DISEÑO Y ADMINISTRACIÓN DE REDES

Área de Ingeniería Telemática

Grado en Ingeniería Informática
Cuarto curso. Primer Semestre.

Departamento de
Ingeniería Electrónica
y Comunicaciones
Universidad Zaragoza

Bloque 1. Interconexión de redes IPv4

Repaso Protocolo Internet (IPv4)
NAT: Network Address Translation
Protocolos de encaminamiento. RIP y OSPF.
Funciones de control. Apoyo en otros protocolos.
Gestión de redes TCP/IP: arquitectura SNMP.

Área de Ingeniería Telemática

Grado en Ingeniería Informática
Cuarto curso. Primer Semestre.

Departamento de Ingeniería Electrónica y Comunicaciones
Universidad Zaragoza

Contenidos

- Repaso Protocolo Internet (IPv4) *Kurose, 4.4, págs.: 323-342, 349-351.*
  - Internet
  - Direccionamiento
  - Encaminamiento
  - Funcionalidad del Protocolo IPv4.
    -PDU -Fragmentación y reensamblado.
  - NAT y NAPT.
- Protocolos de Encaminamiento *Kurose, 4.6, págs.: 371-377.*
  - RIP y OSPF
- Funciones de control: apoyo en otros protocolos
  ICMP: Kurose, 4.4.3, págs.: 343-345.; ARP: Kurose, 5.4.2, págs.: 445-450.; DHCP: Kurose, 4.4.2, págs.: 336-339.
- Gestión de redes TCP/IP: arquitectura SNMP

  Kurose, Capítulo 9, págs.: 735 - 757.

Internet

- **Internet** es un conjunto mundial de redes interconectadas con **protocolos comunes** (TCP/IP) y un **direccionamiento universal** (IP).

- Cada red se incorpora voluntariamente a Internet, y se gestiona de manera autónoma. Sin embargo, existen cierta organización:

  - ISOC (*Internet Society*), asociación internacional para la promoción de la tecnología y servicios Internet (http://www.isoc.org/isoc/)
    - IAB (*Internet Architecture Board*), consejo para el desarrollo técnico de Internet (supervisor de ISOC) (http://www.iab.org/)
    - IETF (*Internet Engineering Task Force*): desarrollo y promoción de estándars (http://www.ietf.org/)
      - RFCs (*Request for Comments*): documentos → http://tools.ietf.org/html
      - IESG (*Internet Engineering Steering Group*) (gestión)
    - IRTF (*Internet Research Task Force*): investigación (*long-term*) (http://www.irtf.org/)
      - IRSG (*Internet Research Steering Group*) (gestión)

  - IANA (*Internet Assigned Numbers Authority*) → ICANN (*Internet Corporation of Assigned Names and Numbers*) (http://www.icann.org/about/):
    - asignación de direcciones, dominios (gestión de servidores raíz)…
    - Delegación por zonas: RIR (*Regional Internet Register*)
      - ARIN, RIPE NCC, APNIC, LACNIC, AfriNIC

Direccionamiento IPv4

- Direccionamiento
  - Asignación de direcciones: Direcciones IPv4 agotadas!
  - *CLASSFUL*
  - *CLASSLESS*:
    - *Subnetting*
    - *Supernetting* (CIDR)
  - NAT (*Network Address Translation*)
  - Resolución de nombres: servicio DNS

- Resolución/Configuración
  - ARP (Proxy ARP, ARP gratuito…)
  - Autoconfiguración: DHCP

Direccionamiento IPv4

- Asignación de direcciones: **IANA** (ahora **ICANN**)
  - Inicialmente la asignación de direcciones IP la realizaba el **NIC (*Network Information Center*)** de forma centralizada.
  - A principios de los 90 se decidió descentralizar esta función creando los llamados **RIR (*Regional Internet Registry*)**.
  - Por crecimiento: delegan en **LIRs (*Local Internet Registry*)**

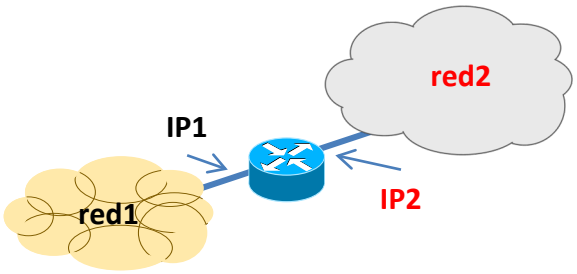
Registro Regional	Área geográfica
ARIN (*American Registry for Internet Numbers*) www.arin.net	EEUU y Canadá
APNIC (*Asia Pacific Network Information Centre*) www.apnic.net	Asia oriental, Pacífico
RIPE (*Réseaux IP Européenes*) www.ripe.net	Europa, Oriente Medio, Asia Central
LACNIC (*Latin American and Caribbean Network Information Center*) www.lacnic.net	América y el Caribe (excepto EEUU y Canadá)
AFRINIC (*African Network Information Center*) www.afrinic.net	África, Oceano Indico

Direccionamiento IPv4

- Asignación de direcciones: **IANA** (ahora **ICANN**)
  - Inicialmente la asignación de direcciones IP la realizaba el **NIC (*Network Information Center*)** de forma centralizada.
  - A principios de los 90 se decidió descentralizar esta función creando los llamados **RIR (*Regional Internet Registry*)**.
  - Por crecimiento: delegan en **LIRs (*Local Internet Registry*)**
- Las direcciones IPv4 asignadas por ICANN se han agotado
  - Los RIR todavía disponen de bloques delegados. Están próximos a agotarse
    - http://www.elpais.com/articulo/tecnologia/IANA/entrega/ultimos/paquetes/direcciones/Internet/IPv4/elpeputec/20110203elpeputec_3/Tes
    - http://www.ripe.net/internet-coordination/ipv4-exhaustion
    - http://www.potaroo.net/tools/ipv4/
  - Hasta ahora… ⇒ soluciones para aprovechar las direcciones IPv4 (NAT, subnetting…)
  - A partir de ahora… ⇒ Protocolo IPv6

Direccionamiento IPv4

- **Identificadores VIRTUALES universales**:
  - Interpretado por el *software*.
  - Independiente del direccionamiento *hardware*.



- Identifican una **conexión de un nodo**
  - Un nodo tendrá tantas direcciones IP como interfaces de conexión a red

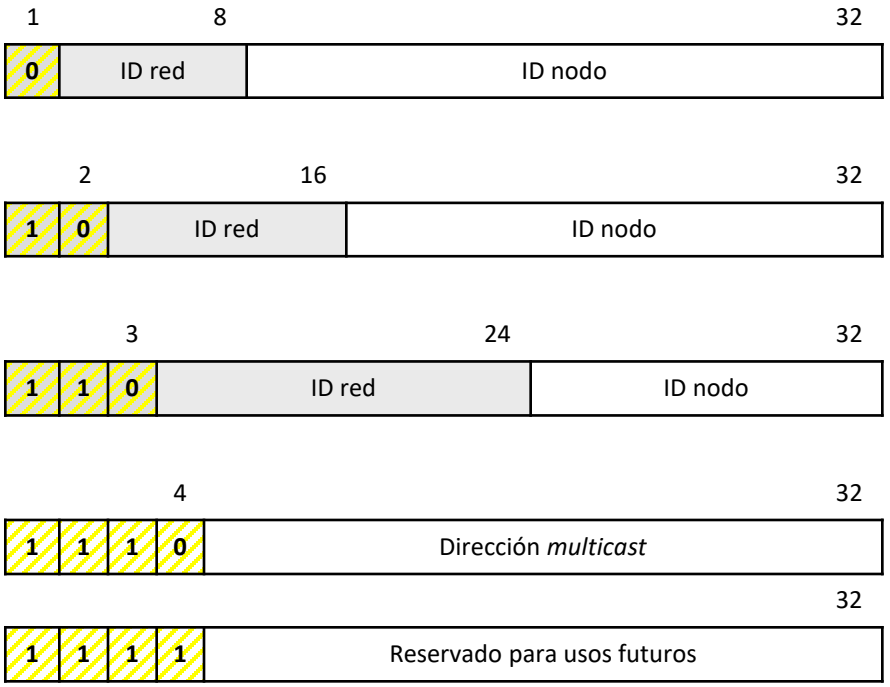
- 32 bits:
  - Notación decimal tomando cada 8 bits como un número decimal y separando los dígitos decimales por puntos

- Significado lógico:
  - *netid*: red
  - *hostid*: nodo en la red

	Identificador de red				Identificador de nodo			
	netid				*Hostid*			
@IP	10011011	11010010			00100110	11110001		
	155	. 210	.		38	. 241		
@red	10011011	11010010			00000000	00000000		
	155	. 210	.		0	. 0		
@bcast	10011011	11010010			11111111	11111111		
	155	. 210	.		255	. 255		

Direccionamiento IPv4

- ***CLASSFUL***: estructura de clases que define el tamaño de red y número de hosts posibles (*netid,hostid*)
  - **Clase A**
    - Pocas redes (126)
    - 16.777.214 nodos/red
  - **Clase B**
    - Redes medianas (16382)
    - 65532 nodos/red
  - **Clase C**
    - Muchas redes
    - 254 nodos/red
  - **Clase D** (multicast)
  - **Clase E** (reservada)



- Los algoritmos de **encaminamiento *CLASSFUL*** determinan la dirección de red en función de la **CLASE** (primeros bits)

Direccionamiento IPv4

- Direcciones especiales, reservadas y privadas

Red o rango	Uso
0.0.0.0	sin especificar (arranque)
000…000.*hostid*	uso en arranque
netid.000…000	@ red (*netid*)
netid.111…111	Difusión (todos nodos de *netid*)
255.255.255.255	Difusión limitada (arranque, red física)
127.X.Y.Z	*loopback* (uso en pruebas)
127.0.0.0	Reservado (fin clase A)
128.0.0.0	Reservado (inicio clase B)
191.255.0.0	Reservado (fin clase B)
192.0.0.0	Reservado (inicio clase C)
224.0.0.0	Reservado (inicio clase D)
240.0.0.0 – 255.255.255.254	Reservado (clase E)
10.0.0.0	Privado (clase A)
172.16.0.0 – 172.31.0.0	Privado (clase B)
192.168.0.0 – 192.168.255.0	Privado (clase C)

Direccionamiento IPv4

- Direcciones especiales, reservadas y privadas

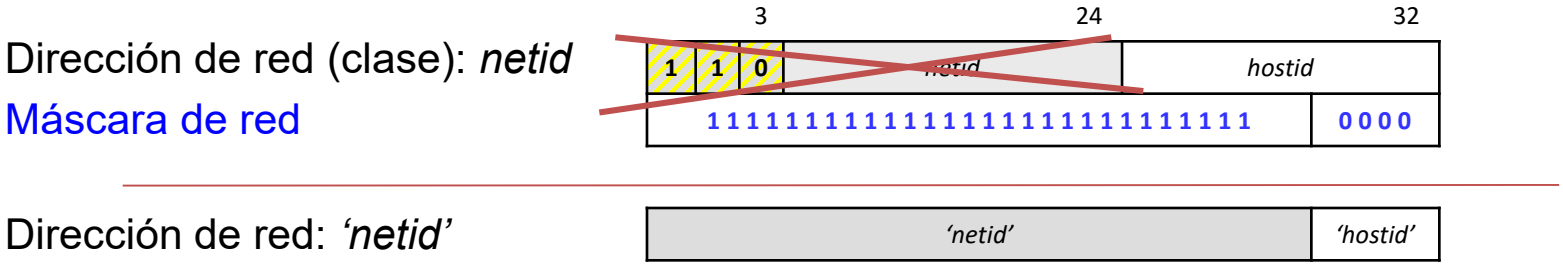
Red o rango	Uso
0.0.0.0	sin especificar (arranque)
000…000.*hostid*	uso en arranque
netid.000…000	@ red (*netid*)
netid.111…111	Difusión (todos nodos de *netid*)
255.255.255.255	Difusión limitada (arranque, red física)
127.X.Y.Z	*loopback* (uso en pruebas)
127.0.0.0	Reservado (fin clase A)
128.0.0.0	Reservado (inicio clase B)
191.255.0.0	Reservado (fin clase B)
192.0.0.0	Reservado (inicio clase C)
224.0.0.0	Reservado (inicio clase D)
240.0.0.0 – 255.255.255.254	Reservado (clase E)

Si no es estrictamente necesaria la CONECTIVIDAD GLOBAL (ej. Red LAN interna)
- no usar @s IP públicas ⇒ @s IP privadas
- Todos podemos usarlas, NO si "salimos" a internet → **NAT** (veremos…)

RFC 1918

Direccionamiento IPv4

- ***CLASSLESS***: **MÁSCARA** define el tamaño de red y número de hosts posibles (*'netid', 'hostid'*)



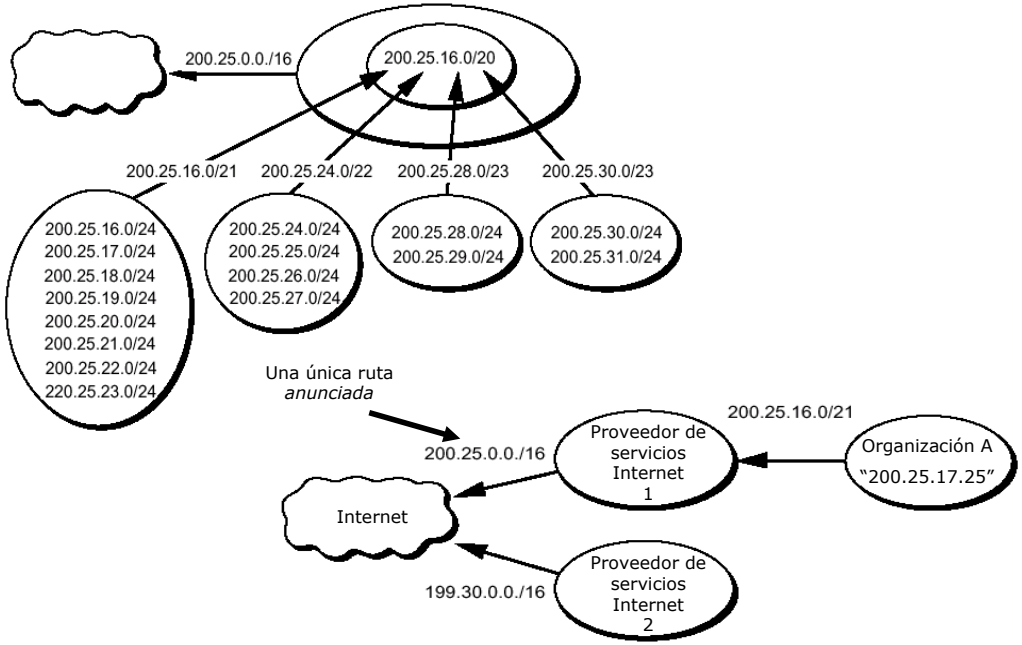
- Dirección de red:
  - *'netid'* > *netid* (clase): ***SUBNETTING*** (subred): subdividir una red en más redes → aprovechar el espacio de red
  - *'netid'* < *netid* (clase): ***SUPERNETTING*** (superred): redes consecutivas formen una red → reducir entradas encaminamiento (CIDR)
    - Agregar, como una única red, direcciones de red **CONSECUTIVAS**

- Los algoritmos de **encaminamiento *CLASSLESS*** usan la **MÁSCARA** (no la clase) para identificar la dirección de red

Internet funciona con CIDR (*Classless Inter Domain Routing*)

Direccionamiento IPv4

- CIDR (*Classless Interdomain Routing*)
  - direccionamiento/encaminamiento en Internet
  - Notación de red: **XXXXX / P** ⇒ **P: prefijo** (número de 1's de la máscara de red)
    - **Direcciones consecutivas** pueden agregarse con un prefijo común: reducción de entradas en la tabla de encaminamiento

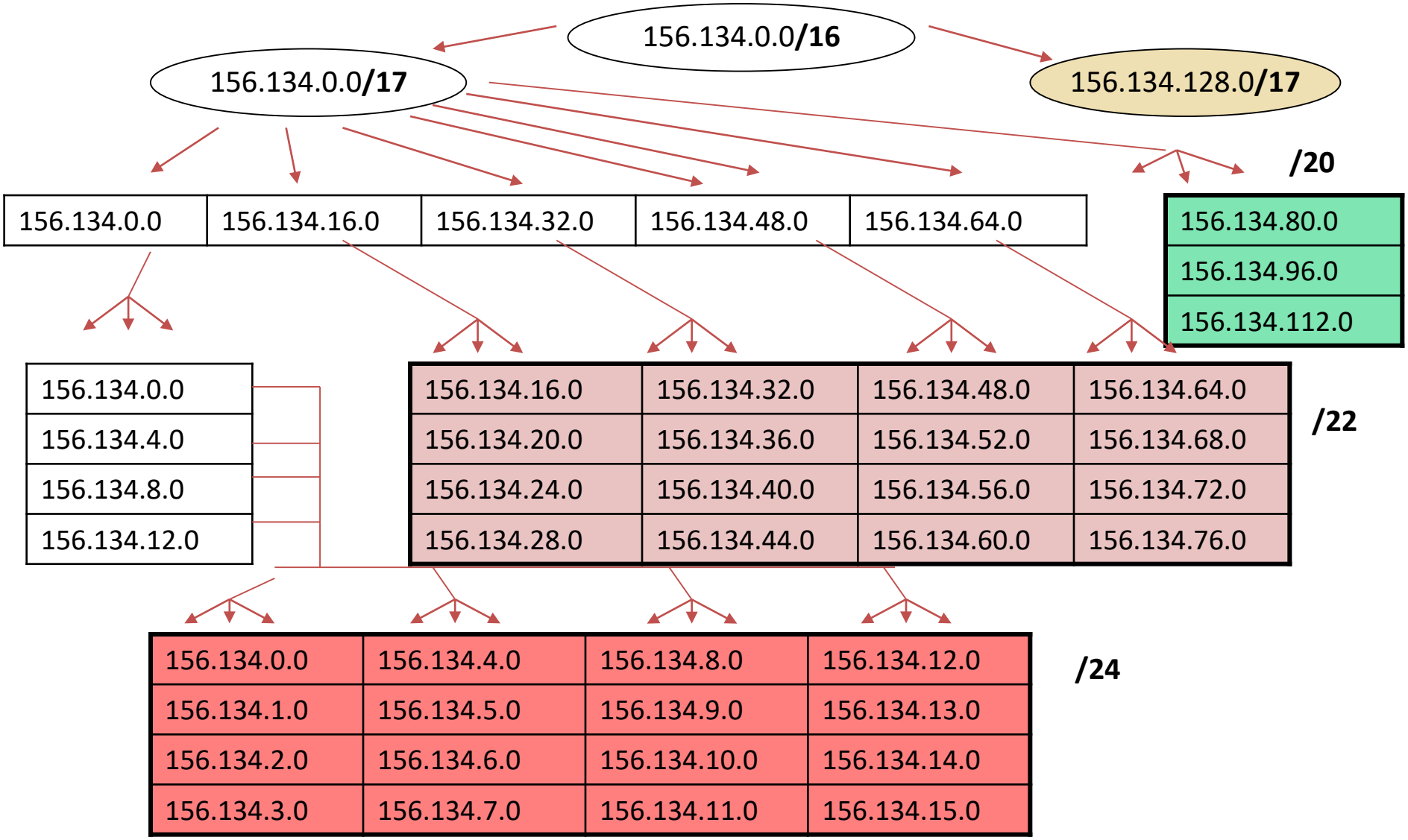


Prefijo	Notación Decimal	Direcciones Individuales	Redes IP
/13	255.248.0.0	512 K	8 Bs or 2048 Cs
/14	255.252.0.0	256 K	4 Bs or 1024 Cs
/15	255.254.0.0	128 K	2 Bs or 512 Cs
/16	255.255.0.0	64 K	1 B or 256 Cs
/17	255.255.128.0	32 K	128 Cs
/18	255.255.192.0	16 K	64 Cs
/19	255.255.224.0	8 K	32 Cs
/20	255.255.240.0	4 K	16 Cs
/21	255.255.248.0	2 K	8 Cs
/22	255.255.252.0	1 K	4 Cs
/23	255.255.254.0	512	2 Cs
/24	255.255.255.0	256	1 C
/25	255.255.255.128	128	1/2 C
/26	255.255.255.192	64	1/4 C
/27	255.255.255.224	32	1/8 C

Direccionamiento IPv4

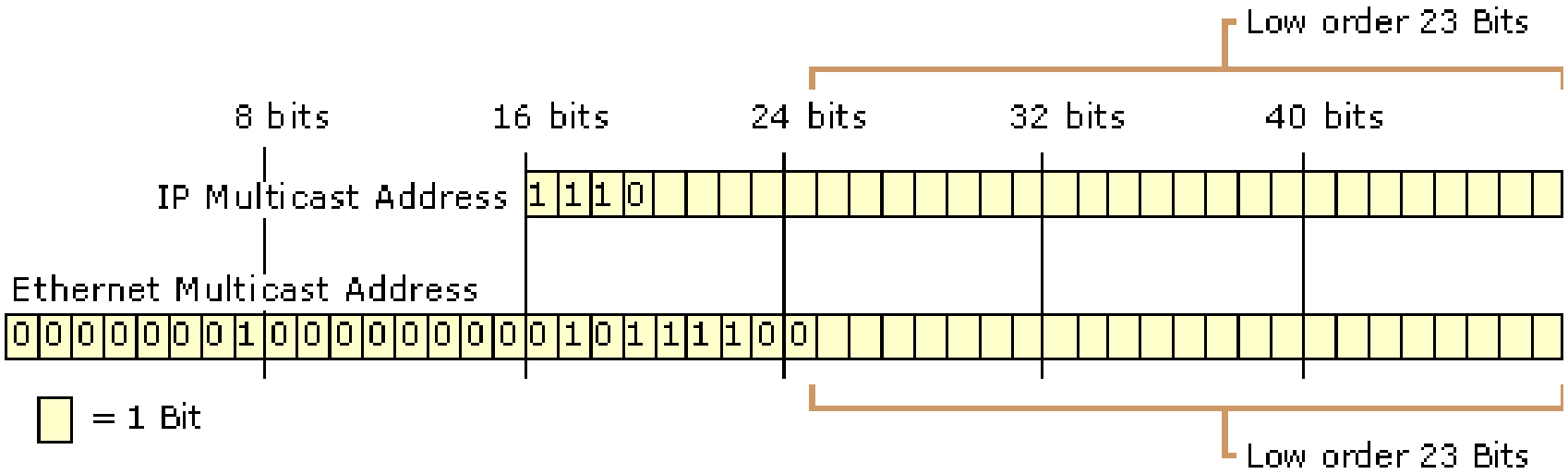
	***CLASSLESS*: MÁSCARA DE RED**	
	SUBRED	**SUPERRED**
Problema	@red se agotan (asignación de una de clase B por dpto. en la misma empresa)	Asignación de @red de pocos nodos (clase C): una empresa grande usa varias: explosión de tablas de rutas
Solución	**SUBDIVIDIR** con la MÁSCARA una red en varias subredes: estructura jerárquica	Asignar @red's **CONSECUTIVAS** y con MÁSCARA definir como una única entrada en la tabla de rutas
OBSERVACIONES	Notación habitual: **@red/prefijo**: **200.25.16.0/21** ⇒ máscara 255.255.248.0	
	Mayor eficiencia: dimensionar correctamente **MÁSCARAS DE TAMAÑO VARIABLE** ***VLSM*** *(Variable Length Subnet Mask)*	Internet: Encaminamiento CIDR: - Agrupación organizaciones - Agrupación proveedores - Agrupación países…
ENCAMINAMIENTO	Requiere ENCAMINAMIENTO *CLASSLESS* (VLSM)	Requiere ENCAMINAMIENTO *CLASSLESS*
	Un **conjunto** de **redes / subredes + 1 máscara** (prefijo) = **una única red** vista "desde fuera"	

Variable Length Subnet Mask (VLSM)



Mapping IP Multicast to MAC-Layer Multicast

To support IP multicasting, the Internet authorities have reserved the multicast address range of 01-00-5E-00-00-00 to 01-00-5E-7F-FF-FF for Ethernet and Fiber Distributed Data Interface (FDDI) media access control (MAC) addresses. As shown in Figure 4.1, the high order 25 bits of the 48-bit MAC address are fixed and the low order 23 bits are variable.



Contenidos

- Repaso Protocolo Internet (IPv4)
  - Internet
  - Direccionamiento
  - **Encaminamiento**
  - Funcionalidad del Protocolo IPv4.
    -PDU - Fragmentación y reensamblado.
  - NAT y NAPT.
- Protocolos de Encaminamiento
  - RIP y OSPF
- Funciones de control: apoyo en otros protocolos
- Gestión de redes TCP/IP: arquitectura SNMP

Encaminamiento

- Función: Definir el camino o ruta a seguir por los datagramas, a través de una o más redes, para que estos alcancen su destino.
  - **Encaminamiento directo**: llegar a "su propia red": Asociación @IP - @PHY
    - Resolución de dirección (ej. ARP)
  - **Encaminamiento indirecto**: llegar a "otra red": Siguiente salto (*router*)
    - Se traduce en encaminamiento directo hacia dicho siguiente salto
    - **Tabla de encaminamiento:**

Red destino	Máscara	Métrica	Siguiente salto	Interfaz de salida

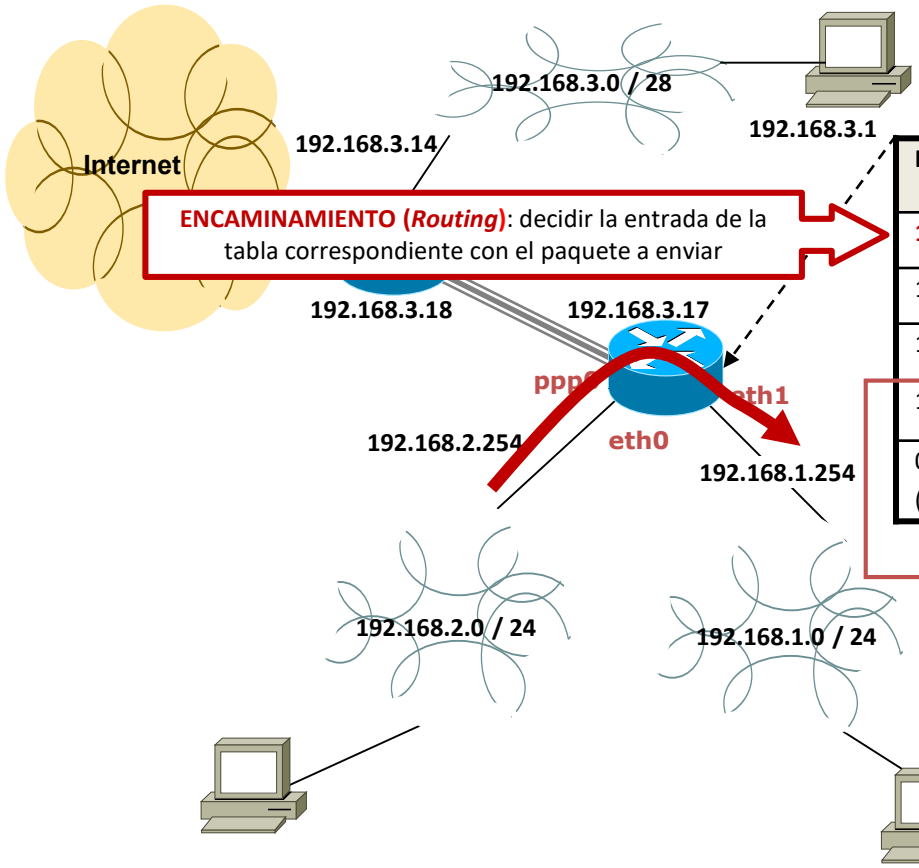
- ***Host*: Tabla: su dirección de red (directo) + *router* por defecto**
  - Si recibe un datagrama que no es para él, lo descarta
  - Si desea enviar un datagrama: a su propia red ⇒ ARP
  - Si desea enviar un datagrama: a otra red destino ⇒ siguiente salto: *router* "por defecto"

- ***Router*: Tabla: las redes a las que está conectado, redes externas + *router* por defecto**
  - Si recibe un datagrama que no es para él, intenta reenviarlo: consulta de la tabla de rutas
  - Si tiene que reenviar en una red propia ⇒ ARP
  - Si tiene que reenviar a una red externa ⇒ consultar la tabla: **comparativa @IP/máscara**
    - Existe una única entrada a la red destino: enviar
    - Existen varias entradas a la red destino: enviar a la de **métrica menor**
    - No existe la entrada explícita: enviar al *router* "por defecto"

Si hay coincidencia con varias, la de **máscara más larga** (más bits coincidentes)
→ ***long prefix match***

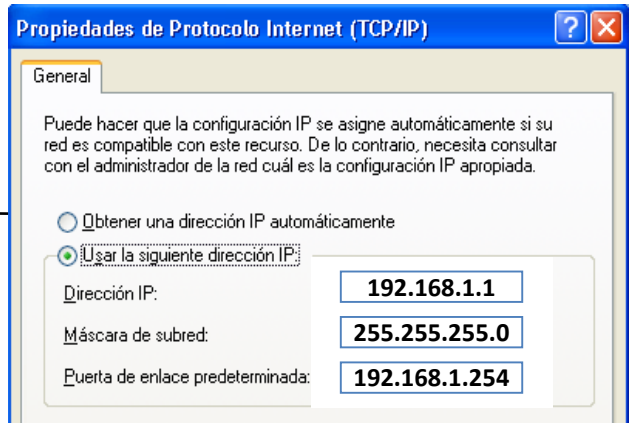
Encaminamiento

REENVÍO (*Forwarding*): Enviar el paquete de una interfaz (IN) a otra (OUT)



Red destino	Máscara	Métrica	Siguiente salto	Interfaz de salida
192.168.1.0	**255.255.255.0**	**0**	*	**eth1**
192.168.2.0	255.255.255.0	0	*	eth0
192.168.3.16	255.255.255.252	0	*	ppp0
192.168.3.0	255.255.255.240	0	192.168.3.18	ppp0
0.0.0.0 (default)	0.0.0.0	0	192.168.3.18	ppp0

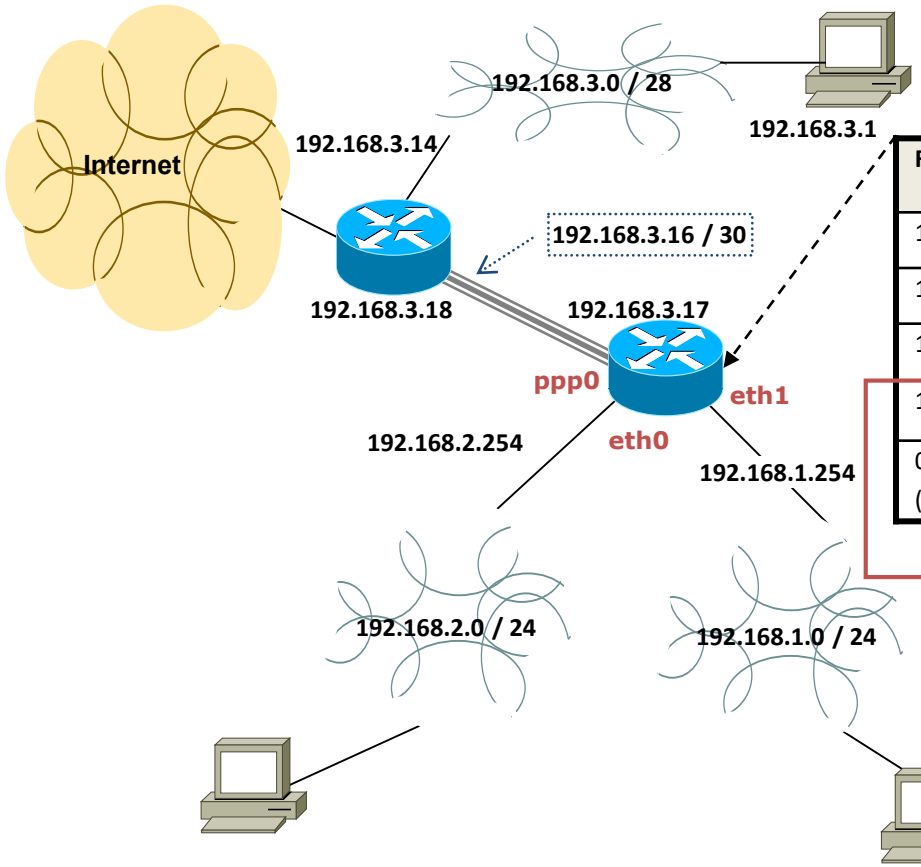
Valdría una única entrada (default)



Encaminamiento

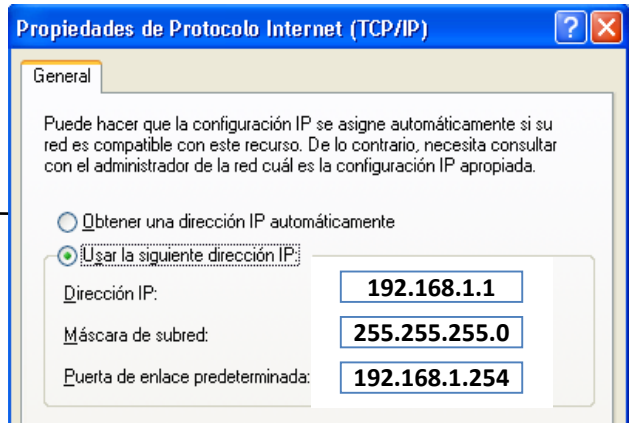
- **Construcción de la tabla de rutas**
  - Dependiendo del tamaño o complejidad de Internet..
    - **Encaminamiento estático:** Rutas fijas establecidas durante el arranque (boot)
      - Útil en casos muy simples, cuando los cambios de encaminamiento son lentos y poco frecuentes
    - **Encaminamiento dinámico:** Inicialización en arranque y actualización por protocolo
      - Protocolos de encaminamiento (intercambio de información entre los *router*)
      - Necesario en grandes redes, con cambios frecuentes y rápidos
  - En definitiva, dos fuentes de información
    - Inicialización (ej. de disco) → *Host* normalmente "congelan" la tabla tras inicializar
    - Actualización (ej. a partir de protocolos) → Los *Router* aprenden información nueva y actualizan las tablas
- **Entradas "especiales"** de la tabla de rutas
  - **Ruta basada en *host*** (*Host-Specific*)
    - Se corresponde con un valor completo de 32-bit: @IP de *host*, no de red.
    - Se puede utilizar para enviar tráfico a un *host* específico a través de un camino concreto.
  - **Ruta basada en *net*** (*Host-Specific*)
    - Se corresponde con un valor de red y su mascara correspondiente.
    - Se puede utilizar para enviar tráfico a una *net* específica a través de un camino concreto.
  - **Ruta por defecto** (*default*) Únicamente se permite una entrada de este tipo
    - Se corresponde con "cualquier" dirección destino (ej. 0.0.0.0/0)
    - Únicamente se utiliza si no hay otra correspondencia en la tabla

Encaminamiento



Red destino	Máscara	Métrica	Siguiente salto	Interfaz de salida
192.168.1.0	255.255.255.0	0	*	eth1
192.168.2.0	255.255.255.0	0	*	eth0
192.168.3.16	255.255.255.252	0	*	ppp0
192.168.3.0	255.255.255.240	0	192.168.3.18	ppp0
0.0.0.0 (default)	0.0.0.0	0	192.168.3.18	ppp0

Valdría una única entrada (*default*)



Contenidos

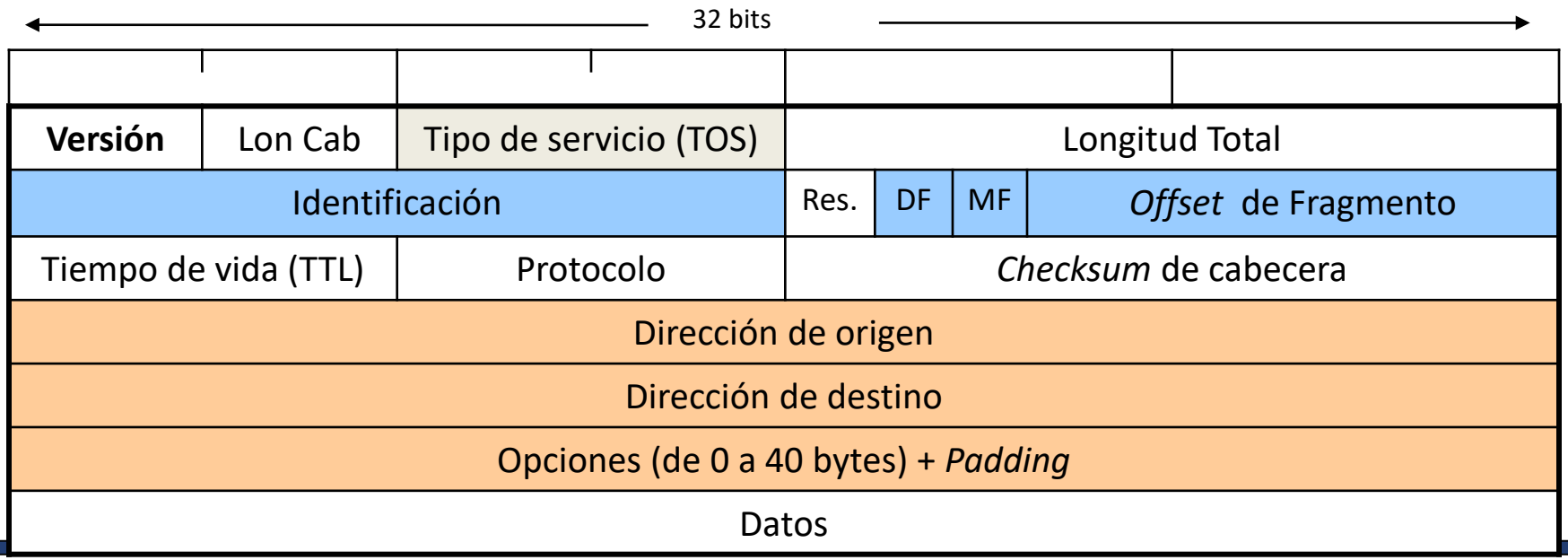
- Repaso Protocolo Internet (IPv4)
  - Internet
  - Direccionamiento
  - Encaminamiento
  - **Funcionalidad del Protocolo IPv4.**

    -PDU - Fragmentación y reensamblado.
  - NAT y NAPT.
- Protocolos de Encaminamiento
  - RIP y OSPF
- Funciones de control: apoyo en otros protocolos
- Gestión de redes TCP/IP: arquitectura SNMP

Funcionalidad: PDU

- **PDU: DATAGRAMA IP**
  - **NO ORIENTADO A CONEXIÓN: QoS *Best effort***
    - No hay conexión extremo a extremo (sin establecimiento, sin información de estado…)
    - Cada paquete tratado de forma individual (encaminamiento según dirección destino)
    - No hay garantías de entrega
  - Tamaño máximo / mínimo = 65535 / 28 (20 de cabecera completa + 8 de fragmento mínimo)
  - Tamaño mínimo MTU recomendable = 576 bytes
  - Tamaño variable de cabecera: Opciones

Diferenciación CIRCUITO VIRTUAL
(recordatorio: Redes Computadores)



Funcionalidad: PDU

Campo	bits	Descripción
Versión	4	Número de versión, incluido para permitir la evolución del protocolo. Actualmente es la versión 4.
Longitud de cabecera	4	Longitud de la cabecera del datagrama en palabras de 32 bits. La longitud mínima es 5. Si cabecera contiene opciones, estas deben ser múltiplo de 32 bits, rellenándose con ceros los bytes no usados (*padding*). ⇒ **LONGITUD VARIABLE**
Tipo de servicio	8	Tipo de servicio deseado.
Longitud total	16	Longitud total del datagrama en octetos.
Número de identificación	16	Permite al destinatario identificar los datagramas. Cada datagrama lleva un identificador diferente, salvo si son fragmentos de un único datagrama
Flags ***Don't Fragment***	1	si tiene el valor 1, no puede ser fragmentado.
More Fragment	1	indica que es un datagrama fragmentado. Todos los fragmentos tienen este bit a 1, excepto el último fragmento.
Offset de fragmento	13	Indica la posición del fragmento **medida en unidades de 8 bytes** (64 bits): En un datagrama, todos los fragmentos, a excepción del último, deben contener un campo de datos múltiplo de 64 bits
Tiempo de vida (TTL)	8	Evita la circulación indefinida de un datagrama. Es decrementado por los *router* y cuando llega a cero es descartado.
Protocolo	8	Indica el protocolo de nivel inmediatamente superior transportado.
***Checksum* de cabecera**	16	Posibilita la detección de errores en la cabecera del datagrama (no en los datos)
Direcciones (emisor/receptor)	32	Indican las direcciones de la fuente y el destino.
Opciones + *Padding*	var.	Usado para codificar opciones pedidas por el emisor. Caso de que la longitud de las opciones no fuera múltiplo de 32 bits, se rellenan con ceros los bits no usados (*padding*).

Funcionalidad: PDU

Tipo de Servicio

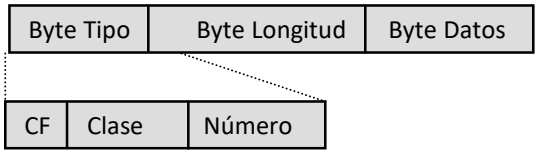
Precedencia	D	T	R	O	O

Bits	0-2	Prioridad	
Bits	3	Retardo	0 – normal, 1 – bajo
Bits	4	Caudal	0 – normal, 1 – alto
Bits	5	Fiabilidad	0 – normal, 1 – alta
Bits	6-7	Reservado	

Números de Protocolo Asignados

1	*Internet Control Message Protocol* (ICMP)
3	*Gateway-to-Gateway Protocol*
6	*Transmission Control Protocol* (TCP)
8	*Exterior Gateway Protocol* (EGP)
11	*Network Voice Protocol*
17	*User Datagram Protocol* (UDP)

Formato de Opciones



Campos en octeto tipo de opción

Bits	0	Flag de copia	0 – No copiar, 1 – copiar
Bits	1-2	Clase de opción	00 – control 01 – reservado 10 – *debug* y medida 11 – reservado
Bits	3-7	Número de opción	(en tabla siguiente)

Opciones en IP

Clase	Número	Longitud	Descripción
0	0	1	Fin de lista de opciones
0	1	1	No operación
0	2	11	Seguridad
0	3	variable	Encaminamiento fuente (*loose*)
0	9	variable	Encaminamiento fuente (estricto)
0	7	variable	Registro de ruta
2	4	variable	Internet *Timestamp*

Funcionalidad: PDU

Opciones en IP

Opción	Función	Máx.	Ej. Windows
Record route	Va anotando en la cabecera IP la ruta seguida por el datagrama	9	Ping –r
Timestamp	Va anotando la ruta y además pone una marca de tiempo en cada salto	4	Ping –s
Strict source routing	La cabecera contiene la ruta paso a paso que debe seguir el datagrama	9	Ping –k
Loose source routing	La cabecera lleva una lista de *router* por los que debe pasar el datagrama, pero puede pasar además por otros	9	Ping -j

El límite de 9 direcciones lo fija el tamaño máximo del campo opciones. En la opción *timestamp* este valor se reduce a 4 porque cada salto anotado ocupa 8 bytes (4 de la dirección y 4 del *timestamp*)

```
ping [-t] [-a] [-n cuenta] [-l tamaño] [-f] [-i TTL] [-v TOS]
       [-r cuenta] [-s cuenta] [[-j lista-host] | [-k lista-host]]
       [-w tiempo de espera] nombre-destino
```

Func.: Fragmentación y reensamblado

- Diversos tamaños de paquete en las diversas redes conectadas en Internet:
  - Necesario acomodar los datagramas IP a los tamaños de la red por donde estos circularán.

- Fragmentación/reensamblado proporcionados por el nivel IP
  - Fragmentación **Intranet**:
    - Los *router* intermedios deben reensamblar y volver a fragmentar:
      - *Buffer* suficientemente grandes
      - Todos los fragmentos deben pasar por el mismo *router* (no encaminamiento dinámico)
    - Aprovecha el máximo tamaño de paquete de cada red.
  - Fragmentación **Internet**:
    - Los *router* no reensamblan: posibilita el encaminamiento dinámico.
    - Pero… se pierde eficiencia en algunas redes

Func.: Fragmentación y reensamblado

- Fragmentación **Internet** (lo que se usa…)

Flag de NO FRAGMENTAR

Fragmentos **múltiplos de 8 bytes** (salvo último)

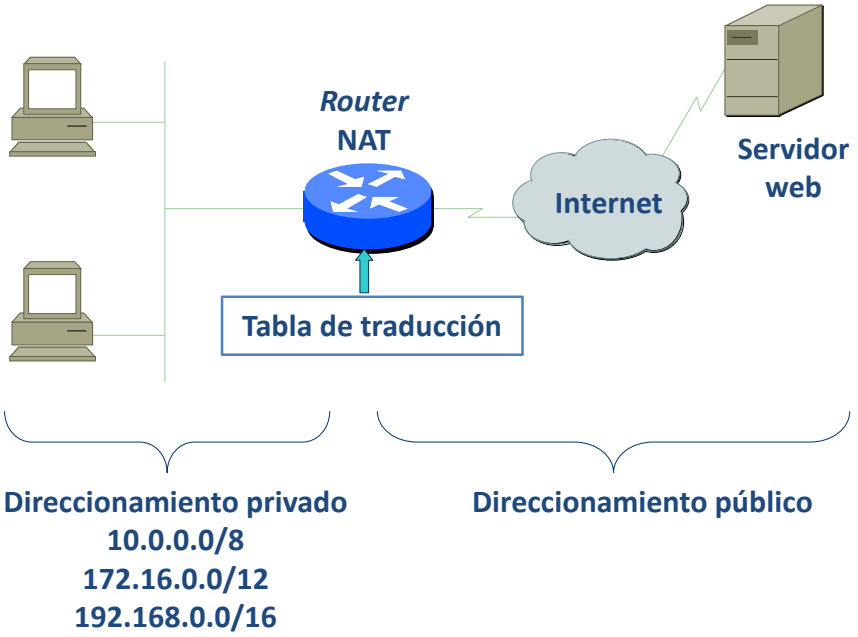
	Id	Long	DF	MF	*Offset*	Datos
Token Ring 4020 – 20 (IPheader) = 4000						
Datagrama Original	XX	4020	0	0	0	ABCDEF GHIJKL MNOP
Ethernet (MTU=1500) 1500 – 20 (IPheader) = 1480 1480 = 185*8						
Fragm. 1	XX	1500	0	1	0	ABCDEF
Fragm. 2	XX	1500	0	1	185	GHIJKL
Fragm. 3	XX	1060	0	0	370	MNOP
PPP (MTU=296) 296 – 20 (IPheader) = 276 272 = 34*8 Múltiplo de 8 más cercano						
Fragm. 3a	XX	292	0	1	370	M
Fragm. 3b	XX	292	0	1	404	N
Fragm. 3c	XX	292	0	1	438	O
Fragm. 3d	XX	244	0	0	472	P

Contenidos

- Repaso Protocolo Internet (IPv4)
  - Internet
  - Direccionamiento
  - Encaminamiento
  - Funcionalidad del Protocolo IPv4.
    -PDU - Fragmentación y reensamblado.
  - **NAT y NAPT.**
- Protocolos de Encaminamiento
  - RIP y OSPF
- Funciones de control: apoyo en otros protocolos
- Gestión de redes TCP/IP: arquitectura SNMP

Direccionamiento: NAT

- NAT (*Network Address Translation*)
  - Traducción de unas direcciones IP en otras de acuerdo a una tabla de equivalencias
    - Para qué
    - Tipos de NAT
    - Problemática de NAT



Direccionamiento: NAT

- NAT (*Network Address Translation*) - ¿Para qué?
  - Limitación en la disponibilidad de direcciones públicas
    - una dirección IP para varios dispositivos
  - "Seguridad"
    - Los bloques RFC-1918 no son 'enrutados' (privados)
      - Los *router* suelen bloquear cualquier paquete con estas direcciones en origen o destino
    - Ningún AS debe publicar estos bloques
    - Se enmascara la topología de la red interna (se puede cambiar)
  - Gestión
    - Protegerse de los cambios de bloques del ISP
  - **Ejemplo típico: Conexión a Internet desde casa**
    - El ISP asigna dinámicamente la dirección IP pública del *router*, permitiendo la conexión
    - El *router* asigna dinámicamente direcciones IP privadas a los equipos de casa

OJO!: Una "ocultación" segura requiere *firewalls* – (filtros de seguridad)

Direccionamiento: NAT

- NAT (*Network Address Translation*) - Tipos de NAT

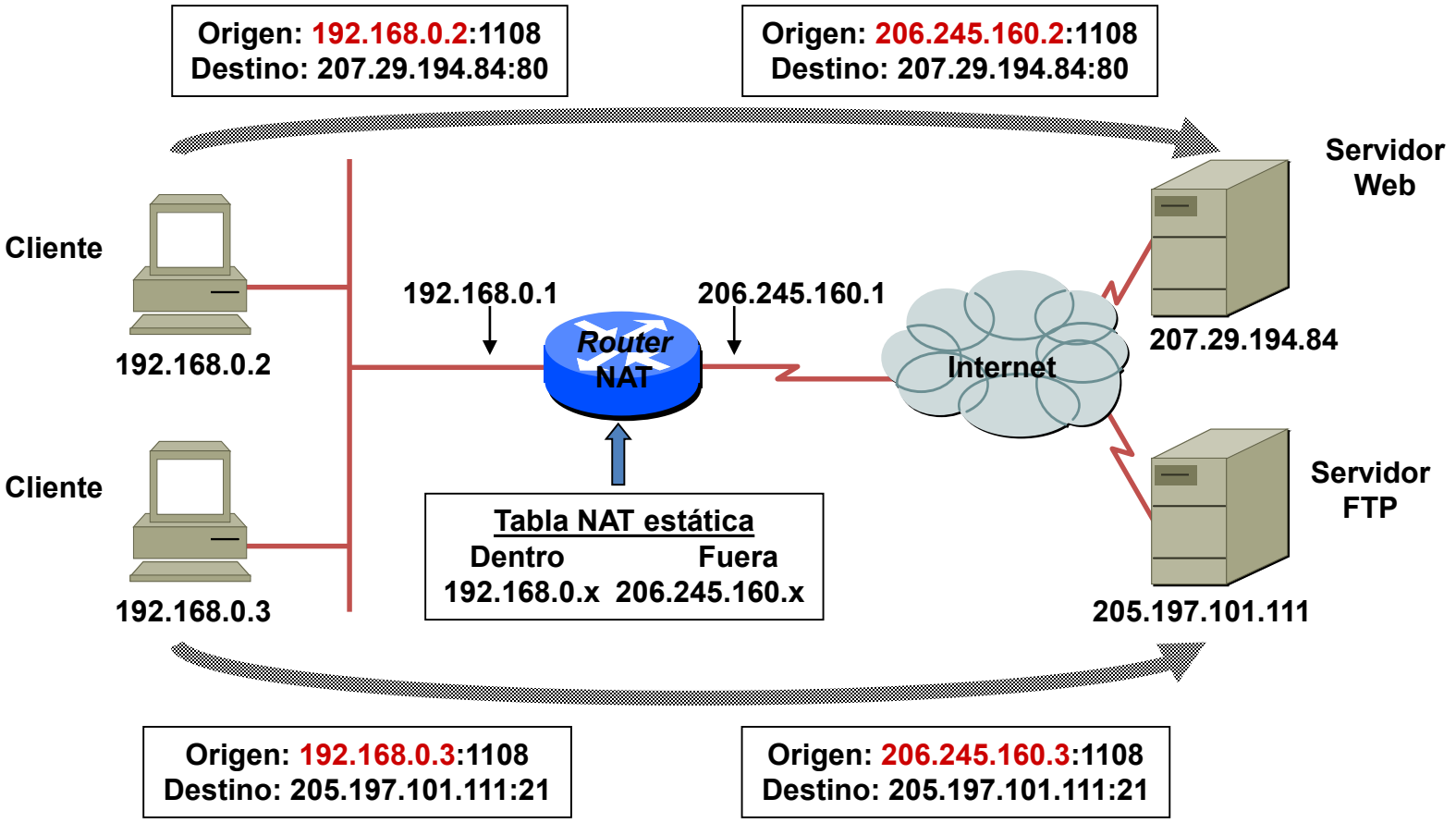
	Estático	Dinámico
NAT Básico **Traduce IP**	**Traducción 1 a 1** El número de direcciones públicas ha de ser igual al de privadas	**Traducción N a m (N>m)** El número de direcciones públicas puede ser menor, pero ha de ser suficiente para el número de ordenadores conectados simultáneamente
NAPT **Traduce IP y puerto** ***(Overloading)*** ***(IP masquerade)***	Uso para conexiones entrantes: permite asociar a una sola dirección diferentes servidores	Típico: **Traducción N a 1 (m P's)** Una sola dirección pública permite la conexión de múltiples ordenadores

Direccionamiento: NAT

- NAT (*Network Address Translation*) – Funcionamiento básico
  - **Reemplazar direcciones IP**
    - Al salir: Cambiar dirección fuente
    - Al entrar. Cambiar dirección destino
  - **Tabla de traducción NAT = recordar las asociaciones**:
    - Asociación de direcciones internas con direcciones externas (públicas, accesibles)
    - Preconfigurada (NAT estático) o creada dinámicamente (entradas con un tiempo de vida)
  - **"los clientes primero"**
    - En el exterior no conocen la traducción de una dirección interna:
      - el cliente inicia la conexión
      - el *router* asigna "dirección interna local" ↔ "dirección externa global".
        - » Desde el exterior podemos conectarnos a la "dirección externa global". Si está la asignación previa, "entramos". Si no hay asociación, desde fuera no podremos conectarnos a dentro (problema P2P!)
  - Para un **servidor interno** (no inicia, sino espera conexión) **ya debe existir la traducción** en NAT previamente y de forma permanente: **mapeo estático**

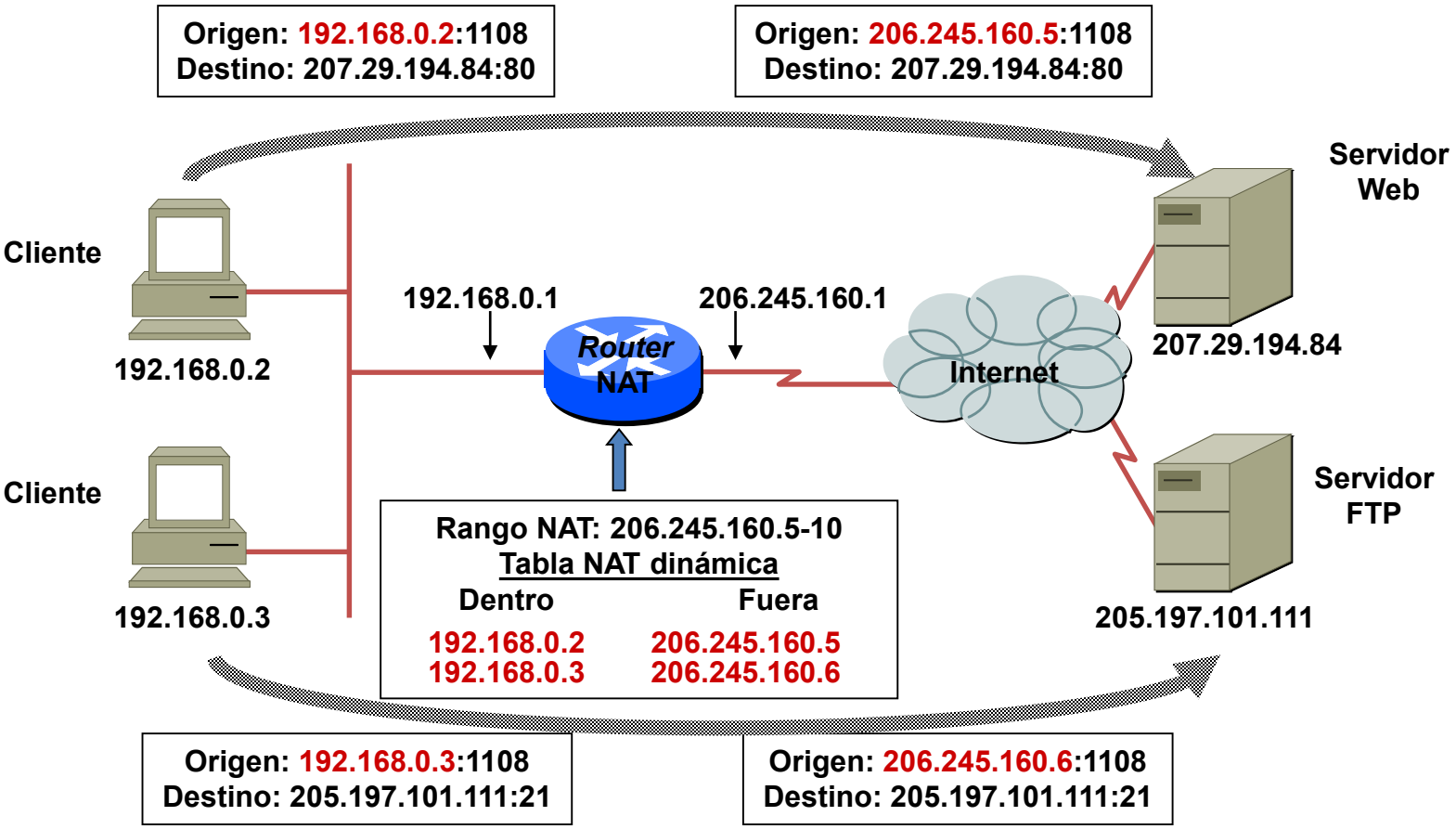
Direccionamiento: NAT

- NAT básico estático



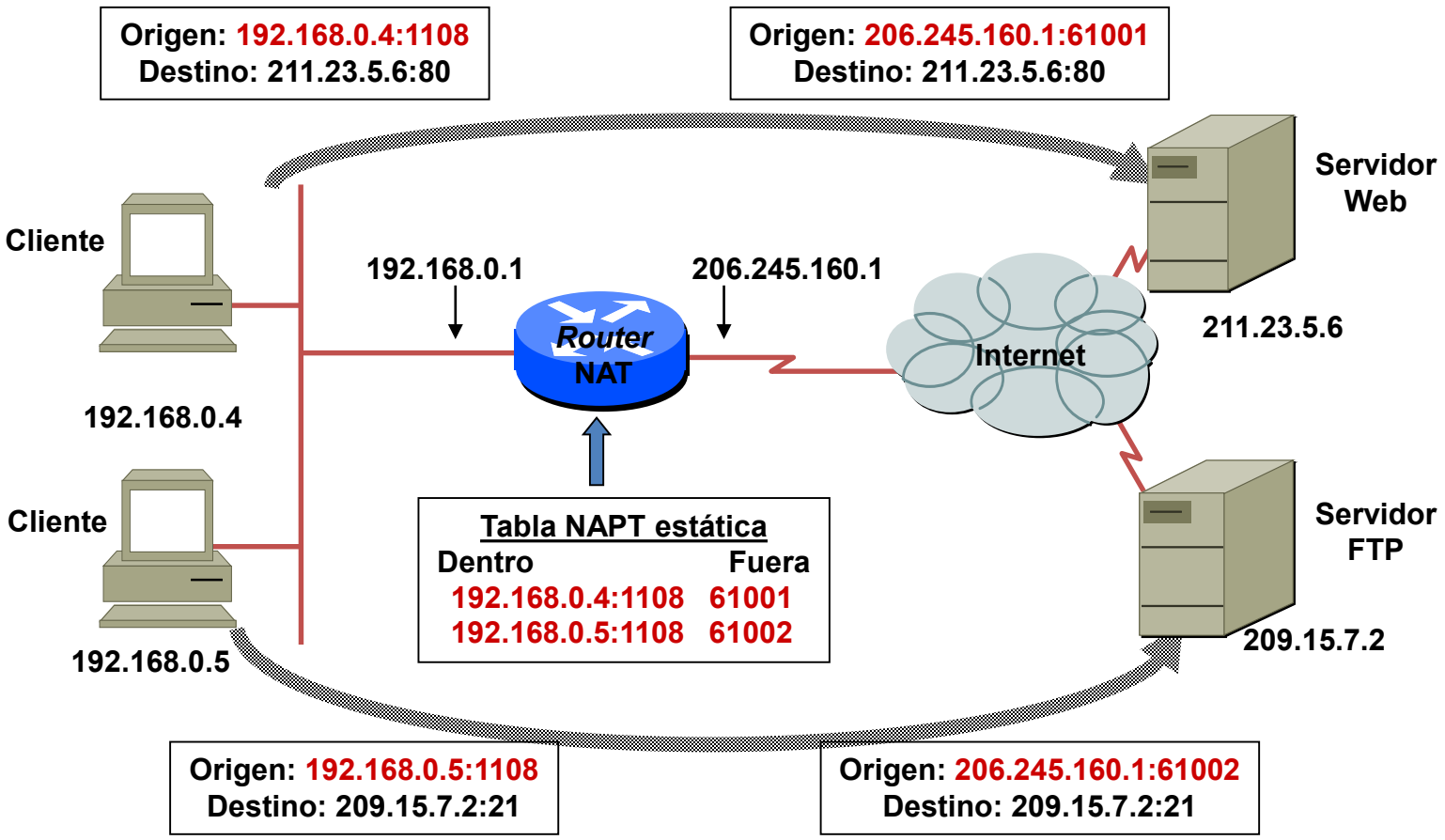
Direccionamiento: NAT

- NAT básico dinámico



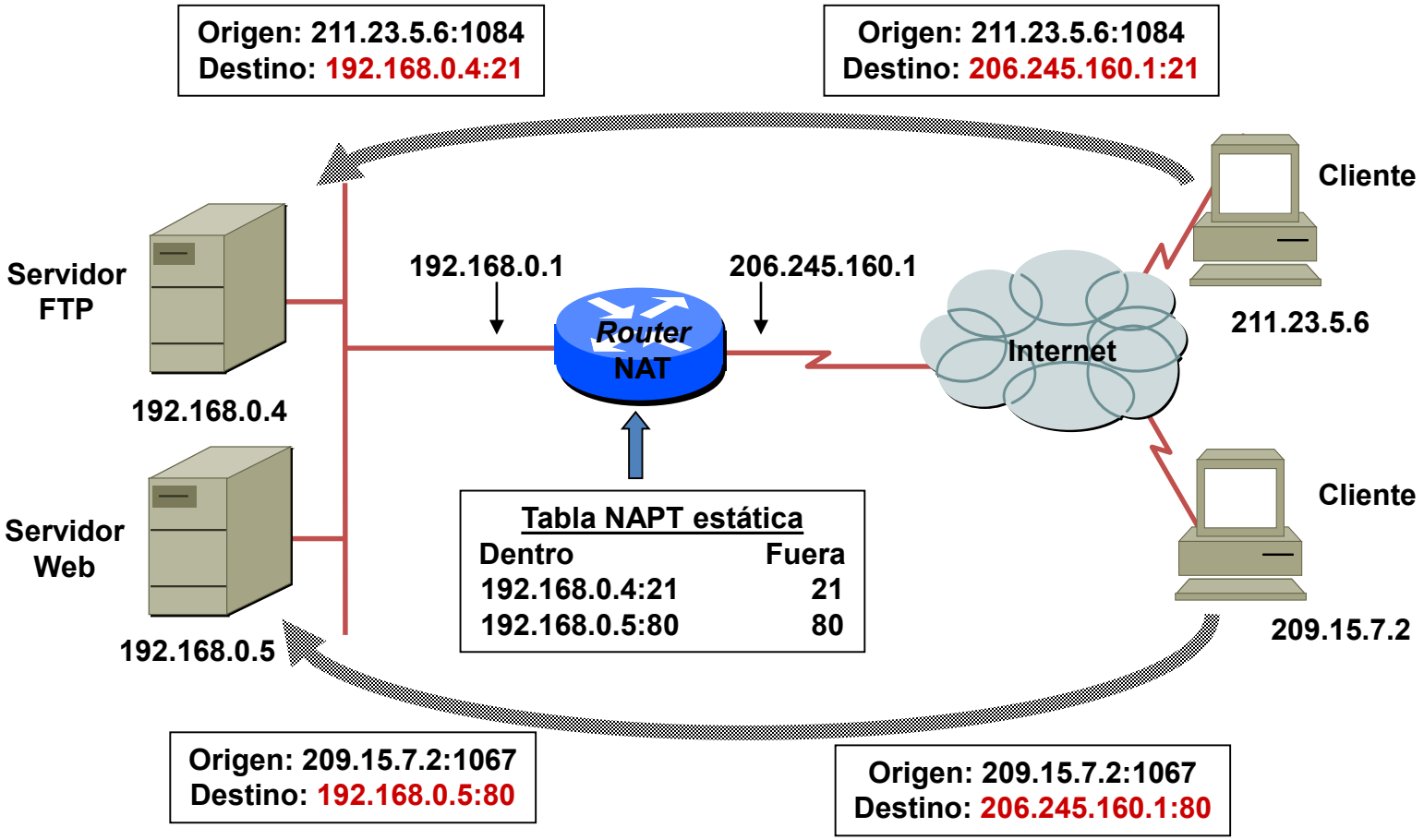
Direccionamiento: NAT

- NAPT dinámico



Direccionamiento: NAT

- NAPT estático



Direccionamiento: NAT

IPv6 para resolver la limitación de direcciones !!

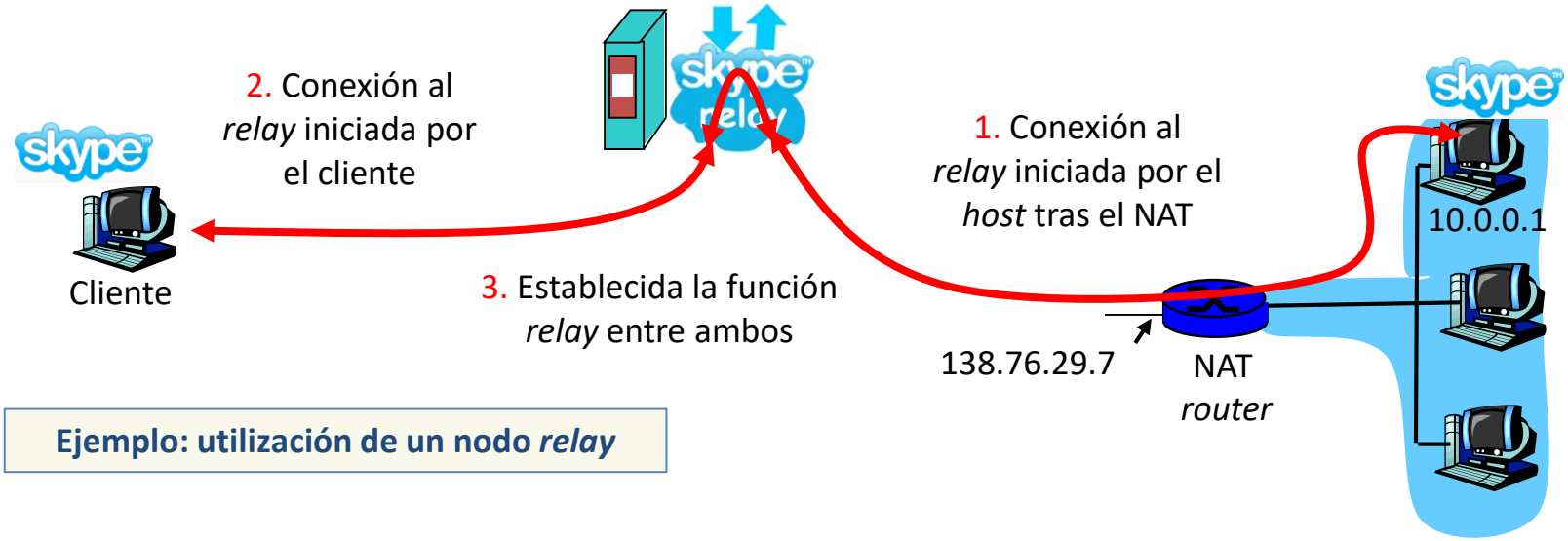
- NAT (*Network Address Translation*) - Problemática de NAT
  - **Los *router* sólo deberían procesar hasta el nivel IP**
    - Protocolos que incluyen @IP/#puerto necesitan ser también modificados: **ICMP** (*Destination Unreachable*); **FTP/UDP**; **H.323** o **SIP** (videoconferencia, VoIP)…
    - Las nuevas implementaciones resuelven algunos de éstos: El *router* tiene que inspeccionar el contenido del paquete IP (más carga de procesamiento) – **ALG(*)**
  - **La traducción NAPT con ICMP no es directa (no hay puertos)**
    - Solución para *query*/*response* (e.g. *ping*): se usa el ID de ICMP como si fuera el puerto
  - **Problemas de cifrado y control de errores**:
    - Cálculo del *checksum* en TCP/UDP sobre una "pseudocabecera" que incluye las direcciones IP
    - AH-IPsec corrobora la integridad de la cabecera, ¿y si se modifica?:
      - Con NAT sólo se puede utilizar la función AH en modo túnel y el NAT se haga antes, o en el mismo dispositivo donde se hace el túnel IPSec.
  - **Aplicaciones P2P**: ¿cómo contacto con un peer sin conocer su dirección IP?
    - Soluciones apoyadas en servidores auxiliares
      - Conexión inversa, STUN, UDP *hole punching*, nodos *relay*…

(*) *Application-level gateway* (ALG) aumenta la funcionalidad de un firewall o NAT. Soporta la traducción de direcciones/puertos en aplicaciones que utilizan esta información en mensajes de control o datos (FTP, SIP, RTSP, etc.) Implica que el *firewall* o NAT trabaja ("entiende") a niveles superiores a IP

Direccionamiento: NAT

IPv6 para resolver la limitación de direcciones !!

- NAT (*Network Address Translation*) - Problemática de NAT
  - **Aplicaciones P2P**: Soluciones apoyadas en servidores auxiliares:
    - Conexión inversa: indicar al destino que inicie él la conexión
    - El cliente detrás del NAT puede averiguar su @IP/#puerto contactando con un servidor conocido (servidor STUN, protocolo STUN)
    - Utilizando UDP (*hole punching*): primero "salir", después, desde fuera "pueden entrar" (salir hacia servidor conocido por los dos pares que colabora con ellos)
    - Nodos *relay*: Los dos pares se conectan a un tercero para realizar la comunicación



Ejemplo: utilización de un nodo *relay*

Contenidos

- Repaso Protocolo Internet (IPv4)
  - Internet
  - Direccionamiento
  - Encaminamiento
  - Funcionalidad del Protocolo IPv4.
    -PDU - Fragmentación y reensamblado.
  - NAT y NAPT.
- **Protocolos de Encaminamiento**
  - **RIP y OSPF**
- Funciones de control: apoyo en otros protocolos
- Gestión de redes TCP/IP: arquitectura SNMP

Encaminamiento dinámico

- **Algoritmos** asociados permiten calcular el camino óptimo, y por tanto decidir la interfaz de salida (e.g algoritmos de mínimo coste)

- **Protocolos** de encaminamiento definen
  - Formato y contenido de los mensajes que se intercambian entre *router*
  - La forma de intercambiar la información (ej.; *unicast*, *broadcast*, *multicast*, ...)
  - La periodicidad del intercambio

- Concepto de **"convergencia"** en un protocolo de encaminamiento
  - Cuando la topología de la red cambia, los *router* deben recalcular las rutas y actualizar las tablas de encaminamiento
  - El **tiempo en que todos los *router* alcanzan un conocimiento homogéneo de la red** se le llama "**tiempo de convergencia**"
  - Tiempos de convergencia grandes implican que los *router* tendrán mayor dificultad para enviar los datagramas por la interfaz más adecuada y por tanto descarte de paquetes
  - La convergencia depende de
    - Distancia en saltos desde el punto en que se produjo el cambio
    - Cantidad de *router* que usan el protocolo dinámico
    - El ancho de banda y la carga de tráfico de la red
    - La carga del *router* (CPU)
    - El protocolo de encaminamiento usado (el algoritmo)
    - La configuración que haga el administrador de la red (e.g.; red con bucles por un mal diseño)

Encaminamiento dinámico - algoritmos

- Algoritmos de encaminamiento:
  - Propagación automática de información de encaminamiento (rutas)
  - Utilización de la información recogida para calcular el camino (y métrica) a cada destino

- Dos algoritmos básicos utilizados en los protocolos de actualización
  - **Vector distancia** (*Distance-vector*) → adquisición iterativa de información distribuida (vecinos y sus rutas)
    - Los *router* van aprendiendo las rutas de sus vecinos y calculan las suyas
  - **Estado del enlace** (*Link-state*) → adquisición de información global
    - Los r*outer* conocen topología de red (información de coste del enlace)
  - También existen híbridos
  - Muchas variaciones en los detalles de implementación

- Encaminamiento jerárquico

Encaminamiento jerárquico

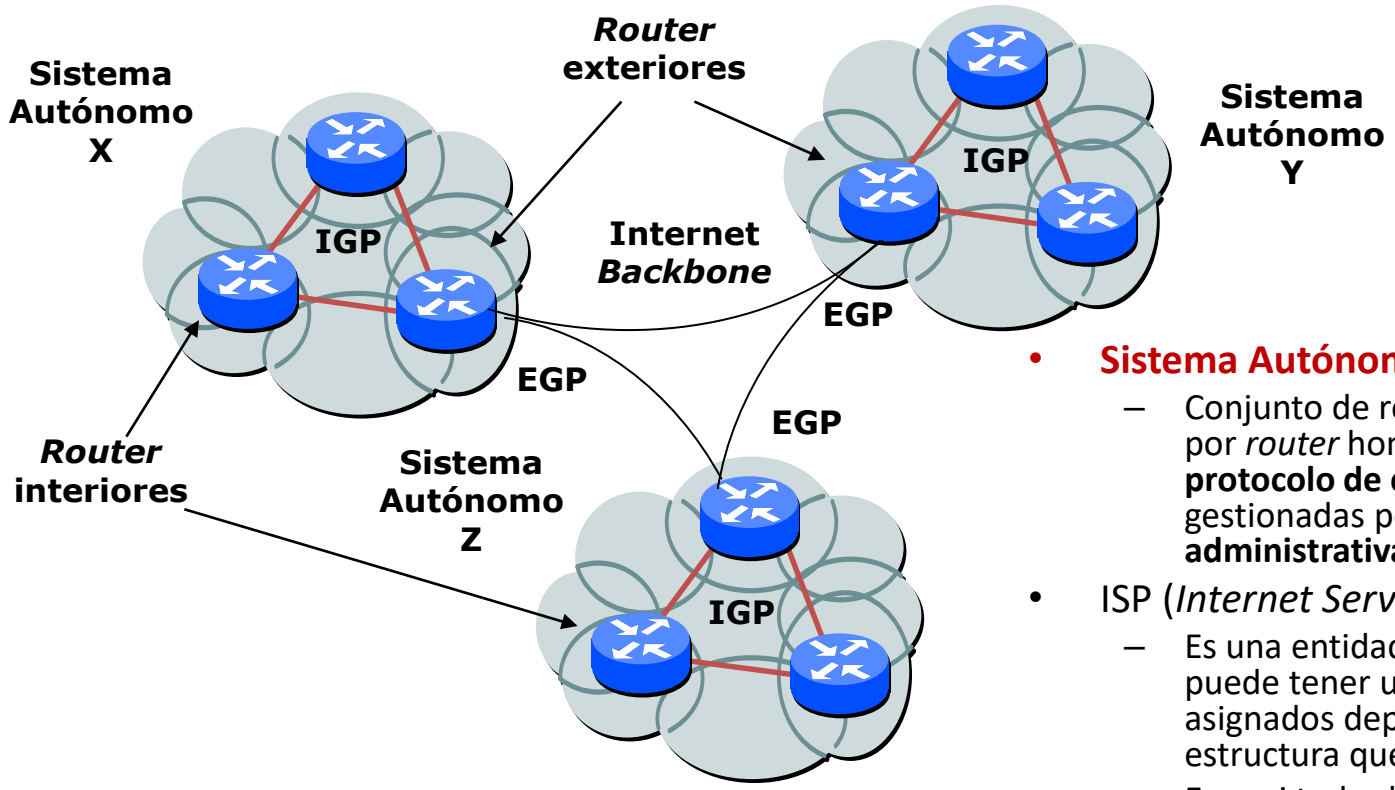
- Algoritmos de encaminamiento vistos: “ideal”
  - Todos los *router* son idénticos
  - La red es plana
  - No es cierto en la realidad

- Es necesario **escalar** las soluciones:
  - Con 200 mill. de destinatarios
    - ¡¡ No se pueden almacenar tantos destinos en las tablas !!
    - ¡¡ Los intercambios de tablas de rutas entre los *router* puede inundar la red !!

- Existe **autonomía administrativa**
  - Internet = red de redes
  - Cada administrador puede querer controlar el encaminamiento de su propia red

- Solución: agregar los *router* en regiones: **Sistema Autónomos** (AS)
  - Los r*outer* en el mismo AS utilizan el mismo protocolo de encaminamiento
  - Los r*outer* en distintos AS pueden utilizar distintos protocolos de encaminamiento

Encaminamiento dinámico - protocolos



ICANN

- Protocolos de encaminamiento:
  - ***Interior Gateway Protocol*** (**IGP**). En un AS
    - RIP, OSPF. (**Intra – AS**)
  - ***Exterior Gateway Protocol*** (**EGP**). Entre ASs
    - EGP, BGP. (**Inter – AS**)

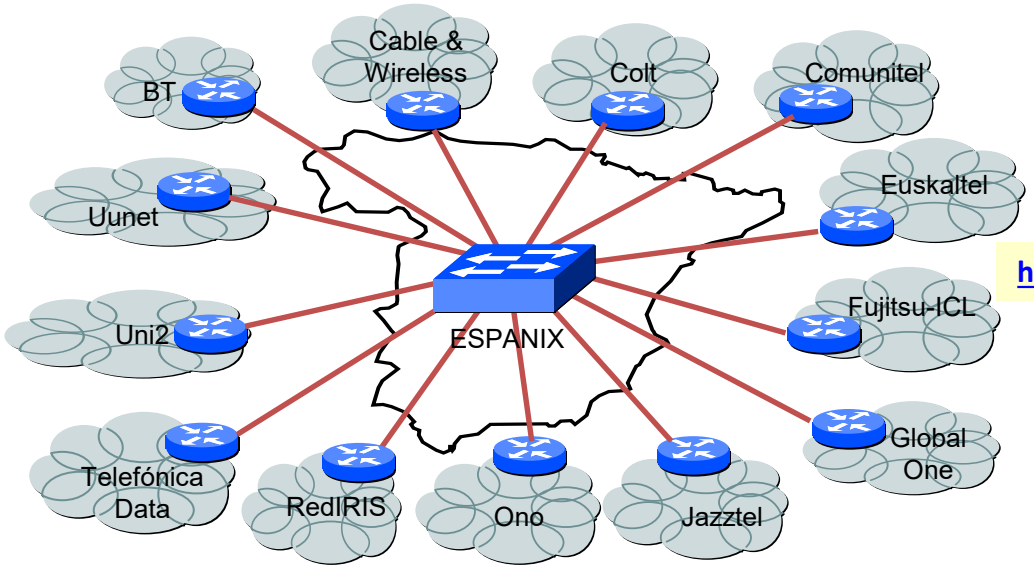
- **Sistema Autónomo** (id: nº de 16 bits):
  - Conjunto de redes interconectadas por *router* homogéneos (**mismo protocolo de encaminamiento**) y gestionadas por **una única entidad administrativa**
- ISP (*Internet Service Provider*)
  - Es una entidad administrativa que puede tener uno o más números AS asignados dependiendo de la estructura que tenga
  - En casi todas las ocasiones un AS se corresponde con un ISP, pero no siempre (e.g.; un AS de tránsito o un punto neutro)

Ejemplo de AS: **RedIRIS** (AS = 766)
http://www.rediris.es/

Encaminamiento dinámico - protocolos

- ***Peering* de ISPs: Punto neutro**
  - La información de encaminamiento entre AS's oculta el encaminamiento interno, de modo que el encaminamiento puede no ser ÓPTIMO entre 2 puntos cualesquiera (ej, 2 ISP's)
  - Solución: establecer acuerdos bilaterales entre proveedores para crear enlaces directos entre sus ASs
  - ¿Muchos pares / acuerdos? ⇒ Punto de interconexión neutro
    - Entidad independiente a la que se conectan los *router* de los ISP's participantes
  - En España: **ESPANIX** (Centro de Proceso de Datos de Banesto, Madrid, desde 1997)

El intercambio de tráfico en el punto neutro debería ocurrir sin restricciones entre todos pero en la práctica los proveedores han de establecer acuerdos bilaterales (no existe realmente un "todos con todos")

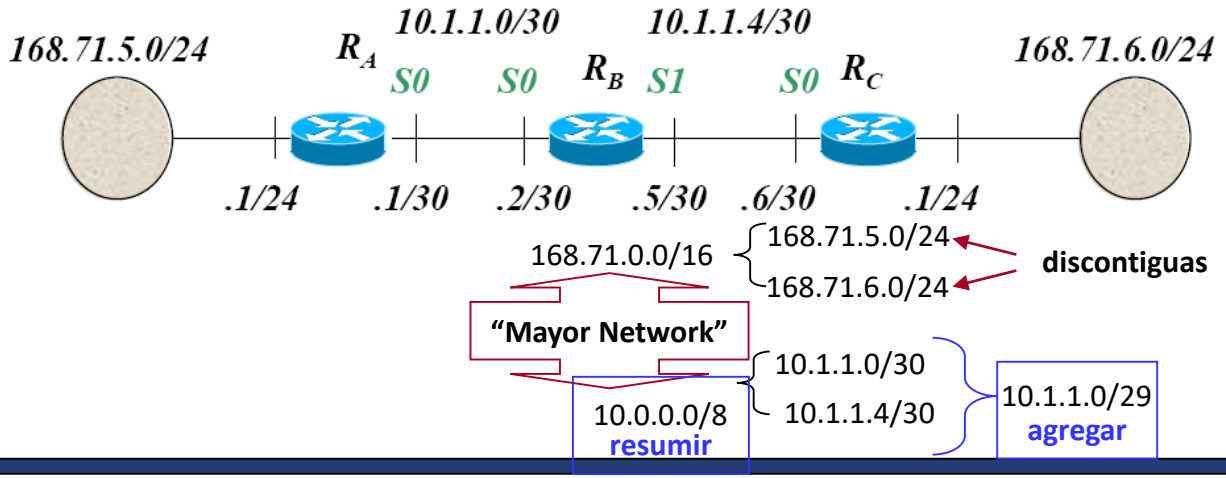


http://www.espanix.net/

http://www.catnix.net/

Encaminamiento dinámico - protocolos

- Algunos conceptos…
  - **Red principal** (“Mayor Network”):
    - se refiere a la porción de red de mayor rango de una dirección IP. Generalmente es la clase a la que pertenece (a no ser que conozcamos la red base a partir de la cual ha sido subdividida (*subnetting*))
  - **Red discontigua:**
    - las direcciones de subred de una red principal se aplican a redes físicas separadas por una red principal distinta
  - **Resumen**: indicar sólo la red principal, aunque haya subredes de esa red principal en la tabla de encaminamiento
  - **Agregación**: Reducir el número de subredes en una porción de red común para comunicar sólo esta porción, por ejemplo, en un paquete de refresco (*update*) de un protocolo de encaminamiento. (puede coincidir con la red principal pero no tiene por qué)



Encaminamiento dinámico - protocolos

- Algunos conceptos…
  - *Classful routing*: protocolos que **no anuncian la máscara**
    - No se puede subdividir la red
    - Cuidado con las redes discontiguas porque RIPv1 e IGRP resumen

      RIPv1, IGRP
  - *Classless routing*: protocolos que **anuncian las máscaras**
    - Se puede subdividir la red usando VLSM en toda la red
    - A pesar de que se anuncian las máscaras hay que tener cuidado con las redes discontiguas si los protocolos resumen o agregan subredes

      RIPv2, OSPF, BGP, EIGRP, etc

Encaminamiento dinámico – protocolos IGP

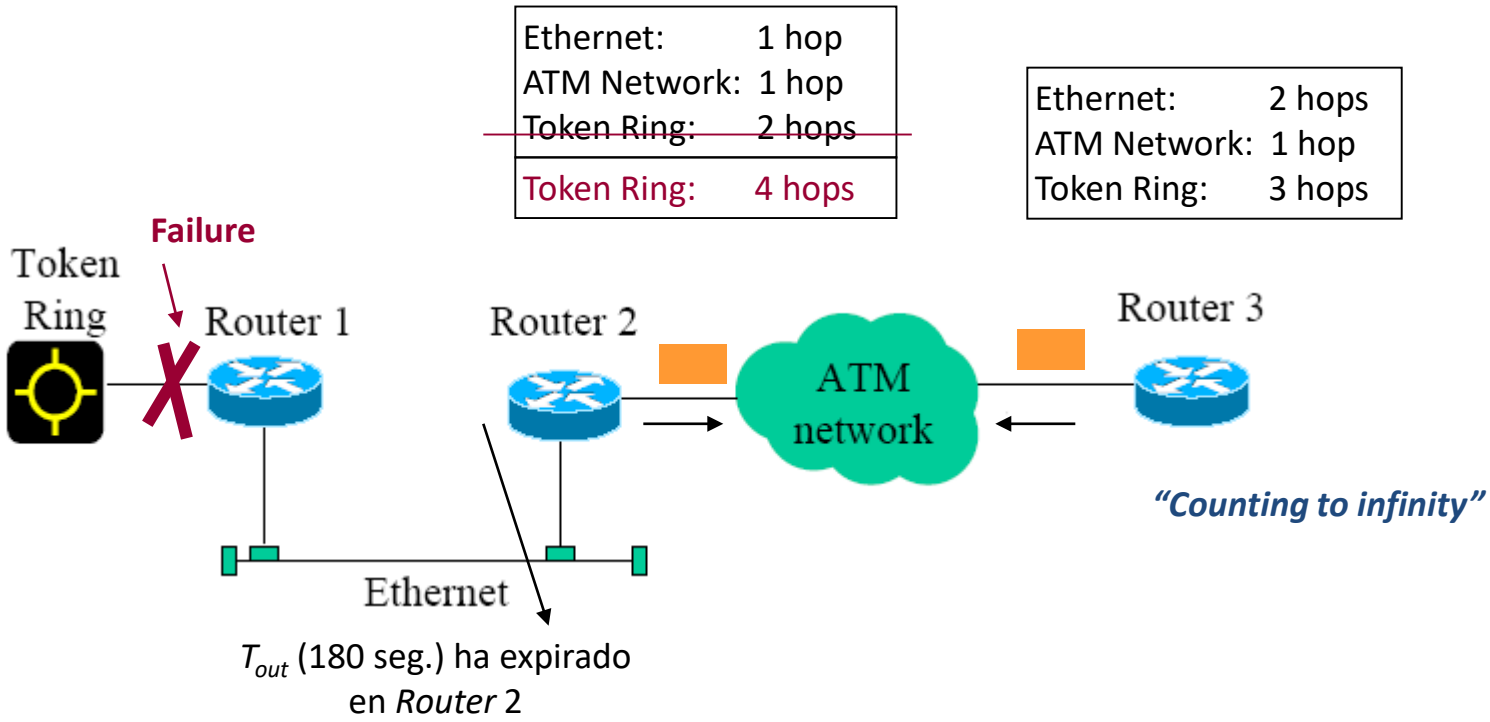
Routing Information Protocol (RIP)	*Open-Short Path First (OSPF)*
***Distance-vector*:** Algoritmo **Bellmand-Ford DISTRIBUIDO**	***Link-state*:** Algoritmo **Dijkstra LOCAL** sobre topología completa
Funciona sobre UDP (puerto 520)	**Funciona sobre IP (protocolo # 89)**
Información **PERIÓDICA** cada 30 seg - Se envía a TODOS lo **VECINOS** - Contiene TABLA COMPLETA (destino, coste) - Coste = # saltos (máximo = 16)	Cada router mantiene: - *Link State Data Base* – LSDB (topología de red) - Tabla de encaminamiento (Dijkstra sobre LSDB) - Se envía información de estado (*Link State Advertisement* - LSA) cada vez que **HAY CAMBIO (envío exige AUTENTICACIÓN)** - A **TODOS LOS *ROUTER*** de la red (participantes) – *flooding*
Detección de caminos redundantes Detección de fallos: *timeout* de actividad de vecinos de 180 seg.	
Fácil de configurar, usar y mantener	- *Flooding* ⇒ ↑ consumo de ancho de banda (eficiencia… depende) - LSA es ↑ información + consumo CPU (Dijkstra)
Útil en redes SENCILLAS (5-10 *router*) – poco escalable	Escalabilidad (crecimiento) gracias a la división en ÁREAS
Métrica no es óptima (sólo número de saltos, y además limitado)	Cualquier métrica (saltos, ancho de banda, retardo…)
Vulnerabilidad: cuenta al infinito (*count to infinity*) Soluciones: - *Split horizon* (*SH*)	
Convergencia puede ser lenta Solución: - *Triggered updates (y SH + Poisson Reverse)*	La información se envía en el momento que la topología de red cambia: con Dijkstra, cada *router* recalcula su tabla - Buena convergencia y reacción ante cambios
RIPv1 no admite máscaras variables	Rutas de red, subred y *host* (VLSM, CIDR)
No permite el uso simultáneo de varias rutas (*multipath*)	Permite varias rutas simultáneas
Sincronización de los *router* en el envío genera bloqueos (tormentas *broadcast*)	
Última versión: **RIPv2**	Última versión: **OSPFv2**

Encaminamiento dinámico (IGP) – protocolo RIP

- ***Count to infinity***
  - Es el síntoma de la creación de bucles en la red.
  - Cuando se van actualizando (reemplazando) las entradas a la tabla para una red inalcanzable, la métrica, al pasar de *router* a *router* (en el bucle) se va incrementando.
  - Limitar el valor ∞ (RIP: 16 saltos) no evita el bucle (Este valor lo que limita es el tamaño - diámetro - máximo de la red)
  - Soluciones:
    - ***Split horizon***: Un *router* NO envía información a otro *router* de las redes que le son comunicadas por ese otro *router*.
    - ***Holddown Timer***: Cuando se aprende que una ruta ha fallado, durante cierto tiempo no se acepta nueva información de esa ruta (evitar bucles transitorios por rutas inestables)
- **Convergencia lenta**
  - ***Poisson Reverse***: (adaptación de *Split Horizon*) Un *router* SÍ envía información a otro *router* de las redes que le son comunicadas por ese otro *router*, PERO con MÉTRICA INFINITO
  - ***Triggered updates***: Para aumentar la convergencia ante cambios en la topología, mejor no esperar las actualizaciones periódicas sino enviar inmediatamente la tabla de rutas actualizada

Encaminamiento dinámico (IGP) – protocolo RIP

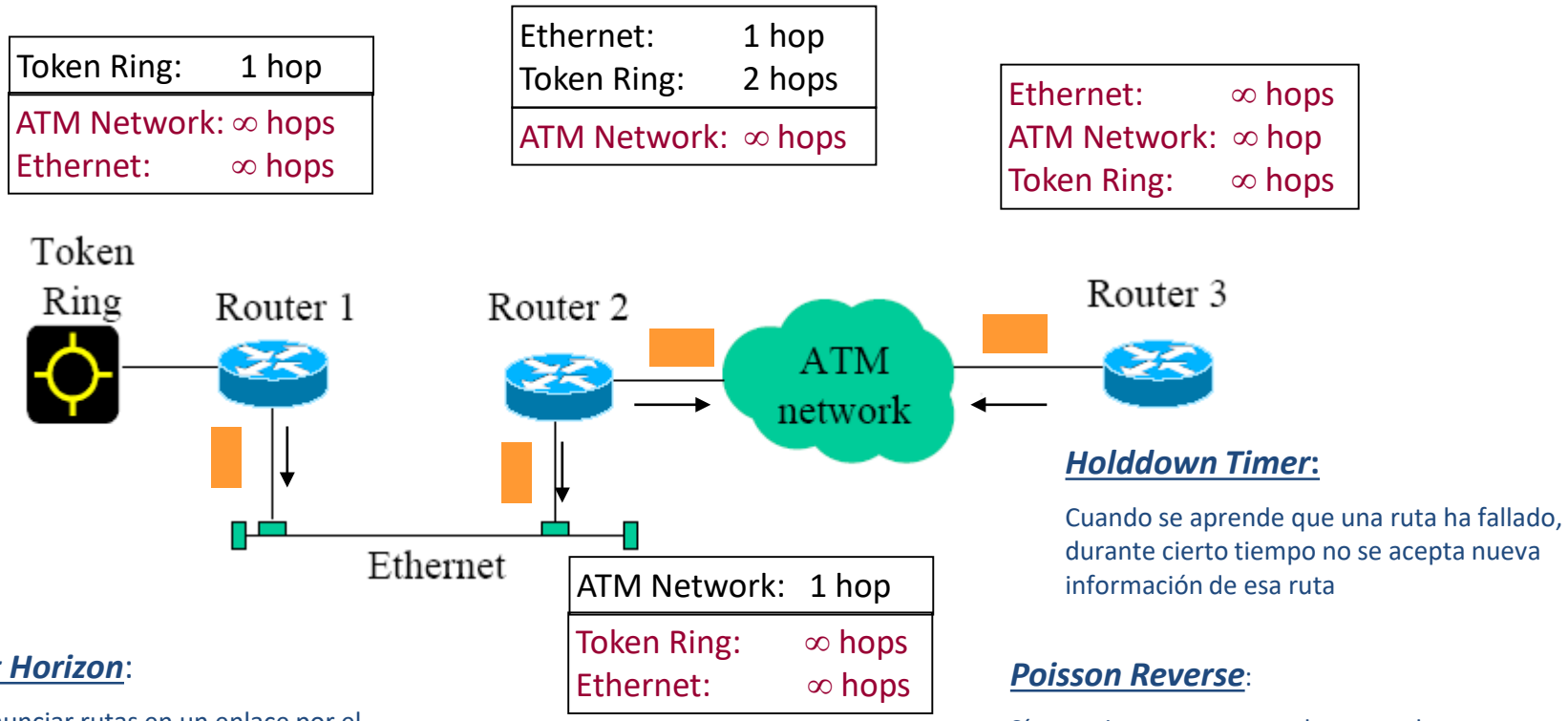
- Ejemplo: *Split Horizon* + *Poisson Reverse*



Cuando en *Router* 2 y *Router* 3 la métrica Token Ring llega a 16 ⇒ inalcanzable

Encaminamiento dinámico (IGP) – protocolo RIP

- Ejemplo: *Split Horizon* + *Poisson Reverse*



***Holddown Timer*:**

Cuando se aprende que una ruta ha fallado, durante cierto tiempo no se acepta nueva información de esa ruta

Split Horizon:

No anunciar rutas en un enlace por el que se recibe la información de dicha ruta

Poisson Reverse:

Sí anunciar rutas en un enlace por el que se recibe la información de dicha ruta, pero COSTE ∞

Encaminamiento dinámico (IGP) – protocolo OSPF

- OSPF: Encaminamiento JERÁRQUICO en ÁREAS (ESCALABILIDAD)
  - ÁREA: Conjunto de redes CONTIGUAS agrupadas
    - Copia separada del algoritmo, su LSDB y su grafo. La topología de un área es invisible fuera del área. Los *router* internos desconocen la topología externa

- OSPF – Funcionamiento "en un área"
  - OSPF puede usarse en diversas topologías (comportamiento distinto):
    - BMA (*Broadcast Multi-Access*): LANs
    - Punto a punto: líneas dedicadas (e.g.; E1)
    - NBMA (*Non-Broadcast Multi-Access*): ATM o FR

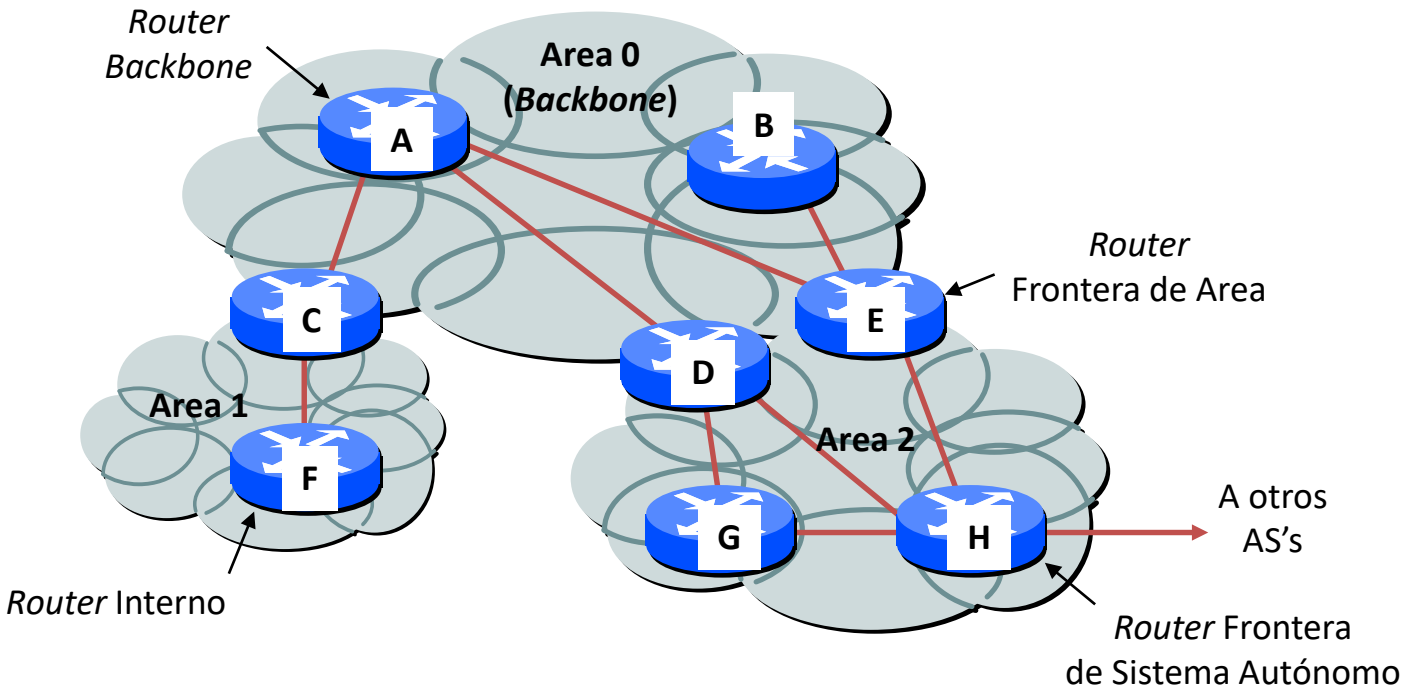
- El funcionamiento general consiste en:
  - Descubrir los vecinos (usando protocolo de HELLO)
  - Enviar LSAs (*LS Advsertisements*) con los cambios en la red
  - Mantener una base de datos con la topología (LSDB) en cada *router*
  - Con el **algoritmo Dijkstra** rellenar la tabla de encaminamiento a partir de la LSDB

Encaminamiento dinámico (IGP) – protocolo OSPF

- OSPF – Funcionamiento en una red multiárea
  - Encaminamiento jerárquico: subdivisión en ÁREAS
    - ***Standard area*** o simple ⇒funcionamiento "OSPF en un área"
    - ***Backbone area*** (de tránsito): área 0 que interconecta otras áreas en un sistema multi-área.
    - ***Stub area***: área que no acepta información de rutas externas al AS. Si los *router* deben conectarse al exterior deben hacerlo usando una ruta por defecto (0.0.0.0)
  - Clases de *router*
    - ***Router backbone***: los que se encuentran en el área 0 o área principal
    - ***Router internos***: pertenecen únicamente a un área
    - ***Router frontera de área***: los que conectan dos o mas áreas (una de ellas necesariamente el *backbone*) – ABR (*Area Border Router*)
    - ***Router frontera de AS***: los que conectan con otros ASes. Pueden estar en el *backbone* o en cualquier otra área – ASBR (*AS Border Router*)
  - Tipos de rutas.
    - ***Intra-área***: las determina directamente el *router*
    - ***Inter-área***: se resuelven en tres fases:
      - Ruta hacia el *router backbone* en el área
      - Ruta hacia el área de destino en el *backbone*
      - Ruta hacia el *router* en el área de destino
    - ***Inter-AS***: se envían al *router* frontera de AS más próximo (empleando alguna de las dos anteriores).

Encaminamiento dinámico (IGP) – protocolo OSPF

- OSPF – Funcionamiento en una red multiárea



Ruta intra-área: D-G-H
Ruta inter-área: F-C,C-A-D,D-G-H
Ruta inter-AS: A-D,D-G-H, H-...

Contenidos

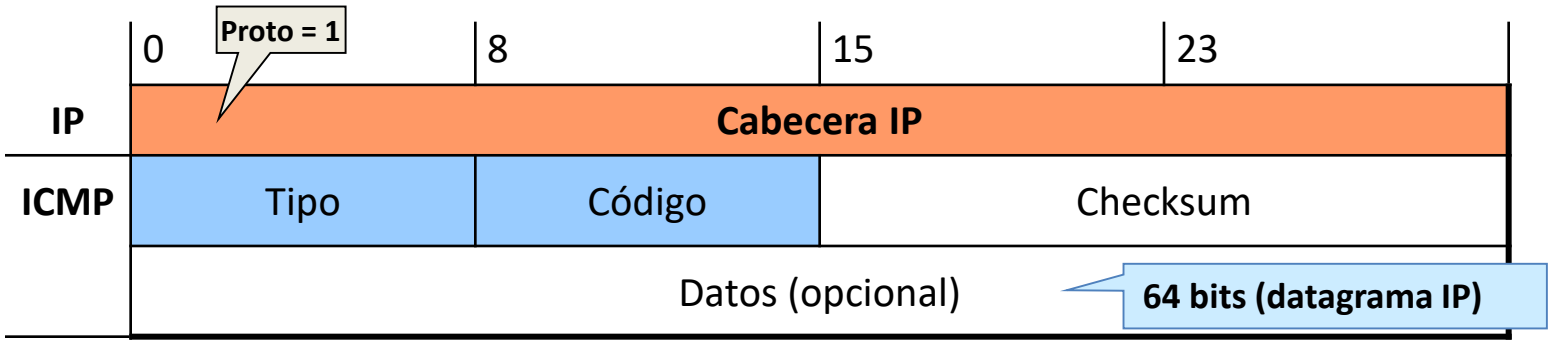
- Repaso Protocolo Internet (IPv4)
  - Internet
  - Direccionamiento
  - Encaminamiento
  - Funcionalidad del Protocolo IPv4.
    -PDU - Fragmentación y reensamblado.
  - NAT y NAPT.
- Protocolos de Encaminamiento
  - RIP y OSPF
- **Funciones de control: apoyo en otros protocolos**
- Gestión de redes TCP/IP: arquitectura SNMP

Protocolos de CONTROL

- **Funcionalidad de control propia del nivel Internet:**
  - protocolo ICMP
- **Configuración IP:**
  - La configuración completa de un *host* incluye:
    - Dirección IP
    - Máscara de red (*classless*) ⇒ @red: encaminamiento directo
    - Dirección de DNS (cualquier servidor DNS – IP pública)
    - *Router* por defecto ⇒ encaminamiento indirecto
  - La configuración completa de un *router* incluye:
    - Dirección IP / máscara de cada interfaz
    - Encaminamiento directo: interfaces de salida para cada red conectado
    - Encaminamiento indirecto: rutas (destino, siguiente salto, métrica) para cada red sin conexión directa
  - Apoyo en protocolos de control:
    - Resolución de direcciones (ARP)
    - Configuración IP (manual, automática: DHCP)

Protocolos de CONTROL - ICMP

- **ICMP (*Internet Control Message Protocol*)**
  - Informa a la fuente original del mensaje sobre situaciones de error o anómalas, siendo esta fuente la que debe referir los errores a niveles superiores que adoptarán las acciones a llevar a cabo. Doble función:
    - **Informativa**
    - **Tratamiento de errores**
  - **Los mensajes ICMP están encapsulados en el campo de datos del datagrama IP**: identifica al nodo (dest) que informa o detectó el error
  - ¿Cómo distinguir el paquete que ha provocado la incidencia, si en ese momento se habían enviado diversos paquetes a distintos destinos?.
    - ICMP puede transportar (datos) los 64 bits iniciales del datagrama que causó el error (información @IP's y puertos)



Protocolos de CONTROL - ICMP

- **ICMP (*Internet Control Message Protocol*)**

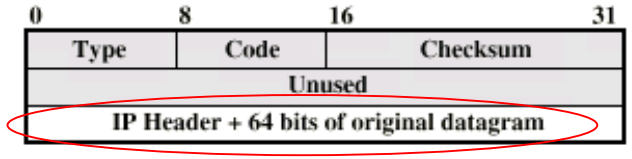
Mensaje (típicos)	Explicación
Destination Unreachable	Red, *host*, protocolo o puerto inaccesible o desconocido
Source quench	Ejerce control de flujo sobre el emisor en casos de congestión. No se utiliza.
Echo request /reply	Test de alcanzabilidad o comprobación de la comunicación (comando ***ping***).
Time exceeded	Datagrama descartado por agotamiento del TTL (comando ***traceroute / tracert*** *(WXP)*)
Redirect	El *router* nos sugiere un camino más óptimo - aunque no deja de reenviar el paquete - (ejemplo. varios *router* en la misma LAN, o cuando un *router* detecta que nos queremos comunicar con un *host* en nuestra misma LAN – sería encaminamiento directo)
Timestamp request /reply	Permite conocer el tiempo de ida y vuelta de un mensaje.

Protocolos de CONTROL - ICMP

Type	
0	***Echo Reply*** (respuesta de eco)
3	***Destination Unreacheable*** (destino inaccesible)
4	***Source Quench*** (disminución del tráfico desde el origen)
5	***Redirect*** (redireccionar - cambio de ruta)
8	***Echo*** **(**solicitud de eco)
11	***Time Exceeded*** (tiempo excedido para un datagrama)
12	***Parameter Problem*** (problema de parámetros)
13	***Timestamp*** (solicitud de marca de tiempo)
14	***Timestamp Reply*** (respuesta de marca de tiempo)
15	***Information Request*** (solicitud de información) – obsoleto
16	***Information Reply*** (respuesta de información) - obsoleto
17	***Addressmask*** (solicitud de máscara de dirección)
18	***Addressmask Reply*** (respuesta de máscara de dirección)

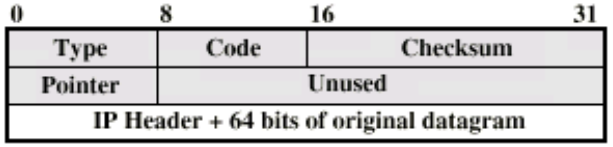
Code	
0	no se puede llegar a la red
1	no se puede llegar al host o aplicación de destino
2	el destino no dispone del protocolo solicitado
3	no se puede llegar al puerto destino o la aplicación destino no está libre
4	se necesita aplicar fragmentación, pero el flag correspondiente indica lo contrario
5	la ruta de origen no es correcta
6	no se conoce la red destino
7	no se conoce el *host* destino
8	el *host* origen está aislado
9	la comunicación con la red destino está prohibida por razones administrativas
10	la comunicación con el *host* destino está prohibida por razones administrativas
11	no se puede llegar a la red destino debido al Tipo de servicio
12	no se puede llegar al *host* destino debido al Tipo de servicio

Protocolos de CONTROL - ICMP

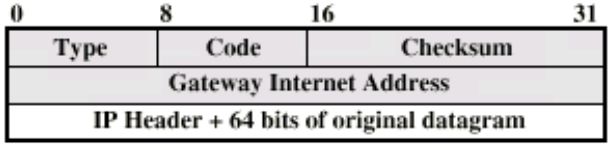


(a) Destination Unreachable; Time Exceeded; Source Quench

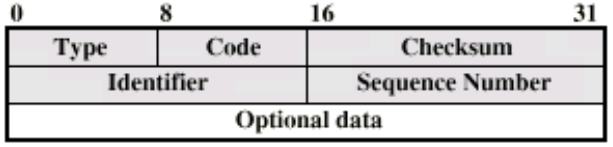
Paquete que ocasionó el error



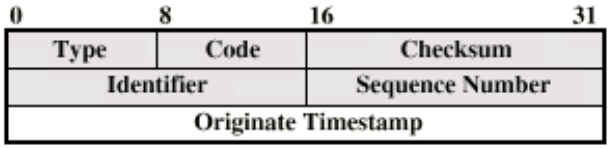
(b) Parameter Problem



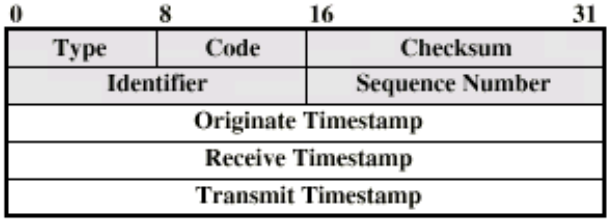
(c) Redirect



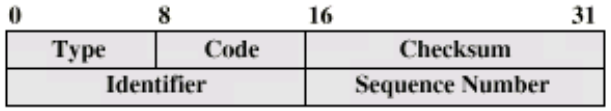
(d) Echo, Echo Reply



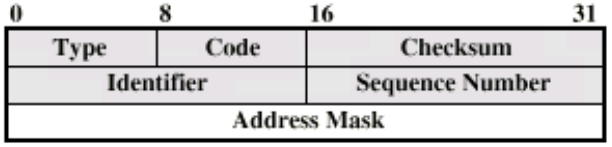
(e) Timestamp



(f) Timestamp Reply



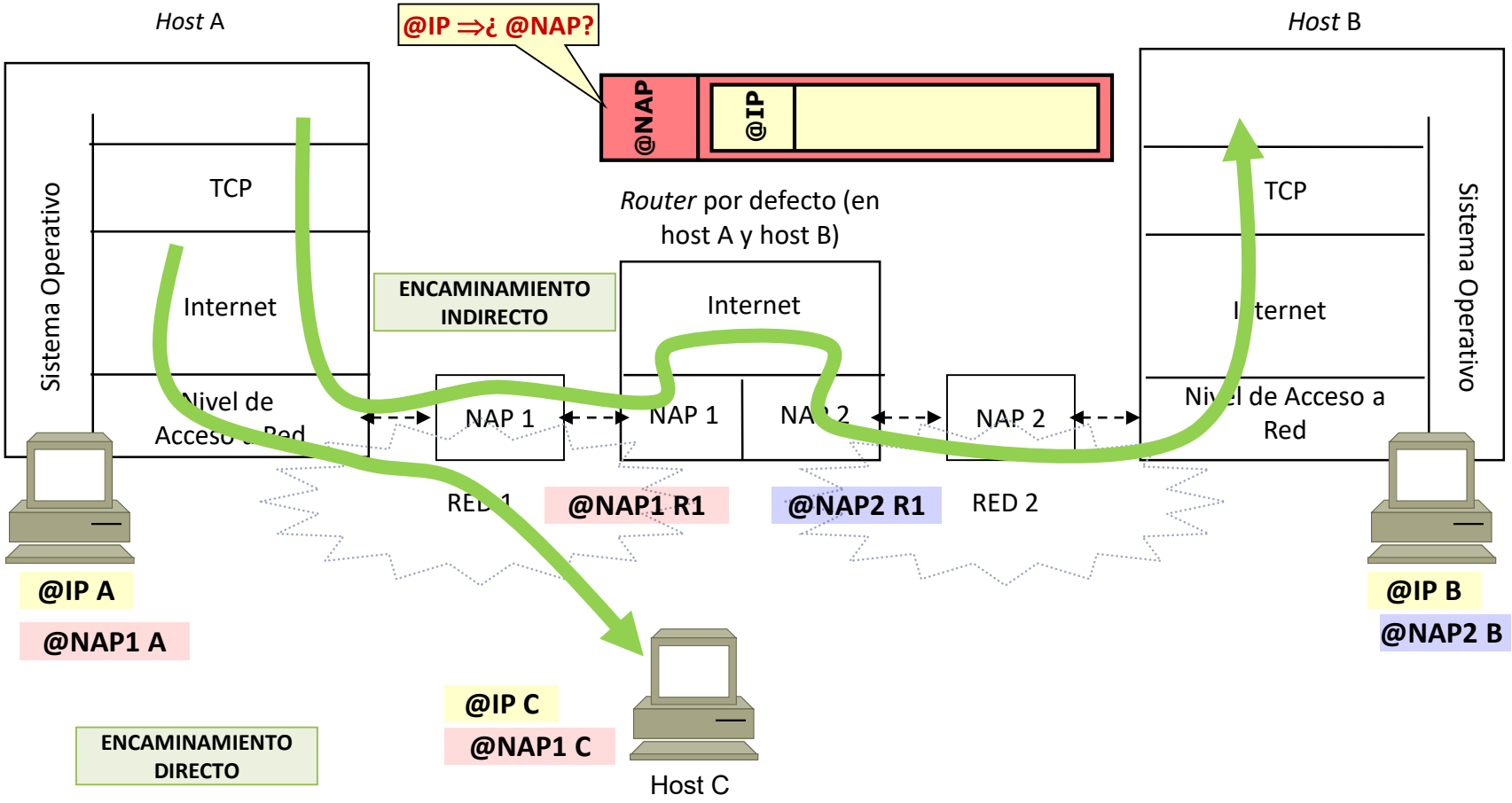
(g) Address Mask Request



(h) Address Mask Reply

Protocolos de CONTROL

- Resolución de direcciones

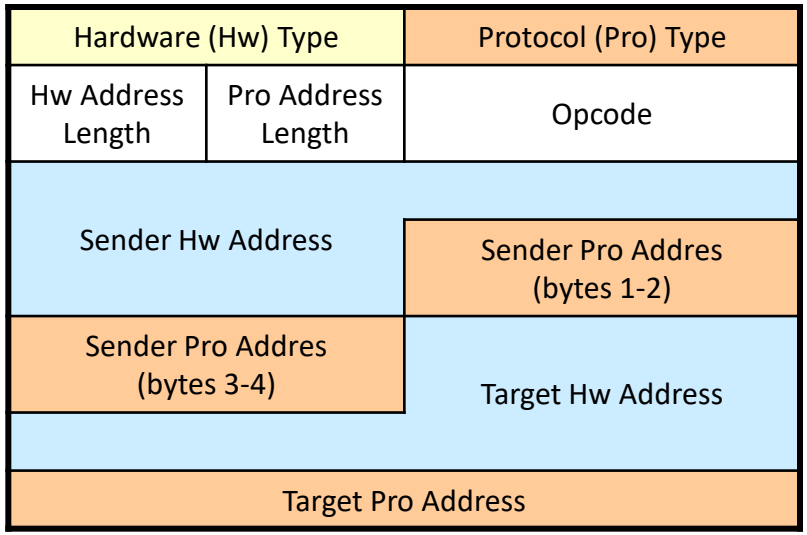


Protocolos de CONTROL

- Resolución de direcciones: soluciones
  - Construir una tabla estática manual de conversión. Ej.: RDSI, X.25, FR, ATM.
  - Crear una tabla dinámica que se mantiene de forma automática en un servidor en el que se registra cada equipo que se conecta a la red. Ej.: ATM.
  - Lanzar una **pregunta *broadcast*** a la red para localizar al propietario de la dirección de red buscada. Solo se puede usar en redes *broadcast*. Ej.: **redes LAN**.

Protocolos de CONTROL - ARP

- ARP: *Address Resolution Protocol*
  - Intercambio de mensajes de petición/respuesta (*Request*/*reply*) para averiguar la @IP de una máquina conociendo su @NAP
    - Válido para algunas tecnologías de acceso



Hw type	Hardware Tyype
1	**Ethernet (10 Mbps)**
6	IEEE 802 Networks
7	ARCNET
15	Frame Relay
16	Asynchronous Transfer Mode (ATM)
17	HDLC
18	Fibre Channel
19	Asynchronous Transfer Mode (ATM)
20	Serial Line

Protocolos de CONTROL - ARP

- ARP: *Address Resolution Protocol*

Hw address type	Especifica el tipo de *hardware* (ej: Ethernet)
Pro address type	Especifica el tipo de protocolo, el mismo que en el campo de tipo EtherType en la cabecera de IEEE 802 (ej: IP)
Hw address length	Especifica la longitud (en bytes) de la dirección *hardware* del paquete. Para IEEE 802.3 e IEEE 802.5 será de 6.
Pro address length	Especifica la longitud (en bytes) de las direcciones del protocolo en el paquete. Para IP será de 4.
Opcode	Especifica si se trata de una petición(1) o una respuesta (2) ARP.
Source/target hw address	Contiene las direcciones física *hardware*. **En IEEE 802.3 (Ethernet) son direcciones de 48 bits**.
Source/target pro address	Contiene las direcciones del protocolo. **En TCP/IP son direcciones IP de 32 bits**. Para el paquete de solicitud, la dirección *hardware* de destino es el único campo indefinido del paquete

Protocolos de CONTROL - ARP

- ARP: *Address Resolution Protocol*
  - Tabla ARP:
    - Evitar envío continuado de *request-reply* para cada paquete IP ⇒ guardar información de asociación ARP-IP
      - Dirección MAC.
      - Tiempo desde el último mensaje ARP.
      - Entrada añadida manualmente o mediante ARP.
      - A qué interfaz pertenece.
    - Añadir entradas: **cuando se recibe un ARP de un nodo conocido** (o para nosotros) ⇒ asociar (@IP fuente - @MAC fuente)
    - Duración limitada:
      - Evitar crecimiento ↑ tabla
      - Evitar uso de información obsoleta
    - Se pueden añadir entradas manualmente
      - Se eliminan al reiniciar
    - Se pueden eliminar entradas manualmente

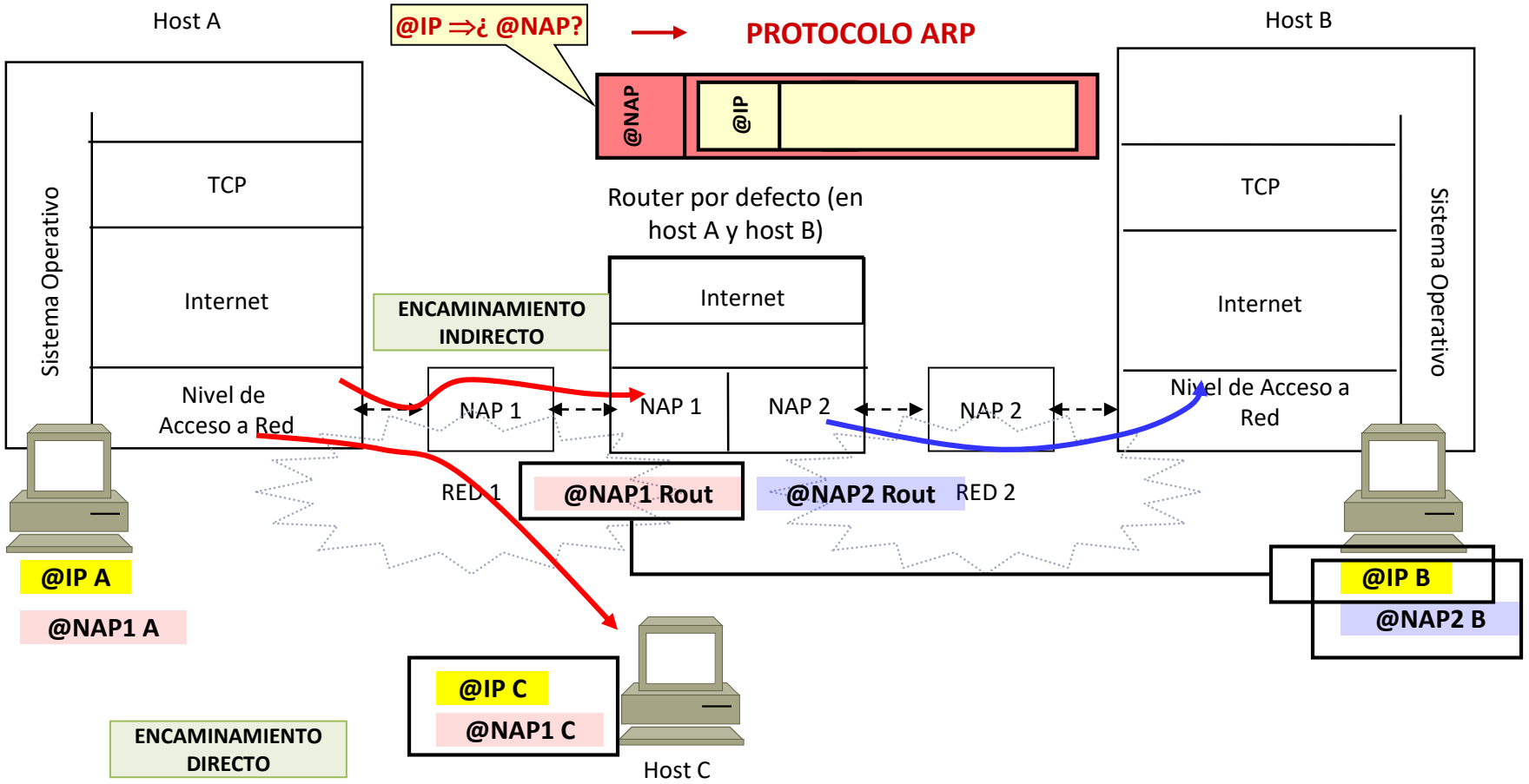
Ojo!, no hacemos caso de cualquier ARP que vemos en la red

arp -a

```
Interfaz: 192.168.1.4 --- 0x3
  Dirección IP          Dirección física      Tipo
  192.168.1.1           00-18-39-8c-bf-f6     dinámico
```

Protocolos de CONTROL - ARP

- Resolución de direcciones



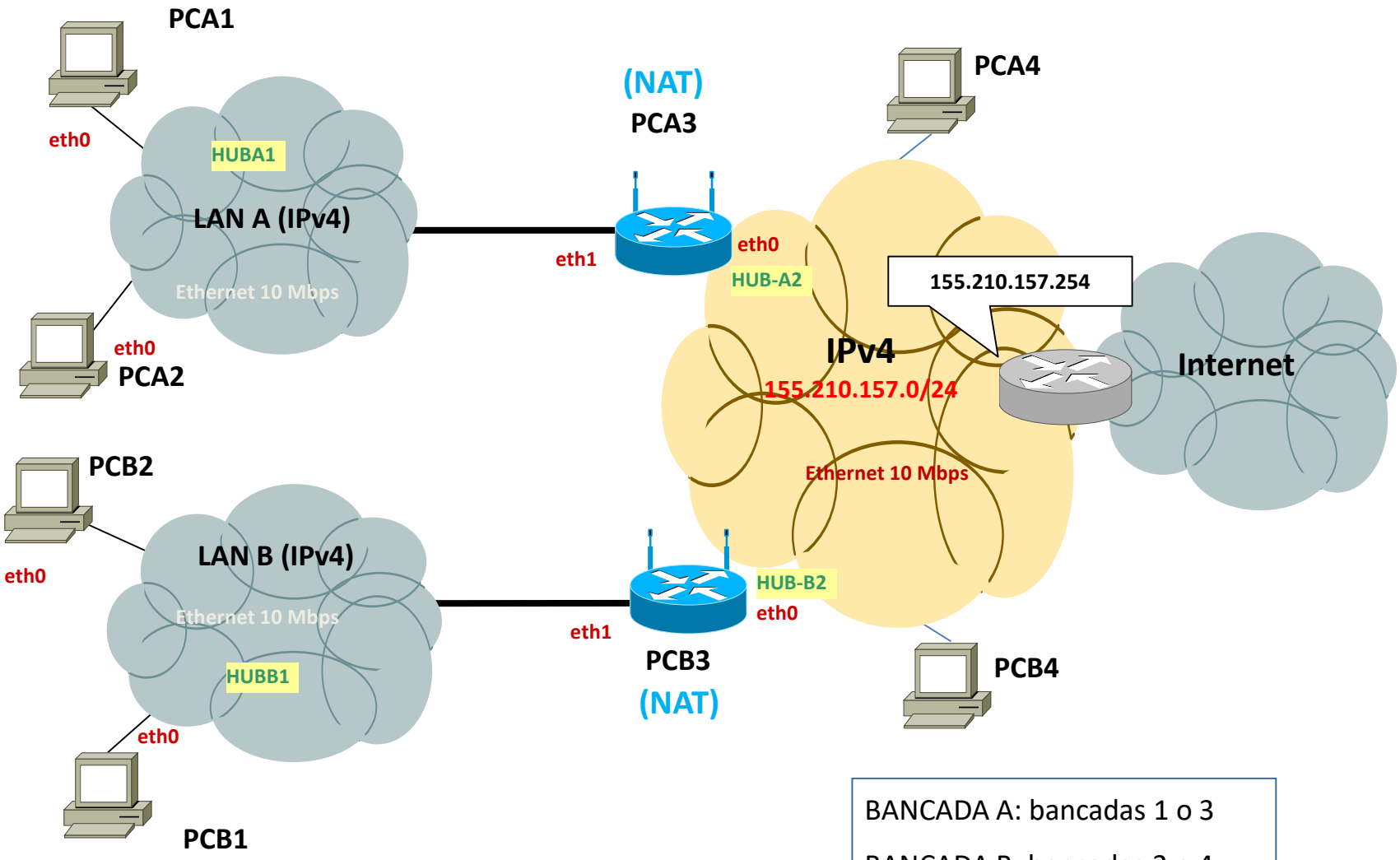
Protocolos de CONTROL - ARP

- ARP: *Address Resolution Protocol* - Casos especiales:
  - **Proxy ARP**
    - Permite que una dirección de red se comparta entre dos redes físicas.
      - Los *host* pertenecientes a 2 redes físicas distintas (*router* R separa) "creen estar en la misma" (misma IP) y usan ARP como encaminamiento directo:
      - El *router* R contesta a las preguntas ARP y así recibe el tráfico entre redes y redirige
        - » Por defecto, nadie responde ARP a una @IP que no es la suya: hay que habilitar la función proxy ARP
    - Utilización:
      - Una red crece y se decide dividirla en dos, pero el *software* de red de los *host* no soporta este cambio.
      - Gestión de movilidad IP (Mobile IP)
  - **ARP gratuito**
    - Las estaciones de trabajo envían una solicitud ARP preguntando sobre su propia dirección IP con el propósito de detectar direcciones IP duplicadas y forzar a que todas las estaciones de trabajo actualicen la entrada de las direcciones correspondientes.
    - Utilización:
      - Permiten detectar direcciones IP en conflicto (ej. DHCP)
      - Ayudan a actualizar las tablas ARP de otras máquinas (gestión de movilidad)
      - Informan a los *switch* de la dirección MAC de una máquina conectada a un puerto (aprendizaje)

Protocolos de CONTROL - ARP

- ARP: *Address Resolution Protocol* - Casos especiales:
  - **Proxy ARP**
    - Ventajas:
      - Fácil de incorporar a un *router* sin modificar los demás (y sus tablas de encaminamiento)
      - Debería usarse donde los *host* no puedan configurarse con un *router* por defecto (no "inteligencia de encaminamiento")
    - Desventajas
      - *Host* desconocen la topología física real y usan ARP, lo que implica:
        - » Incremento del tráfico ARP en el segmento de red
        - » Tablas ARP mayores
        - » Riesgo de ARP *spoofing*
        - » No funciona en redes que no usan resolución mediante ARP
        - » No es generalizable para todo tipo de topologías de red (por ejemplo, más de dos)

Protocolos de CONTROL



Protocolos de CONTROL – Configuración IP

- Arranque del sistema (*bootstraping*) – configuración de red:
  - Proceso por el cual se inicia una computadora y en la cual se carga una imagen del SO:
    - El ordenador carga un programa sencillo de arranque (*boot*)
  - ¿Y la configuración de red (@IP, máscaras, servidor DNS, puerta de enlace…)?
    - SO con disco suelen tener almacenado esta información en ficheros del SO
    - Puede ser necesario **obtener la información de un servidor externo**

Sobre nivel de acceso a red

RARP (*Reverse Address Resolution Protocol*)

- El servidor RARP ha de estar en la misma LAN que el cliente.
- RARP permite encontrar ÚNICAMENTE LA DIRECCIÓN IP (dada una dirección física) de manera ESTÁTICA

Sobre TCP/IP

BOOTP (*BOOTstrap Protocol*)

- El servidor y el cliente pueden estar en LANs diferentes (encaminamiento IP)
- Permite suministrar TODOS LOS PARÁMETROS de configuración al cliente.
- La asignación de direcciones es ESTÁTICA

Extensión de BOOTP →**DHCP (*Dynamic Host Configuration Protocol*)**

- Permite ADEMÁS una asignación DINÁMICA

Protocolos de CONTROL – DHCP

- **DHCP (*Dynamic Host Configuration Protocol*)** **Sobre UDP *client port* (68) - *server port* (67)**
  - Extensión de BOOTP (compatible!) pero no "da", sino "alquila" las direcciones IP: **RFC 2131** **RFC 2132**
    - **Manual**: asignación de una dirección especifica para una máquina específica (equivale a BOOTP).
    - **Automático**: permite asignar direcciones permanentes (también estático).
    - **Dinámico**: asigna bajo demanda una dirección de un pool durante un tiempo limitado negociado (*lease time*)
  - Es lo más parecido a la autoconfiguración (*plug-and-play*)
- Funcionamiento del protocolo
  - Dos pasos
    - Descubrimiento de servidor DHCP
    - Diálogo cliente – servidor para negociar la asignación IP

    Sin configuración IP: mensajes con @IP dest = 255.255.255.255 y @IP src = 0.0.0.0.
  - DHCP *relay*: si el servidor está en otra LAN
    - *Relay*, recibe el *broadcast* del cliente y lo reenvía *unicast* al servidor
    - *Relay* recibe respuesta *unicast* del servidor y lo reenvía *broadcast* al cliente
    - *Relay* usa 'giaddr' para poner su dirección (@IP del interfaz del cliente)

    Relay* ≠ *Forward
    Relay ⇒ "capturar" el paquete, modificar (ej. Cambio IP fuente) y reenviar
  - DHCP usa caches para sistemas con disco y así optimizar las peticiones cuando rebota
- Problemas DHCP con DNS:
  - Cambio de @IP (misma u otra red) ⇒ no puede mantener su nombre y dominio
    - Solución ⇒ DDNS (*Dynamic* DNS)

Protocolos de CONTROL – DHCP

- **DHCP (*Dynamic Host Configuration Protocol*)**
  - Lista de opciones configurables:
    - Dirección del servidor DNS
    - Nombre DNS
    - Puerta de enlace de la dirección IP
    - Dirección de Publicación Masiva (*broadcast address*)
    - Máscara de subred
    - Tiempo máximo de espera del ARP (Protocolo de Resolución de Direcciones según siglas en inglés)
    - MTU (Unidad de Transferencia Máxima según siglas en inglés) para la interfaz
    - Servidores adicionales
      - Servidores y dominios NIS (Servicio de Información de Red según siglas en inglés)
      - Servidores NTP (Protocolo de Tiempo de Red según siglas en inglés))
      - Servidor SMTP
      - Servidor TFTP
      - Nombre del servidor WINS

Protocolos de CONTROL – DHCP

- **DHCP (*Dynamic Host Configuration Protocol*)**

RFC 2131

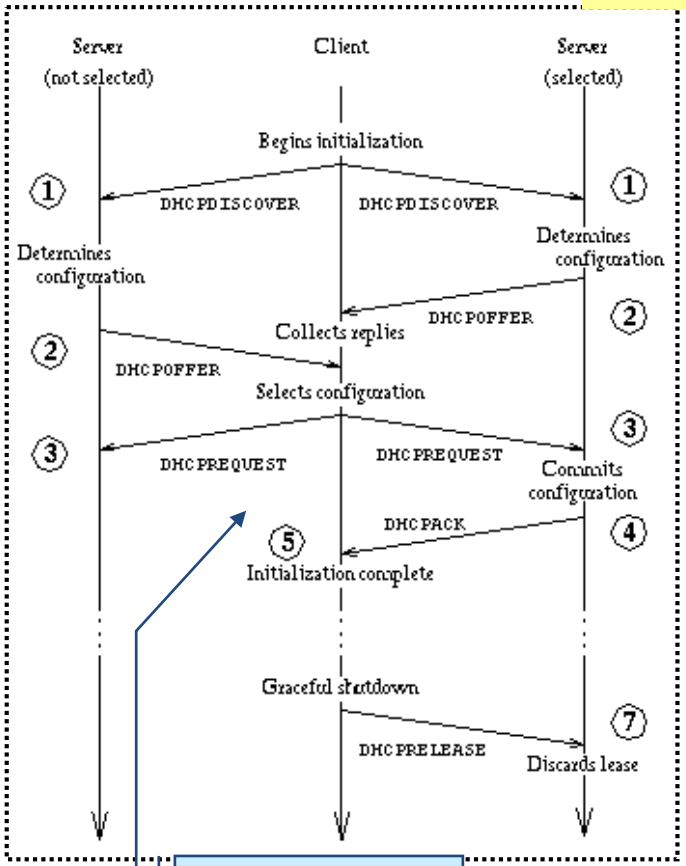
BROADCAST
1. Cliente DHCP → Servidor DHCP: **DhcpDiscover**.
   - Cliente puede especificar preferencias (*lease time*, @IP…)

BROADCAST
UNICAST
2. Un servidor DHCP → cliente **DhcpOffer**
   - Indica @IP disponible: campo 'yiaddr'
   - Todavía no hay acuerdo !!!
3. Client recibe 1 o + DhcpOffer':
   - Selección de uno: toma @IP del servidor

BROADCAST
4. Cliente → Servidor DHCP elegido: **DhcpRequest**
   - Opción DHCP *identifier*: @IP antes guardada (del servidor)

BROADCAST
UNICAST
5. Servidor verifica ¿DCHP *Identifier* = su @IP?: sí ⇒
   - **DhcpAck** (@IP disponible)
   - DhcpNack (@IP en uso)
6. Cliente recibe
   - DhcpAck: puede usar @IP
     - Si hay algún problema con esa IP: Cliente → Servidor DHCP DhcpDecline (volver a 1)
   - DhcpNak: volver al paso 1

UNICAST
7. Cliente → Servidor: **DhcpRelease**
   1. Liberar la @IP antes del fin del "lease time"



Contenidos

- Repaso Protocolo Internet (IPv4)
  - Internet
  - Direccionamiento
  - Encaminamiento
  - Funcionalidad del Protocolo IPv4.
    -PDU - Fragmentación y reensamblado.
  - NAT y NAPT.
- Protocolos de Encaminamiento
  - RIP y OSPF
- Funciones de control: apoyo en otros protocolos
- **Gestión de redes TCP/IP: arquitectura SNMP**

Gestión de redes TCP/IP: arquitectura SNMP

- Introducción
- Conceptos generales del modelo SNMP
- Información de gestión SNMP

SNMP. Introducción

La gestión de redes en TCP/IP

La primera herramienta para gestión de Internet fue **ICMP** (*Internet Control Message Protocol*):

– Transferencia de **mensajes de control** entre los *router* y los *host* que soportan IP.

– Se puede comprobar la conectividad (*echo/echo-reply)* y estimar retardos y fiabilidad (*time stamp/time stamp-reply*) de una ruta en la red.

– Programa **PING** (*Packet Internet Groper*) desarrollado a partir de ICMP. Programa **Traceroute.**

En 1987 se propone **SGMP** (*Simple Gateway Monitoring Protocol*) para monitorizar router.

SNMP. Introducción

Surgieron tres aproximaciones al problema de gestión:

- **High-Level Entity Management System (HEMS):** surgió como una generalización del Host Monitoring Protocol (HMP).
- **Simple Network Management Protocol (SNMP):** nació como una versión mejorada de SGMP.
- **CMIP over TCP/IP (CMOT):** donde se intentó integrar Common Management Information Protocol (CMIP) y los servicios y estructuras de bases de datos de ISO.

SNMP. Introducción

- En 1988, se propone **SNMP** (Simple Network Management Protocol), versión extendida de SGMP y destinada a su utilización a corto plazo. SNMP ha evolucionado en paralelo a TCP/IP.

- Posteriormente apareció **RMON** (Remote Monitor), que permite monitorizar globalmente una subred, no sólo sus dispositivos.

- Se han propuesto diversas **extensiones** (estándar y privadas) a la MIB de SNMP. Algunas deficiencias de SNMP se intentan resolver con SNMPv2 y SNMPv3

Conceptos generales del modelo SNMP

Elementos del modelo SNMP:

- Estación gestora
- Agente
- Base de información de gestión (MIB)
- Protocolo de gestión de red

Conceptos generales del modelo SNMP

ESTACIÓN GESTORA:

- Sirve como interfaz con el gestor humano.

- Contiene:

  - conjunto de aplicaciones de gestión para análisis de datos, recuperación de fallos, etc.
  - interfaz para monitorización y control de la red.
  - traducción de las órdenes del gestor a los dispositivos de monitorización y control remotos.
  - base de datos de información extraída de las MIB de todas las entidades gestionadas.

Conceptos generales del modelo SNMP

AGENTE:

- Los dispositivos de la red (*host*, *hub*, puentes y *router*) pueden tener agentes SNMP.
- Responden a órdenes de la estación gestora y de forma asíncrona le envían información importante no solicitada (*traps*).

Conceptos generales del modelo SNMP

BASE DE INFORMACIÓN DE GESTIÓN (MIB):

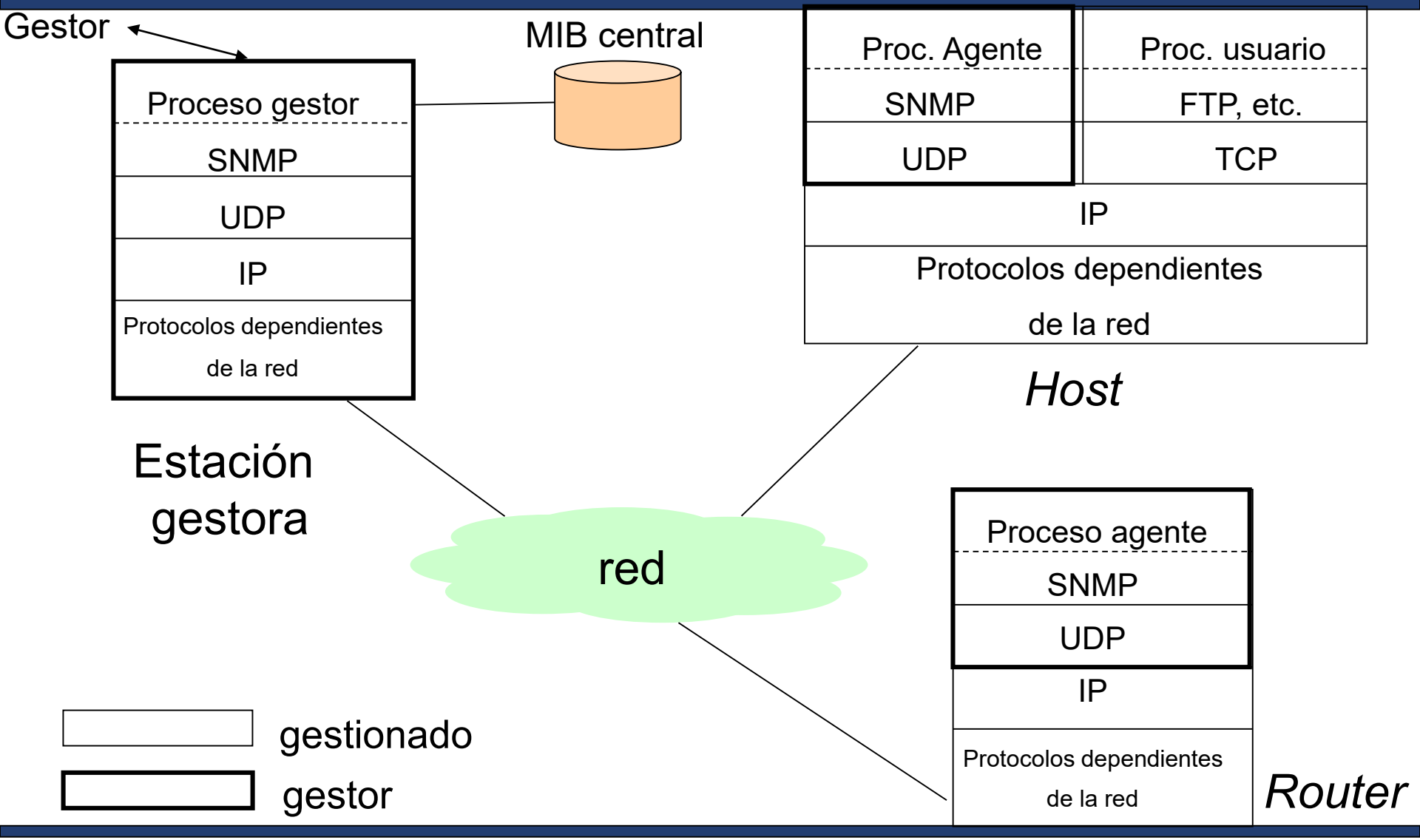
- Los recursos de la red a gestionar se representan mediante objetos.
- Colección de objetos = MIB.
- Los MIB forman un conjunto de puntos de acceso para la estación gestora.
- Los objetos están estandarizados para una clase concreta (p.e., un mismo tipo de objetos se usa para gestionar varios *router*).

Conceptos generales del modelo SNMP

PROTOCOLO DE GESTIÓN DE RED

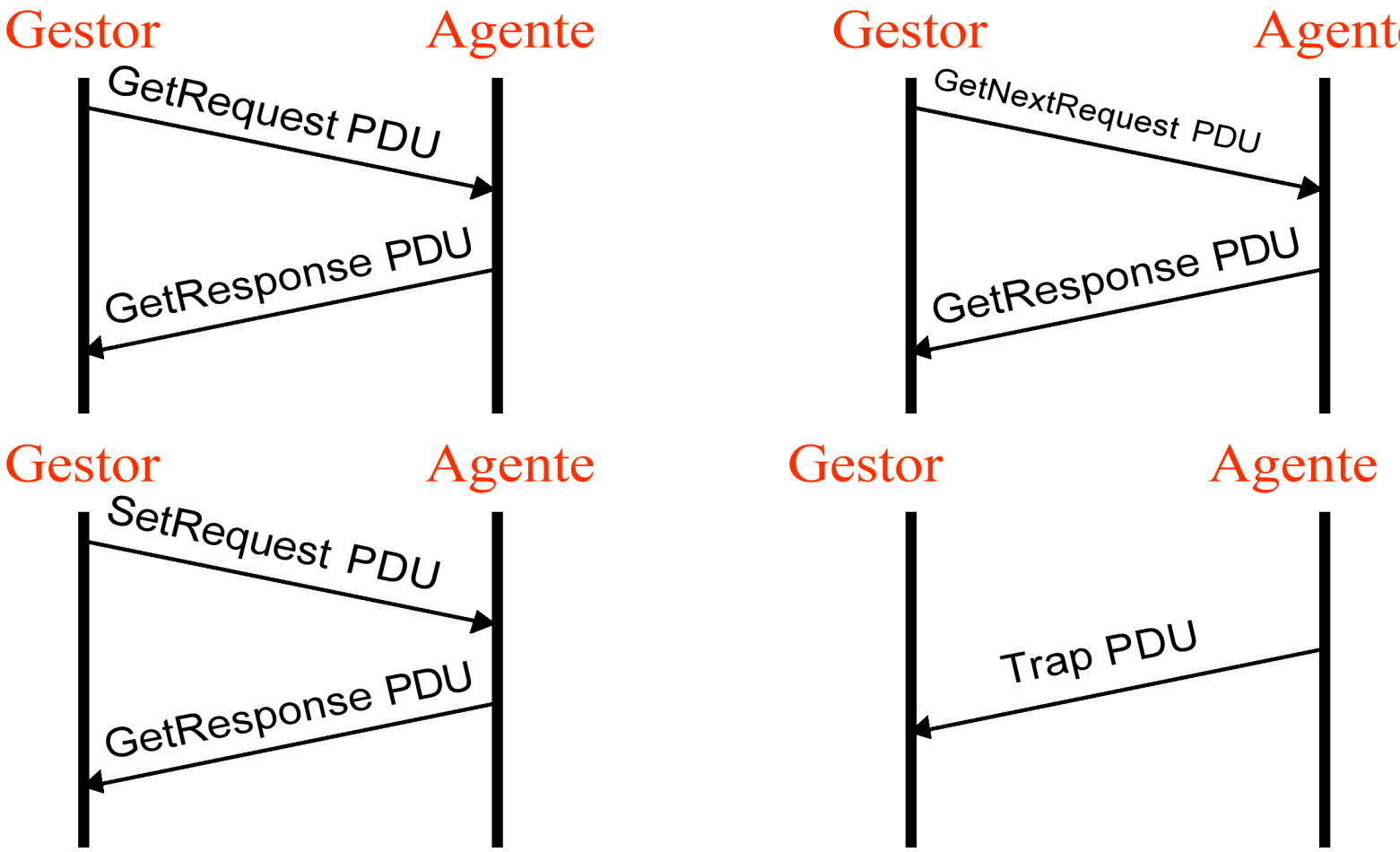
- La estación gestora y los agentes gestores se comunican utilizando un protocolo.
- En las redes TCP/IP este protocolo es SNMP. SNMP es un protocolo de nivel de aplicación dentro del modelo TCP/IP.
- SNMP funciona sobre UDP. Se trata por tanto de un protocolo no orientado a conexión.
- Incluye las siguientes funciones:
  - ***Get***: la estación gestora extrae el valor de un objeto del agente.
  - ***Set:*** la estación gestora fija el valor de un objeto del agente
  - ***Trap:*** permite a un agente notificar a la estación gestora eventos significativos

Conceptos generales del modelo SNMP



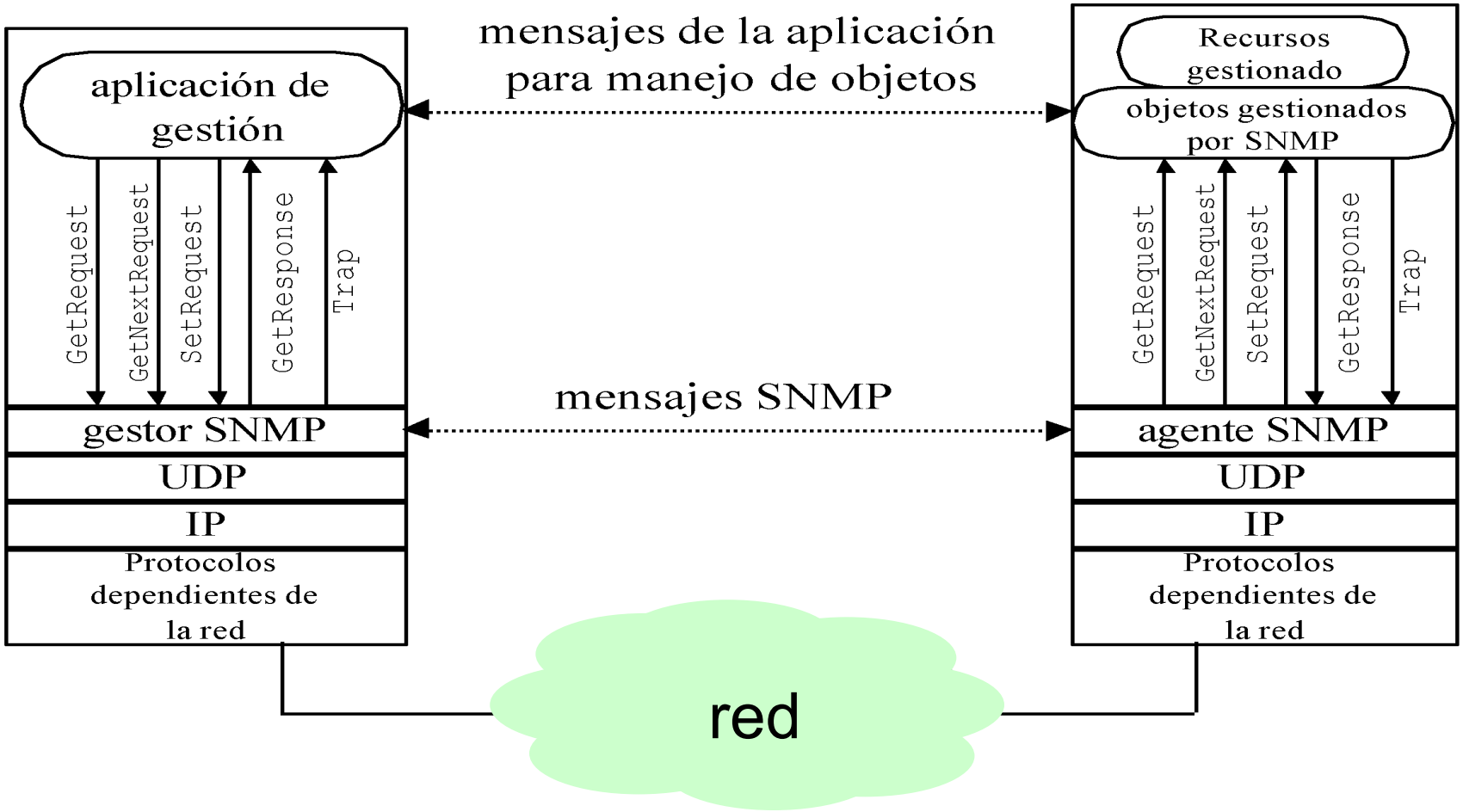
Conceptos generales del modelo SNMP

Mensajes SNMP



Conceptos generales del modelo SNMP

Funcionamiento SNMP



Conceptos generales del modelo SNMP

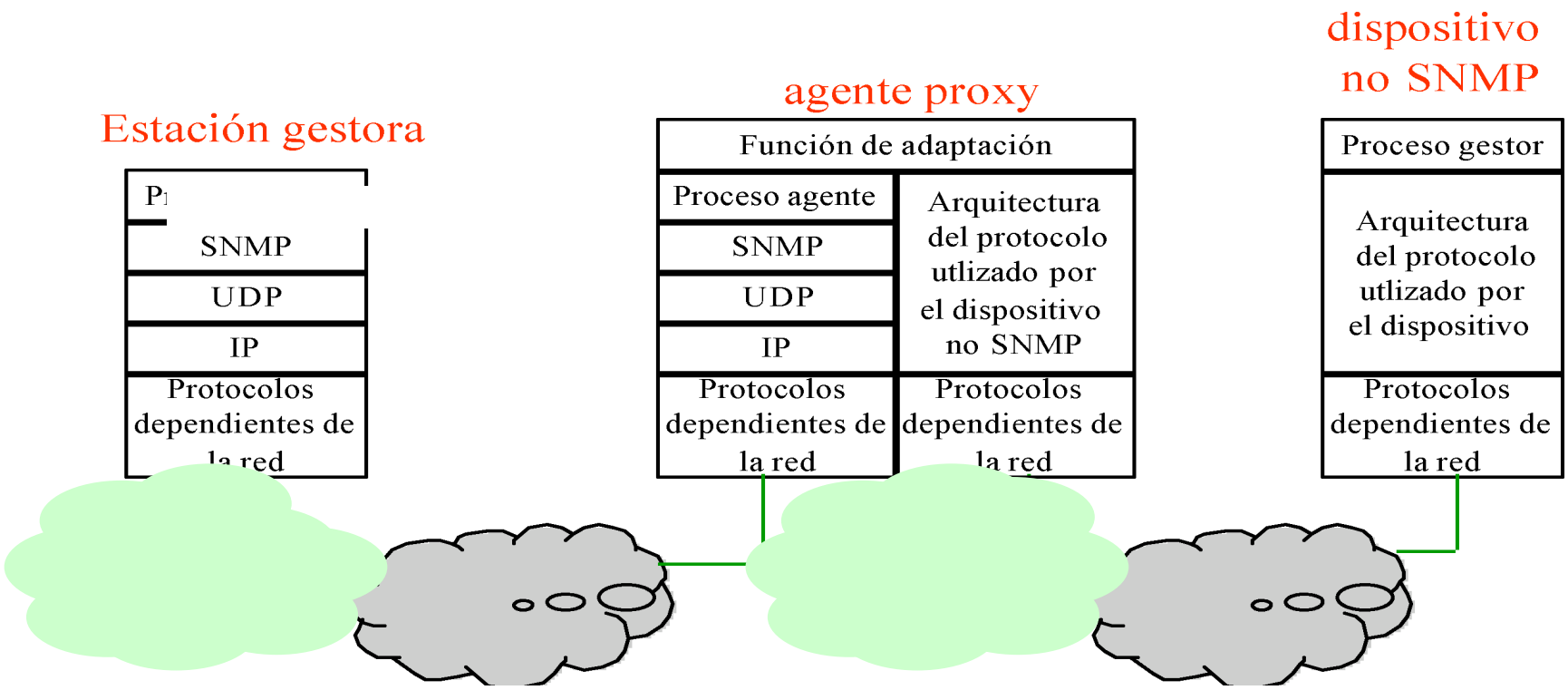
TRAP-DIRECTED POLLING

- Si una estación gestora es responsable de un gran número de agentes y cada agente mantiene un gran número de objetos, entonces no es práctico que la estación gestora sondee todos sus agentes y todos sus objetos.
- Para evitar este inconveniente se utiliza el ***trap-directed polling***.
- La estación gestora realiza un **sondeo inicial** y a **intervalos** poco frecuentes (un día) para recoger información clave (características del interfaz y estadísticas básicas).
- Cada agente notifica a la estación gestora los eventos no usuales (p.e., el agente cae y se reactiva, un enlace falla, condición de sobrecarga, ...).
- La comunicación de estos eventos se realiza mediante ***traps***.
- Una vez alertada, la estación gestora puede iniciar una consulta sobre el agente para diagnosticar el problema.

Conceptos generales del modelo SNMP

PROXIES

- Algunos puentes, módems, etc. no soportan los protocolos TCP/IP.
- Algunos sistemas pequeños (PCs, controladores programables) tienen TCP/IP para sus aplicaciones, pero no se quiere cargarlos con SNMP, el agente y el mantenimiento del MIB.



Información de gestión SNMP

- La gestión en TCP/IP (al igual que en OSI) se basa en el manejo de una “base de datos” (MIB) que contiene la información de los elementos a gestionar.
- Cada recurso gestionable es un **objeto**.
- La MIB es una colección estructurada de objetos en forma de **árbol.**
- Cada sistema (servidor, router, etc) mantiene una MIB con información sobre los recursos del mismo.
- La estación gestora **monitoriza** los recursos de la red *leyendo* los valores de los objetos en la MIB y puede **controlar** los recursos *modificando* esos valores.

Información de gestión SNMP

- El objeto u objetos usados para representar un recurso particular debe ser el mismo en todos los sistemas.

  Ejemplo:
  El número de conexiones TCP abiertas en un periodo:

  conexiones abiertas = conexiones pasivas + conexiones activas

- Se debe utilizar un esquema de representación común que permita la interoperabilidad de todos los elementos.

  Para ello se debe definir una estructura de información de gestión (**SMI**).

Información de gestión SNMP

- La estructura de información de gestión o SMI (Structure of Management Information), especificada en RFC1155, define el marco general en que una MIB puede ser definida y construida.
- La SMI identifica los **tipos de datos** que pueden ser usados en la MIB y especifica cómo los recursos dentro de ella se representan y nombran.
- La filosofía utilizada en la SMI consiste en proporcionar **simplicidad** y **escalabilidad** a la MIB.

Información de gestión SNMP

- Una MIB sólo puede almacenar datos de dos tipos:
  - **escalares**
  - **matrices bidimensionales** de escalares

- SNMP sólo puede recuperar de una MIB **escalares** o **entradas individuales** en una tabla.

- SMI no soporta creación o recuperación de estructuras de datos complejas. En la gestión OSI se tiene mayor funcionalidad.

Información de gestión SNMP

SMI: definición de la estructura de información

Para proveer una forma estándar de representar la información de gestión, la SMI debe:

Proveer una técnica estándar de definición de la estructura de la MIB.

Proveer una técnica estándar de definición de los objetos individuales (sintaxis/valor).

Proveer una técnica estándar para codificar los valores de los objetos.

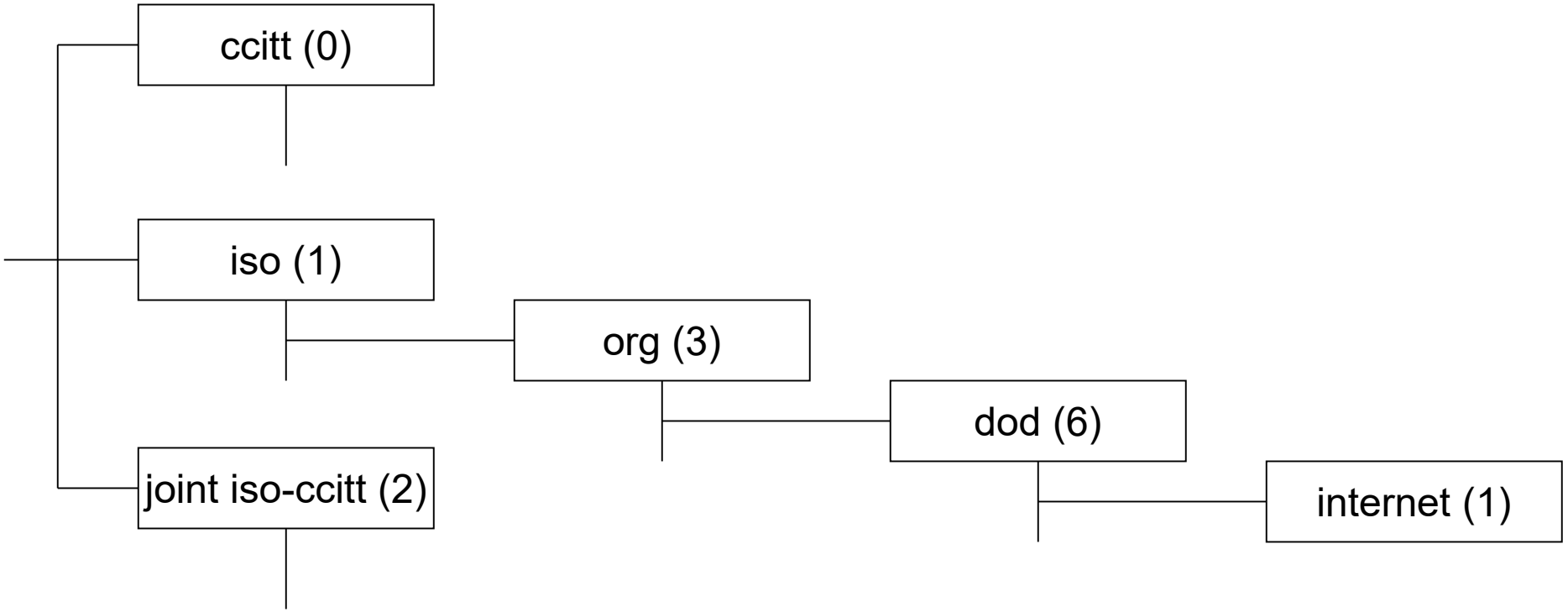
Información de gestión SNMP

Estructura de la MIB

- Los objetos gestionados en SNMP se estructuran en forma de **árbol** organizados según jerarquías.
- Cada **objeto** representa un recurso, una actividad o una información relacionada.
- Dentro de la estructura del árbol los objetos se agrupan en **grupos** relacionados: grupos de objetos IP, grupos de objetos TCP, ...
- A cada objeto de una MIB se le asocia un identificador del tipo ASN.1 **OBJECT IDENTIFIER**.
- El identificador único permite nombrar al objeto.

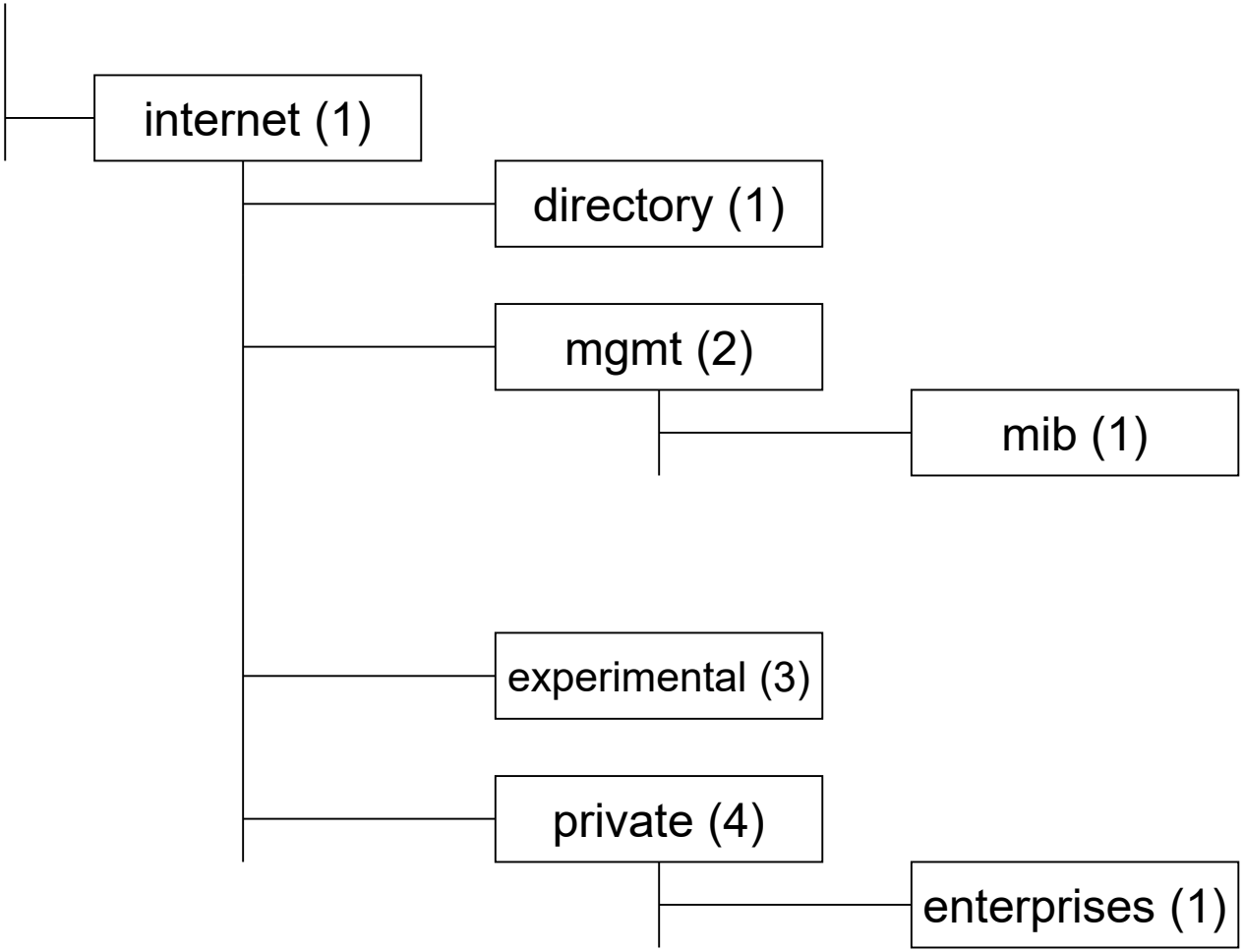
Información de gestión SNMP

Estructura de la MIB



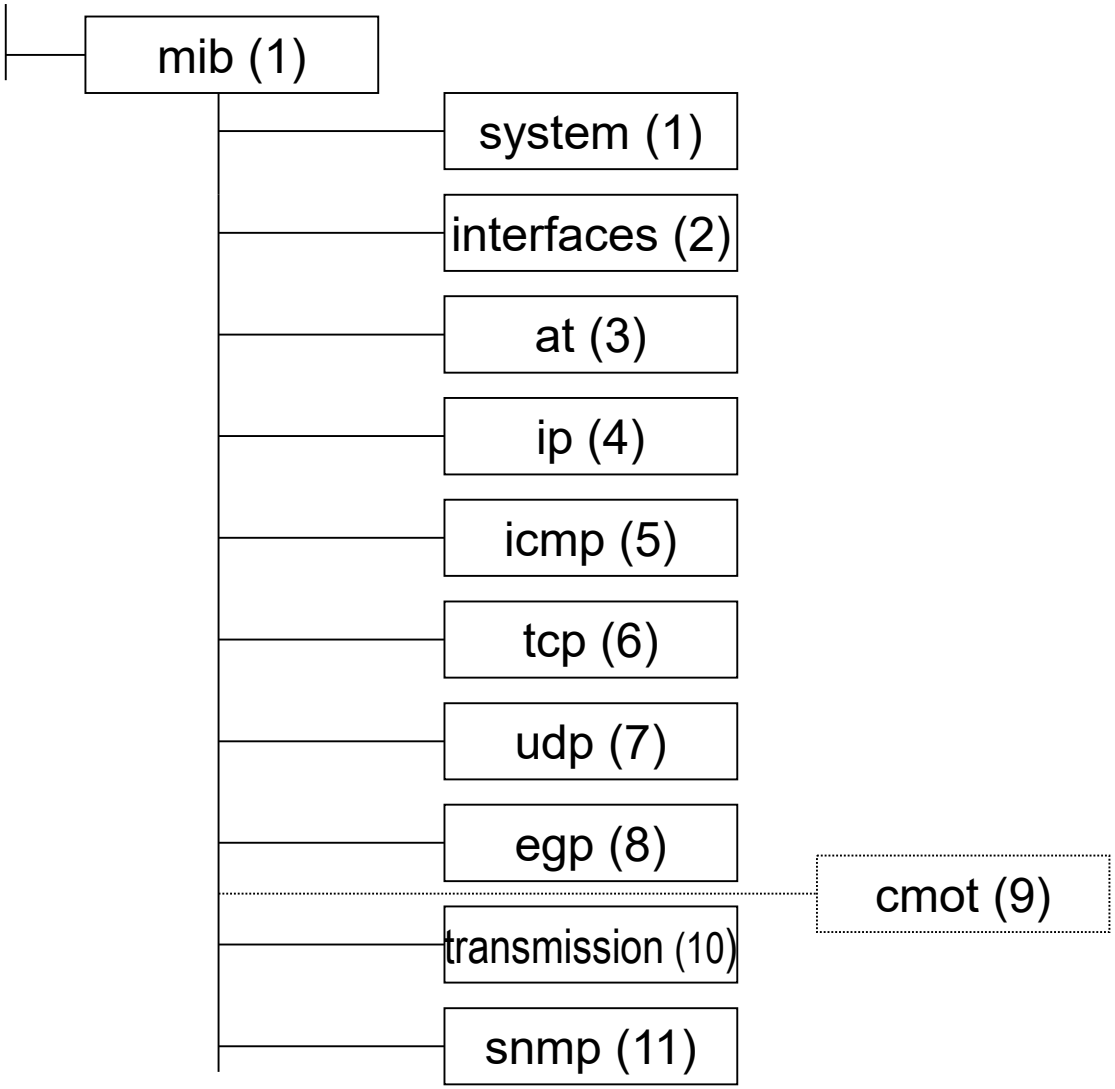
Información de gestión SNMP

Estructura de la MIB



Información de gestión SNMP

Estructura de la MIB



SNMP sobre Linux

El formato de instrucción en Linux para ejecutar comandos SNMP es el siguiente:

comando @IP_entidad community object_instance_ID

Ejemplo:

snmpget 155.210.157.5 public system.sysDescr.0
system.sysDescr.0 = 3Com SuperStack II

SNMP sobre Linux

COMANDOS SNMP BAJO LINUX

Las siguientes funciones permiten establecer comunicaciones a través de SNMP entre un gestor y un agente en el entorno Linux.

snmpd: "*daemon*" que ejecuta el agente SNMP para permitir comunicarse al PC a través de mensajes del protocolo.

snmpget: comando que ejecuta la función *get* de SNMP de forma que la estación gestora extrae el valor de un objeto del agente.

snmpset: comando que ejecuta la función *set* de SNMP de forma que la estación gestora fija el valor de un objeto del agente.

SNMP sobre Linux

COMANDOS SNMP BAJO LINUX

***snmpgetnext*:** comando que ejecuta la función *get* de SNMP de forma que la estación gestora extrae el valor de un objeto del agente, que es el siguiente en el orden lexicográfico de la estructura de la base de información.

***snmptrap*:** comando que ejecuta la función *trap* de SNMP de forma que permite a un agente notificar a la estación gestora eventos significativos, mediante el envío de un *trap*.

***snmptrapd*:** "*daemon*" que activa la recepción de *trap*.

SNMP sobre Linux

***snmptranslate*:** comando que permite traducir información de los objetos de la base de información (conversión entre valores SMI y forma simbólica), mostrar la descripción de los objetos, etc.

***snmpwalk*:** comando basado en la función snmpgetnext que obtiene información de objetos de la base de información.

***snmpnetstat*:** comando que muestra el estado de la red utilizando SNMP.

Además de estas funciones, hay otros comandos de uso menos frecuente: snmptest, snmpdelta, snmptable, snmpstatus, etc.