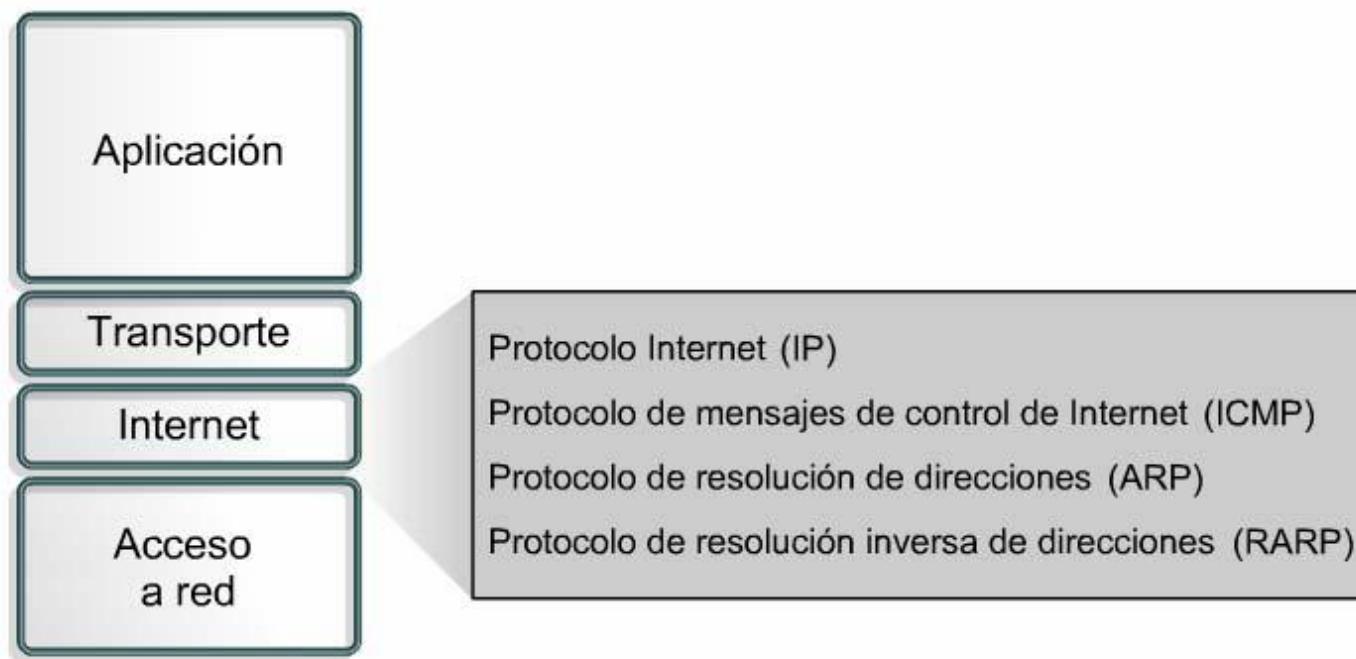
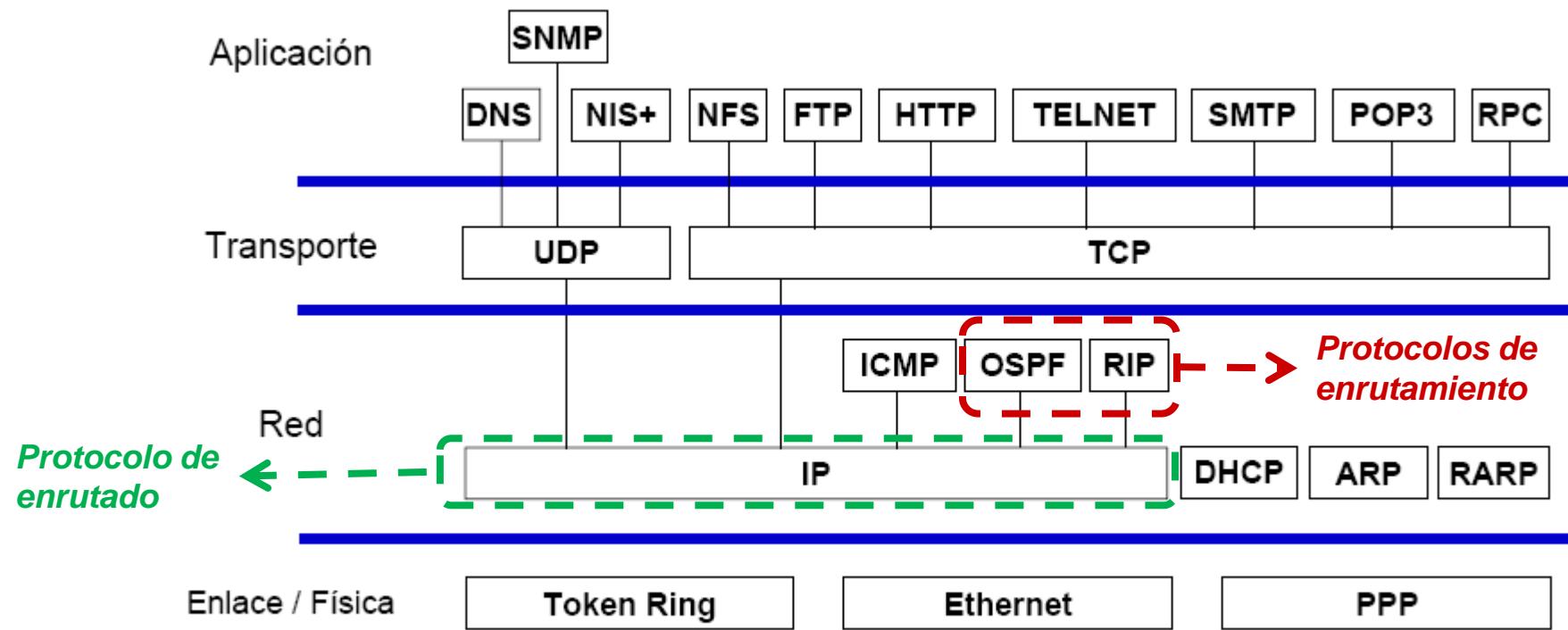


# Tema 5: Interconexión de redes.

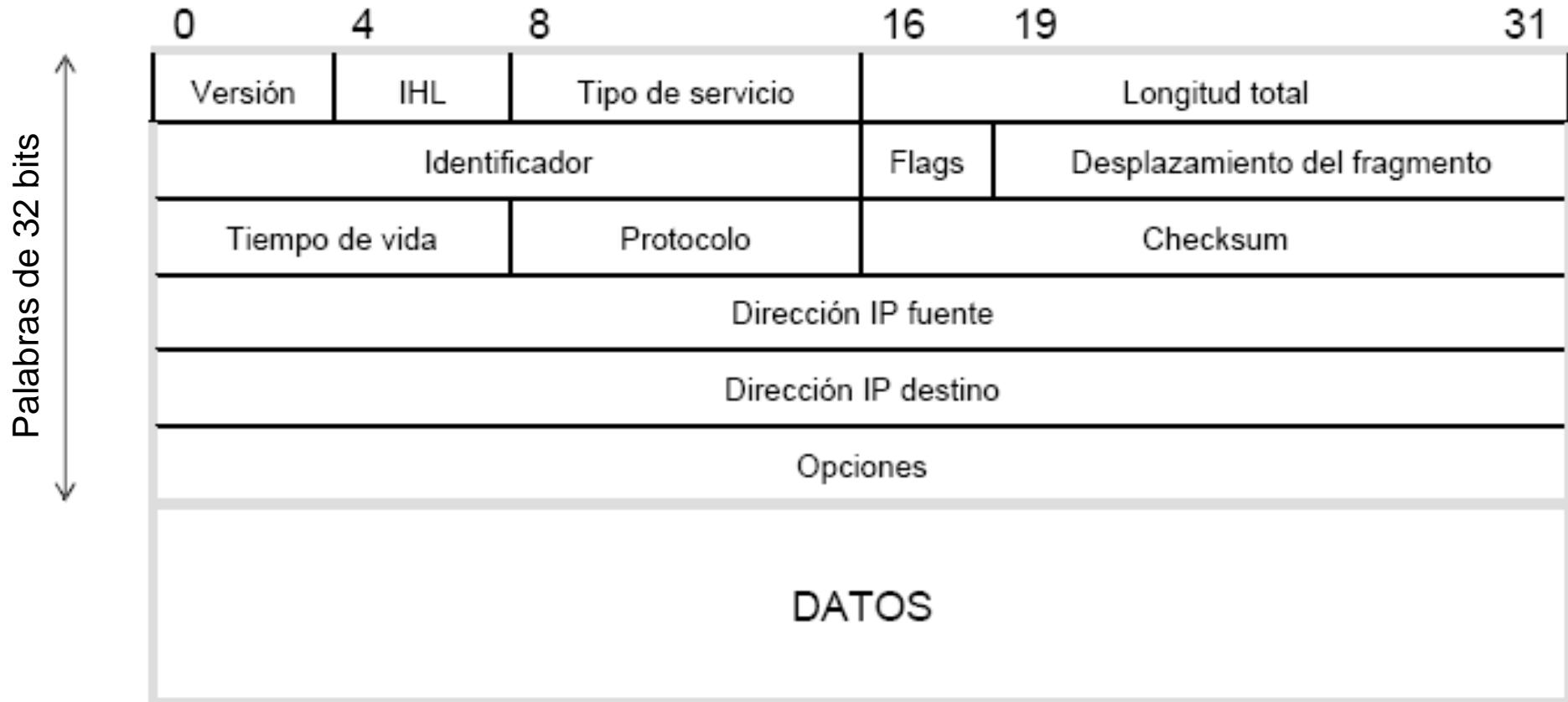
Fundamentos de Redes de Computadores

Grado en Ingeniería Informática (2º curso)  
Escuela Técnica Superior de Ingeniería. Universidad de Huelva





- **Protocolo de red de Internet (Internet Protocol)**
  - Proporciona un servicio básico de entrega de paquetes
  - Protocolo **no orientado a conexión** (no fiable)
    - No realiza detección ni recuperación de paquetes perdidos o erróneos
    - No garantiza que los paquetes lleguen en orden
    - No garantiza la detección de paquetes duplicados
- **Funciones básicas del protocolo IP**
  - Fragmentación y reensamblaje de paquetes
    - División del paquetes en fragmentos de un tamaño aceptable por la red
  - **Direccionamiento y encaminamiento de datagramas**
    - Encaminado de paquetes atendiendo a información de tabla de rutas
    - La construcción de tablas de rutas puede ser:
      - Manual (routing estático)
      - Mediante algún protocolo de routing dinámico: RIP, OSPF, BGP, etc.



## Campos de la cabecera IP

- **Versión:** Valor=4 (IPv4)
- **IHL**
  - Longitud de la cabecera, en palabras de 32 bits.
  - Campo IHL ocupa 4 bits
  - Tamaño máximo de la cabecera = 15 palabras (60 bytes)
- **Tipo de servicio**

0	1	2	3	4	5	6	7
		Prioridad	Calidad de servicio (QoS)			Reserv.	

  - **Prioridad**
    - Especifica la prioridad del datagrama (hasta 8 niveles).
    - Un paquete de alta prioridad debe ser reexpedido por un router antes que un paquete de baja prioridad (aunque este llegase antes)
  - **QoS:** Puede tomar los siguientes valores
    - 1000 Minimizar retardo
    - 0100 Maximizar rendimiento (velocidad de transmisión)
    - 0010 Maximizar fiabilidad (seguridad en la entrega)
    - 0001 Minimizar coste monetario
    - 0000 Servicio normal

- **Longitud total**
  - Longitud del datagrama (cabecera + datos) medida en bytes.
  - Campo Longitud Total ocupa 16 bits
  - Longitud máxima del datagrama: 216 bytes = 64 Kbytes
- **Identificador**
  - Número de 16 bits que identifica al datagrama
- **Flags**
  - **MF (More Fragments)**: si está a 1 indica que hay más fragmentos
  - **DF (Don´t Fragment)**: si es 1 prohíbe la fragmentación
- **Desplazamiento del fragmento. OFFSET**
  - Nº secuencia del fragmento (unidades = 8 bytes)
- **Tiempo de vida (TTL, *Time To Live*)**
  - Nº encaminadores que puede atravesar el paquete
  - Cuando TTL=0 el paquete debe ser descartado

- **Protocolo**
  - Protocolo transportado sobre IP.
    - 1: Internet Control Message Protocol (ICMP)
    - 2: Internet Group Management Protocol (IGMP)
    - 6: Transmission Control Protocol (TCP)
    - 8: Exterior Gateway Protocol (EGP)
    - 17: User Datagram Protocol (UDP)
    - 41: IP Version 6 (IPv6)
    - 89: Open Shortest Path First (OSPF)
- **Checksum**
  - Suma de control de la cabecera
- **Direcciones IP origen y destino**
  - Identifican al equipo (host) emisor y receptor del paquete
- **Opciones**
  - Campo opcional, con opciones especiales.
  - Ejemplos: encaminamiento de origen, sello de ruta, sello de tiempo, etc.
  - Tamaño máximo del campo opciones: 10 palabras.

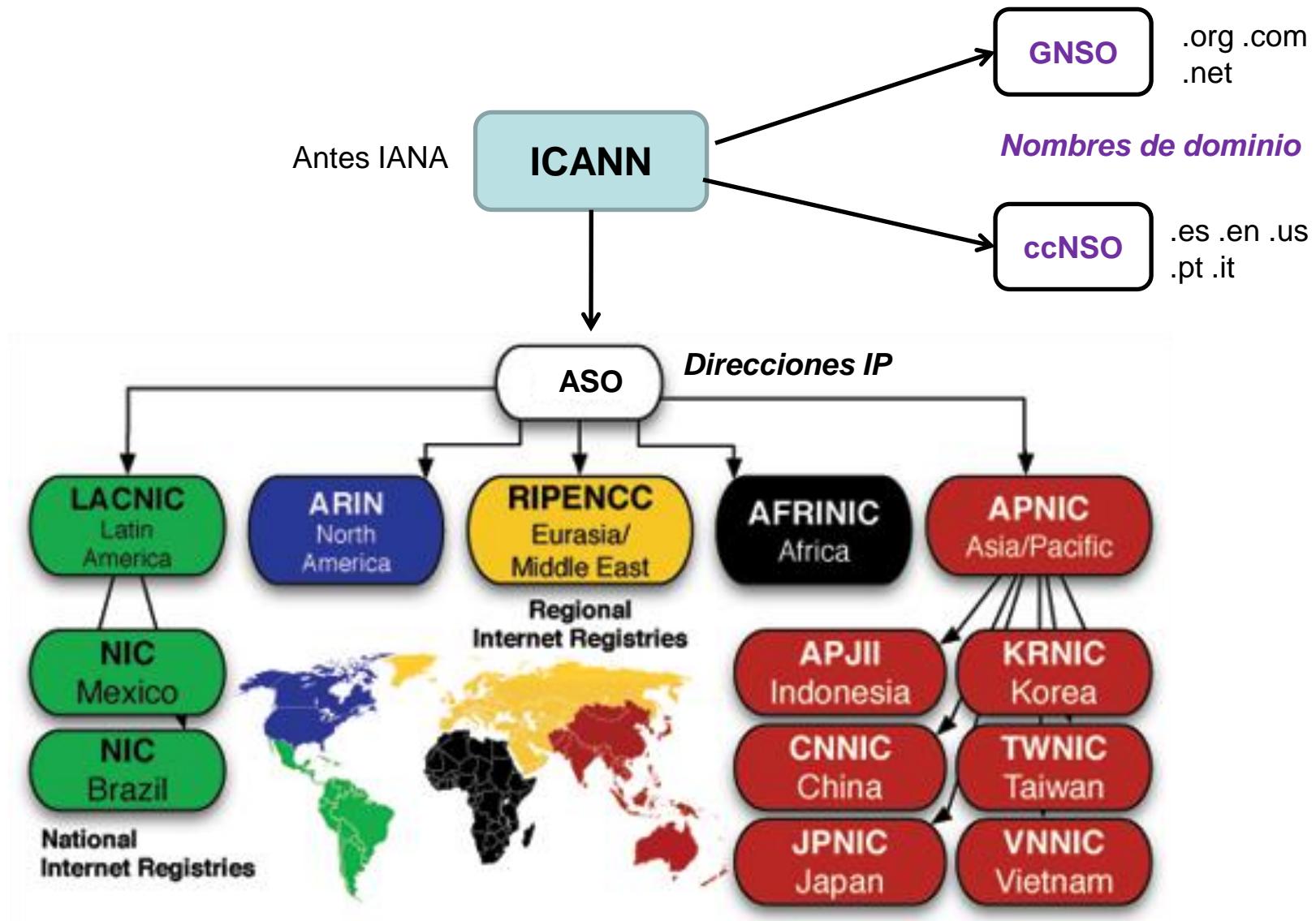
## Direcciones IPv4

- Constan de 4 bytes (32 bits).
- Se adjunta una máscara que indica qué parte es común en esa red
- Para expresarlas se utiliza la “notación de punto”.

**$128.2.7.9 = 10000000 . 00000010 . 00000111 . 00001001$  (IP)**

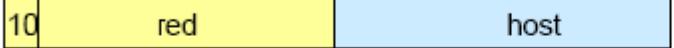
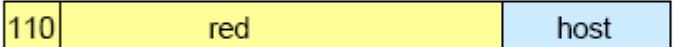
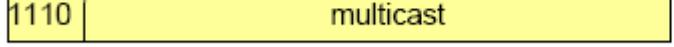
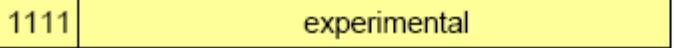
**$255.255.252.0 = 11111111 . 11111111 . 11111100 . 00000000$  (máscara)**

- Tipos de direcciones IPv4
  - Unicast: Un único host (único equipo).
  - Multicast: Un grupo de hosts (empiezan por 224....)
  - Broadcast: Todos los hosts dentro de mi red local ( $r.r.255.255$ )



- La parte que identifica a la red es fija para cada red y es necesario solicitarla a una de las entidades regionales de registro de Internet (RIR, Regional Internet Registries):
  - **ARIN (American Registry for Internet Numbers)**: es responsable de administrar y registrar las direcciones IP en NorteAmérica
  - **RIPE (Reseaux IP Europeens)**: es responsable de administrar y registrar las direcciones IP en Europa y Oriente Medio
  - **APNIC (Asia Pacific Network Information Center)**: es responsable de administrar y registrar las direcciones IP en la región de AsiaPacífico
  - ...
- **ICANN (antes IANA) -> ASO -> RIR (ARIN, RIPE, APNIC...)**
- Las operadoras lo piden al RIR, y éstas los distribuyen a sus clientes (a un módico precio).

- Direcciones IPv4 basadas en clases (classfull)

<b>Clase A</b>	7 bits  [ 0 - 127 ]	24 bits	$2^7 = 128$ redes $2^{24} = 16.777.216$ hosts Ejemplo: <b>26.56.120.9</b>
<b>Clase B</b>	14 bits  [ 128 - 191 ] [ 0 - 255 ]	16 bits	$2^{14} = 16.384$ redes $2^{16} = 65.536$ hosts Ejemplo: <b>147.96.50.110</b>
<b>Clase C</b>	21 bits  [ 192 - 223 ] [ 0 - 255 ] [ 0 - 255 ]	8 bits	$2^{21} = 2.097.152$ redes $2^8 = 256$ hosts Ejemplo: <b>217.6.95.44</b>
<b>Clase D</b>	28 bits  [ 224 - 239 ] [ 0 - 255 ] [ 0 - 255 ] [ 0 - 255 ]		Ejemplo: <b>224.0.0.1</b>
<b>Clase E</b>	28 bits 		

- CIDR: Classless Interdomain Routing: uso de subredes y máscaras.

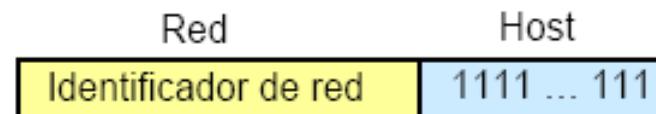
## Direcciones IP especiales (1)

### Loopback (127.x.y.z)

- Direcciones de bucle interno (loopback)
- Casi todas las máquinas usan como dirección de loopback la **127.0.0.1**
- También la 0.0.0.0 es considerada de loopback, aunque está en desuso.

### Direcciones broadcast (terminadas en 11...111)

- Se utilizan para enviar un paquete a todas las máquinas de la red local
- Formato de las direcciones broadcast
- Todos los bits de identificador de host se ponen a valor 1
- Último valor del rango de direcciones de la red



## Direcciones IP especiales (2)

### Ejemplos de direcciones broadcast

- Red de clase A:  
Red (8) Host (24)  

00011011.	1111111.11111111.11111111
-----------	---------------------------

 = 27.255.255.255
  
- Red de clase B:  
Red (16) Host (16)  

10001110.01011000.	11111111.11111111
--------------------	-------------------

 = 142.88.255.255
  
- Red de clase C:  
Red (24) Host (8)  

11000111.01000011.11101111.	11111111
-----------------------------	----------

 = 199.67.239.255
  
- Red sin clase (n=14):  
Red (18) Host (14)  

01011010.00100000.10	111111.11111111
----------------------	-----------------

 = 90.32.191.255
  
- Red sin clase (n=7):  
Red (27) Host (5)  

10001111.00011010.00000111.011	11111
--------------------------------	-------

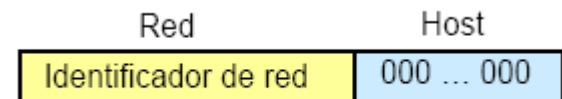
 = 143.26.7.127
  
- Dir. broadcast universal: 255.255.255.255

## Direcciones IP especiales (3)

- **Direcciones de red (terminadas en 00...000)**
    - Se utilizan para representar a una red completa en las tablas de encaminamiento
    - Nunca se utilizan como dirección destino ni se asignan a un host concreto
    - Ejemplo de tabla de rutas en Linux (orden **netstat -nr**)

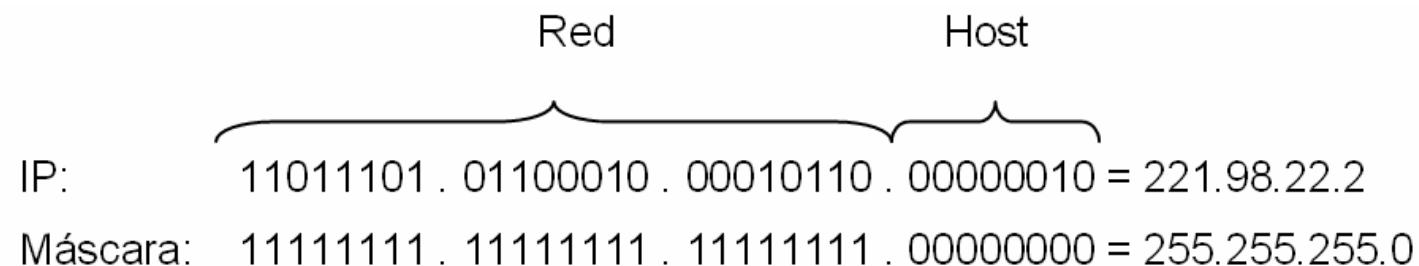
Kernel IP routing table					
Destination	Gateway	Genmask	Flags	Iface	
192.168.1.0	0.0.0.0	255.255.255.0	U	eth0	
192.168.2.0	0.0.0.0	255.255.255.0	U	eth1	
0.0.0.0	192.168.1.1	255.255.255.0	UG	eth0	

- **Formato de las direcciones de red**
    - Todos los bits de identificador de host se ponen a valor 0
    - Primer valor del rango de direcciones de la red



## Máscaras de red (1)

- **La máscara de red indica:**
  - Qué parte de la dirección IP identifican a la red
    - Bits de la máscara a 1
  - Qué parte de la dirección IP identifican al ordenador (host) dentro de la red
    - Bits de la máscara a 0
- **Ejemplo**
  - Dirección de clase C: 221.98.22.2
  - Máscara: 255.255.255.0
  - Notación alternativa: 221.98.22.2/24
  - El valor **/24** indica la longitud de la parte de red (nº de unos de la máscara)



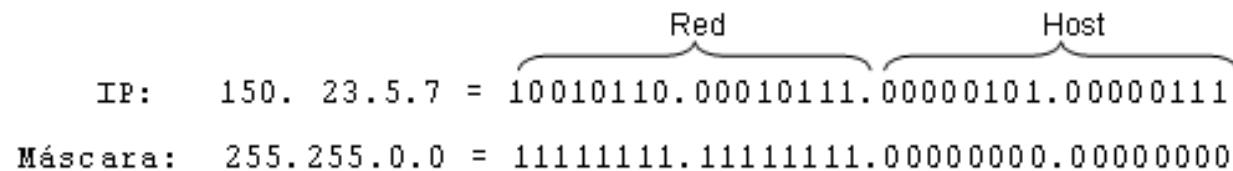
- Máscaras de red (2). Ejemplos de máscaras**

				Notación alternativa
Red (8)	Host (24)			
Dir. de clase A =	<b>00011011.00000111.10000010.00000011</b>	= 27.7.130.3		
Máscara =	<b>11111111.00000000.00000000.00000000</b>	= 255.0.0.0		
				<b>27.7.130.3/8</b>
Red (16)	Host (16)			
Dir. de clase B =	<b>10001110.01011000.00001100.00000100</b>	= 142.88.12.4		
Máscara =	<b>11111111.11111111.00000000.00000000</b>	= 255.255.0.0		
				<b>142.88.12.4/16</b>
Red (24)	Host (8)			
Dir. de clase C =	<b>11000111.01000011.11101111.00000110</b>	= 199.67.239.6		
Máscara =	<b>11111111.11111111.00000000.00000000</b>	= 255.255.255.0		
				<b>199.67.239.6/24</b>
Red (18)	Host (14)			
Dir. sin clase =	<b>01011010.00100000.10000011.00000101</b>	= 90.32.131.5		
Máscara =	<b>11111111.11111111.11000000.00000000</b>	= 255.255.192.0		
				<b>90.32.121.5/18</b>
Red (27)	Host (5)			
Dir. sin clase =	<b>10001111.00011010.00000111.01100011</b>	= 143.26.7.99		
Máscara =	<b>11111111.11111111.11111111.11100000</b>	= 255.255.255.224		
				<b>143.26.7.99/27</b>

**Subnetting:** Supongamos la red de la clase B: 150.23.0.0

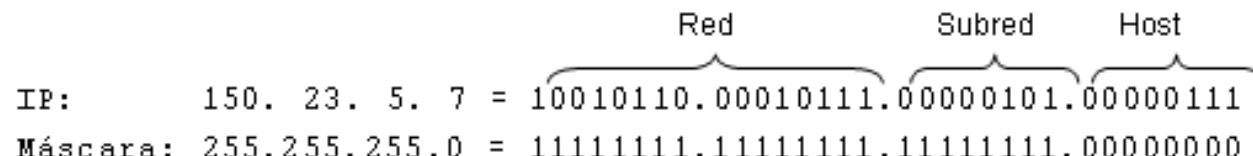
- Tenemos 16 bits para identificar a host ( $2^{16}$  hosts)

IP: 150. 23. 5. 7 = 10010110.00010111.00000101.00000111  
Máscara: 255.255.0.0 = 11111111.11111111.00000000.00000000



- Esta red se puede dividir, por ejemplo, en 256 subredes con 256 hosts cada una
- Usamos 8 bits para identificar a la subred ( $2^8 = 256$  subredes)
- Usamos 8 bits para identificar a host ( $2^8 = 256$  hosts)
- Nos queda la siguiente organización:
  - Subred 0: 150.23.0.0 (Dpto. de administración)
  - Subred 1: 150.23.1.0 (Dpto. de RRHH)
  - .....
  - Subred 255: 150.23.255.0 (Dpto. comercial)
- Por tanto la máscara de subred adecuada es la siguiente:

IP: 150. 23. 5. 7 = 10010110.00010111.00000101.00000111  
Máscara: 255.255.255.0 = 11111111.11111111.11111111.00000000



Sea la red de la clase C: 192.168.44.0. Se desea dividir la red en 8 subredes de igual tamaño. ¿Cuál ha de ser la máscara para esas 8 subredes?

- a) ¿Es la misma para las 8?
- b) ¿Cuáles son los primeros 3 octetos de esa máscara?
- c) ¿Cuántas direcciones contendrá cada subred?

3 bits para identificar la subred ( $2^3 = 8$  subredes)

5 bits para identificar el host ( $2^5 = 32$  hosts por subred)

IP:	192.168.44.x	=	11000000.10101000.00101100.	ssshhhhh
Máscara:	255.255.255.224	=	11111111.11111111.11111111.11100000	

Organización resultante:

**Subred 192.168.44.0**

- hosts: de 192.168.44.1 al 192.168.44.30
- broadcast : 192.168.44.31

**Subred 192.168.44.32**

- hosts: de 192.168.44.33 al 192.168.44.62
- broadcast : 192.168.44.63

**Subred 192.168.44.64**

- hosts: de 192.168.44.65. al 192.168.44.94
- broadcast: 192.168.44.95

**Subred 192.168.44.96**

- hosts: de 192.168.44.97 al 192.168.44.126
- broadcast: 192.168.44.127

**Subred 192.168.44.128**

- hosts: de 192.168.44.129 al 192.168.44.158
- broadcast: 192.168.44.159

**Subred 192.168.44.160**

- hosts: de 192.168.44.161 al 192.168.44.190
- broadcast: 192.168.44.191

**Subred 192.168.44.192**

- hosts: de 192.168.44.193 al 192.168.44.222
- broadcast: 192.168.44.223

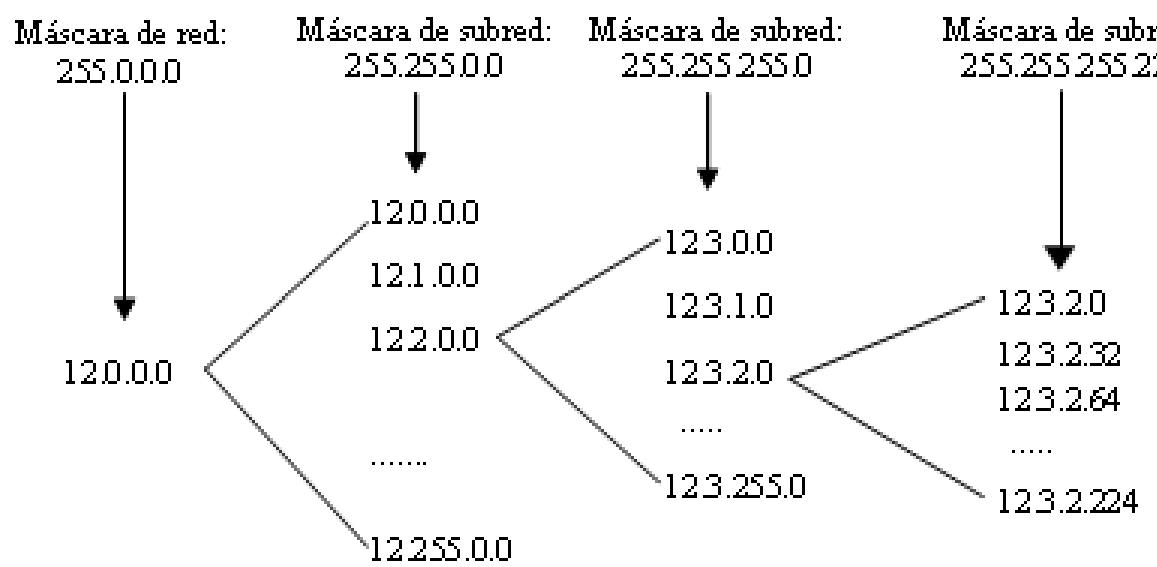
**Subred 192.168.44.224**

- hosts: de 192.168.44.225 al 192.168.44.254
- broadcast: 192.168.44.255

## VLSM: Máscaras de subred de dirección variable

- Muchas organizaciones utilizan un sistema de organización jerárquica de direcciones de red
- La red se divide en subredes, las cuales, a su vez se pueden dividir en varias subredes
- La longitud de la máscara de subred puede ser variable, en función de la subred en la que nos encontremos

**Ejemplo 1:** Supongamos la red de clase A 12.0.0.0, organizada de la siguiente manera



Dirección de subred	Máscara de subred
12.0.0.0	255.255.0.0
12.3.0.0	255.255.255.0
12.3.2.0	255.255.255.0
12.3.3.0	255.255.255.0
12.3.4.0	255.255.255.0
.....	.....
12.3.255.0	255.255.255.0
12.3.2.0	255.255.255.224
12.3.2.32	255.255.255.224
.....	.....
12.3.2.224	255.255.255.224

- **Ejemplo 2:** Supongamos que una empresa con una red de clase C (200.21.32.0) quiere dividir el espacio de direcciones en cinco subredes del siguiente tamaño:
  - Subred 1: 60 hosts
  - Subred 2: 50 hosts
  - Subred 3: 40 hosts
  - Subred 4: 30 hosts
  - Subred 5: 20 hosts

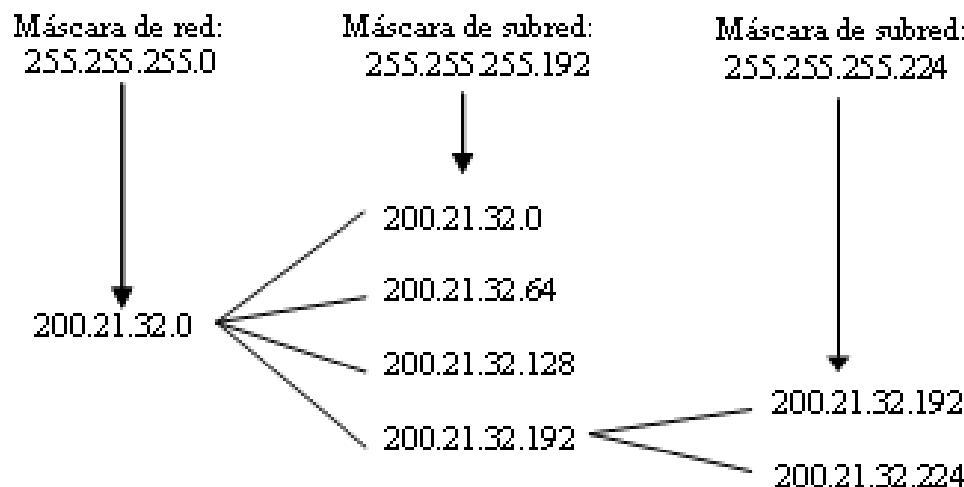
Se puede adoptar la siguiente solución

Dividir la red en 4 ( $2^2$ ) subredes de 62 ( $2^6-2$ ) hosts cada una

Máscara = 11111111 . 11111111 . 11111111 . 11000000 = **255.255.255.192**

Subdividir una de las redes en dos subredes de 30 ( $2^4-2$ ) hosts cada una

Máscara = 11111111 . 11111111 . 11111111 . 11100000 = **255.255.255.240**



Dirección de subred	Máscara de subred
200.21.32.0	255.255.255.192
200.21.32.64	255.255.255.192
200.21.32.128	255.255.255.192
200.21.32.192	255.255.255.224
200.21.32.224	255.255.255.224

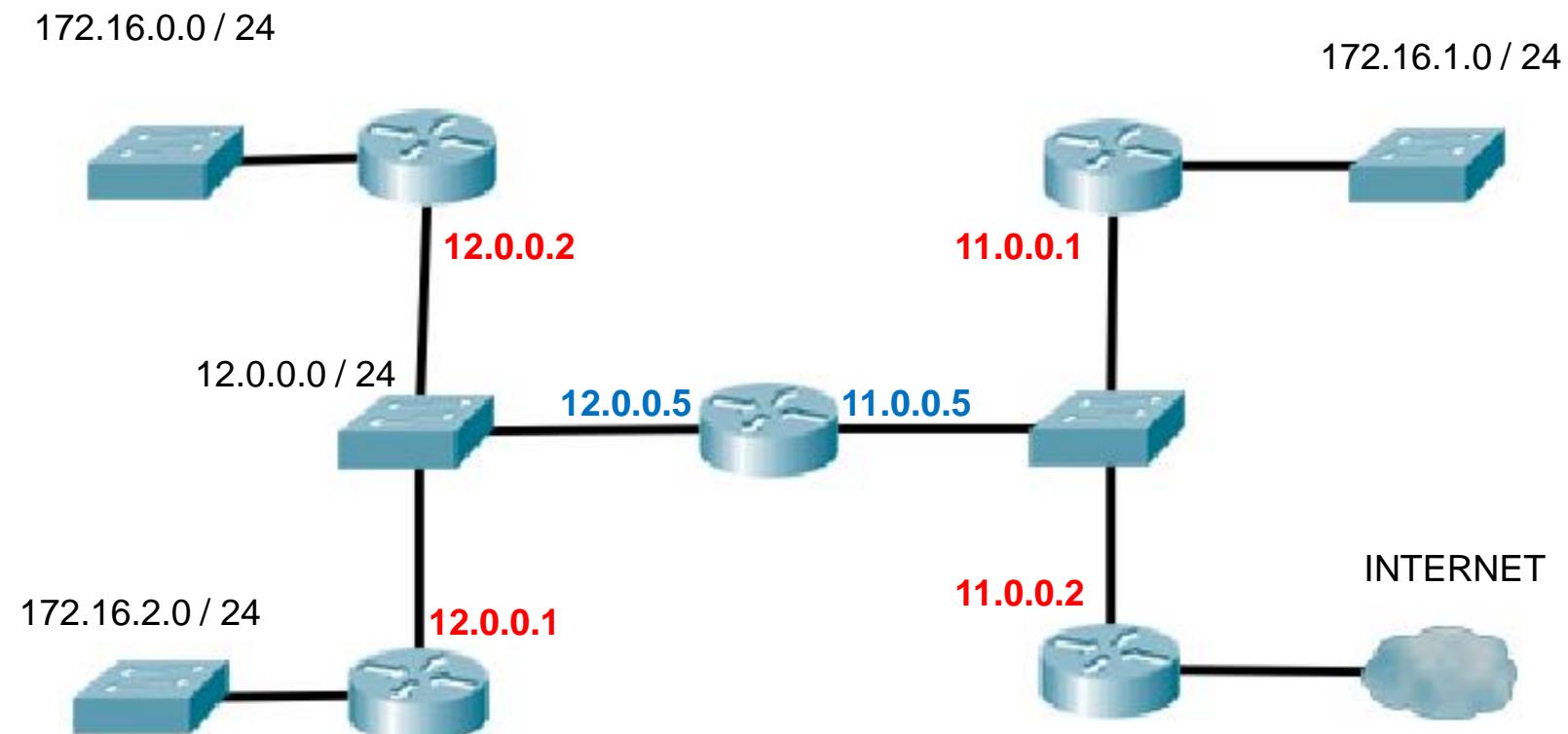
Los routers utilizan una “**tabla de enrutamiento**” que incluye:

- **Dirección de red** de destino.
- **IP** (o Interfaz usada, si es punto a punto) **del próximo router** hacia la red destino.
- **Métrica**: distancia a la red destino.

El router evalúa la mejor ruta hasta la red destino en función de dicha métrica.

Las tablas son transmitidas a otros routers y recalculadas en función de dicha información periódicamente (**rutas dinámicas**). Existen varios protocolos para ello: RIP, IGRP, EIGRP, BGP, OSPF, denominados protocolos de “**enrutamiento**” en contraste con IP al que se le suele llamar protocolo de “**enrutado**”.

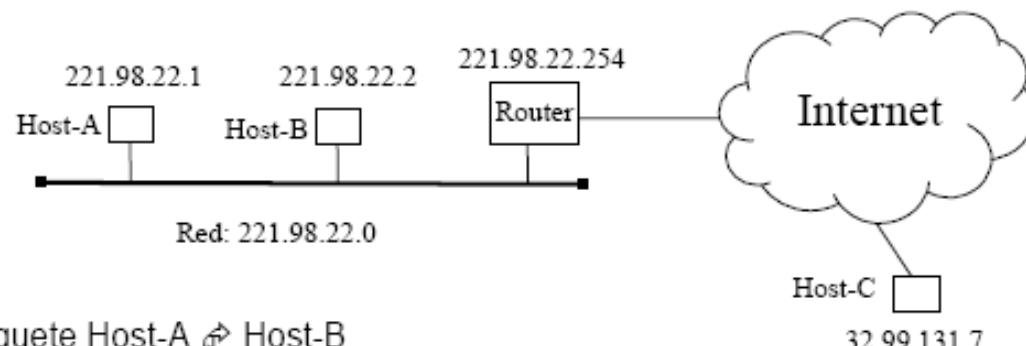
También se pueden añadir rutas de forma manual (**rutas estáticas**)



La siguiente tabla de un router se interpreta así:

11.0.0.0/8 is variably subnetted, 2 subnets, 2 masks  
C 11.0.0.0/24 is directly connected, GigabitEthernet0/0  
L 11.0.0.5/32 is directly connected, GigabitEthernet0/0  
12.0.0.0/8 is variably subnetted, 2 subnets, 2 masks  
C 12.0.0.0/24 is directly connected, GigabitEthernet0/1  
L 12.0.0.5/32 is directly connected, GigabitEthernet0/1  
172.16.0.0/24 is subnetted, 3 subnets  
S 172.16.0.0/24 [1/1] via 12.0.0.2  
S 172.16.1.0/24 [1/1] via 11.0.0.1  
S 172.16.2.0/24 [1/1] via 12.0.0.1  
S\* 0.0.0.0/0 [1/0] via 11.0.0.2

Subred	IP siguiente salto	Métrica
11.0.0.0/24	local	0
12.0.0.0/30	local	0
172.16.0.0/24	12.0.0.2	1
172.16.1.0/24	11.0.0.1	1
172.16.2.0/24	12.0.0.1	1
El resto (salida por defecto)	11.0.0.2	-



Enviar paquete Host-A  $\rightarrow$  Host-B

Host-A envía paquete directamente a través de su red local

Enviar paquete Host-A  $\rightarrow$  Host-C

Host-A envía el paquete al router y este se encargará de encaminarlo hasta su destino

Para saber como tiene que tratar el paquete, el Host-A tiene que realizar las siguientes operaciones:

Aplicar la máscara de red a la dirección destino

convierte la dirección del host destino en una dirección de red

Consultar la tabla de encaminamiento

decide a quien debe entregar el paquete (host destino o router)

La máscara de red es en nuestro caso el siguiente valor:

255.255.255.0

Aplicación de la máscara: realizar Y-lógica bit a bit entre la dirección destino y la máscara:

Dirección del Host-B

221.98.22.2	= 11011101 . 01100010 . 00010110 . 00000010
<u>255.255.255.0</u>	<u>= 11111111 . 11111111 . 11111111 . 00000000</u>
221.98.22.0	= 11011101 . 01100010 . 00010110 . 00000000

Dirección del Host-C

32.99.131.7	= 00100000 . 01100010 . 10000011 . 00000111
<u>255.255.255.0</u>	<u>= 11111111 . 11111111 . 11111111 . 00000000</u>
32.99.131.0	= 00100000 . 01100010 . 10000011 . 00000000

El Host-A consulta su tabla de encaminamiento (orden **netstat -nr**)

Destination	Gateway
-----	-----
221.98.22.0	0.0.0.0
0.0.0.0 (default)	221.98.22.254
127.0.0.1	127.0.0.1

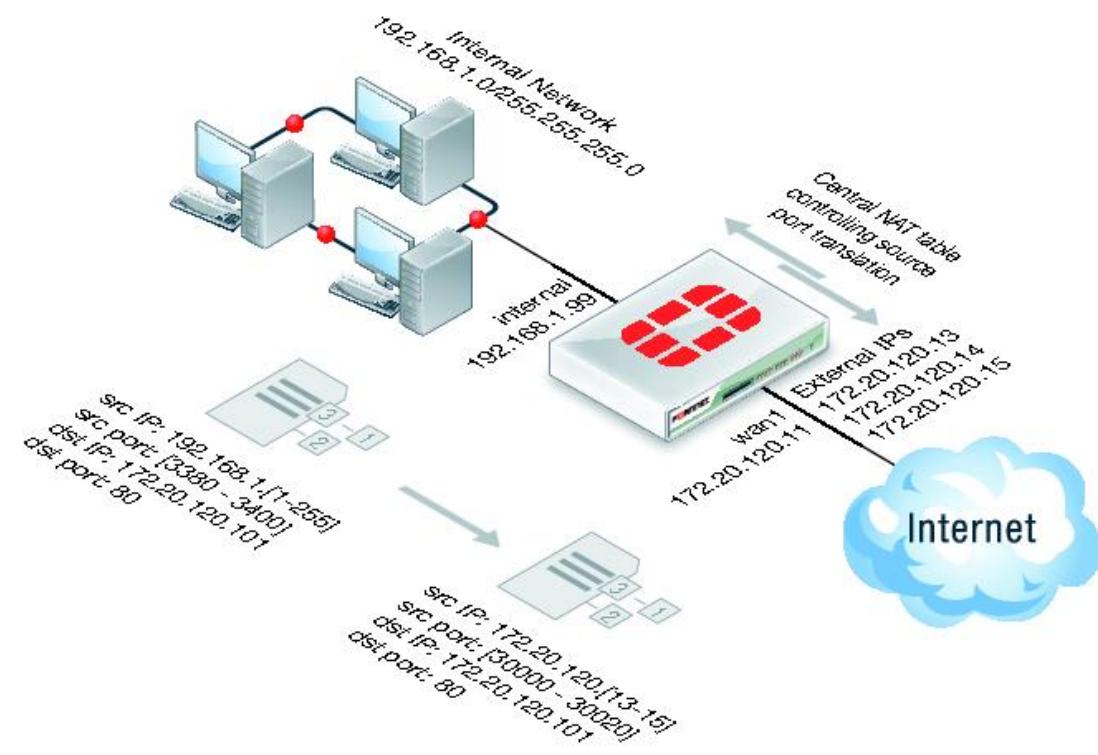
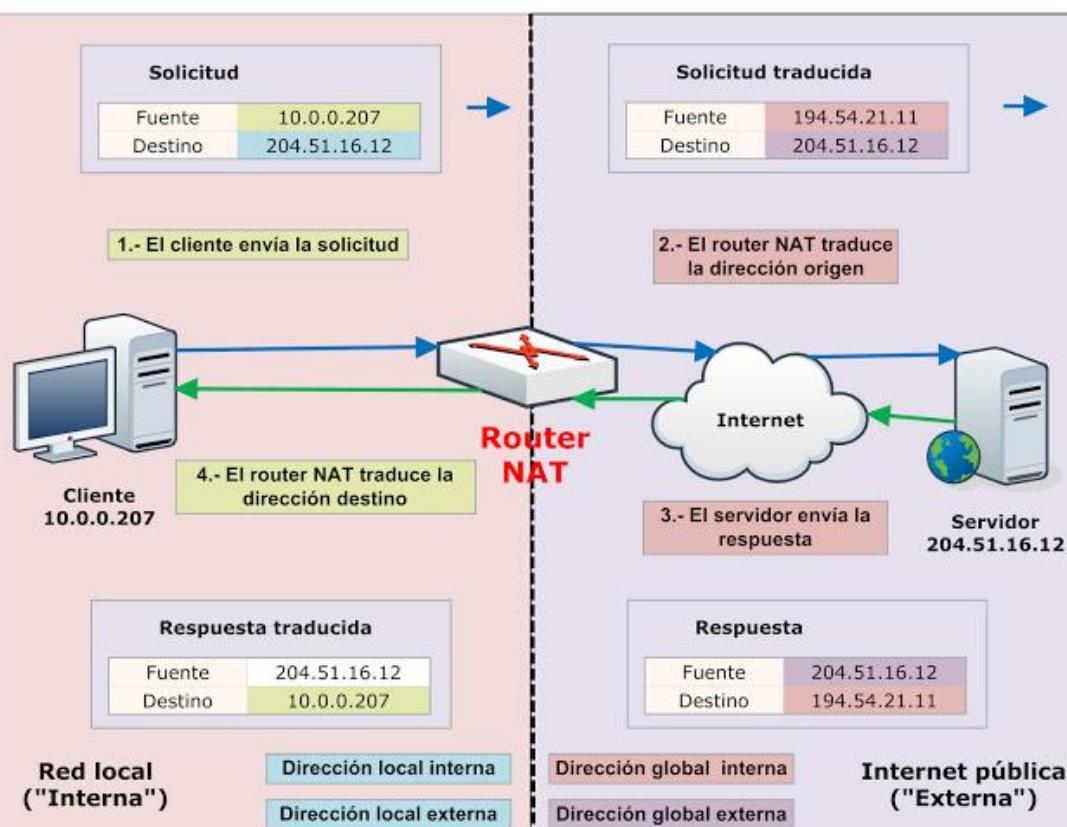
Si la dirección de Gateway asociada es el propio remitente (0.0.0.0), el paquete se debe enviar directamente a través de la red local

Si la dirección de Gateway asociada es la de un router, se usa ese router para enviar el paquete a su destino

Si la dirección no aparece en la tabla se usa el **Default Router** para enviar el paquete a su destino

## Direcciones reservadas para redes privadas

- No se puede salir a Internet con una dirección IP privada.
- Se pueden asignar a redes conectadas a Internet sólo si es a través de un router que hace traducción de direcciones de red (**NAT**).
- Tipos de NAT: Por puerto (PAT) o por IP dinámicas.
- Los rangos de direcciones IP privadas son los siguientes:
  - **10.0.0.0 – 10.255.255.255** equivale a 1 red privada de clase A
  - **172.16.0.0 – 172.31.255.255** equivale a 16 redes privadas de clase B
  - **192.168.0.0 – 192.168.255.255** equivale a 256 redes privadas de clase C

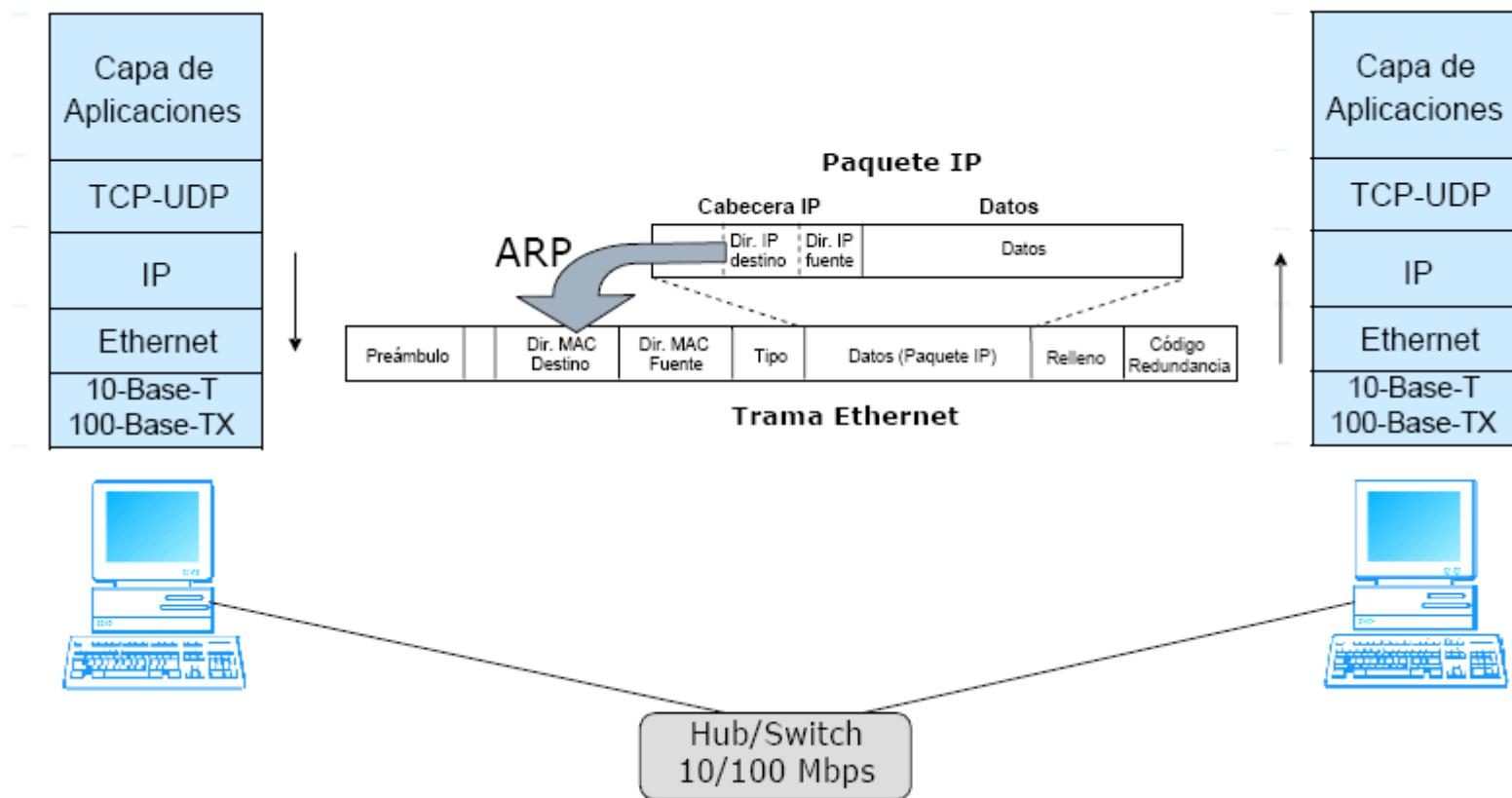


- **0.x.x.x** sólo usado como dirección origen en equipos que aún no tienen IP
- **100.64-127.x.x** Para NAT.
- **169.254.x.x** Enlace local, es decir, IP provisional mientras no se ha obtenido una por DHCP.
- **192.0.2.x** Para documentación y test en entorno local
- **192.88.99.x** Para retransmisión IPv6 a IPv4
- **198.18-19.x.x** Para tests de comunicación entre redes privadas
- **198.51.100.x** Para tests y documentación
- **203.0.113.x** Para tests y documentación
- **224-239.x.x.x** Multicast (por ejemplo RIP, OSPF...)
- **240-254.x.x.x** Reservado para uso futuro

Éstas no hay que memorizarlas, tan sólo están a título ilustrativo

## ARP: Address Resolution Protocol

- Protocolo de resolución de direcciones: Dada una IP, averigua la MAC asociada (en esa LAN).



## El protocolo ARP (Address Resolution Protocol)

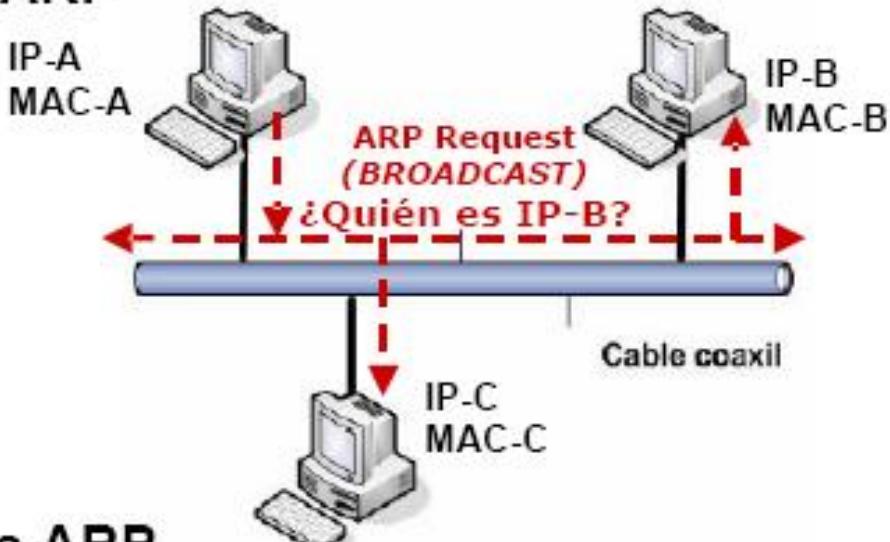
La tabla ARP Mantiene las direcciones IP de las últimas máquinas con las que nos hemos comunicado y las direcciones Ethernet asociadas

Ver la tabla arp en windows: c:\arp -a

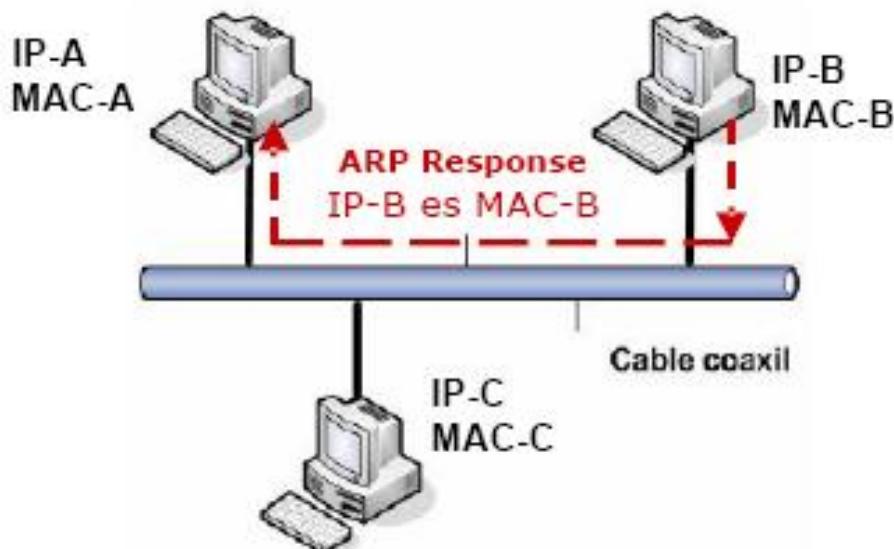
Net to Media Table					
Device	IP Address	Mask	Flags	Phys Addr	
le0	147.96.48.203	255.255.255.255		00:00:b4:c3:c8:f4	
le0	147.96.37.196	255.255.255.255		00:a0:24:57:78:3e	
le0	147.96.48.217	255.255.255.255		00:20:18:2f:1d:60	

- Funcionamiento de ARP: Si Host A quiere enviar un paquete a Host B:
  - Host A consulta su tabla ARP para ver si la dirección MAC de Host B está.
  - Si no está entonces envía un mensaje a todos los equipos de esa LAN (broadcast) preguntando por la dirección MAC de Host B (ARP Request).
  - Host B responde a Host A informándole de su dirección IP ARP Response

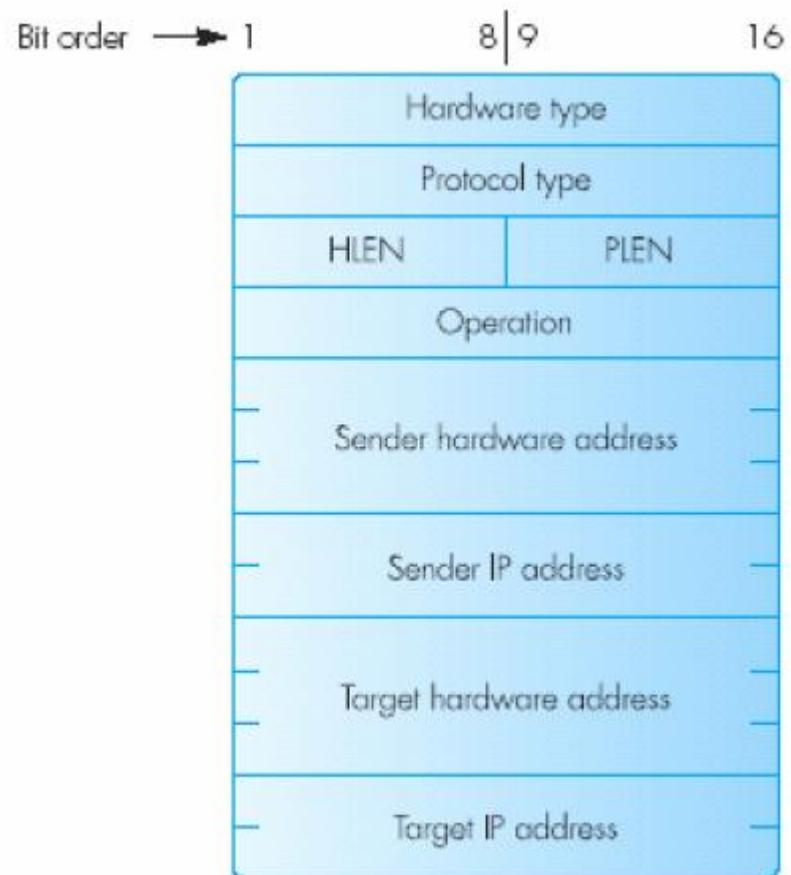
## Pregunta ARP



## Respuesta ARP



- **Formato de paquetes ARP**



HLEN = Hardware address length

PLEN = IP address length

Operation = 1 ARP request

= 2 ARP response

= 3 RARP request

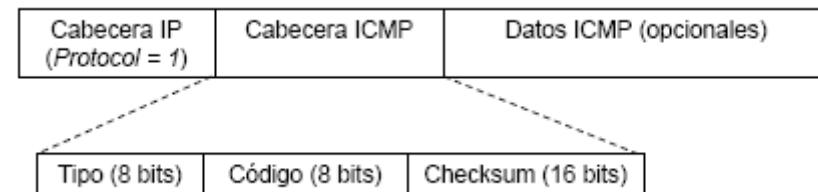
= 4 RARP response

- **RARP** o reverse ARP hace justo lo contrario, envía su MAC para que un servidor le indique su IP
- **BootP** es una evolución de RARP y además de la IP ofrece la puerta de enlace y la DNS.
- **DHCP** es la evolución de BootP y es el protocolo que se usa actualmente para la configuración automática de los equipos.

- **ICMP = Internet Control Message Protocol**
  - Es un protocolo para el intercambio de mensajes de control en la red.
  - Los mensajes ICMP se pueden clasificar en dos tipos:
- **Mensajes de error**
  - Permiten informar de situaciones de error en la red
    - Ejemplos: destino inalcanzable, tiempo excedido, problema de parámetro, etc.
- **Mensajes informativos**
  - Permiten intercambiar información sobre la presencia o el estado de un determinado sistema
    - Ejemplos: mensajes de ECHO (ping), anuncio o solicitud de router, redirecciones, etc.

# Formato de mensajes ICMP

- Los mensajes ICMP se transmiten dentro de paquetes IP
- El formato de los mensajes ICMP es el siguiente:



- La cabecera ICMP contiene la siguiente información:
  - **Tipo** (8 bits): Indica el tipo del mensaje ICMP
  - **Código** (8 bits): Ofrece información adicional sobre el contenido del mensaje. Su significado depende del tipo del mensaje.
  - **Checksum** (16 bits): Es un campo para detectar errores en el mensaje ICMP.

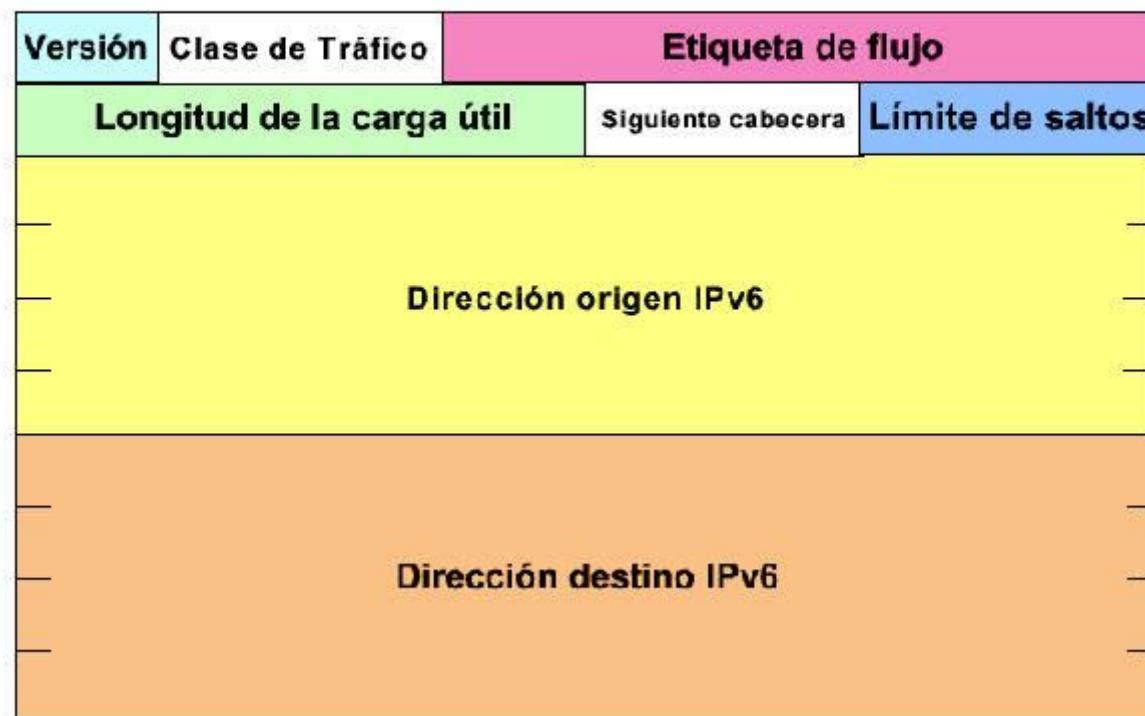
	<b>Tipo</b>	<b>Significado</b>
Mensajes Informativos	0	Echo Reply
	5	Redirect
	8	Echo Request
	9	Router Solicitation
	10	Router Advertisement
Mensajes de error	3	Destination Unreachable
	4	Source Quench
	11	Time Exceeded
	12	Parameter Problem

**Mensajes ICMP Echo Request y Echo Reply** Se utilizan para ver si un computador es alcanzable  
Para generar este tipo de paquetes se utiliza la orden **ping**  
Formato de los mensajes Echo Request/Echo Reply

Tipo (0/8)	Código (0)	Checksum
Identificador	Nº de secuencia	
Datos		

- **Tipo** = 8 Echo Request; 0 Echo Reply
- **Código** = 0
- **Identificador.** Permite establecer la correspondencia entre mensajes de Echo Request y Echo Reply.
  - Cada mensaje Echo Reply contiene el mismo identificador que su correspondiente Echo Request
- **Secuencia.** También se utiliza para establecer la correspondencia entre el Echo Request y el Echo Reply, cuando se envían varios Echo Requests consecutivos con el mismo identificador.
  - El mensaje Echo Reply contiene el mismo nº de secuencia que su correspondiente Echo Request
- **Datos.** Contiene un número determinado de bytes, generados aleatoriamente por la herramienta de diagnóstico.
  - El tamaño de este campo se puede especificar como un parámetro de la orden **ping**

## Cabecera de paquete IPv6



Surgió debido al agotamiento de direcciones IP de la versión anterior. En 1992 se solicitaron propuestas para IPng (nueva generación) que pasó a llamarse IPv6 en los RFC 2460 y 2373. Sus características son:

- **1 Direcciones de 128 bits** (16 octetos) = 670 mil billones direcciones/mm<sup>2</sup> de la superficie de La Tierra
- 2 Mayor flexibilidad de direccionamiento:

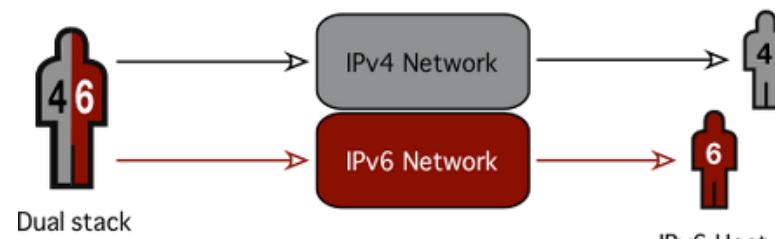
Dirección “**anycast**” o de “difusión por proximidad”: dirigido al equipo más próximo de un conjunto (todos tienen la misma IP)

Direcciones Multicast con campo de ámbito.

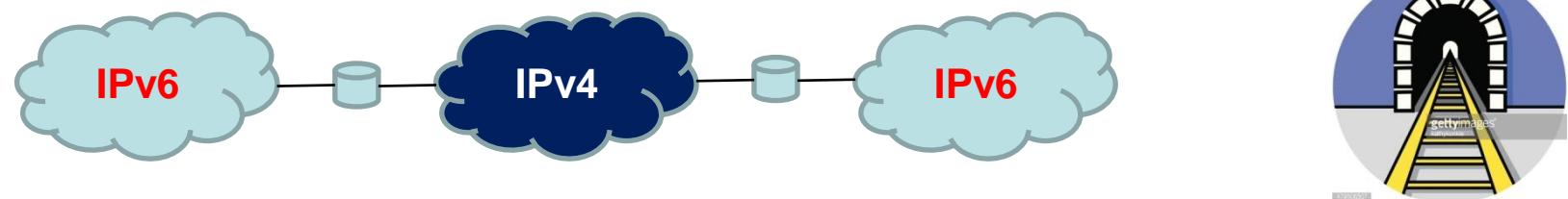
- 3 Etiquetado de “**flujo de datos**” para mejor tratamiento de voz y vídeo.
- 4 **QoS** asociados a los flujos de datos.
- 5 **Cabecera simplificada** con respecto a la versión 4.
- 6 Las opciones se tratan como **cabeceras insertables** antes de los datos.
- 7 **Privacidad** de datos.

Evolución: Hoy día, los principales sistemas operativos (Windows Vista, Linux, MacOs) incorporan IPv6. Los servidores raíz DNS ya proveen registros con IPv6. Las IPv4 públicas se agotaron en 2013. Tan sólo queda que las grandes operadoras migren sus equipos progresivamente.

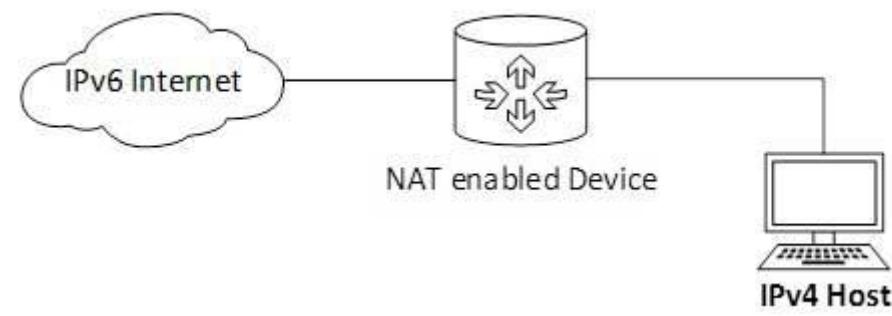
- **Doble pila o “dual-stack”:** IPv4 e IPv6 coexisten en la misma red. Los equipos tienen simultáneamente asignadas direcciones IPv4 e IPv6 y procesan los paquetes según su tipo.



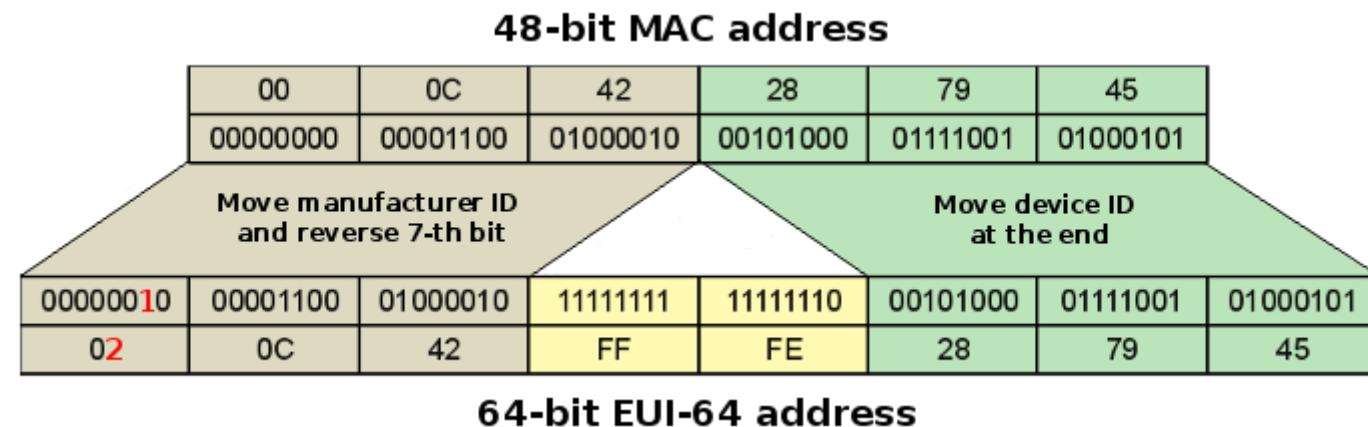
- **Tunneling:** El paquete IPv6, que viene de una red o “isla” IPv6, se encapsula dentro de un paquete IPv4, para atravesar la red IPv4 hasta llegar a la otra “isla” IPv6.



- **Traducción:** La traducción de direcciones de red 64 (NAT64) permite que los dispositivos IPv6 se comuniquen con dispositivos IPv4 mediante una técnica similar al NAT para IPv4. Un paquete IPv6 se traduce en un paquete IPv4, y viceversa.



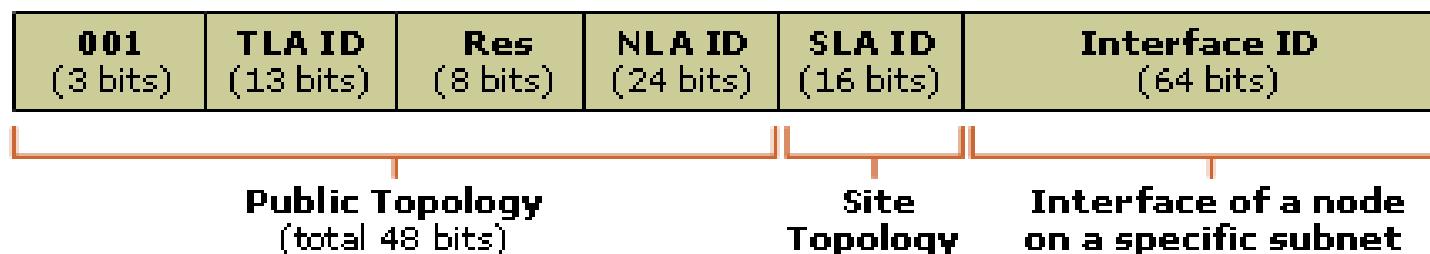
- **Link-local:** Direcciones que sólo tienen sentido dentro de la misma subred local.
  - FE80::/10
  - Normalmente se forman con la **MAC**, pero el administrador puede elegir **cualquiera**.



- Ejemplo: FE80::020C:42FF:FE28:7945, o FE80::1
- Existe un protocolo (DAD) para la detección de direcciones duplicadas
- *Equivale a 0.0.h.h o 169.254.h.h en IPv4*

- **Global:** Direcciones visibles en Internet.

- 2000::/3



- 1. Prefijo de enrutamiento global(48bits). Lo dan IANA+ISP
  - 2. Prefijo de subred (16bits). Lo elige el usuario.
  - 3. Identificador de NIC. Se forman con la **MAC**, o a **voluntad**.
- *Equivale a direcciones públicas en IPv4*

- Loopback: 0::1/128 (Equivale a 127.0.0.1 de IPv4)
- Dirección ausente: 0::0/128
- Dirección **privada** (“unique local”): fc00::/7
  - Equivale a 10.x.x.x, 172.16-31.x.x, 192.168.x.x de IPv4
- **Multicast:** ff::/8
  - ff02::1/128 Broadcast (Equivale a r.r.255.255 de IPv4)
  - ff02::2/128 Routers (Equivale a 224.0.0.2 de IPv4)
  - ff02::[id.de NIC|últimos 24 bits id. de grupo] /128
- *Direcciones “de transición IPv4”*
  - 0::1.2.3.4/96 “compatibles”.
  - 0::ff:ff:1.2.3.4/96 “mapeadas”.

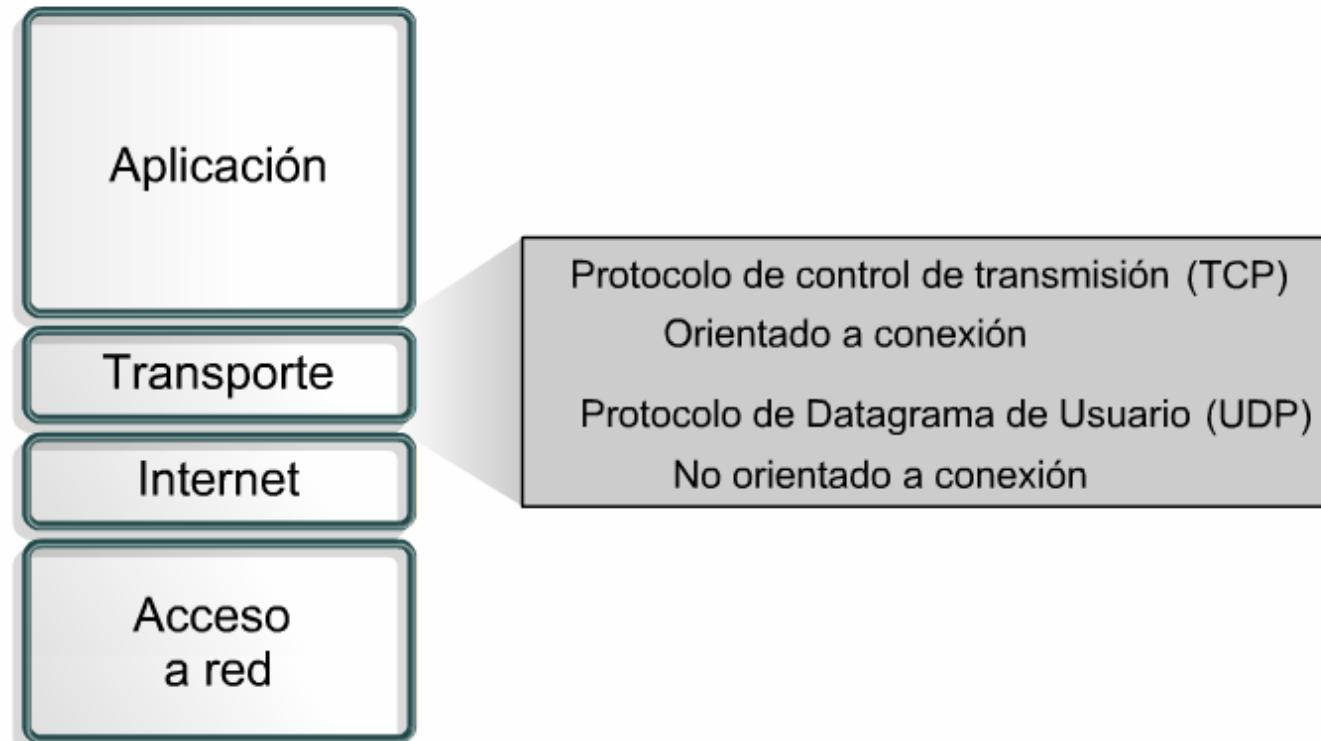
## ¿Cómo se otorgan las direcciones?

- Configurándolas manualmente.
- Usando **DHCPv6**
- Usando paquetes ICMPv6 con SLAAC (stateless address autoconf).
- Mixta de las dos últimas (**dirección vía SLAAC** y **DNS vía DHCP**)

Regla no escrita: a los routers se les suele dar la dirección 1.

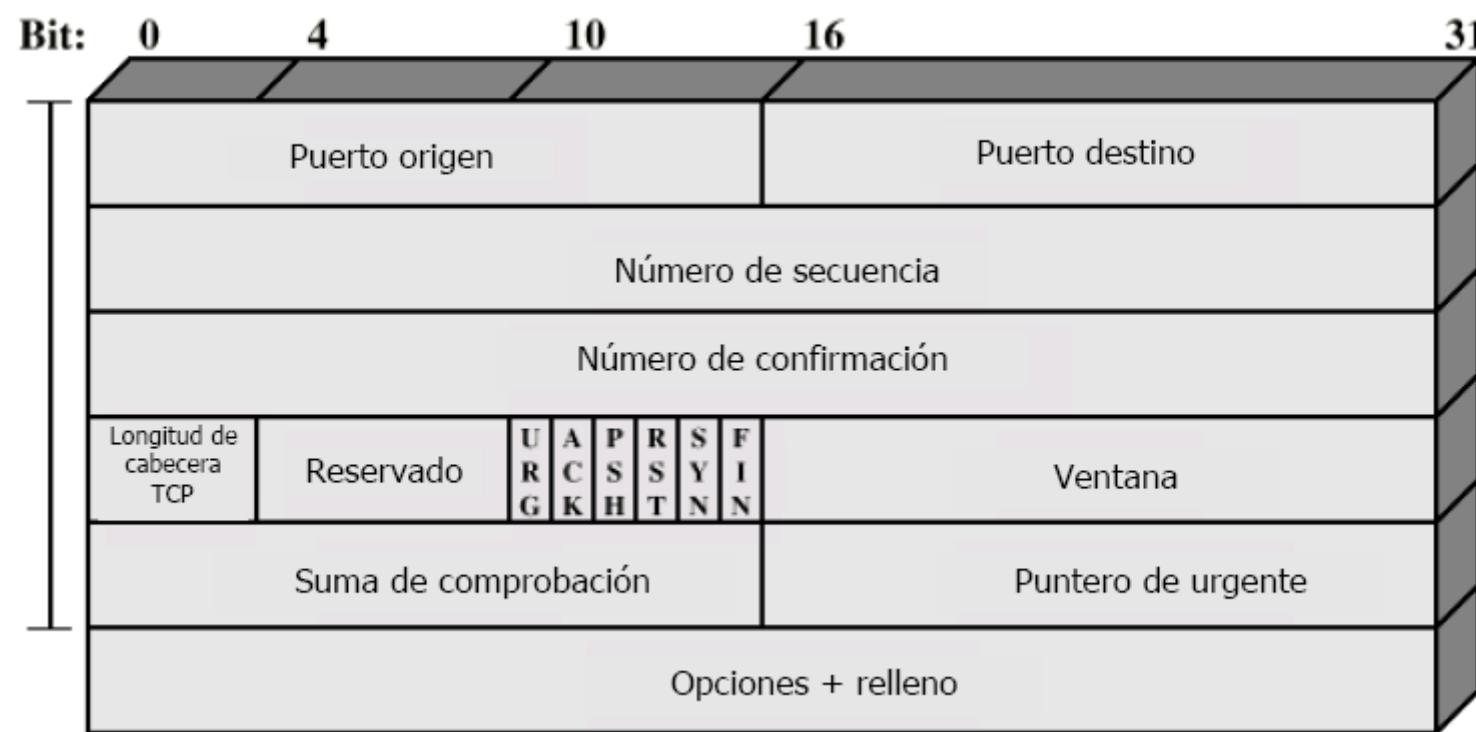
En ipv6, la primera y la última IP de una red **sí** se pueden usar.

En IPv6 también se puede hacer **subnetting**, pero no suele hacerse dado que la máscara /64 da para redes y hosts de sobra.



- **El protocolo TCP (Transmission Control Protocol)**
  - Protocolo de transporte "orientado a conexión", es decir, garantiza un servicio extremo a extremo fiable:
    - Detectar/retransmitir segmentos de datos perdidos o erróneos
    - Detectar y descartar segmentos duplicados
    - Ordenar los segmentos en el destino y pasarlo de forma ordenada a la capa de aplicación
  - Utilizado en aplicaciones en las que la seguridad en la entrega es más importante que la rapidez
  - Soporta: FTP, HTTP, Telnet, SMTP, DNS, etc.
- **El protocolo UDP (User Datagram Protocol)**
  - Protocolo de transporte "sin conexión": no garantiza un servicio extremo a extremo fiable
  - Utilizado en aplicaciones en las que la rapidez en la entrega es más importante que la seguridad, o bien, donde la seguridad está garantizada por ser local.
  - Soporta: DNS, SNMP, RIP, etc.

- Unidad de transferencia: Segmento
- Fases en una transmisión mediante TCP
  - Establecimiento de conexión (bits SYN, ACK)
  - Transferencia de datos
  - Cierre de conexión (bits FIN, ACK; o RST)
- Mecanismos de control de errores en TCP
  - Cada segmento lleva un número de secuencia de 32 bits (NS) que indica la posición que ocupa el primer byte del segmento dentro del mensaje original
  - Cuando el receptor recibe un segmento de datos correcto y sin errores, envía una confirmación (NR) al emisor
  - Si transcurrido un tiempo ( $T_{out}$ ) desde que se envió el segmento, el emisor no recibe confirmación, entonces retransmite de nuevo el segmento



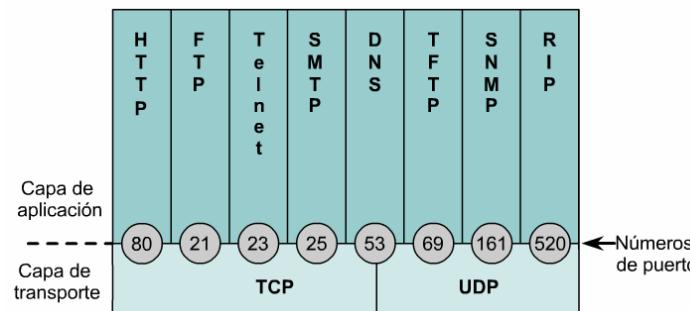
- **Campos de la cabecera del segmento TCP**
- **Puerto origen y destino:**
  - Identifican los puertos extremos de la conexión
- **Nº de secuencia:**
  - Indica la posición del primer byte del segmento con respecto al mensaje original (stream)
- **Nº de confirmación:**
  - Indica el nº de secuencia del siguiente byte que se espera recibir, confirmando los anteriores.
- **Longitud de la cabecera:** medida en palabras de 32 bits
- **Flag URG y puntero urgente:**
  - Si URG=1, el segmento transporta datos urgentes a partir del nº de byte especificado en el campo **puntero urgente**. **Estos datos no pueden esperar en el buffer de recepción hasta que la aplicación los demande, deben ser entregados a ella inmediatamente.**
- **Flag ACK:**
  - Si ACK=1, el segmento transporta una confirmación válida en el campo de superposición

- **Campos de la cabecera del segmento TCP**
- **Flag PUSH:**
  - Si PUSH=1, indica que los datos deben ser enviados inmediatamente.
  - Si PUSH=0, los datos se pueden almacenar en un buffer de transmisión y ser enviados cuando se tengan datos suficientes para enviar un segmento con tamaño igual a MSS.
- **Flag SYN:**
  - Flag utilizado en el establecimiento de la conexión.
  - Significa que los extremos deben sincronizar los números de secuencia iniciales de la transmisión
- **Flag RST:**
  - Flag utilizado para abortar una conexión
- **Flag FIN:**
  - Flag utilizado en la finalización de la conexión
- **Código de redundancia:**
  - Checksum de todas las palabras de 16 bits de la parte de datos
- **Ventana:**
  - Permite negociar dinámicamente el tamaño de la ventana de transmisión
- **Opciones:**
  - Permite negociar parámetros adicionales de la conexión, por ejemplo el tamaño máximo del segmento (MSS)

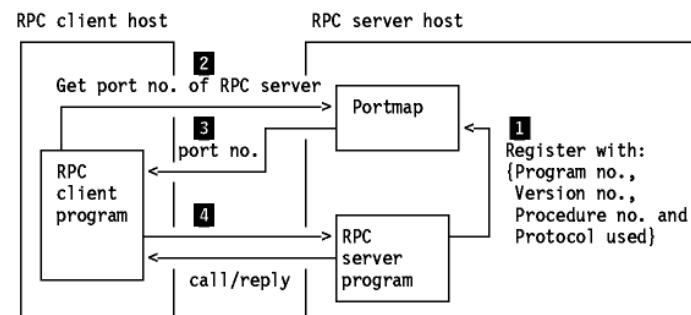
- **El cliente:** Es la aplicación que solicita la conexión con la máquina remota (se activa sólo cuando se necesita).
- **El servidor:** Es la aplicación que recibe y acepta la solicitud de conexión del cliente (tiene que estar siempre activa y escuchando).
- **“Sockets” en una conexión TCP**
  - Cliente:
    - **Dirección IP:**
      - Identifica a la máquina cliente (la que solicita la conexión)
    - **Puerto:**
      - Identifica al proceso cliente dentro de la máquina cliente (ej. el navegador Mozilla)
  - Servidor:
    - **Dirección IP :**
      - Identifica a la máquina servidora (la que acepta la conexión)
    - **Puerto:**
      - Identifica al proceso servidor dentro de la máquina servidora (ej. servidor web Apache)
- **El comando netstat muestra los sockets conectados: C:/netstat -an**

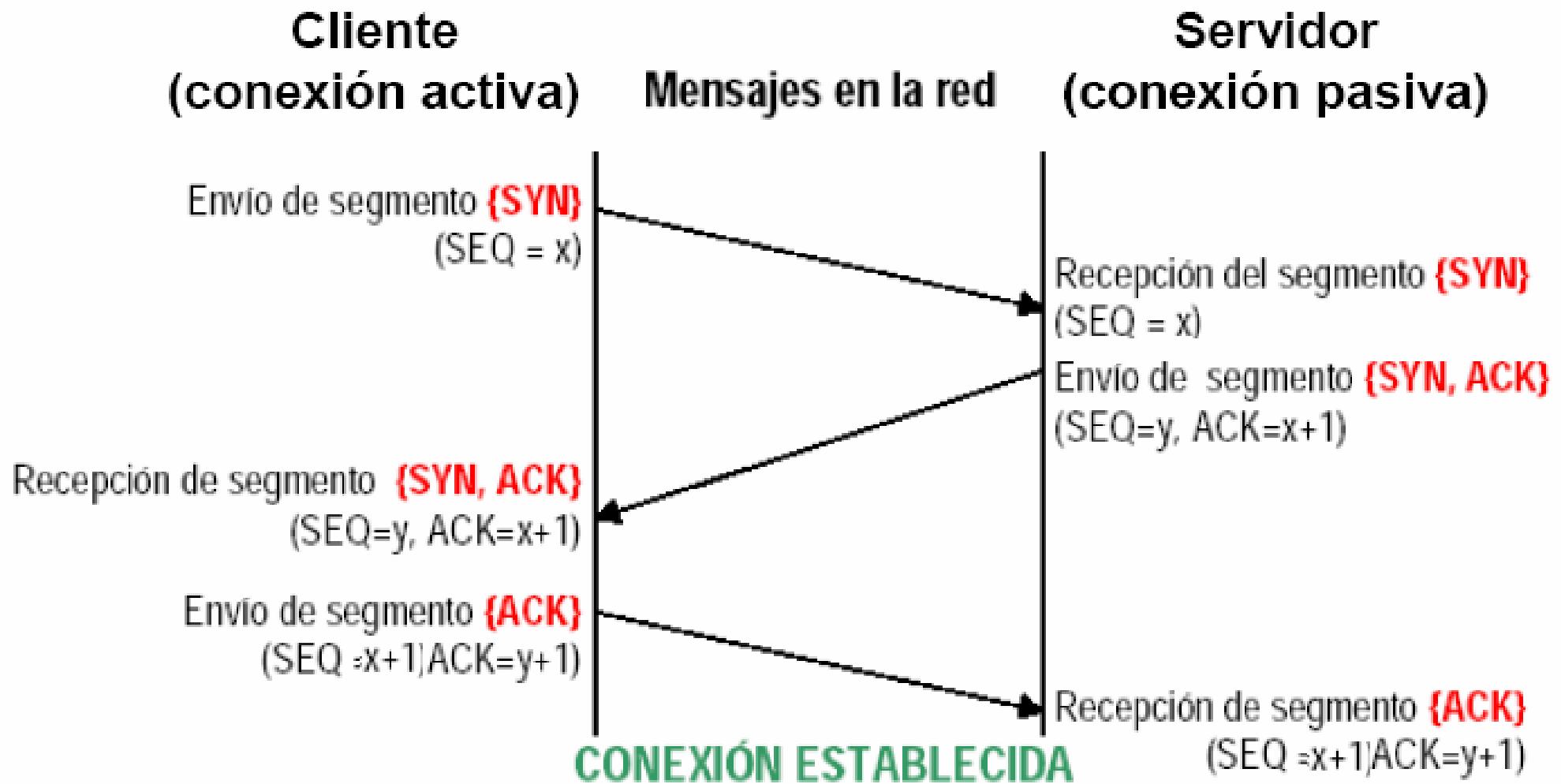
- **El puerto del cliente**
  - Cuando un proceso cliente solicita una conexión TCP, el SO de la máquina cliente le asigna un número de puerto libre. Este número de puerto es siempre un número mayor que 1024.
- **El puerto del servidor**
  - El cliente debe conocer de antemano el número de puerto del servidor para establecer la conexión. Para ello existen dos soluciones:
    - Que el servidor utilice un "puerto bien conocido" (wellknown ports) o pre-asignado. Los números inferiores a 1024 corresponden a números de puerto bien conocidos.

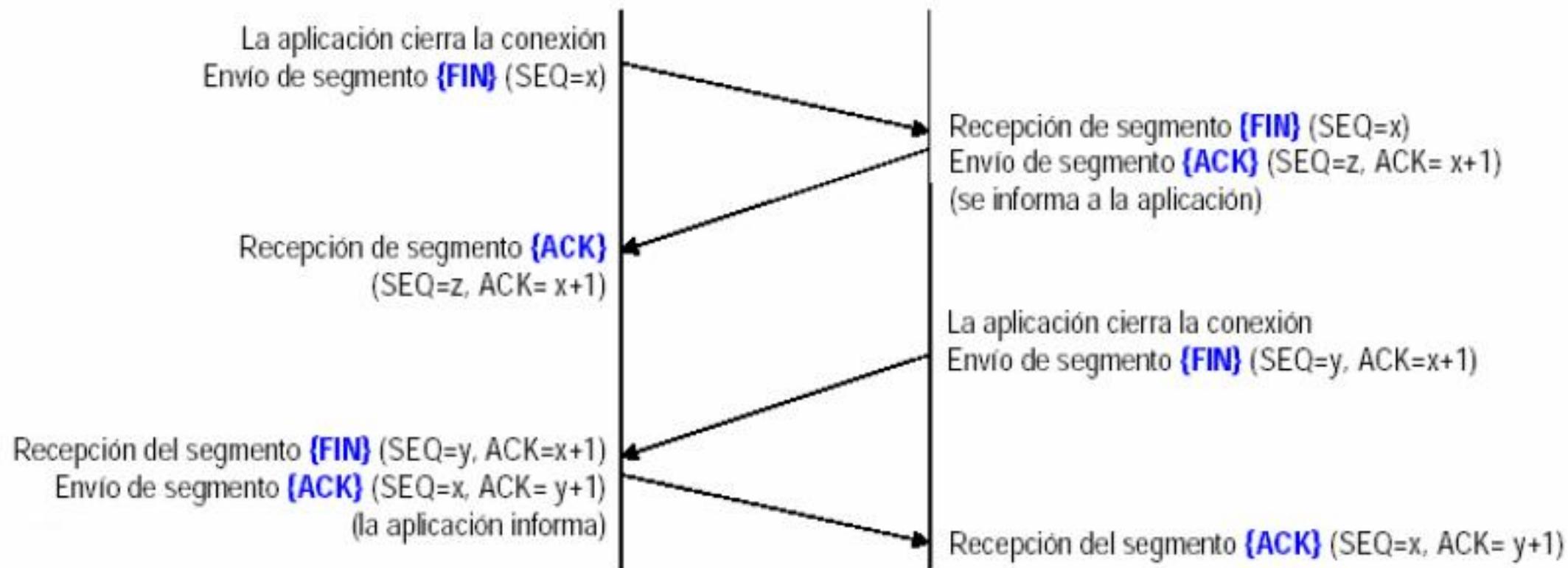
FTP: 21  
SSH: 22  
TELNET: 23  
SMTP: 25  
HTTP: 80  
POP3: 110

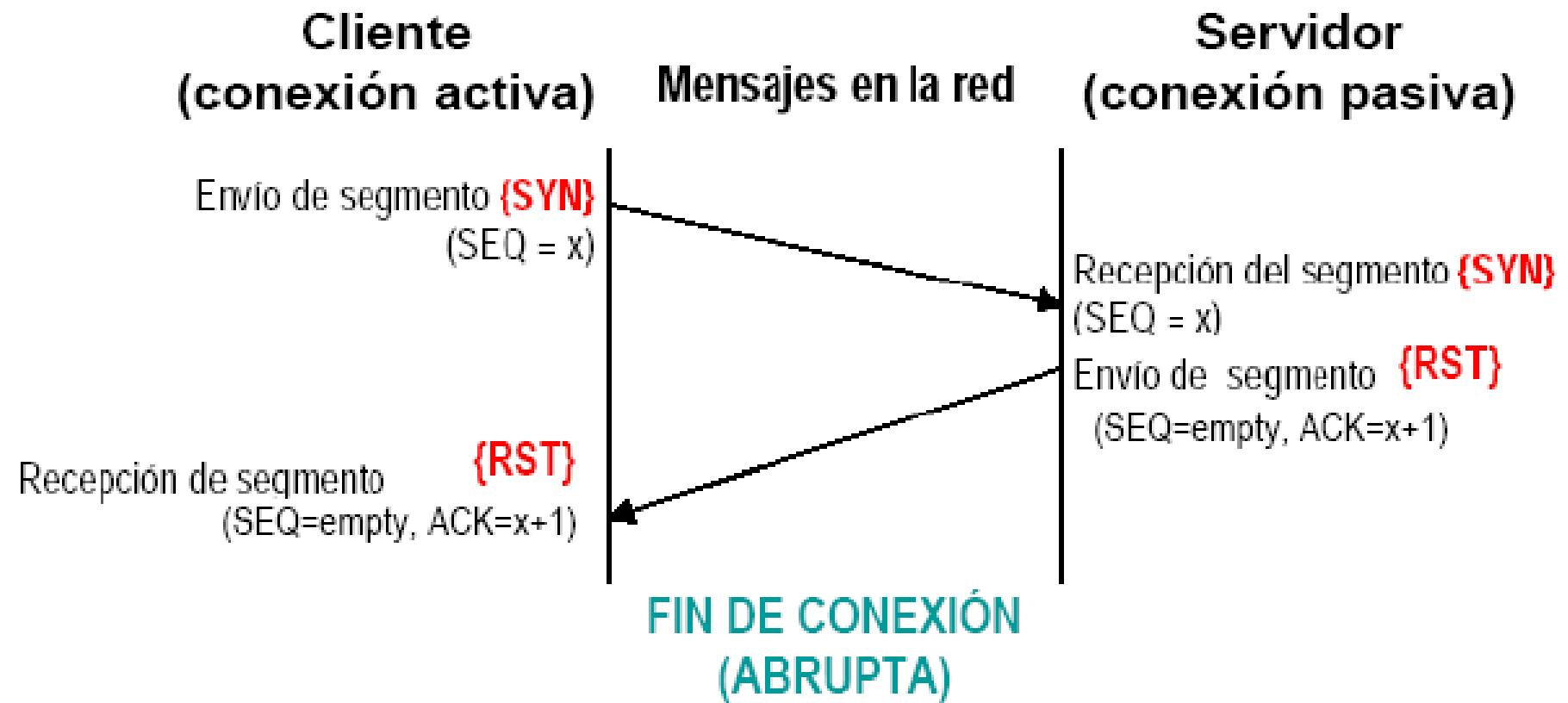


- Usar el servicio de llamada a procedimiento remoto RPC (Remote Procedure Call).









RST se puede enviar en cualquier momento desde cualquier lado (cliente o servidor) y no espera respuesta.  
RST suele usarse cuando un servidor deniega el servicio (caso de la ilustración).

- **Características de UDP**

UDP NO es fiable:

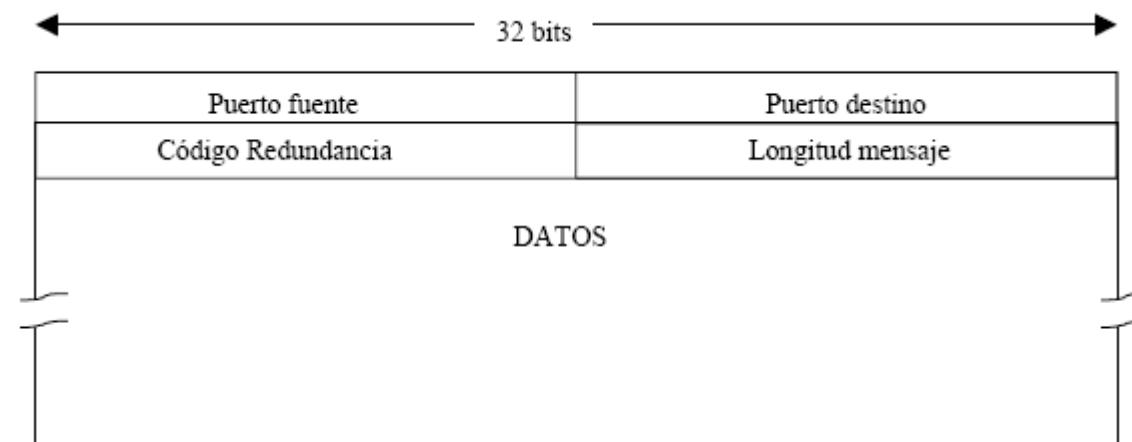
- UDP NO numera los segmentos.
- El receptor NO envía confirmación de la recepción de los mismos
- UDP no garantiza recuperación de paquetes perdidos o erróneos ni evita la duplicidad de paquete

UDP proporciona puertos y segmentación, pero nada más.

UDP es más rápido (no requiere tanto procesamiento) y reduce la carga de tráfico  
(cabeceras más pequeñas)

Ideal para “streaming”, es decir, flujos de video o voz donde la pérdida puntual de segmentos es menos relevante que la latencia

- **Formato del paquete UDP**

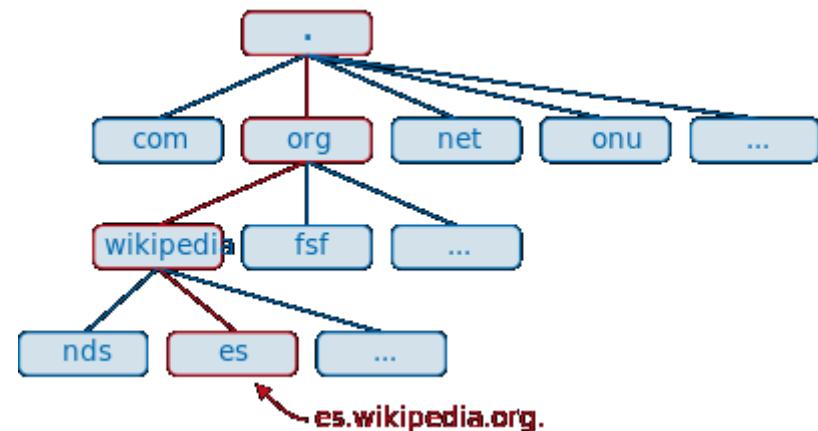
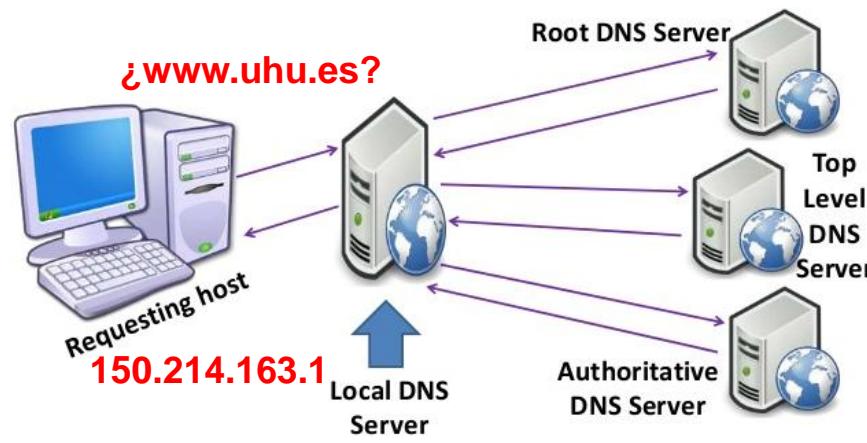


- Telnet (23TCP)
  - Servicio de terminal virtual.
  - No cifrado, hoy día se usa **SSH(22)**
  - Permite negociación de opciones.
- TFTP(UDP69)
  - Implementa control de flujo.
- FTP(TCP20-21)
  - Usa un puerto del servidor para control (21) y otro para datos (20).
  - Permite explorar directorios, transferencia múltiple, etc...
- SMTP(T25Y587)
  - SMTP es el protocolo utilizado para envío de correo desde la aplicación cliente o entre servidores.
- POP3(T110Y995)
  - POP3 es el protocolo utilizado para descargar correo desde los servidores.
- DNS(UDP53,TCP53)
  - Servicio de Resolución de Nombres
- HTTP(T80)
  - Protocolo de Transferencia de Hypertexto (HTML: Hypertext Markup Language).
  - Permite transferir y visualizar los contenidos de las páginas web desde la aplicación Navegador
- SNMP (TCP161-162)
  - Monitorización de la red



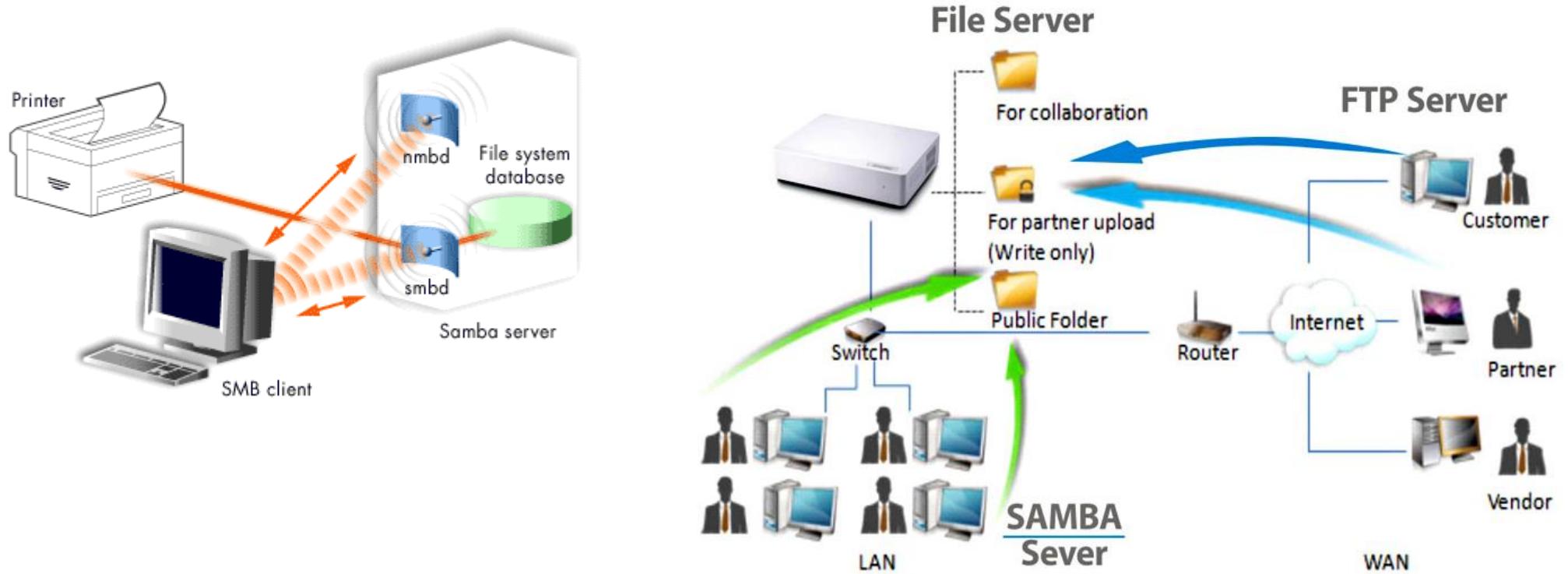
- Transferencia de archivos**
    - TFTP ♦
    - FTP ♦
    - NFS
  - Correo electrónico**
    - SMTP
  - Conexión remota**
    - Telnet ♦
    - rlogin
  - Administración de red**
    - SNMP ♦
  - Gestión de nombres**
    - DNS ♦
- ♦ utilizado por el router

# DNS



- Dado un nombre ([www.uhu.es](#)) te devuelven su IP. Comando **nslookup nombre**.
- La IP de al menos un DNS debe configurarse en los PCs (nuestro ISP nos la debe dar).
- Los DNS almacenan los registros de nombres de forma jerárquica (búsqueda recursiva). Además, mantienen unos registros temporales en función de las peticiones que van recibiendo (búsqueda en la caché).
- Los registros están normalizados y pueden ser:
  - A o AAAA: nombre de dominio – IP, el tipo de registro más común.
  - CNAME: un alias de ese mismo nombre de dominio
  - MX: servidor de correo
  - NS y SOA: información DNS de la zona.
  - PTR: IP- nombres de dominios asociados a esa IP.
- El acceso al registro de nombres es parecido a las IPs, parte del ICANN y de forma jerárquica se va concediendo a las diferentes empresas (ISPs).

# FTP y SAMBA



- Un servidor FTP sirve para subir/bajar archivos. Comando **ftp IP**
  - Una vez logueado (anonymous si no hay clave) : **put**, **get**, **dir** para subir, bajar, listar archivos
- El protocolo FTP es la forma de comunicar cliente y servidor.
- SAMBA es la implementación libre del protocolo SMB que windows utiliza para compartir archivos en red.
- Con SAMBA, al igual que a archivos se pueden acceder a otros recursos (impresora, fax).