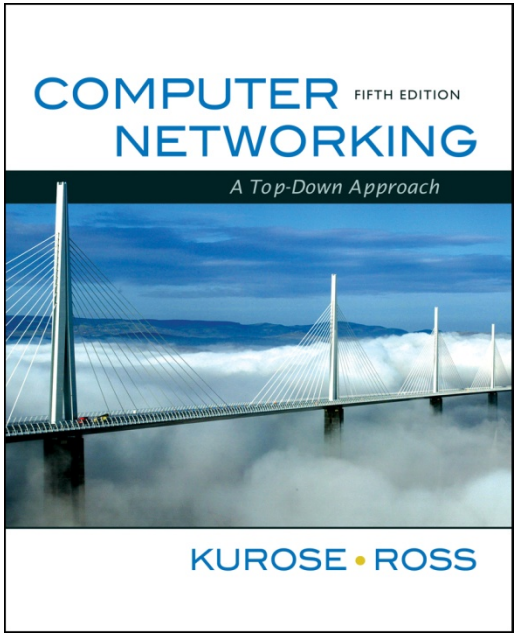
Capa de red
Nivel 3
(capítulo 4 del libro)



A note on the use of these ppt slides:

We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) in substantially unaltered form, that you mention their source (after all, we'd like people to use our book!)
- If you post any slides in substantially unaltered form on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR

Computer Networking: A Top Down Approach
5th edition.
Jim Kurose, Keith Ross
Addison-Wesley, April 2009.

Capítulo 4: Capa de red

Objetivos:

- Entender los servicios que da la capa de red:
  - Modelos (CV y datagramas)
  - Reenvío o conmutación (forwarding) frente a enrutado (routing)
    - Como funciona un router
  - Algoritmos de enrutado
  - Broadcast, multicast
- Protocolo de Internet (IP)

Capítulo 4: Capa de red

4. 1 Introducción

4.2 Circuitos virtuales y datagramas

4.3 Que hay dentro de un router

4.4 IP: Protocolo de Internet

- Formato
- Direccionamiento IPv4
- DHCP y NAT
  - Accesibilidad tras NAT
- ICMP
- IPv6

4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector distancia
- Enrutamiento jerárquico

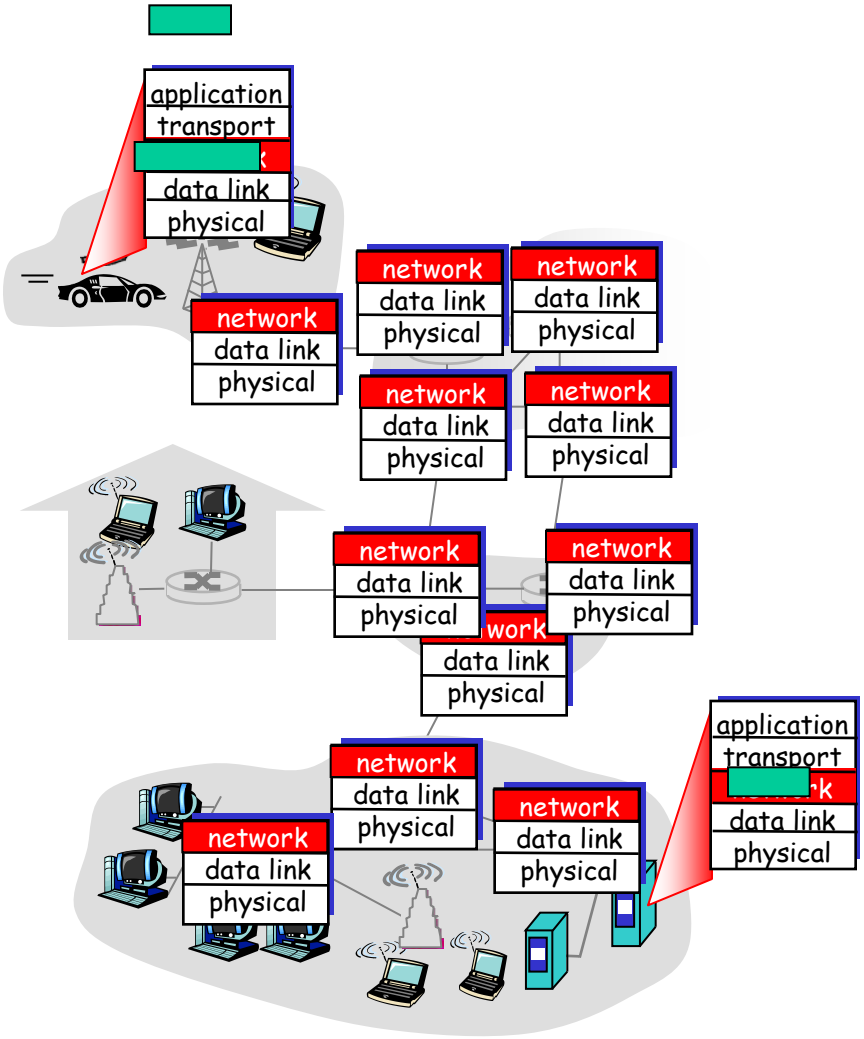
4.6 Enrutamiento en Internet

- RIP
- OSPF
- BGP
- Tipo de relaciones

4.7 Enrutamiento Broadcast y multicast

Capa de red

- **Hacer llegar segmentos entre equipos no adyacentes**
- El lado emisor añade información (cabeceras)
- En recepción, se pasa la información a la capa de transporte
- Cada equipo final implementa la capa de red y superiores, los routers solo hasta la capa de red
- Los routers analizan las cabeceras para determinar el siguiente salto
- Rec: PDU (*protocol data unit*)



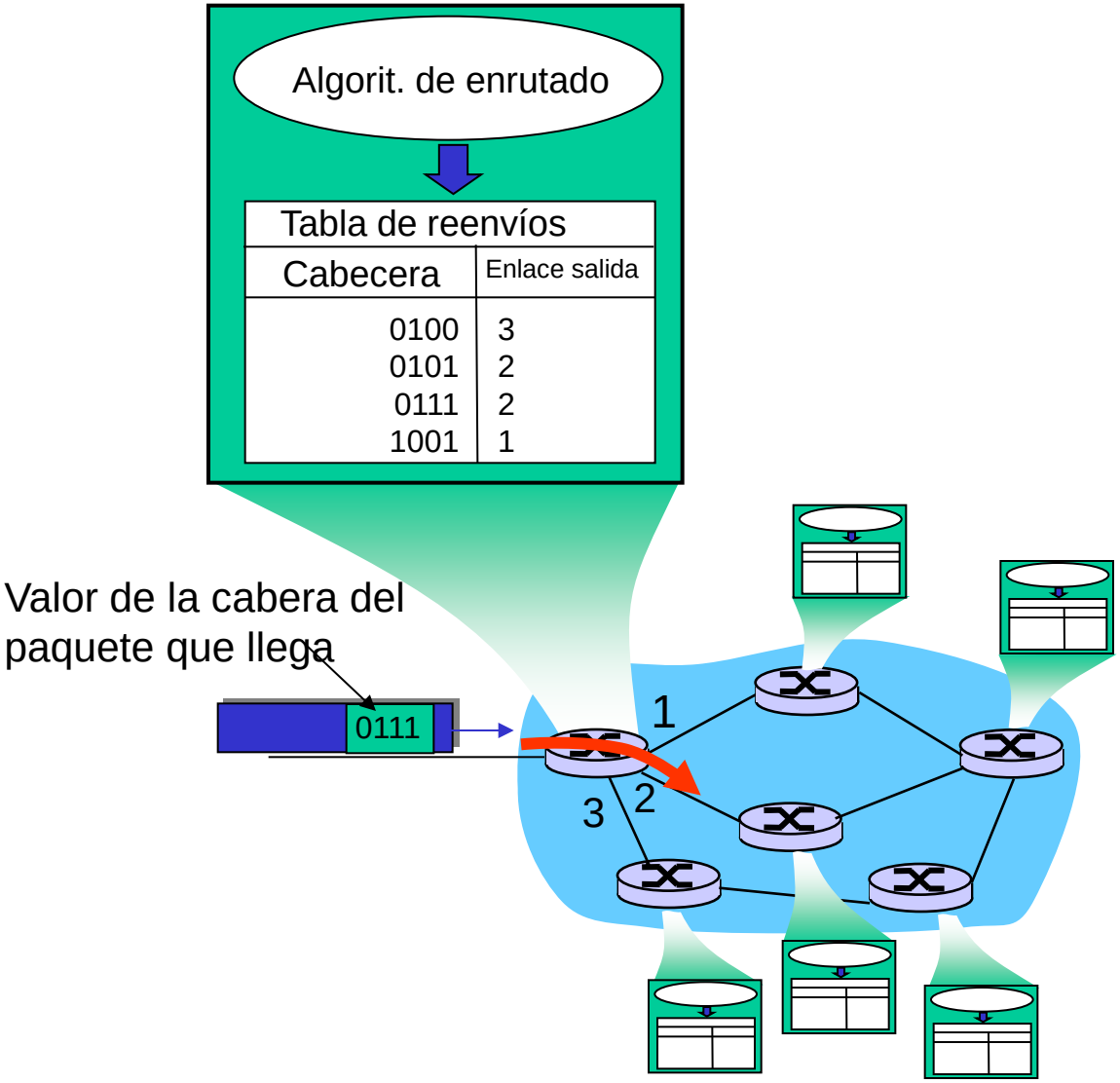
La funcionalidad clave de la capa de red

Analogía:

- Reenvío *(o conmutación)*: : proceso de realizar un cambio de carretera en un nudo
- Enrutamiento *(routing)*: proceso de planificar un viaje

- *Mover datagramas* entre equipos no adyacentes:*
  - *Reenvío* mover paquetes desde las interfaces de entrada a las apropiadas de salida del router
    - Diseño del hardware
  - *Enrutamiento* determinar que interfaz de salida debe ser elegida dado el destino del paquete
    - *Algoritmos de enrutado*
    - *Direccionamiento*
    - *Protocolos de administración (ICMP)*

Interacción entre reenvío y enrutado



Establecimiento de la conexión

- 3º importante (sub)función en algunas arquitecturas de redes:
  - Antes del flujo de datagramas, los dos máquinas finales y los routers implicados en la ruta establecen una conexión
  - Se requiere colaboración por tanto de los routers
  - ATM, frame relay, X.25

Modelos de servicios en redes

Q: ¿Que servicios se pueden ofrecer al transportar datagramas desde el emisor al receptor?

[compromisos a la hora de hacer llegar bytes que es el servicio fundamental]

Ejemplos de servicios para datagramas individualmente:

- Garantía de entrega
- Garantía de entrega con un retardo menos de 40 ms
- [...]

Ejemplos de servicios para flujos de datagramas:

- Entrega en orden de los datagramas
- Ancho de banda mínimo garantizado para el flujo
- Restricciones en los tiempos entre paquetes (*jitter*)

Modelos de servicios en redes:

Arquitecura Red	Modelos Servicio	Garantias ?				Información Congestión
		Bandwidth	Perd.	Orden	Tiempo	
Internet	best effort	none	no	no	no	no (inferida perdidas*)
ATM	CBR	constant rate	yes	yes	yes	congestion no
ATM	VBR	guaranteed rate	yes	yes	yes	congestion yes
ATM	ABR	guaranteed minimum	no	yes	no	no
ATM	UBR	none	no	yes	no	

Capítulo 4: Capa de red

4. 1 Introducción

4.2 Circuitos virtuales y datagramas

4.3 Que hay dentro de un router

4.4 IP: Protocolo de Internet

- Formato
- Direccionamiento IPv4
- DHCP y NAT
  - Accesibilidad tras NAT
- ICMP
- IPv6

4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector distancia
- Enrutamiento jerárquico

4.6 Enrutamiento en Internet

- RIP
- OSPF
- BGP
- Tipo de relaciones

4.7 Enrutamiento Broadcast y multicast

Redes de circuitos virtuales y de datagramas

- Manera de enfocar la implementación de un nivel 3
- Una red de datagramas proporciona un servicio sin conexión *host* a *host*
- Las redes de VC proporcionan un servicio orientados a la conexión
- Es análogo a la capa de enlace/transporte pero:
  - Servicio: host-to-host (no adyacentes, no procesos)
  - No elección: la red proporciona uno u otro
  - Implementación: en los routers del núcleo de la red
  - Análogo conmutación paquetes /conmutación de circuitos (sin canal dedicado)

Circuitos virtuales (CV/VC)

Similar a un circuito telefónico

- Procesos de configuración (**setup**) por cada conexión antes de que los datos puedan ser transmitidos
- Cada paquete transporta un identificador de VC (que no es el la dirección del host destino [que sí es necesario en el setup])
- Cada router del camino (*path*) mantiene información del estado de cada conexión que lo atraviesa
- Recursos de los enlaces y los routers (ancho de banda, buffers) pueden ser reservados para el VC (recursos dedicados proporcionan QoS adecuado para servicios predecibles)

Implementación de VC

Un VC consiste de:

1. Camino (*path*) desde el origen al destino
2. Número de VC, un número por cada enlace a lo largo del camino
3. Entradas en la tabla de reenvíos en cada router del camino

- Los paquetes transportan el número del VC al cual pertenecen (y no la dirección destino)
- El número de VC típicamente cambia en cada enlace (no centralizado y tamaño limitado)
  - El nuevo número de VC tras atravesar un salto es asignado por la tabla de reenvíos

VC tabla de reenvios

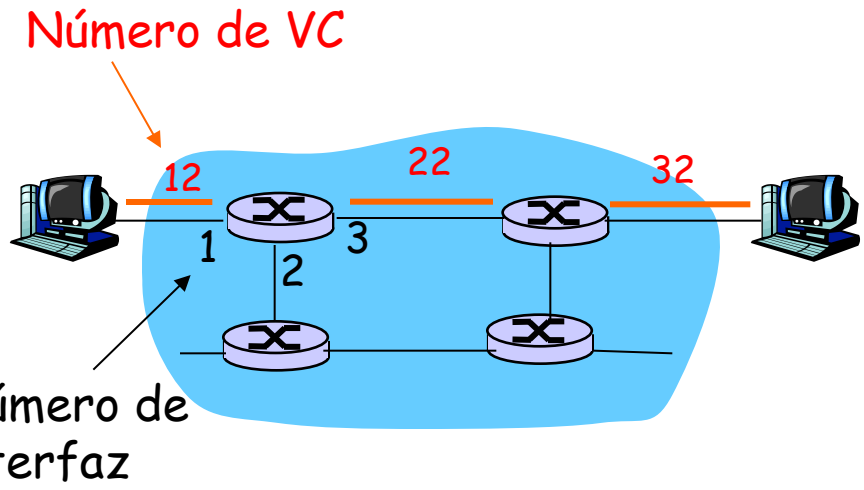


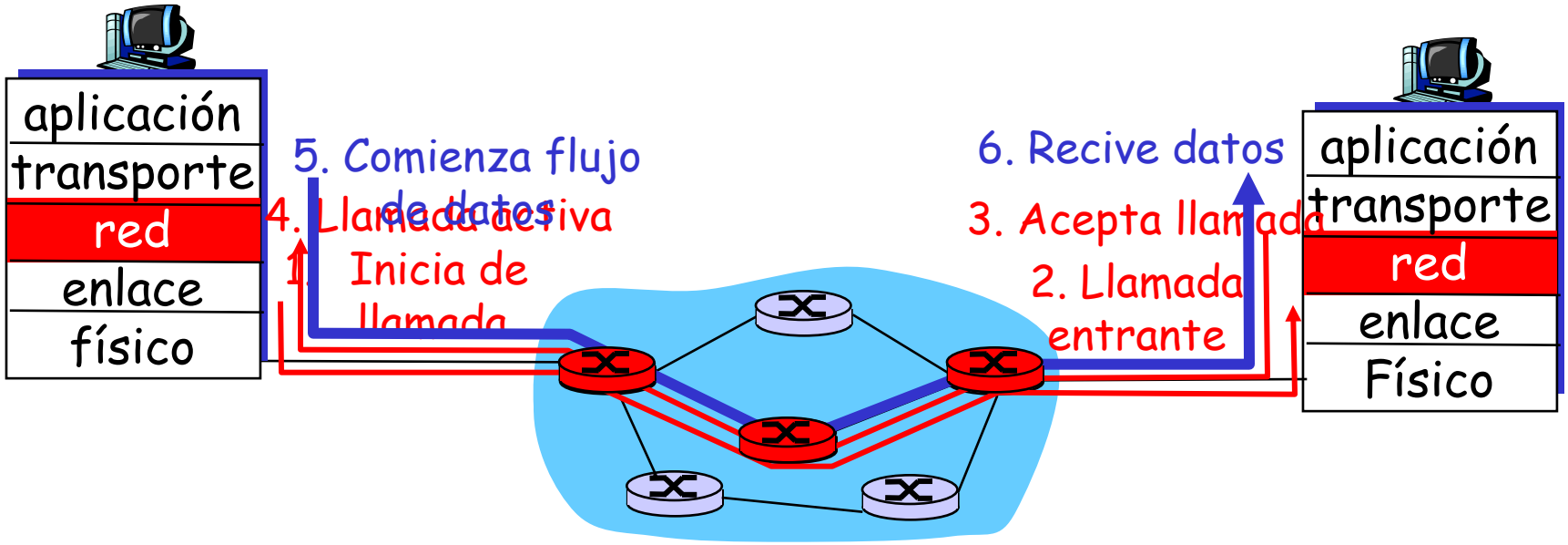
Tabla de reenvios en el router (arriba-izq)

Interface entrada	#VC de entrada	Interface de salida	#VC salida
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
…	…	…	…

Los router mantienen información del estado de la conexión

Circuitos virtuales: protocolos de señalización

- Necesarios para inicialización y finalización de un VC: configuración, transferencia y terminación
- Usado en ATM, frame-relay, X.25
- Nuestro punto de vista genérico
- No usado en IP



Redes de datagramas

- **No inicialización de conexión** al nivel de la capa de red
- Los routers no saben nada de las conexiones extremo-a-extremo (end-to-end/host-to-host)
  - No existe concepto de conexión (a este nivel)
- Los paquetes se reenvían usando la dirección de la máquina destino
  - Por tanto **los paquetes pueden tomar distintos caminos**

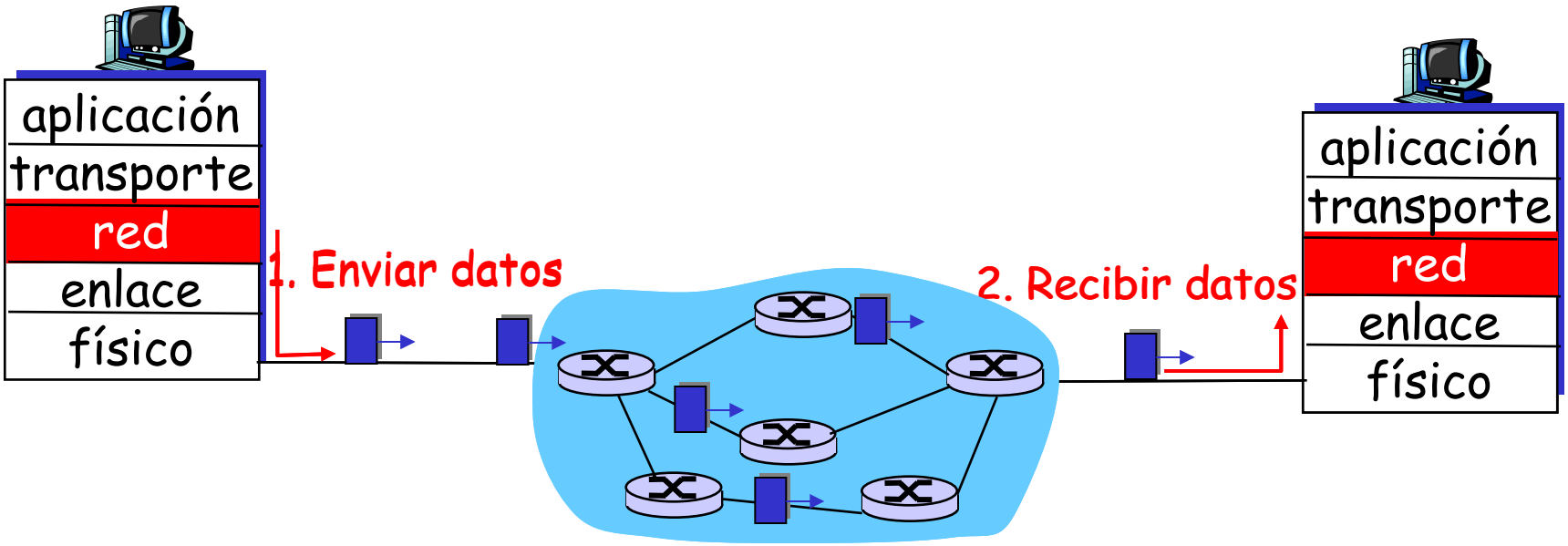


Tabla de reenvio en red de datagramas

En concreto en IP 4000 millones de direcciones, de modo que listar cada una de ellas no es una opción - Rango de direcciones

Dirección IP destino in la cabecera del paquete recien llegado

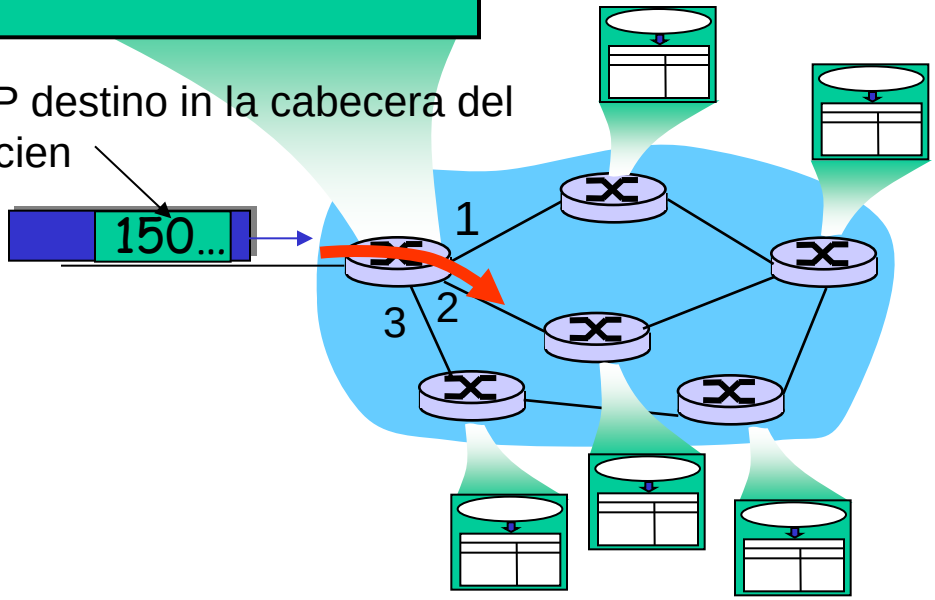


Tabla de reenvio de datagramas

Rango de direcciones	Interface de red
11001000 00010111 00010000 00000000 hasta **11001000 00010111 00010111 11111111**	0
11001000 00010111 00011000 00000000 hasta **11001000 00010111 00011000 11111111**	1
11001000 00010111 00011001 00000000 hasta **11001000 00010111 00011111 11111111**	2
En otro caso	3

Q: ¿Pero que pasa si los rangos no se dividen tan cómodamente?

Coincidencia del prefijo más largo

Coincidencia del prefijo más largo

Cuando se busca en la tabla de reenvíos un dirección destino dada, se suele imponer el prefijo *más largo* que coincide con esa dirección en concreto

Rango direcciones destino	Interface de red
11001000 00010111 00010*** ********	0
11001000 00010111 00011000 ********	1
11001000 00010111 00011*** ********	2
En otro caso	3

Ejemplos:

DA: 11001000 00010111 00010110 10100001 ¿Que interface? :

DA: 11001000 00010111 00011000 10101010 ¿Que interface? :

DA: 11001000 00010111 00111000 10101010 ¿Que interface? :

Redes de Datagramas o VC: ¿por qué? ¿Qos?

Datagramas (e.g. IP)

- Datos intercambiados por ordenadores
  - **Servicio flexible/elástico**
- Sistemas finales inteligentes
  - Capaces de ejecutar control sobre el rendimiento y **recuperarse de errores**
- Más flexible ante cambios en los niveles inferiores al ser más simple
- IntServ/DiffServ

VC (e.g. ATM)

- Evolución de la telefonía
- Adecuado para llamadas telefónicas:
  - **Servicio predecible**
- **Señalización inicial** (¿pesada?)
- **Menos robusta ante fallos**
- Sistemas finales "no inteligentes"
  - Teléfonos
  - La complejidad se deja en la red

Capítulo 4: Capa de red

4. 1 Introducción

4.2 Circuitos virtuales y datagramas

4.3 Que hay dentro de un router

4.4 IP: Protocolo de Internet

- Formato
- Direccionamiento IPv4
- DHCP y NAT
- ICMP
- IPv6

4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector distancia
- Enrutamiento jerárquico

4.6 Enrutamiento en Internet

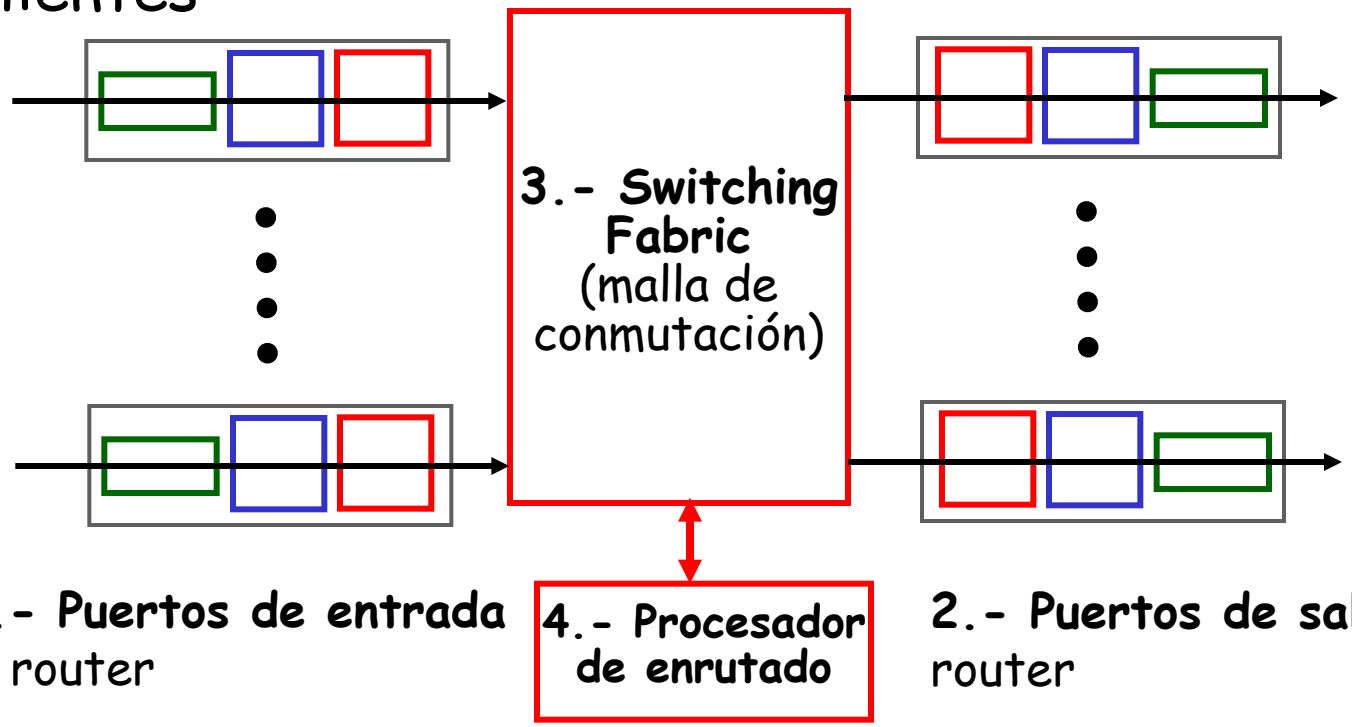
- RIP
- OSPF
- BGP
- Tipo de relaciones

4.7 Enrutamiento Broadcast y multicast

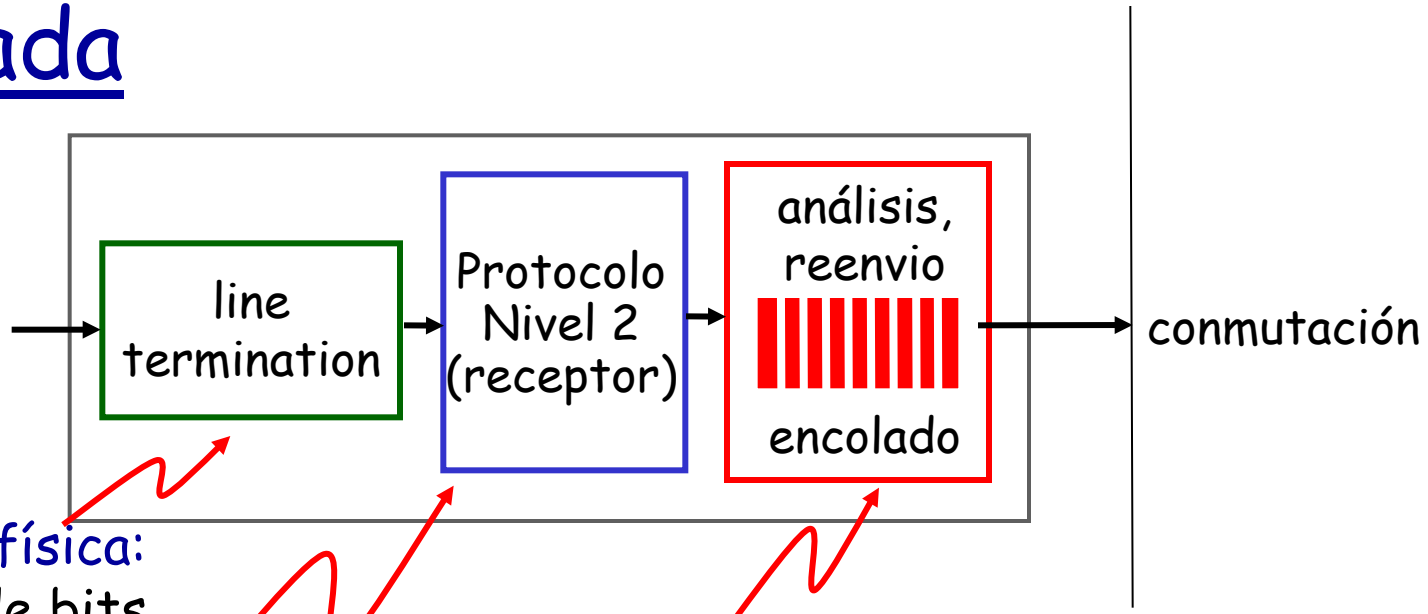
¿Cómo funciona un router?

Dos funciones clave:

- Ejecutar algoritmos/protocolos de **enrutado** (RIP, OSPF, BGP)*
- **Reenvío** de datagramas de los enlaces entrantes y salientes



Funcionalidades de los puertos de entrada



Capa física:
Recepción de bits

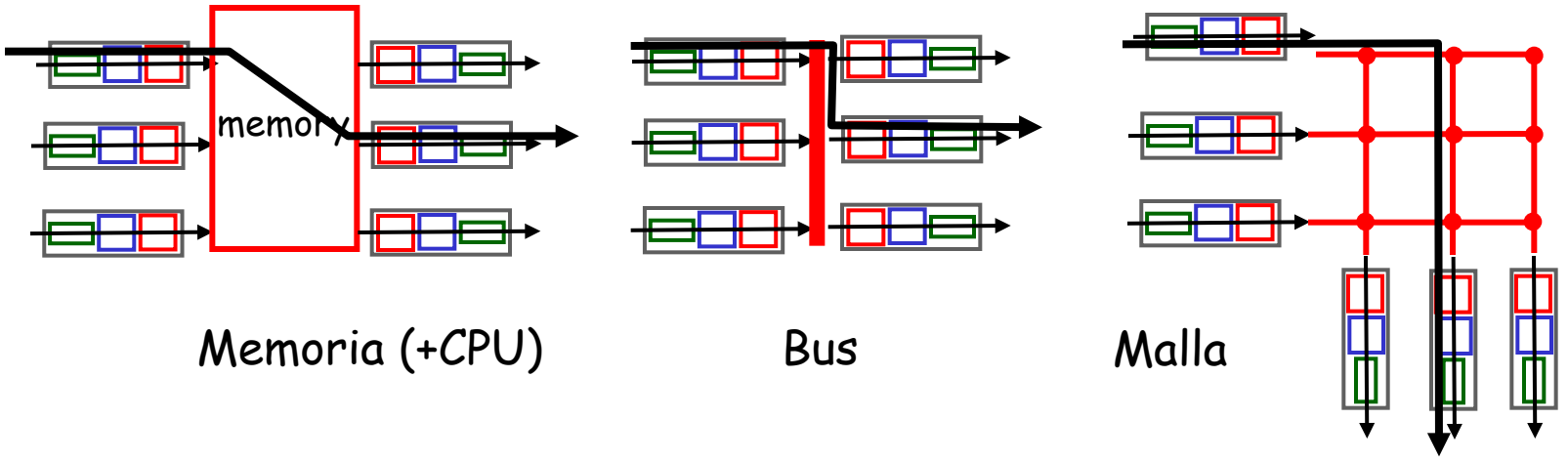
Capa de enlace:
Ethernet (por ejemplo)

Conmutación (típicamente descentralizada tabla reenvíos local*):

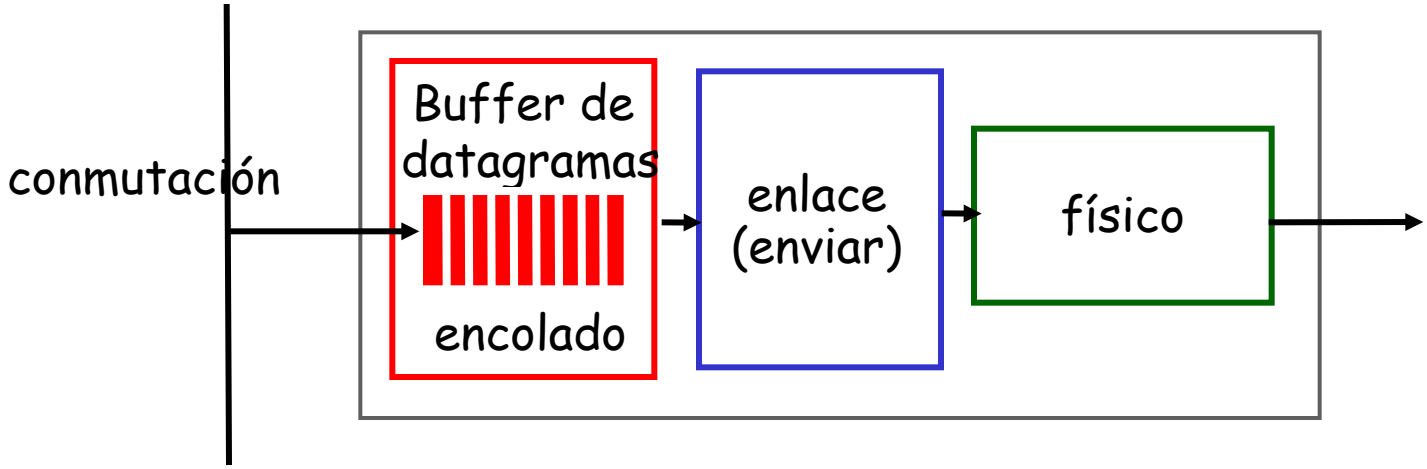
- **Dada una dirección destino buscar el puerto de salida adecuado** y copiarlo allí
- **Objetivo: completar el proceso a tasa de línea** ('line speed')
- **Encolado**: Si los datagramas llegan más rápido que la tasa de reenvío estos se encolaran en buffers de entrada

Tipos de mallas (entramado) de conmutación

- Transferir paquetes desde el buffer de entrada al buffer de salida (bajo planificación)
- Velocidad de conmutación: tasa a la que los paquetes pueden ser transferidos desde los puertos de entrada a los de salida
  - ¿Dadas N entradas: velocidad de conmutación debería ser al menos N veces la tasa de línea por enlace?

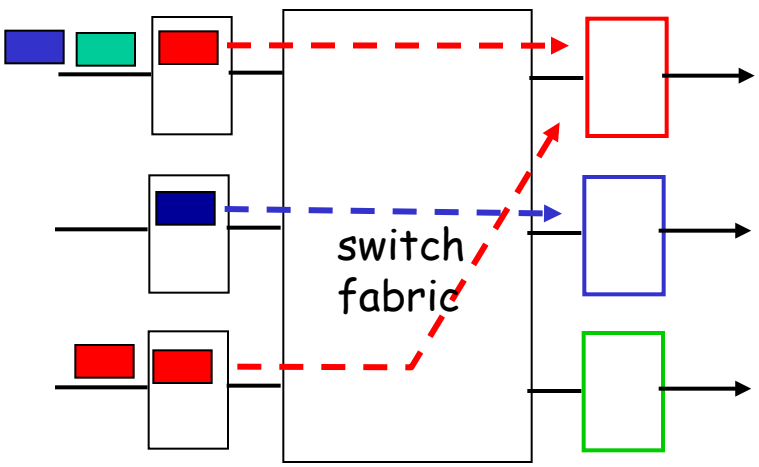


Puertos de salida

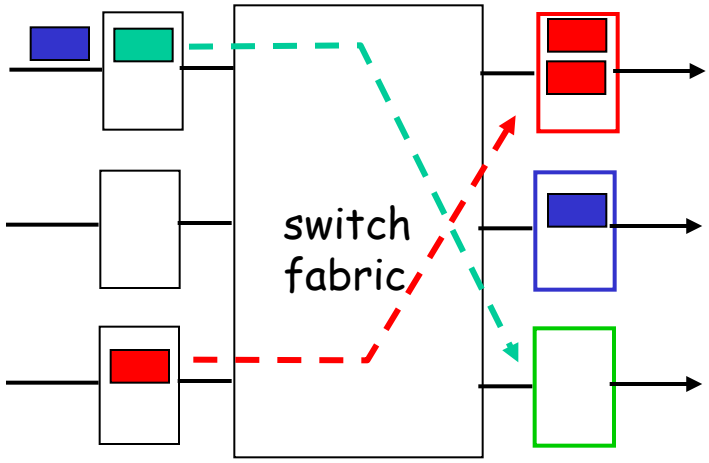


- *Se requieren buffers* cuando los datagramas llegan más rápido que la velocidad de transmisión
- *Planificador de paquetes* debe determinar que paquete de los buffers de entrada transmitir en instante de tiempo dado

Puertos de salida



En t, varios paquetes
por ser conmutados



Despúes de conmutar
un paquete

- *Buffering* cuando la velocidad de conmutación (y la tasa de llegadas temporalmente también) es mayor que la tasa de línea de salida
- *Encolado (retardo) y perdidas debido a que el buffer del puerto de salida se desborda: QoS*

¿Cuánto buffer?

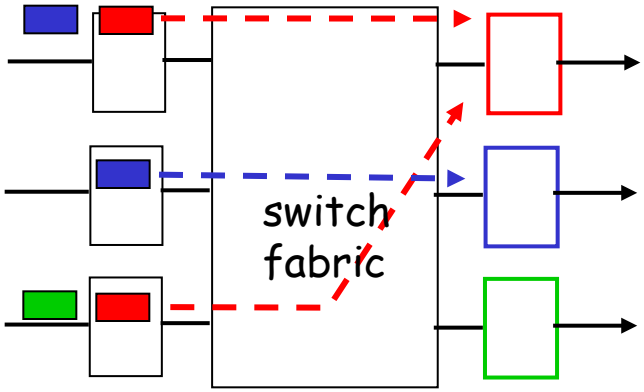
- RFC 3439 "rule of thumb": buffer equivalente a la capacidad del enlace (C) dividida entre RTT típico (250 ms)
  - e.g., C = 10 Gpbs link: 2.5 Gbit buffer
- Otras recomendaciones (N flujos):

$$\frac{RTT \cdot C}{\sqrt{N}}$$

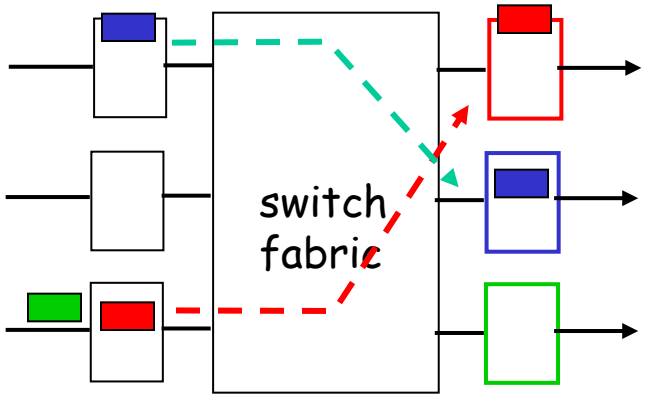
- RED: Detección aleatoria temprana

Encolado en el puertos de entrada

- Conmutación más lenta que el agregado de la tasa de llegadas: el encolado puede suceder en los puertos de entrada
  - *Retardo y perdidas (buffer overflow!)*
- Bloqueo Head-of-the-Line (HOL) : Datagramas bloqueados en la primera posición de un buffer, bloquean a su vez otros datagramas



Disputa por el Puerto de salida



Un tiempo de paquete después: paquete verde experimenta bloqueo HOL

Problema 10

Coincidencia prefijo	Interfaz
00	0
010	1
011	2
10	2
11	3

Problema 11

Coincidencia prefijo	Interfaz
1	0
10	1
111	2
Otro caso	3

Capítulo 4: Capa de red

4. 1 Introducción

4.2 Circuitos virtuales y datagramas

4.3 Que hay dentro de un router

4.4 IP: Protocolo de Internet

- Formato
- Direccionamiento IPv4
  - DHCP y NAT
    - Accesibilidad tras NAT
- ICMP
- IPv6

4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector distancia
- Enrutamiento jerárquico

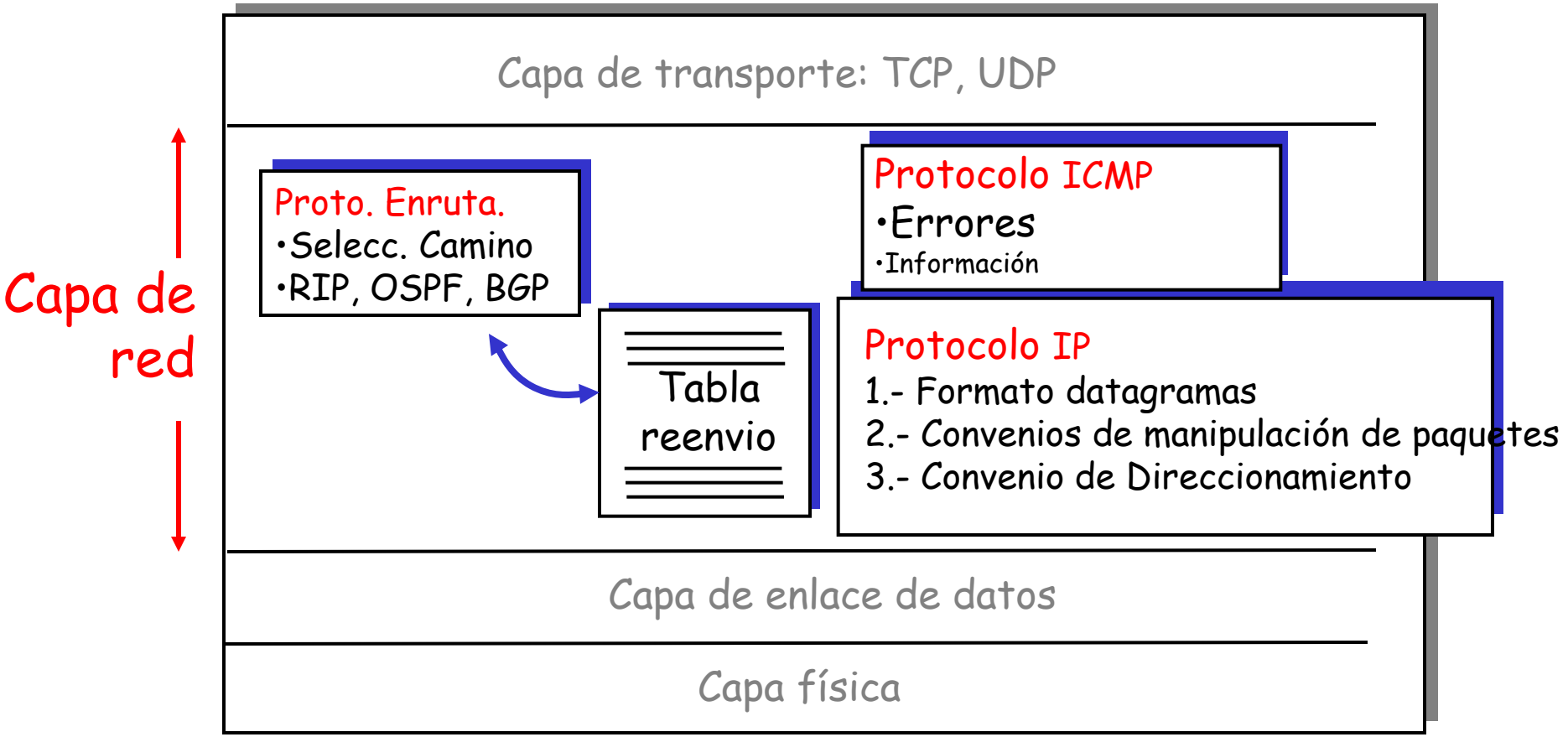
4.6 Enrutamiento en Internet

- RIP
- OSPF
- BGP
- Tipo de relaciones

4.7 Enrutamiento Broadcast y multicast

Protocolo de Internet (IP)

Funcionalidades: enrutamiento (direccionamiento+protocolos enrutamiento), fragmentacion e ICMP:



Capítulo 4: Capa de red

4. 1 Introducción

4.2 Circuitos virtuales y datagramas

4.3 Que hay dentro de un router

4.4 IP: Protocolo de Internet

- Formato
- Fragmentación
- Direccionamiento IPv4
- DHCP y NAT
  - Accesibilidad tras NAT
- ICMP
- IPv6

4.5 Algoritmos de enrutamiento

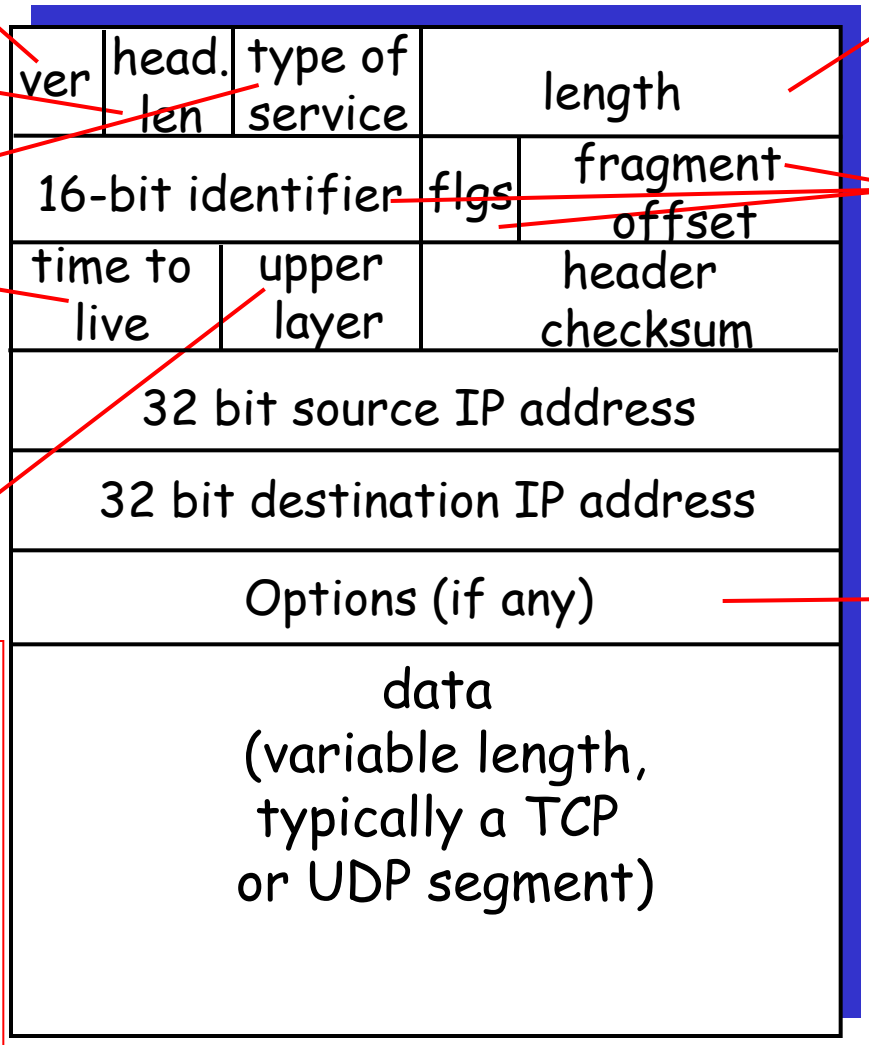
- Estado de enlaces
- Vector distancia
- Enrutamiento jerárquico

4.6 Enrutamiento en Internet

- RIP
- OSPF
- BGP
- Tipo de relaciones

4.7 Enrutamiento Broadcast y multicast

Formato datagrama IP (sintaxis y semántica)



Número de versión

Longitud cabecera (palabras de 4 bytes: 20-60B)

Tipo de servicio (DiffServ)

Tiempo de vida máximo número de saltos restantes (decremental por salto)

Protocolo de la capa superior

Longitud total del datagrama (bytes)

Fragmentación Y reesamblado (desplazamiento en bloques de 8B)

De forma opcional se pueden añadir marcas de tiempo, routers visitados…

Cuanta sobrecarga (overhead) con IP?

- 20* bytes de IP
- + Transporte (TCP==20) + capa de aplic.

Capítulo 4: Capa de red

4. 1 Introducción

4.2 Circuitos virtuales y datagramas

4.3 Que hay dentro de un router

4.4 IP: Protocolo de Internet

- Formato
- Fragmentación
- Direccionamiento IPv4
- DHCP y NAT
  - Accesibilidad tras NAT
- ICMP
- IPv6

4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector distancia
- Enrutamiento jerárquico

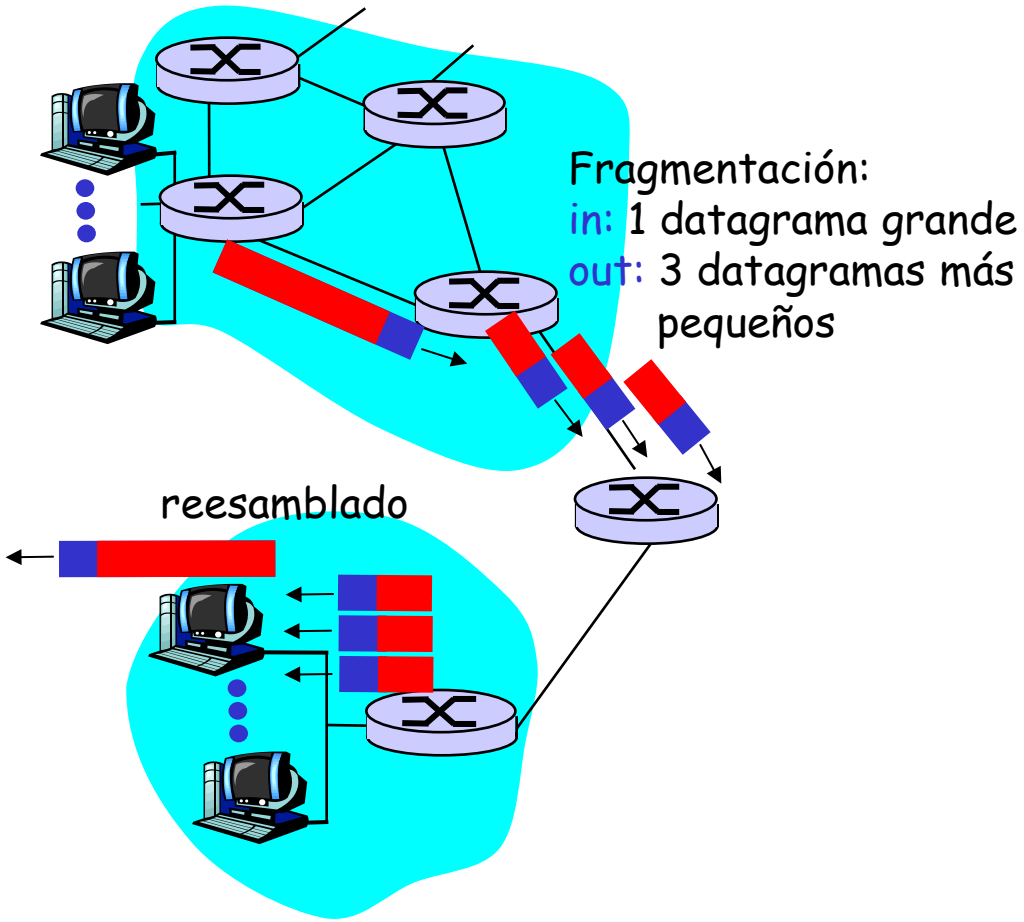
4.6 Enrutamiento en Internet

- RIP
- OSPF
- BGP
- Tipo de relaciones

4.7 Enrutamiento Broadcast y multicast

IP Fragmentación y reesamblado

- Los enlaces de las redes tiene definida una **MTU** (max. transfer unit)- tamaño máximo a transportar por el nivel:
  - Diferentes tipos de enlaces: diferentes MTUs
- Datagramas IP grandes son divididos ("fragmentados") en algún router de la red
  - Un datagrama pasa a ser varios datagramas
  - El ensamblado se realiza solo en el destino final
  - Varios de los campos de IP están destinados para identificar y ordenar fragmentos que originariamente fueron un solo datagrama



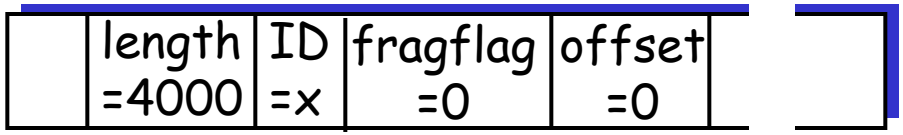
IP Fragmentación y reesamblado

Ejemplo

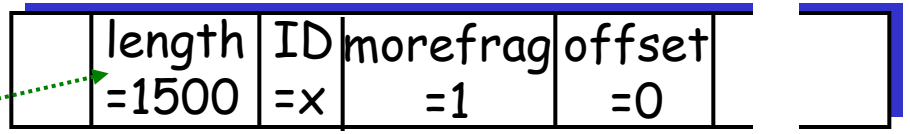
- 4000 byte de datagrama, 20B de cabecera
- MTU = 1500 bytes

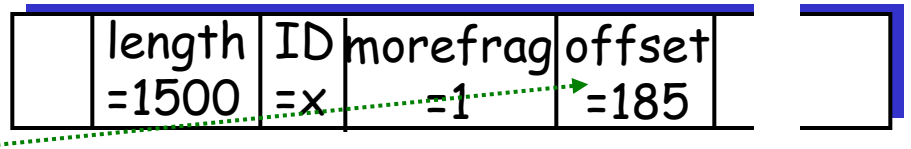
1480 bytes en el campo de datos

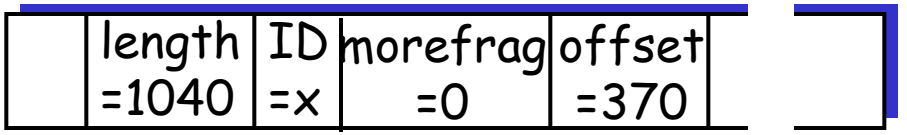
offset = 1480/8
¡Multiplos de 8B!
(menos el último)



Un datagrama grande se convierte en varios más pequeños, siempre del tamaño máximo posible (en IP)







Fragmentación: pros y contras

- Independecia y flexibilidad de niveles inferiores
- Trabajo significativo en los extremos
- Trabajo extra en los routers que fragmentan
- Debilidad ante ataques
  - Sin final de datagrama
  - Numeración incorrecta

Capítulo 4: Capa de red

4. 1 Introducción

4.2 Circuitos virtuales y datagramas

4.3 Que hay dentro de un router

4.4 IP: Protocolo de Internet

- Formato
- Fragmentación
- Direccionamiento IPv4
- DHCP y NAT
  - Accesibilidad tras NAT
- ICMP
- IPv6

4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector distancia
- Enrutamiento jerárquico

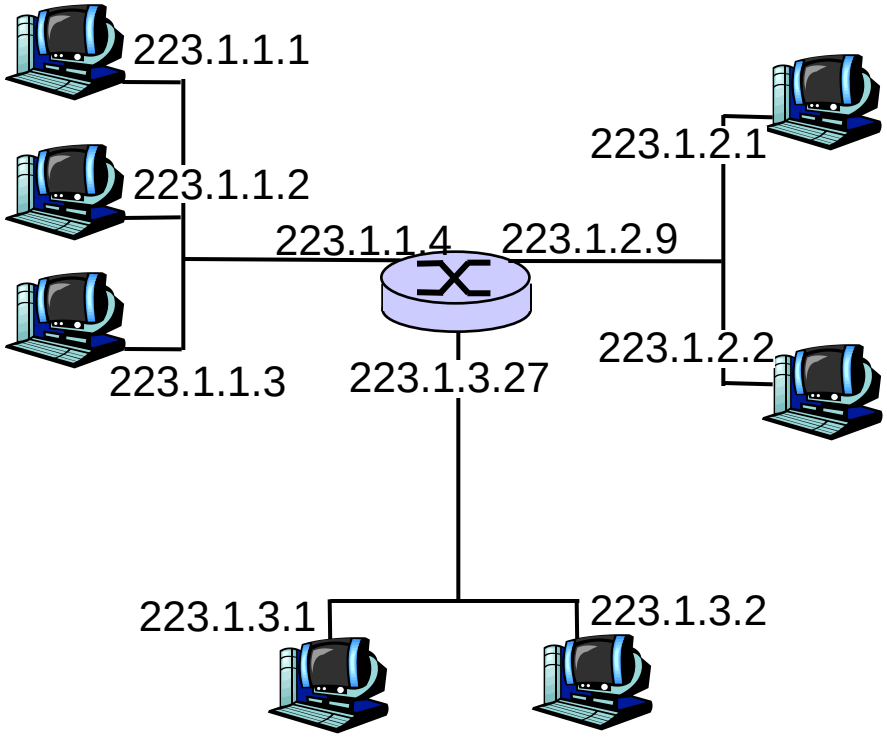
4.6 Enrutamiento en Internet

- RIP
- OSPF
- BGP
- Tipo de relaciones

4.7 Enrutamiento Broadcast y multicast

Direccionamiento IP: Introducción

- Direcciones IP: identificador de 32-bits para hosts e interfaces (¿únicos?)
- *Interface:* conexión entre los hosts y los routers con el enlace físico
  - Los routers tienen generalmente varias interfaces
  - Los hosts (PC, equipos finales) suelen tener una
  - Debe haber una dirección IP asignada a cada interface
  - Notación:
    - Decimal con punto
    - Binario
    - Entero decimal

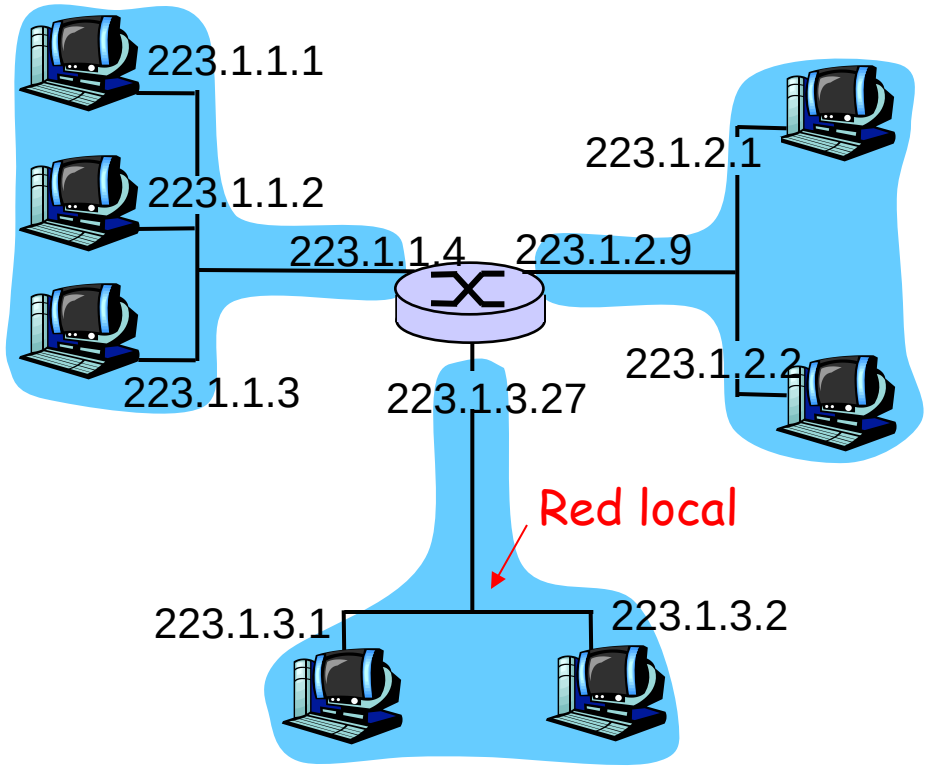


223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

Redes

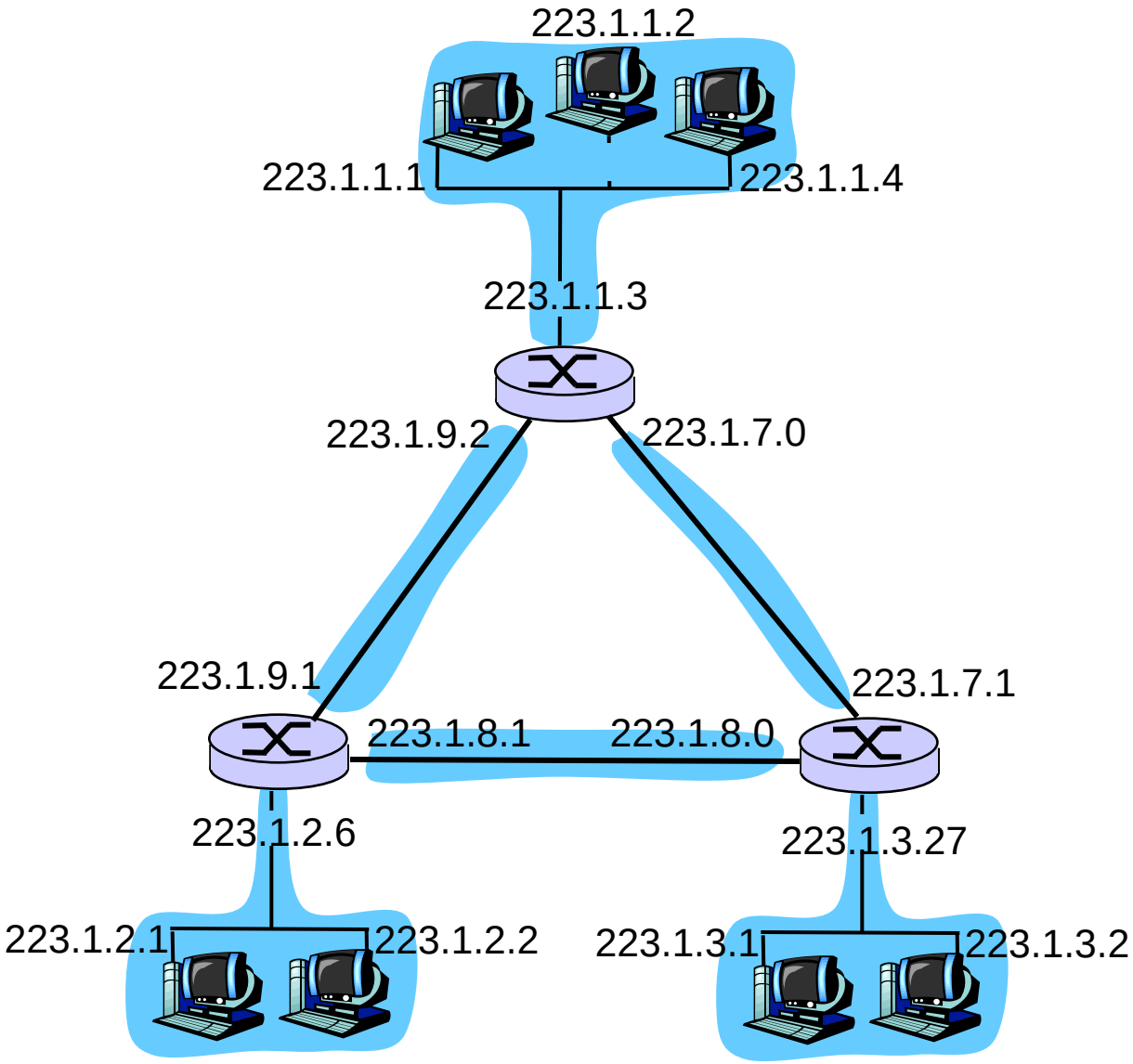
- Dirección IP:
  - Parte de red (bits altos)
  - Parte del host/interface (bits bajos)
- *¿Que es una red?*
  - Interfaces que contienen la misma parte de red en sus direcciones IP
- *¿Que es una red local?*
  - Elementos que pueden comunicarse entre ellos sin la intervención de un router



1 red compuesta a su vez de 3 (sub)redes (que son redes locales)

Redes

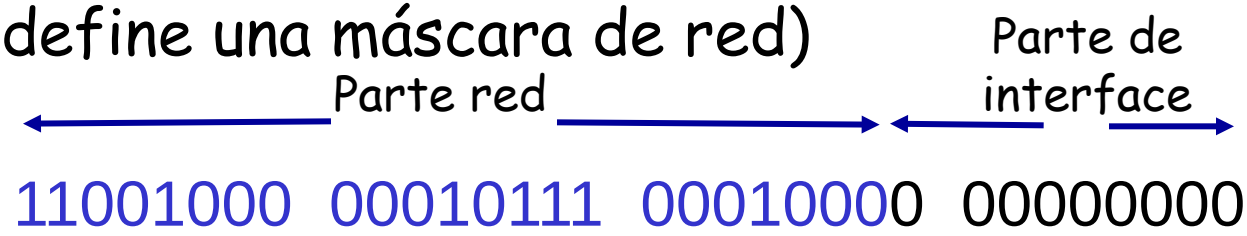
¿Cuantas?



Estrategia de definición de reangos IP: CIDR

CIDR: Enrutamiento entre dominios sin clases

- La parte de la dirección de red es de longitud arbitraria (y por tanto también los de la interface)
- Formato de las direcciones: a.b.c.d/x, donde x es el número de bits pertenecientes a la red (y define una máscara de red)



200.23.16.0/23

Notación máscaras*

- Notación CIDR
  - 200.23.16.0/23 (23 bits de red + 9 bits de interface, 512 interfaces)
- Notación Máscara en decimal
  - 255.255.254.0 (==/23)
- Notación Máscara en binario
  - 11111111 11111111 11111110 00000000 (==/23)

Direccionamiento con clase*

- A==/8 (16777216 hosts)
- B==/16 (65534 host)
- C==/24 (256)
- Muy poco flexible, las clases A y B suelen resultar grandes y las C pequeñas

Direcciones IP reservadas*

- No pueden asignarse a las interfaces direcciones que empiecen por 127.
  - Dirección *loopback*
- No asignar direcciones con campos de interface con todos los bits iguales a 0 o a 1.
  - Campo de interface de la dirección a 0, identifican rangos:
    - Ejemplos 150.244.0.0 (incorrecta si x.x.x.x/?), 150.244.65.0 (?)
  - Campo de interface de la dirección con todos los bits a 1, identifica todas las estaciones de la red. Se denominan dirección de difusión o *broadcast* :
    - Ejemplos 150.244.255.255 (?), 150.244.65.255 (?)

Direcciones Privadas*

- Conjunto de direcciones que por convenio (RFC 1918) solo se pueden usar de manera interna en una red (los router no encaminarán nunca* ese tráfico hacia el exterior)

Rango IP	# Direcciones	CIDR y máscara de subred
10.0.0.0 – 10.255.255.255	16.777.216	10.0.0.0/8 (255.0.0.0)
172.16.0.0 – 172.31.255.255	1.048.576	172.16.0.0/12 (255.240.0.0)
192.168.0.0 – 192.168.255.255	65.536	192.168.0.0/16 (255.255.0.0)

Configuración equipo IP*

- Las interfaces configuradas con IP requieren:
  - Dirección IP
  - Dirección IP del router configurado como puerta de enlace (gateway en los host se suele llamar, router predeterminado o router del primer salto)
  - Máscara de red (identificar si la IP es parte de la red local o no)
- En los host típicamente también DNS

Definiciones asignatura

- Direcciones **IP** válidas
- Direcciones **IP** pertenecen al rango/red
- Direcciones **IP** asignables
- Ejemplos:
  - **Rango/Red** 1.2.0.0/23
    - **2**.2.0.4 -> válida pero no pertenece a esta red
    - **257**.2.0.4 -> no válida
    - 1.2.**1**.**255** -> válida, sí pertenece rango, pero no asignable
    - 1.2.**0.0** -> válida, pertenece rango, pero no asignable
    - 1.2.**0.1** -> válida, pertenece rango, y asignable
    - 1.2.**0.255** -> válida, pertenece rango, y asignable
  - 192.168.1.1 -> válida y asignable detrás de un NAT
  - 127.0.0.1 -> válida pero no asignable

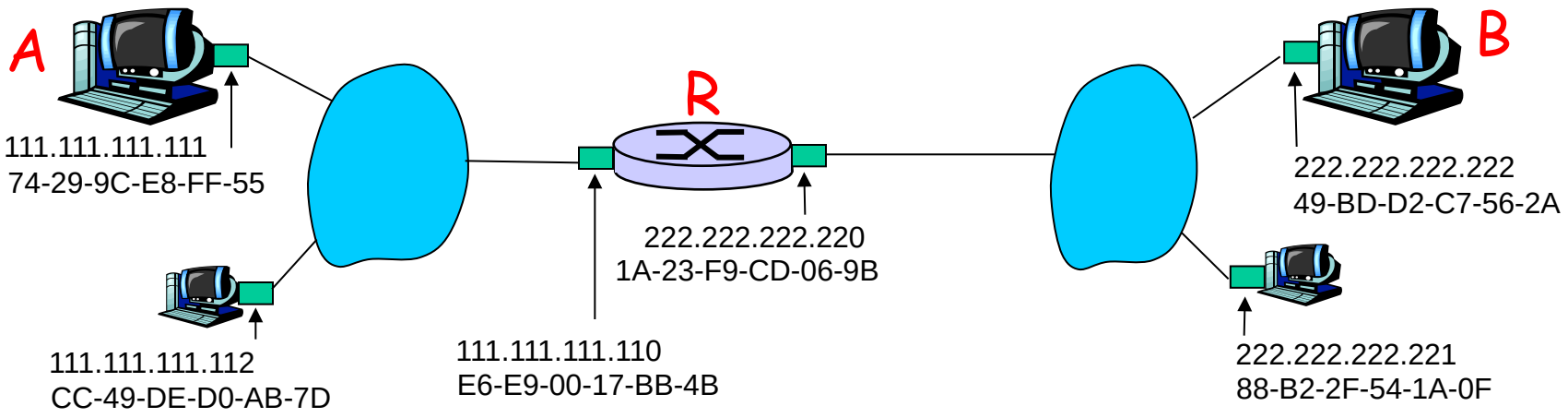
Definiciones

- Los rango IP los vamos a definir
  - Por una dirección en formato de rango + máscara
- Un rango IP asociado a una interface
  - Se forma de uno o varios rangos IP donde aplica la regla de preferencia a prefijo más largo
- Aunque de forma general/informal los rangos asociados a una interface pueden además definirse como:
    - Un conjunto con definición intensiva o extensiva
      - Ej. extensiva: {1.1.1.1, 1.1.1.2, 1.1.1.3}
      - Ej. intensiva: [Entre 1.1.1.1 y 1.1.1.3]
    - Con "comodines"

Repaso ARP.- Addressing: routing to another LAN

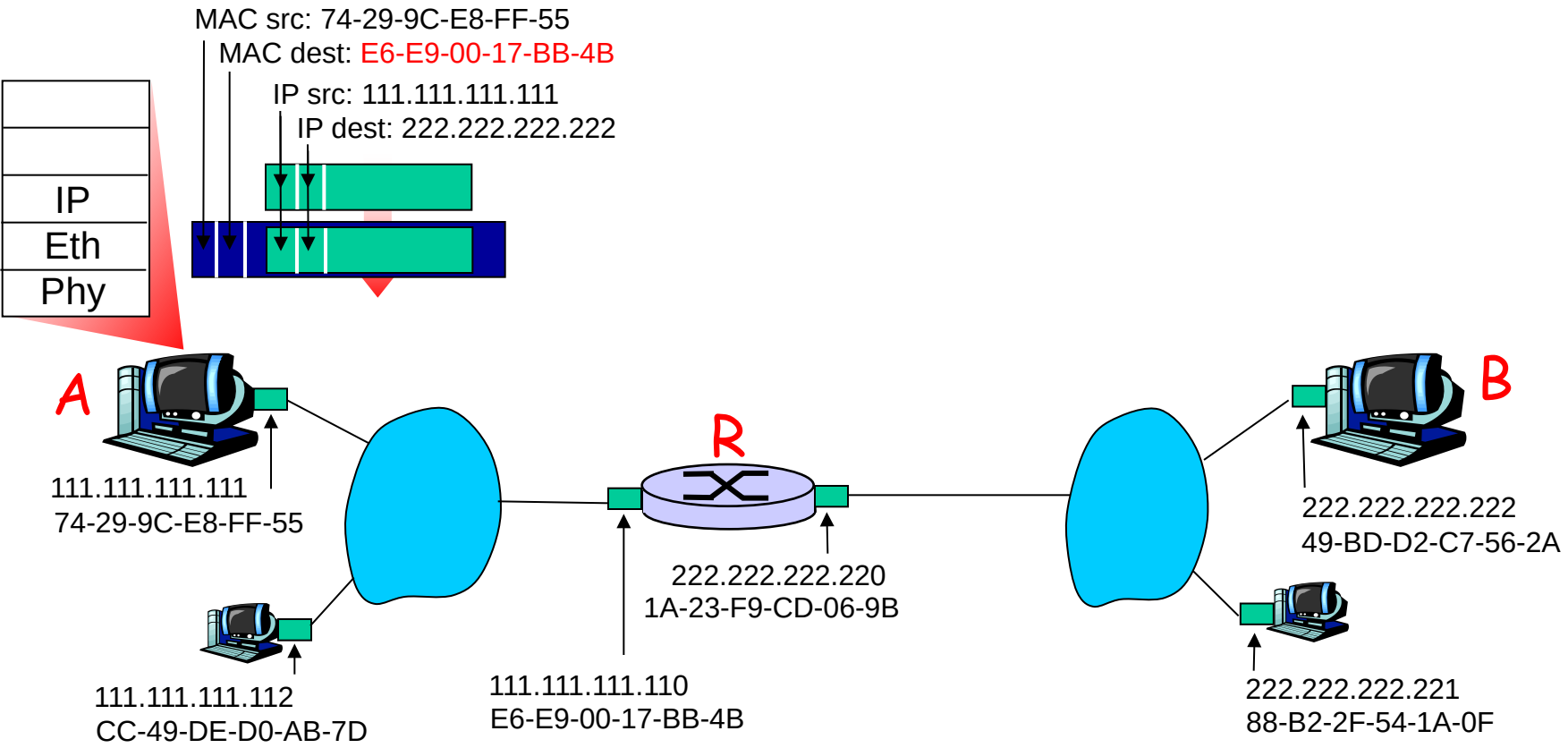
walkthrough: send datagram from A to B via R.

- focus on addressing - at both IP (datagram) and MAC layer (frame)
- assume A knows B's IP address
- assume A knows B's MAC address (how?)
- assume A knows IP address of first hop router, R (how?)
- assume A knows MAC address of first hop router interface (how?)



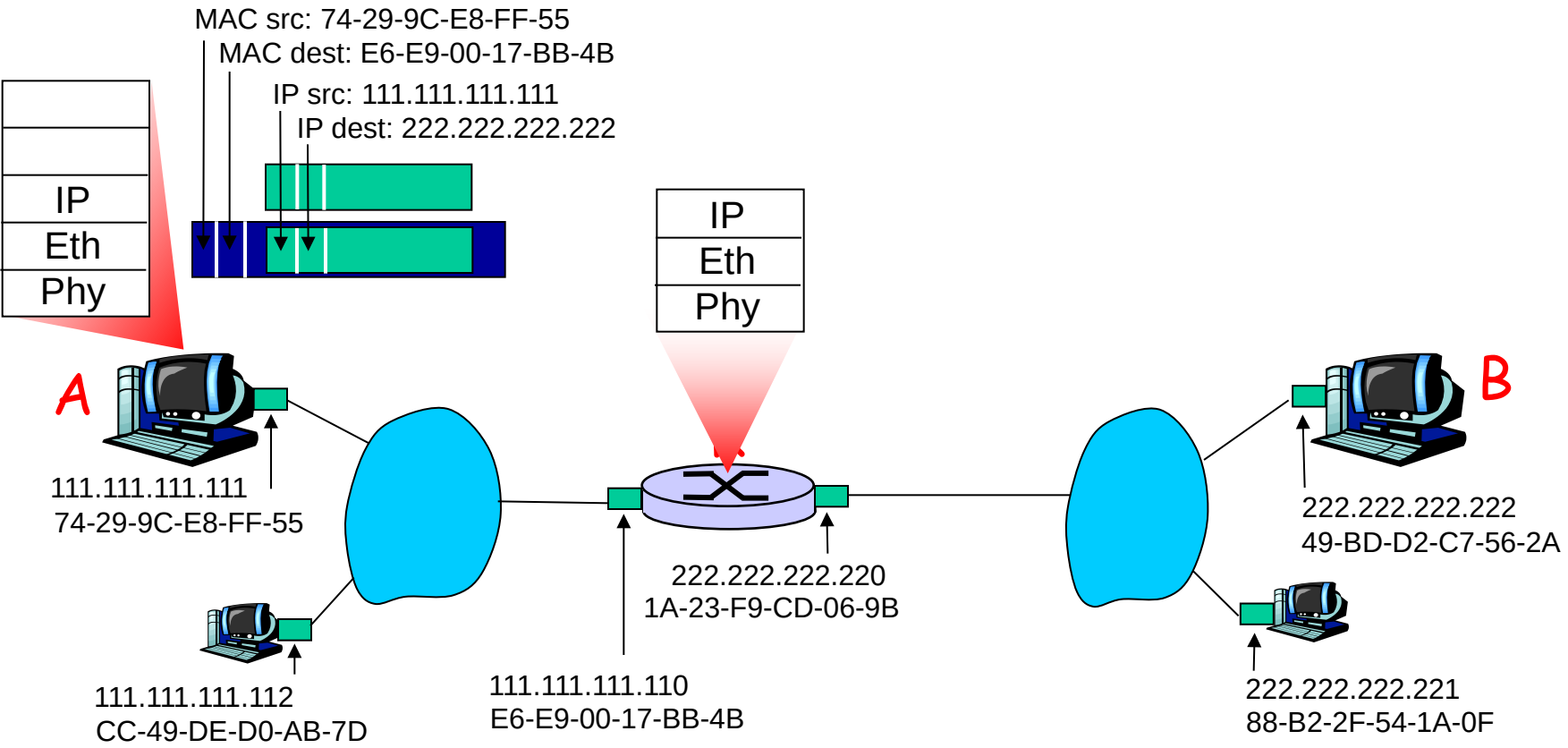
RC5. ARP : routing to another LAN

- A creates IP datagram with IP source A, destination B
- A creates link-layer frame with R's MAC address as dest, frame contains A-to-B IP datagram



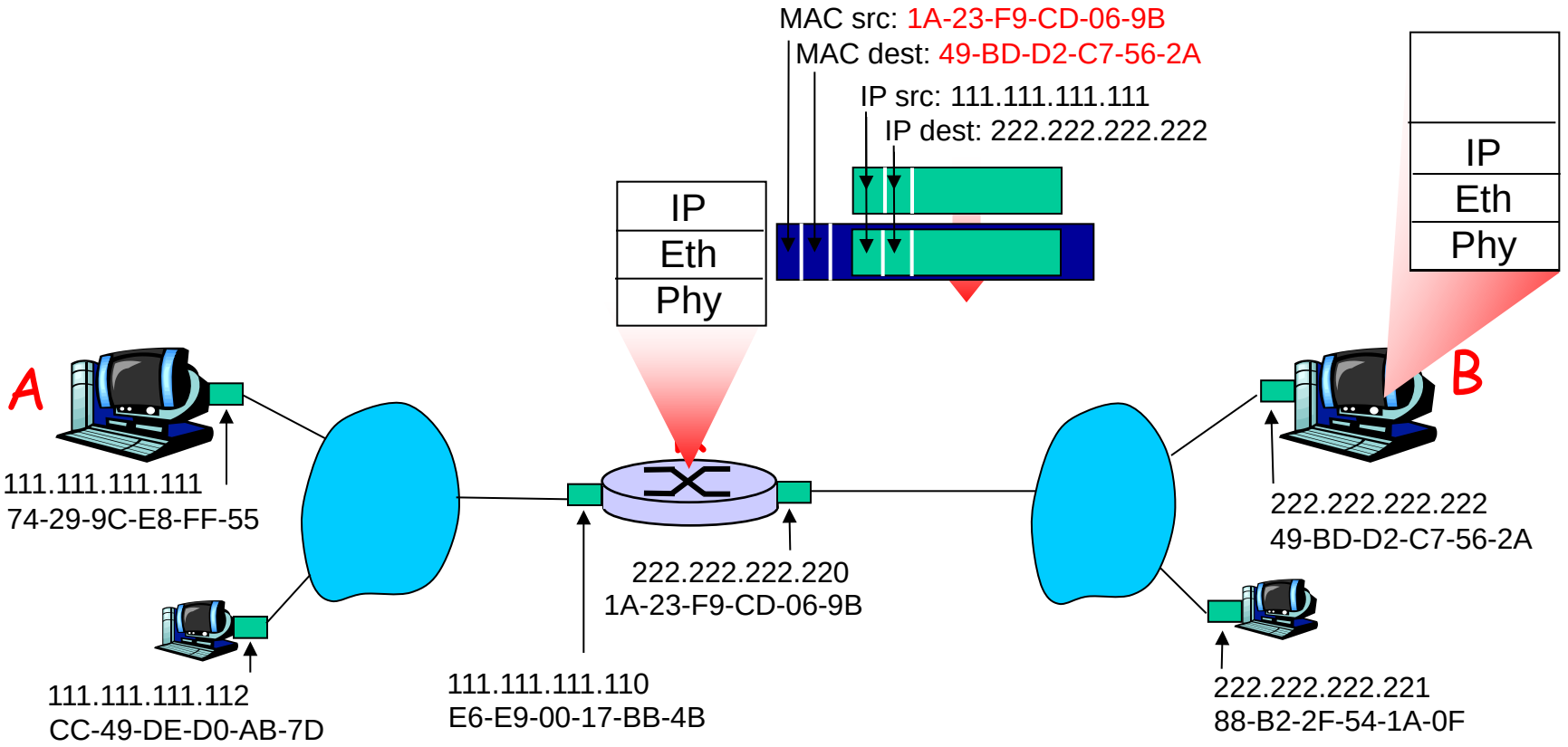
RC5. ARP : routing to another LAN

- frame sent from A to R
- frame received at R, datagram removed, passed up to IP



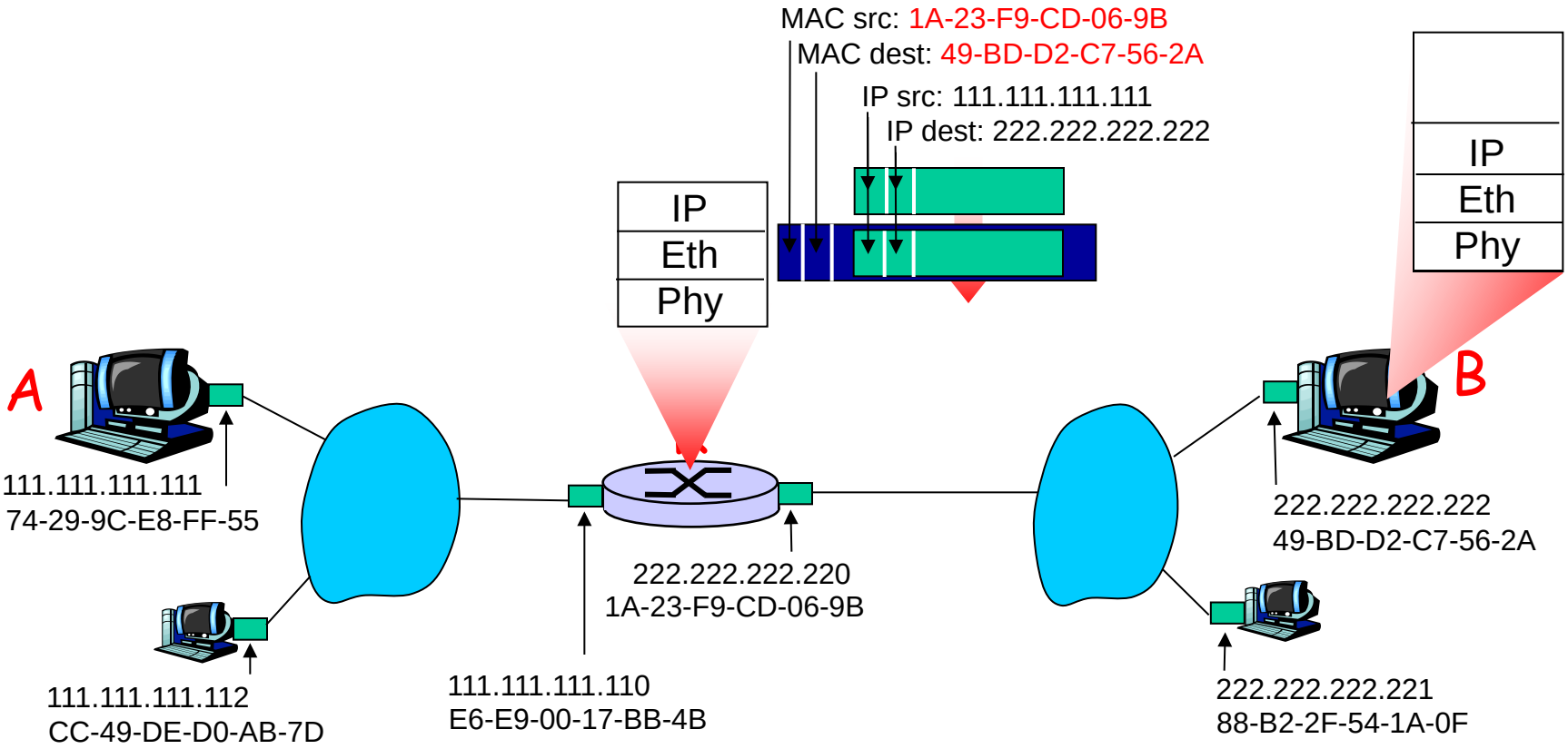
RC5. ARP : routing to another LAN

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram



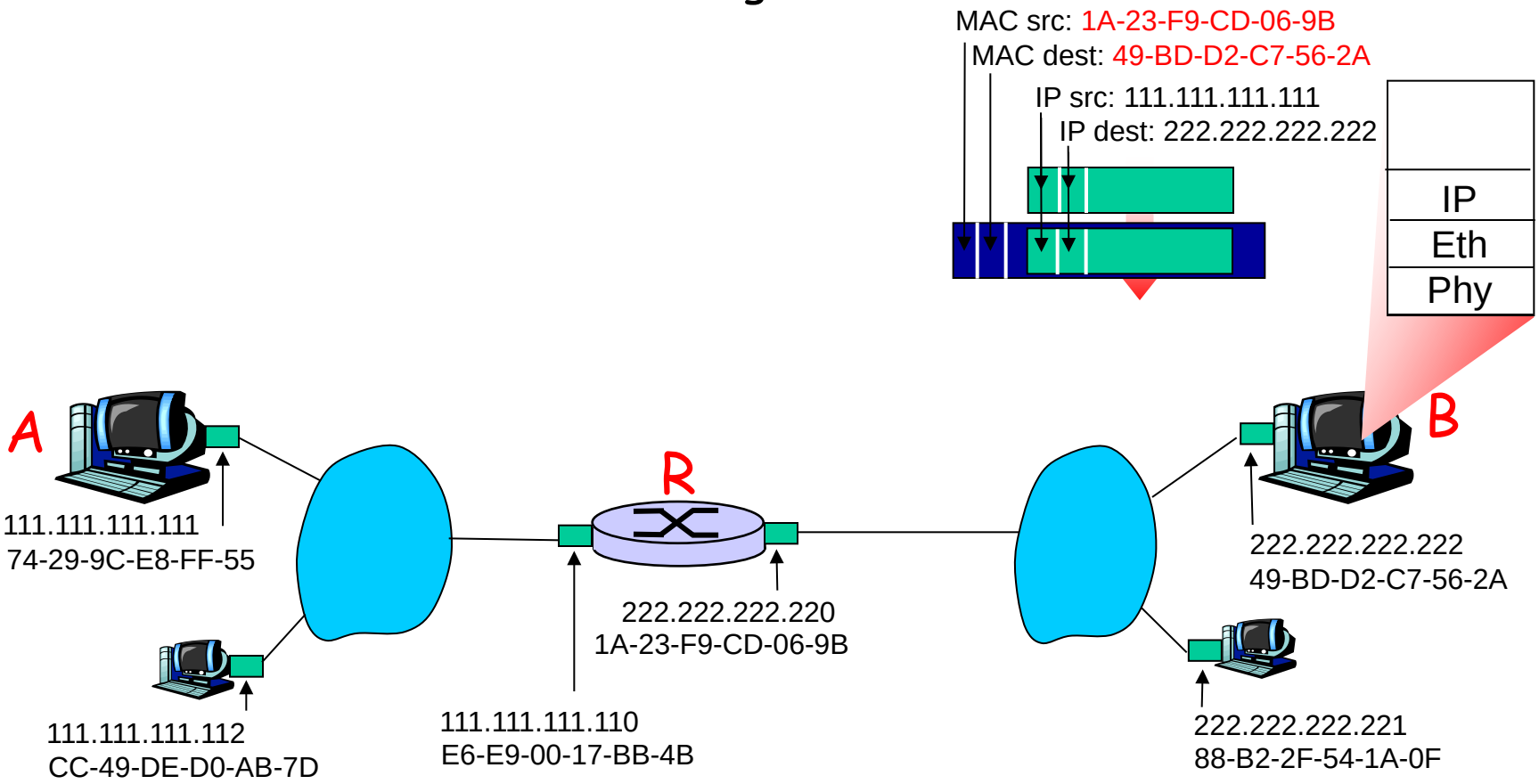
RC5. ARP : routing to another LAN

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram



RC5. ARP : routing to another LAN

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram



Direcciónes IP: ¿cómo conseguir una?

1.- Cómo la obtiene un host (parte de interface)

2.- Cómo la obtiene una ISP (parte red)

1.- Un host:

- Manualmente en la configuración de la tarjeta
  - Windows: control-panel->network->configuration->tcp/ip->properties
  - UNIX: /etc/rc.config [/etc/network/interfaces]
- DHCP: Dynamic Host Configuration Protocol: Un servidor facilita dinámicamente direcciones: "plug-and-play"

DHCP: Dynamic Host Configuration Protocol

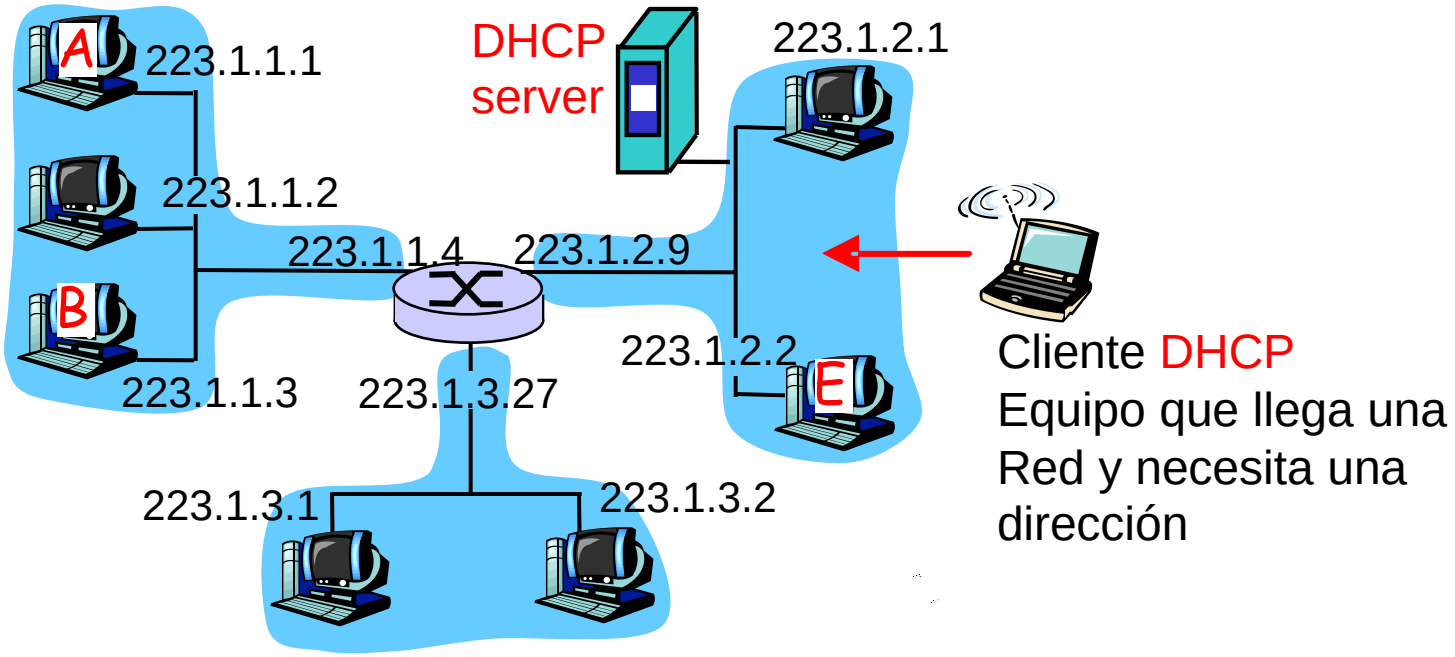
Objetivo: Permitir a los host obtener dinámicamente ("sobre la marcha") su dirección IP cuando estos se unen a la red

- Protocolo cliente servidor
- Cuestiones de seguridad…
- Permite re-usar direcciones (similar a la multiplexación estadística)
- Permite soporte a usuarios móviles (aunque de forma limitada al moverse de red local)
- Requiere retransmisión fuera de la red local)

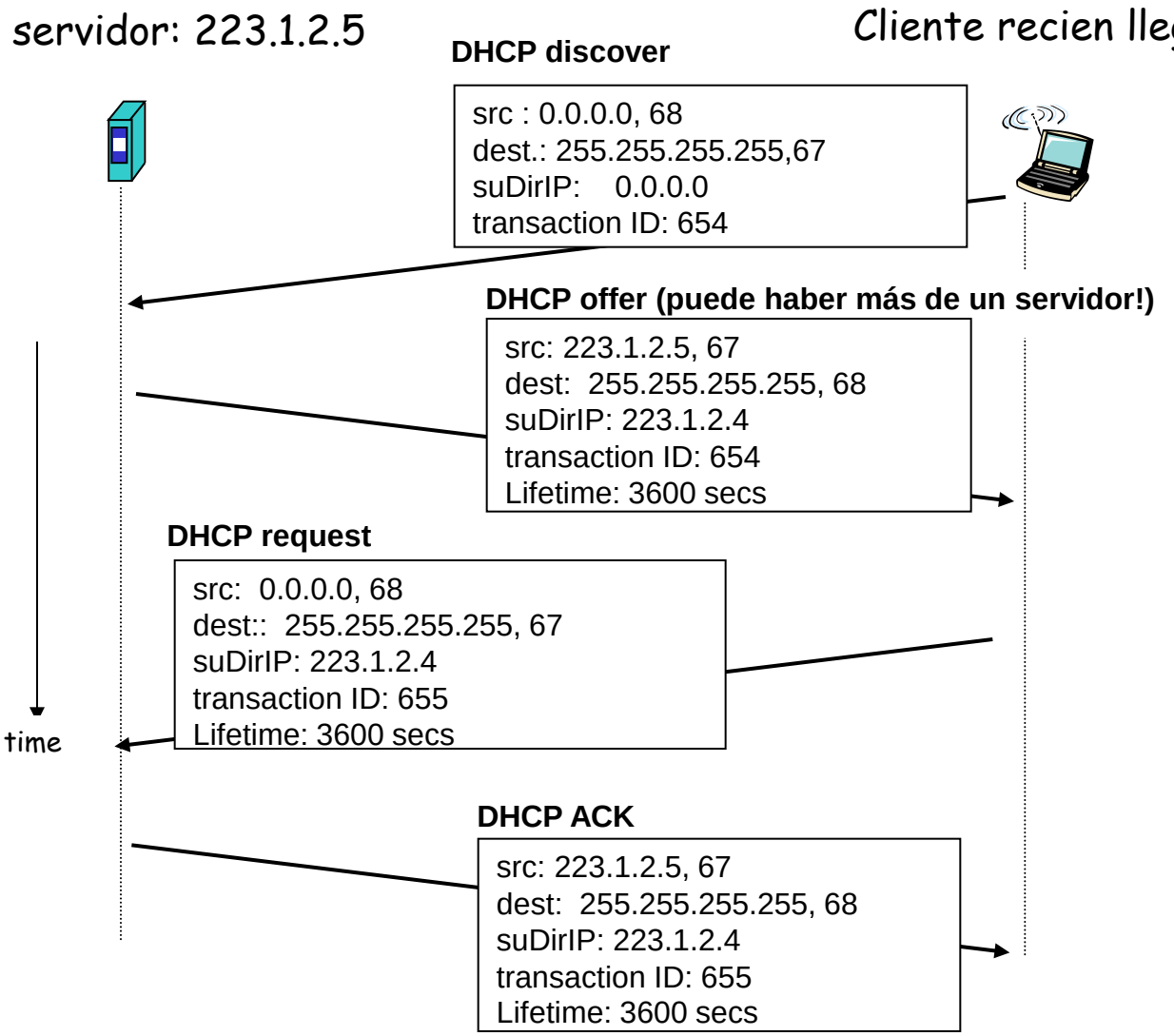
DHCP Funcionamiento:

- host broadcasts "DHCP discover" msg [optional]
- DHCP server responds with "DHCP offer" msg [optional]
- host requests IP address: "DHCP request" msg
- DHCP server sends address: "DHCP ack" msg

DHCP escenario cliente-servidor



DHCP cliente-servidor (servidor en misma red local)*

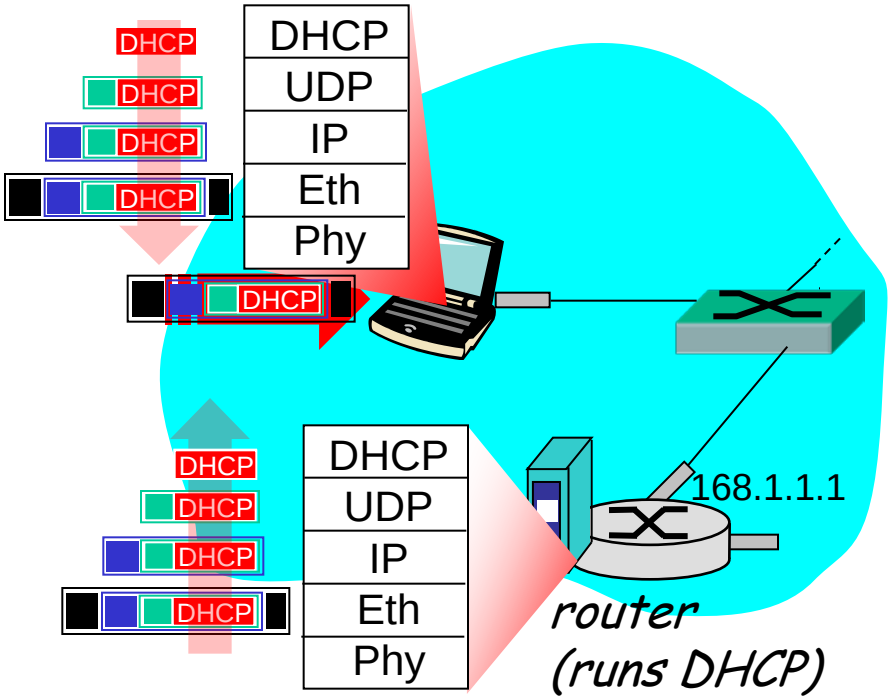


DHCP: devuelve más que la dirección IP

DHCP puede devolver más que la dirección IP:

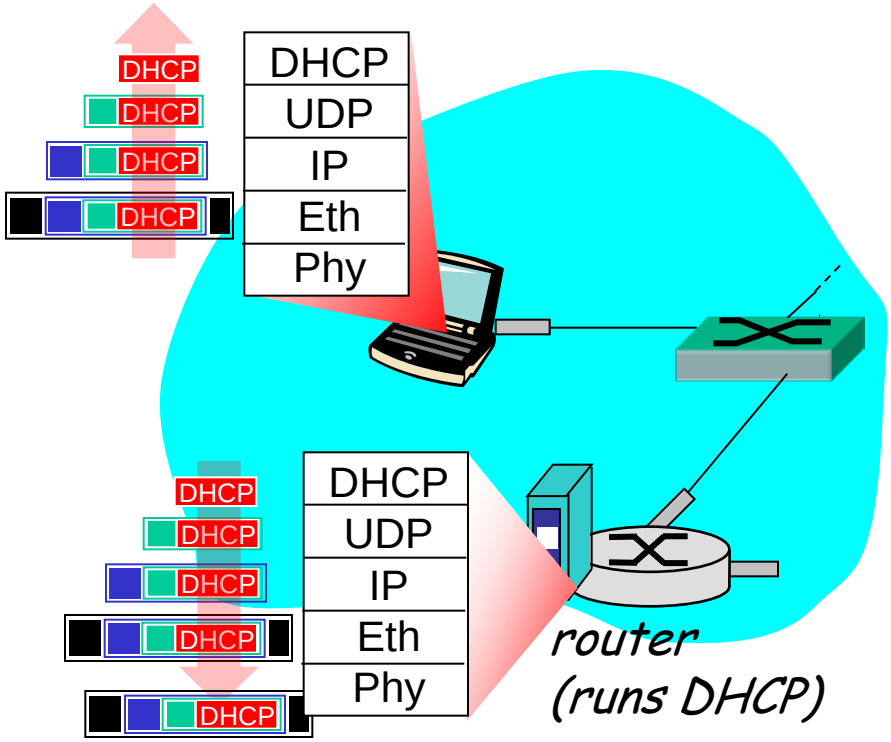
- Dirección de la puerta de enlace predeterminada (o gateway) [lo veremos más adelante]
- Nombre y dirección IP del servidor de nombres (DNS)
- Máscara de red (como hemos visto: indicación del número de bits que representan la red y la interface)

DHCP: ejemplo & Wireshark



- El portátil necesita una IP, la dirección del router que da acceso a Internet (addr of first-hop router), dirección del DNS
- La petición DHCP se encapsula en UDP, en IP y en Ethernet
- Se envía en difusión Ethernet (dest: FFFFFFFFFFFF) en la LAN, que debe recivir el router ejecutando DHCP
- Desencapuslamiento del paquete

DHCP: ejemplo & Wireshark



- El servidor envía un DHCP ACK con la información solicitada

Direcciones IP: Como conseguirlas?

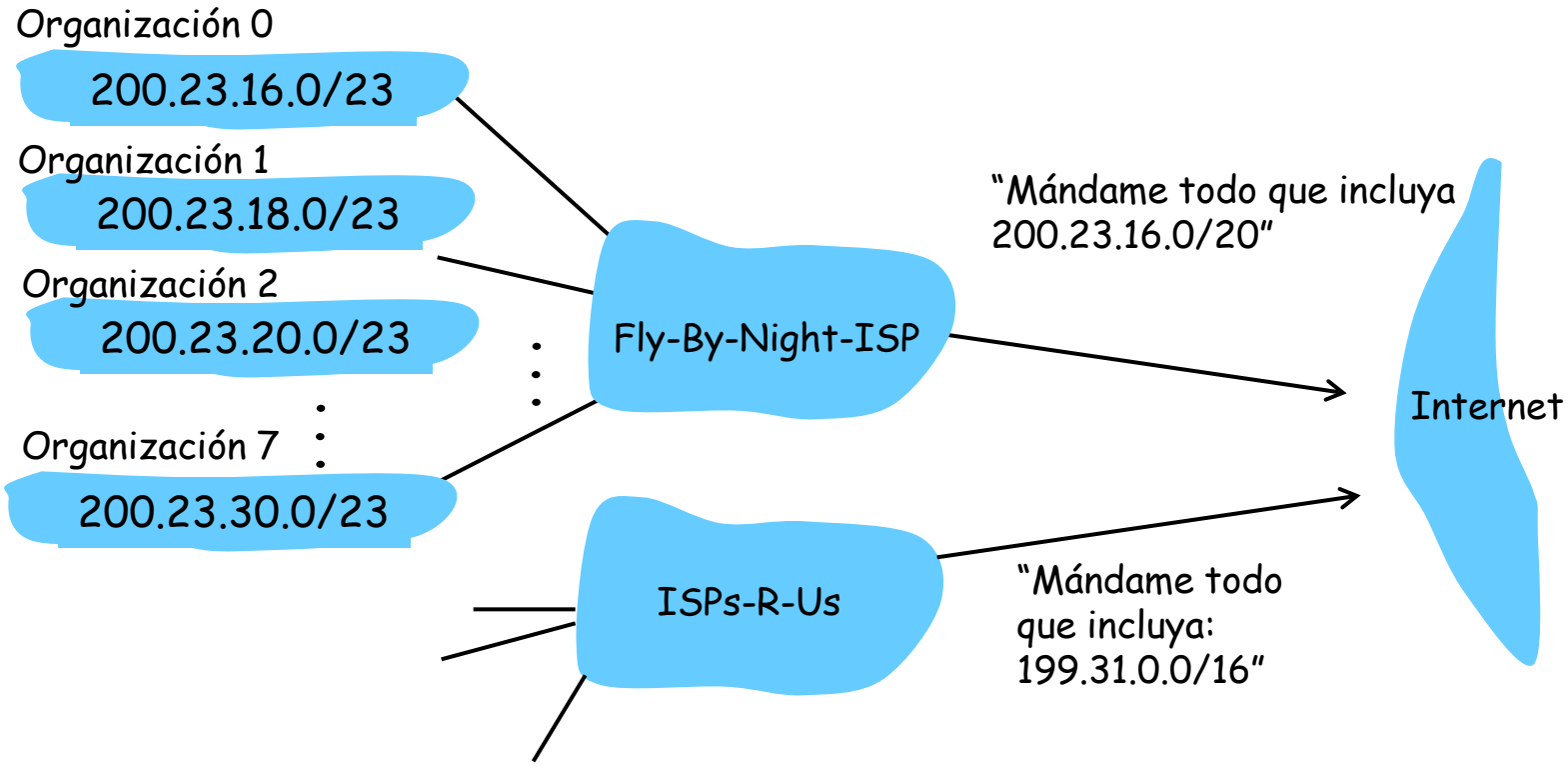
Q: Como obtener la parte de la dirección IP de red?

A: Las ISP reciben bloques de direcciones IP

ISP's block	11001000 00010111 00010000	00000000	200.23.16.0/20
Organización 0	11001000 00010111 0001000**0**	00000000	200.23.16.0/23
Organización 1	11001000 00010111 0001001**0**	00000000	200.23.18.0/23
Organización 2	11001000 00010111 0001010**0**	00000000	200.23.20.0/23
...	.....	....	....
Organización 7	11001000 00010111 0001111**0**	00000000	200.23.30.0/23

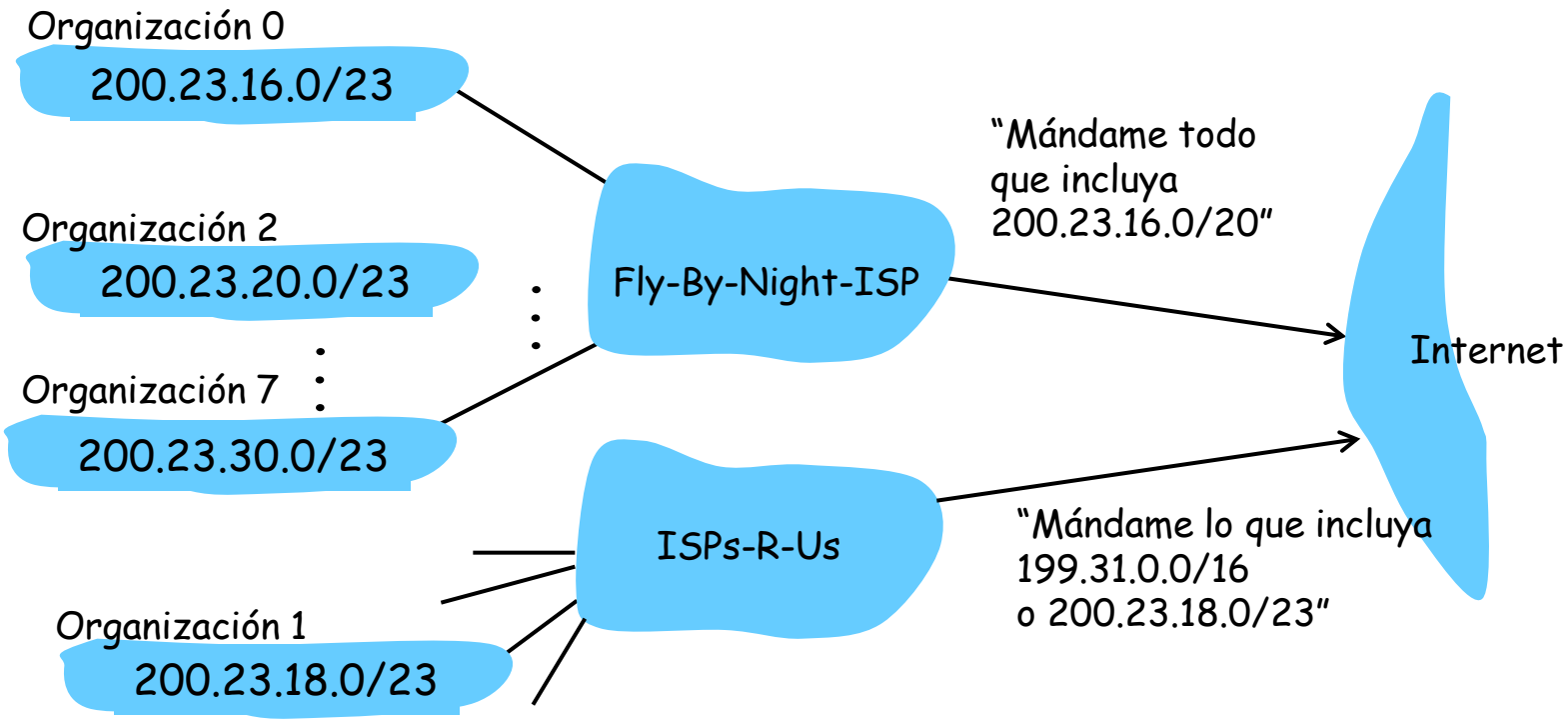
Direccionamiento jerárquico: **agregación de rutas IP**

Direccionamiento jerárquico permite el anuncio de las rutas eficientemente:



Cambios en el direccionamiento IP jerarquico

IP aplica la regla de dar preferencia al prefijo más largo
ISPs-R-Us tiene una ruta más específica a Organización 1 y permite cambios "fáciles"



Administración de Internet

Q: Quien facilita a las ISPs los bloques de direcciones?

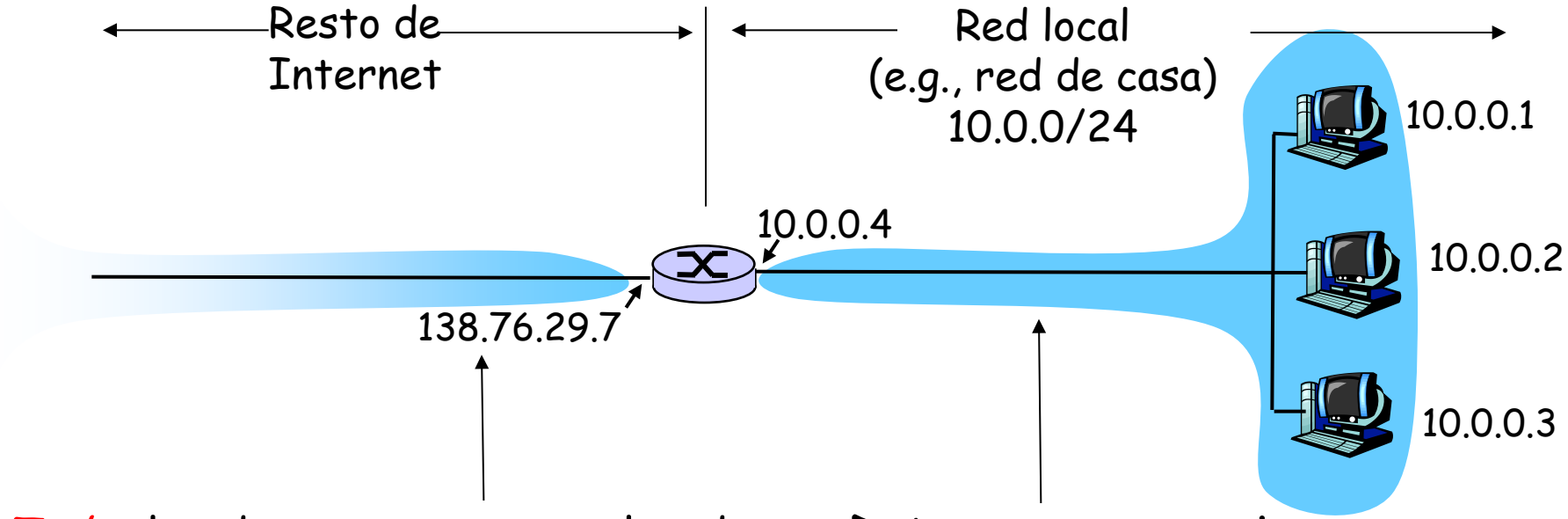
A: ICANN: Internet Corporation for Assigned Names and Numbers

- Asigna direcciones
- Gestiona los DNS
- Asigna nombres de los dominios y disputas
- La gestión esta parcialmente dividida en áreas geográficas
- Historia: IANA (Internet Assigned Numbers Authority)

Enrutado entre redes IP

- En caso de conexión a varios routers IP, se puede requerir una configuración IP más compleja: varias entradas con distintos rangos (IP_rango+máscara) donde aplica la regla del prefijo más largo.

NAT: Network Address Translation



Todos los datagramas que salen de la red local tienen una única dirección NAT : 138.76.29.7 (por ejemplo), y diferente puerto (nivel transporte) origen

Datagramas con origen o destino en esta red. Tienen 10.0.0/24 direcciones para origen y destino

NAT: Network Address Translation

- Motivación: Los usuarios de las redes locales usan una única dirección IP en lo que respecta al mundo "exterior":
  - El ISP solo facilita una IP a cada red doméstica/oficina
  - Puede modificar las direcciones internas sin notificación al resto del mundo
  - Puede cambiar la parte de red sin cambios en las direcciones internas
  - Los elementos dentro de la red local no son directamente accesibles ni visibles para el resto (lo que implica mayor seguridad)

NAT: Network Address Translation

Implementación Un router NAT debe:

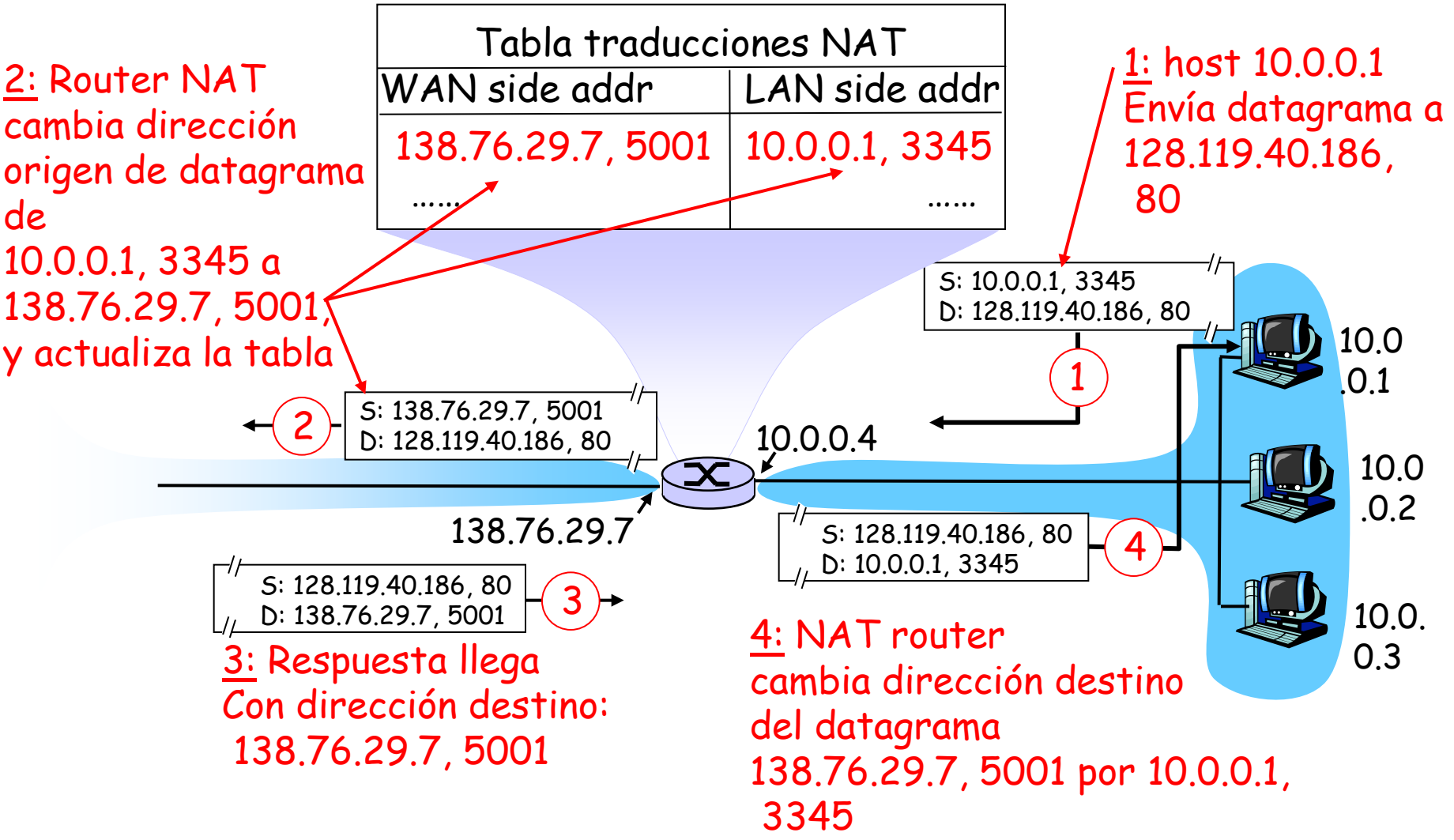
- *Datagramas salientes: reemplazar* (IP origen, puerto origen) de cada datagrama saliente a (dirección NAT IP, nuevo puerto)

  . . . Clientes/servidores remotos responderán usando (dirección NAT IP, nuevo puerto) como dirección destino

- *Almacenar (en la tabla de traducciones NAT)* cada (IP origen, puerto origen) a (dirección NAT IP, nuevo puerto) como pareja

- *Datagramas entrantes: remplazar* (dirección NAT IP, nuevo puerto) en el campo destino de cada datagrama entrante con la correspondiente entrada (IP origen, puerto) almacenada en la tabla de traducciones NAT

NAT: Network Address Translation

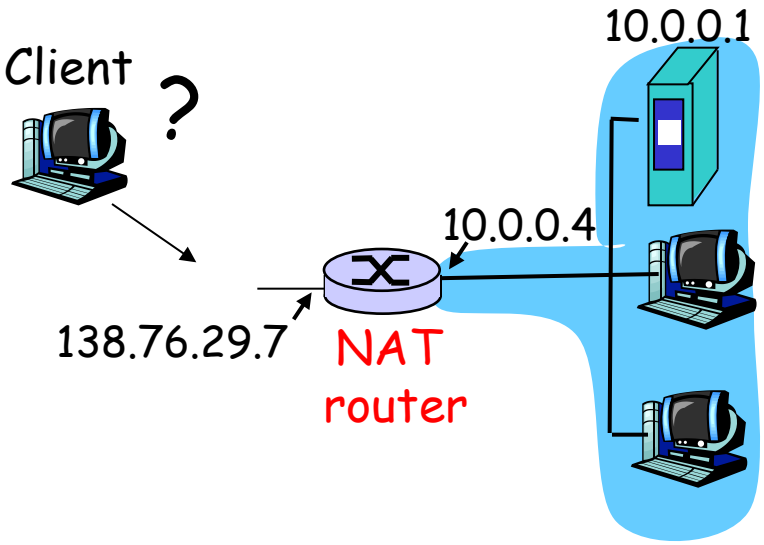


NAT: Network Address Translation

- Los campos puerto de TCP y UDP son un campo de 16 bits:
  - Más de 60000 conexiones simultaneas con una sola dirección IP!
- Sin embargo, NAT tiene su parte oscura:
  - Los routers de nivel 3 no deberían tocar campos del nivel 4 (y si no son TCP, UDP? Checksums?)
  - Los puertos deben determinar procesos
  - Dificulta la accesibilidad externa (Internet) [enfoque terminal – terminal]
    - Como vamos a ver importante en P2P o VoIP
  - Sensible a errores (reinicio del router)
  - Dificulta la monitorización (la relación host <> usuario no es válida).
  - IPv6

Accesibilidad tras NAT

- Un cliente quiere conectarse a un servidor (u otro cliente) cuya dirección IP es 10.0.0.1 (privada)
  - Sin embargo, el cliente no puede usar esa dirección como destino
  - Solo es visible para el cliente la IP NAT: 138.76.29.7
- **Solución 1: estáticamente** redirigir toda conexión entrante a un puerto dado a una dirección concreta:
  - e.g., (138.76.29.7, puerto 2500) siempre redirigido a 10.0.0.1 puerto 25000
    - Útil para servidores y imposible con aplicaciones con puertos aleatorios (evitar ser filtradas)
    - Exige conocimientos



Accesibilidad tras NAT

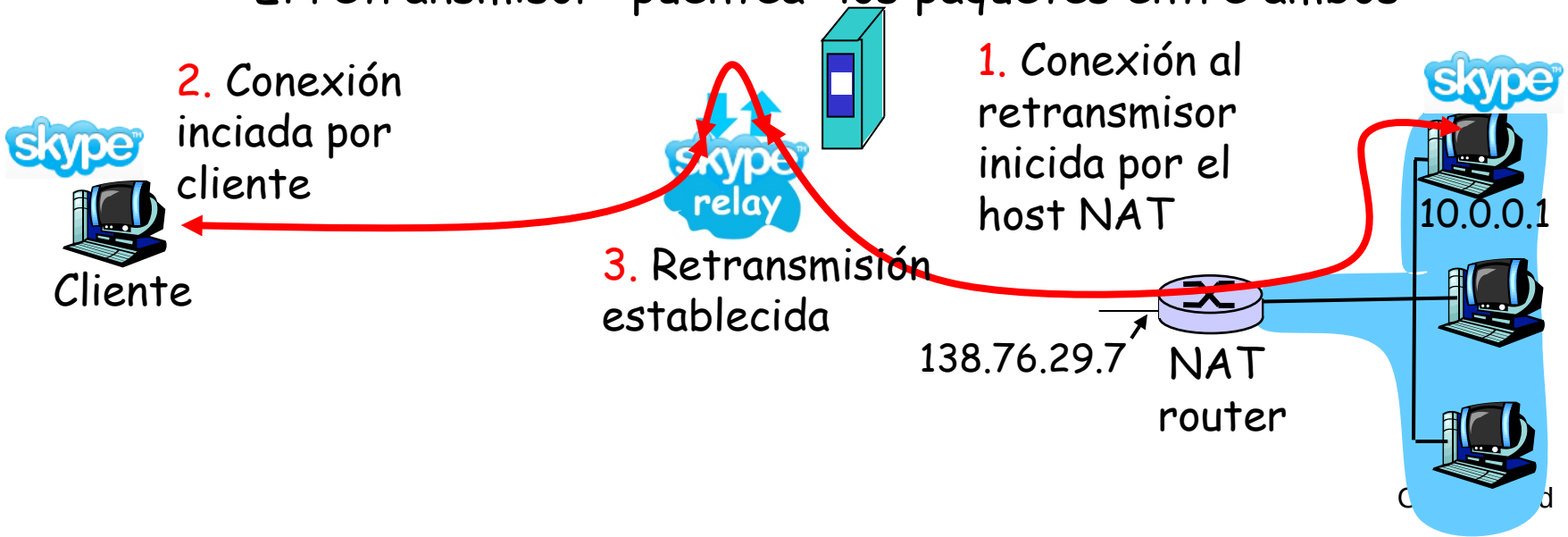
- **Solución 2: NAT transversal simple**
- En el caso de que un extremo no este detrás de un router NAT se puede utilizar un host intermedio
- Este host intermedio debe ser accedido por el extremo con NAT (iniciar sesión de muchas aplicaciones)
- El extremo sin NAT, contacta con el host intermedio, el host con NAT es avisado que debe iniciar una conexión con el extremo sin NAT que es públicamente accesible (inversión de la conexión)

Accesibilidad tras NAT

- **Solución 3: NAT transversal simétrico (STUN).**
- En el caso de que ambos extremos estén detrás de un router NAT ambos extremos pueden añadir una entrada de forma "artificial" en la tabla NAT y darla a conocer al otro extremo.
- Se necesita un servidor (IP pública) intermedio con el que ambos extremos abren una conexión
  - Tras que uno de ellos haya hecho saber al servidor que quiere conectarse con el otro.
- Está conexión crea una entra en la tabla NAT ¿de qué tipo?¿con qué campos?
- El servidor comunica a los extremos de la entrada creada en la tabla del otro extremo.
- Se inicia comunicación.
- Ciertas dudas en la seguridad y su funcionamiento… ¿UDP?

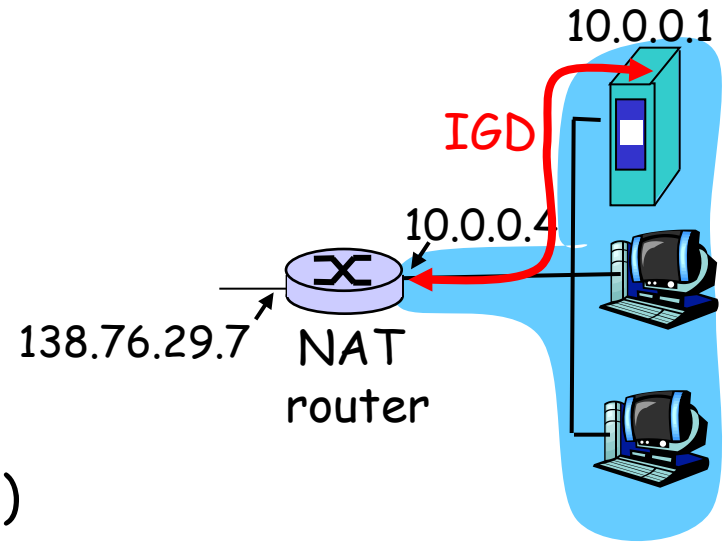
Accesibilidad tras NAT

- **Solución 4: NAT transversal con retransmisores (TURN) → ¿usados en Skype?**
  - Clientes con NAT establecen conexiones a nodos de retransmisión (a veces llamados supernodos, y frecuentemente no-parte de la red propia de Skype)
  - El otro extremo también se conecta al nodo retransmisor
  - El retransmisor "puentea" los paquetes entre ambos



Accesibilidad tras NAT

- **Solución 5: Internet Gateway Device (IGD) Protocol del conjunto de protocolos Universal Plug and Play (UPnP) .**
- Permite gestionar y hacer pública la tabla de traducciones (a veces mediante un nodo intermedio):
  - Usa la IP pública 138.76.29.7
  - Añadir/Borrar el "mapeo" de puertos (Con tiempos de duración)
  - En definitiva, resulta como la configuración estática anterior pero de forma automática y dinámica modifica la configuración de los puertos NAT
  - Exige colaboración con el router
    - ¿redes domesticas?¿oficina?



Capítulo 4: Capa de red

4. 1 Introducción

4.2 Circuitos virtuales y datagramas

4.3 Que hay dentro de un router

4.4 IP: Protocolo de Internet

- Formato
- Fragmentación
- Direccionamiento IPv4
- DHCP y NAT
  - Accesibilidad tras NAT
- ICMP
- IPv6

4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector distancia
- Enrutamiento jerárquico

4.6 Enrutamiento en Internet

- RIP
- OSPF
- BGP
- Tipo de relaciones

4.7 Enrutamiento Broadcast y multicast

ICMP: Internet Control Message Protocol

- Usado por hosts y routers para comunicar información de nivel de red:
  - Comunicación errores: protocolo, host, red, network, puerto inalcanzable (unreachable)
  - Peticiones/Respuesta de echo (usado por ping)
- Capa de red pero sobre IP:
  - ICMP es encapsulado en datagramas IP (campo protocolo =1)
  - Por tanto tan poco fiable como IP
  - Subprotocolo de la capa de red

Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

Formato Mensajes ICMP*

TIPO	CODIGO	DESCRIPCION	ERROR
0	0	Respuesta de Eco	NO
3	Varios	Destino inalcanzable	SI
4	0	Source quench	SI
5	Varios	Redirección	SI
8	0	Solicitud de Eco	NO
9	0	Aviso de Ruta	NO
10	0	Solicitud de Ruta	NO
11	Varios	Plazo de tiempo excedido	SI
12	Varios	Problemas con parámetros	SI
13	0	Solicitud de marca de tiempo	NO
14	0	Respuesta de marca de tiempo	NO
17	0	Solicitud de mascara de red	NO
18	0	Respuesta de mascara de red	NO

ICMP*

- Formato de mensajes de ICMP

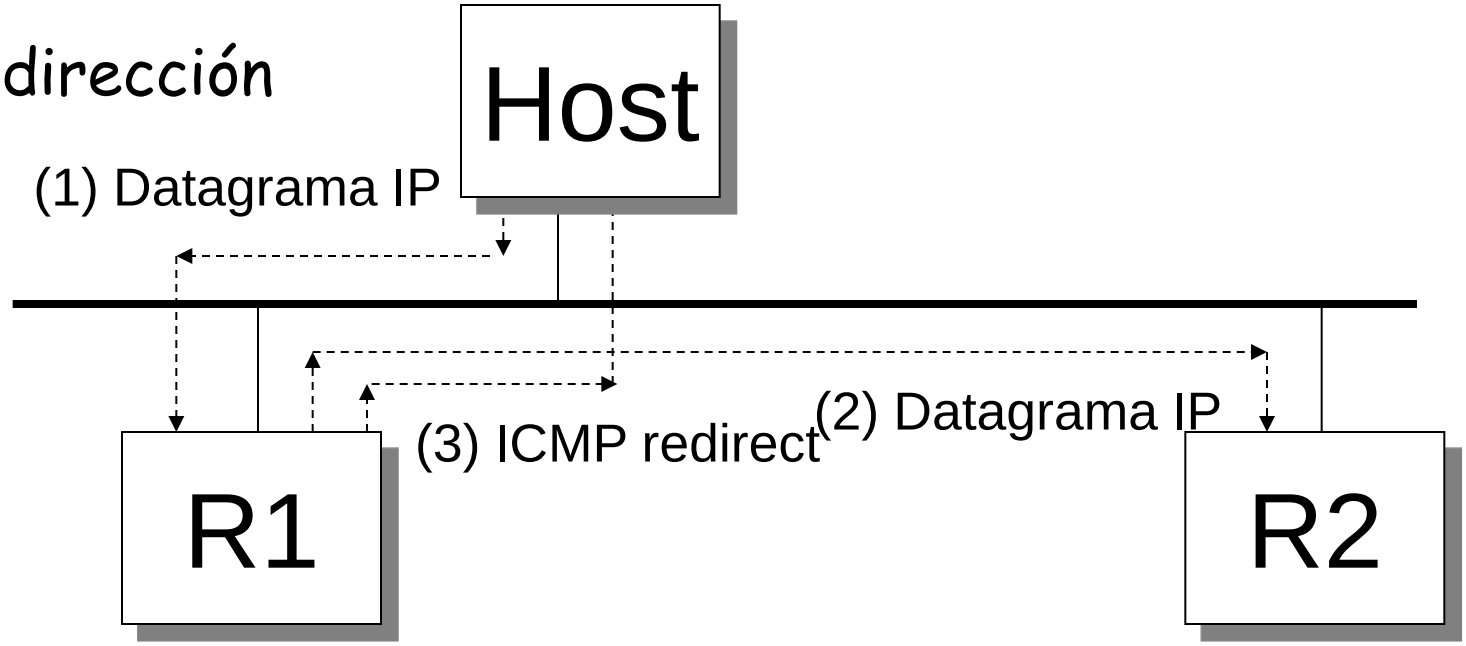
- Los campos Tipo y Código definen la clase de mensaje

Mensajes ICMP*

- Un mensaje de error ICMP nunca se genera como respuesta a:
  - Otro mensaje de error de ICMP.
  - Un datagrama con destino IP de broadcast
  - Un datagrama con dirección física broadcast
  - Un fragmento del datagrama distinto del primero
  - Un datagrama cuya dirección IP fuente no sea la de un único host
- En los mensajes ICMP de error se copia la cabecera y los 8 primeros bytes de datos del datagrama IP que provocó el error

Ejemplos de mensajes ICMP*

- Redirección



- (1) El host envía un datagrama IP a R1, debido a que R1 es el router por defecto.
- (2) R1 recibe el datagrama y decide que R2 es el router adecuado. Cuando lo envía a R2 se da cuenta que está utilizando la misma red física por donde ha llegado el datagrama
- (3) R1 envía un ICMP redirect al host, indicando que los siguientes datagramas con el mismo destino se deben enviar a R2 (definición de mascaras por ejemplo)

Traceroute (tracert) e ICMP

- Puede haber dustintas versiones:
  - Basadas en UDP/TCP
  - ICMP
- El origen envia paquetes al destino con:
  - Primero con el TTL =1
  - Luego TTL=2, etc.
- Cuando un datagrama dado llega a un router:
  - El router descarta el datagrama
  - Y envia aviso: mensaje ICMP (tipo 11, código 0)
  - Los mensajes ICMP incluyen el nombre del router y la dirección IP
- Cuando vuelve la respuesta, se calcula el RTT
- traceroute suele ejecutarse 3 veces

Criterios de parada

- Cuando finalmente el paquete llega a destino, se considera el último salto:
  - Tipicamente cuando en vez de devolver un paquete tipo 11 código 0, se obtiene un ping (replay) o…
  - Destino devuelve puerto "inalcanzable" si el puerto no es accesible (fue escogido uno "raro*")
- Se para el programa

Capítulo 4: Capa de red

4. 1 Introducción

4.2 Circuitos virtuales y datagramas

4.3 Que hay dentro de un router

4.4 IP: Protocolo de Internet

- Formato
- Fragmentación
- Direccionamiento IPv4
- DHCP y NAT
  - Accesibilidad tras NAT
- ICMP
- IPv6

4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector distancia
- Enrutamiento jerárquico

4.6 Enrutamiento en Internet

- RIP
- OSPF
- BGP
- Tipo de relaciones

4.7 Enrutamiento Broadcast y multicast

IPv6

- Motivación: El conjunto de direcciones de 32-bits van a ser próximamente completamente alocadas
- Otras motivaciones:
  - Optimizar la cabecera para ayudar en el procesado/reenvío
  - Capacidades adicionales para dar QoS

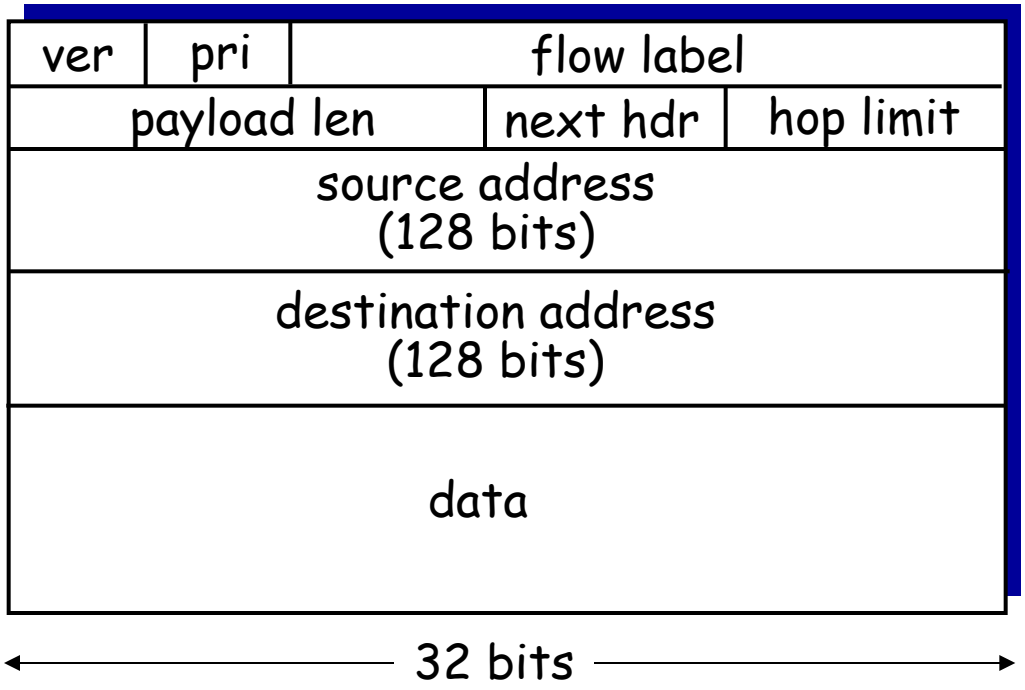
  El formato de datagramas:
  - Cabecera fija de 40 Bytes
  - Fragmentación no permitida

IPv6 cabecera

Priority: identifica prioridades entre los datagramas de un flujo

Flow Label: identifica datagramas en el mismo flujo (concepto de "flujo" no bien definido)

Next header: identifica protocolo superior para datos



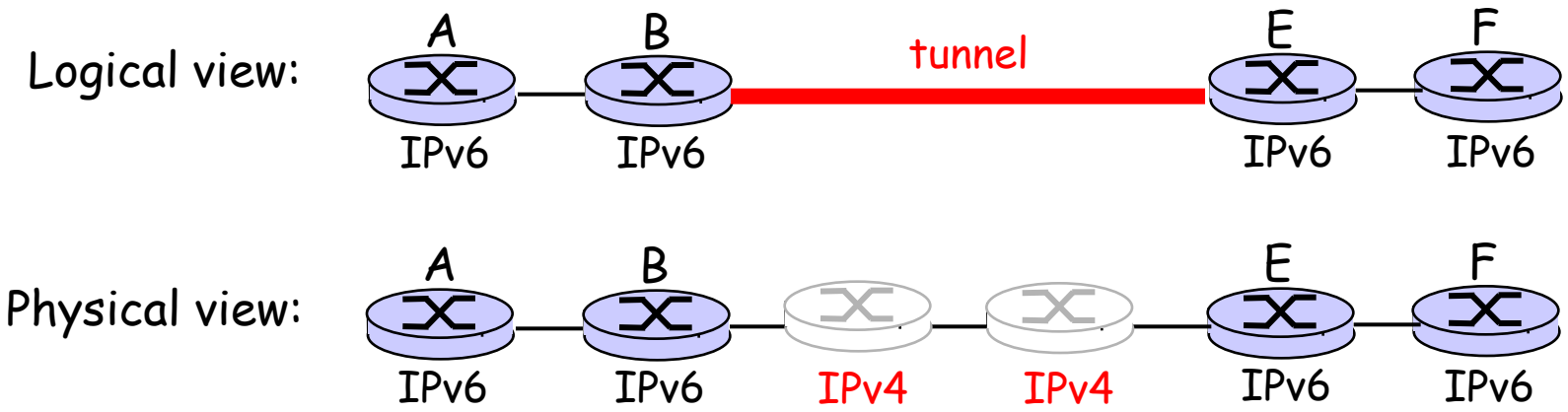
Otros cambios desde IPv4

- *Checksum*: eliminado para reducir el costo de procesado por salto
- *Options:* permitidas, pero fuera de la cabecera, indicado por el campo "Next Header"
- *ICMPv6:* nueva versión de ICMP
  - Mensajes nuevos, e.g. "Packet Too Big"
  - Funciones de gestión de grupos

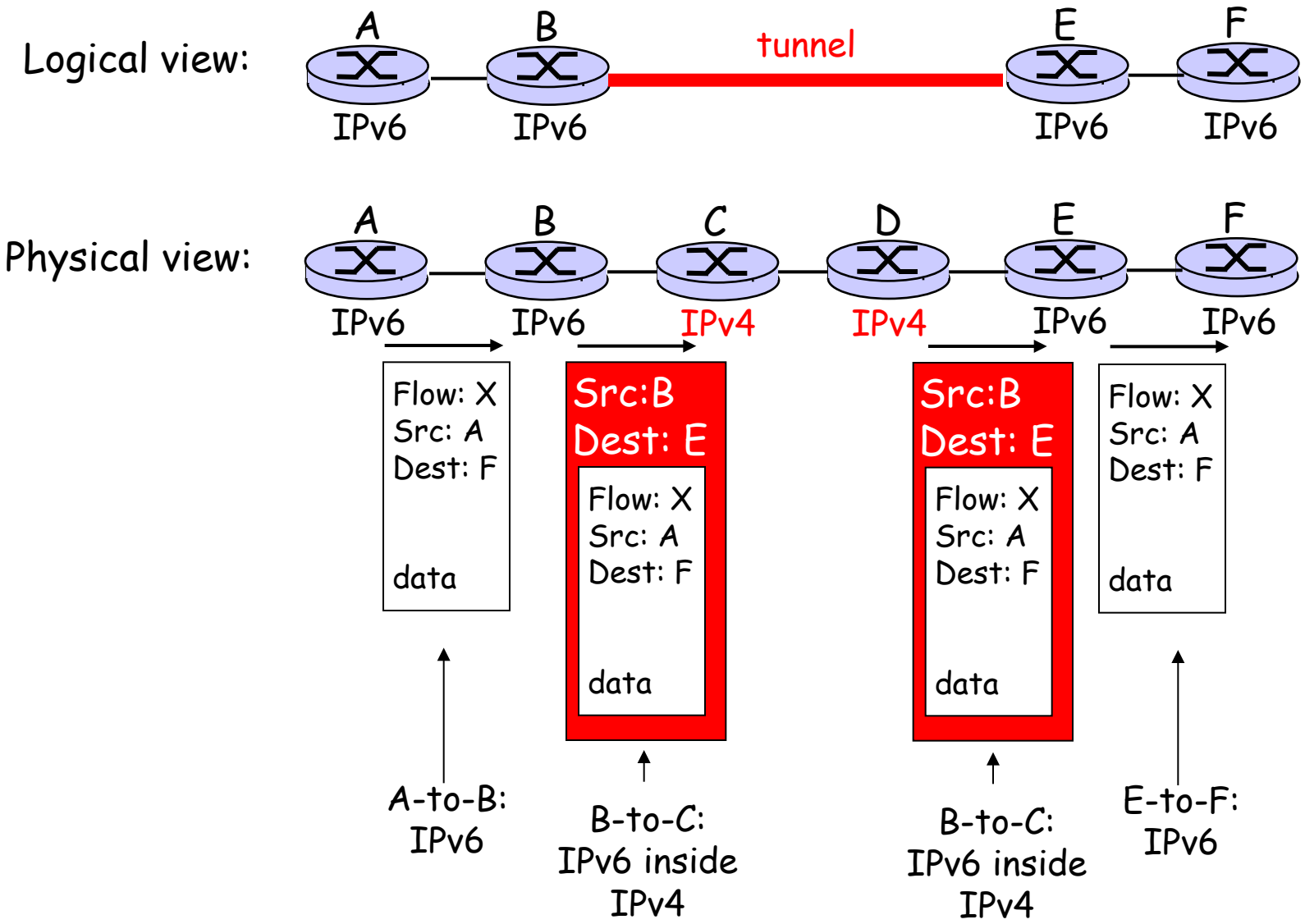
Transición desde IPv4 a IPv6

- No todos los routers pueden ser actualizados simultaneamente:
  - ¿Día fijo?
  - ¿Como van a operar ambos protocolos a la vez?
- *Tunelado:* IPv6 es transportado como carga útil (payload, datos) en IPv4 entre routers IPv4
- *¿Pila dual?*

Tunneling


Logical view:
A
B
tunnel
E
F
IPv6
IPv6
IPv6
IPv6
Physical view:
A
B
E
F
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6

Tunneling


Logical view:
A
B
tunnel
E
F
IPv6
IPv6
IPv6
IPv6
Physical view:
A
B
C
D
E
F
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6
Flow: X
Src: A
Dest: F
data
Src:B
Dest: E
Flow: X
Src: A
Dest: F
data
Src:B
Dest: E
Flow: X
Src: A
Dest: F
data
Flow: X
Src: A
Dest: F
data
A-to-B:
IPv6
B-to-C:
IPv6 inside
IPv4
B-to-C:
IPv6 inside
IPv4
E-to-F:
IPv6

Capítulo 4: Capa de red

4. 1 Introducción

4.2 Circuitos virtuales y datagramas

4.3 Que hay dentro de un router

4.4 IP: Protocolo de Internet

- Formato
- Fragmentación
- Direccionamiento IPv4
- DHCP y NAT
  - Accesibilidad tras NAT
- ICMP
- IPv6

4.5 Algoritmos de enrutamiento

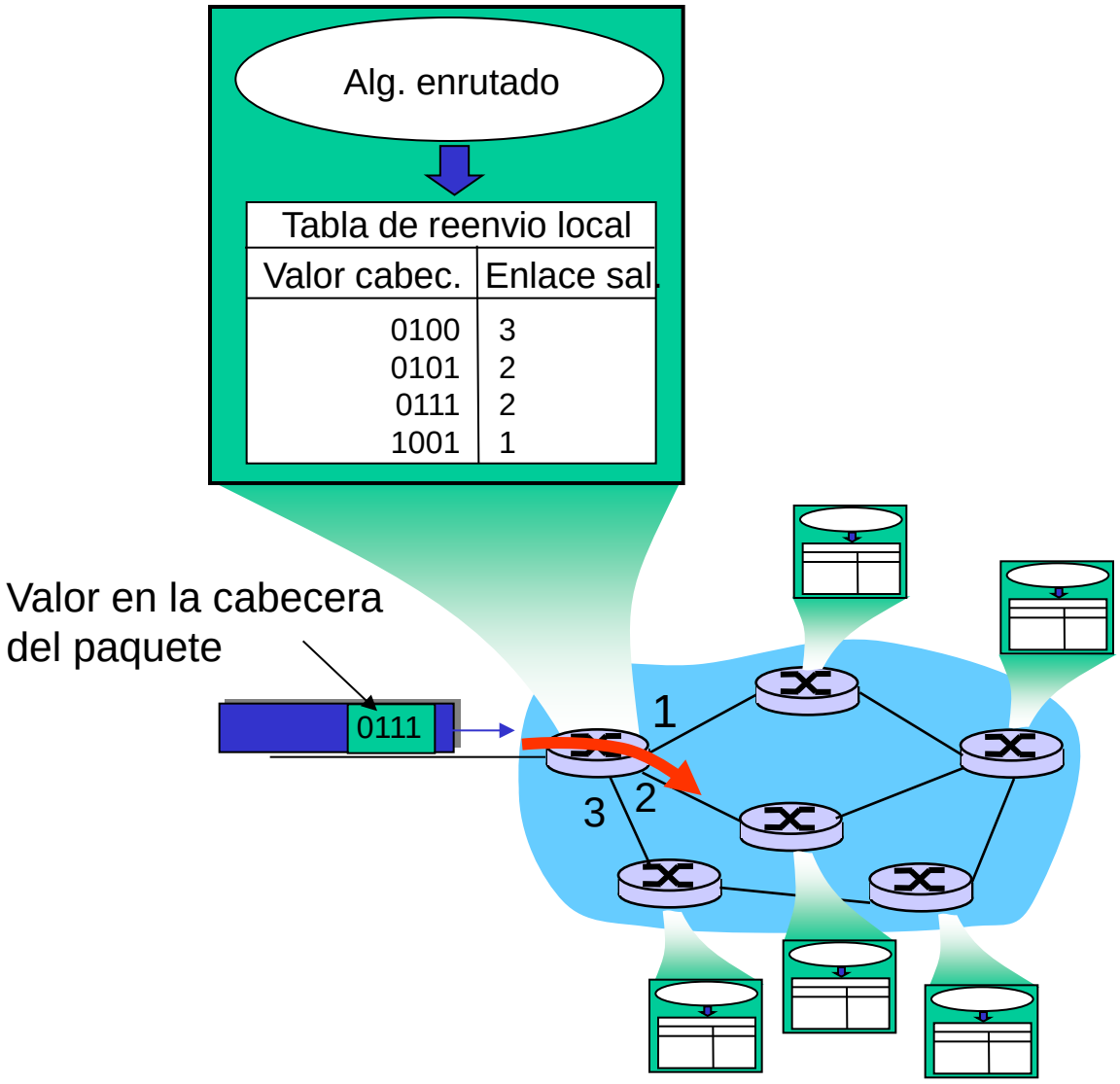
- Estado de enlaces
- Vector distancia
- Enrutamiento jerárquico

4.6 Enrutamiento en Internet

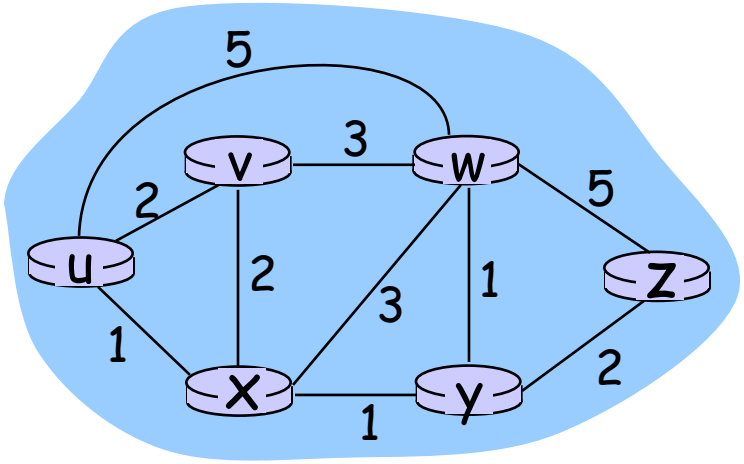
- RIP
- OSPF
- BGP
- Tipo de relaciones

4.7 Enrutamiento Broadcast y multicast

Interacción entre enrutado y reenvio



Grafos



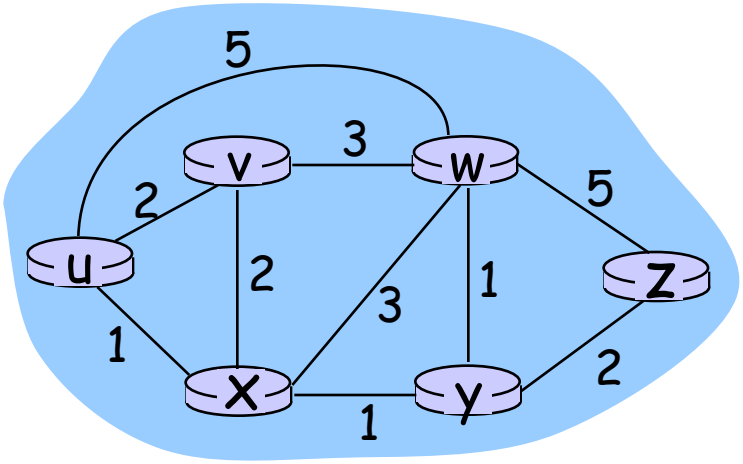
Grafo: G = (N,E)

N (nodos) = conjunto de routers = { u, v, w, x, y, z }

E (aristas (edge)) = conjunto de enlaces ={ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }

Usar grafos es útil pues aplica el conocimiento de este área al mundo de las redes
Ejemplo: P2P, donde N es un conjunto de usuarios y E es un conjunto de conexiones TCP

Grafos: coste



- c(x,x') = coste de enlace (x,x')

  - e.g., c(w,z) = 5

- Coste puede ser 1, o inversamente relacionado con el ancho de banda, o la congestión

Ruta = secuencía de nodos $(x_1, x_2, x_3, \ldots, x_p)$ tal que cada una de las parejas (x_1, x_2) (x_2, x_3)... sean aristas pertenecientes a E

Coste de una ruta $(x_1, x_2, x_3, \ldots, x_p) = c(x_1,x_2) + c(x_2,x_3) + \ldots + c(x_{p-1},x_p)$

Pregunta: Como encontrar las **rutas de menor coste**, e.g., entre u - z ?

Algoritmos de enrutado tienen como objeto encontrarlas

Métricas*

- El concepto de trayectoria más corta implica una métrica. Se han propuesto posibles métricas*:
  - Número de saltos
  - Distancia física
  - Retardo medio de encolamiento y transmisión de un paquete
  - Longitud media de las colas
  - Coste de comunicaciones
  - Tráfico medio

Clasificación de algoritmos

¿Información global o descentralizada?

Global:

- Todos los routers conocen la topología completa y el "coste" de los enlaces
- Algoritmos de estado de enlace

Descentralizado:

- Un router conoce sólo a los routers vecinos con enlaces físicos y su coste
- Proceso iterativo de cálculo que implica el intercambio de información con los vecinos
- Algoritmos basados en vector distancia

¿Dinámico o estático?

Estático:

- La información en los routers cambia poco con el tiempo

Dinámico:

- Los routers cambian más rápido
  - Actualizaciones periódicas
  - Responden a los cambios en el coste de los enlaces

Capítulo 4: Capa de red

4. 1 Introducción

4.2 Circuitos virtuales y datagramas

4.3 Que hay dentro de un router

4.4 IP: Protocolo de Internet

- Formato
- Fragmentación
- Direccionamiento IPv4
- DHCP y NAT
- ICMP
- IPv6

4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector distancia
- Enrutamiento jerárquico

4.6 Enrutamiento en Internet

- RIP
- OSPF
- BGP
- Tipo de relaciones

4.7 Enrutamiento Broadcast y multicast

Cálculo de la trayectoria más corta

Algoritmo de Dijkstra

- La topología de la red y el coste de los enlaces son conocidos por todos los nodos
  - Se consigue vía **difusión** del estado de los enlaces
  - Todos los nodos tienen la **misma información**
- Calcula los caminos de mínimo coste de un nodo ("origen") a todos los demás nodos
  - Define la tabla de reenvios de dicho nodo
- Es iterativo: después de k iteraciones, encuentra el camino de mínimo coste a k destinos
- OSPF

Notación:

- c(i,j): coste asociado al enlace del nodo i al nodo j. Es infinito si no hay nodos vecinos directamente conectados
- D(v): valor actual del coste del camino entre el nodo origen y el nodo destino v
- p(v): nodo predecesor en el camino desde el origen hasta el nodo v. Es decir, el anterior nodo a v
- N: conjunto de nodos cuyo camino de coste mínimo es conocido definitivamente.
- u: nodo analizado

Algoritmo de Dijkstra

1 ***Inicialización:***
2 N' = {u}
3 Para todos los nodos v
4 Si v es vecino directo de u
5 entonces D(v) = c(u,v)
6 Si no D(v) = infinito
7
8 ***Repetir***
9 Encontrar w no incluido en N' tal que D(w) es un mínimo
10 Añadir w a N'
11 Actualizar D(v) para todos los v vecinos directos de w y no en N':
12 D(v) = min(D(v), d(w) + c(w,v))
13 /* El nuevo coste a v es o bien el valor anterior a v o el coste
14 del camino más corto a w conocido más el coste de w a v */
15 ***Hasta que todos los nodos estén en N' (==N)***

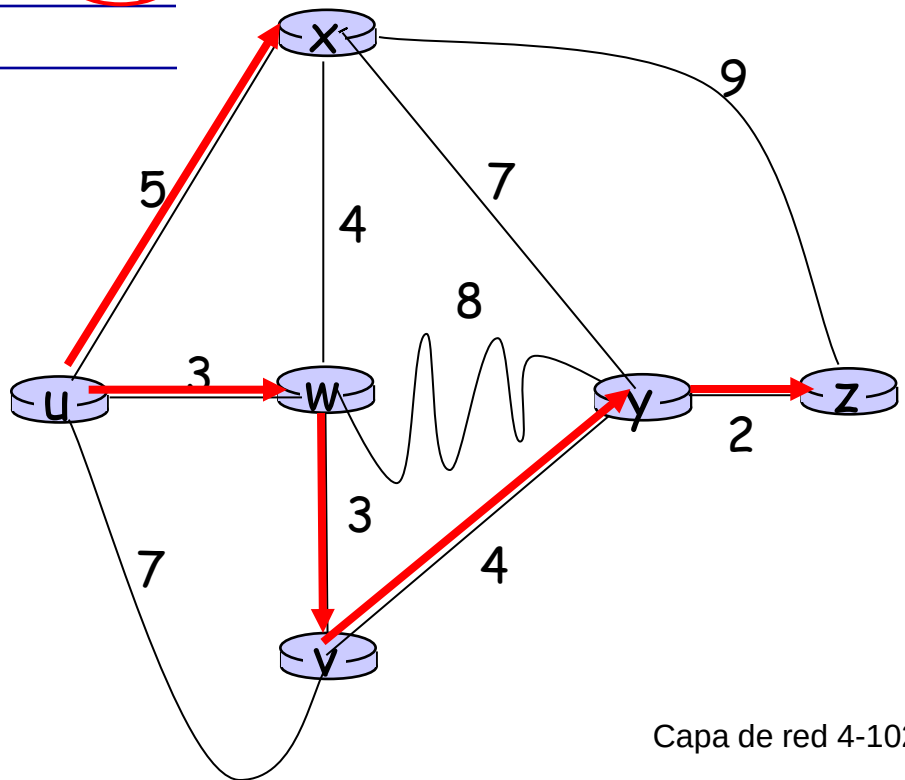
Algoritmo de Dijkstra: ejemplo

Paso	N'	D(**v**) p(v)	D(**w**) p(w)	D(**x**) p(x)	D(**y**) p(y)	D(**z**) p(z)
0	u	7,u	3,u	5,u	∞	∞
1	uw	6,w		5,u	11,w	∞
2	uwx	6,w			11,w	14,x
3	uwxv				10,v	14,x
4	uwxvy					12,y
5	uwxvyz					

Coste actual para llegar de u→(·)
Previo(·)

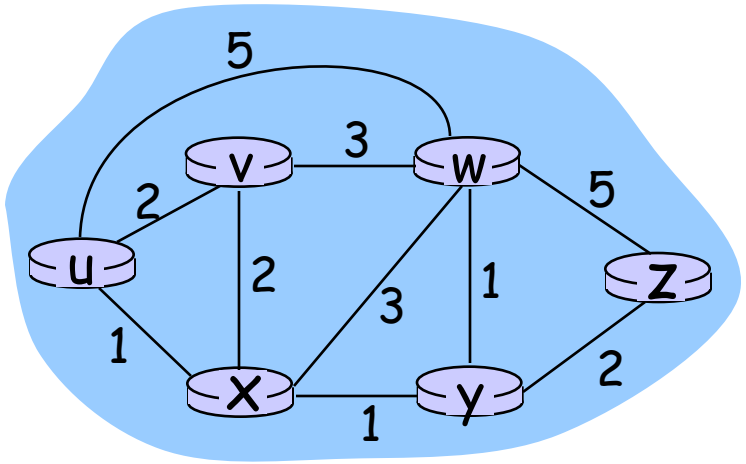
Notas:

- Construir los caminos más cortos usando el nodo anterior
- En caso de empate escoger arbitrariamente



Algoritmo de Dijkstra: otro ejemplo

Paso	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



Dijkstra: otro ejemplo (2)

Árbol de rutas mínimas desde u:

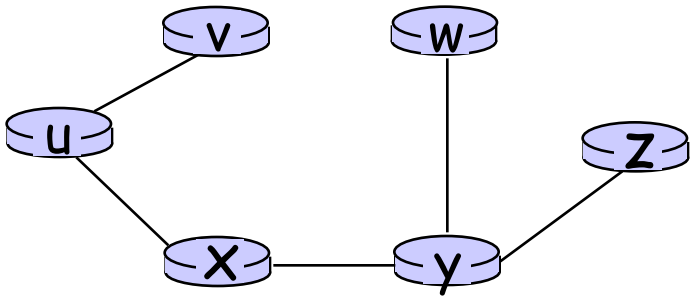


Tabla de reenvios resultante :

destino	enlace
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

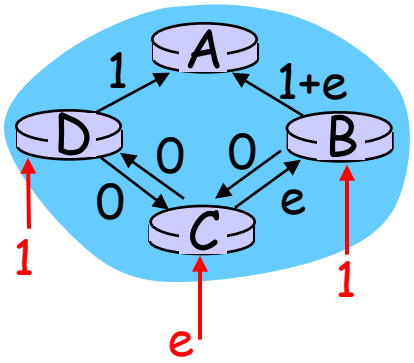
Algoritmo de Dijkstra, discusión

Complejidad de algoritmos: n nodos

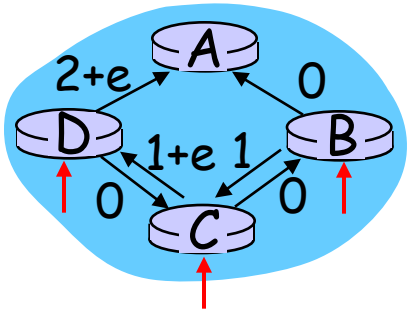
- En cada iteración es necesario analizar todos los nodos w que no estén en N'
- n(n+1)/2 compariciones: $O(n^2)$

Posibles oscilaciones:

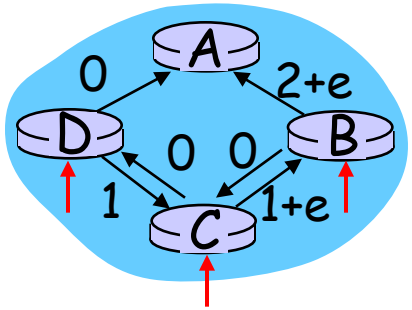
- e.g., coste del enlace = cantidad de tráfico que transporta (RIP, OSPF, BGP no sensibles carga)



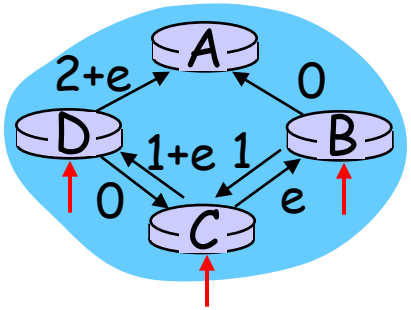
initially



… recompute routing



… recompute



… recompute

Capítulo 4: Capa de red

4. 1 Introducción

4.2 Circuitos virtuales y datagramas

4.3 Que hay dentro de un router

4.4 IP: Protocolo de Internet

- Formato
- Fragmentación
- Direccionamiento IPv4
- DHCP y NAT
- ICMP
- IPv6

4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector distancia
- Enrutamiento jerárquico

4.6 Enrutamiento en Internet

- RIP
- OSPF
- BGP
- Tipo de relaciones

4.7 Enrutamiento Broadcast y multicast

Algoritmo vector de distancias

Ecuación de Bellman-Ford

Definición

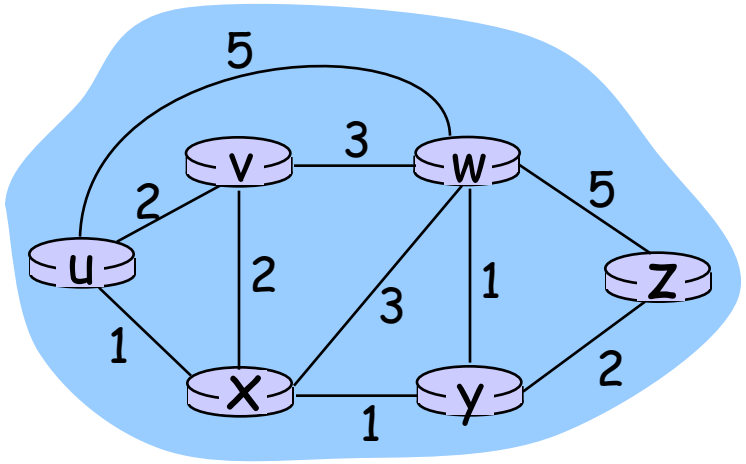
$d_x(y)$:= coste ruta de menor coste de x a y

Entonces

$$d_x(y) = \min_v \{c(x,v) + d_v(y)\}$$

donde min toma todos los vecinos v de x

Bellman-Ford ejemplo



Teniendo en cuenta (distancia vecinos):
$d_v(z) = 5$, $d_x(z) = 3$, $d_w(z) = 3$

Ecuación B-F establece:

$$d_u(z) = \min \{ c(u,v) + d_v(z), \; c(u,x) + d_x(z), \; c(u,w) + d_w(z) \}$$
$$= \min \{2 + 5, \; 1 + 3, \; 5 + 3\} = 4$$

Nodo del que se obtiene el mínimo debe ser el siguiente salto en la ruta más corta ➜ tabla de reenvíos

Algoritmo de vector de distancias

- $D_x(y)$ = estimación del coste mínimo desde x a y
  - x mantiene un vector de distacias $\mathbf{D_x} = [D_x(y): y \in N]$
- Nodo x:
  - Conoce el costo para alcanzar cada vecino v: $c(x,v)$
  - Mantiene el vector distancias de cada vecino. Para cada vecino v, x mantiene $\mathbf{D_v} = [D_v(y): y \in N]$

Algoritmo vector de distancias

Idea básica:

- "De vez en cuando", cada nodo manda su propio vector de distancias (estimado) a sus vecinos
- Cuando x recibe un nuevo vector de distancias (estimado) desde sus vecinos, actualiza su propio DV usando la ecuación de B-F:

$$D_x(y) \leftarrow \min_v\{c(x,v) + D_v(y)\} \quad \text{para cada nodo } y \in N$$

- En situaciones normales, la estimación converge al valor real de coste mínimo $d_x(y)$

Algoritmo de encaminamiento basado en vector de distancia

Iterativo:

- Continúa mientras los nodos se intercambien información
- Auto-parada: no hay "señal de parada"

Asíncrono:

- No es necesario que los nodos intercambien información en unos momentos determinados ni siguiendo un orden fijo

Distribuido:

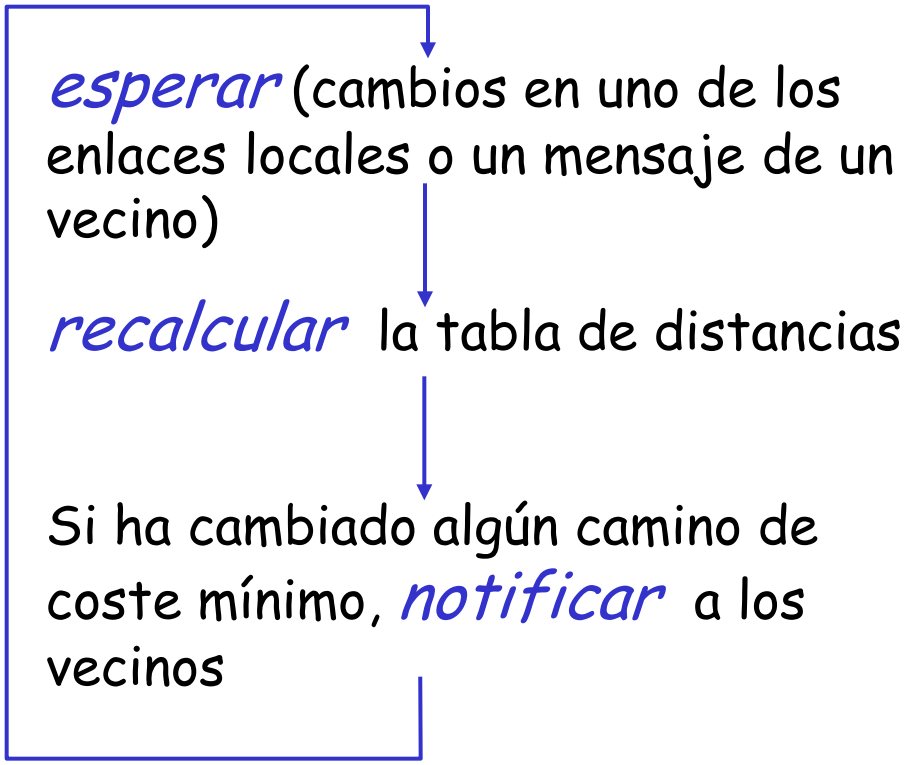
- Cada nodo intercambia información sólo con los vecinos inmediatos

Estructura de datos: tabla de distancias

- Cada nodo tiene una tabla propia
- Cada posible destino ocupa una fila
- Cada vecino inmediato ocupa una columna

Algoritmo de vector de distancia

En cada nodo:



$D_x(y) = \min\{c(x,y) + D_y(y),\ c(x,z) + D_z(y)\}$
$= \min\{2+0\ ,\ 7+1\} = 2$

$D_x(z) = \min\{c(x,y) + D_y(z),\ c(x,z) + D_z(z)\}$
$= \min\{2+1\ ,\ 7+0\} = 3$

node x table

desde \ coste a	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

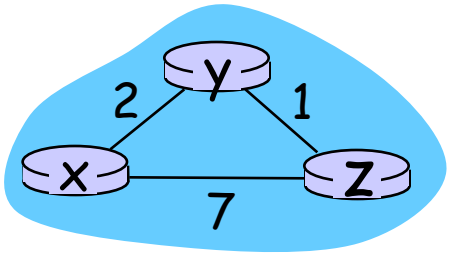
desde \ coste a	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

node y table

desde \ coste a	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

node z table

desde \ coste a	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0



time

$D_x(y) = \min\{c(x,y) + D_y(y),\ c(x,z) + D_z(y)\}$
$= \min\{2+0\ ,\ 7+1\} = 2$

$D_x(z) = \min\{c(x,y) + D_y(z),\ c(x,z) + D_z(z)\}$
$= \min\{2+1\ ,\ 7+0\} = 3$

node x table

Coste a

desde	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

desde	x	y	z
x	0	2	3(y)
y	2	0	1
z	7	1	0

desde	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

node y table

coste a

desde	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

desde	x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

desde	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

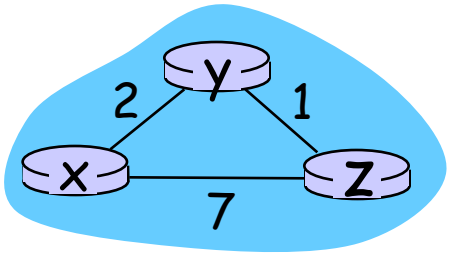
node z table

coste a

desde	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

desde	x	y	z
x	0	2	7
y	2	0	1
z	3	1	0

desde	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

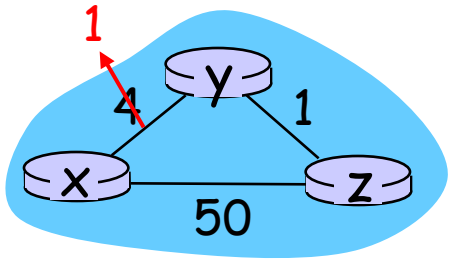


tiempo

Vector de distancias : cambios en los costes de los enlaces

Cambios en los costes:

- Un nodo detecta cambio en el coste local
- Actualiza información de enrutado, recalcula vector de distancias (VD)
- Si VD cambia, se notifica a los vecinos



"Las buenas noticias viajan rápido"

t_0: *y* detecta cambios en los costes de los enlaces, actualiza su VD, informa a sus vecinos

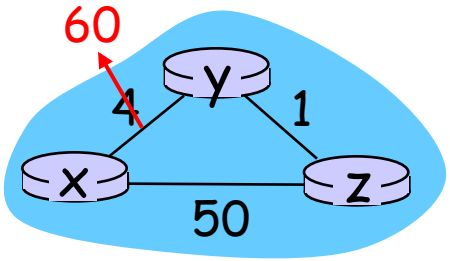
t_1: *z* recibe la actualización desde *y*, actualiza la tabla, computa el nuevo coste mínimo a *x*, envía a sus vecinos su VD.

t_2: *y* recibe la actualización de *z*, actualiza su tabla de distancias. Los costes no cambian: para, así que no manda mensaje a z.

Vector de distancias : cambios en los costes de los enlaces

Frente a cambios:

- Las buenas noticas viajan rápido
- Malas noticias viajan lento problema de "cuenta al infinito"!
- Puede exigir muchas iteraciones hasta que el algoritmo pare (leer texto)



Inversa envenenada:

- Si Z enruta a través de Y para llegar a X :
  - Z dice a Y que su distancia a X es infinita (de ese modo Y no enrutará a través de Z para llegar a X)
- Soluciona esto el problema? NO, cuando se involucra a más nodos o ciertas topologías (e.g., en línea)

Malas noticias

c(x,y)=4 c(y,z)=1 c(x,z)=50 (t0)
c(x,y)=60 c(y,z)=1 c(x,z)=50 (t2)

Tabla y (t0)

desde \ coste a	x	y	z
x	0	4	5
y	4	0	1
z	5	1	0

Tabla y (t1)

desde \ coste a	x	y	z
x			
y	**6**	0	1
z	5	1	0

$D_y(x) = \min\{c(y,x) + D_x(x), c(y,z)+D_z(x)\}$ **(t3)**

$= \min\{60+0 , 1+5\} = 6!$

Tabla z (t1)

desde \ coste a	x	y	z
x			
y	4	0	1
z	5	1	0
	∞		

Tabla z (t2)

desde \ coste a	x	y	z
x			
y	6	0	1
z	**7**	1	0

...

Comparación de algoritmos: Estado de enlace (EE) vs. vector de distancias (VD)

Complejidad de los mensajes

- EE: con n nodos, E enlaces, O(nE) mensajes enviados cada ciclo
- VD: intercambio de mensajes entre vecinos únicamente

Velocidad de convergencia

- EE: Algoritmo $O(n^2)$ que requiere O(nE) mensajes
- VD: tiempo de convergencia variable
  - Puede haber bucles de encaminamiento
  - problema de cuenta al infinito

Robustez: ¿qué pasa si hay errores en el router?

EE:

- Un nodo puede enviar costes de *enlace* erróneos
- Recuperables en la siguiente ejecución
- Cada nodo calcula sólo su propia tabla (errores en el cálculo solo dañinos de forma local)

VD:

- Un nodo puede enviar costes de *rutas* incorrectos
- Los errores se propagan por la red y difícil recuperarse
- La tabla de un nodo se usa para configurar el resto de las tablas (errores cálculos)

Capítulo 4: Capa de red

4. 1 Introducción

4.2 Circuitos virtuales y datagramas

4.3 Que hay dentro de un router

4.4 IP: Protocolo de Internet

- Formato
- Fragmentación
- Direccionamiento IPv4
- DHCP y NAT
- ICMP
- IPv6

4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector distancia
- Enrutamiento jerárquico

4.6 Enrutamiento en Internet

- RIP
- OSPF
- BGP
- Tipo de relaciones

4.7 Enrutamiento Broadcast y multicast

Enrutamiento jerárquico

El estudio hasta ahora ha sido una idealización

- Router idénticos: mismo algoritmo de enrutado, todos accesibles entre ellos, todos se conocen entre ellos…
- *… no es cierto en la práctica, (la jerarquización ya fue introducido cuando hablamos de direcionamiento)*

Escala: millones de hosts (routers) conectados a Internet

- No se puede almacenar todas las direcciones!
- Los mensajes necesarios por los algoritmos serían enormes y frecuentes!

Autonomía administrativa

- Internet = red de redes
- Cada administrador de red suele querer controlar el de su propia red (AS)
- Se podría considerar como conjuntos de subredes

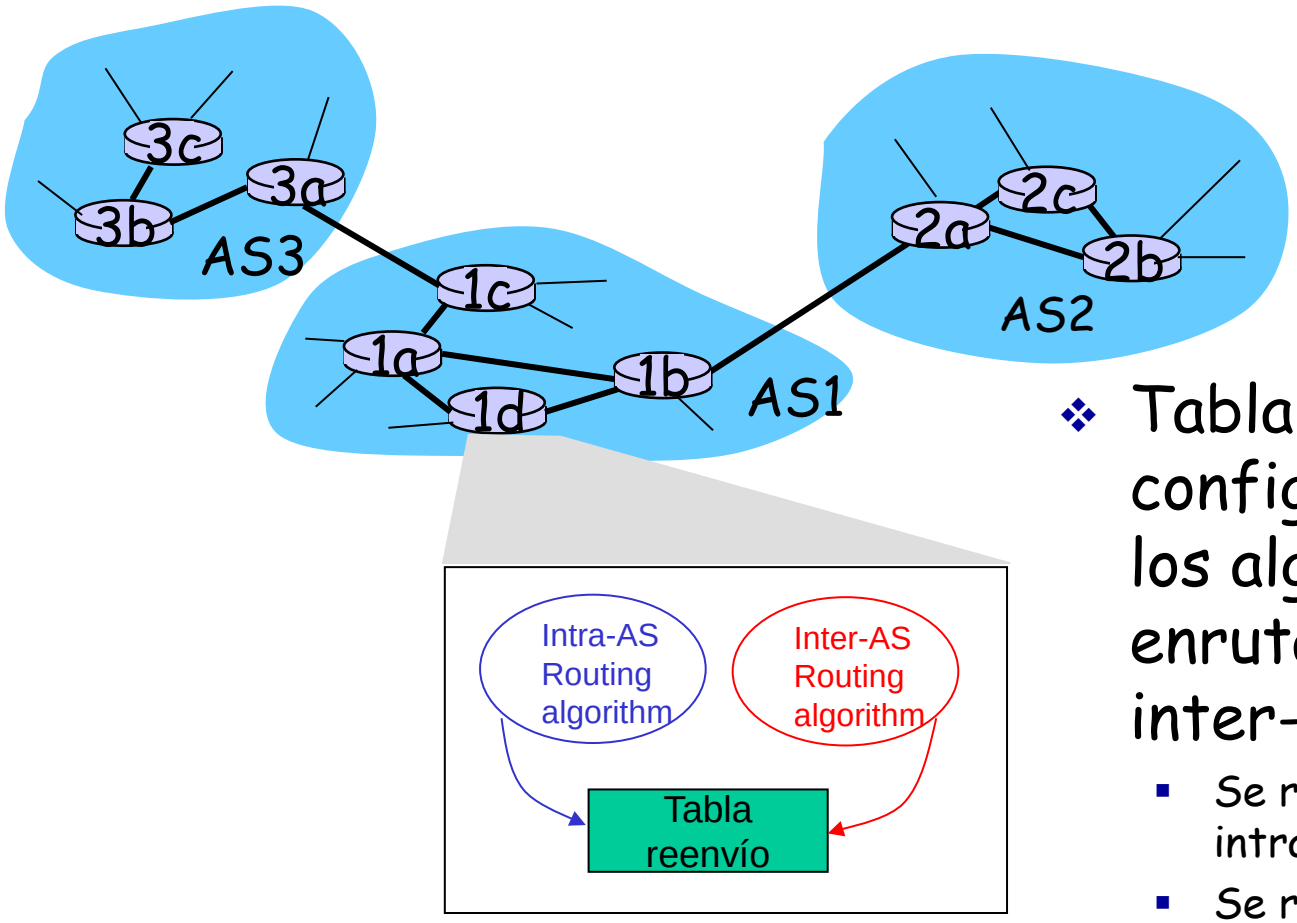
Enrutamiento jerárquico

- Grupo de routers formando, "sistemas autónomos" (AS)
- Los routers en el mismo AS ejecutan mismos algoritmos de enrutado:
  - Protocolo de enrutado "intra-AS"
  - Routers en diferentes AS pueden ejecutar diferentes algoritmos intra-AS

Pasarela o puerta de enlace (gateway)

- Puerta de enlace : Permite conectar una red a otra
- En los "bordes" de los AS
- Están conectados a otros routers de otros AS ¿Cómo?
- Distinguir (¿?) de puerta de enlace predeterminada

AS interconectadas



- Tabla de reenvíos está configurada tanto por los algoritmos de enrutado intra- e inter-AS
  - Se requiere información intra-As para destino locales
  - Se requiere información **tanto** inter-AS y intra-As para destinos externos

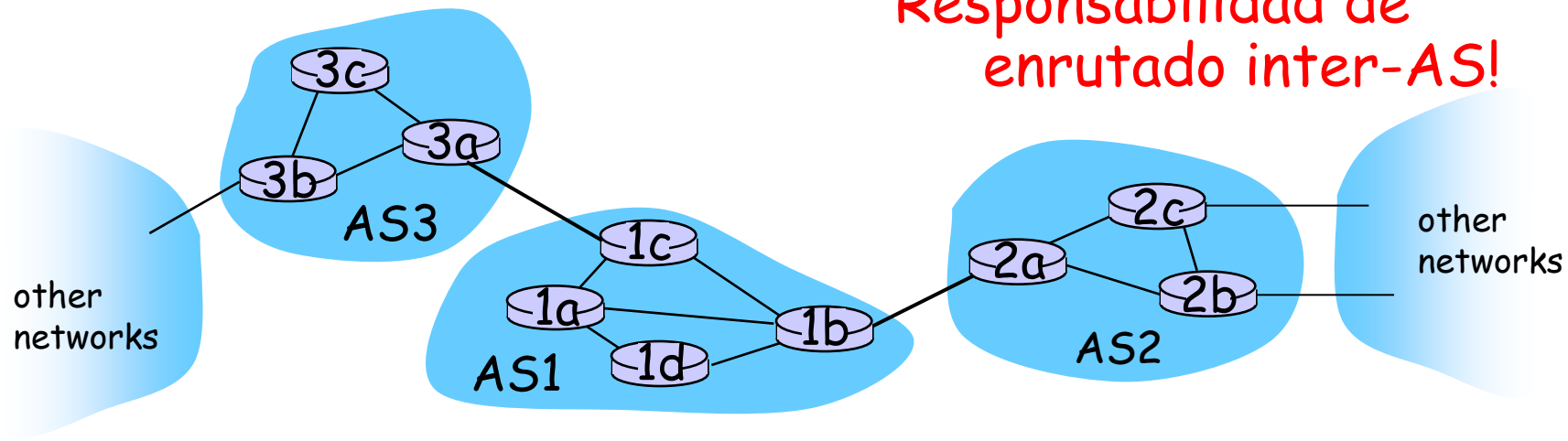
Tareas Inter-AS

- Supón que un router en AS1 recibe un datagrama destinado fuera de AS1:
  - El router debería reenviar el paquete a una puerta de enlace, pero… ¿a cual?

AS1 debe (nodo en AS1):

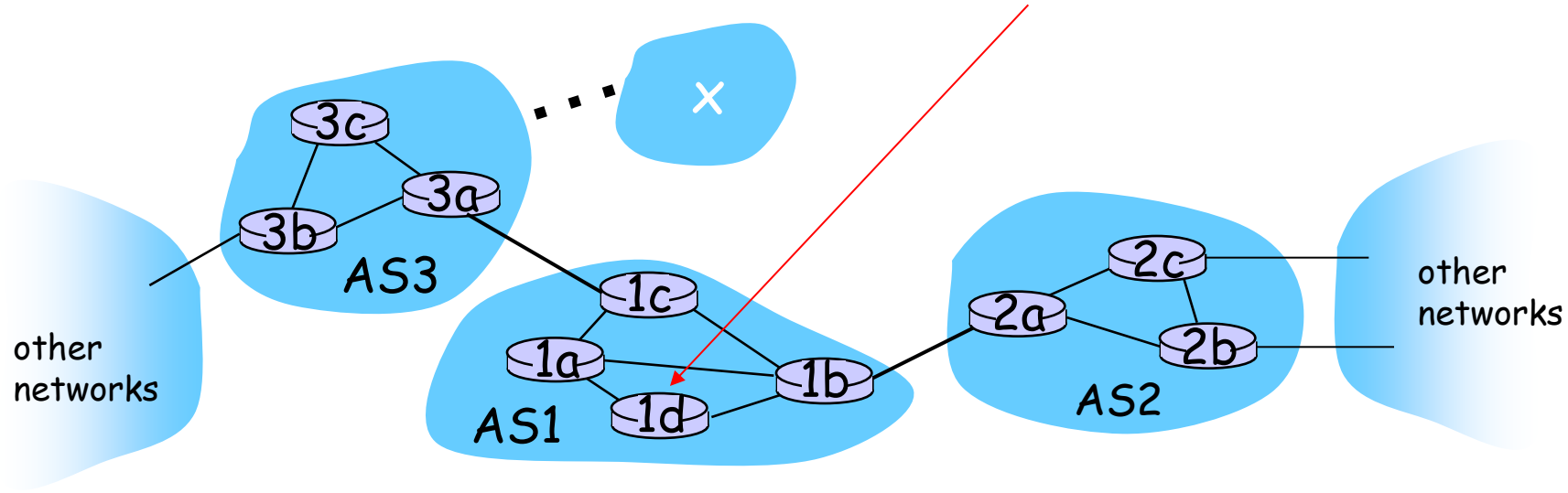
1. Aprender que destinos son alcanzables mediante AS2 y cuales porAS3 [sino bastaría tener una entrada por defecto]
2. Propagar esta información de "alcanzabilidad" a todos los routers en AS1

Responsabilidad de enrutado inter-AS!



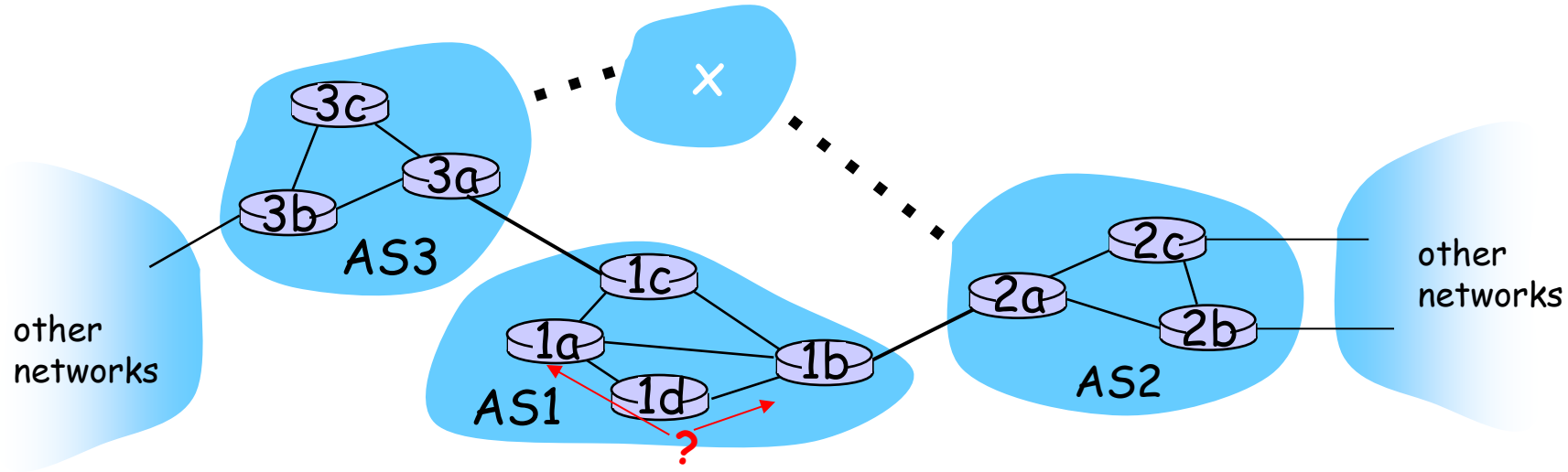
Ejemplo: Definiendo tabla de reenvío en router 1d

- AS1 aprende (vía protocolo **inter-AS**) que la red [esto es en particular, prefijo IP] *x* es alcanzable vía AS3 (gateway 1c) pero no vía AS2.
  - Protocolo inter-AS propaga la info. a todos los routers internos
- Router 1d determina usando info. de enrutado intra-AS que su interface *I* es la de menor coste a 1c.
  - Inserta una entrada en la tabla de reenvios *(x,I)*



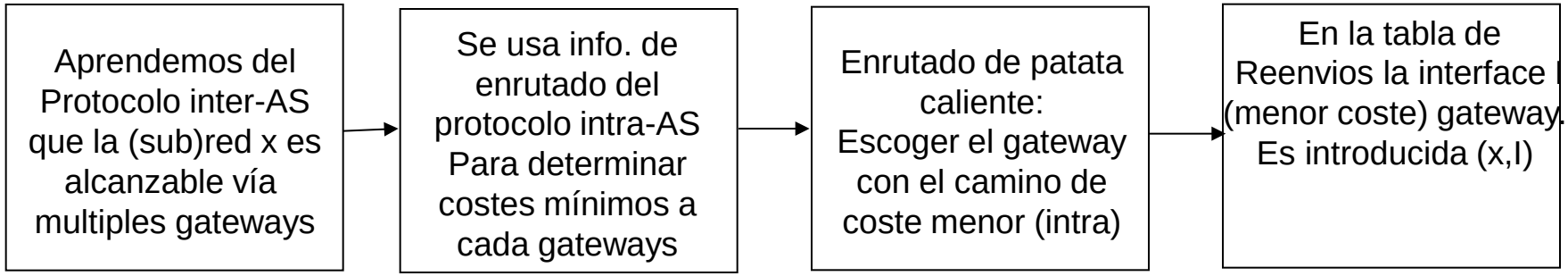
Ejemplo: Escogiendo entre multiples ASes

- Ahora, AS1 aprende mediante el protocolo inter-AS que la subred *x* es alcanzable desde AS3 *y* desde AS2.
- Para configurar la tabla de reenvío, el router 1d debe determinar que puerta de enlace debe reenviar el paquete hacía el destino x



Ejemplo: Escogiendo entre multiples ASs*

- Para configurar la tabla de reenvíos, el router 1d debe determinar hacia que puerta de enlace debe ser dirigido para el destino x
- Opciones:
  - *hot potato routing* [patata caliente]: enviar el paquete hacia la puerta de enlace de menor costo, tal como se ha definido mediante el algoritmo **intra-routing**
  - Minimizar el número de AS en la ruta
  - Azar
  - Precio/Política



Capítulo 4: Capa de red

4. 1 Introducción

4.2 Circuitos virtuales y datagramas

4.3 Que hay dentro de un router

4.4 IP: Protocolo de Internet

- Formato
- Fragmentación
- Direccionamiento IPv4
- DHCP y NAT
- ICMP
- IPv6

4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector distancia
- Enrutamiento jerárquico

4.6 Enrutamiento en Internet

- RIP
- OSPF
- BGP
- Tipo de relaciones

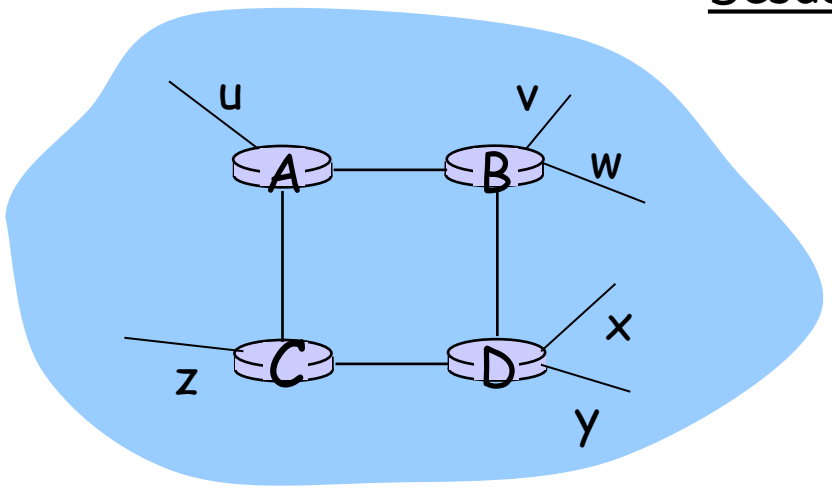
4.7 Enrutamiento Broadcast y multicast

Enrutado Intra-AS

- También conocido como Interior Gateway Protocols (IGP) [protocolo de pasarela (=puerta de enlace) interior (ojo!)]
- Los protocolos de enrutado Intra-AS (ojo!) más comunes :
  - **RIP**: Routing Information Protocol
  - **OSPF**: Open Shortest Path First
  - IGRP: Interior Gateway Routing Protocol (Cisco proprietary)

RIP (Routing Information Protocol)

- Introducido en versiones de UNIX en 1982 basado en trabajo de Xerox, actualmente al menos versión 2!
- Se basa en el algoritmo de **vector de distancias**:
  - Métrica de distancia: **# saltos** (max = 15 saltos), cada enlace tiene, por tanto, coste 1
  - VDs son intercambiados entre los vecinos cada (aprox.) 30 sec [conocidos como anuncios RIP]
  - Cada anuncio lista hasta 25 destinos *subredes (: prefijos IP)*



Desde router A a prefijo destino:

Prefijo	saltos
u	1
v	2
w	2
x	3
y	3
z	2

RIP: Ejemplo (router D)

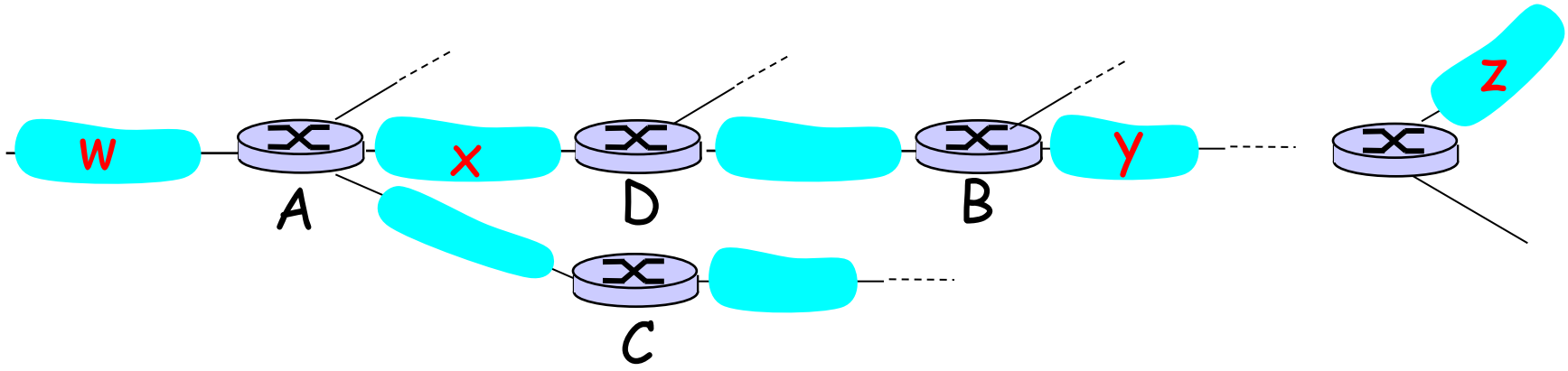


Tabla de enrutamiento en RIP == vector de distancias (en este caso se muestra el vector propio tras posibles actualizaciones) + tabla de reenvío

Subred destino	Sig. router	# saltos a dest
w	A	2
y	B	2
z	B	7
x	--	1
....	....	....

RIP: Ejemplo

Anuncio A-D

dest.	Sigui.	saltos
w	-	1
x	-	1
z	C	4
....	..	...

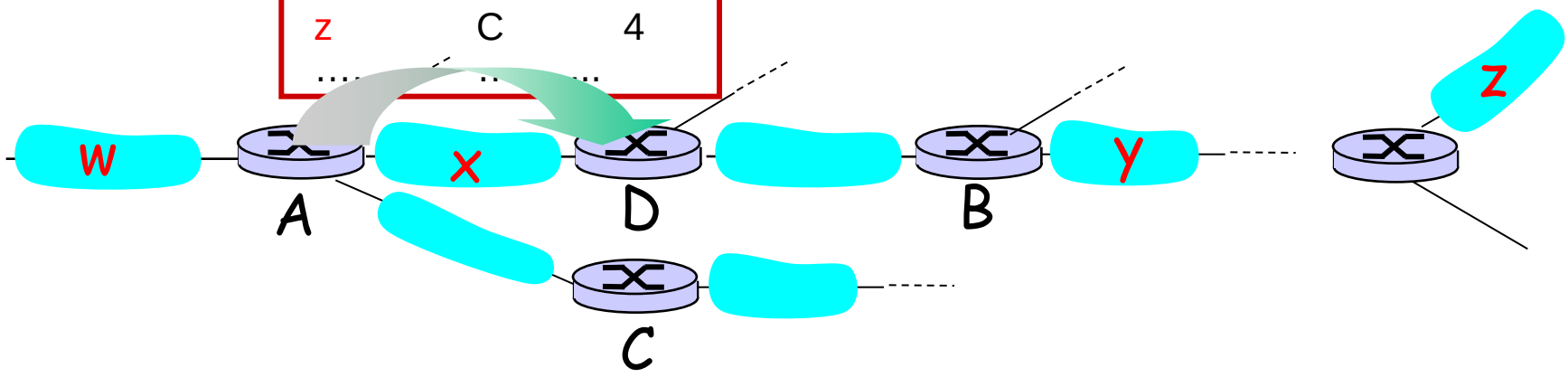


Tabla de enrutamiento router D

Subred destino	Sig. router	# saltos a dest
w	A	2
y	B	2
z	B → A	7 → 5
x	--	1
....	....	....

RIP: Fallos en los enlaces y recuperación

Si no se reciben anuncios después de 180 s : el vecino y su enlace directo se consideran "muertos"

- Las rutas vía ese vecino son inválidas (modifica tabla enrutamiento local)
- Se informa a los vecinos
- Los vecinos nos responden (si las tablas cambian)
- *Emplea poison reverse* (inversa envenenada), en este caso : distancia infinita = 16 saltos

OSPF (Open Shortest Path First)

- "open": Público de libre acceso
- Usa algoritmo de **estado de los enlaces**
  - Se envían paquetes en inundación a todo el AS
  - De este modo se recupera la topología y los costes (típicamente el administrador de red decide la métrica saltos/inv de la capacidad e.g.)
  - Las rutas se computa usando Dijkstra
  - Se envía información por cada cambio o al menos 1 vez cada 40 minutos

Enrutado Inter-AS (Internet): BGP

- BGP (Border Gateway Protocol): El protocolo aceptado por todos como algoritmo de enrutado inter-AS
  - Se define como el *pegamento* que une Internet
- BGP permite a cada AS:
  - eBGP: obtener la alcanzabilidad de red de sus ASs vecinos
  - iBGP: Propagar esta información a todos los routers del AS.
  - Determinar rutas
- Permite a las redes darse a conocer en Internet!

BGP selección de ruta

- Si los routers tienen más de una opción para un AS destino, la selección en BGP es en este orden [¿controlable?]:
  1. Política establecida por el gestor : muy frecuente, falta de colaboración (competitividad)
  2. Menor número de AS : vector de distancias
  3. Puerta de enlace con ruta más corta dentro del AS : patata caliente
  4. Criterios adicionales

Por qué diferente enrutado Intra- y Inter-AS?

Política:

- Inter-AS: La administración puede controlar el tráfico que la atraviesa, en términos económicos y competitivos
- Intra-AS: Una sola administración

Escala:

- Los router dentro de un sistema autónomo pueden ser pocos, lo que motiva un algoritmo tipo EL, en cambio, ante un AS grande (o el propio Internet), es más razonable un algoritmo de VD
- Rendimiento:
  - Intra-AS: esencialmente el objetivo final
  - Inter-AS: Primero términos políticos

Capítulo 4: Capa de red

4. 1 Introducción

4.2 Circuitos virtuales y datagramas

4.3 Que hay dentro de un router

4.4 IP: Protocolo de Internet

- Formato
- Fragmentación
- Direccionamiento IPv4
- DHCP y NAT
- ICMP
- IPv6

4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector distancia
- Enrutamiento jerárquico

4.6 Enrutamiento en Internet

- RIP
- OSPF
- BGP
- Tipo de relaciones

4.7 Enrutamiento Broadcast y multicast

Tipo de relaciones

- Peering (==entre iguales/pares)
  - Ambos extremos se intercambian rutas de sus redes y clientes
  - En principio sin coste pero cumpliendo requisitos
    - http://www.att.com/peering/
- Proveedor-cliente
  - El proveedor permite servir como transito para el tráfico entrante y saliente de sus clientes
  - El cliente paga por este tráfico
- ¿Cómo? → Puntos de intercambio
    - IXP
    - Enlace directo

Capítulo 4: Capa de red

4. 1 Introducción

4.2 Circuitos virtuales y datagramas

4.3 Que hay dentro de un router

4.4 IP: Protocolo de Internet

- Formato
- Fragmentación
- Direccionamiento IPv4
- DHCP y NAT
- ICMP
- IPv6

4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector distancia
- Enrutamiento jerárquico

4.6 Enrutamiento en Internet

- RIP
- OSPF
- BGP
- Tipo de relaciones

4.7 Enrutamiento Broadcast y multicast