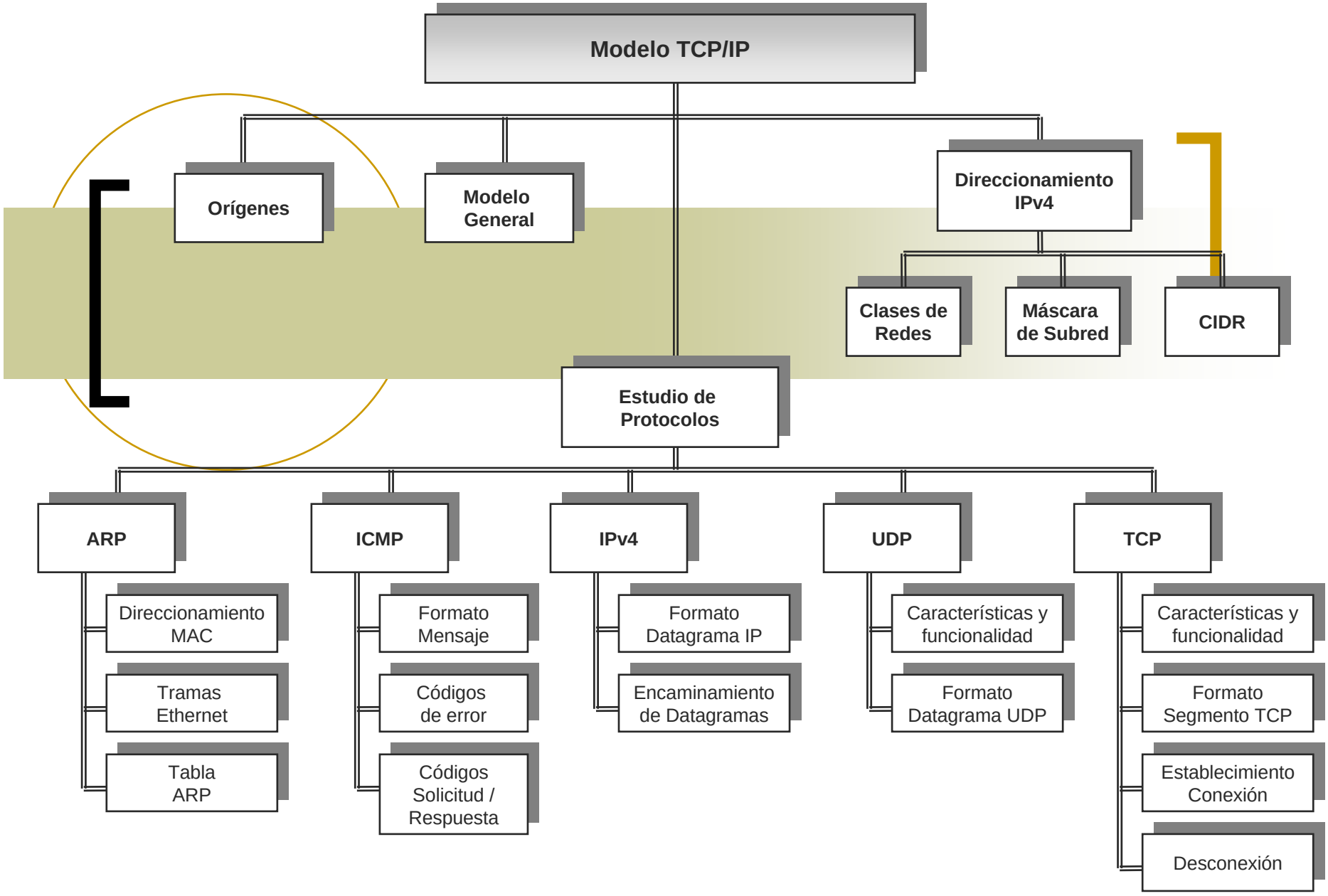

Modelo TCP/IP
Orígenes
Modelo General
Direccionamiento IPv4
Clases de Redes
Máscara de Subred
CIDR
Estudio de Protocolos
ARP
ICMP
IPv4
UDP
TCP
Direccionamiento MAC
Tramas Ethernet
Tabla ARP
Formato Mensaje
Códigos de error
Códigos Solicitud / Respuesta
Formato Datagrama IP
Encaminamiento de Datagramas
Características y funcionalidad
Formato Datagrama UDP
Características y funcionalidad
Formato Segmento TCP
Establecimiento Conexión
Desconexión

Arquitectura TCP/IP

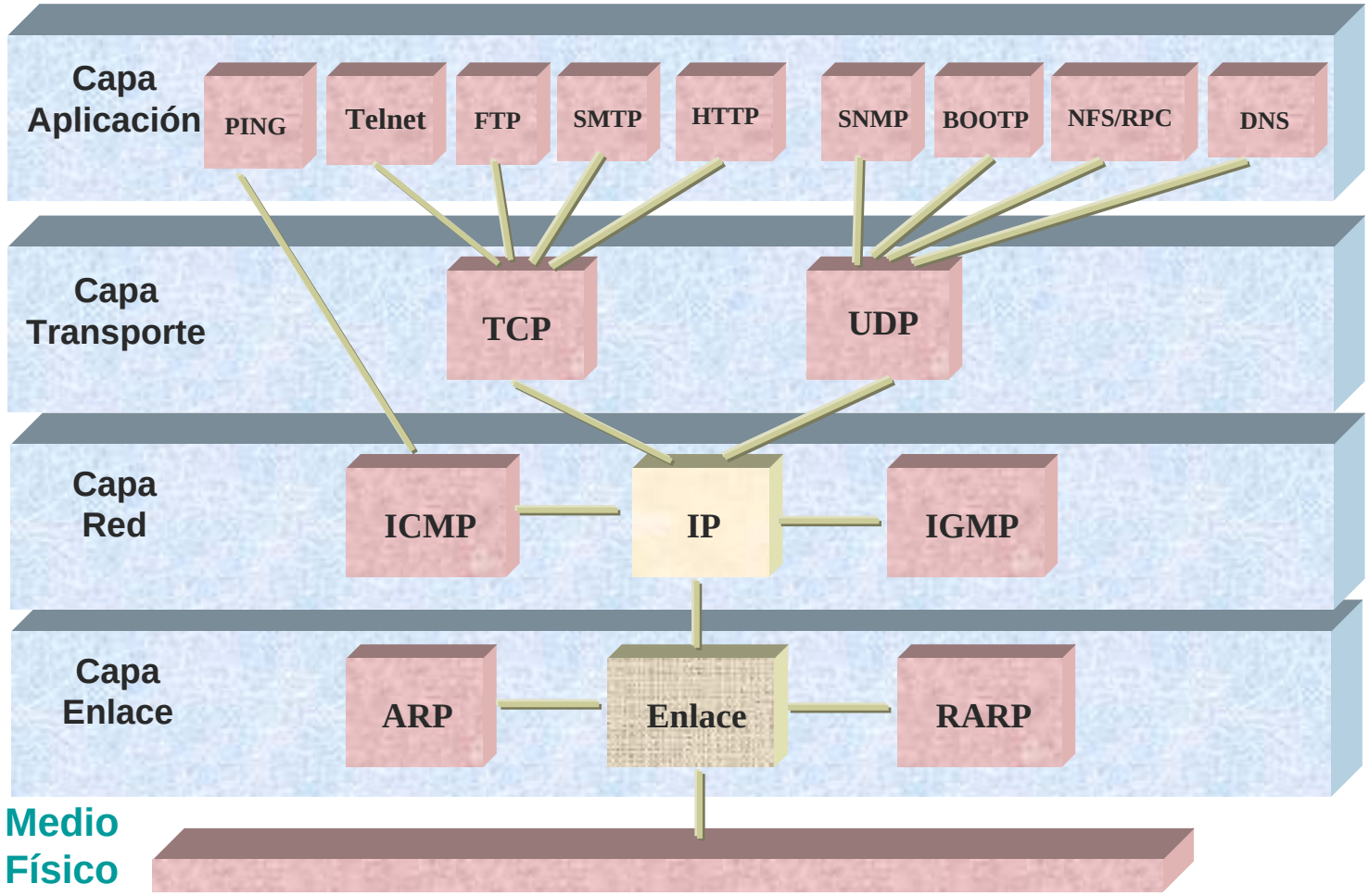
ÍNDICE

- Objetivo:
  - Describir y analizar la familia de protocolos TCP-IP

- ¿Qué vamos a estudiar?
  - Direccionamiento MAC/IP.
  - Protocolos ARP, RARP
  - Protocolo de señalización ICMP
  - Protocolo de red IP.
  - Protocolos de transporte TCP/UDP.

- Documentación adicional
  - ***RFC*** (Request For Comments): http://www.rfc-editor.org/rfc.html
  - "***TCP/IP Illustrated. Vol.1***" W.R. Stevens, Addsion Wesley

Familia Protocolos TCP/IP



Protocolos (1)

- El origen de esta familia de protocolos fue la red ARPANET (en ella se desarrollaron los conceptos fundamentales de diseño y gestión de redes)
- Los niveles más bajos (enlace y físico) no están implementados ya que se diseñó para no depender de una red física concreta
  - Los protocolos ARP (Adress Resolution Protocol) y RARP (Reverse Adress Resolution Protocol) se encargan de enlazar los sistemas de direccionamiento IP y el de la red física utilizada

Protocolos (2)

- Capa de Red:
  - La base de la familia de protocolos es el nivel de Red (***IP*** *Internet Protocol*). Es un protocolo de conmutación de paquetes muy sencillo, de tipo datagrama, de forma que se pueda implementar en cualquier tipo de máquina.
  - A diferencia de OSI, no se distingue claramente entre servicio, interfaz y protocolo (está entremezclado)
  - Existen actualmente dos versiones IPv4, IPv6
  - Protocolos de apoyo:
    - ***ICMP*** (Internet Control Message Protocol) : comunicación de mensajes entre nodos de la red
    - ***IGMP*** (Internet Group Mangement Protocol) : envío de mensajes a grupos de usuarios

Protocolos (3)

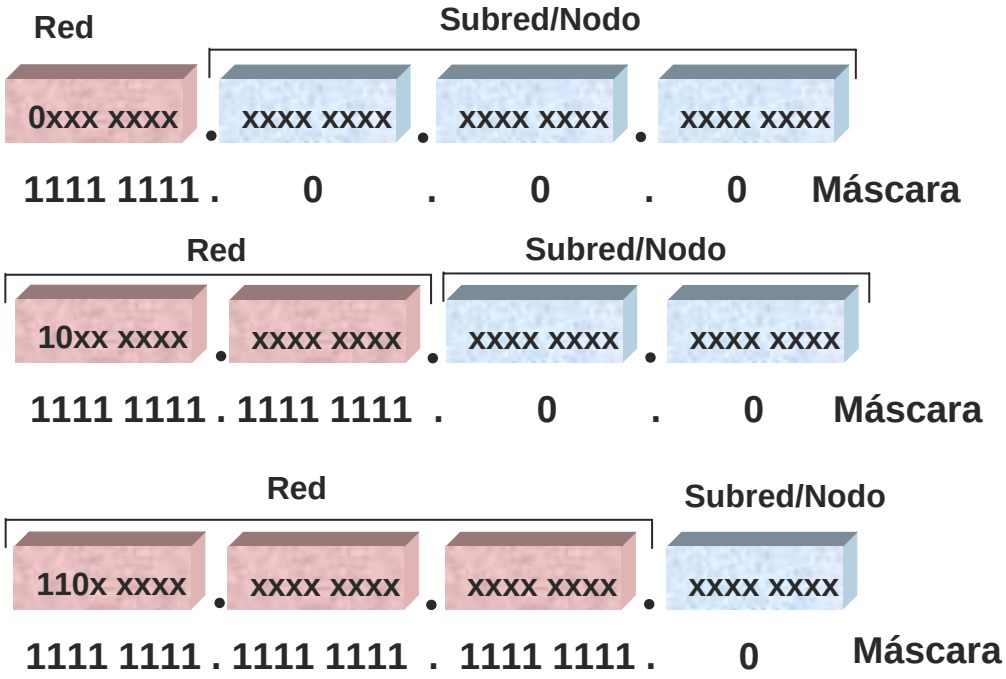
- La capa de transporte:
  - Implementa protocolos *extremo a extremo* (entre nodo origen y destino de la información). Se definen dos protocolos:
  - *TCP* (Transmission Control Protocol):
    - Es un protocolo orientado a la conexión con control de errores
    - Se encarga también del control de flujo
    - Troceado y reensamblado de flujos (garantía de secuenciamiento)
  - *UDP* (User Datagram Protocol) :
    - Es un protocolo sin conexión (*datagrama*)
    - No realiza control de errores
    - No garantiza el secuenciamiento de la información
    - Es muy rápido
    - Útil para peticiones aisladas, o transmisión de audio o vídeo.

Protocolos (4)

- Capa Aplicación: (protocolos de alto nivel)
  - Protocolos basados en ICMP:
    - ***PING*** : solicitud de eco
  - Protocolos basados en TCP:
    - ***TELNET***: terminal remoto
    - ***FTP*** (File Transfer Protocol): transmisión de ficheros
    - ***SMTP*** (Simple Mail Transfer Protocol): correo electrónico
    - ***HTTP*** (HyperText Transfer Protocol: páginas web
    - ***RPC*** (Remote Procedure Call): ejecución de procesos remotos
  - Protocolos basados en UDP:
    - ***SNMP*** (Simple Network Management Protocol): gestión de red
    - ***BOOTP*** : arranque remoto
    - ***DNS*** (Domain Name System)
    - ***RPC*** (Remote Procedure Call): ejecución de procesos remotos
    - ***NFS/RPC*** (Network File System) (gestión de ficheros en red)

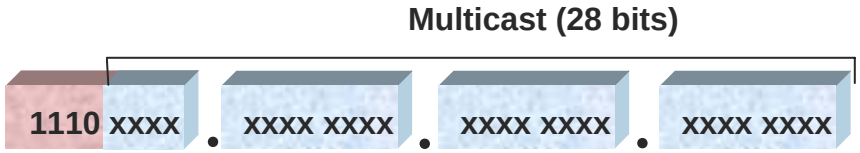
Direccionamiento IPv4 (1)

- Las direcciones IPv4 tienen una longitud de 32 bits (4 bytes)
- Tienen dos partes :
  - Parte de red
  - Parte de subred/nodo
- Clases de direcciones:
  - Clase A
  - Clase B
  - Clase C



Direccionamiento IPv4 (2)

- Clase D



- Clase E

Rangos de direcciones IP

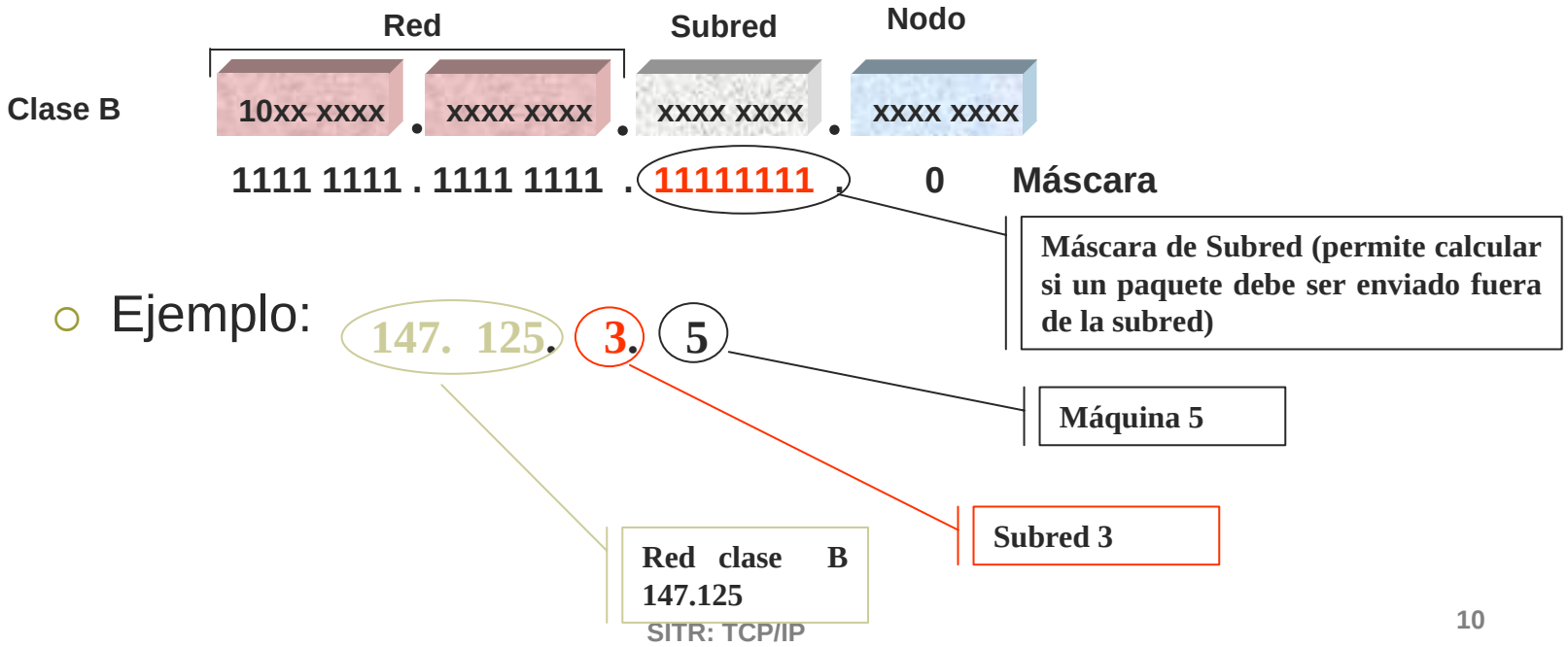
Clase	Nodos por clase	Máscara	Dir. Comienzo	Dir. final
A	16.777.214	255.0.0.0	0.0.0.0	127.255.255.255
B	65534	255.255.0.0	128.0.0.0	191.255.255.255
C	254	255.255.255.0	192.0.0.0	223.255.255.255
D	-	-	224.0.0.0	239.255.255.255
E	-	-	240.0.0.0	255.255.255.255

Rangos de direcciones IP reservadas

Clase	Máscara	Dir. Comienzo	Dir. Final
A	255.0.0.0	10.0.0.0	10.255.255.2555
B	255.255.0.0	172.16.0.0	172.31.255.255
C	255.255.255.0	192.168.0.0	192.168.255.255

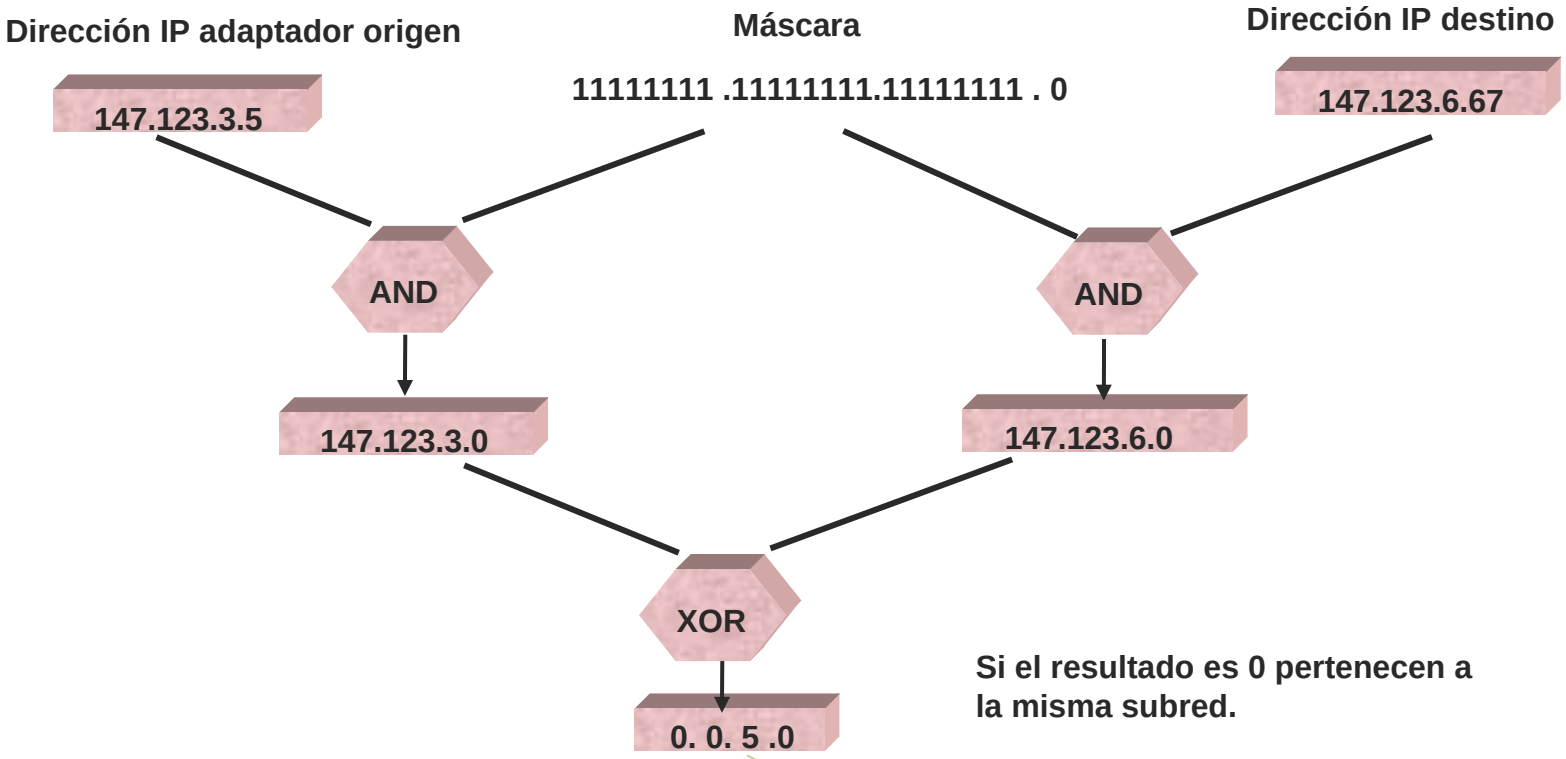
Direccionamiento IPv4 (3)

- Máscara de Subred:
  - Una red se puede dividir en subredes
  - El campo de Subred/Máquina se puede dividir en:
    - Direccionamiento de subredes
    - Direccionamiento de máquinas (dentro de la subred)



Direccionamiento IPv4 (4)

- Utilización de la máscara de red:



Direccionamiento IPv4 (5)

- Direccionamiento de máquina
  - Existen dos valores de los bits del campo de máquina reservados:
    - 0000 0000 (todos los bits a 0) identifica la red/subred
    - 1111 1111 (todos los bits a 1) dirección de broadcast (envío simultáneo de paquetes a todas las máquinas conectadas a la subred)

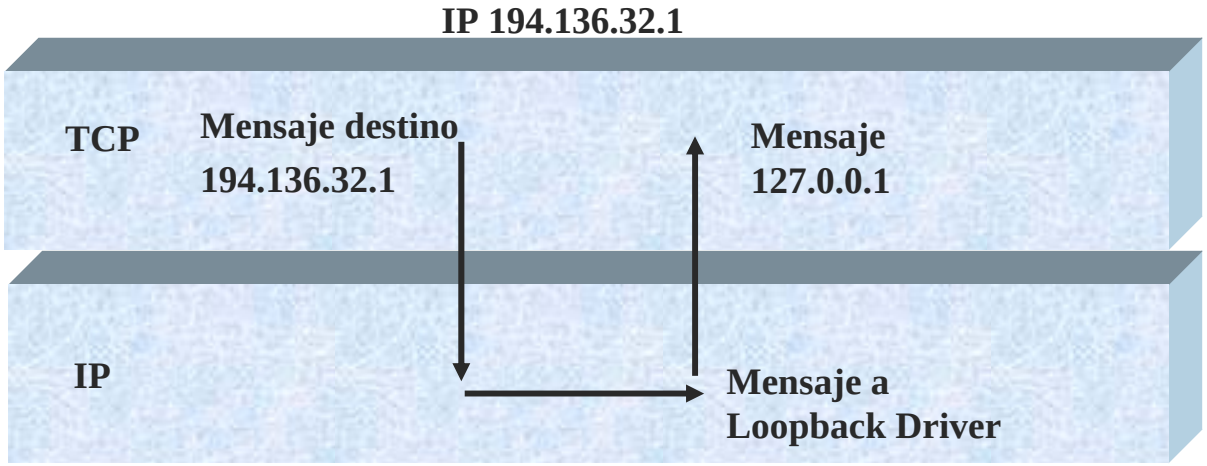
      Por seguridad el uso de paquetes de broadcast está limitado a la subred.

  - Ejemplo:

255.	255.	255.	0	**Máscara de Subred**
147.	125.	3.	0	**Identificador de la subred 147.125.3**
147.	125.	3.	255	**Paquete de broadcast a toda la subred 147.125.3**
Clase B		**Subred**		

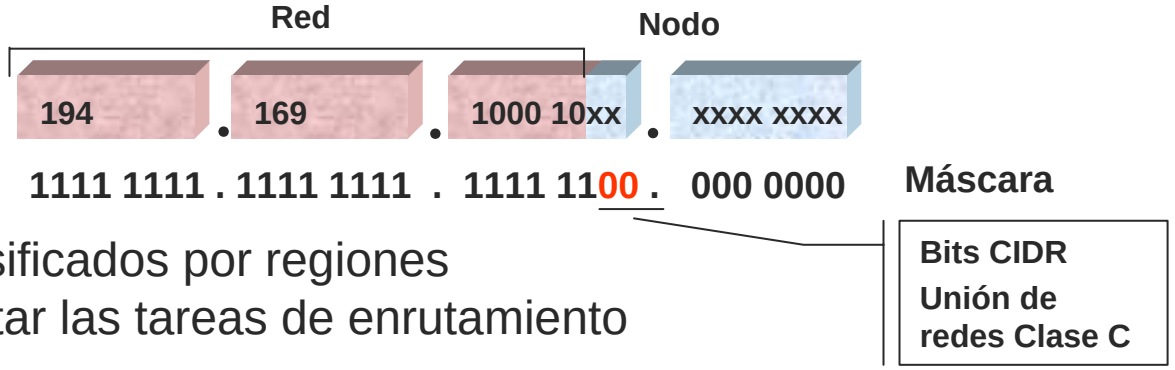
Direccionamiento IPv4 (6)

- Bucle local (Loopback Driver):
  - Es una interface que permite a un cliente y un servidor estar situados en la misma máquina.
  - Se reserva el identificador de red *127*.
  - Generalmente se usa la dirección *127.0.0.1*
  - Se le asigna el nombre '*localhost*'



Direccionamiento IPv4 CIDR

- Implementa un método de enrutamiento interdominios sin clases (RFC 1519)
- Las redes de clase B son demasiado grandes y las de clase C son demasiado pequeñas
- CIDR permite unir varias redes de clase C contiguas, marcando los bits de la máscara de red como máquina (bit a 0)



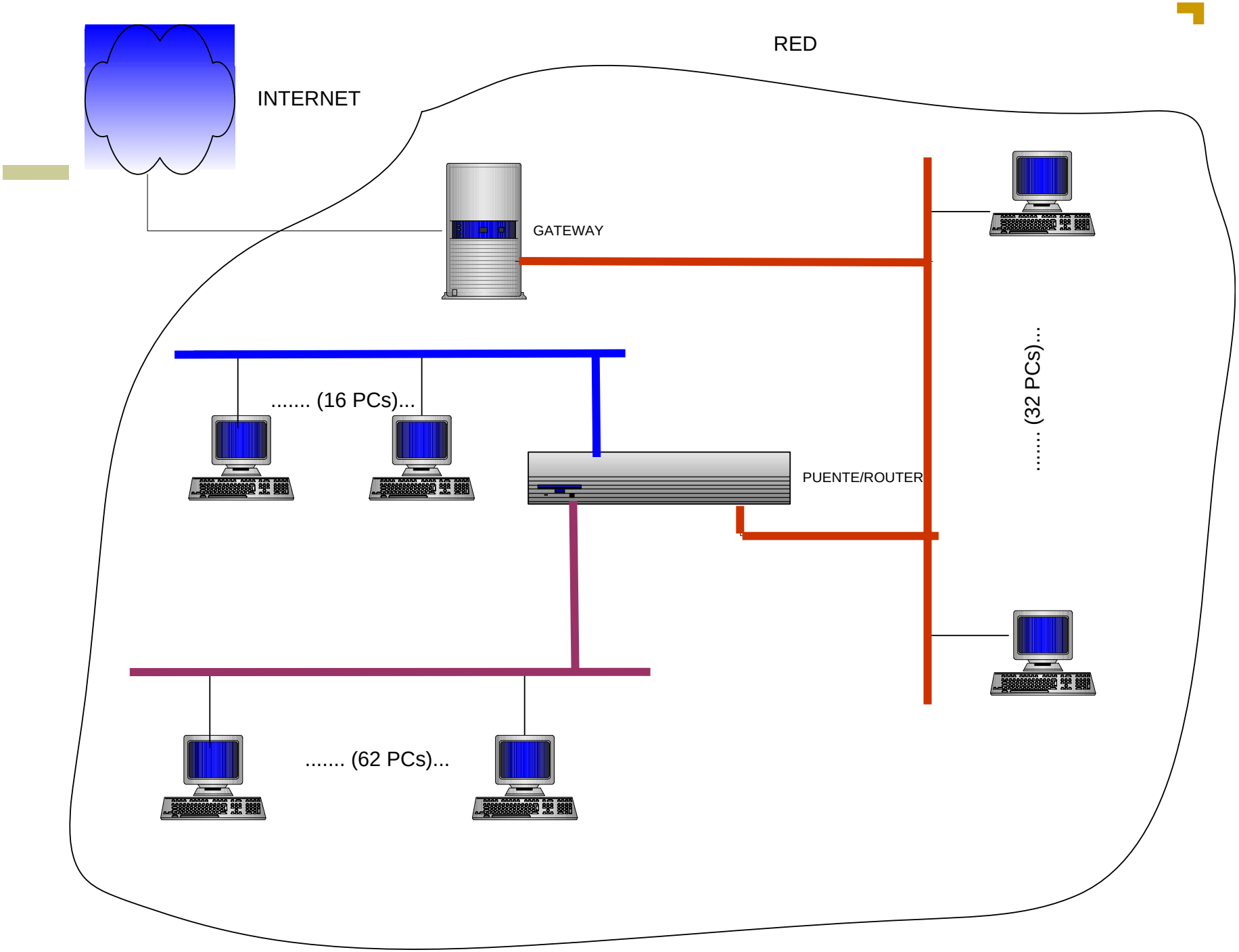
- Están clasificados por regiones para facilitar las tareas de enrutamiento

Región	Dir. Comienzo	Dir. final
Europa	194.0.0.0	195.255.255.255
Norteamérica	198.0.0.0	199.255.255.255
Centro y Sudamérica	200.0.0.0	201.255.255.255
Asia y Pacífico	202.0.0.0	203.255.255.255
Reserva	204.0.0.0	205.255.255.255

Ejemplo Direccionamiento IP

Para la red mostrada en la figura se pide:

- **Asignar una dirección IP de Red de clase C**
- **Especificar la máscara de subred adecuada para cada una de las subredes**
- **Asignar las direcciones IP de cada subred.**
- **Asignar las direcciones IP a cada nodo (Estaciones y Routers).**


RED
INTERNET
GATEWAY
..... (16 PCs)...
PUENTE/ROUTER
..... (32 PCs)...
..... (62 PCs)...

Nivel de Enlace LLC/MAC (1)

- Como comentamos, TCP/IP no define el nivel de enlace
- Estudiaremos un caso muy usual: disponer de un adaptador de Red Ethernet
- Existen dos tipos de tramas :
  - IEEE 802.2/802.3(RFC 1042)
  - Ethernet (RFC 894)

Encapsulación IEEE 802.2/802.3 (RFC 1042)

MAC - Destino	MAC - Fuente	Long.	DSAP	SSAP	Cntl	Org. Code	Tipo	Datos: Datagrama IP	CRC
6	6	2	1	1	1	3	2	38-1492	4

Cabecera

Encapsulación Ethernet (RFC 894)

MAC - Destino	MAC- Fuente	Tipo	Datos: Datagrama IP	CRC
6	6	2	46 - 1500	4

Cabecera

Nivel de Enlace LLC/MAC (2)

- MTU (Maximun Transmission Unit)
  - Depende del tipo de adpatador de red
  - Si un datagrama IP es mayor que el MTU se fragmenta (a nivel IP)
  - Se puede averiguar ejecutando el comando '*netstat -i*'

```
$ netstat -i

Name   Mtu   Net/Dest      Address        Ipkts  Ierrs Opkts  Oerrs Collis Queue
lo0    8232  loopback      localhost      1375284 0     1375284 0     0      0

hme0   1500  lorca         lorca          7246270 70    6586492 1     619142 0
```

Direccionamiento MAC

- Para que un datagrama IP alcance su destino es preciso conocer su dirección física
- Se trata de una combinación de 48 bits agrupados en 6 bytes,
- Los tres primeros bytes identifican al fabricante
- Los tres últimos bytes identifican la tarjeta
- La dirección MAC es única e irrepetible
- Existe una dirección reservada que tiene todos los bits a 1 FF:FF:FF:FF:FF:FF, se utiliza para enviar una trama a todas las máquinas de la red (broadcast)

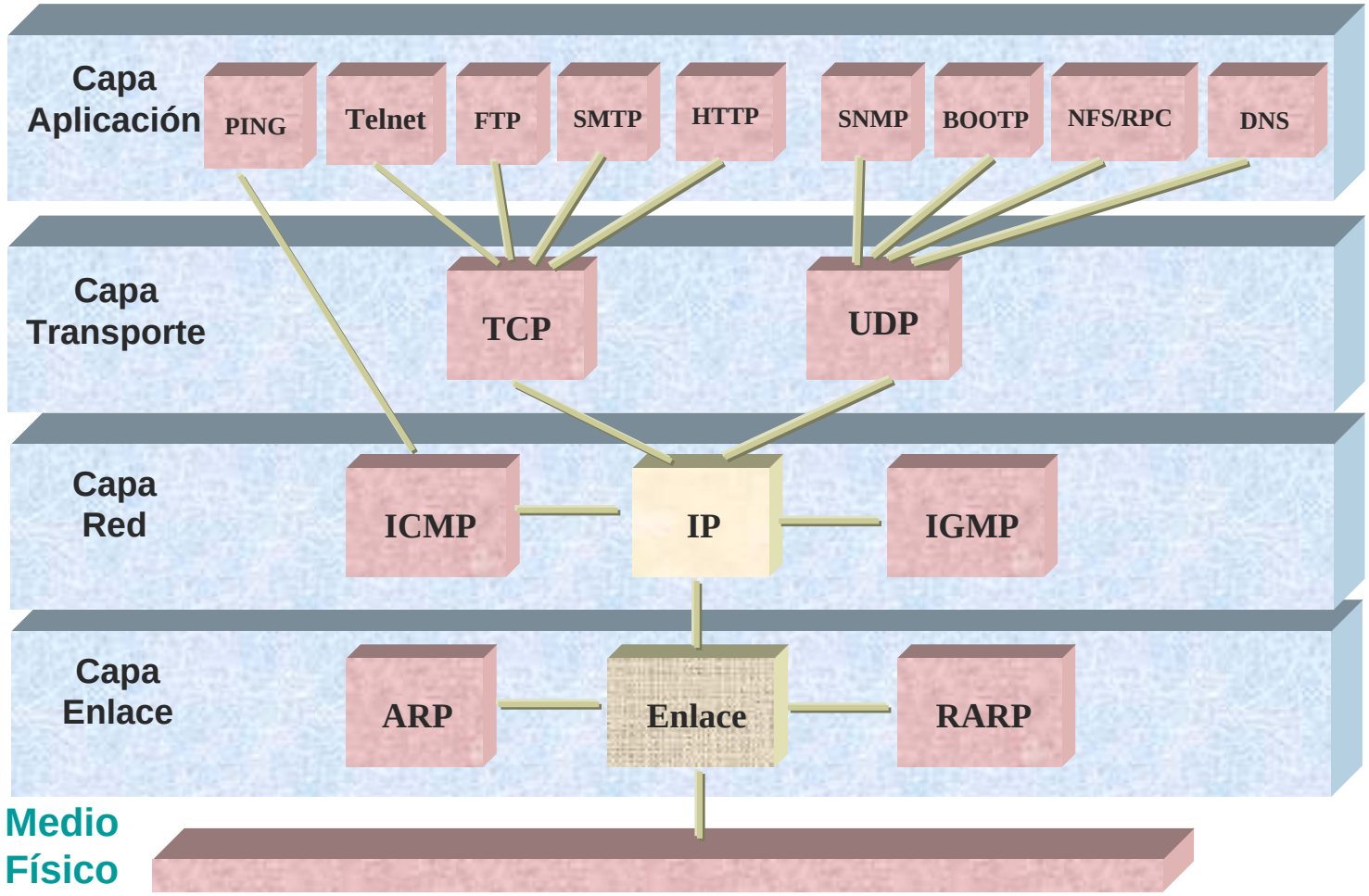
02:60:8C:0D:0A:5C

Fabricante: 3Com

Direccionamiento MAC

- Para conocer la dirección MAC propia basta con interrogar al adaptador de red
  - Windows 95: **winipcfg**
  - Windows NT/2000/XP: **ipconfig**
  - UNIX/Linux: **ifconfig**

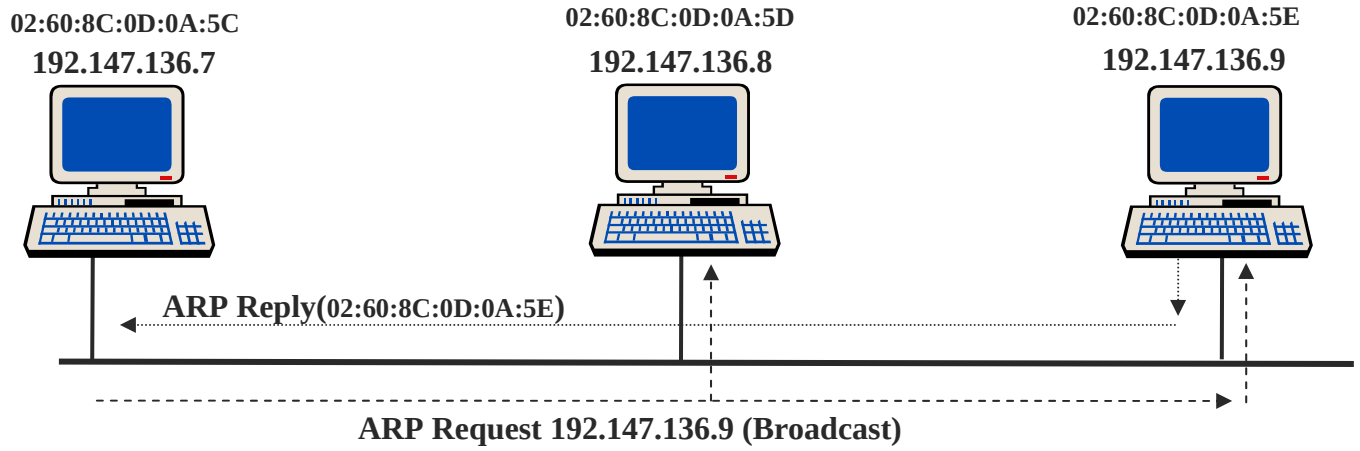
Familia Protocolos TCP/IP

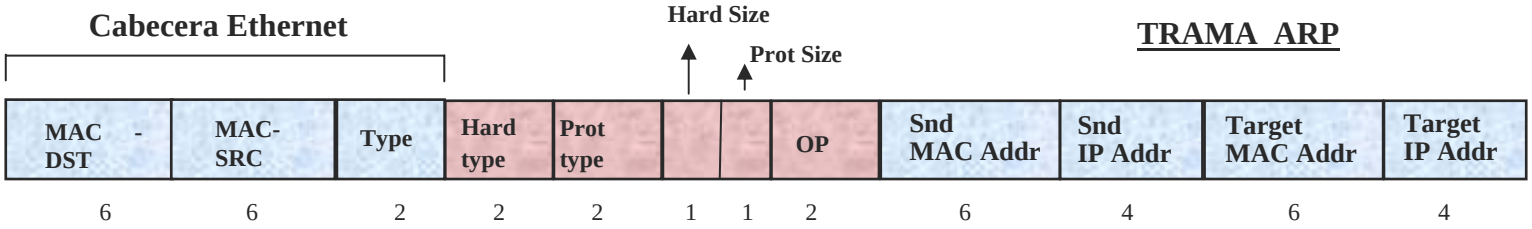


Protocolo ARP (1)

- Para enviar un datagrama necesitamos también la dirección MAC de destino
- Protocolo *ARP* *(Address Resolution Protocol)* obtiene la dirección MAC correspondiente a una dirección IP
  - Envía una trama de '*Petición ARP*' a todas las máquinas de la red (broadcast)
  - El protocolo ARP de la máquina destino verifica que el solicitante pide su dirección IP.
  - Emite una trama de '*Respuesta ARP*' incorporando su dirección MAC e IP
  - Se recibe la respuesta guardando el resultado en una tabla, pudiendo emitir a partir de entonces datagramas IP

Protocolo ARP (2)





Type: 0x0806 (trama ARP)
Hard type: 1 (Ethernet)
Prot. type: 0x0800 (IP)
Hard size: 6 (bytes dir. MAC)
Prot size: 4 (bytes dir. IP)

OP: tipo de operación
- **1: ARP request**
- **2: ARP reply**
- **3: RARP request**
- **4: RARP reply**

Protocolo ARP (3)

- Los datos se almacenan en una tabla dinámica:
  - el comando ***arp -a*** nos muestra la tabla de ARP con la relación entre direcciones MAC e IP
  - El plazo normal de expiración de una entrada es de 20 minutos después de su creación (*Depende del S.O.)*

```
$ arp -a
Net to Media Table
Device   IP Address               Mask      Flags   Phys Addr
------ -------------------- --------------- ----- ---------------
hme0   193.147.136.216      255.255.255.255       00:60:08:64:39:07
hme0   193.147.136.1        255.255.255.255       00:e0:1e:3d:fe:83
hme0   192.168.2.20         255.255.255.255       00:a0:c9:78:75:bb
hme0   lorca                255.255.255.255 SP    08:00:20:a2:55:fe

hme0   BASE-ADDRESS.MCAST.NET 240.0.0.0       SM    01:00:5e:00:00:00
```

- El Protocolo *RARP* (Reverse Address Resolution Protocol) se utiliza para obtener una dirección IP en sistemas sin disco duro.

Ejemplo: Protocolo ARP

Petición/Respuesta ARP

Símbolo del sistema

```
C:\>ping -n 1 192.168.60.10

Haciendo ping a 192.168.60.10 con 32 bytes de datos:

Respuesta desde 192.168.60.10: bytes=32 tiempo<1m TTL=128

Estadísticas de ping para 192.168.60.10:
    Paquetes: enviados = 1, recibidos = 1, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms

C:\>arp -a

Interfaz: 192.168.60.40 --- 0x4
  Dirección IP          Dirección física      Tipo
  192.168.60.1          00-17-31-bb-06-ab     dinámico
  192.168.60.10         00-c0-9f-a7-3d-6d     dinámico
```

(Untitled) - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
1	0.000000	00:19:b9:0e:1f	ff:ff:ff:ff:ff:	ARP	Who has 192.168.60.10? Tell 192.168.60.40
2	0.000159	00:c0:9f:a7:3d	00:19:b9:0e:1f:	ARP	192.168.60.10 is at 00:c0:9f:a7:3d:6d

```
⊞ Frame 1 (42 bytes on wire, 42 bytes captured)
⊞ Ethernet II, Src: 00:19:b9:0e:1f:50 (00:19:b9:0e:1f:50), Dst: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)
⊟ Address Resolution Protocol (request)
    Hardware type: Ethernet (0x0001)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (0x0001)
    Sender MAC address: 00:19:b9:0e:1f:50 (00:19:b9:0e:1f:50)
    Sender IP address: 192.168.60.40 (192.168.60.40)
    Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.60.10 (192.168.60.10)
```

```
0000  ff ff ff ff ff ff 00 19  b9 0e 1f 50 08 06 00 01   ........ ...P....
0010  08 00 06 04 00 01 00 19  b9 0e 1f 50 c0 a8 3c 28   ........ ...P..<(
0020  00 00 00 00 00 00 c0 a8  3c 0a                     ........ <.
```

File: "C:\DOCUME~1\LUIS~1.JIM\CONFIG~1\Temp\etherXXXXa07384" 158 Bytes 00:00:00 | Packets: 2 ... | Profile: Default

(Untitled) - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

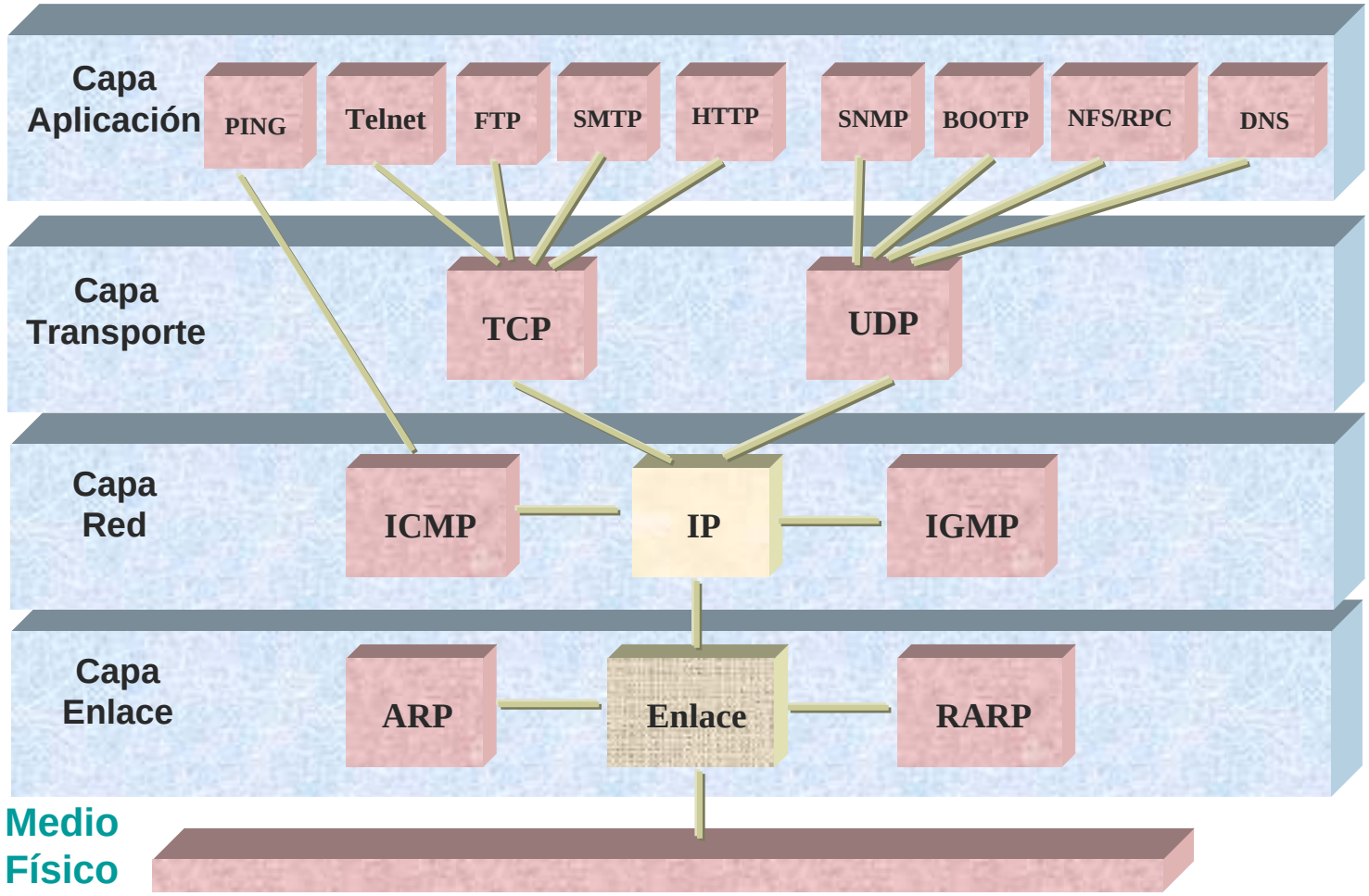
No.	Time	Source	Destination	Protocol	Info
1	0.000000	00:19:b9:0e:1f	ff:ff:ff:ff:ff:	ARP	Who has 192.168.60.10? Tell 192.168.60.40
2	0.000159	00:c0:9f:a7:3d	00:19:b9:0e:1f:	ARP	192.168.60.10 is at 00:c0:9f:a7:3d:6d

```
⊞ Frame 2 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: 00:c0:9f:a7:3d:6d (00:c0:9f:a7:3d:6d), Dst: 00:19:b9:0e:1f:50 (00:19:b9:0e:1f:50)
⊟ Address Resolution Protocol (reply)
    Hardware type: Ethernet (0x0001)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (0x0002)
    Sender MAC address: 00:c0:9f:a7:3d:6d (00:c0:9f:a7:3d:6d)
    Sender IP address: 192.168.60.10 (192.168.60.10)
    Target MAC address: 00:19:b9:0e:1f:50 (00:19:b9:0e:1f:50)
    Target IP address: 192.168.60.40 (192.168.60.40)
```

```
0000  00 19 b9 0e 1f 50 00 c0  9f a7 3d 6d 08 06 00 01   .....P.. ..=m....
0010  08 00 06 04 00 02 00 c0  9f a7 3d 6d c0 a8 3c 0a   ........ ..=m..<.
0020  00 19 b9 0e 1f 50 c0 a8  3c 28 00 00 00 00 00 00   .....P.. <(......
0030  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

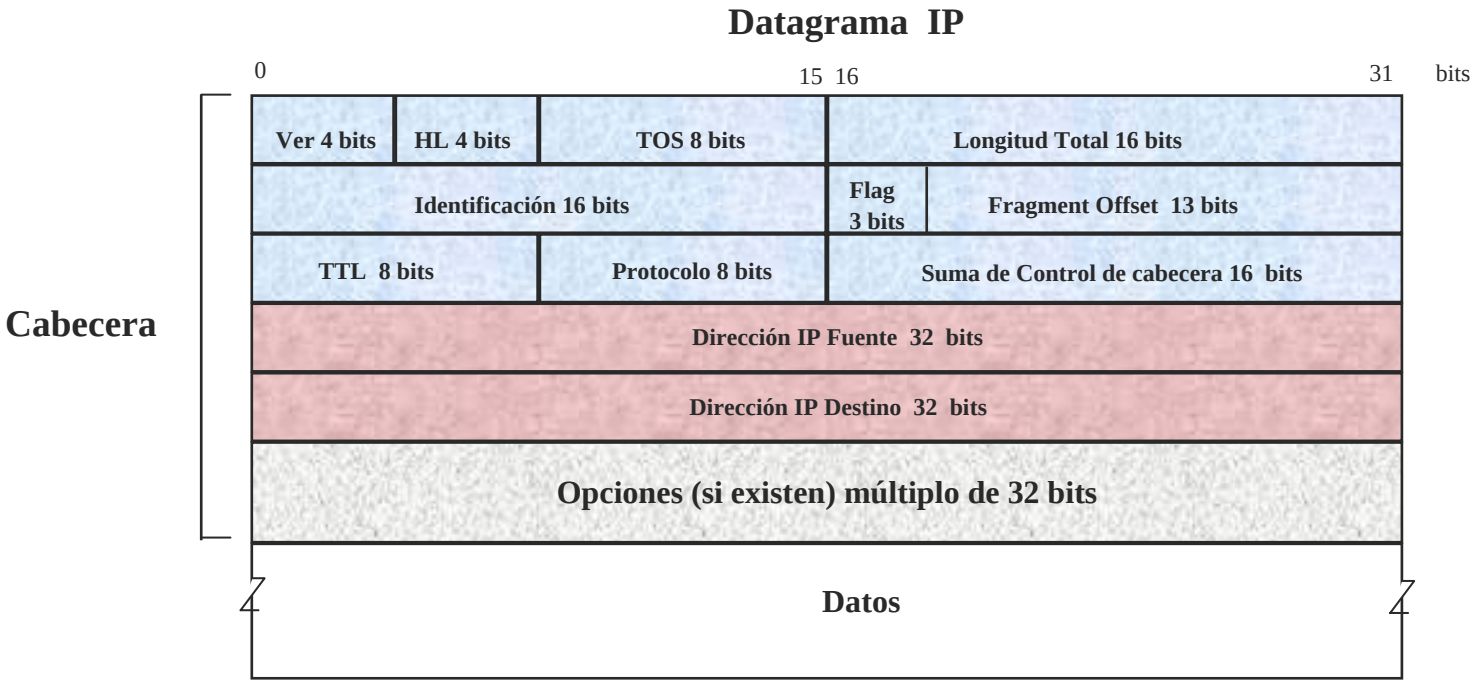
File: "C:\DOCUME~1\LUIS~1.JIM\CONFIG~1\Temp\etherXXXXa07384" 158 Bytes 00:00:00 | Packets: 2 ... | Profile: Default

Familia Protocolos TCP/IP



Protocolo IPv4

- Datagrama IP (RFC 791)
  - El tamaño normal de la cabecera es de 20 bytes, a no ser que presente opciones
  - El tamaño máximo del datagrama se ajusta para que quepa en el MTU del nivel de enlace.



Cabecera IPv4 (1)

- Ver: indica la versión (4)
- HL: longitud de la cabecera en palabras (4 bytes), normalmente vale 5 (20 bytes) pero puede alcanzar las 15 palabras (60 bytes) cuando se utilizan las opciones de marcado de rutas.
- TOS: Tipo de servicio. Los tres primeros bits son ignorados, los 4 bits siguientes representan el TOS y el último vale siempre 0.
  - Cada bit selecciona un tipo de comportamiento (solo uno puede estar activado):
    - 1: Minimiza el Retardo
    - 2: Minimiza el Caudal
    - 3: Maximiza la fiabilidad
    - 4: Minimiza Coste
  - Si todos están a **0** significa servicio normal
- Longitud Total: tamaño en bytes del datagrama IP incluyendo la cabecera (máximo valor 65535)

Cabecera IPv4 (2)

- Identificación:permite marcar de forma única cada datagrama enviado por una máquina (16 bits)
  - Asegura la secuenciación
  - Asociado al mecanismo de fragmentación, su valor se copia en todos los trozos de datagrama que componen el datagrama original)
- Flag:
  - El tercer bit indica (si está a uno) que hay fragmentación '**more fragments'** (se fragmenta si se ha superado el MTU)
  - El último fragmento coloca el bit a 0
  - El segundo bit del capo **Flag** se denomina **'don't fragment'**. Si está activado, fuerza un mensaje de error si el datagrama debe ser fragmentado y lo elimina.
- Fragment Offset:
  - Índice del fragmento a partir del datagrama original (en bloques de 8 bytes)
  - El campo longitud total es reajustado al tamaño de cada fragmento
  - Cada fragmento lleva su propia cabecera IP pero con el mismo identificador de datagrama

Cabecera IPv4 (3)

- TTL: (Time To Live) tiempo de expiración del datagrama
  - Cada nodo por el que pasa decrementa este campo
  - Si alcanza el valor 0 el datagrama es destruido y se genera un mensaje ICMP
  - Permite evitar que un datagrama se quede atrapado en un bucle.
- Protocolo: identifica el protocolo de que utiliza el datagrama IP
  - TCP, UDP, ICMP, IGMP, ...
- Suma de Control: (suma complemento a 1 de palabras de 16 bits)
  - Permite comprobar errores en la cabecera (no tiene en cuenta los datos)
- Direcciones IP fuente y destino
- Opciones:
  - Permiten almacenar un registro de rutas, grabación de tiempos de paso por los routers, restricciones de seguridad, especificar la ruta, ...
  - Está limitado a 40 bytes

Ejemplo: Protocolo IP

Fragmentación IP

Símbolo del sistema

```
C:\>ping -n 1 -l 2000 192.168.60.10

Haciendo ping a 192.168.60.10 con 2000 bytes de datos:

Respuesta desde 192.168.60.10: bytes=2000 tiempo=1ms TTL=128

Estadísticas de ping para 192.168.60.10:
    Paquetes: enviados = 1, recibidos = 1, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 1ms, Máximo = 1ms, Media = 1ms

C:\>
```

(Untitled) - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.60.40	192.168.60.10	IP	Fragmented IP protocol (proto=ICMP 0x01, off=0)
2	0.000015	192.168.60.40	192.168.60.10	ICMP	Echo (ping) request
3	0.000904	192.168.60.10	192.168.60.40	IP	Fragmented IP protocol (proto=ICMP 0x01, off=0)
4	0.000946	192.168.60.10	192.168.60.40	ICMP	Echo (ping) reply

```
⊞ Frame 1 (1514 bytes on wire, 1514 bytes captured)
⊞ Ethernet II, Src: 00:19:b9:0e:1f:50 (00:19:b9:0e:1f:50), Dst: 00:c0:9f:a7:3d:6d (00:c0:9f:a7:3d:6d)
⊟ Internet Protocol, Src: 192.168.60.40 (192.168.60.40), Dst: 192.168.60.10 (192.168.60.10)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 1500
    Identification: 0x0584 (1412)
  ⊞ Flags: 0x02 (More Fragments)
    Fragment offset: 0
    Time to live: 128
    Protocol: ICMP (0x01)
  ⊞ Header checksum: 0x161a [correct]
    Source: 192.168.60.40 (192.168.60.40)
    Destination: 192.168.60.10 (192.168.60.10)
    Reassembled IP in frame: 2
⊟ Data (1480 bytes)
    Data: 0800497804002E006162636465666768696A6B6C6D6E6F70...
```

File: "C:\DOCUME~1\LUIS~1.JIM\CONFIG~1\Temp\etherXXXXa06552" 4240 Bytes 00:00:00 | Packets: 4 ... | Profile: Default

(Untitled) - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

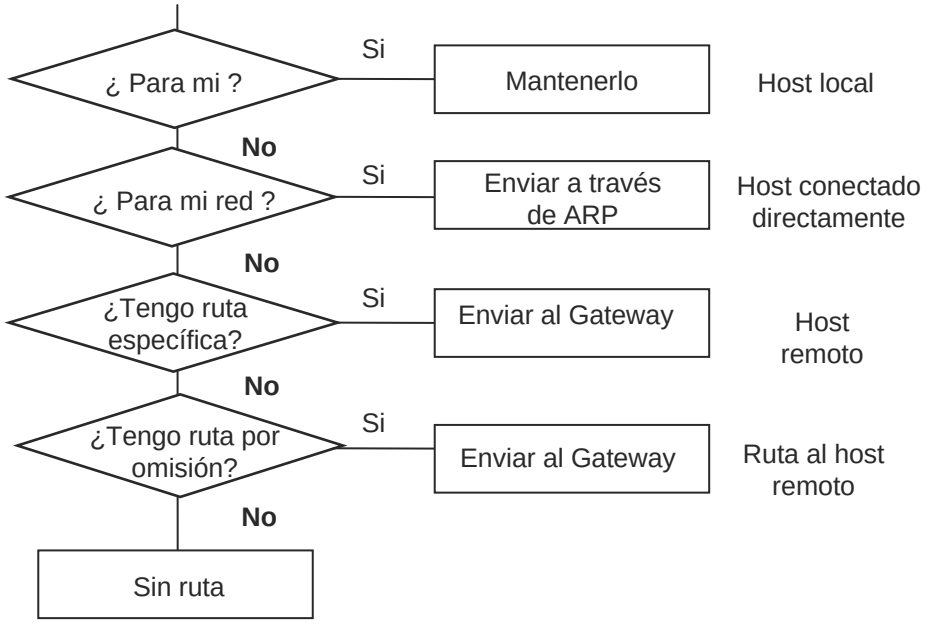
No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.60.40	192.168.60.10	IP	Fragmented IP protocol (proto=ICMP 0x01, off=0)
2	0.000015	192.168.60.40	192.168.60.10	ICMP	Echo (ping) request
3	0.000904	192.168.60.10	192.168.60.40	IP	Fragmented IP protocol (proto=ICMP 0x01, off=0)
4	0.000946	192.168.60.10	192.168.60.40	ICMP	Echo (ping) reply

```
⊞ Frame 2 (562 bytes on wire, 562 bytes captured)
⊞ Ethernet II, Src: 00:19:b9:0e:1f:50 (00:19:b9:0e:1f:50), Dst: 00:c0:9f:a7:3d:6d (00:c0:9f:a7:3d:6d)
⊟ Internet Protocol, Src: 192.168.60.40 (192.168.60.40), Dst: 192.168.60.10 (192.168.60.10)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 548
    Identification: 0x0584 (1412)
  ⊞ Flags: 0x00
    Fragment offset: 1480
    Time to live: 128
    Protocol: ICMP (0x01)
  ⊞ Header checksum: 0x3919 [correct]
    Source: 192.168.60.40 (192.168.60.40)
    Destination: 192.168.60.10 (192.168.60.10)
  ⊞ [IP Fragments (2008 bytes): #1(1480), #2(528)]
⊞ Internet Control Message Protocol
```

File: "C:\DOCUME~1\LUIS~1.JIM\CONFIG~1\Temp\etherXXXXa06552" 4240 Bytes 00:00:00 | Packets: 4 ... | Profile: Default

Enrutamiento IP (1)

- Mecanismo:




```
$ netstat -rn

Routing Table:
  Destination           Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
193.147.136.0        193.147.136.16        U        3   2619  hme0
224.0.0.0            193.147.136.16        U        3      0  hme0
default              193.147.136.1         UG       0  12386

127.0.0.1            127.0.0.1             UH       01300499  lo0
```

Enrutamiento IP (2)

- Tabla de Rutas:
  - Dirección IP destino: máquina o una red
  - Dirección IP del siguiente router: al que se debe enviar el datagrama
  - Flags:
    - **G** la dirección de IP de destino es una red (ruta indirecta)
    - **H** la dirección de IP de destino es una máquina (ruta directa)
    - **U** la ruta está en servicio
    - **D** la ruta ha sido creada por redirección
    - **M** ruta modificada por redirección

  - Debe existir una ruta por defecto (default gateway)

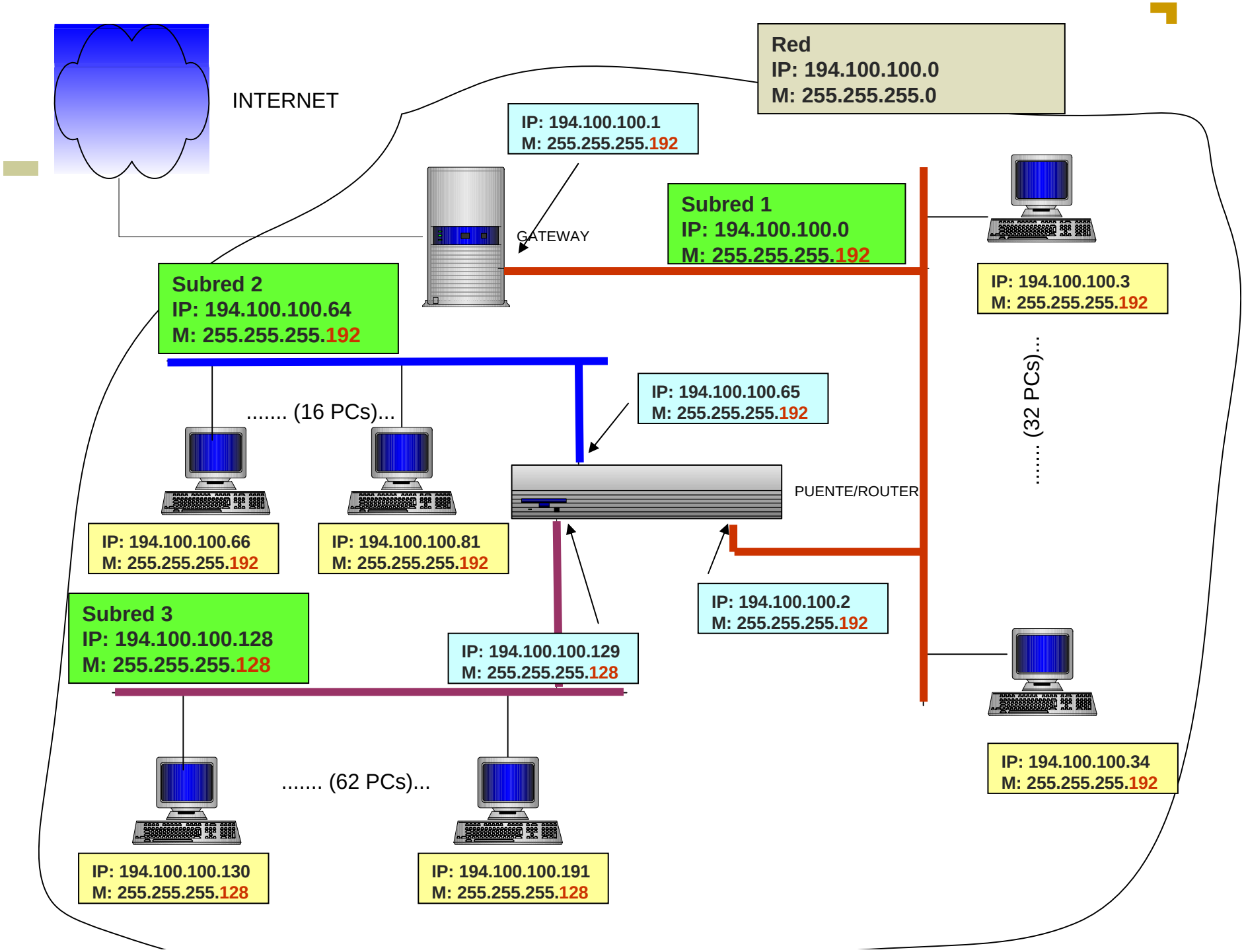
Enrutamiento IP (3)

- Algoritmos de enrutamiento:
  - Se pueden crear rutas manualmente (estático)
    - Se utiliza el comando '*route*'
  - La tabla de enrutamiento se actualiza también dinámicamente (demonios '*routed*' y '*gated*')
  - Existen diferentes algoritmos según el tipo de ruta (se basan todos ellos en el intercambio de las tablas entre routers):
    - **RIP** (Routing Information Protocol):
      - Usa una métrica basada en saltos (rutas corta, 16 saltos)
      - Utiliza datagramas UDP para pasar la información entre routers
    - **HELLO**: usa una métrica basada en retardos (rutas largas)
    - **BGP** (Border Gateway Protocol)
    - **EGP** (Exterior Border Protocol)

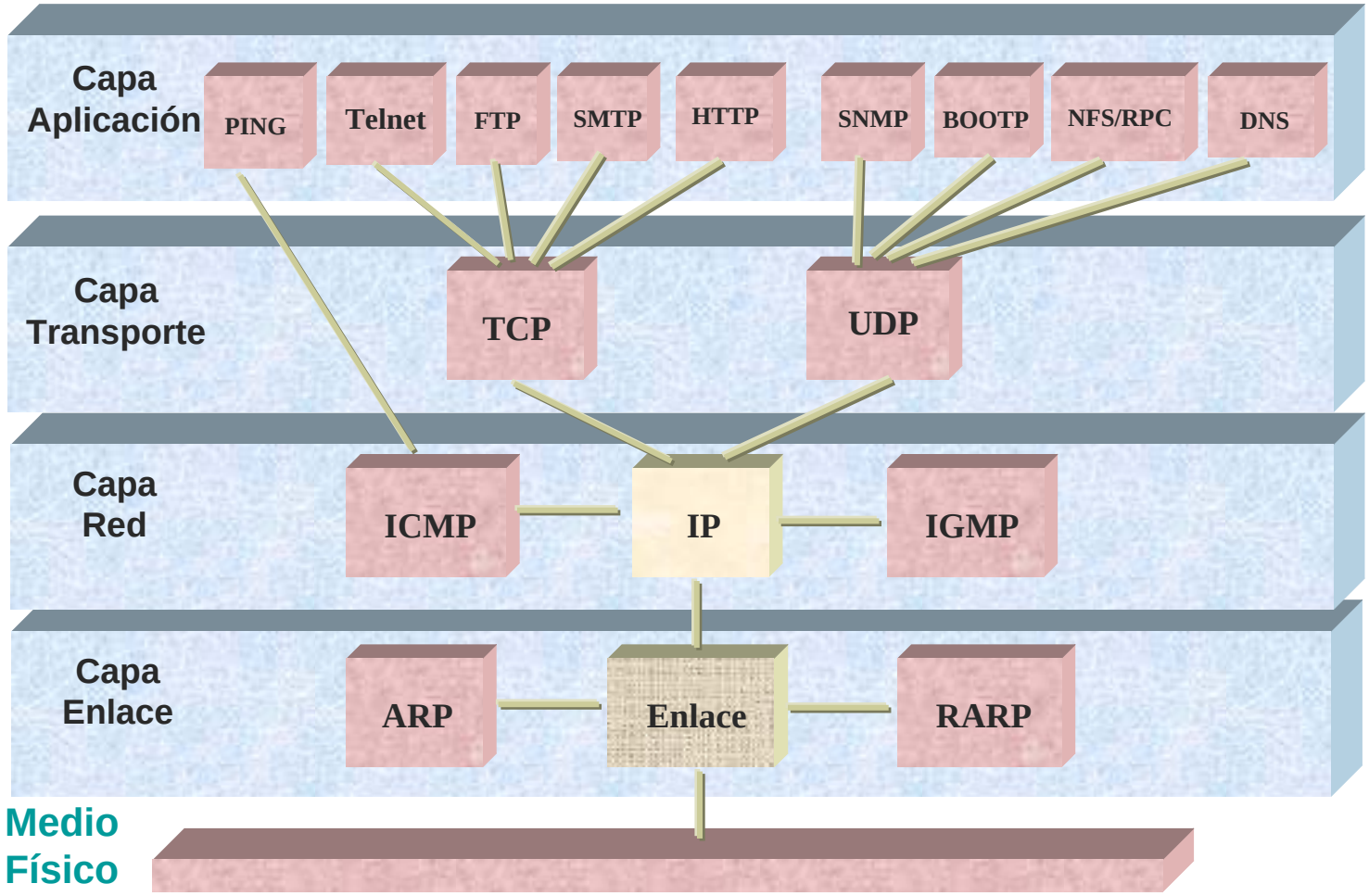
Ejemplo Enrutamiento IP

Para la red mostrada en la figura se pide:

- **Asignar la dirección IP de la pasarela (router) por defecto de cada nodo .**
- **Tabla de enrutamiento del router interno de la red**

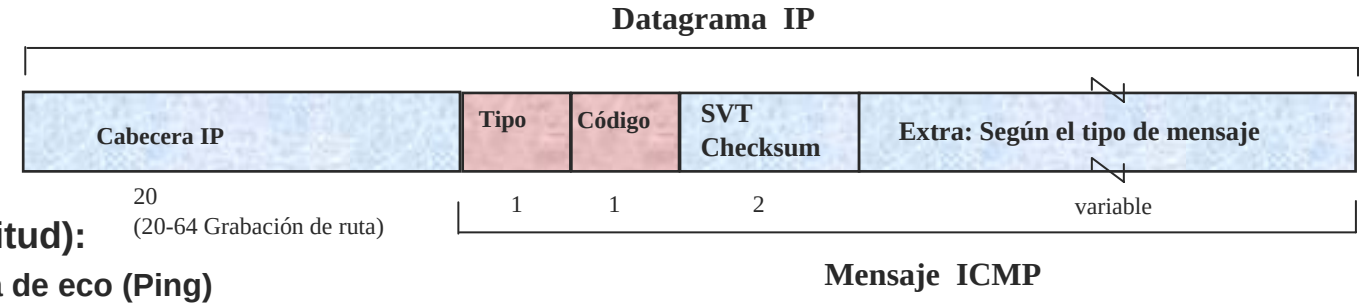

INTERNET
Red
IP: 194.100.100.0
M: 255.255.255.0
IP: 194.100.100.1
M: 255.255.255.192
GATEWAY
Subred 1
IP: 194.100.100.0
M: 255.255.255.192
IP: 194.100.100.3
M: 255.255.255.192
..... (32 PCs)...
Subred 2
IP: 194.100.100.64
M: 255.255.255.192
IP: 194.100.100.65
M: 255.255.255.192
..... (16 PCs)...
PUENTE/ROUTER
IP: 194.100.100.66
M: 255.255.255.192
IP: 194.100.100.81
M: 255.255.255.192
IP: 194.100.100.2
M: 255.255.255.192
Subred 3
IP: 194.100.100.128
M: 255.255.255.128
IP: 194.100.100.129
M: 255.255.255.128
IP: 194.100.100.34
M: 255.255.255.192
..... (62 PCs)...
IP: 194.100.100.130
M: 255.255.255.128
IP: 194.100.100.191
M: 255.255.255.128

Familia Protocolos TCP/IP



Protocolo ICMP (1)

- Tiene como misión comunicar los mensajes de error, solicitudes y respuestas entre nodos de la red (RFC 792)
- Los mensajes son enviados dentro de un datagrama IP



Tipo (Solicitud):

0: Respuesta de eco (Ping)
8: Petición de eco (Ping)
9: Aviso de Router
10: Solicitud de Router
13: Petición 'timestamp'
14: Respuesta 'timestamp'
15: Pet. de información (obsoleto)
16: Resp. de información (obsoleto)
17: Petición máscara dirección
18: Respuesta máscara dirección

Tipo (Error):

3: Destino Inaccesible
4: Cadencia demasiado elevada
5: Redirección
11: Tiempo sobrepasado (TTL)
12: Problema de parámetros o forma

Código:

0: Solicitud
X: Especifica diferentes tipos de errores

type	*code*	Description	Query	Error
0	0	echo reply (Ping reply, Chapter 7)	•	
3		destination unreachable:		
	0	network unreachable (Section 9.3)		•
	1	host unreachable (Section 9.3)		•
	2	protocol unreachable		•
	3	port unreachable (Section 6.5)		•
	4	fragmentation needed but don't-fragment bit set (Section 11.6)		•
	5	source route failed (Section 8.5)		•
	6	destination network unknown		•
	7	destination host unknown		•
	8	source host isolated (obsolete)		•
	9	destination network administratively prohibited		•
	10	destination host administratively prohibited		•
	11	network unreachable for TOS (Section 9.3)		•
	12	host unreachable for TOS (Section 9.3)		•
	13	communication administratively prohibited by filtering		•
	14	host precedence violation		•
	15	precedence cutoff in effect		•
4	0	source quench (elementary flow control, Section 11.11)		•
5		redirect (Section 9.5):		
	0	redirect for network		•
	1	redirect for host		•
	2	redirect for type-of-service and network		•
	3	redirect for type-of-service and host		•
8	0	echo request (Ping request, Chapter 7)	•	
9	0	router advertisement (Section 9.6)	•	
10	0	router solicitation (Section 9.6)	•	
11		time exceeded:		
	0	time-to-live equals 0 during transit (Traceroute, Chapter 8)		•
	1	time-to-live equals 0 during reassembly (Section 11.5)		•
12		parameter problem:		
	0	IP header bad (catchall error)		•
	1	required option missing		•
13	0	timestamp request (Section 6.4)	•	
14	0	timestamp reply (Section 6.4)	•	
15	0	information request (obsolete)	•	
16	0	information reply (obsolete)	•	
17	0	address mask request (Section 6.3)	•	
18	0	address mask reply (Section 6.3)	•	

Protocolo ICMP (2)

- Existen dos tipos de mensajes:
  - Solicitud ICMP:
    - Se solicita o se informa de un acontecimiento concreto que no es causa de ningún error
    - Ejemplo: el envío de un eco, o la presencia de un router
  - Notificación de Error ICMP:
    - Por ejemplo: una máquina inaccesible, una fragmentación requerida, etc...
    - Un mensaje de error no puede ser producido por otro mensaje de error
    - Un mensaje de error contiene siempre la cabecera IP (20 bytes) y los 8 primeros bytes del datagrama que lo provocó (permite asociar el mensaje a un protocolo TCP/UDP y un puerto (usuario))

Protocolo ICMP (3)

- Situaciones que no producen un mensaje de error(evita las tormentas de ‘broadcast’):
  - Un mensaje de error ICMP
  - Un datagrama IP o una trama enlace destinado a una dirección ‘broadcast’
  - Un fragmento recibido fuera de secuencia
  - Un datagrama cuya dirección fuente no ha sido emitida por una sola máquina

Ejemplo: Protocolo ICMP

Petición de Eco (ping)

Símbolo del sistema

```
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>ping -n 2 192.168.60.10

Haciendo ping a 192.168.60.10 con 32 bytes de datos:

Respuesta desde 192.168.60.10: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.60.10: bytes=32 tiempo<1m TTL=128

Estadísticas de ping para 192.168.60.10:
    Paquetes: enviados = 2, recibidos = 2, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms

C:\>_
```

(Untitled) - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.60.40	192.168.60.10	ICMP	Echo (ping) request
2	0.000236	192.168.60.10	192.168.60.40	ICMP	Echo (ping) reply
3	0.999974	192.168.60.40	192.168.60.10	ICMP	Echo (ping) request
4	1.000187	192.168.60.10	192.168.60.40	ICMP	Echo (ping) reply

```
⊞ Frame 1 (74 bytes on wire, 74 bytes captured)
⊞ Ethernet II, Src: 00:19:b9:0e:1f:50 (00:19:b9:0e:1f:50), Dst: 00:c0:9f:a7:3d:6d (00:c0:9f:a7:3d:6d)
⊞ Internet Protocol, Src: 192.168.60.40 (192.168.60.40), Dst: 192.168.60.10 (192.168.60.10)
⊟ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0 ()
    Checksum: 0x115c [correct]
    Identifier: 0x0400
    Sequence number: 14336 (0x3800)
  ⊟ Data (32 bytes)
      Data: 6162636465666768696A6B6C6D6E6F707172737475767761...

0000  00 c0 9f a7 3d 6d 00 19  b9 0e 1f 50 08 00 45 00   ....=m.. ...P..E.
0010  00 3c eb 75 00 00 80 01  55 c8 c0 a8 3c 28 c0 a8   .<.u.... U...<(..
0020  3c 0a 08 00 11 5c 04 00  38 00 61 62 63 64 65 66   <....\.. 8.abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                     wabcdefg hi
```

File: "C:\DOCUME~1\LUIS~1.JIM\CONFIG~1\Temp\etherXXXXa06232" 384 Bytes 00:00:01 | Packets: 4 ... | Profile: Default

(Untitled) - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

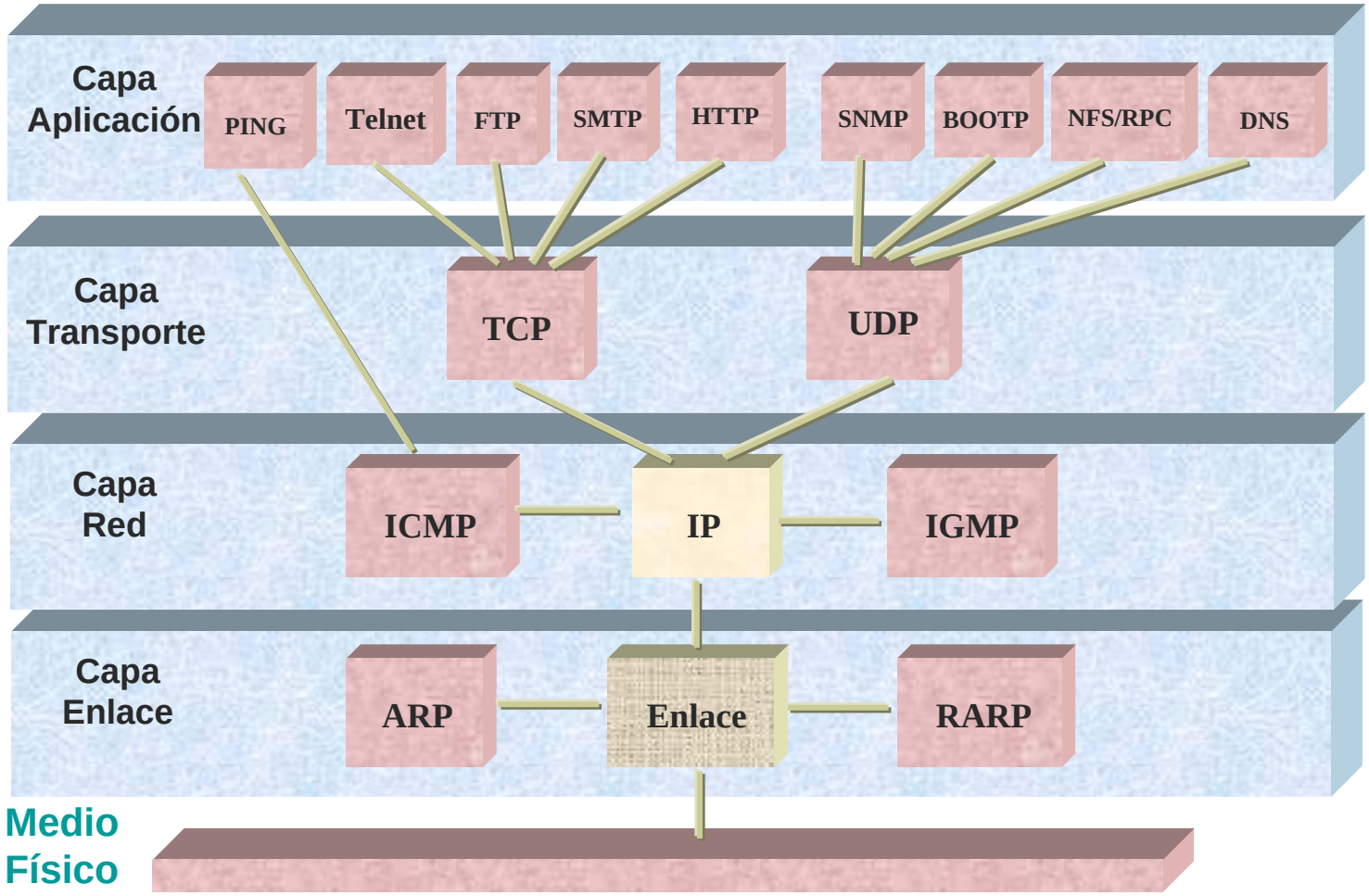
No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.60.40	192.168.60.10	ICMP	Echo (ping) request
2	0.000236	192.168.60.10	192.168.60.40	ICMP	Echo (ping) reply
3	0.999974	192.168.60.40	192.168.60.10	ICMP	Echo (ping) request
4	1.000187	192.168.60.10	192.168.60.40	ICMP	Echo (ping) reply

```
⊞ Frame 2 (74 bytes on wire, 74 bytes captured)
⊞ Ethernet II, Src: 00:c0:9f:a7:3d:6d (00:c0:9f:a7:3d:6d), Dst: 00:19:b9:0e:1f:50 (00:19:b9:0e:1f:50)
⊞ Internet Protocol, Src: 192.168.60.10 (192.168.60.10), Dst: 192.168.60.40 (192.168.60.40)
⊟ Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0 ()
    Checksum: 0x195c [correct]
    Identifier: 0x0400
    Sequence number: 14336 (0x3800)
  ⊟ Data (32 bytes)
      Data: 6162636465666768696A6B6C6D6E6F707172737475767761...
```

```
0000  00 19 b9 0e 1f 50 00 c0  9f a7 3d 6d 08 00 45 00   .....P.. ..=m..E.
0010  00 3c cb ab 00 00 80 01  75 92 c0 a8 3c 0a c0 a8   .<...... u...<...
0020  3c 28 00 00 19 5c 04 00  38 00 61 62 63 64 65 66   <(...\.. 8.abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                     wabcdefg hi
```

File: "C:\DOCUME~1\LUIS~1.JIM\CONFIG~1\Temp\etherXXXXa06232" 384 Bytes 00:00:01 | Packets: 4 ... | Profile: Default

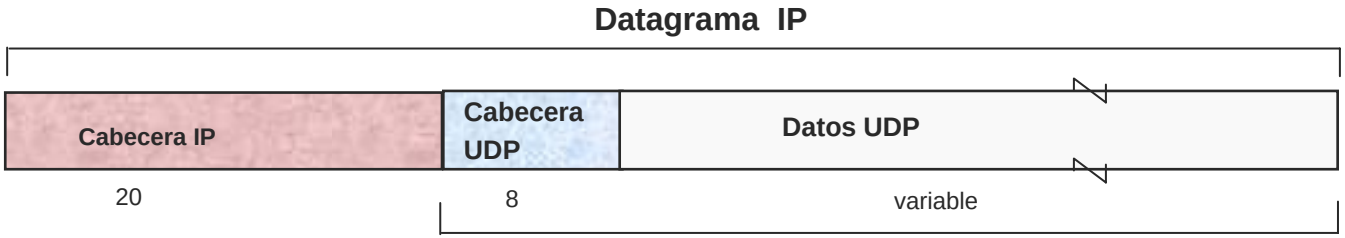
Familia Protocolos TCP/IP

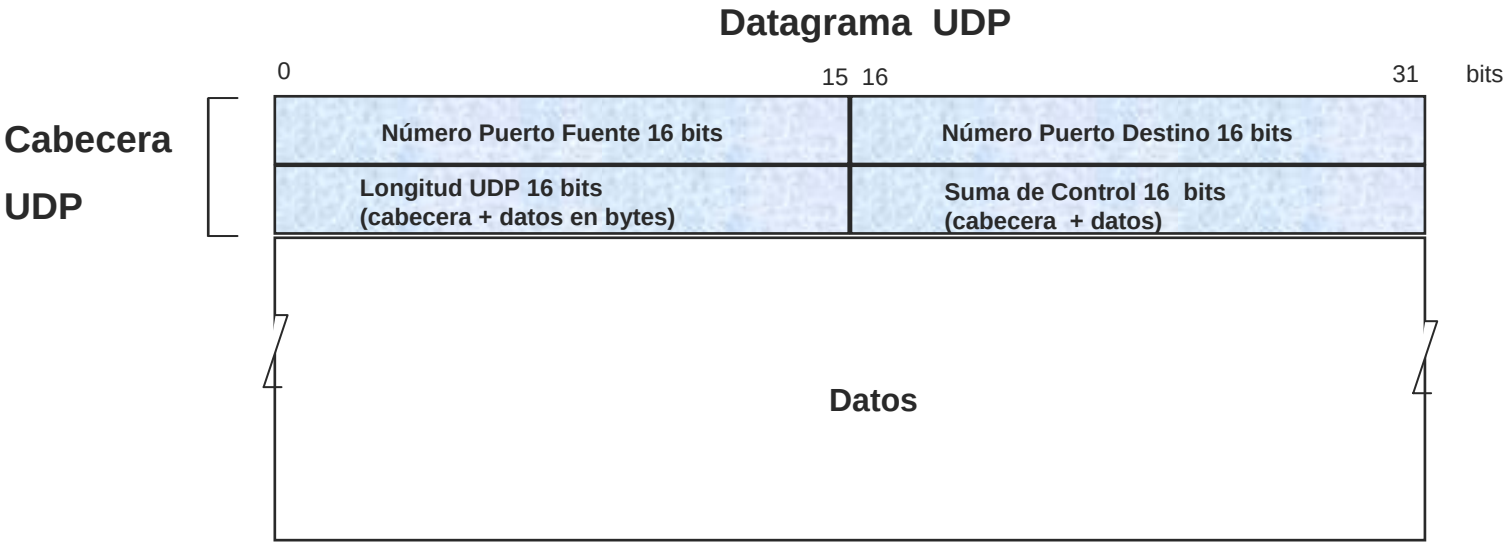


Capa Transporte: UDP

- UDP es un protocolo no orientado a la conexión
  - Cada operación de salida efectuada por un proceso genera un único datagrama UDP
  - Cada datagrama es autocontenido, no se mezclan.
  - Cada datagrama UDP genera un datagrama IP aunque este puede ser fragmentado dependiendo del MTU
  - UDP no ofrece garantía de secuencia (los datagramas pueden llegar desordenados)
  - UDP no ofrece garantía de fiabilidad(los datagramas pueden ser incorrectos o no llegar)
  - UDP no ofrece control de flujo
  - Es misión del protocolo de nivel aplicación el control de flujo y errores

Datagrama UDP





Ejemplo: Protocolo UDP

Solicitud DNS:

Conversión de nombre simbólico a dirección IP

(Untitled) - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.60.40	192.168.60.1	DNS	Standard query A lorca.umh.es
2	0.001729	192.168.60.1	192.168.60.40	DNS	Standard query response A 193.147.145.36

```
⊞ Frame 1 (72 bytes on wire, 72 bytes captured)
⊞ Ethernet II, Src: 00:19:b9:0e:1f:50 (00:19:b9:0e:1f:50), Dst: 00:17:31:bb:06:ab (00:17:31:bb:06:ab)
⊞ Internet Protocol, Src: 192.168.60.40 (192.168.60.40), Dst: 192.168.60.1 (192.168.60.1)
⊟ User Datagram Protocol, Src Port: 16303 (16303), Dst Port: 53 (53)
    Source port: 16303 (16303)
    Destination port: 53 (53)
    Length: 38
  ⊞ Checksum: 0x2a1f [correct]
⊟ Domain Name System (query)
    [Response In: 2]
    Transaction ID: 0xdc9e
  ⊞ Flags: 0x0100 (Standard query)
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ⊟ Queries
    ⊞ lorca.umh.es: type A, class IN
```

File: "C:\DOCUME~1\LUIS~1.JIM\CONFIG~1\Temp\etherXXXXa05160" 216 Bytes 00:00:00 | Packets: 2 ... | Profile: Default

(Untitled) - Wireshark

File Edit View Go Capture Analyze Statistics Help

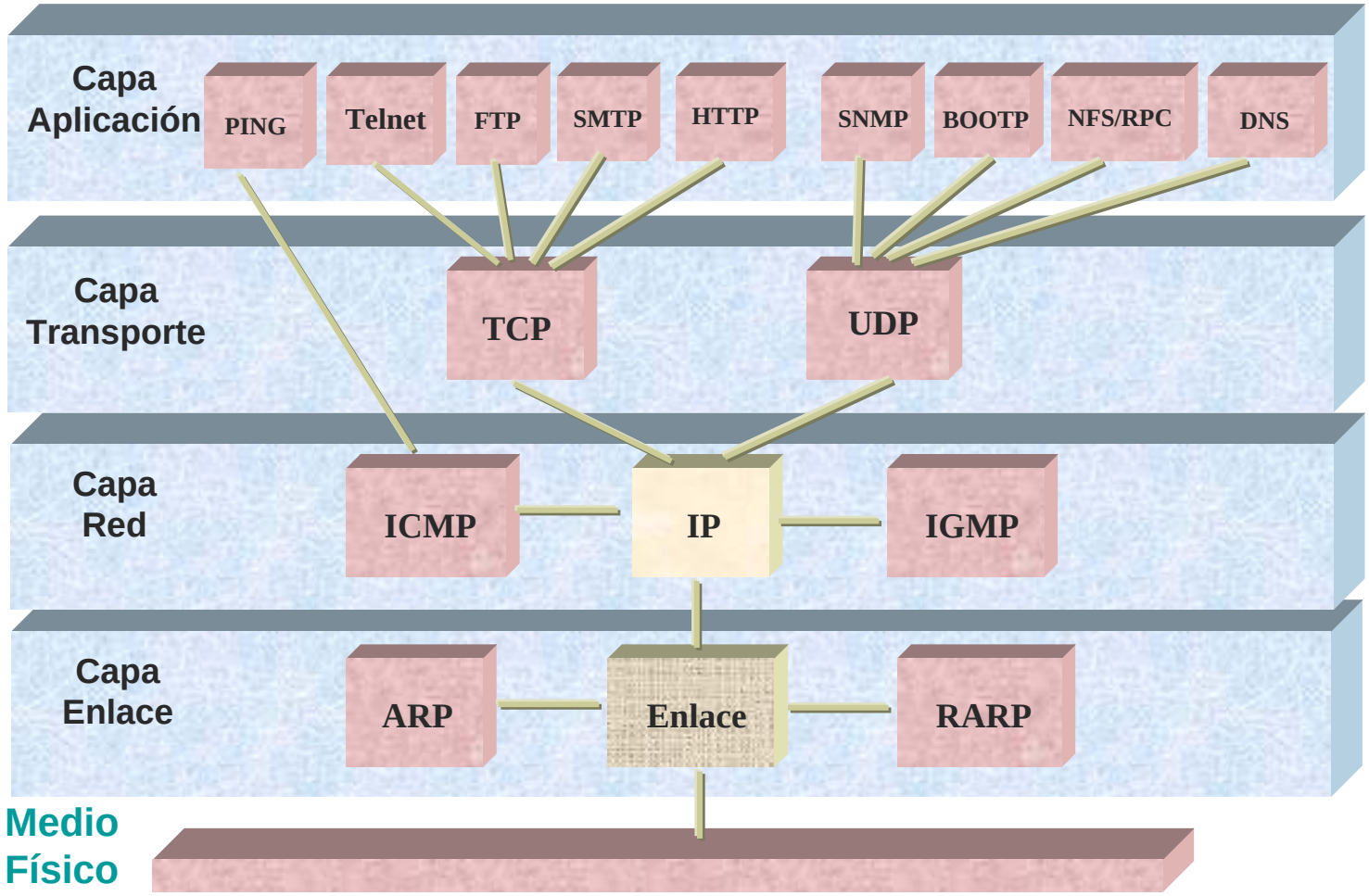
Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.60.40	192.168.60.1	DNS	Standard query A lorca.umh.es
2	0.001729	192.168.60.1	192.168.60.40	DNS	Standard query response A 193.147.145.36

```
⊞ Frame 2 (88 bytes on wire, 88 bytes captured)
⊞ Ethernet II, Src: 00:17:31:bb:06:ab (00:17:31:bb:06:ab), Dst: 00:19:b9:0e:1f:50 (00:19:b9:0e:1f:50)
⊞ Internet Protocol, Src: 192.168.60.1 (192.168.60.1), Dst: 192.168.60.40 (192.168.60.40)
⊟ User Datagram Protocol, Src Port: 53 (53), Dst Port: 16303 (16303)
    Source port: 53 (53)
    Destination port: 16303 (16303)
    Length: 54
  ⊞ Checksum: 0xefaf [correct]
⊟ Domain Name System (response)
    [Request In: 1]
    [Time: 0.001729000 seconds]
    Transaction ID: 0xdc9e
  ⊞ Flags: 0x8580 (Standard query response, No error)
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  ⊟ Queries
    ⊞ lorca.umh.es: type A, class IN
  ⊟ Answers
    ⊞ lorca.umh.es: type A, class IN, addr 193.147.145.36
```

File: "C:\DOCUME~1\LUIS~1.JIM\CONFIG~1\Temp\etherXXXXa05160" 216 Bytes 00:00:00 | Packets: 2 ... | Profile: Default

Familia Protocolos TCP/IP



Capa Transporte: TCP (1)

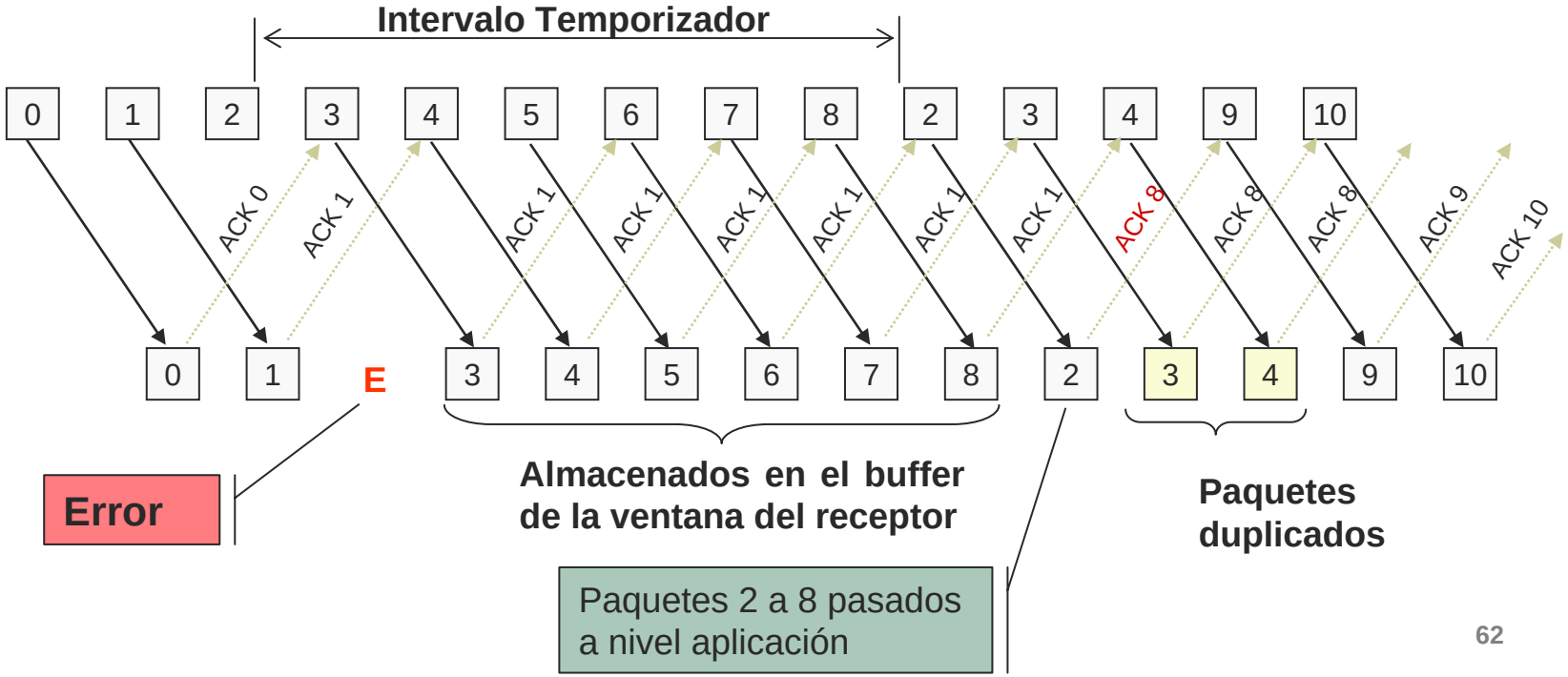
- Servicio de flujo (stream) de bytes, orientado a la conexión y fiable
  - Precisa una etapa previa de conexión y una posterior de desconexión a la transmisión de datos
  - El tamaño de los mensajes es fijado por TCP (si el mensaje es más grande se fragmenta)
  - La unidad de información se denomina 'segmento'
  - No hay fronteras entre mensajes (flujo)
  - Control de Flujo
  - Secuenciamiento de Datos
  - Control y Recuperación de Errores

Capa Transporte: TCP (2)

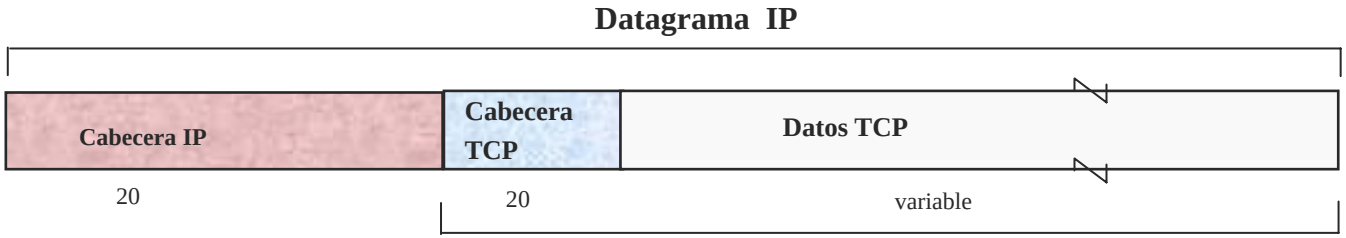
- Es un protocolo de ventana deslizante no selectivo
  - Cuando TCP emite un segmento, mantiene un temporizador esperando su asentimiento por el otro extremo. Si expira el temporizador se **reenvía** el segmento
  - Si TCP recibe datos del otro extremo de la conexión, emite un asentimiento
  - TCP mantiene una suma de control en su cabecera y datos. Si aparece un segmento inválido, se rechaza y **no se emite asentimiento**
  - Los segmentos TCP son enviados como datagramas IP, por tanto pueden llegar desordenados. Los segmentos TCP llevan un número de secuencia por lo que TCP los **reordena** si es necesario.

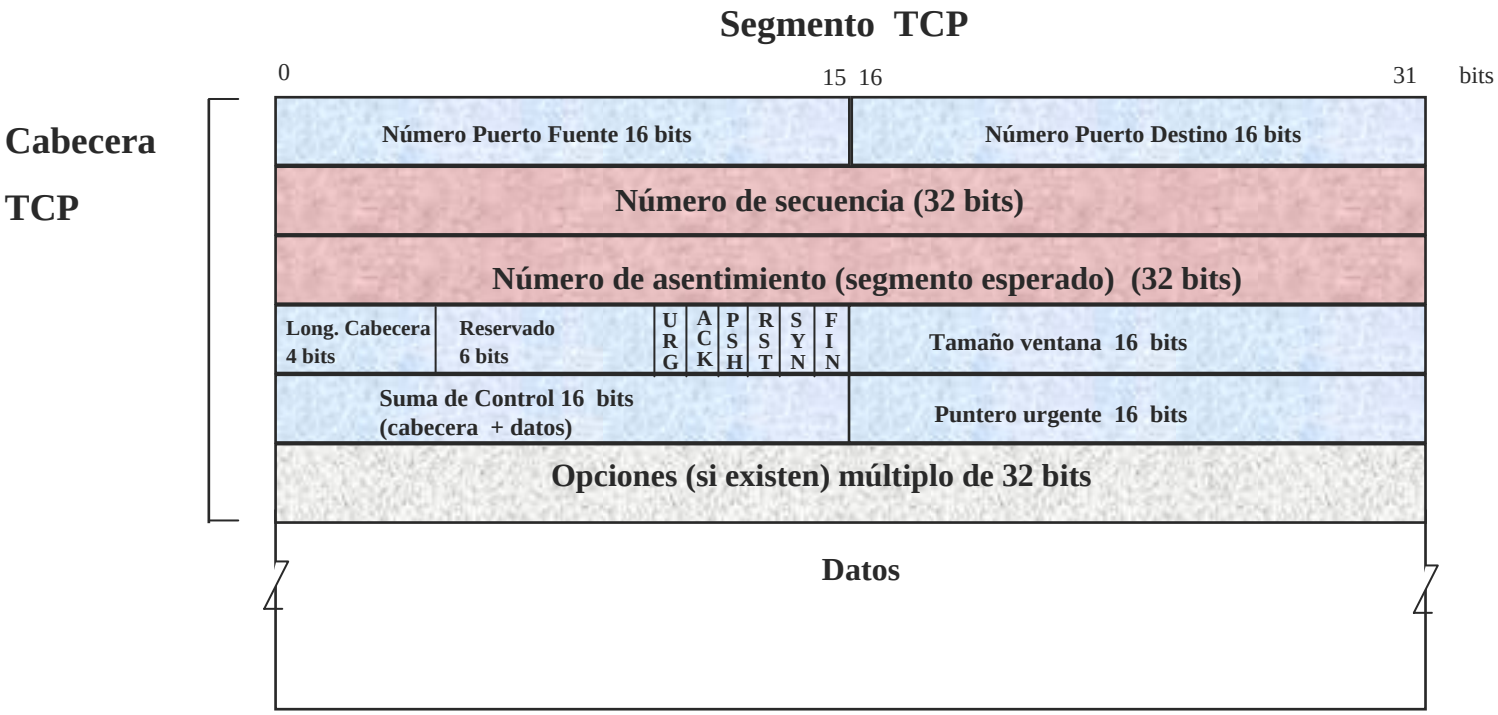
Capa Transporte: TCP (3)

- Los segmentos TCP pueden llegar duplicados. TCP los elimina
- TCP proporciona un control de flujo (tamaño de la ventana). Adecua emisor rápido con un receptor lento
- Una conexión TCP actúa como un flujo de bytes. TCP no interpreta su contenido



Segmento TCP





Cabecera TCP

- Tipos de trama:
  - **SYN** : establecimiento de la conexión
  - **ACK** : indica que incorpora un asentimiento (se aprovecha la transmisión de datos para emitir los asentimientos
  - **URG** : indica que el puntero urgente es válido (offset que se debe añadir al número de secuencia)
  - **PSH** :el receptor debe pasar los datos a la aplicación tan rápido como pueda
  - **RST** : reiniciar la conexión
  - **FIN** : el emisor finaliza la emisión de datos

Opciones TCP

- Opciones:

Fin de lista de opciones

Tipo=0
1 byte

Operación nula

Tipo=1
1 byte

Tamaño máximo de segmento

Tipo=2	long=4	Tamaño máximo segmento (**MSS**)
1 byte	1 byte	2 bytes

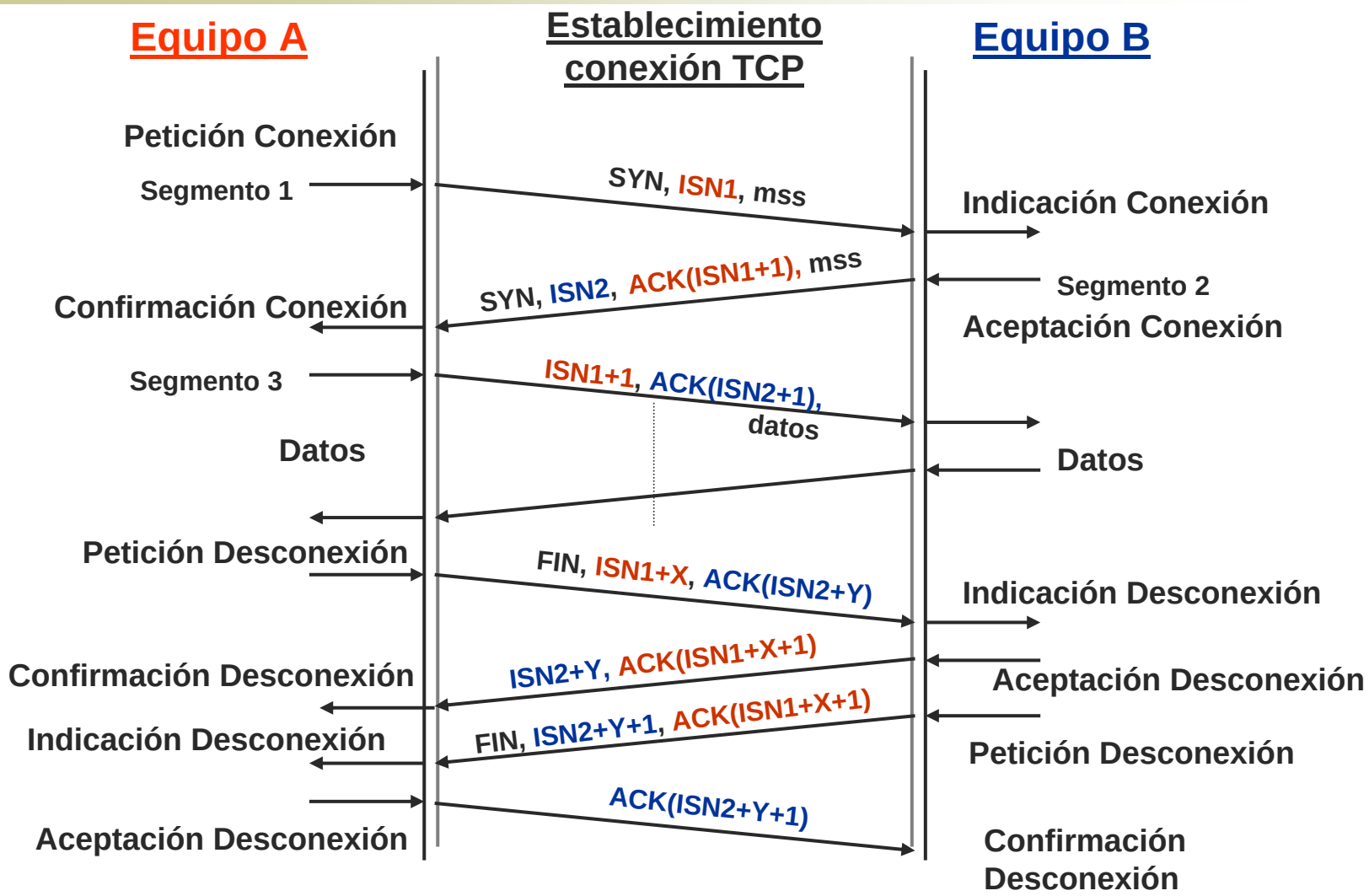
Factor de escala de la ventana

Tipo=3	long=3	valor
1 byte	1 byte	1 byte

Grabación de Tiempo

Tipo=8	long = 10	Valor tiempo	tiempo respuesta de eco
1 byte	1 byte	4 bytes	4 bytes

Conexión-Desconexión TCP



Broadcom NetXtreme Gigabit Ethernet Driver (Microsoft's Packet Scheduler) : Capturing - Wireshark

File Edit View Go Capture Analyze Statistics Help

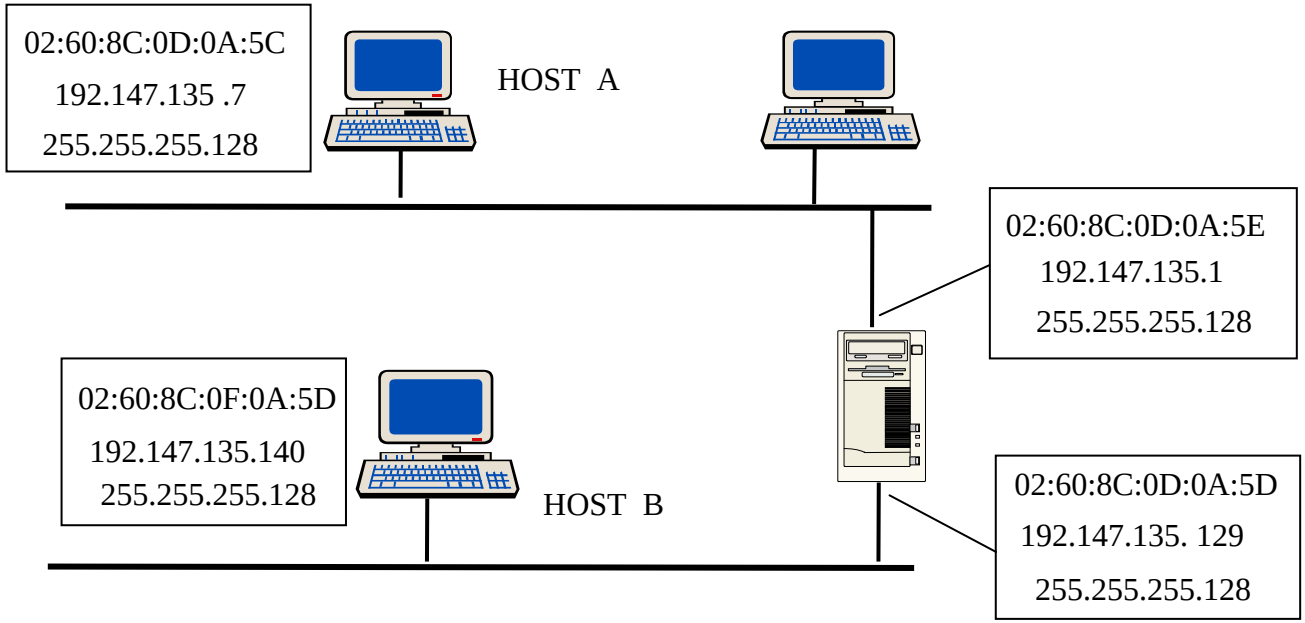
Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.60.40	192.168.60.10	TCP	40509 > 6666 [SYN] Seq=0 Win=65535 Len=0 MSS=1460
2	0.000429	192.168.60.10	192.168.60.40	TCP	6666 > 40509 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
3	0.000482	192.168.60.40	192.168.60.10	TCP	40509 > 6666 [ACK] Seq=1 Ack=1 Win=65535 Len=0
4	29.592870	192.168.60.40	192.168.60.10	TCP	40509 > 6666 [PSH, ACK] Seq=1 Ack=1 Win=65535 Len=9
5	29.797451	192.168.60.10	192.168.60.40	TCP	6666 > 40509 [ACK] Seq=1 Ack=10 Win=65526 Len=0
6	36.325286	192.168.60.10	192.168.60.40	TCP	6666 > 40509 [PSH, ACK] Seq=1 Ack=10 Win=65526 Len=9
7	36.446237	192.168.60.40	192.168.60.10	TCP	40509 > 6666 [ACK] Seq=10 Ack=10 Win=65526 Len=0
8	68.047166	192.168.60.40	192.168.60.10	TCP	40509 > 6666 [FIN, ACK] Seq=10 Ack=10 Win=65526 Len=0
9	68.047391	192.168.60.10	192.168.60.40	TCP	6666 > 40509 [ACK] Seq=10 Ack=11 Win=65526 Len=0
10	68.047879	192.168.60.10	192.168.60.40	TCP	6666 > 40509 [FIN, ACK] Seq=10 Ack=11 Win=65526 Len=0
11	68.047905	192.168.60.40	192.168.60.10	TCP	40509 > 6666 [ACK] Seq=11 Ack=11 Win=65526 Len=0

```
⊞ Frame 11 (54 bytes on wire, 54 bytes captured)
⊞ Ethernet II, Src: 00:19:b9:0e:1f:50 (00:19:b9:0e:1f:50), Dst: 00:c0:9f:a7:3d:6d (00:c0:9f:a7:3d:6d)
⊞ Internet Protocol, Src: 192.168.60.40 (192.168.60.40), Dst: 192.168.60.10 (192.168.60.10)
⊟ Transmission Control Protocol, Src Port: 40509 (40509), Dst Port: 6666 (6666), Seq: 11, Ack: 11, Len: 0
    Source port: 40509 (40509)
    Destination port: 6666 (6666)
    Sequence number: 11    (relative sequence number)
    Acknowledgement number: 11    (relative ack number)
    Header length: 20 bytes
  ⊞ Flags: 0x10 (ACK)
    Window size: 65526
  ⊞ Checksum: 0xf99d [validation disabled]
  ⊟ [SEQ/ACK analysis]
      [This is an ACK to the segment in frame: 10]
      [The RTT to ACK the segment was: 0.000026000 seconds]
```

```
0000  00 c0 9f a7 3d 6d 00 19  b9 0e 1f 50 08 00 45 00   ....=m.. ...P..E.
0010  00 28 b6 93 40 00 80 06  4a b9 c0 a8 3c 28 c0 a8   .(..@... J...<(..
0020  3c 0a 9e 3d 1a 0a a3 c3  74 f7 76 d2 45 a2 50 10   <..=.... t.v.E.P.
0030  ff f6 f9 9d 00 00                                  ......
```

Broadcom NetXtreme Gigabit Ethernet Driver (Microsoft's Packet Scheduler) : <live capture in progress> File: C... | Packets: 11 Displ... | Profile: Default

Ejercicio Protocolos TCP/IP

Describir la secuencia de mensajes y protocolos involucrados en una transmisión UDP entre el nodo A y el nodo B descritos en la figura.