TEMA 2
CAPA DE RED

Fundamentos de Redes

2021/2022

➢Bibliografía Básica:

Capítulos 6 Y 9, Pedro García Teodoro, Jesús Díaz Verdejo y Juan Manuel López Soler. ***TRANSMISIÓN DE DATOS Y REDES DE COMPUTADORES***, Ed. Pearson, 2014, ISBN**:** 978-0-273-76896-8

➢Para saber más…

Capítulo 4 James F. Kurose y Keith W. Ross. ***COMPUTER NETWORKING. A TOP-DOWN APPROACH***, 5ª Edición, Addison-Wesley, 2010, ISBN: 9780136079675.

Tema 2. CAPA DE RED

1. **Introducción**
2. Conmutación
3. El protocolo IP
4. Asociación con capa de enlace: el protocolo ARP
5. El protocolo ICMP
6. Autoconfiguración de la capa de red (DHCP)

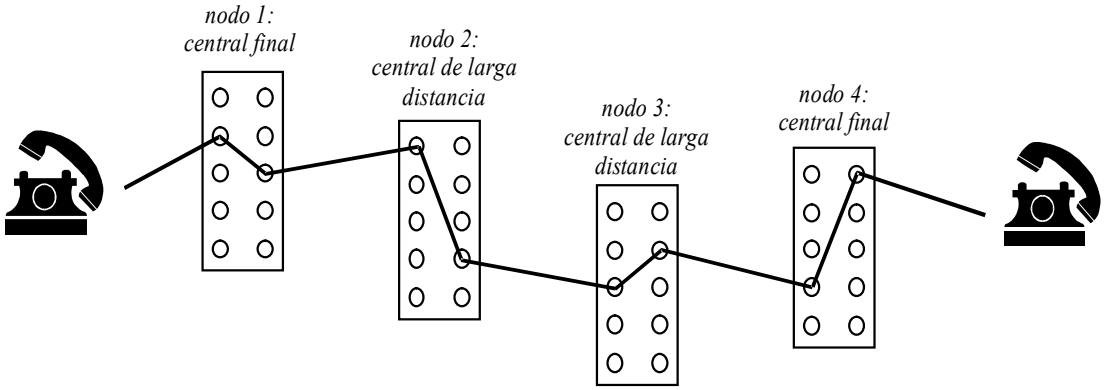
- Funciones y servicios de la capa de red en TCP/IP
  - El objetivo de la capa de red en Internet es **la interconexión de redes**, con independencia de la tecnología subyacente!!
  - **Conmutación**: acción de cursar tráfico entre los nodos de la red
  - **Encaminamiento**: encontrar la mejor ruta hasta el destino
  - En el modelo OSI el control de congestión se realiza en esta capa
- Ejemplos de protocolos de red:
  - X.25 https://es.wikipedia.org/wiki/Norma_X.25
  - IP

Tema 2. CAPA DE RED

1. Introducción
2. **Conmutación**
3. El protocolo IP
4. Asociación con capa de enlace: el protocolo ARP
5. El protocolo ICMP
6. Autoconfiguración de la capa de red (DHCP)

2. CONMUTACIÓN

- Conmutación = acción de cursar tráfico para establecer o determinar un camino que permita transmitir información extremo a extremo
- Diferentes tecnologías de conmutación
  - Conmutación de Circuitos
  - Conmutación de Paquetes: datagramas o circuitos virtuales
- **Conmutación de circuitos**
  - Ej. Teléfono
  - Es un servicio orientado a conexión → exige un establecimiento de conexión previo a la transmisión



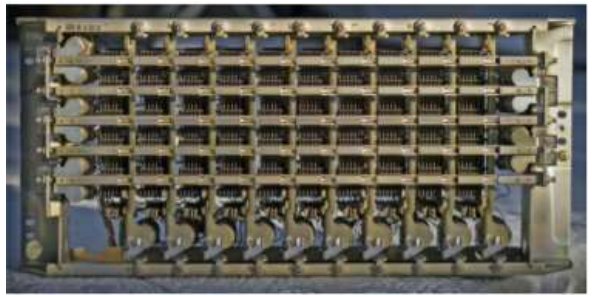
  - Pasos: (i) Conexión, (ii) Transmisión, (iii) Desconexión
  - Recursos dedicados. Facilita comunicaciones tiempo-real. No hay contención (contienda por acceder al medio) .
  - Retraso para establecimiento de la llamada. Poca flexibilidad para adaptarse a cambios. Poco tolerante a fallos.

- Conmutación de circuitos

Ventajas

- **La transmisión se realiza en tiempo real**, adecuado para voz
- **Uso permanente de recursos,** el circuito se mantiene durante toda la sesión
- **No hay contención**, no hay contienda para acceder al medio
- **El circuito es fijo**, no hay decisiones de encaminamiento una vez establecido
- **Simplicidad en la gestión de los nodos intermedios**.

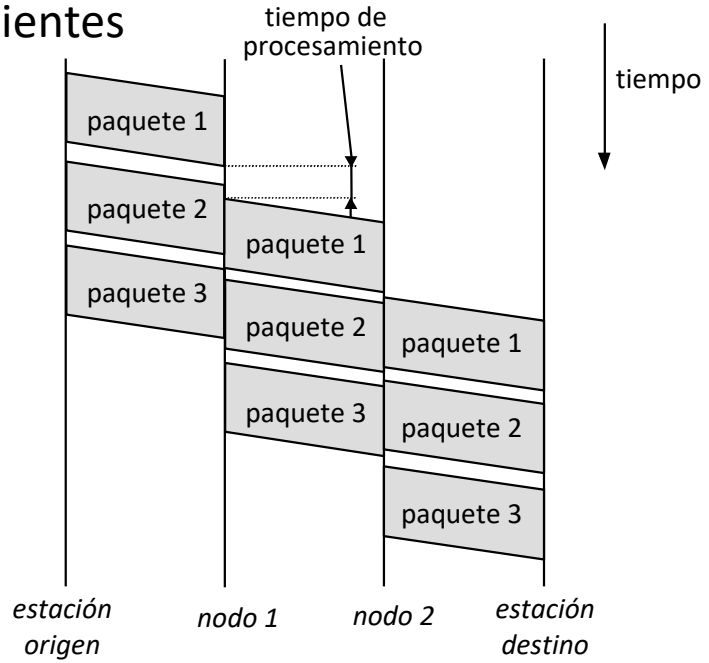
Desventajas



- **Retraso en el inicio de la comunicación**.
- **En ocasiones uso no eficiente de recursos**.
- **El circuito es fijo.** No se reajusta la ruta de comunicación.

- MÁS INFORMACIÓN: Cap. 10 del libro de W. Stallings, 7ª Edición.

- **Conmutación de paquetes (datagramas)**:
  - Envío en unidades de datos (paquetes) independientes
  - No hay conexión
  - En cada salto: almacenamiento y re-envío
  - Cada paquete debe contener en su cabecera las direcciones origen y destino
  - Los paquetes pueden seguir rutas diferentes y pueden llegar desordenados



- **Conmutación de paquetes con circuitos virtuales**:
  - Usado en redes ATM (tecnología en desuso para redes troncales)
  - Orientado a conexión. Pasos: (i) Conexión, (ii) Transmisión, (iii) Desconexión
  - No hay asignación de recursos como en conmutación de circuitos

➢ Ejemplo de ejercicio a realizar de la lista de problemas resueltos

Estime el tiempo involucrado en la transmisión de un mensaje de datos para la técnica de conmutación de paquetes mediante datagramas considerando los siguientes parámetros:

M: longitud en bits del mensaje a enviar.

V: velocidad de transmisión de las líneas en bps.

P: longitud en bits de los paquetes.

H: bits de cabecera de los paquetes.

N: número de nodos intermedios entre las estaciones finales.

D: tiempo de procesamiento en segundos en cada nodo.

R: retardo de propagación, en segundos, asociado a cada enlace.

Tema 2. CAPA DE RED

1. Introducción
2. Conmutación
3. **El protocolo IP**
4. Asociación con capa de enlace: el protocolo ARP
5. El protocolo ICMP
6. Autoconfiguración de la capa de red (DHCP)

- IPv4 está especificado en el RFC 791 (1349, 2474, 6864)
  - Es un protocolo para la **interconexión** de redes (también llamadas subredes).
  - Resuelve el **encaminamiento** en Internet: encontrar la ruta para llegar al destino.
  - Es un protocolo **salto a salto**. Involucra a *hosts* y *routers.*
  - Ofrece un servicio **no orientado a conexión y no fiable**:
    - No hay negociación o "*handshake*", no hay una conexión lógica entre las entidades.
    - No existe control de errores, ni control de flujo, ni control de congestión.
  - La unidad de datos (paquete) de IP se denomina **datagrama** = cabecera + datos.
  - IP es un protocolo de **máximo esfuerzo** ("*best-effort*") o buena voluntad: los datagramas se pueden perder, duplicar, retrasar o llegar desordenados.
  - IP gestiona la **fragmentación**: adaptar el tamaño del datagrama a las diferentes *Maximum Transfer Units* (MTUs) de las subredes necesarias hasta llegar al destino.

- Cada entidad IP se identifica por su dirección IP:

Servidor Webmail
130.206.192.39

www.youtube.com
172.194.34.206

Servidor Spotify
78.31.8.101

www.google.com = 172.194.34.209

www.ugr.es= 150.214.204.25
dns3.ugr.es = 150.214.191.10
pop.ugr.es = 150.214.20.3

3. EL PROTOCOLO IP

- Internet adopta un direccionamiento **jerárquico** que simplifica las tablas de *routing*.
- Las direcciones IP (32 bits) tienen dos partes bien diferencias:

  un identificador de la subred y un identificador del dispositivo (host) dentro de esa subred.
- Cada subred tiene un identificador único en la intranet (para dir. privadas) o en internet (para públicas)
- Cada dispositivo tiene un identificador único en la subred.
- La **máscara de red** es un patrón de 1s que determina qué bits pertenecen al identificador de subred

  Dirección IP ➔ 200.27.4.112 = 11001000.00011011.00000100.01110000

  Máscara ➔ 255.255.255.0 = 11111111.11111111.11111111.00000000

  La máscara se puede representar de forma compacta, por ejemplo 200.27.4.112/24
- Dada una IP, para obtener la dirección o identificador de la subred, se realiza una operación lógica &:

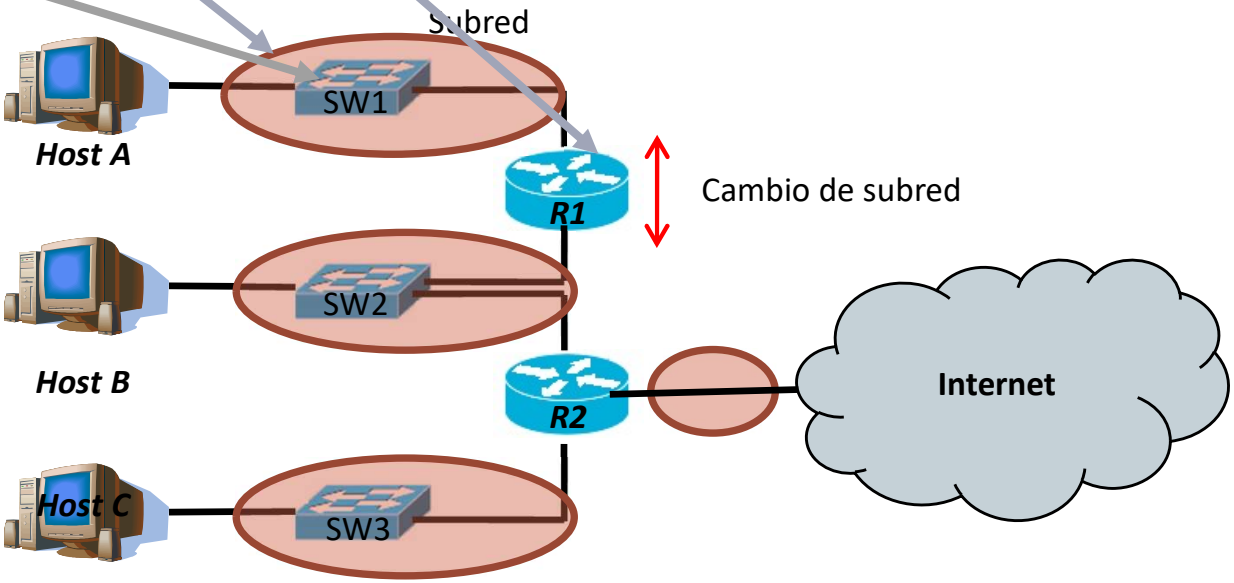
  200.27.4.112 = 11001000.00011011.00000100.01110000

  & &

  255.255.255.0 = 11111111.11111111.11111111.00000000

  Subred ➔ 200.27.4.0 = 11001000.00011011.00000100.00000000

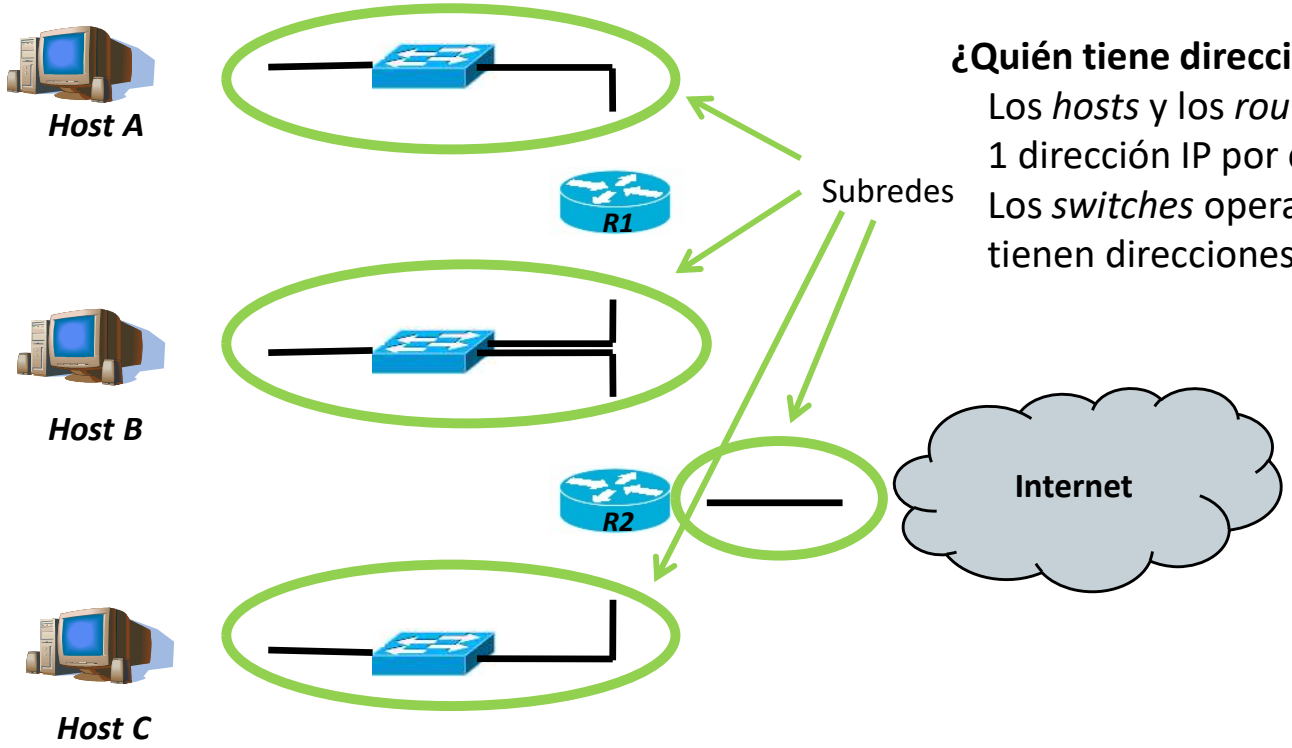
- Podemos considerar Internet como un conjunto de subredes interconectadas
- ¿Qué es una subred? líneas de transmisión e infraestructuras de red que permiten la conexión DIRECTA de dispositivos IP sin intermediarios (routers)
- ¿Qué es un *switch*? ¿Qué es un *router*?



Computer Networking. A Top-down Approach. de James F. Kurose y Keith W. Ross:

"Para determinar las subredes, separe cada interfaz de los hosts y routers, creando redes aisladas. Dichas redes aisladas se corresponden con las subredes."

- ¿Qué es una subred? ¿Qué es un *switch*? ¿Qué es un *router*?



¿Quién tiene direcciones IP?

Los *hosts* y los *routers* tienen 1 dirección IP por cada interfaz.

Los *switches* operan en cada 2 → NO tienen direcciones IP

Computer Networking. A Top-down Approach. de James F. Kurose y Keith W. Ross:

"Para determinar las subredes, separe cada interfaz de los hosts y routers, creando redes aisladas. Dichas redes aisladas se corresponden con las subredes."

- ¿Cómo se elige la máscara? ➔ Según el número de dispositivos previsibles en la subred, tal que se ajusta para no desaprovechar direcciones. Recuérdese: cada subred tiene un identificador único en nuestra intranet (direcciones privadas) o en internet (públicas).

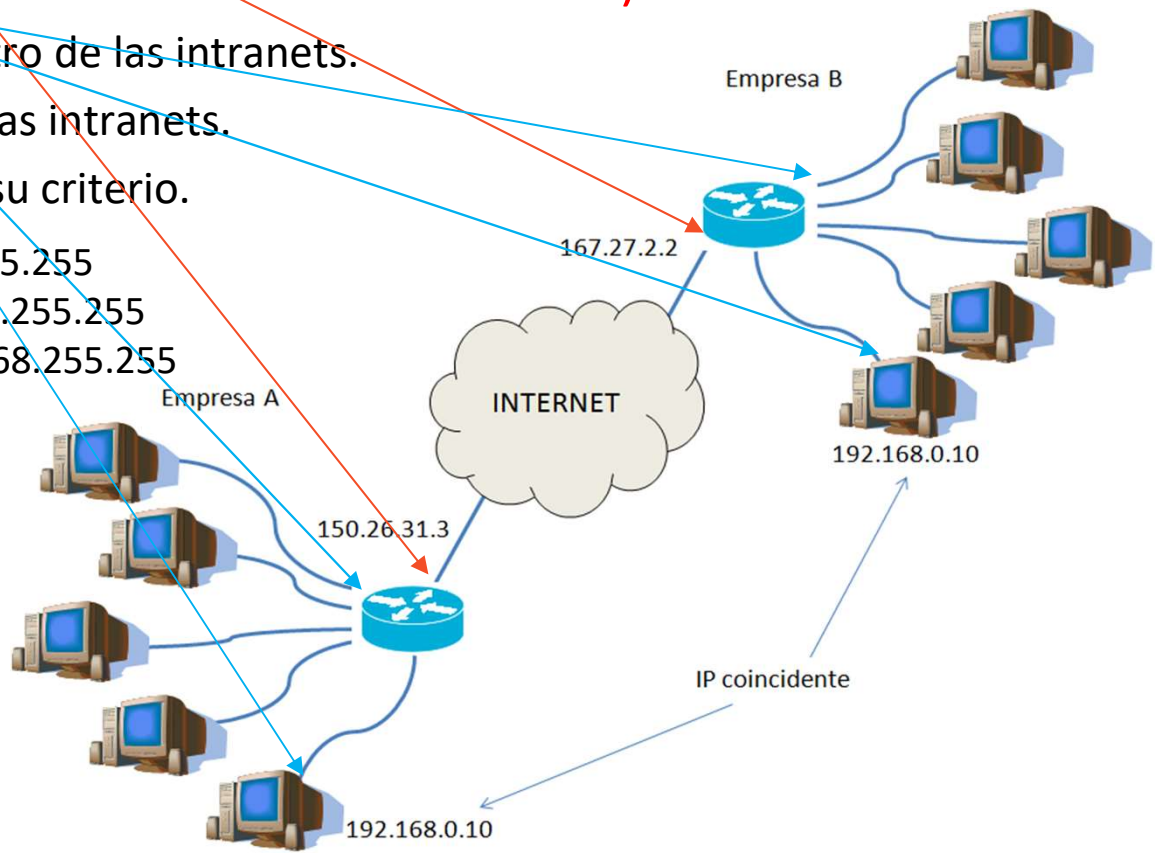
  Dirección IP ➔ 200.27.4.112 = 11001000.00011011.00000100.01110000

  Máscara ➔ 255.255.255.0 = 11111111.11111111.11111111.00000000

- # dispositivos = $2^{\#\ ceros} - 2$ ➔ ej. 8 ceros (/24) permite 254 dispositivos
  - El -2 viene de que la primera (000…0) y última (111…1) están reservadas.
    Por ejemplo en la subred 200.27.4.0/24 no se pueden asignar como id. de dispositivo
    - 200.27.4.0 = 11001000.00011011.00000100.00000000 ➔ Reservada (subred)
    - 200.27.4.1 = 11001000.00011011.00000100.00000001 ➔ Dispositivo #1
    - …
    - 200.27.4.254 = 11001000.00011011.00000100.11111110 ➔ Dispositivo #254
    - 200.27.4.255 = 11001000.00011011.00000100.11111111 ➔ Reservada (difusión)

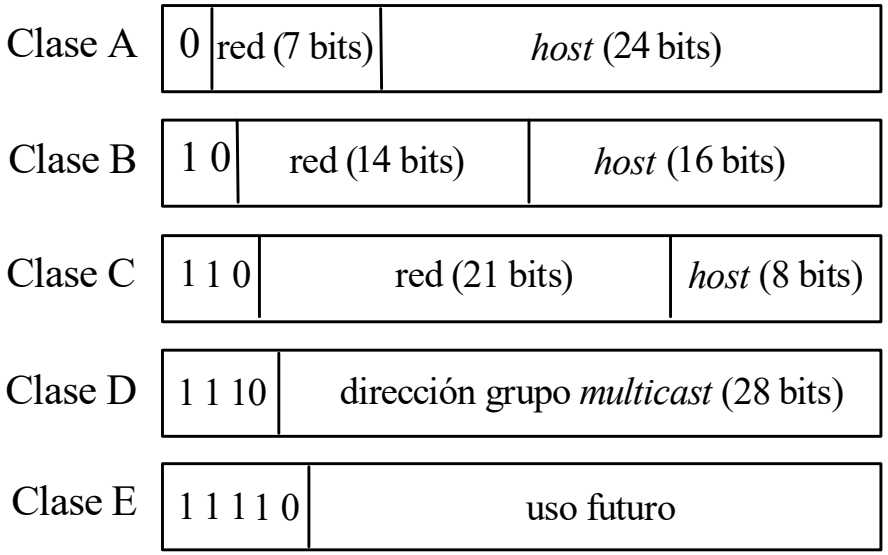
- Direcciones **públicas** (identificador único en internet)
  - Cada dirección se asigna a sólo 1 dispositivo en toda la Internet global.
  - Se asignan centralizadamente
- Direcciones **privadas** (identificador único en la intranet)
  - Sólo sirven para tráfico dentro de las intranets.
  - Se pueden repetir en distintas intranets.
  - Las asigna el usuario según su criterio.

10.0.0.0/8 ➔ de 10.0.0.0 a 10.255.255.255
172.16.0.0/16 ➔ de 172.16.0.0 a 172.31.255.255
192.168.0.0/24 ➔ de 192.168.0.0 a 192.168.255.255



- Direcciones IPv4: CLASES (ver RFC 1166 y 5737)
  - Los *hosts* y *routers* tienen una IP por cada una de sus interfaces.
  - 32 bits, notación decimal con puntos. Ejemplo: 192.168.212.60
  - Originariamente se definieron 5 clases de direcciones IP
  - Clases A,B,C → Jerárquicas a dos niveles:

    identificador de subred + identificador de dispositivo (*host*)



Tradicionalmente se definieron 5 clases de direcciones:

Rangos:

Clase A (/8) →	0.0.0.0–127.255.255.255	⇒	128 redes x 16.777.216 *hosts*
Clase B (/16) →	128.0.0.0–191.255.255.255	⇒	16.384 redes x 65.536 *hosts*
Clase C (/24) →	192.0.0.0–223.255.255.255	⇒	2.097.152 redes x 256 *hosts*
Clase D →	224.0.0.0–239.255.255.255	→	para *multicast*
Clase E →	240.0.0.0–255.255.255.255	→	usos futuros

Reglas especiales: (# dispositivos = $2^{\#\ ceros} - 2$)

host = 00…0 ⇒ identifica a una red, nunca es una dirección origen, no se usa para dispositivos

host = 11…1 ⇒ difusión en la red especificada, es una dirección destino, no se usa para dispositivos

127.0.0.0 ⇒ autobucle (*loopback*)

Para evitar ambigüedades el identificativo de dispositivo (host) no debe ser ni 255 ni 0

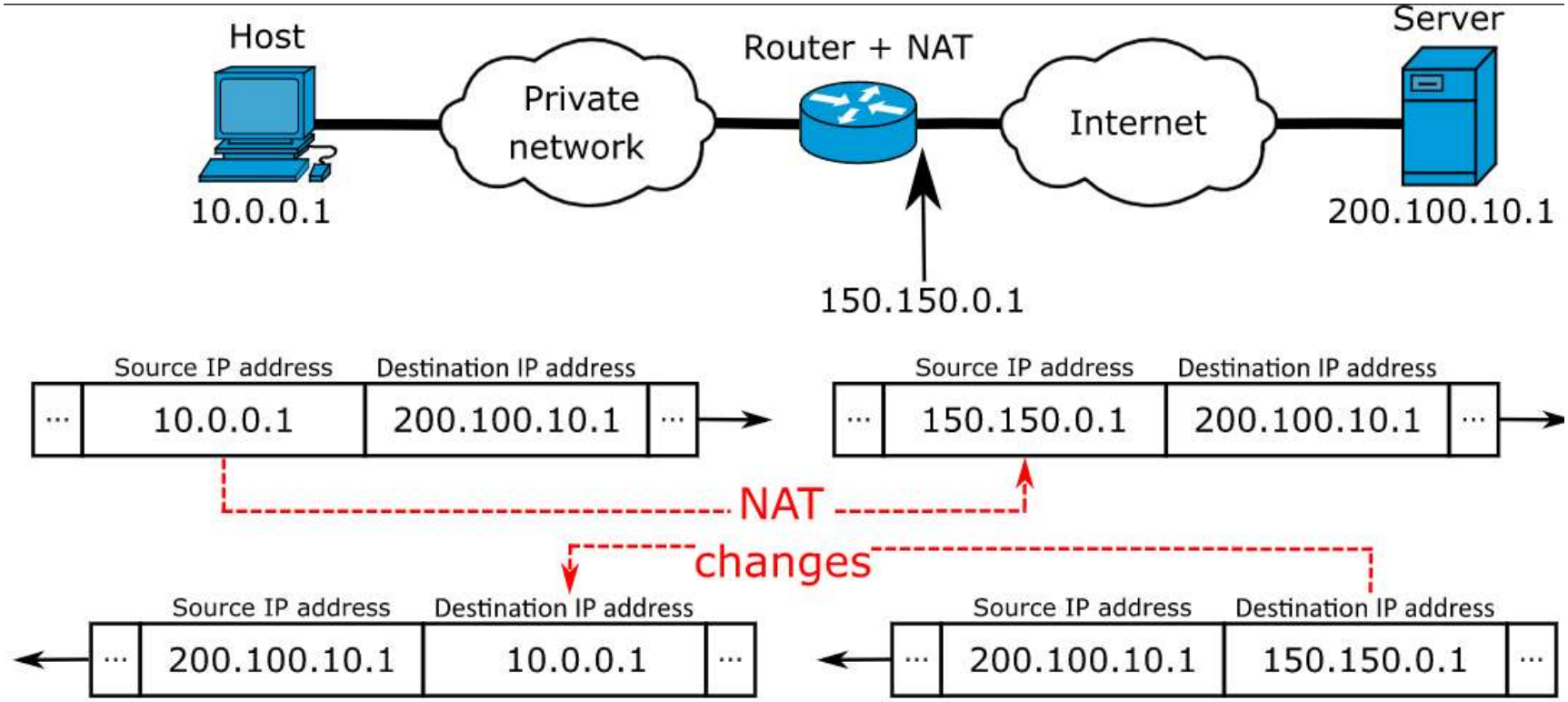
Reserva de direcciones privadas (RFC1918):

Clase A →	10.0.0.0 →	1 Red privada clase A
Clase B →	172.16.0.0 – 172.31.0.0 →	16 redes privadas clase B
Clase C →	192.168.0.0 – 192.168.255.0 →	256 redes privadas clase C

Gestión/asignación: IANA *(www.iana.org) ahora gestionada por ICANN (www.icann.org)*

3. EL PROTOCOLO IP

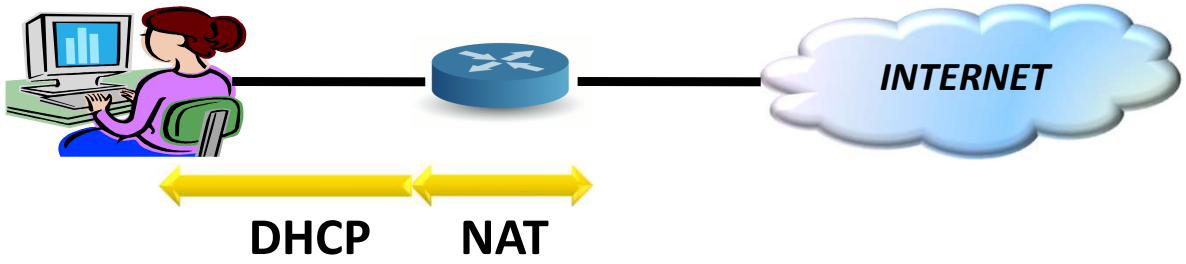
- NAT ("*Network Address Translation*") RFC 1631, 2663, 3022:



NAT es un método para reasignar un espacio de direcciones IP (típicamente privadas) a otro (públicas) modificando la dirección IP de los paquetes mientras se retransmiten a través de un *router*

3. EL PROTOCOLO IP

- NAT ("*Network Address Translation*") RFC 1631, 2663, 3022
- En su uso habitual, el *router* NAT reemplaza las direcciones IP privadas origen salientes por direcciones públicas, y al revés con las entrantes.



- Para ello el NAT usa una "**Tabla de Traducciones**", que con la ayuda de una reasignación de puertos, permite deshacer los cambios en el trafico entrante.
- IMPORTANTE: No se suelen instalar servidores (detrás de un NAT) con direcciones privadas, pues no serían accesibles desde el exterior. Una posible solución es usar STUN (*Session Traversal Utilities for* NAT): protocolo cliente/servidor que permite a clientes NAT encontrar su dirección IP pública, el tipo de NAT en el que se encuentra y el puerto asociado en la tabla de NAT con el puerto local.
- *Entonces…¿cómo funcionan aplicaciones como Whatsapp desde una IP privada?*

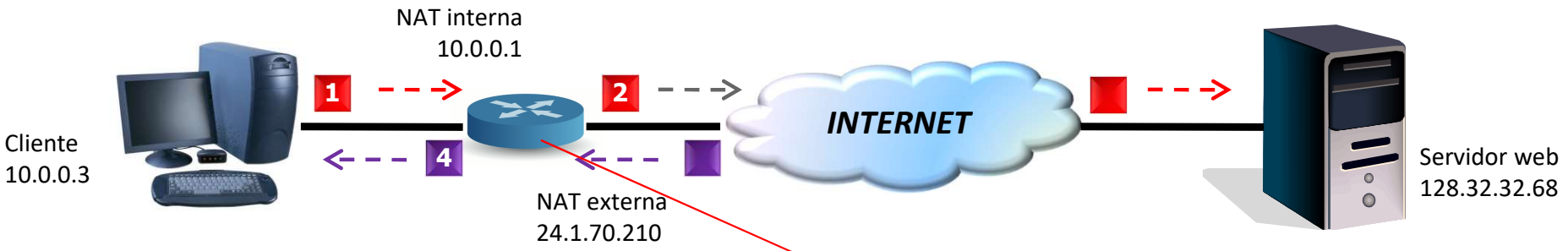
3. EL PROTOCOLO IP

PROTO	TCP
SADDR	10.0.0.3
DADDR	128.32.32.68
SPORT	1049
DPORT	80
FLAGS	SYN
CKSUM	0x1636

1. El cliente intenta conectarse al servidor web 128.32.32.68 y envía un paquete SYN con su dirección IP interna 10.0.0.3 (privada).

PROTO	TCP
SADDR	24.1.70.210
DADDR	128.32.32.68
SPORT	40960
DPORT	80
FLAGS	SYN
CKSUM	0x2436

2. El dispositivo NAT ve la configuración del paquete, añade una nueva entrada a su tabla de traducción. Luego modifica el paquete usando su dirección IP externa (pública), cambia el puerto y el chequeo de integridad del paquete.



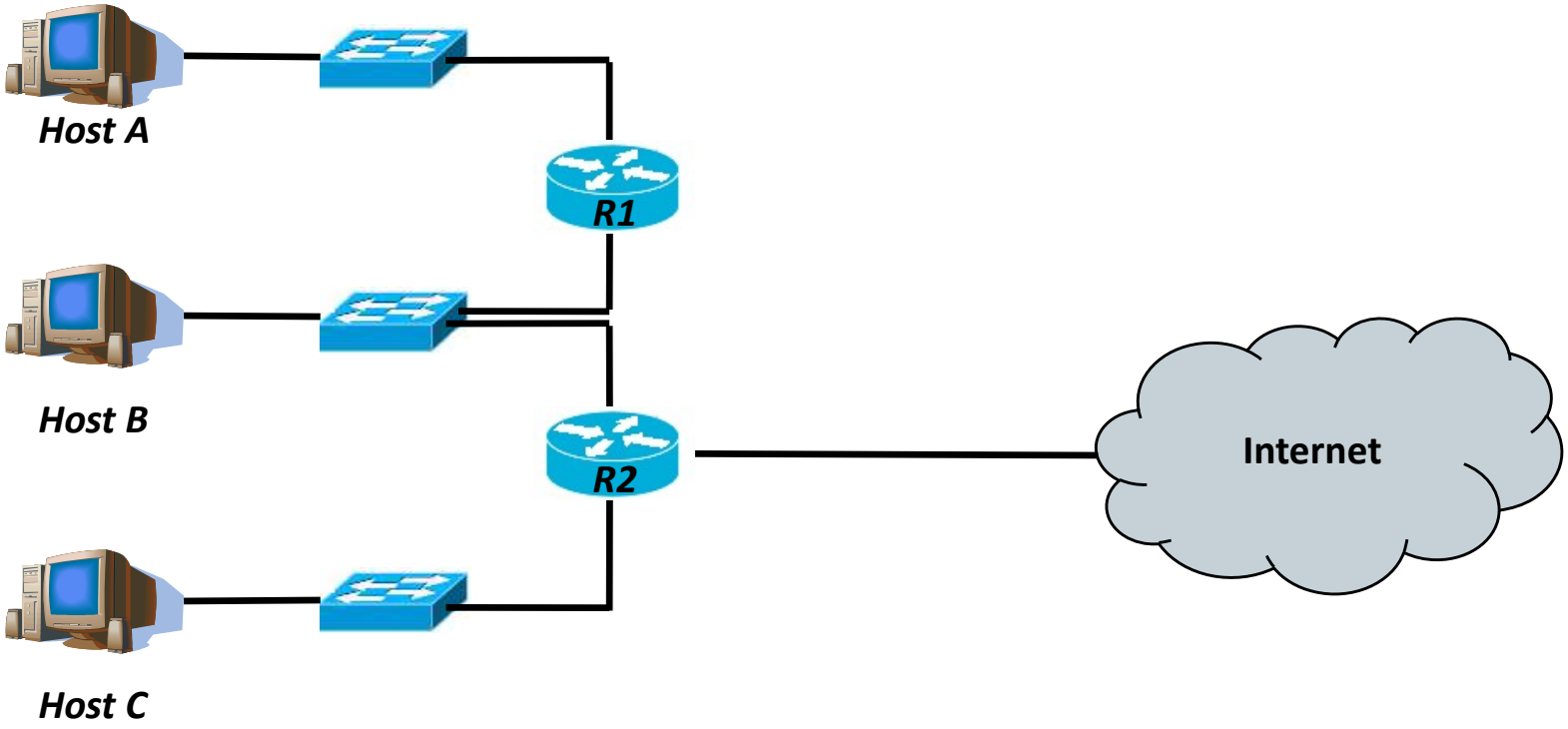
PROTO	TCP
SADDR	128.32.32.68
DADDR	10.0.0.3
SPORT	80
DPORT	1049
FLAGS	SYN, ACK
CKSUM	0x7841

4. El dispositivo NAT mira su tabla de traducción, y encuentra la que corresponde a direcciones y puertos origen/destino. Reescribe el paquete utilizando los puertos y direcciones internas.

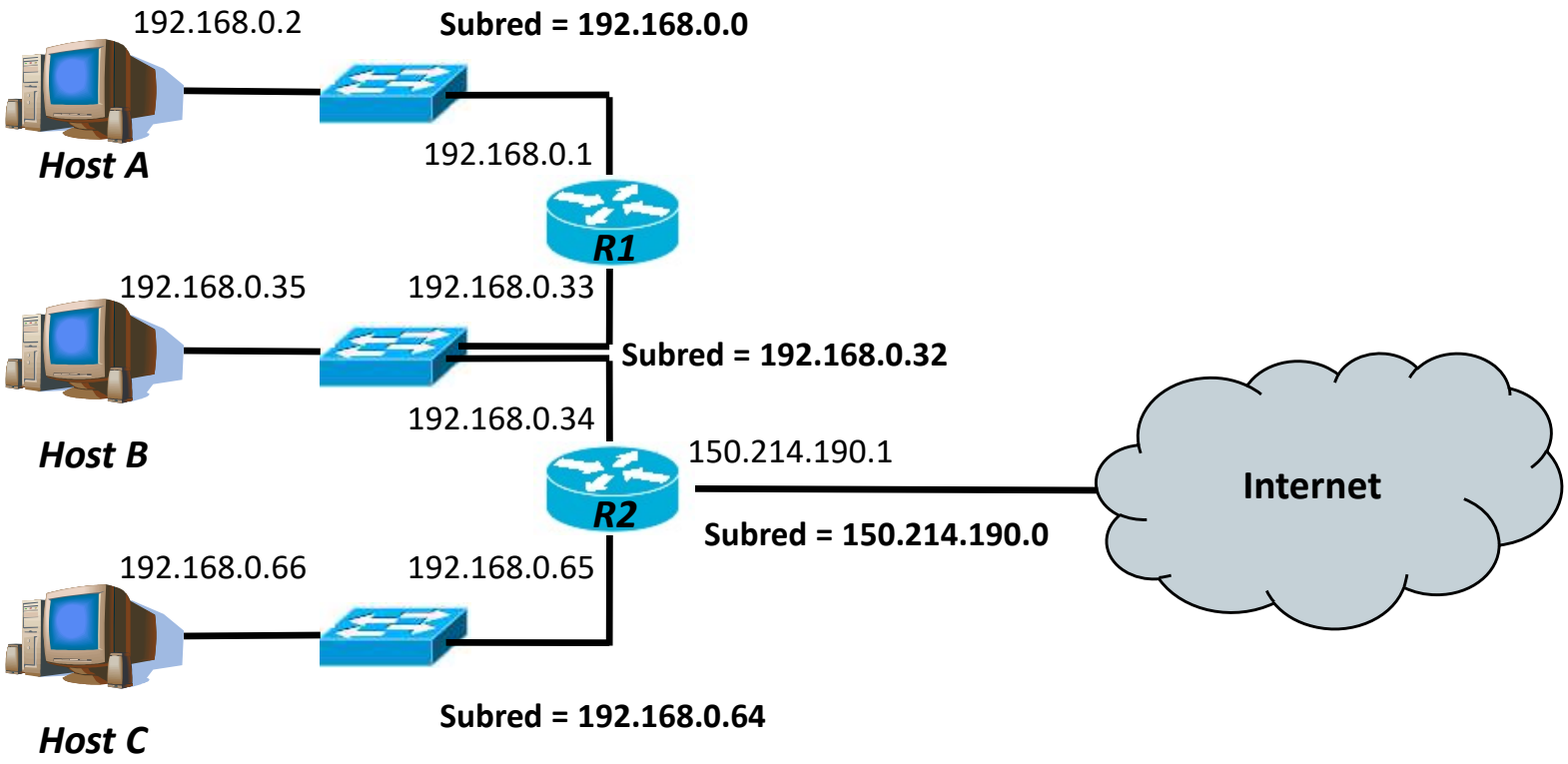
Original	NAT
10.0.0.3:1049	24.1.70.210:40960
...	...

PROTO	TCP
SADDR	128.32.32.68
DADDR	24.1.70.210
SPORT	80
DPORT	40960
FLAGS	SYN, ACK
CKSUM	0x8041

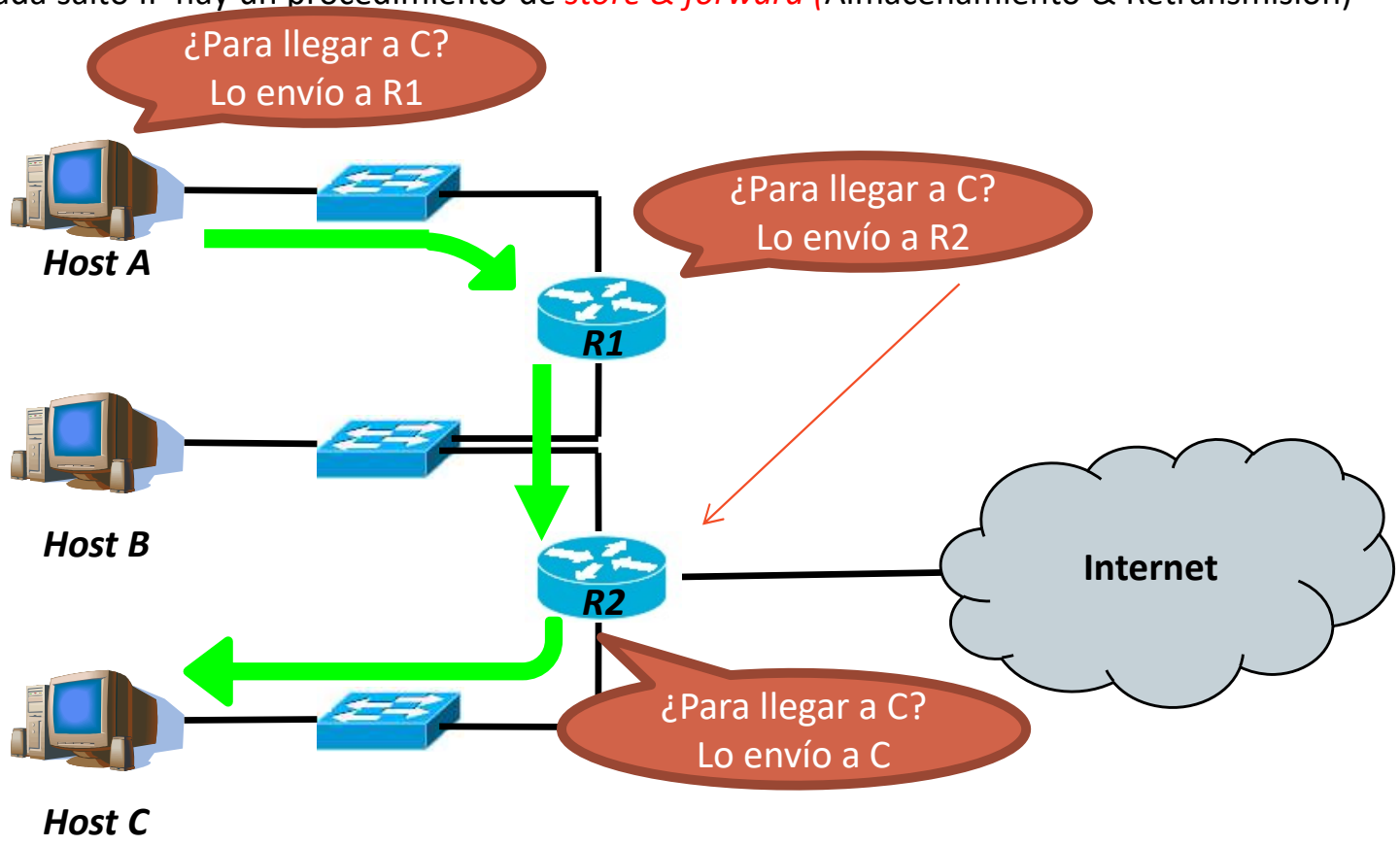
- Ejercicio: Asignar direcciones
  - Supongamos 3 subredes corporativas con 30 dispositivos cada una
  - Usemos direcciones privadas 192.168.0.0
  - Supongamos una subred de acceso con direccionamiento público (asignado por el ISP)



- Asignar direcciones → minimizar las IPs no utilizadas + simplificar el encaminamiento
  - Subredes corporativas: 30 dispositivos, direcciones privadas 192.168.0.0 ➔ 5 ceros, /27
  - Subred de acceso: dirección pública (ISP) ➔ 2 ceros, /30, por ejemplo 150.214.190.0



- El encaminamiento:
  - Encontrar el mejor camino para llevar la información (paquete a paquete) de un origen a un destino dado.
  - Se realiza paquete a paquete y salto a salto, en función de la IP destino del paquete y de las tablas de encaminamiento residentes en cada una de las entidades IP (host origen y *routers).*
  - En cada salto IP hay un procedimiento de *store & forward (*Almacenamiento & Retransmisión)



- Modos de encaminamiento: directo y no directo.
- Cada dispositivo (*host o router*) tiene una tabla de encaminamiento
- Un *router* suele estar en varias redes distintas, un *host* suele estar en solo una red
- Al consultar la tabla, en caso de conflicto, se elige la ruta con la máscara más larga
- El *default* no es obligatorio pero simplifica mucho las tablas

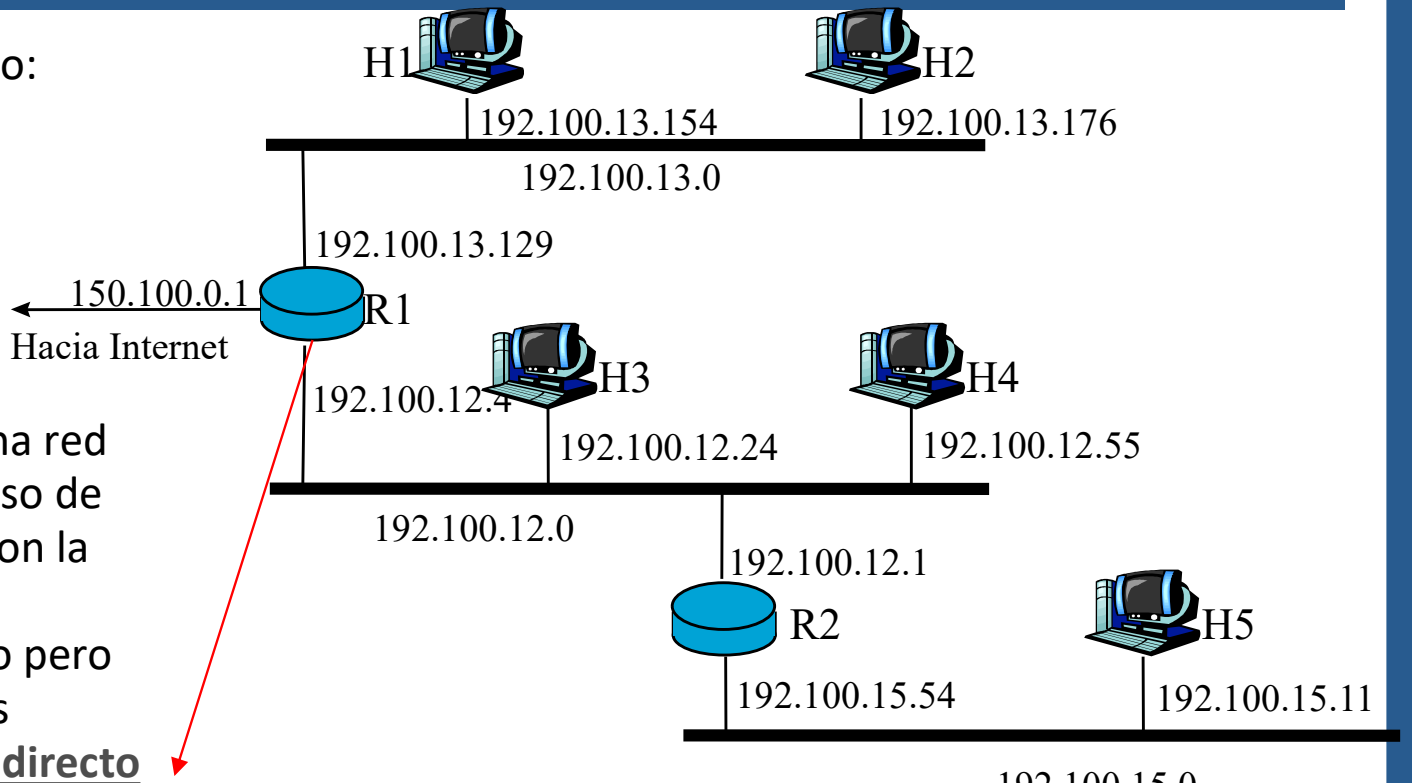


Tabla de R1, * = routing directo

i	Destino (D_i)	Salto siguiente (S_i)	Máscara (M_i)	Flags	Interfaz(I_i)
1	127.0.0.1	*	255.255.255.255	H	lo
2	192.100.12.0	*	255.255.255.0	-	eth0
.	192.100.13.0	*	255.255.255.0	-	eth1
	150.100.0.0	*	255.255.255.252	-	eth2
.	192.100.15.0	192.100.12.1	255.255.255.0	G	eth0
N	Default	150.100.0.2	0.0.0.0	G	eth2

- Si no hay fragmentación y no hay "traducción de direcciones" (NAT), el datagrama (salvo el TTL, las opciones y el campo de comprobación) no se modifica en el camino.

- Proceso de encaminamiento en los nodos IP (salto a salto) por cada datagrama:

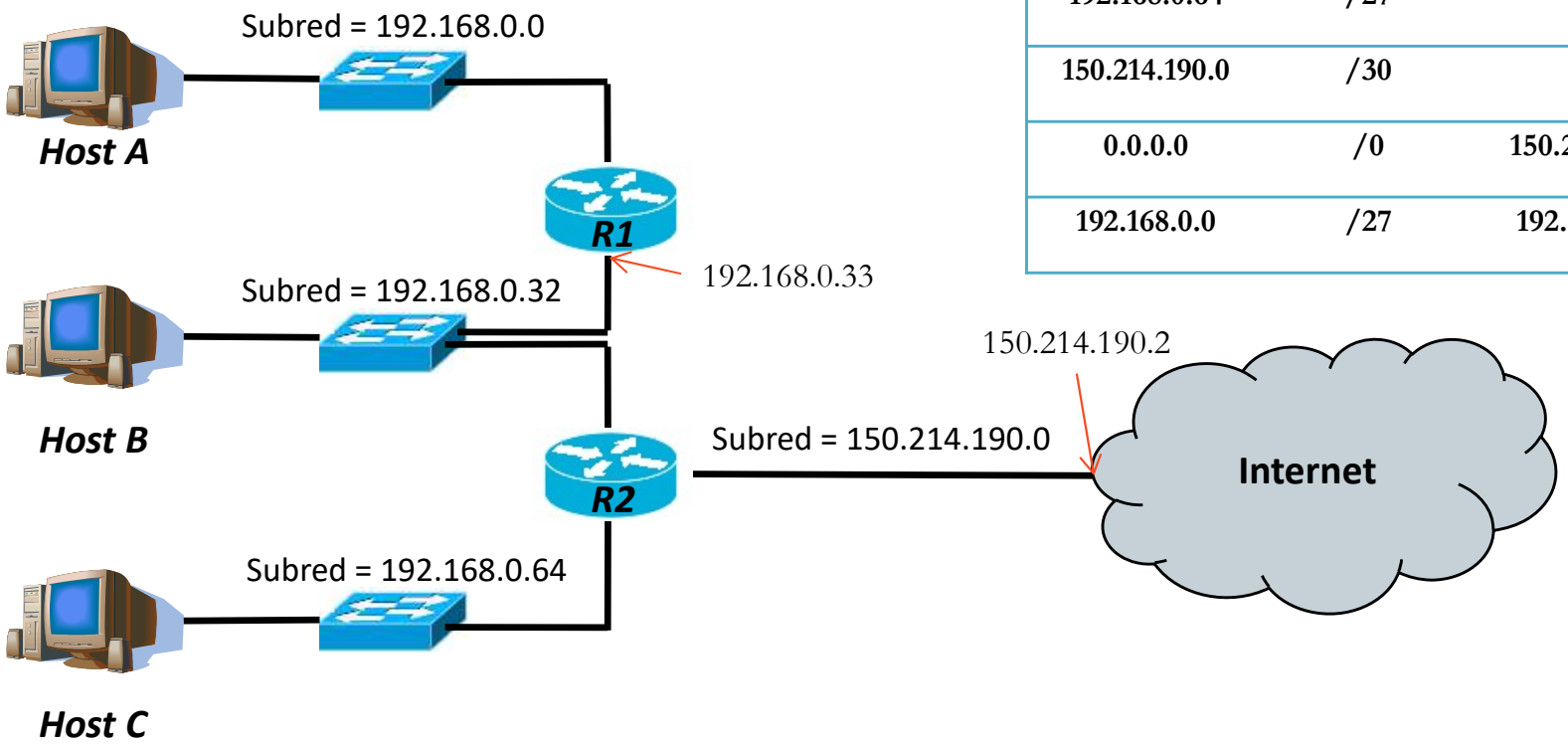
  - Se extrae la dirección destino: IP_DESTINO del datagrama
  - Por cada entrada i de la tabla de encaminamiento, con i =1,..,N, se calcula

    IPi = IP_DESTINO AND (&) MASCARA_i

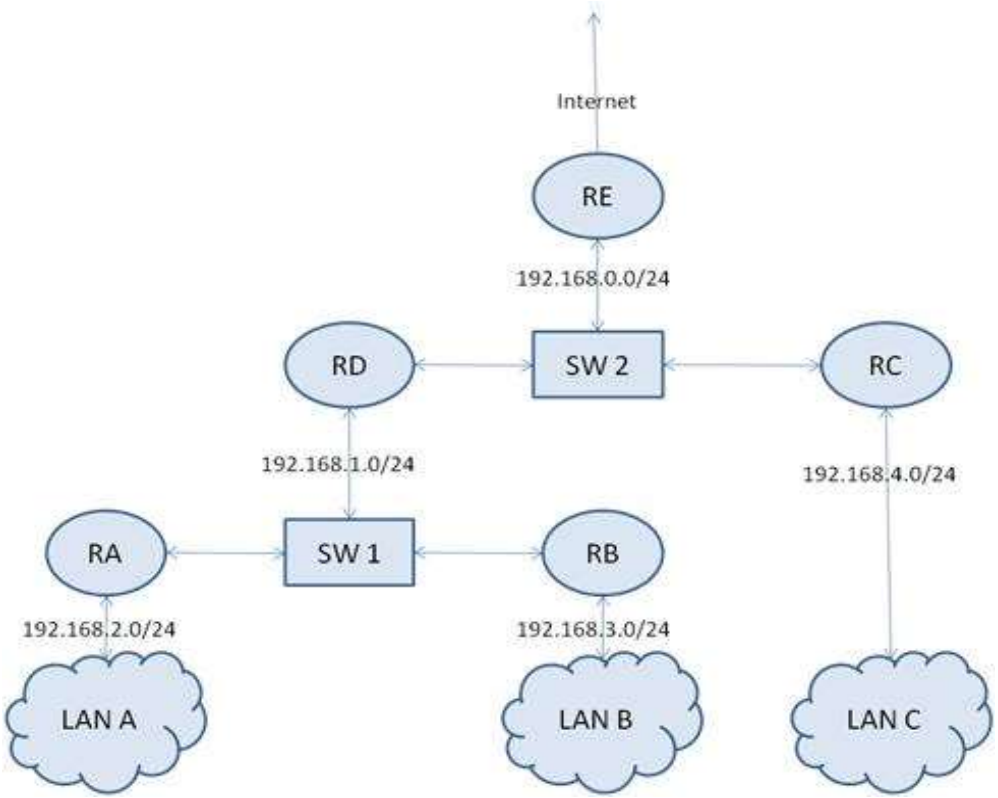
  - Si IPi == Di (campo destino de la tabla de encaminamiento) y
    si es *routing* directo (*) → reenviar el datagrama al destino final por la interfaz I_i
    o si no es *routing* directo→ reenviar el datagrama al "**salto siguiente**" por la I_i
  - Si hay varias coincidencias, se elige el destino Di con la máscara más larga
  - Si se ha barrido toda la tabla y no hay coincidencia con ninguna fila → error (posible mensaje ICMP)
  - Para encapsular el datagrama en la trama física correspondiente, se debe consultar la tabla ARP (ver más adelante) y en caso de no conocer la dirección física se envía un *broadcast* con protocolo ARP para obtener la dirección física.

- **Ejercicio: Diseñar manualmente la tabla de encaminamiento en R2 → 3 Pasos:**
  1. Incorporar todas las redes directamente conectadas.
  2. Incorporar la entrada por defecto
  3. Añadir todas las entradas adicionales necesarias.

Dirección IP destino	Máscara	Siguiente nodo
192.168.0.32	/27	-
192.168.0.64	/27	-
150.214.190.0	/30	-
0.0.0.0	/0	150.214.190.2
192.168.0.0	/27	192.168.0.33

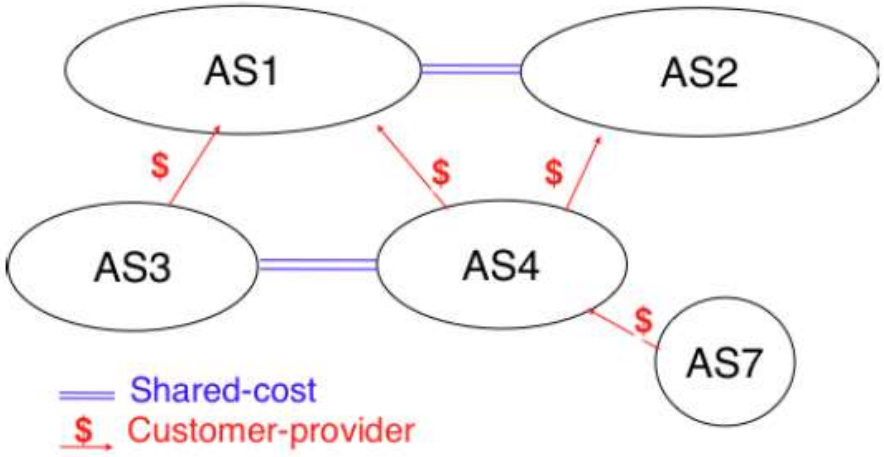


7. Imagine una situación donde hay cinco routers RA-RE. RA, RB y RC se conectan cada uno a una red local A, B y C, siendo cada router única puerta de enlace de cada red. RA, RB y RD están conectados entre sí a través de un switch. RC, RD y RE están conectados entre sí a través de un switch. RE conecta a Internet a través de la puerta de acceso especificada por el ISP. Especifique tablas de encaminamiento en los routers. Asigne a voluntad las direcciones IP e interfaces necesarias.



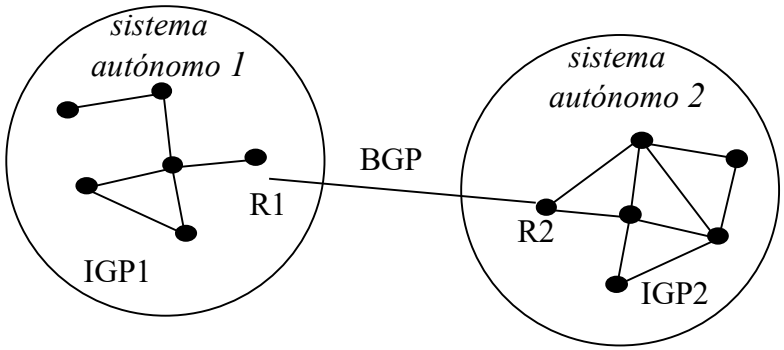
3. EL PROTOCOLO IP

- Para facilitar la administración y aumentar la escalabilidad Internet se jerarquiza en **Sistemas Autónomos** (SA).
- Un **SA** es un conjunto de redes y *routers* administrados por una única autoridad que define cómo es el intercambio de tablas (*routing* interno) dentro del SA
- En cada SA existe un *router*, denominado *router* exterior, responsable de informar a los otros SAs sobre las redes accesibles a través del SA
- Cada SA se identifica por un entero de 16 bits (DESDE 2007 ES 32-BITS). Por ejemplo Rediris = AS766

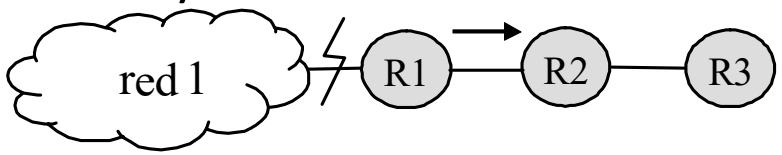


- Intercambio automático de tablas de encaminamiento
  - Se definen 2 niveles para el intercambio de tablas:
    - Protocolos IGP: el administrador tiene libertad para elegir el protocolo de intercambio de tablas entre los *routers* dentro del SA.

      Ejemplos de protocolos IGP: **RIP, OSPF, HELLO, IS-IS, IGRP, EIGRP**
    - Protocolo EGP (norma única en Internet) para el intercambio de información entre SA. Todos los "*routers* exteriores" usan el protocolo único en Internet: **BGP**



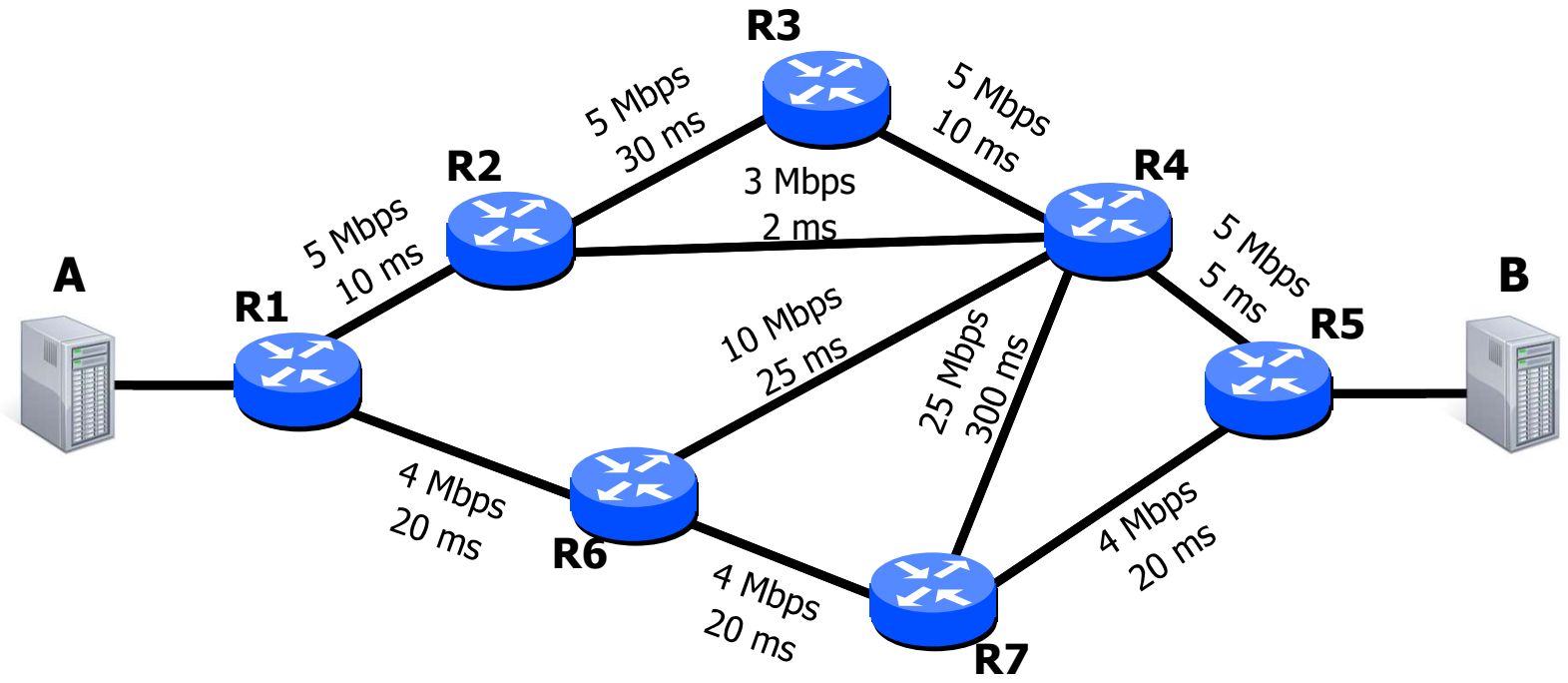
- RIP ("Routing Information Protocol" RFC 1058, 2453, 4822)
  - Protocolo de la capa de aplicación (opera sobre UDP puerto 520).
  - Adopta un algoritmo *vector-distancia* (métrica basada en número de saltos).
  - Periódicamente (por defecto cada 30 segundos) cada *router* RIP recibe y envía a todos sus vecinos (dirección *multicast* 224.0.0.9) los vectores-distancia para todos los posibles destinos.
  - De entre ellos, para un destino dado en la tabla de encaminamiento, se selecciona como salto siguiente (gateway) el vecino que anuncie el menor coste a ese destino, actualizando la métrica para ese destino, sumando 1 al coste anunciado.
  - Problema las malas noticias tardan en propagarse.
  - Problema de la convergencia lenta y la cuenta al infinito.
  - Soluciones:
    Split horizon
    Hold down
    Poison reverse
  - Ver > man routed (SO Linux)



- Open Short-Path First (RFC 2328)
  - Basado en estado del enlace (coste α 1 / velocidad del enlace). Se informa a todos sobre el coste de los vecinos
  - A partir del grafo se aplica Dijkstra
  - Permite rutas alternativas y balanceo de carga
  - Para ser escalable define áreas independientes
  - Minimiza difusión mediante *routers* designados en canales compartidos
  - Mensajes: hello, database description, link status request/update/ack

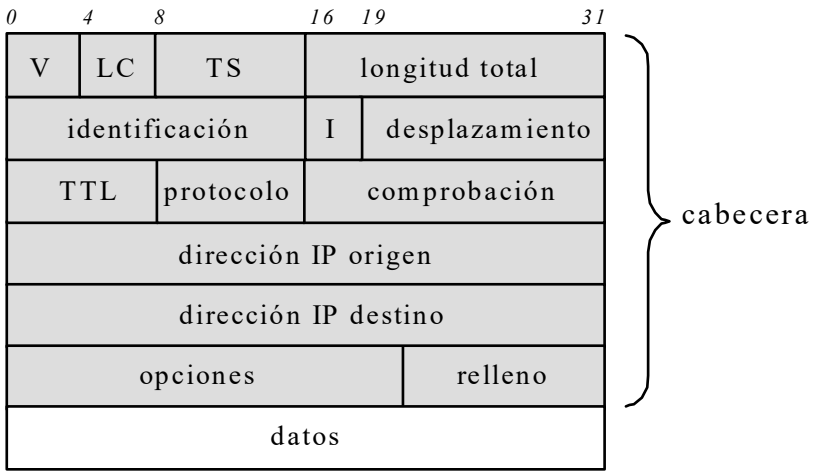
- Open Short-Path First (RFC 2328)

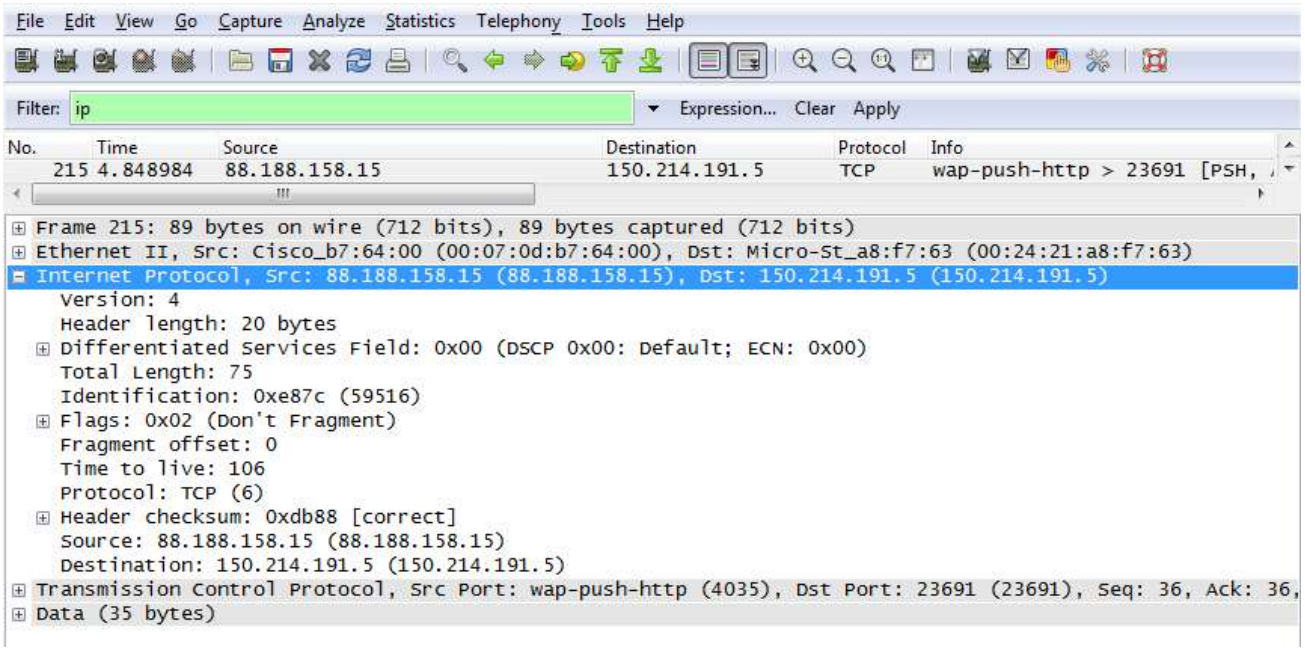
  Ejemplo para RIP y OSPF



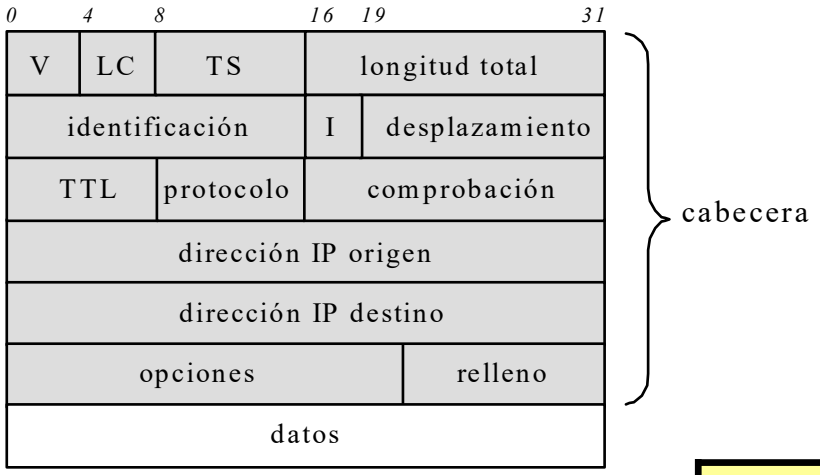
3. EL PROTOCOLO IP

- Formato de datagrama





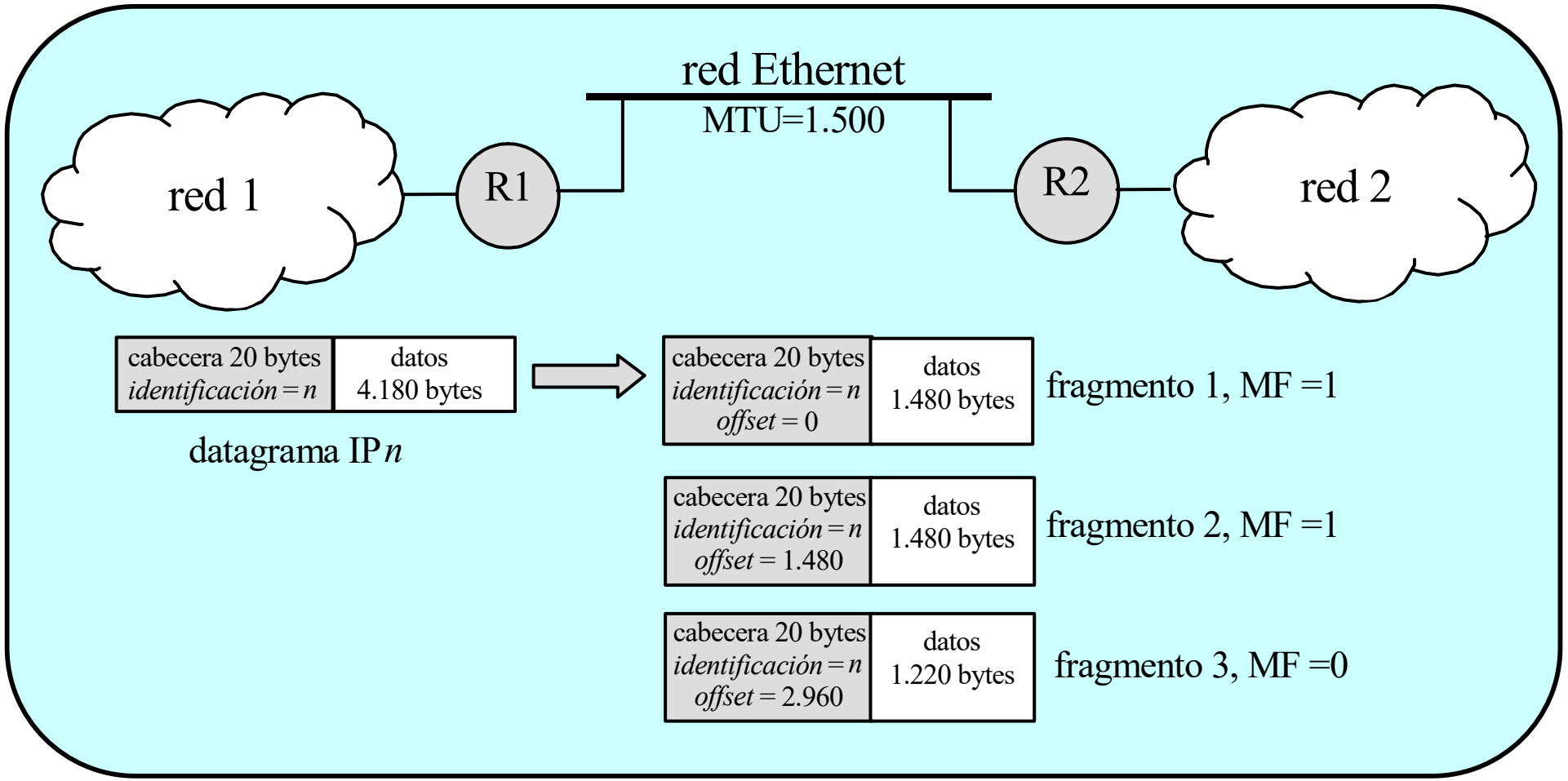
- Formato de datagrama



- Fragmentación IPv4:
  - Tamaño máximo del datagrama: 2^{16}-1 = 65.535 bytes.
  - Es necesario adaptarse a la MTU (Maximum Transfer Unit) de cada subred
  - El ensamblado sólo se puede hacer en el destino final
  - **desplazamiento**: offset respecto del comienzo del paquete.
  - **indicadores (I)**: "Don´t Fragment", "More Fragments".

Nivel de enlace	MTU (bytes)
PPP normal	1500
PPP bajo retardo	296
X.25	1600 (RFC 1356)
Frame Relay	1600 (normalmente)
Ethernet DIX	1500
Ethernet LLC-SNAP	1492
Token Ring 4 Mb/s	4440 (THT 8ms)
Classical IP over ATM	9180

- Fragmentación IPv4:

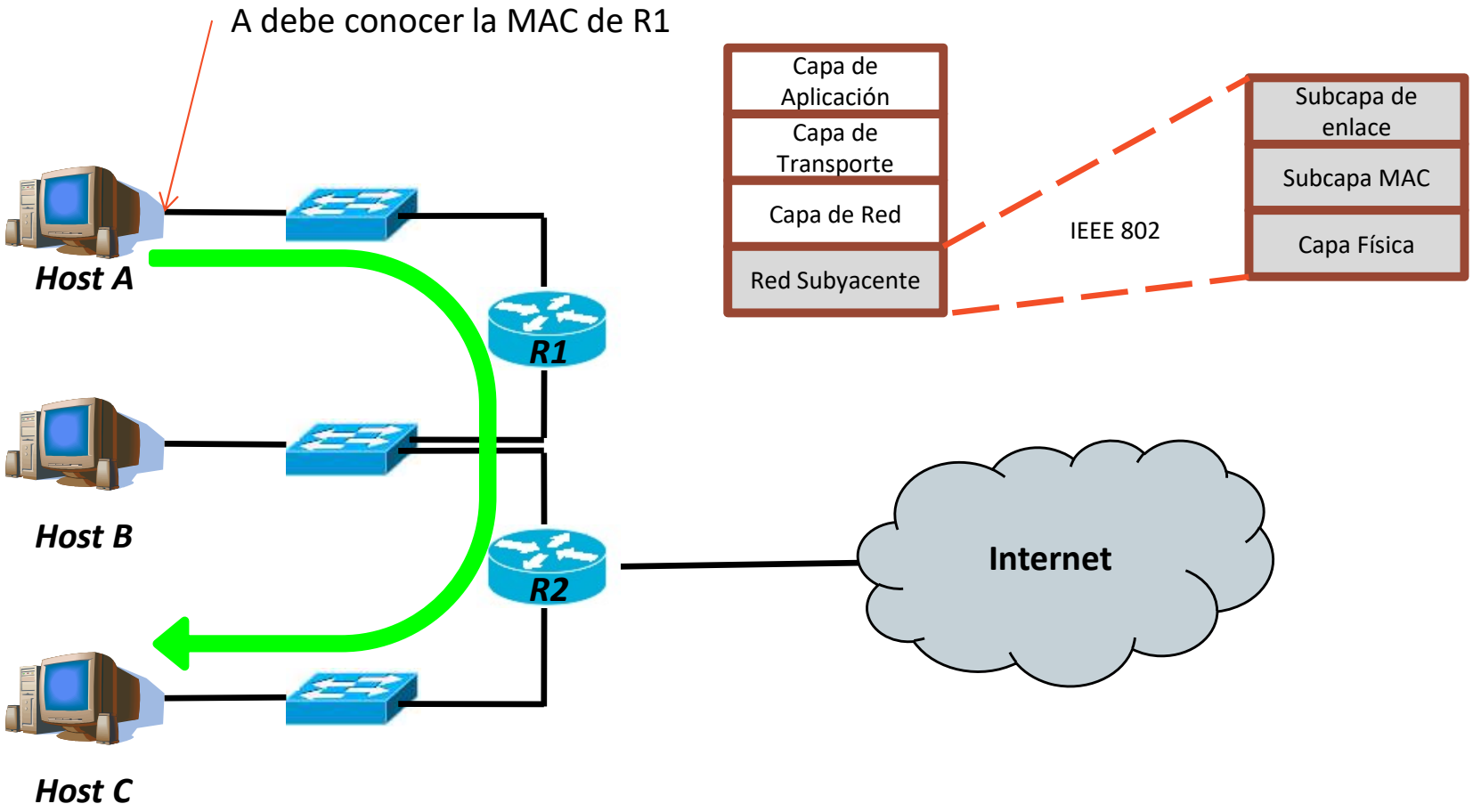


Tema 2. CAPA DE RED

1. Introducción
2. Conmutación
3. El protocolo IP
4. **Asociación con capa de enlace: el protocolo ARP**
5. El protocolo ICMP
6. Autoconfiguración de la capa de red (DHCP)

4. EL PROTOCOLO ARP

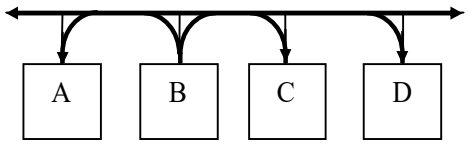
- Direcciones MAC
  - Tras la redirección IP ➔ Enviar a la MAC del siguiente nodo



4. EL PROTOCOLO ARP

- Direcciones MAC
  - Tras consultar la tabla de encaminamiento ➔ enviar el datagrama a la dirección *Medium Access Control* (MAC) del siguiente nodo. Se usan en redes Ethernet (cableadas) y Wifi
  - Formato de las MAC (6 bytes): HH-HH-HH-HH-HH-HH ➔ ej. 00-24-21-A8-F7-6A
  - Son únicas, asignadas por IEEE en lotes de 2^{24} para cada fabricante
  - Existe definida una dirección de difusión (*broadcast*) FF-FF-FF-FF-FF-FF
  - Protocolo: *Address Resolution Protocol* (ARP)

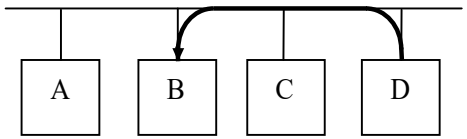
    Obtener la dir. MAC a partir de la IP: (a) y (b)



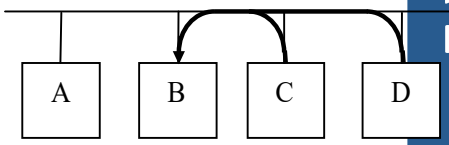
(a)

  - Protocolo: *Rerverse* ARP (RARP)

    Obtener la IP a partir de la MAC: (a) y (c)



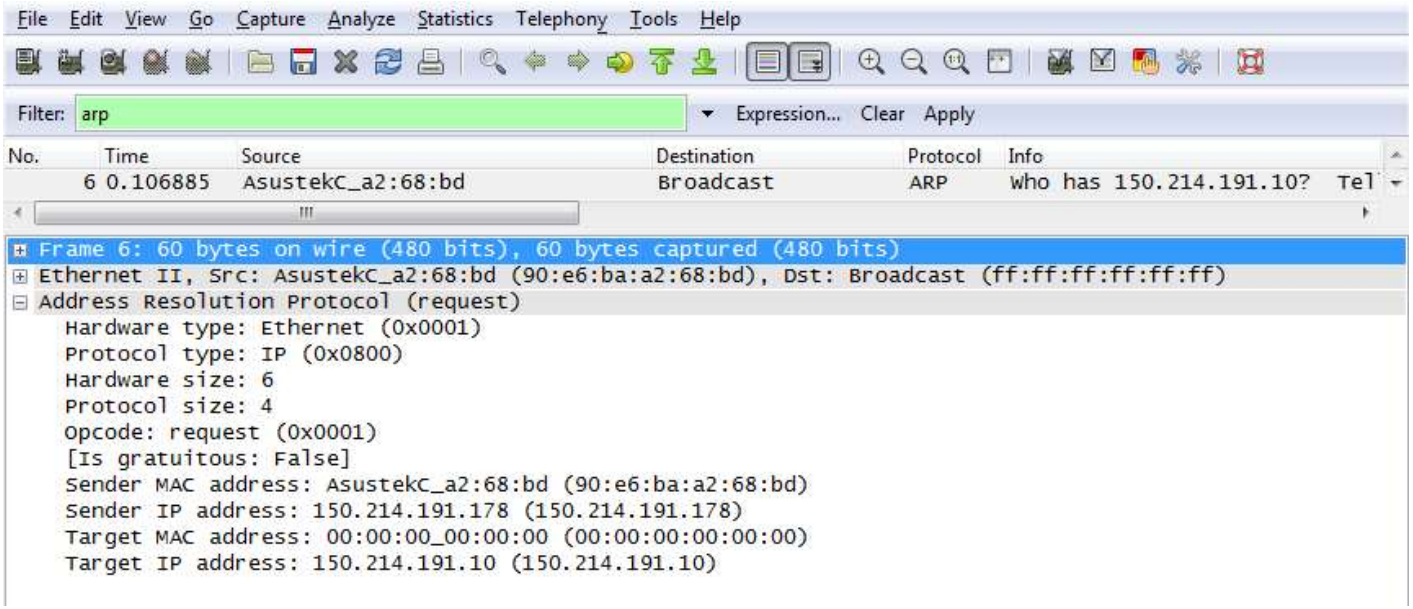
(b)



(c)

- Formato ARP:

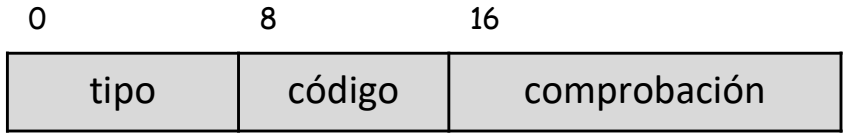
0	8	16	31
Htipo		Ptipo	
Hlen	Plen	Operación	
Hemisor (bytes 0-3)			
Hemisor (bytes 4-5)		Pemisor (bytes 0-1)	
Pemisor (bytes 2-3)		Hsol (bytes 0-1)	
Hsol (bytes 2-5)			
Psol (bytes 0-3)			



Tema 2. CAPA DE RED

1. Introducción
2. Conmutación
3. El protocolo IP
4. Asociación con capa de enlace: el protocolo ARP
5. **El protocolo ICMP**
6. Autoconfiguración de la capa de red (DHCP)

- ICMP (*Internet Control Message Protocol*)
  - Informa sobre situaciones de error en IP ➔ es un protocolo de señalización
  - Suelen ir (excepto eco y solicitudes) hacia el origen del datagrama IP original
  - ICMP se encapsula en IP
  - Cabecera de 32 bits
    - Tipo (8 bits): tipo de mensaje
    - Código (8 bits): subtipo de mensaje
    - Comprobación (16 bits)

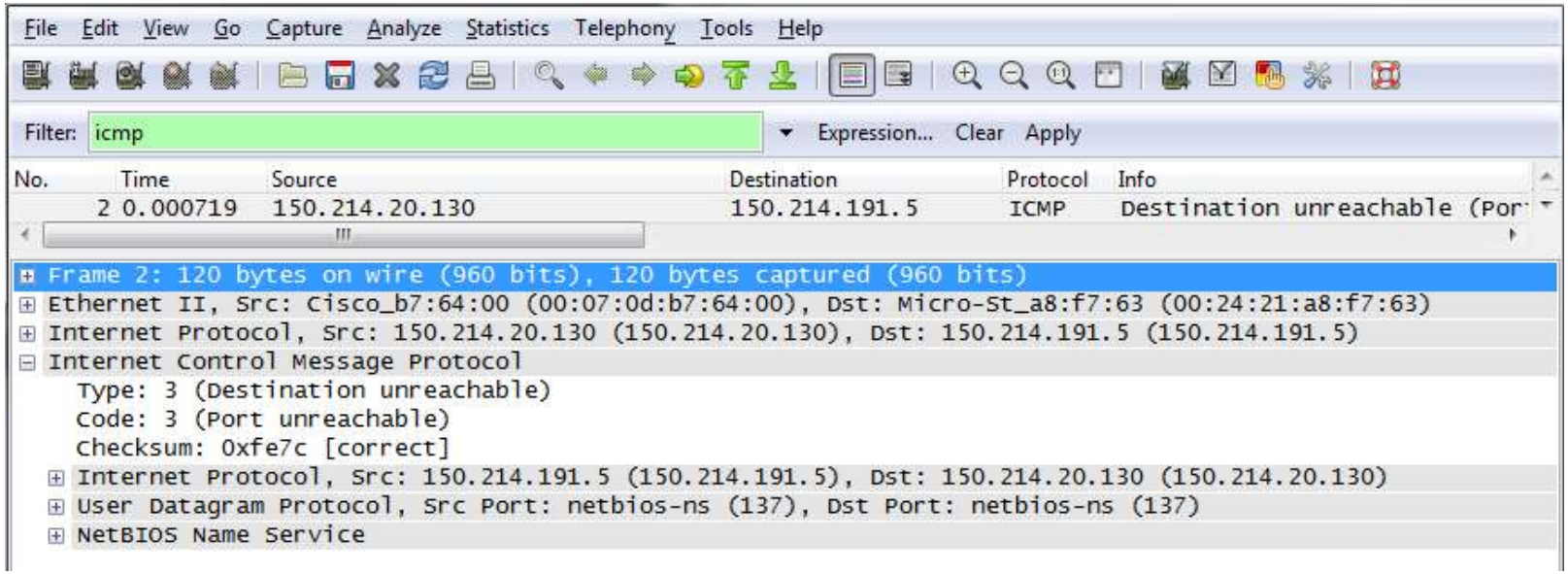


Mensajes ICMP:

Campo *tipo*	Mensaje ICMP
8/0	Solicitud/respuesta de eco (ping)
3	Destino inalcanzable
4	Ralentización del origen
5	Redireccionamiento
11	Tiempo de vida excedido
12	Problema de parámetros
13/14	Solicitud/respuesta de sello de tiempo
17/18	Solicitud/respuesta de máscara de red

5. EL PROTOCOLO ICMP

- ICMP (*Internet Control Message Protocol*)
  - informa sobre situaciones de error ➔ señalización
  - Hacia el origen del datagrama IP.
  - Se encapsula en IP
  - Cabecera de 32 bits. Incluye la cabecera del datagrama que ha disparado el mensaje



Tema 2. CAPA DE RED

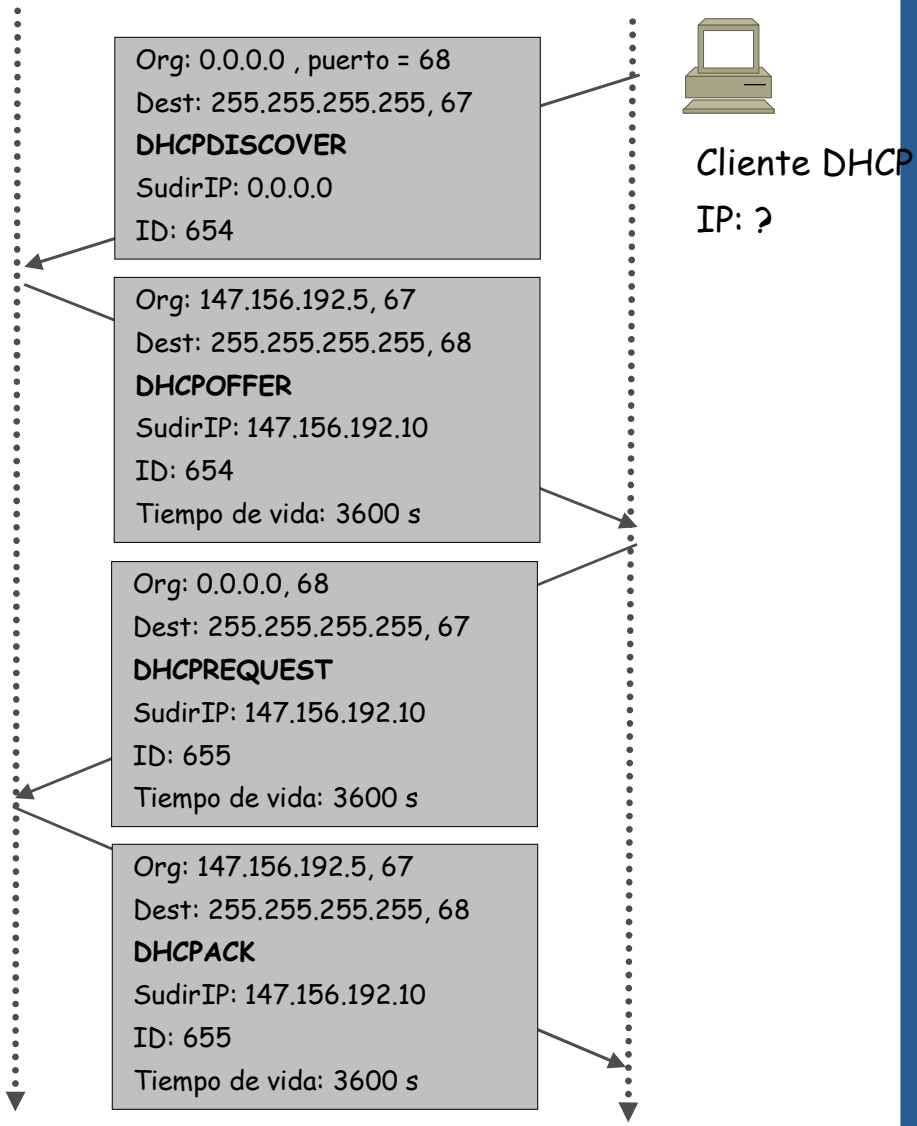
1. Introducción
2. Conmutación
3. El protocolo IP
4. Asociación con capa de enlace: el protocolo ARP
5. El protocolo ICMP
6. **Autoconfiguración de la capa de red (DHCP)**

DHCP (Dynamic Host Configuration Protocol)

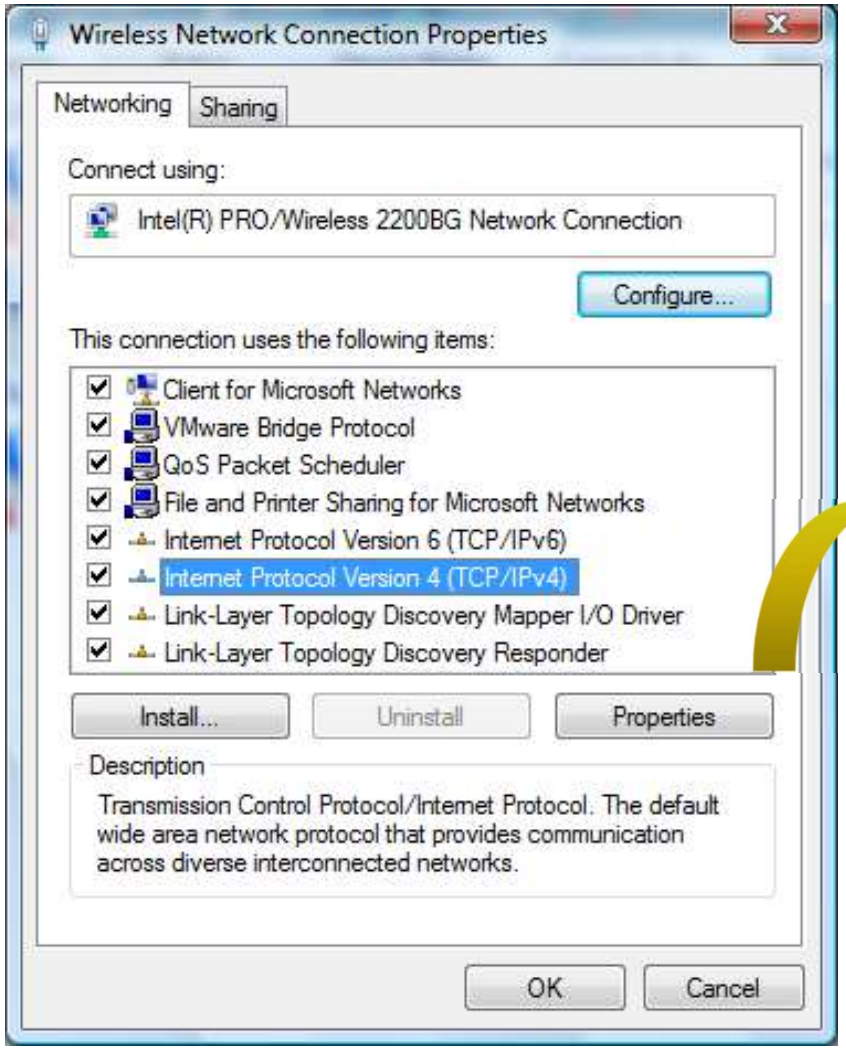
Servidor DHCP
147.156.192.5

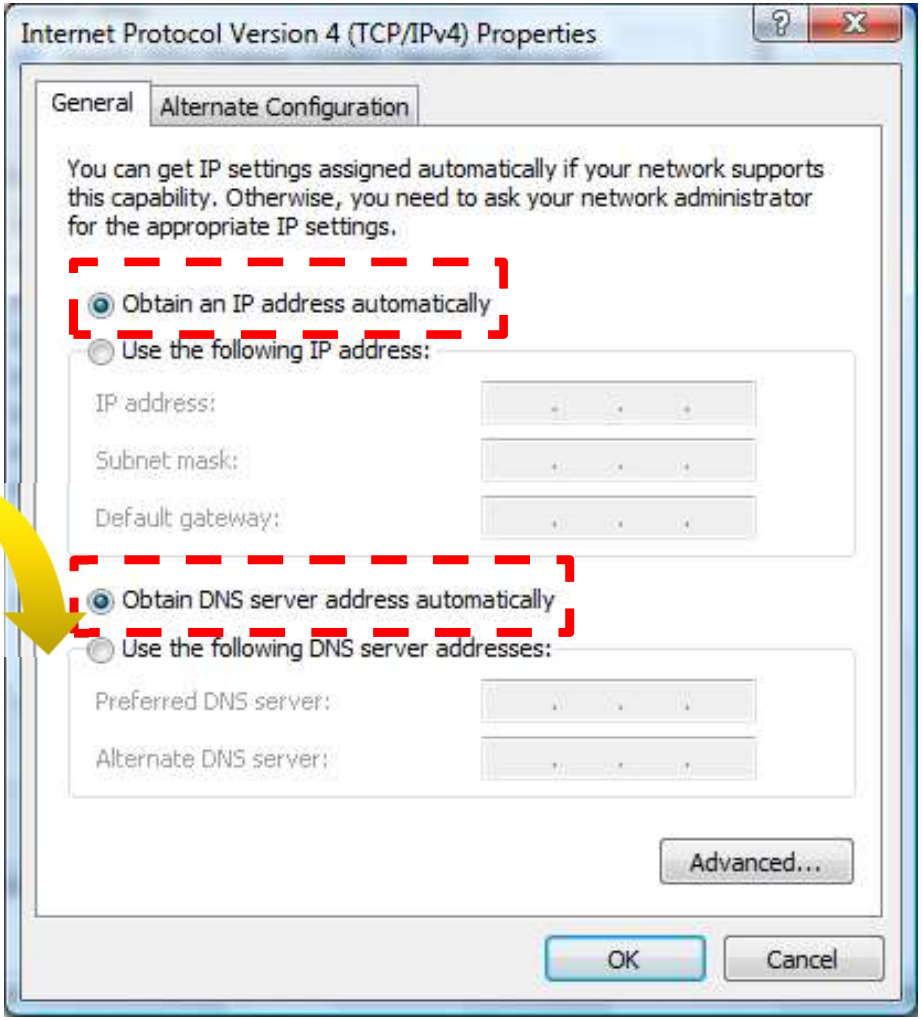
Para asignar las direcciones se usa **DHCP** (RFC 2131-3396), protocolo usuario de UDP (**puerto 67**)

- El host (cliente) envía un mensaje *broadcast*: "DHCP discover"
- El server DHCP responde con un mensaje "DHCP offer"
- El host solicita una dirección IP, mensaje "DHCP request"
- El server DHCP envía la dirección IP: mensaje "DHCP ack"



Configuración de un cliente MS Windows:





6. Autoconfiguración de la capa de red: DHCP

Configuración de un cliente Linux (Fedora Core distribution):

```
# Sample /etc/sysconfig/network-scripts/ifcfg-eth0 :

DEVICE=eth0
BOOTPROTO=dhcp
HWADDR=00:0C:29:CE:63:E3
ONBOOT=yes
TYPE=Ethernet
```

Configuración de un servidor de Linux (*dhcpd*):

```
# Sample /etc/dhcpd.conf

default-lease-time 600;max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option domain-name-servers 192.168.1.1, 192.168.1.2;
option domain-name "mydomain.org";
subnet 192.168.1.0 netmask 255.255.255.0 {
  range 192.168.1.10 192.168.1.100;
  range 192.168.1.150 192.168.1.200;
}

# Static IP address assignment
host haagen {
  hardware ethernet 08:00:2b:4c:59:23;
  fixed-address 192.168.1.222;
}
```

TEMA 2
CAPA DE RED

Fundamentos de Redes

2021/2022