

TCP (Transmission Control Protocol)

El Protocolo de Control de Transmisión es un protocolo de comunicación de la Capa de Transporte que proporciona un servicio de transporte de datos de flujo confiable. Se caracteriza porque a través de TCP, podemos obtener transmitir un flujo de bytes exacto entre 2 host que desean comunicarse, luego de establecer una conexión (donde un cliente informa el desea de transmitir datos al otro, y este último lo confirma) entre ambos equipos. Durante la transferencia, los software del protocolo en ambas máquinas continúan comunicándose para verificar que los datos recibidos son correctos. Los bytes son entregados en el mismo orden en que fueron enviados, sin duplicados, pérdida de paquetes ni errores. El protocolo adecúa el tamaño de envío de los datagramas para que sea eficiente sobre la red.

Para poder entregar los paquetes en orden y sin pérdida de datos se utiliza una técnica de "confirmación con retransmisión" en la cual el receptor confirma la llegada correcta de los mensajes con un ACK (acknowledgement), y el emisor no envía el siguiente paquete hasta que recibió esta confirmación de entrega. A la vez, el emisor dispara un timer al momento de enviar cada paquete de tal forma de que si este expira antes de recibir la confirmación ACK, retransmite el paquete hacia el destino y aguarda la nueva confirmación de recepción.

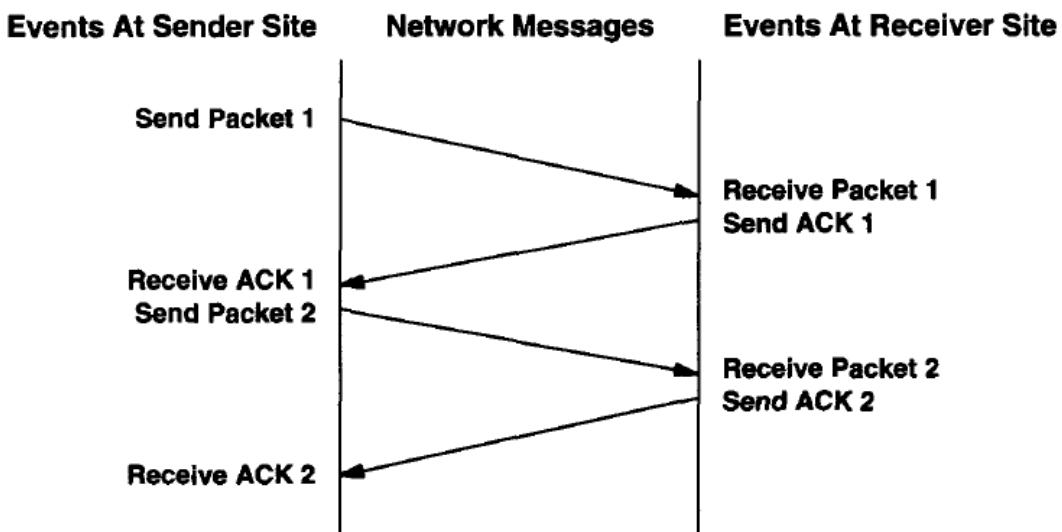


Figure 13.1 A protocol using positive acknowledgement with retransmission in which the sender awaits an acknowledgement for each packet

En combinación con esta técnica, para evitar paquetes duplicados y asegurar el correcto orden, usualmente los protocolos confiables asignan números de secuencia a cada paquete que el receptor devuelve en el ACK.

Para lograr una transmisión eficiente, TCP divide la secuencia de bytes a enviar en segmentos. Cada. El mecanismo de “ventana deslizable” permite enviar varios segmentos antes de recibir un ACK. Para esto ordena los bytes a ser enviados y utiliza 3 punteros asociados a toda conexión (para definir la ventana deslizable).

Para controlar el flujo en TCP, cada ACK de recibido especifica cuantos bytes han sido recibidos, y un mensaje “window advertisement” que especifica cuantos bytes el receptor está preparado para recibir (dependiendo del tamaño del buffer), por lo que el tamaño de ventana varía con el tiempo, permitiendo lograr un flujo eficiente.

Estructura de un Segmento TCP

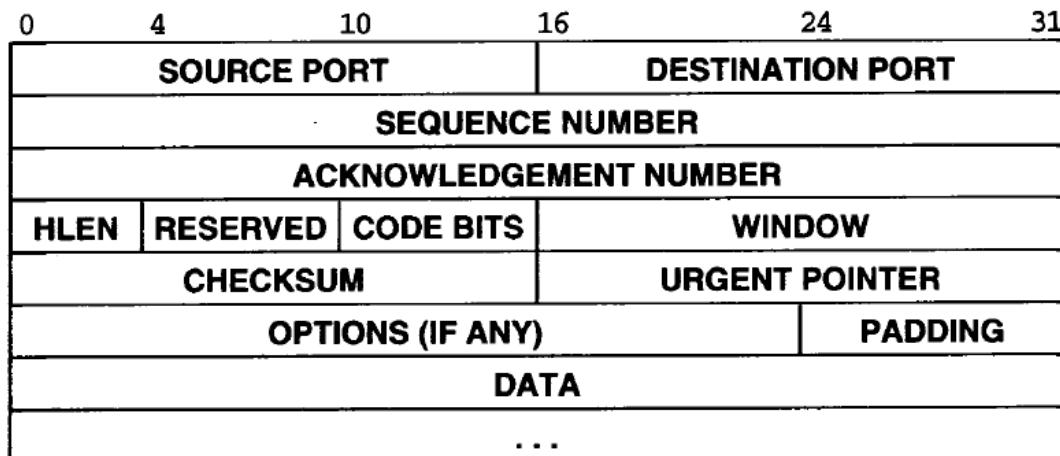


Figure 13.7 The format of a TCP segment with a TCP header followed by data. Segments are used to establish connections as well as to carry data and acknowledgements.

Each segment is divided into two parts, a header followed by data. The header, known as the *TCP header*, carries the expected identification and control information. Fields *SOURCE PORT* and *DESTINATION PORT* contain the TCP port numbers that identify the application programs at the ends of the connection. The *SEQUENCE NUMBER* field identifies the position in the sender's byte stream of the data in the segment. The *ACKNOWLEDGEMENT NUMBER* field identifies the number of the octet that the source expects to receive next. Note that the sequence number refers to the stream flowing in the same direction as the segment, while the acknowledgement number refers to the stream flowing in the opposite direction from the segment.

El “tipo” de segmento TCP (la carga de datos que lleva) se define según el campo **CODE BITS**.

Some segments carry only an acknowledgement while some carry data. Others carry requests to establish or close a connection. TCP software uses the 6-bit field labeled **CODE BITS** to determine the purpose and contents of the segment. The six bits tell how to interpret other fields in the header according to the table in Figure 13.8.

Bit (left to right)	Meaning if bit set to 1
URG	Urgent pointer field is valid
ACK	Acknowledgement field is valid
PSH	This segment requests a push
RST	Reset the connection
SYN	Synchronize sequence numbers
FIN	Sender has reached end of its byte stream

Aunque TCP sea un protocolo con conexión orientado a un flujo de datos confiable (en orden), a veces es importante poder enviar información por fuera de ese flujo, normalmente prioritaria, es decir que no deba esperar a que se procesen los segmentos previamente enviados antes de su atención. Por ej, sería muy útil que las señales (interrupciones o “aborts”) pudieran tener prioridad para recuperar el control de un programa que lee la entrada en la máquina destino, y que se encuentra “colgado”. Al tener prioridad y pasar por sobre el flujo de los segmentos TCP ya enviados, podría abortar el programa en cuestión para recuperar su control. Esto se realiza seteando los **CODE BITS** de la cabecera TCP en **URG**, con lo cual, el segmento enviado se considera “fuera de banda”, y se procesa al recibirse lo antes posible, independientemente de su posición en el flujo de datos. Además de marcarse URG en los code bits, se llenará el campo **URGENT POINTER** de la cabecera TCP, de forma de que especifique la posición en el segmento donde terminan los datos urgentes.

No todos los segmentos enviados tendrán el mismo tamaño. Al igual que la longitud de la cabecera TCP, pueden variar. Para negociar la longitud de los segmentos con el otro extremo de la conexión, TCP utiliza el campo **OPTIONS**.

El campo **CHECKSUM** de la cabecera TCP se utiliza para verificar la integridad de los datos como así también de la cabecera TCP

“En el campo **acknowledgment (ACK)** de un segmento TCP se especifica el número de secuencia del siguiente octeto que el receptor espera recibir.”

La forma de considerar cuando hacer **expirar un timer** al no recibir un ACK esperado, depende de las características de la red (que varían momento a momento dependiendo del tráfico), y se suelen utilizar algoritmos de retransmisión adaptativos para encontrar el timeout óptimo de forma que las transacciones no se enlentecan tanto sin requerir un número considerable de retransmisiones (lo que acabaría por ocupar una mayor capacidad del canal)

Compartiendo puertos en conexiones TCP

Ej: los servidores web utilizan por defecto el puerto TCP 80 y pueden recibir conexiones simultáneas.

Puede parecer extraño que dos conexiones utilicen al mismo tiempo el puerto TCP 53 en la máquina 128.10.2.3, pero no hay ambigüedad. Debido a que el TCP asocia los mensajes entrantes con una conexión en vez de hacerlo con un puerto de protocolo, utiliza ambos puntos extremos para identificar la conexión apropiada. La idea importante que se debe recordar es:

Como el TCP identifica una conexión por medio de un par de puntos extremos, varias conexiones en la misma máquina pueden compartir un número de puerto TCP.

Estableciendo una conexión TCP

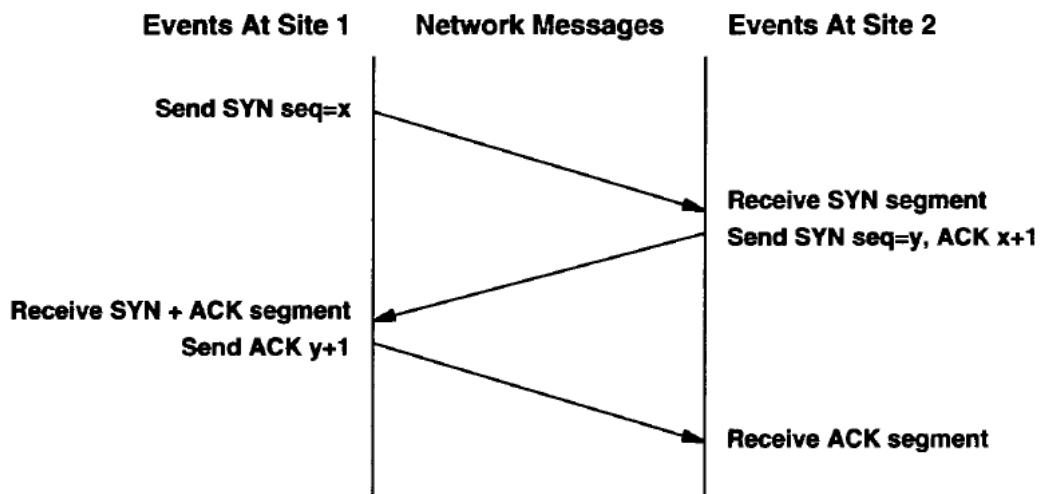


Figure 13.13 The sequence of messages in a three-way handshake. Time proceeds down the page; diagonal lines represent segments sent between sites. SYN segments carry initial sequence number information.

The first segment of a handshake can be identified because it has the SYN[†] bit set in the code field. The second message has both the SYN bit and ACK bits set, indicating that it acknowledges the first SYN segment as well as continuing the handshake. The final handshake message is only an acknowledgement and is merely used to inform the destination that both sides agree that a connection has been established.

Cuando 2 máquinas quieren establecer una conexión TCP, el emisor envía un segmento activando el bit SYN de los code bits de la cabecera TCP, y envía su Número de Secuencia Inicial "x" al destinatario. (Este número de secuencia se calculará aleatoriamente e identificará cada segmento del flujo entrante en el mismo sentido desde A, identificando así únicamente los paquetes enviados desde el emisor. Cada segmento futuro enviado desde A comenzará a contarse desde este número). Este almacena el número de secuencia para la conexión de A (x), y responde un segmento activando el bit SYN y enviando un propio (y también aleatorio) número de secuencia inicial (y) que identificará el inicio de la cuenta del flujo B → A en esta conexión. A la vez envía en el campo ACK un número equivalente a x aumentado en 1 (para confirmar así a A que se aceptó su número de secuencia inicial). Si todo está OK, A recibe este mensaje y confirma el establecimiento final de la conexión devolviendo un ACK a B de valor y + 1.

Cerrando una conexión TCP

Para terminar una conexión TCP, se usa la operación *close*. Recordemos que como las conexiones TCP son full dúplex, requiere cerrar ambos sentidos la conexión independientemente. Para hacerlo, el host que desea cerrar conexión envía un segmento con el bit **FIN** activado. Una vez recibido, se notifica al programa de aplicación y se envía un ACK al receptor, mientras se aguarda una respuesta (y quizás la intervención del usuario). Una vez que la aplicación confirma el cierre de la conexión, el receptor se envía un segundo ACK pero ahora con el bit FIN activado, señalando que ahora sí definitivamente, la conexión Emisor -> Receptor ha sido cerrada. Esto no afecta la recepción de mensajes en sentido Emisor <- Receptor hasta que este último finalice la conexión.

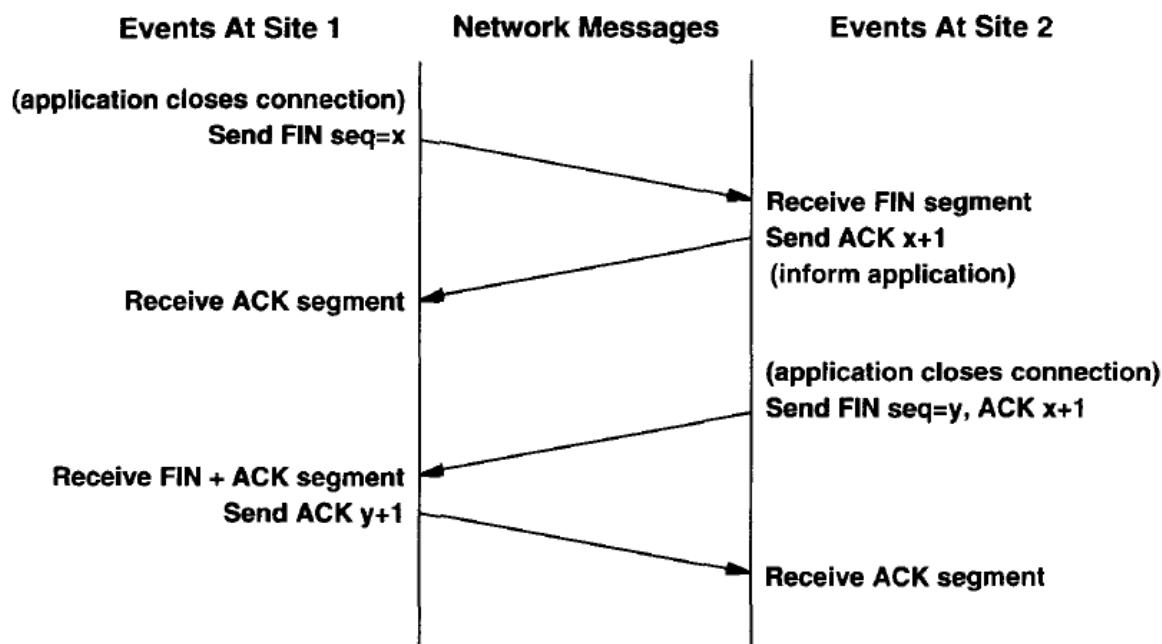


Figure 13.14 The modified three-way handshake used to close connections.

The site that receives the first FIN segment acknowledges it immediately and then delays before sending the second FIN segment.

Resetear una conexión TCP

Si la conexión TCP sufre una ruptura anormal, TCP permite resetear la conexión cuando uno de los interlocutores envía activado el bit **RST** de los code bits de la cabecera TCP. El receptor responde abortando la conexión. Esta es una forma de abortar la conexión en ambas direcciones inmediatamente. Los buffers y recursos que eran utilizados por la conexión se liberan.

UDP (User Datagram Protocol)

The User Datagram Protocol (UDP) provides an unreliable connectionless delivery service using IP to transport messages between machines. It uses IP to carry messages, but adds the ability to distinguish among multiple destinations within a given host computer.

An application program that uses UDP accepts full responsibility for handling the problem of reliability, including message loss, duplication, delay, out-of-order delivery, and loss of connectivity. Unfortunately, application programmers often ignore these problems when designing software. Furthermore, because programmers often test network software using highly reliable, low-delay local area networks, testing may not expose potential failures. Thus, many application programs that rely on UDP work well in a local environment but fail in dramatic ways when used in a larger TCP/IP internet.

12.4 Format Of UDP Messages

Each UDP message is called a *user datagram*. Conceptually, a user datagram consists of two parts: a UDP header and a UDP data area. As Figure 12.1 shows, the header is divided into four 16-bit fields that specify the port from which the message was sent, the port to which the message is destined, the message length, and a UDP checksum.

0	16	31
UDP SOURCE PORT	UDP DESTINATION PORT	
UDP MESSAGE LENGTH	UDP CHECKSUM	
DATA		
...		

12.10 Summary

Most computer systems permit multiple application programs to execute simultaneously. Using operating system jargon, we refer to each executing program as a *process*. The User Datagram Protocol, UDP, distinguishes among multiple processes within a given machine by allowing senders and receivers to add two 16-bit integers called protocol port numbers to each UDP message. The port numbers identify the source and destination. Some UDP port numbers, called *well known*, are permanently assigned and honored throughout the Internet (e.g., port 69 is reserved for use by the trivial file transfer protocol *TFTP* described in Chapter 26). Other port numbers are available for arbitrary application programs to use.

UDP is a thin protocol in the sense that it does not add significantly to the semantics of IP. It merely provides application programs with the ability to communicate using IP's unreliable connectionless packet delivery service. Thus, UDP messages can be lost, duplicated, delayed, or delivered out of order; the application program using UDP must handle these problems. Many programs that use UDP do not work correctly across an internet because they fail to accommodate these conditions.

Información Anexa

Teorema de Shannon:

Considerando todas las posibles técnicas de codificación de niveles múltiples y polifásicas, el teorema de Shannon-Hartley indica que la **capacidad del canal C** es:

$$C = B \log_2 \left(1 + \frac{S}{N} \right)$$

donde:

- B es el ancho de banda del canal en Hertz.
- C es la capacidad del canal (tasa de bits de información bit/s)
- S es la potencia de la señal útil, que puede estar expresada en vatios, milivatios, etc., (W, mW, etc.)
- N es la potencia del ruido presente en el canal, (mW, μ W, etc.) que trata de enmascarar a la señal útil.

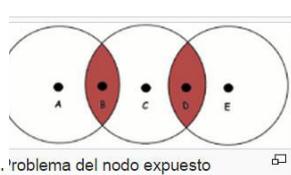
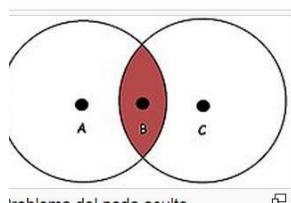
Problemas en redes inalámbricas:

Problemática en redes inalámbricas [\[editar\]](#)

En las redes inalámbricas proceder a la escucha del medio y por lo tanto detectar las colisiones producidas, puede resultar complicado. Esto se manifiesta en dos problemáticas:

- **Problema del nodo oculto:** una estación puede creer que el canal (medio) está libre cuando en realidad está ocupado por otra estación a la que no oye. En la siguiente imagen se muestra como A y C transmiten hacia B ya que ambos detectaron que el canal estaba libre. Sin embargo B escucha a ambos nodos, dando lugar a una colisión.
- **Problema del nodo expuesto:** una estación puede creer que el canal está ocupado cuando en realidad lo está ocupando otra estación que no interferiría en su transmisión a otro destino. En la figura se muestra como C está comunicándose con B. Como D detecta que el canal está ocupado, no puede transmitir hacia E, cuando lo idóneo sería que sí pudiese.

Estos problemas fueron resueltos con la implementación del protocolo [CSMA/CA](#) (MultiAccess Collision Avoidance)



Tramas en redes Ethernet:

Trama Ethernet				
cabecera Ethernet	cabecera IP (20 bytes)	cabecera TCP (20 bytes)	datos	checksum Ethernet

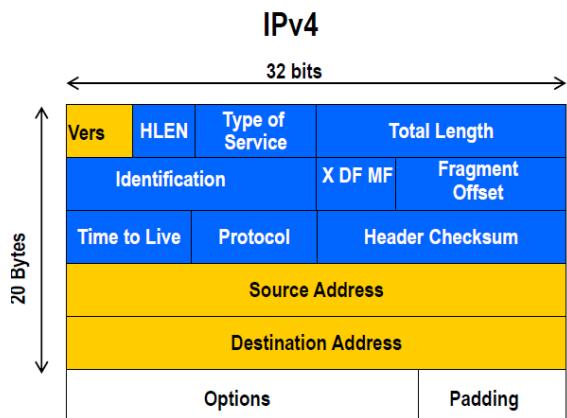
Y los principales campos que la forman son:

Campos de la trama Ethernet						
?	1	6	6	2	46-1500	4
Preámbulo	Inicio de delimitador de trama	Dirección Destino	Dirección Origen	Tipo	Datos	Secuencia de verificación de trama

- **Preámbulo:** Patrón de unos y ceros que indica a las estaciones receptoras que una trama es Ethernet o IEEE 802.3. La trama Ethernet incluye un byte adicional que es el equivalente al campo Inicio de Trama (SOF) de la trama IEEE 802.3.
- **Inicio de trama (SOF):** Byte delimitador de IEEE 802.3 que finaliza con dos bits 1 consecutivos, y que sirve para sincronizar las porciones de recepción de trama de todas las estaciones de la red. Este campo se especifica explícitamente en Ethernet.
- **Direcciones destino y origen:** Incluye las direcciones físicas (MAC) únicas de la máquina que envía la trama y de la máquina destino. La dirección origen siempre es una dirección única, mientras que la de destino puede ser de broadcast única (trama enviada a una sola máquina), de broadcast múltiple (trama enviada a un grupo) o de broadcast (trama enviada a todos los nodos).
 - **Tipo (Ethernet):** Especifica el protocolo de capa superior que recibe los datos una vez que se ha completado el procesamiento Ethernet.
 - **Longitud (IEEE 802.3):** Indica la cantidad de bytes de datos que sigue este campo.

Cabecera de Paquetes IPv4 (/datagramas IP, capa de red(3))

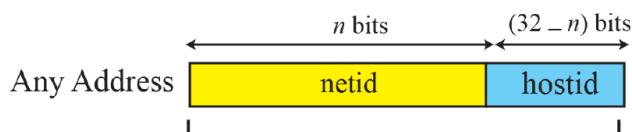
Cabecera IPv4: formato (C7)



IPv4: 12 campos fijos que pueden tener opciones => cabecera de tamaño entre 20, $5*32/8$, y 60 bytes

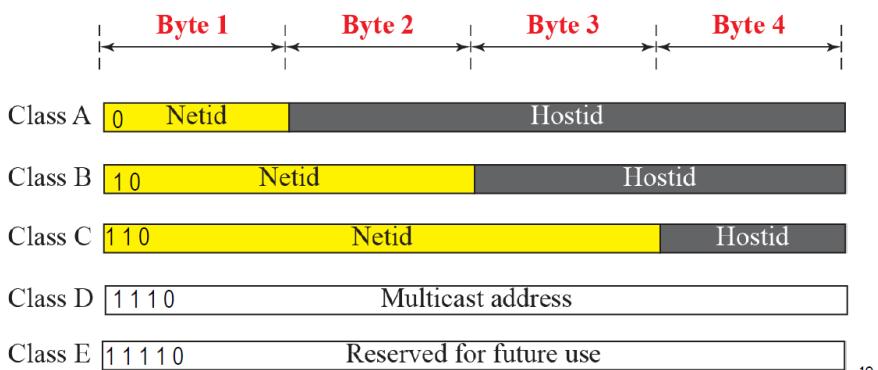
- Vers: 4.
- **Long. Cabecera (HLEN):** en palabras de 32 bits (mínimo 5, máximo 15)
- **Longitud total:** mide la cabecera + datos en octetos, máximo $2^{16} = 65535$ bytes.
- **Identificación, DF (Don't Fragment), MF, Desplaz. Fragmento:** campos de fragmentación; x bit sin uso
- **Tiempo de vida:** contador de saltos hacia atrás (se descarta cuando es cero)
- **Checksum:** de la cabecera (no incluye los datos)

Direcciones IP (C4)



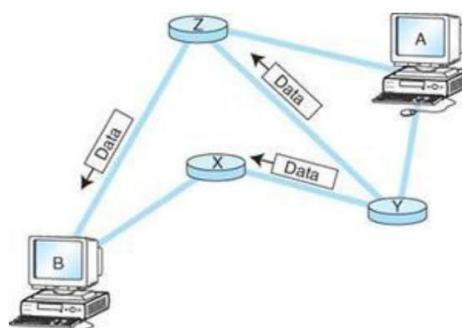
Class A: $n = 8$
 Class B: $n = 16$
 Class C: $n = 24$

Direcciones IPv4:



10

Ruteo de datagramas (C8)



Ruteo de paquetes I

Ruteo: Proceso de selección de un camino para el envío de paquetes, y el router es el dispositivo que realiza la selección. Existen dos formas:

- 1) Entrega directa
- 2) Entrega indirecta

8

1 vs 2: misma o distinta red?



Cómo sabe el Tx si el destino está en su misma red?

Network addresses are the key (netid,0)

- El Tx compara la porción de red de la IP de destino con la suya propia. **Misma dirección de red => entrega directa.**

2. Ruteo: entrega indirecta



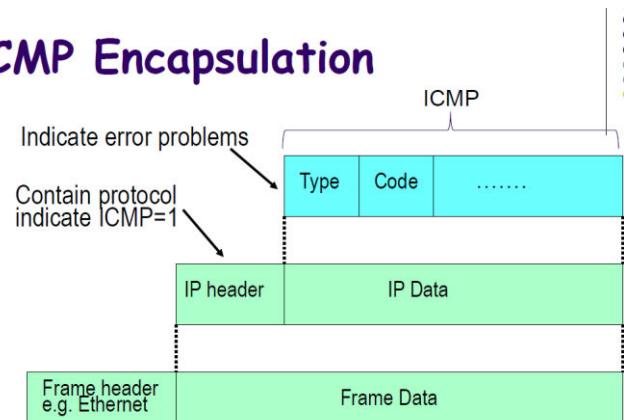
- El datagrama pasa de ruter a ruter hasta que llega a uno conectado directamente a la red destino
- ¿Cómo sabe un **host** a qué ruter enviar el datagrama?
- ¿Cómo saben los **ruters** la ruta por la que debe pasar el datagrama hasta llegar a la red destino?

Tanto los Host como los routers usan **tablas de ruteo** que tienen una entrada por cada posible red IP destino indicando:

- a) que la entrega es directa, o
- b) la IP del ruter que constituye el sgte salto en la ruta hasta el destino

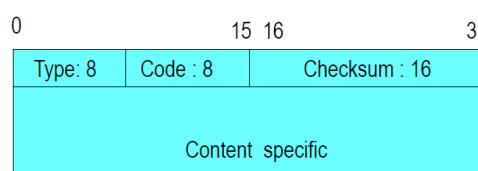
ICMP Encapsulation

Encapsulamiento de Mensajes ICMP
(Internet Control Message Protocol)



OBS: Un Mx ICMP no puede generar Mxs de errores ICMP

ICMP: Mx



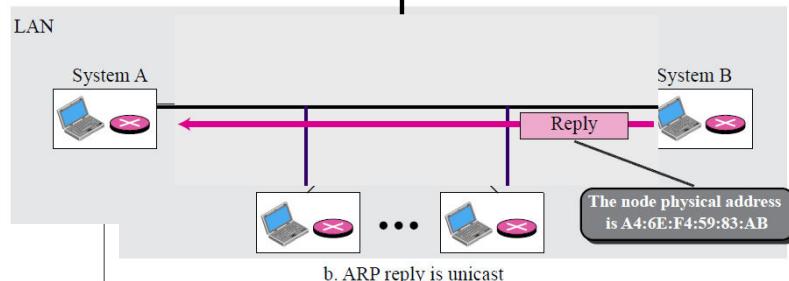
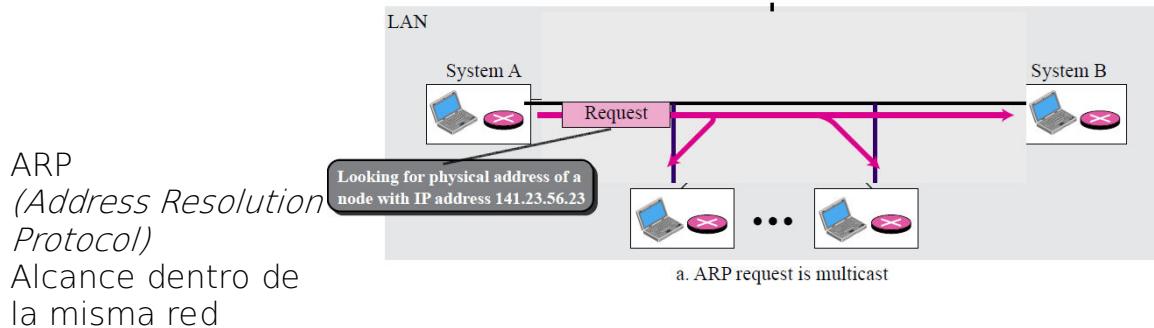
- **Type:** relevant ICMP message
- **Code:** more details information
- **Checksum:** covers ICMP header/data (not IP header)

Estructura de un ICMP
y códigos

Type	Code	Meaning
0/8	0	echo reply/echo request
3	0	network unreachable
3	1	host is unreachable
3	3	port is unreachable
4	0	source quench
5	0	redirect
9/10	0	router discovery/advertisement
11	0	time exceed
12	0	parameter problem
13/14	0	time stamp request/reply
17/18	0	network request/reply

The ICMP router discovery messages are called "Router Advertisements" and "Router Solicitations". Each router periodically multicasts a Router Advertisement from each of its multicast interfaces, announcing the IP address(es) of that interface. Hosts discover the addresses of their neighboring routers simply by listening for advertisements. When a host attached to a multicast link starts up, it may multicast a Router Solicitation to ask for immediate advertisements, rather than waiting for the next periodic ones to arrive; if and only if no advertisements are forthcoming, the host may retransmit the solicitation a small number of times, but then must desist from sending any more solicitations. Any routers that subsequently start up, or that were not discovered because of packet loss or temporary link partitioning, are eventually discovered by reception of their periodic (unsolicited) advertisements.

ARP operation



Implantación de ARP

- ARP realiza:
 - Transformación de dirección IP en dirección física.
 - Encapsulación de ARP packet into the frame
 - Responde solicitudes.

➤ Al inicio se realiza una consulta de una memoria intermedia ARP para ver si existe dirección física del destino. Si no, envía requerimiento ARP.

➤ Cuando una consulta ARP llega, extrae dirección IP y dirección física del transmisor. Si no existe esta información en su memoria intermedia lo almacenará.



El mensaje ARP se encapsula directamente en la trama física

Objetivos de Diseño (IETF) IPv6

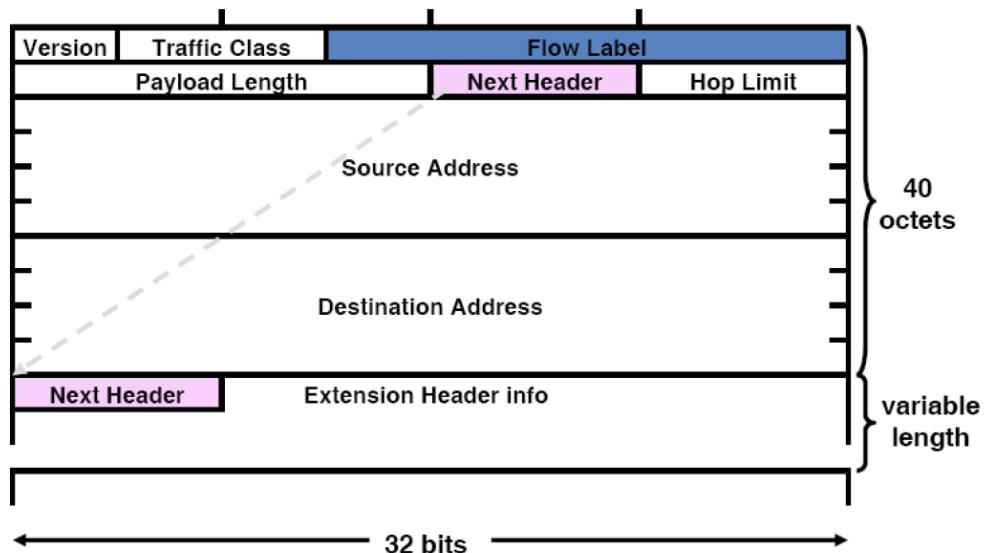
Objetivos de diseño de IPv6

Julio de 1999: Constitución oficial del IPv6 Forum.

- Datagrama Eficiente: Base + Extensión
- Direccionamiento: Mayor número de direcciones
- Fragmentación en origen-destino
- Identificador de flujos (QoS)
- Mecanismos que faciliten la configuración (plug-and-play)
- Seguridad incorporada: autenticación y cifrado
- Compatibilidad con IPv4

Cabecera Base-Extensión IPv6

Cabecera IPv6



Diferencias entre headers
IPv4 vs IPv6

Cuadro de diferencias de Headers

IPv4 Header 20 bytes+opt	IPv6 Header 40 bytes
Version	=
Internet Header Length	Removido, la longitud es fija (40 bytes)
Type of Service	Reemplazado por Traffic Class en IPv6.
Total Length	Reemplazado por el campo Payload Length, que sólo indica la longitud del payload.
Identification Fragmentation Flags Fragment Offset	Removido, la lÍx de fragmentación está en la cabecera de Fragmentation.
Time to Live	Reemplazado por el campo Hop Limit.
Protocol	Reemplazado por el campo Next Header.
Header Checksum	Removido puesto que la detección de bit-level Error es realizada en el paquete completo por la capa de enlace
Source Address 32 bits	Incrementado a 128 bits.
Destination Address	Incrementado a 128 bits.
Options	Reemplazado por cabeceras de extensión

IPv6: Cabeceras

● Resumen de Ventajas:

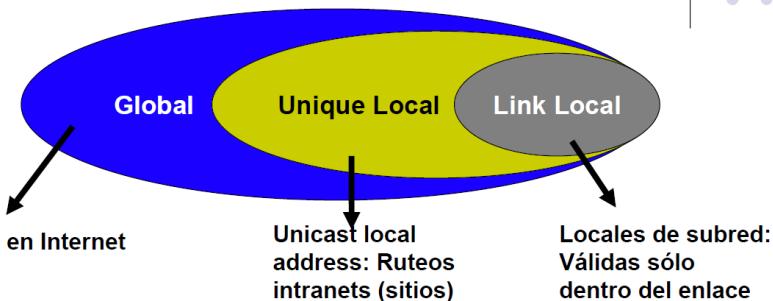
- Escalable en el número/tipo de opciones
- Procesamiento ordenado/eficiente
- Distribución de la complejidad
 - Cabeceras procesadas por elementos de red intermedios (**routers**)
 - Cabeceras procesadas en **destino**

Direcciones IPv6: Scope

OBS: Las direcciones se asignan a interfaces, no a nodos!

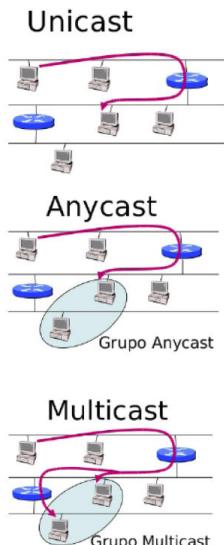
- Las interfaces pueden tener varias direcciones
- Alcance de las direcciones:
 - Link Local (locales a subred) FE80 :: /10
 - Unicast local address (locales a organización, sitios) FC00 :: /7
 - Globales 2000 :: /3

Direcciones: Alcance

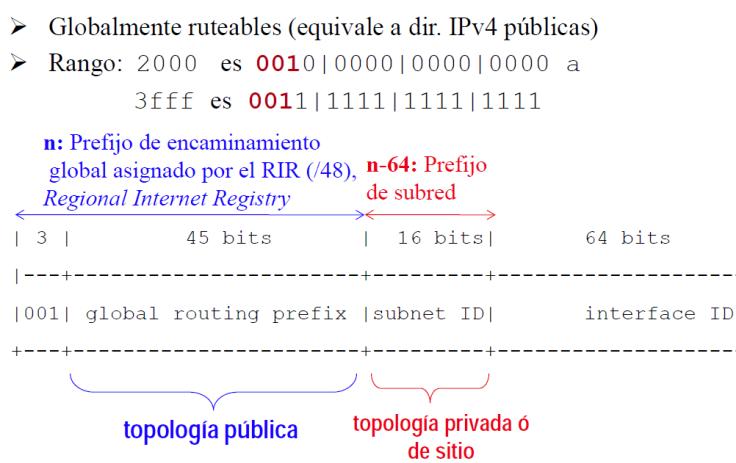


Comunicaciones

- **Unicast:** identifican a una interfaz de un solo nodo (de uno a otro)
- **Anycast:** identifican a un conjunto de interfaces, en general de nodos distintos (de uno a alguno ...)
- **Multicast:** identifican a un conjunto de interfaces (de uno a todos los del grupo ...)

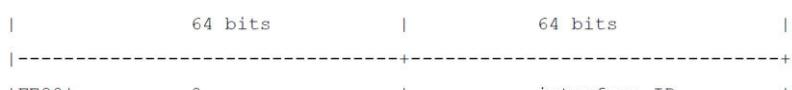


Unicast Global 2000::/3



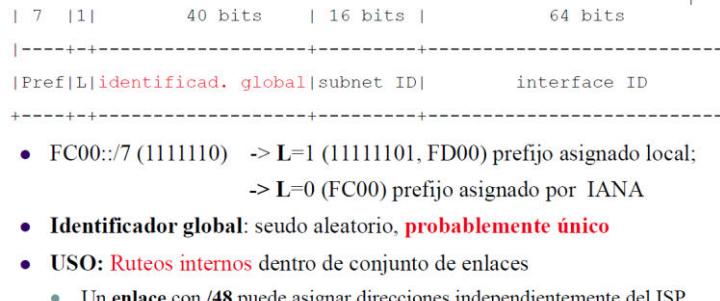
No routeables:

Unicast Link-Local



- Sólo válidas en el enlace local donde la interfaz está conectada

Routeables sólo dentro de la organización:
Unicast Unique-Local: Ex Sitio (ULA)



Unicast: Identificador de interfaz (IID)

- Deben ser únicos dentro del mismo prefijo de subred
- El mismo IID se puede usar en múltiples interfaces de un mismo nodo si están en subredes distintas
- Normalmente se usa un IID de 64 bits generado:
 - Manualmente
 - Autoconfiguración stateless
 - Basado en la MAC (Formato EUI-64)

Unicast: IID EUI-64

- Si la MAC es de 64 bits (EUI-64):
 1. Hacer el complemento del bit IID: U/L; 7mo bit más significativo ($0 \rightarrow 1$). En gral, U/L=0 (Universalmente administrada)
- Si la MAC es de 48 bits (IEEE 802):
 1. Agregar FF-FE en el medio (3er-4to byte) → EUI 64
 2. Cambiar el IID: bit U/L

Anycast

- Identifica un grupo de interfaces
- Asignadas a partir de direcciones *unicast* (igual sintaxis)
- Usos:
 - Descubrir servicios en la red (DNS, proxy HTTP, etc.);
 - Balanceo de carga;
 - Localizar routers que proveen acceso a una determinada subred;
 - Un paquete enviado a esta dir. es entregado al router más próximo al origen dentro de la misma subred.
- Todos los routers deben aceptar la dir. *Anycast Subnet-Router* formada por : **prefijo de la subred + el IID=0**
(ej., 2001:db8:cafe:dad0::/64)

Multicast

-
- The diagram shows a multicast IPv6 address structure with the following fields:
112 bits | group ID | flags | scope
+-----+-----+-----+-----+
| 11111111 | flgs | scop |
+-----+-----+-----+
- Identifica un grupo de interfaces.
 - Prefijo **FF** + 4 bits de *flags* + 4 bits que definen el alcance de la dirección *multicast*.
 - El soporte para *multicast* es obligatorio en todos los nodos IPv6.
 - La dirección *multicast* deriva del bloque **FF00::/8**.
 - Los 112 bits restantes se utilizan para identificar el grupo *multicast*.

Multicast Solicited-Node

- Todos los nodos deben formar parte de este grupo;
- Se forma agregando el prefijo **FF02::1:FF00:0000/104** a los 24 bits más a la derecha del IID;
- Utilizado por el protocolo de Descubrimiento de Vecinos (*Neighbor Discovery*).

Direcciones de una interfaz

address type	Binary prefix	IPv6 notation
nspecified	00...0 (128 bits)	::/128
oopback	00...1 (128 bits)	::1/128
multicast	11111111 (8 bits)	FF00::/8
link-Local unicast	1111111010	FE80::/10
unique-Local unicast	11111110	FC00::/7
global Unicast	001	2000::/3



ICMPv6

Internet Control Message Protocol

- Mismas funciones que ICMPv4 (pero no compatibles)
 - Informar características de la red;
 - Realizar diagnósticos;
 - Informar errores en el procesamiento de paquetes.
- Está después del encabezado base y extensión (si los hay)

- Utilizado para las funcionalidades IPv6:
 - Gestión de grupos multicast (Multicast Listener Discovery)
 - **Descubrimiento de vecinos** (Neighbor Discovery, ND)
 - Movilidad IPv6
 - Descubrimiento de la Path MTU (Maximun Transmisit Unit)

IPv6 Base	Extensión NH=58	ICMPv6
-----------	-----------------	--------

Tipo	Código	Checksum
Datos		

ICMPv6

Tipo	Nombre	Descripción	Tipo	Nombre	Descripción
1	Destination Unreachable	Indica fallas en la entrega del paquete (dirección o puerto desconocido) o problemas de comunicación.	128	Echo Request	Utilizados por el comando ping.
2	Packet Too Big	Indica que el tamaño del paquete es mayor que la Unidad Máxima de Transferencia (MTU) enlace.	129	Echo Reply	
3	Time Exceeded	Indica que el Límite de Direcccionamiento o de ensamblaje del paquete fue excedido.	130	Multicast Listener Query	Utilizados en la gestión de grupos <i>multicast</i> .
4	Parameter Problem	Indica un error en alguno de los campos del encabezado IPv6 o que el tipo indicado en el Siguiente Encabezado no fue reconocido.	131	Multicast Listener Report	
100-101		Uso experimental	132	Multicast Listener Done	
102-126		No utilizados	133	Router Solicitation	
127		Reservado para la expansión de mensajes ICMPv6	134	Router Advertisement	
			135	Neighbor Solicitation	Utilizados con el protocolo de Descubrimiento de Vecinos.
			136	Neighbor Advertisement	
			137	Redirect Message	
			138	Router Renumbering	Utilizado en el mecanismo de redirección(Renumbering) de routers.
			139	ICMP Node Information Query	
			140	ICMP Node Information Response	Utilizados para descubrir datos sobre nombres y direcciones, actualmente están limitados a herramientas de diagnóstico, depuración y gestión de redes.

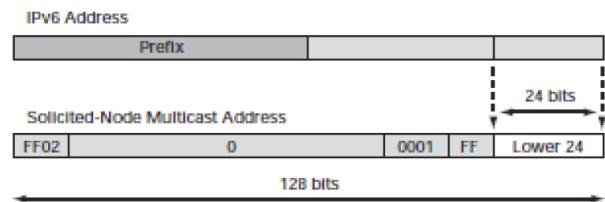
Multicast Solicited-Node

Descubrimiento de vecinos (RFC 4861)



- Utilizado por los *hosts* y *routers* para:
 - A. Determinar la MAC de los nodos de la red;
 - B. Encontrar routers vecinos;
 - C. Determinar prefijos y otros datos de configuración de la red;
 - D. Detectar direcciones duplicadas;
 - E. Determinar la accesibilidad de los routers;
 - F. Redireccionamiento de paquetes;
 - G. Autoconfiguración de direcciones

Figure 13: IPv6 Solicited-Node Multicast Address Format



- For example, the *solicited-node multicast address* corresponding to the IPv6 address 2037::01:800:200E:8C6C is **FF02::1:FF0E:8C6C**.

ND A- Descubrimiento de MACs

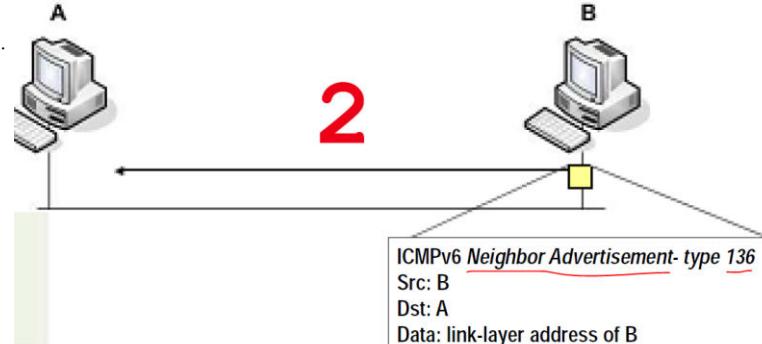


ND A- Descubrimiento de MACs

- **Determina la MAC de los vecinos del mismo enlace** (ARP IPv4)
- Usa dir. **Multicast solicited-node** en lugar de broadcast.

El vecino responde enviando un mensaje NA informando su dirección MAC.

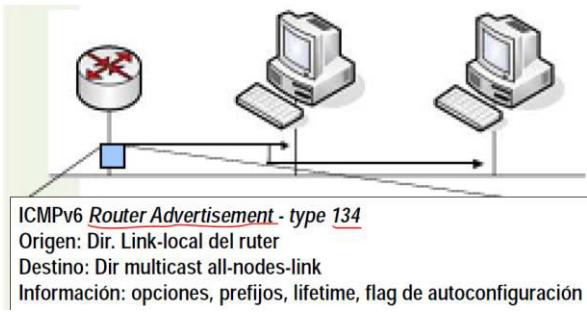
El host envía un mensaje NS: informa su MAC y solicita la MAC del vecino.



ND B- Descubrimiento de routers y prefijos: FF02::1

- Localizar routers vecinos dentro del mismo enlace.

- Determina prefijos y parámetros relacionados con la autoconfiguración de direcciones.
- Los routers envían periódicamente mensajes RA a multicast all-nodes scope link.



ND G-Autoconfiguración

Mecanismo seguido por un HOST para autoconfigurar interfaces IPv6

- Stateful:** provista por un servidor de dir. (DHCPv6)
- Stateless RFC 4862:** un HOST genera su propia IP a partir de información de los routers (Router Advertisment: prefijos asociados al enlace) y la dirección MAC.
 - Genera una dirección para cada prefijo informado en los mensajes RA.
 - Si no hay routers en la red solamente, genera una dirección *link local*: FE80::/64+IID(MAC).

20

Path MTU Discovery: PMTUD

- Path MTU Discovery – Busca garantizar que el paquete sea del mayor tamaño posible.
- Fragmentación – Permite enviar paquetes mayores que la MTU de un enlace.
 - IPv4 – Todos los routers pueden fragmentar paquetes mayores que la MTU del siguiente enlace => IPv4 puede fragmentar más de una vez durante su trayecto.
 - IPv6 – La fragmentación se realiza solamente en el origen.**
- Todos los nodos IPv6 deben soportar PMTUD.

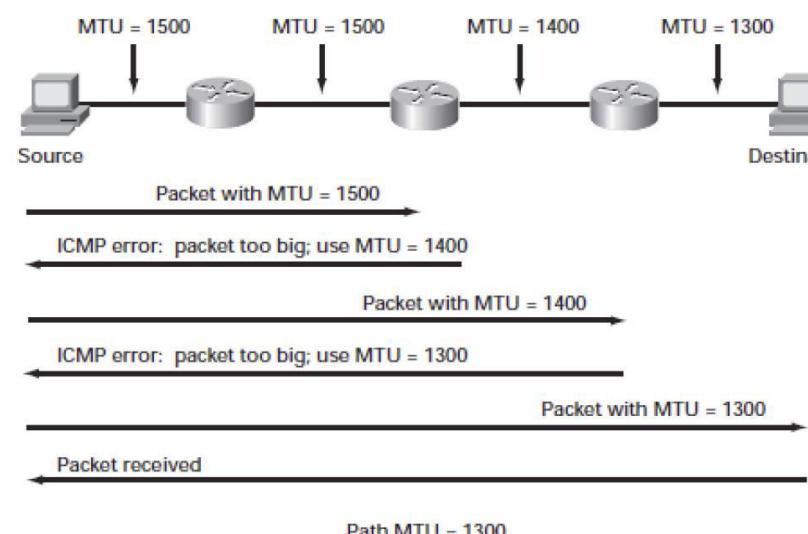
PMTU Discovery Process

- Send a message to the destination with MTU of your link;
- If receive a ICMP error message, then resend the message with the new MTU;
- Do 1 and 2 until response from destination;
- Last MTU is the Path MTU.

NOTA: Los paquetes enviados a un grupo multicast utilizan un tamaño igual a la menor PMTU de todo el conjunto de destinos

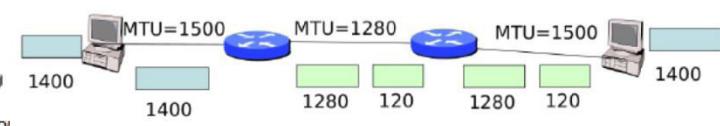
Ej. Fragmentación (ABCs of IP Version 6)

Figure 16: Path MTU Discovery

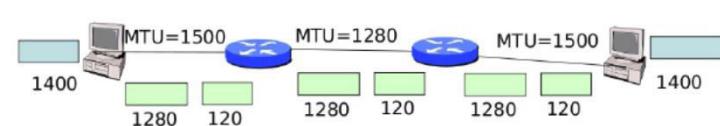


. Fragmentación

IPv4



IPv6



Calidad de Servicio:

- 2 campos para QoS:



- **Clase de Tráfico:** utilizado en DiffServ (DSCP)
- **Identificador de Flujo:** identifica flujos desde un origen a un destino con una determinada QoS

Modelos:

1. Best Effort Internet
2. Integrated Services (IntServ)
3. Differentiated Services (DiffServ)



Coexistencia y transición

Estas técnicas de transición se dividen en 3 categorías:

- **Doble pila**

- Provee soporte a ambos protocolos en el mismo dispositivo.

- **Tunelización**

- Permite el tráfico de paquetes IPv6 sobre la estructura de la red IPv4 existente.

- **Traducción**

- Permite la comunicación entre nodos que solo soportan IPv6 y nodos que solo soportan IPv4..

30

Doble Pila

Una red doble pila es una infraestructura capaz de encaminar ambos tipos de paquetes.

Exige analizar algunos aspectos:

- Configuración de los servidores de DNS;
- Configuración de los protocolos de enrutamiento;
- Configuración de los *firewalls*;
- Cambios en la administración de las redes.

Tunneling

- También llamado encapsulamiento.
- El contenido del paquete IPv6 se encapsula en un paquete IPv4.
- Se pueden clasificar de la siguiente manera:
 - *Router-a-Router*
 - *Host-a-Router*
 - *Router-a-Host*
 - *Host-a-Host*

Neighbor Discovery

Define varios mecanismos, entre ellos:

descubrir routers, prefijos y parámetros, autoconfiguración de direcciones, resolución de direcciones, determinación del siguiente salto, detección de nodos inalcanzables, detección de direcciones duplicadas o campos, redirección, balanceo de carga entrante, direcciones anycast, y anuncioación de proxies.

ND define cinco tipos de mensajes ICMPv6:

- **Solicitud de router** (Router Solicitation). Generado por una interfaz cuando es activada, para pedir a los routers que se anuncien inmediatamente. Tipo de mensaje ICMPv6 133 código 0.
- **Anunciación de router** (Router Advertisement). Generado por los routers periódicamente (entre cada 4 y 1800 segundos) o como consecuencia a una solicitud de router, a través de multicast, para informar de su presencia así como de otros parámetros de enlace y de Internet, comoprefijos (uno o varios), tiempo de vida, configuración de direcciones, límite de salto sugerido, etc. Es importante para permitir la reenumeración. Tipo de mensaje ICMPv6 134 código 0.
- **Solicitud de vecino** (Neighbor Solicitation). Generado por los nodos para solicitar la dirección en la capa de enlace de la tarjeta de su vecino, o para verificar que el nodo vecino es alcanzable, así como para detectar las direcciones duplicadas. Las solicitudes son multicast cuando el nodo necesita resolver una dirección y unicast cuando el nodo quiere verificar que el vecino es alcanzable. Tipo de mensaje ICMPv6 135 código 0.
- **Anunciación de vecino** (Neighbor Advertisement). Generado por los nodos como respuesta a la solicitud de vecino. Tipo de mensaje ICMPv6 136 código 0.
- **Redirección** (ICMP Redirect). Generado por los routers para informar a los hosts de un mejor salto para llegar a un destino. Tipo de mensaje ICMPv6 137 código 0.

Good Luck!! ☺ – © 2015 – “El CEO”