School of Computer and Communication Sciences.

Hedrick, C. (1988, junio). "RFC-1058: Routing Information Protocol". Piscataway: Rutgers University.

IANA Reserved Multicast Address List. Disponible en web: <http://www.iana.org/assignments/multicast-addresses>. [Fecha de consulta: 12 abril de 2010.]

Information Sciences Institute (1981, septiembre). "RFC-791: Internet Protocol Specification". Marina del Rey: Information Sciences Institute, University of Southern California.

Intel Corporation (1999, 20 septiembre). "Preboot Execution Environment (PXE) Specification".

Kent, S.; Seo, K. (2005, diciembre). "RFC-4301: Security Architecture for the Internet Protocol". BBN Technologies.

Kurose, J.; Ross, K. (2005). *Computer Networking: a Top-Down Approach Featuring the Internet*. Amherst: Department of Computer Science, University of Massachusetts.

Malkin, G. (1998, noviembre). "RFC-2453: RIP Version 2". Bay Networks.

Moy, J. (1994, marzo). "RFC-1584: "Multicast Extensions to OSPF".

Moy, J. (1998, abril). "RFC-2328: OSPF Version 2". Ascend Communications.

Postel, J. (1981, septiembre). "RFC-792: Internet Control Message Protocol (ICMP)". ISI,

Rekhter, Y.; Li T.; Hares, S. (2006, enero). "RFC-4271: A Border Gateway Protocol 4 (BGP-4)".

Reynolds, J. (2002, enero). "RFC-3232: Assigned Numbers: RFC 1700 is Replaced by an On-line Database".

Reynolds, J.; Postel; J. (1994, octubre). "RFC-1700: Assigned Numbers".

Srisuresh, P.; Egevang, K. (2001, enero). "RFC-3022: Traditional IP Network Address Translator (Traditional NAT)". Intel Corporation,

Srisuresh, P.; Holdrege, M. (1999, agosto). "RFC-2663: IP Network Address Translator (NAT) Terminology and Considerations". Murray Hill: Lucent Technologies,

Tanenbaum, A. S. (2003). *Redes de computadores* (4.ª ed.). Nueva York: Prentice-Hall Professional Technical Reference.