

# Capítulo 4

## La capa de red

*Redes de computadores*  
*Bloque 1*

### 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

All material copyright 1996-2010  
J.F Kurose and K.W. Ross, All Rights Reserved



*Redes de computadores: Un enfoque descendente, 5ª edición.*  
Jim Kurose, Keith Ross  
Pearson Educación,  
2010.

## Capítulo 4: la capa de red

### Objetivos del capítulo:

- ❖ Comprender los principios subyacentes a los servicios de la capa de red:
  - modelos de servicio en la capa de red
  - reenvío <-> enrutamiento
  - cómo funciona un router
  - enrutamiento (= selección de camino)
  - difusión, multidifusión
- ❖ instanciación, implementación en Internet

## Capítulo 4: la capa de red

### 4.1 Introducción

#### 4.2 Redes de circuitos virtuales y de datagramas

#### 4.3 Las tripas de un router

#### 4.4 IP: Internet Protocol

- Formato de datagramas
- Direccionamiento IPv4
- ICMP
- IPv6

#### 4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector de distancias
- Enrutamiento jerárquico

#### 4.6 Enrutamiento en Internet

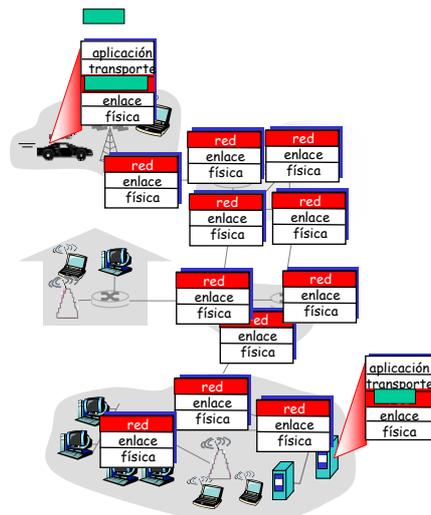
- RIP
- OSPF
- BGP

#### 4.7 Enrutamiento por difusión y multidifusión

Capa de red 4-3

## Capa de red

- ❖ transporta segmentos del host emisor al receptor
- ❖ en el lado emisor, encapsula segmentos en datagramas
- ❖ en el receptor, entrega segmentos a la capa de transporte
- ❖ los protocolos de la capa de red están en *cada* host y router
- ❖ el router examina los campos de la cabecera de todos los datagramas IP que lo atraviesan



Capa de red 4-4

## Dos funciones clave en la capa de red

❖ *reenvío*: mover paquetes de la entrada del router a la salida apropiada

❖ *enrutamiento*: determinar la ruta a tomar por los paquetes desde el origen al destino

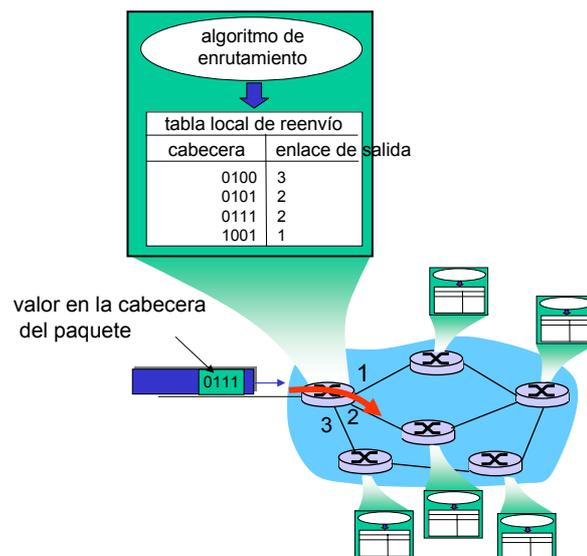
- *algoritmos de enrutamiento*

analogía:

❖ *enrutamiento*: planear un viaje desde Madrid a Valencia

❖ *reenvío*: proceso de elegir cada cruce y/o salida de la autopista

## Relación enrutamiento <-> reenvío



## Configuración de conexión

- ❖ 3ª importante función en *algunas* arquitecturas de red:
  - ATM, frame relay, X.25
- ❖ antes de que comience el flujo de datagramas, los dos hosts terminales **más** todos los routers intermedios establecen una conexión virtual
  - routers involucrados
- ❖ serv. de conexión en capa de transp. <-> capa de red:
  - **red:** entre dos hosts (puede involucrar routers en el caso de circuitos virtuales)
  - **transporte:** entre dos procesos

Capa de red 4-7

## Modelo de servicio de red

**P:** ¿Qué *modelo de servicio* para el "canal" que transporta los datagramas del origen al destino?

### posibles servicios para datagramas individuales:

- ❖ entrega garantizada
- ❖ entrega garantizada con retardo acotado (p. ej. < 40 ms)

### posibles servicios para un flujo de datagramas:

- ❖ entrega de datagramas en orden
- ❖ ancho de banda mínimo garantizado
- ❖ restricciones en la fluctuación entre paquetes

Capa de red 4-8

## Modelos de servicio de la capa de red:

Arquitectura de red	Modelo de servicio	¿Garantías?			¿Informa sobre congestión?	
		Ancho de banda	P	O		T
Internet	“como puedas”	ninguna velocidad	no	no	no	no (inferido por pérdidas)
ATM	CBR	constante	sí	sí	sí	no hay congestión
ATM	VBR	velocidad garantizada	sí	sí	sí	no hay congestión
ATM	ABR	mínimo garantizado	no	sí	no	sí
ATM	UBR	ninguna	no	sí	no	no

Capa de red 4-9

## Capítulo 4: la capa de red

### 4.1 Introducción

### 4.2 Redes de circuitos virtuales y de datagramas

### 4.3 Las tripas de un router

### 4.4 IP: Internet Protocol

- Formato de datagramas
- Direccionamiento IPv4
- ICMP
- IPv6

### 4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector de distancias
- Enrutamiento jerárquico

### 4.6 Enrutamiento en Internet

- RIP
- OSPF
- BGP

### 4.7 Enrutamiento por difusión y multidifusión

Capa de red 4-10

## Servicios de la capa de red con y sin conexión

- ❖ la red de datagramas proporciona un servicio de red sin conexión
- ❖ la red de circuitos virtuales (VC) proporciona un servicio de red con conexión
- ❖ analogías con la capa de transporte, pero:
  - **servicio:** host-a-host
  - **sin elección:** cada red da uno u otro
  - **implementación:** en el núcleo de la red

Capa de red 4-11

## Circuitos virtuales

"la ruta origen-destino es como un circuito telefónico"

- orientación a rendimiento
  - acciones de la red a lo largo de la ruta origen-destino
- 
- ❖ establecimiento de llamada antes del flujo de datos
  - ❖ cada paquete porta un identificador de VC (y no la dirección del host destino)
  - ❖ *cada* router en la ruta mantiene el "estado" de *cada* conexión activa
  - ❖ los recursos de enlaces y routers (ancho de banda, buffers) pueden *reservarse* a un VC (recurso dedicado = servicio predecible)

Capa de red 4-12

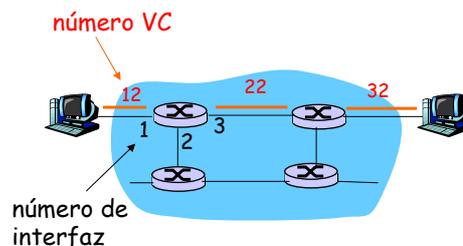
## Implementación de VC

un VC consiste en:

1. una ruta desde origen a destino
  2. números de VC, a razón de un número por cada enlace a lo largo de la ruta
  3. entradas en las tablas de reenvío de los routers atravesados
- ❖ un paquete que pertenece a un VC porta el número de VC (y no la dirección del destino)
  - ❖ el número de VC puede cambiar en cada enlace
    - el nuevo número sale de la tabla de reenvío

Capa de red 4-13

## Tabla de reenvío para un VC



### Tabla de reenvío del router noroeste:

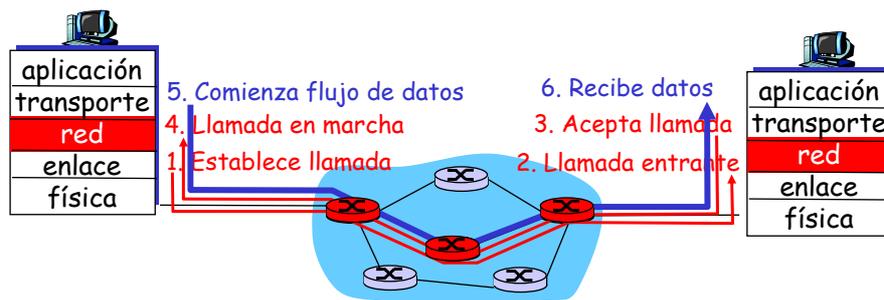
Interfaz entrada	Nº VC entrada	Interfaz salida	Nº VC salida
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
...	...	...	...

**¡Los routers mantienen la info de conectividad activa!**

Capa de red 4-14

## VC: protocolos de señalización

- ❖ usados para establecer, mantener y desconectar un VC
- ❖ usados en ATM, frame-relay, X.25
- ❖ no usados en Internet a día de hoy



Capa de red 4-15

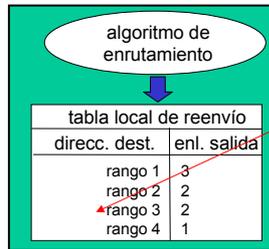
## Redes de datagramas

- ❖ no hace falta establecimiento de llamada en la capa de red
- ❖ routers: no guardan estado de las conexiones entre terminales
  - no existe el concepto de "conexión"
- ❖ los paquetes se reenvían usando la dirección de host destino
  - los paquetes entre el mismo par origen-destino pueden llevar cada uno una ruta distinta



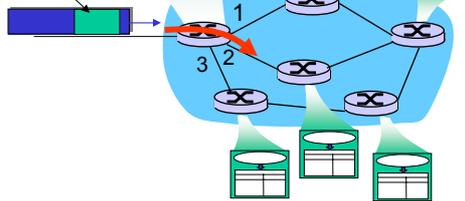
Capa de red 4-16

## Tabla de reenvío de datagramas



¡4 mil millones de direcciones IP!  
Se guardan *rangos* de direcciones y no direcciones particulares (agregación de entradas)

dirección IP destino en cabecera del paquete entrante



Capa de red 4-17

## Tabla de reenvío de datagramas

Rango de direcciones destino	Link Interface
11001000 00010111 00010000 00000000 a	0
11001000 00010111 00010111 11111111	
11001000 00010111 00011000 00000000 a	1
11001000 00010111 00011000 11111111	
11001000 00010111 00011001 00000000 a	2
11001000 00010111 00011111 11111111	
en otro caso	3

P: ¿Qué pasa si los rangos no se pueden separar así de bien?

Capa de red 4-18

## Regla del prefijo más largo

### Coincidencia del prefijo más largo

al buscar en la tabla de reenvío una dirección de destino dada, hay que usar el prefijo con la coincidencia más larga

Rango de direcciones destino	Link interface
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001000 00010111 00011*** *****	2
en otro caso	3

### Ejemplos:

DA: 11001000 00010111 0001**0110** 10100001    ¿Qué interfaz?

DA: 11001000 00010111 0001**1000** 10101010    ¿Qué interfaz?

## ¿Por qué redes VC y de datagramas?

### Internet (datagrama)

- ❖ Intercambio de datos entre computadores
  - Servicio "elástico", no necesita temporización estricta
- ❖ sistemas terminales "inteligentes" (computadores)
  - pueden adaptarse, controlar, recuperarse de errores
  - el núcleo de la red se mantiene simple, la complejidad está en la "frontera"
  - diferentes características
  - dar servicio igualitario es difícil

### ATM (VC)

- ❖ evolución a partir de la telefonía
- ❖ en la conversación humana:
  - requisitos de temporización estricta y de fiabilidad
  - servicios garantizados
- ❖ terminales "tontos"
  - teléfonos...
  - la complejidad va *dentro* de la red

## Capítulo 4: la capa de red

### 4.1 Introducción

### 4.2 Redes de circuitos virtuales y de datagramas

### 4.3 Las tripas de un router

### 4.4 IP: Internet Protocol

- Formato de datagramas
- Direccionamiento IPv4
- ICMP
- IPv6

### 4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector de distancias
- Enrutamiento jerárquico

### 4.6 Enrutamiento en Internet

- RIP
- OSPF
- BGP

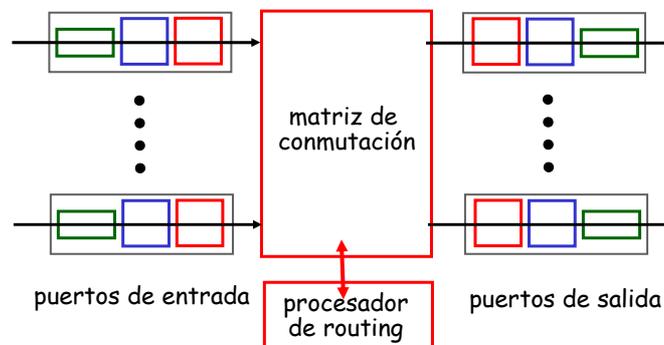
### 4.7 Enrutamiento por difusión y multidifusión

Capa de red 4-21

## Arquitectura de un router

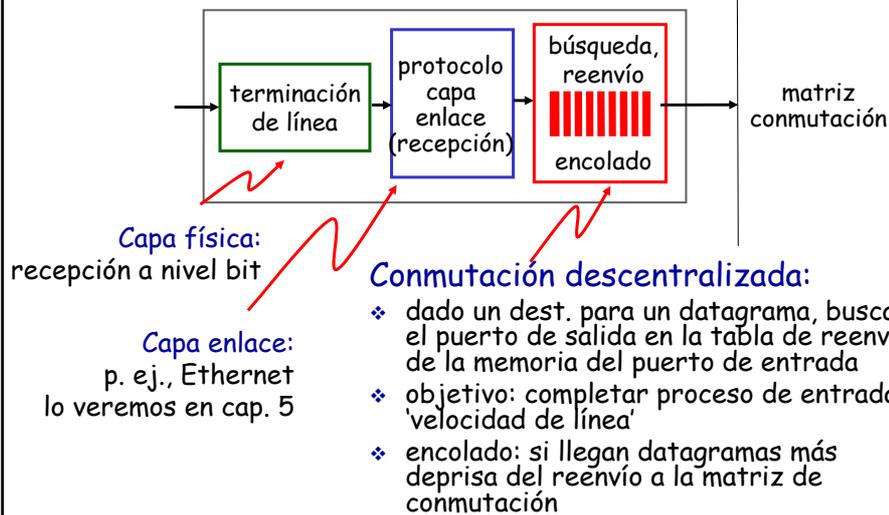
dos funciones clave en el router:

- ❖ ejecutar prot./alg. de routing (RIP, OSPF, BGP)
- ❖ *reenviar* datagramas del enlace de ent. al de sal.



Capa de red 4-22

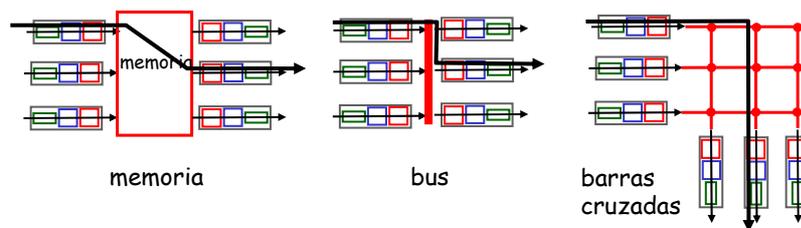
## Funciones del puerto de entrada



Capa de red 4-23

## Matriz de conmutación

- ❖ transferir el paquete del buffer de entrada al buffer de salida correcto
- ❖ velocidad de conmutación: a la que se pueden transferir paquetes desde las entradas a las salidas
  - se suele medir en múltiplos de la velocidad de línea
  - N entradas: deseable velocidad de conmutación = N veces la velocidad de línea
- ❖ tres tipos de matriz de conmutación

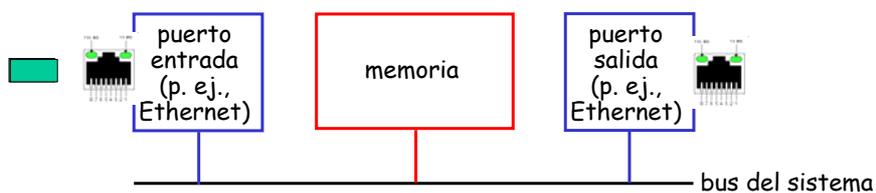


Capa de red 4-24

## Conmutación usando memoria

### Routers de primera generación:

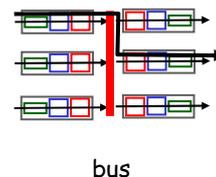
- ❖ computadores tradicionales, la conmutación la hace la CPU directamente
- ❖ se copia el paquete a la memoria del sistema
- ❖ velocidad limitada al ancho de banda de la memoria (hay que cruzarlo dos veces por datagrama)



Capa de red 4-25

## Conmutación por bus

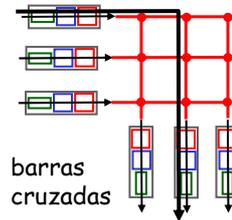
- ❖ de entrada a salida, el datagrama viaja por un bus compartido
- ❖ **contención del bus:** la conmutación limitada por el ancho de banda del bus
- ❖ Cisco 5600, bus de 32 Gbps: suficiente para routers corporativos o de acceso



Capa de red 4-26

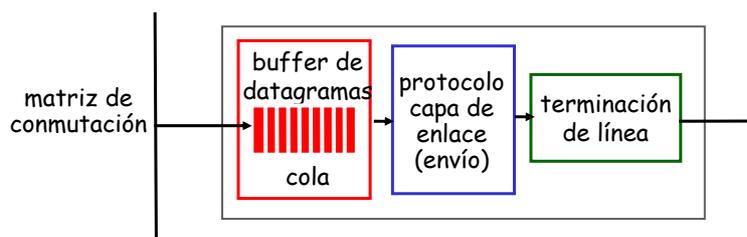
## Conmutación mediante una red

- ❖ no sufre limitaciones de ancho de banda
- ❖ redes tipo Banyan, barras cruzadas, u otros tipos (desarrollados inicialmente para conectar procesadores en sistemas multiprocesador)
- ❖ diseño avanzado: fragmentación del datagrama en celdas de long. fija, conmuta celdas por la matriz
- ❖ Cisco 12000: 60 Gbps a través de una red



Capa de red 4-27

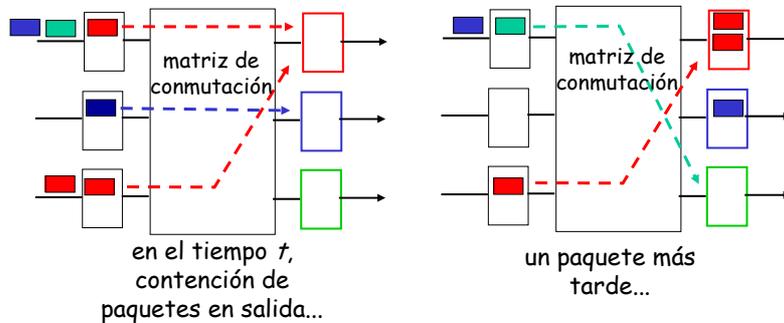
## Puertos de salida



- ❖ **buffering**: se necesita cuando los datagramas llegan de la matriz más rápido de lo que se pueden transmitir
- ❖ **planificación de paquetes**: elige un datagrama de entre los que están a la cola para transmitirlo

Capa de red 4-28

## Cola en el puerto de salida



- ❖ buffering si la velocidad de llegada excede la velocidad de línea de salida
- ❖ *imposibles colas (=retraso) y pérdidas de paquetes, por desbordamiento en el buffer!*

Capa de red 4-29

## ¿Cuánto buffering?

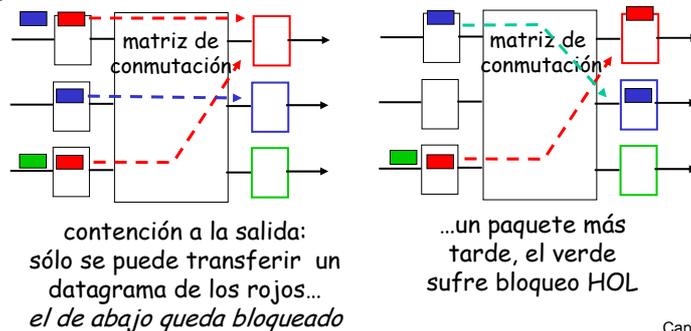
- ❖ Según RFC 3439 a "ojo de buen cubero": tamaño del buffer igual al "típico" RTT (digamos 250 ms) multiplicado por la capacidad del enlace  $C$ 
  - ej., enlace  $C = 10$  Gpbs  $\rightarrow$  2.5 Gbit buffer
- ❖ recomendación reciente: con  $N$  flujos, buffering igual a

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

Capa de red 4-30

## Cola del puerto de entrada

- ❖ si la matriz admite datos más despacio que la combinación de todos los puertos de entrada -> habrá colas en los puertos de entrada
  - *implica retrasos y tal vez pérdidas si se desborda el buffer*
- ❖ **Bloqueo en la cabeza de línea (HOL):** un datagrama bloqueado al principio de la cola impide avanzar a los que están detrás



Capa de red 4-31

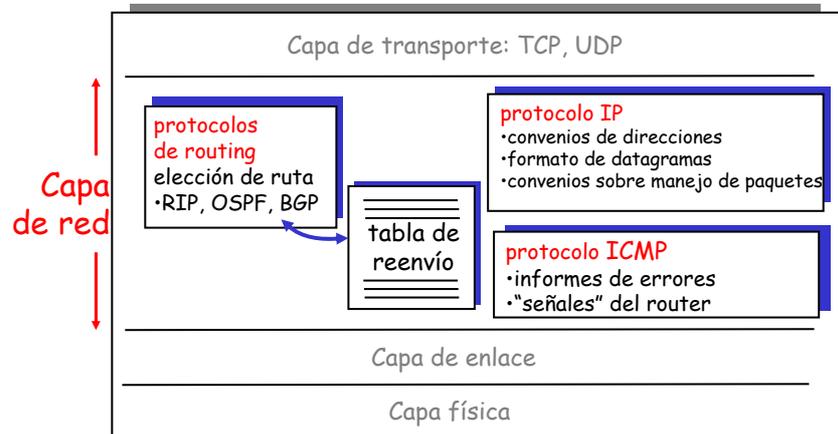
## Capítulo 4: la capa de red

- 4.1 Introducción
- 4.2 Redes de circuitos virtuales y de datagramas
- 4.3 Las tripas de un router
- 4.4 IP: Internet Protocol**
  - Formato de datagramas
  - Direccionamiento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de enrutamiento
  - Estado de enlaces
  - Vector de distancias
  - Enrutamiento jerárquico
- 4.6 Enrutamiento en Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Enrutamiento por difusión y multidifusión

Capa de red 4-32

## Capa de red en Internet

Funciones de la capa de red en host y router:



Capa de red 4-33

## Capítulo 4: la capa de red

### 4.1 Introducción

### 4.2 Redes de circuitos virtuales y de datagramas

### 4.3 Las tripas de un router

### 4.4 IP: Internet Protocol

- Formato de datagramas
- Direccionamiento IPv4
- ICMP
- IPv6

### 4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector de distancias
- Enrutamiento jerárquico

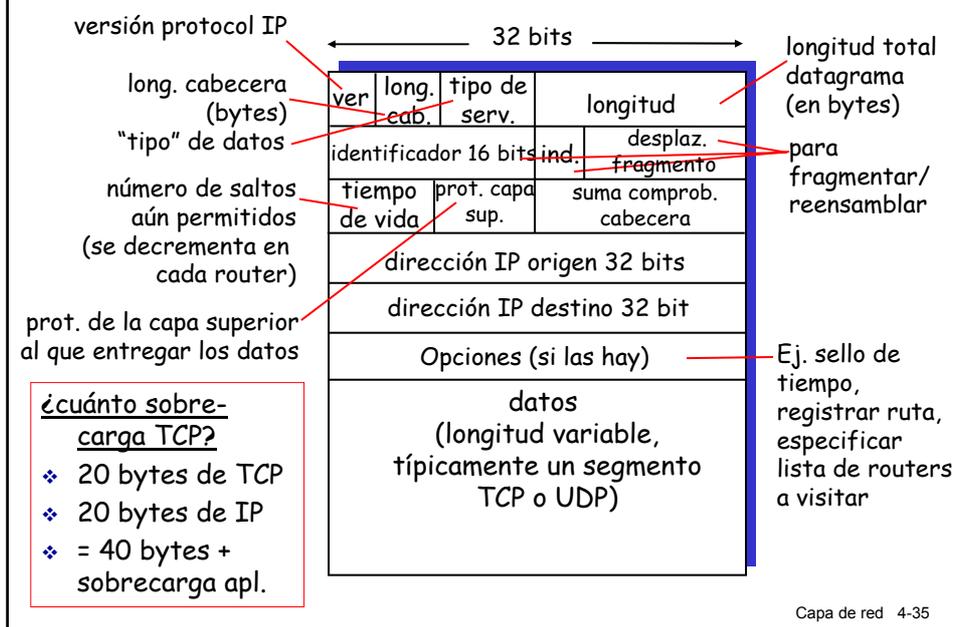
### 4.6 Enrutamiento en Internet

- RIP
- OSPF
- BGP

### 4.7 Enrutamiento por difusión y multidifusión

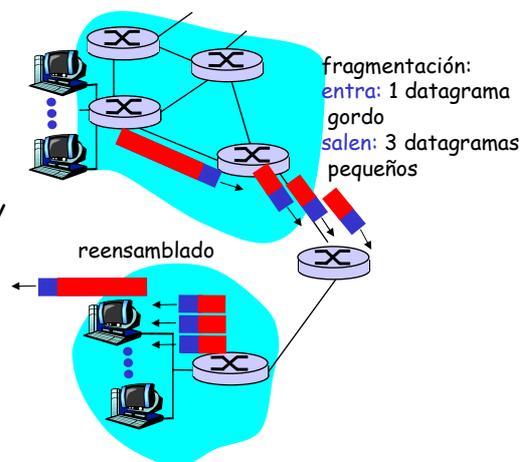
Capa de red 4-34

## Formato de datagrama IP



## IP: Fragmentación & Reensamblado

- ❖ los enlaces de red están limitados por la MTU (max. transmission unit): la trama más larga posible en el nivel de enlace
  - diferentes enlaces => diferentes MTUs
- ❖ un datagrama IP "gordo" hay que dividirlo ("fragmentarlo") en la red
  - un datagrama pasa a ser varios datagramas
  - "reensamblado" únicamente en destino
  - los bits de la cabecera IP identifican y ordenan los fragmentos de trama



## IP: Fragmentación & Reensamblado

### Ejemplo

- ❖ datagrama de 4000 bytes
- ❖ MTU = 1500 bytes

1480 bytes en el campo de datos

despl. = 1480/8

long.	ID	indicador	despl.
=4000	=x	=0	=0

Un datagrama gordo se convierte en varios datagramas pequeños

long.	ID	indicador	despl.
=1500	=x	=1	=0
=1500	=x	=1	=185
=1040	=x	=0	=370

## Capítulo 4: la capa de red

### 4.1 Introducción

### 4.2 Redes de circuitos virtuales y de datagramas

### 4.3 Las tripas de un router

### 4.4 IP: Internet Protocol

- Formato de datagramas
- **Direccionamiento IPv4**
- ICMP
- IPv6

### 4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector de distancias
- Enrutamiento jerárquico

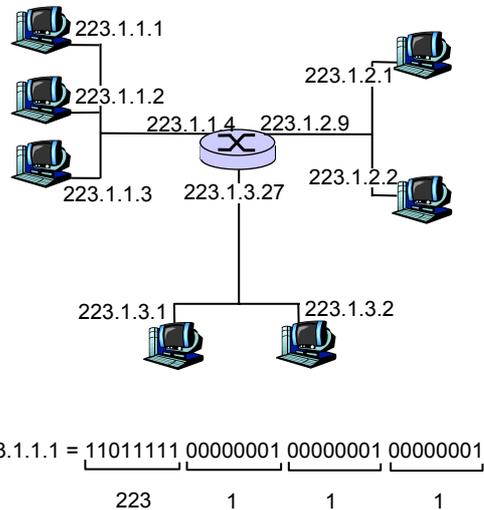
### 4.6 Enrutamiento en Internet

- RIP
- OSPF
- BGP

### 4.7 Enrutamiento por difusión y multidifusión

## Direccionamiento IP: introducción

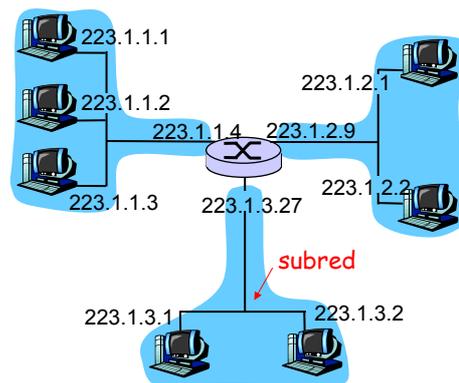
- ❖ **dirección IP:** identificador de 32 bits para hosts e *interfaces* de router
- ❖ **interfaz:** conexión entre host/router y enlace físico
  - un router tiene típicamente muchas interfaces
  - un host típicamente sólo una interfaz
  - la dirección IP se asocia con *cada* interfaz



Capa de red 4-39

## Subredes

- ❖ **dirección IP:**
  - parte subred (bits más significativos)
  - parte host (bits menos significativos)
- ❖ **P: ¿Qué es una subred?**
  - conjunto de dispositivos con la misma subred en su dirección IP
  - pueden "verse" entre sí sin la intervención de un router



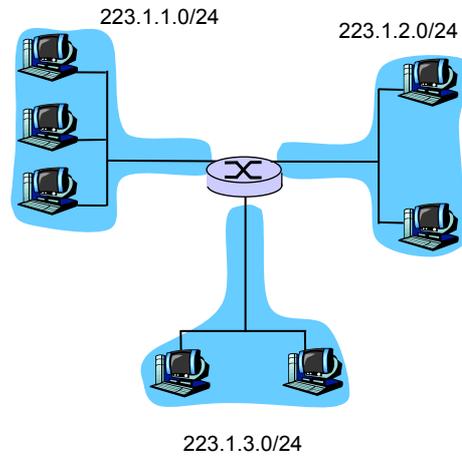
la red consiste en 3 subredes

Capa de red 4-40

# Subredes

## Receta

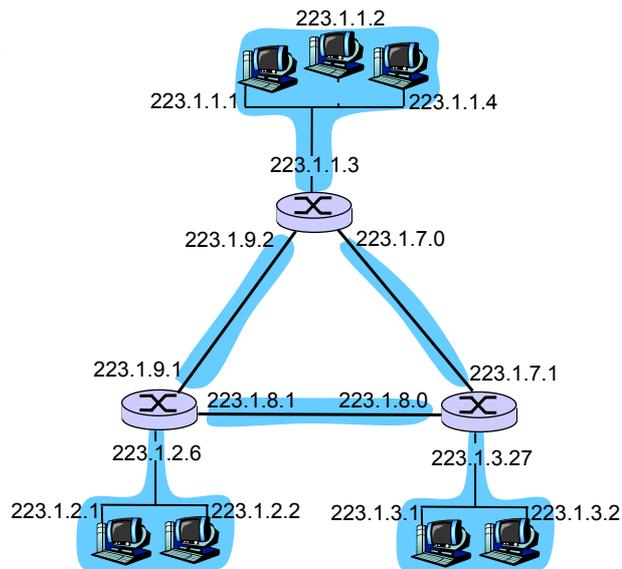
- ❖ para determinar las subredes, separar cada interfaz de host o router, creando islas de redes aisladas
- ❖ cada red aislada se denomina **subred**.



Máscara de subred: /24

# Subredes

¿Cuántas?



## direccionamiento IP: CIDR

### CIDR: Classless InterDomain Routing

- porción subred de longitud arbitraria
- formato de direcciones: **a.b.c.d/x**, donde x es el nº de bits en la porción de subred de la dirección



## direcciones IP: ¿cómo consigo una?

### P: ¿Cómo consigue un *host* una dirección IP?

- ❖ fijada por el administrador del sistema en un fichero
  - Windows: panel-de-control -> red -> configuración -> tcp/ip -> propiedades
  - UNIX: /etc/rc.config
- ❖ **DHCP:** Dynamic Host Configuration Protocol: consigue dinámicamente una desde un servidor
  - "plug-and-play"

## DHCP: Dynamic Host Configuration Protocol

**Objetivo:** permite a un host obtener de un servidor dinámicamente su dirección IP en el momento de agregarse a la red

Puede renovar el "préstamo" de su dirección

Permite re-usar direcciones (los nodos apagados no consumen dirección)

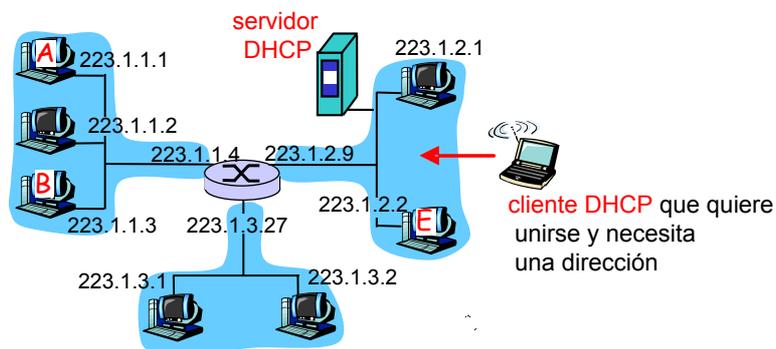
Da soporte a usuarios móviles, que quieren unirse a la red (más brevemente)

resumen de mensajes de DHCP:

- el host difunde "DHCP discover" [opcional]
- el servidor DHCP contesta con "DHCP offer" [opcional]
- el host pide dirección IP: "DHCP request"
- el servidor DHCP envía la dirección: "DHCP ack"

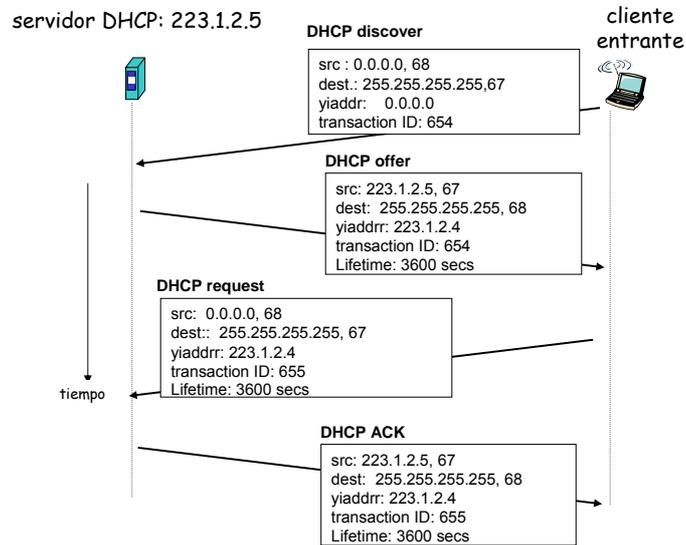
Capa de red 4-45

## escenario DHCP cliente-servidor



Capa de red 4-46

## escenario DHCP cliente-servidor



, a de red 4-47

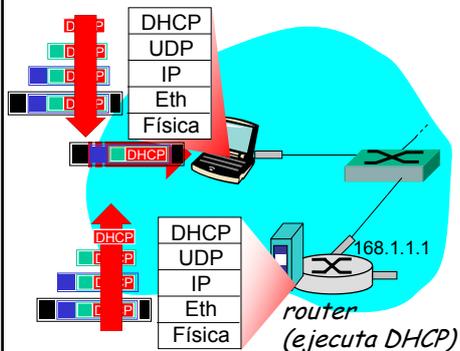
## DHCP: más que direcciones IP

DHCP puede devolver más datos, además de la IP:

- dirección del router de primer salto
- nombre y dirección IP del servidor DNS
- máscara de red (especificando la porción de subred y de host en la dirección IP)

Capa de red 4-48

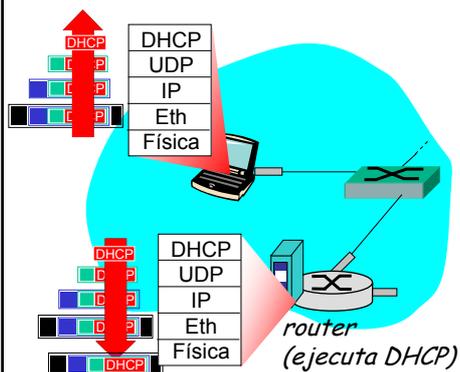
## DHCP: ejemplo



- ❖ para conectar el portátil necesita una IP, la IP del router de primer salto y la IP del servidor DNS: hay que usar DHCP
- ❖ la solicitud DHCP se encapsula en UDP, que se encapsula en IP, que se encapsula en 802.1 Ethernet
- ❖ la trama Ethernet (dest: FFFFFFFF) difundida en la red se recibe en el route que ejecuta el servicio DHCP
- ❖ de la trama Ethernet se extrae la trama IP, de ahí la UDP y de ahí la solicitud DHCP

Capa de red 4-49

## DHCP: ejemplo



- ❖ el servidor DHCP emite un mensaje DHCP ACK conteniendo la IP del cliente, la IP del router de primer salto y la IP del servidor DNS
- ❖ el mensaje se encapsula y se reenvía al cliente, el cual extrae la información
- ❖ ahora el cliente sabe su IP, nombre e IP del servidor DNS e IP del router de primer salto

Capa de red 4-50

## DHCP: captura con Wireshark (LAN casera)

Message type: **Boot Request (1)**  
 Hardware type: Ethernet  
 Hardware address length: 6  
 Hops: 0  
**Transaction ID: 0x6b3a11b7**  
 Seconds elapsed: 0  
 Bootp flags: 0x0000 (Unicast)  
 Client IP address: 0.0.0.0 (0.0.0.0)  
 Your (client) IP address: 0.0.0.0 (0.0.0.0)  
 Next server IP address: 0.0.0.0 (0.0.0.0)  
 Relay agent IP address: 0.0.0.0 (0.0.0.0)  
**Client MAC address: Wistron\_23:68:8a (00:16:d3:23:68:8a)**  
 Server host name not given  
 Boot file name not given  
 Magic cookie: (OK)  
 Option: (t=53,l=1) **DHCP Message Type = DHCP Request**  
 Option: (61) Client identifier  
 Length: 7; Value: 010016D323688A;  
 Hardware type: Ethernet  
 Client MAC address: Wistron\_23:68:8a (00:16:d3:23:68:8a)  
 Option: (t=50,l=4) Requested IP Address = 192.168.1.101  
 Option: (t=12,l=5) Host Name = "nomad"  
**Option: (55) Parameter Request List**  
 Length: 11; Value: 010F03062C2E2F1F21F92B  
**1 = Subnet Mask; 15 = Domain Name**  
**3 = Router; 6 = Domain Name Server**  
 44 = NetBIOS over TCP/IP Name Server  
 .....

solicitud

Message type: **Boot Reply (2)**  
 Hardware type: Ethernet  
 Hardware address length: 6  
 Hops: 0  
**Transaction ID: 0x6b3a11b7**  
 Seconds elapsed: 0  
 Bootp flags: 0x0000 (Unicast)  
**Client IP address: 192.168.1.101 (192.168.1.101)**  
 Your (client) IP address: 0.0.0.0 (0.0.0.0)  
**Next server IP address: 192.168.1.1 (192.168.1.1)**  
 Relay agent IP address: 0.0.0.0 (0.0.0.0)  
 Client MAC address: Wistron\_23:68:8a (00:16:d3:23:68:8a)  
 Server host name not given  
 Boot file name not given  
 Magic cookie: (OK)  
**Option: (t=53,l=1) DHCP Message Type = DHCP ACK**  
**Option: (t=54,l=4) Server Identifier = 192.168.1.1**  
**Option: (t=1,l=4) Subnet Mask = 255.255.255.0**  
**Option: (t=3,l=4) Router = 192.168.1.1**  
**Option: (6) Domain Name Server**  
 Length: 12; Value: 445747E2445749F244574092;  
 IP Address: 68.87.71.226;  
 IP Address: 68.87.73.242;  
 IP Address: 68.87.64.146  
**Option: (t=15,l=20) Domain Name = "hsd1.ma.comcast.net."**

respuesta

Capa de red 4-51

## direcciones IP: ¿cómo consigo una?

**P:** ¿cómo consigue la red la parte subred de la IP?

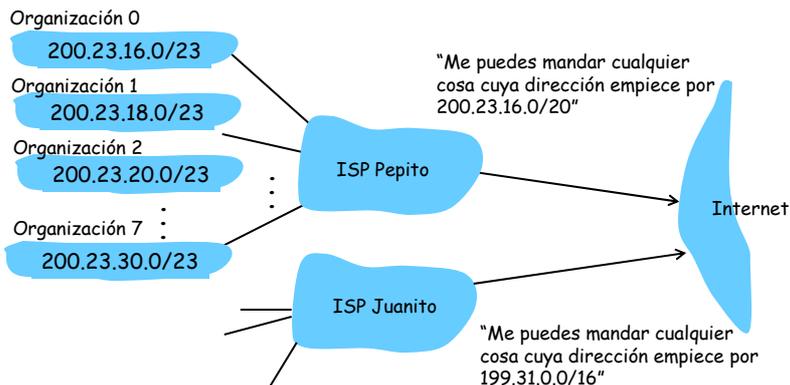
**R:** gracias a la porción de direcciones reservada a su ISP (proveedor de servicio internet)

bloque ISP	<u>11001000 00010111 00010000</u> 00000000	200.23.16.0/20
Organización 0	<u>11001000 00010111 00010000</u> 00000000	200.23.16.0/23
Organización 1	<u>11001000 00010111 00010010</u> 00000000	200.23.18.0/23
Organización 2	<u>11001000 00010111 00010100</u> 00000000	200.23.20.0/23
...	.....	....
Organización 7	<u>11001000 00010111 00011110</u> 00000000	200.23.30.0/23

Capa de red 4-52

## direccionamiento jerarquizado: agregación de rutas

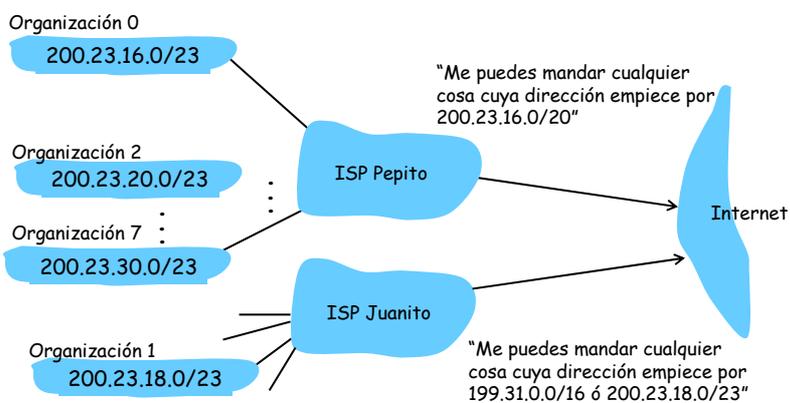
El direccionamiento jerarquizado permite propagar de modo eficiente la información de rutas:



Capa de red 4-53

## direccionamiento jerarquizado: rutas más específicas

ISP Juanito tiene una ruta más específica hacia la Organización 1



Capa de red 4-54

## direccionamiento IP: la última palabra...

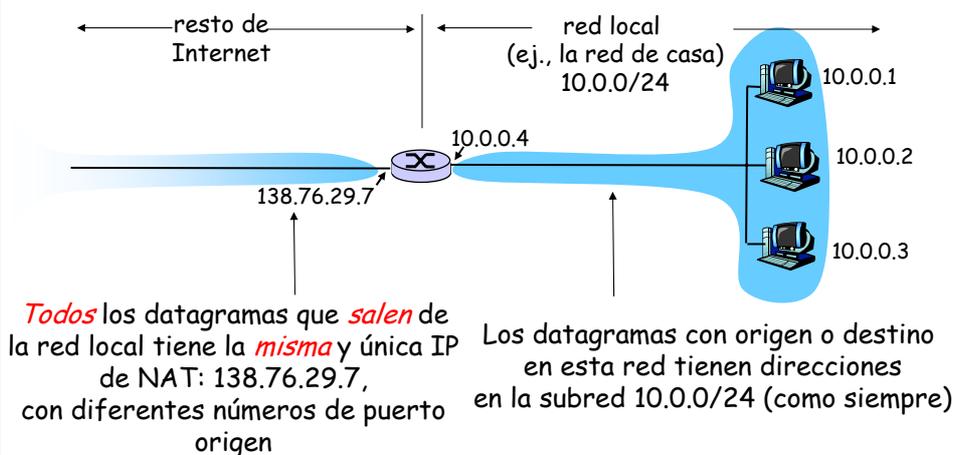
**P:** ¿De dónde saca un ISP su bloque de direcciones?

**R:** **ICANN:** Internet Corporation for Assigned Names and Numbers

- reserva direcciones
- gestiona DNS
- asigna nombres de dominios, resuelve disputas

Capa de red 4-55

## NAT: Network Address Translation



Capa de red 4-56

## NAT: Network Address Translation

- ❖ **Motivación:** la red local usa una sola IP desde el punto de vista del mundo exterior:
  - no se necesita un rango de direcciones (que nos daría el ISP): basta una IP para todos los sistemas
  - se pueden modificar las direcciones locales en cualquier momento, sin notificar a nadie de fuera
  - se puede cambiar de ISP sin tocar nada de la configuración de la red local
  - los sistemas dentro de la red local no son "visibles" desde fuera (no se pueden direccionar), lo que añade seguridad

Capa de red 4-57

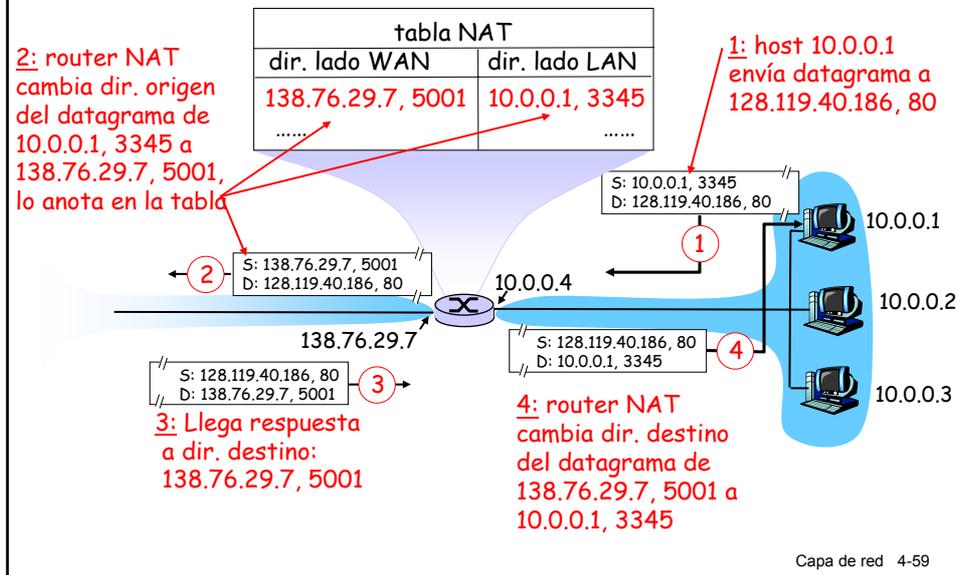
## NAT: Network Address Translation

**Implementación:** un router NAT debe:

- *datagramas salientes: reemplazar* (IP origen, n° de puerto) de cada datagrama saliente a (IP NAT, nuevo n° de puerto)
  - ... clientes/servidores remotos responderán usando (IP NAT, nuevo n° de puerto) como dirección destino
- *registrar (en la tabla NAT)* cada par (IP origen, n° de puerto) a (IP NAT, nuevo n° de puerto)
- *datagramas entrantes: reemplazar* (IP NAT, nuevo n° de puerto) en el campo destino de cada datagrama (IP origen, n° de puerto) almacenado en la tabla NAT

Capa de red 4-58

## NAT: Network Address Translation

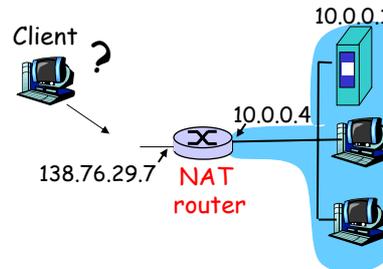


## NAT: Network Address Translation

- ❖ el campo n<sup>o</sup>-de-puerto es de 16 bits:
  - ¡60.000 conexiones simultáneas con una sola IP para toda la red!
- ❖ ...pero NAT tiene sus detractores:
  - los routers debieran procesar solo hasta la capa 3
  - viola el argumento "terminal-a-terminal"
    - la posibilidad de NAT ha de tenerse en cuenta al diseñar aplicaciones, por ejemplo, tipo P2P
  - el problema de falta de direcciones debe resolverse por la vía de usar IPv6

## problema para "atravesar" NAT

- ❖ un cliente quiere conectarse a un servidor con IP 10.0.0.1
  - la IP del servidor 10.0.0.1 es local a la red (el cliente no la puede usar como dir. destino)
  - la única IP visible externamente es la de NAT: 138.76.29.7
- ❖ solución 1: configurar NAT "a fuego" para reenviar el tráfico dirigido a cierto puerto hacia el servidor:
  - p. ej., paquetes (123.76.29.7, 2500) se reenvían siempre a (10.0.0.1, 25000)

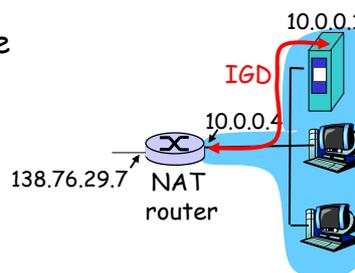


Capa de red 4-61

## problema para "atravesar" NAT

- ❖ solución 2: protocolo Universal Plug and Play (UPnP) Internet Gateway Device (IGD). Permite que un host de la red local:
  - ❖ conozca la IP pública (138.76.29.7)
  - ❖ añada/borre correspondencias (IP local, puerto) (con plazo de validez)

la idea es automatizar la configuración NAT puramente estática



Capa de red 4-62

## problema para "atravesar" NAT

- ❖ solución 3: con intermediarios (usado en Skype)
  - el cliente en red local establece conexión con el intermediario
  - el cliente externo se conecta también al intermediario
  - el intermediario que "media" entre ambas conexiones



Capa de red 4-63

## Capítulo 4: la capa de red

- 4.1 Introducción
- 4.2 Redes de circuitos virtuales y de datagramas
- 4.3 Las tripas de un router
- 4.4 IP: Internet Protocol**
  - Formato de datagramas
  - Direccionamiento IPv4
  - **ICMP**
  - IPv6
- 4.5 Algoritmos de enrutamiento
  - Estado de enlaces
  - Vector de distancias
  - Enrutamiento jerárquico
- 4.6 Enrutamiento en Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Enrutamiento por difusión y multidifusión

Capa de red 4-64

## ICMP: Internet Control Message Protocol

❖ usado por hosts & routers para comunicar info a nivel de red	<u>Tipo</u>	<u>Código</u>	<u>Descripción</u>
	0	0	respuesta de eco
▪ informe de errores: host, red, puerto, protocolo inalcanzables	3	0	red dest. inalcanzable
	3	1	host dest. inalcanzable
▪ solicitud/respuesta de eco (usada para "ping")	3	2	protocolo dest. inalcanzable
	3	3	puerto dest. inalcanzable
❖ protocolo "por encima" de IP:	3	6	red dest. desconocida
	3	7	host dest. desconocido
▪ los mens. ICMP se transportan en datagramas IP (como si fuera TCP, o UDP)	4	0	regulación en origen (control congestión - no usado)
	8	0	solicitud de eco (ping)
❖ <b>mensaje ICMP:</b> tipo, código, más primeros 8 bytes del datagrama IP que causó el error	9	0	anuncio de ruta
	10	0	descubrimiento de ruta
	11	0	TTL caducado
	12	0	cabecera IP errónea

Capa de red 4-65

## Traceroute e ICMP

- ❖ El origen manda una serie de segmentos UDP al destino
    - el primero con TTL =1
    - el segundo, TTL =2, etc.
    - n° de puerto "absurdo"
  - ❖ Cuando el enésimo datagrama llega al router:
    - el router lo descarta...
    - ...y envía al origen un mensaje ICMP (tipo 11, código 0)
    - el mensaje ICMP incluye nombre e IP del router
  - ❖ cuando llega el mensaje ICMP, el origen calcula RTT
  - ❖ traceroute repite esto tres veces
- criterio de parada
- ❖ el segmento UDP llega por fin al host destino
  - ❖ el destino devuelve el mensaje ICMP "puerto inalcanzable" (tipo 3, código 3)
  - ❖ cuando el origen detecta este mensaje ICMP, se para.

Capa de red 4-66

## Capítulo 4: la capa de red

### 4.1 Introducción

### 4.2 Redes de circuitos virtuales y de datagramas

### 4.3 Las tripas de un router

### 4.4 IP: Internet Protocol

- Formato de datagramas
- Direccionamiento IPv4
- ICMP
- IPv6

### 4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector de distancias
- Enrutamiento jerárquico

### 4.6 Enrutamiento en Internet

- RIP
- OSPF
- BGP

### 4.7 Enrutamiento por difusión y multidifusión

Capa de red 4-67

## IPv6

❖ **Motivación inicial:** las direcciones de 32 bits se van a acabar pronto

❖ **Motivación adicional:**

- el nuevo formato de cabecera ayuda a acelerar el procesamiento/reenvío
- también facilita la QoS

**Formato del datagrama:**

- cabecera de longitud fijada a 40 bytes
- no se permite fragmentación

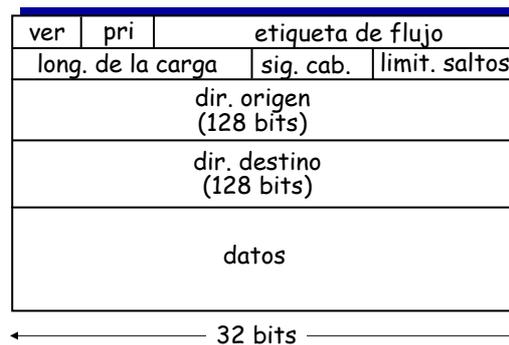
Capa de red 4-68

## Cabecera IPv6 (cont)

**Prioridad:** identifica prioridad entre los datagramas de de un flujo

**Etq. de "flujo":** identifica datagramas en el mismo "flujo" (concepto de "flujo" vagamente definido)

**Cabecera siguiente:** identifica prot. de capa superior



Capa de red 4-69

## Otros cambios desde IPv4

- ❖ **Suma de comprobación:** se elimina, para reducir tiempo de proceso por salto
- ❖ **Opciones:** permitidas, pero fuera de la cabecera, indicadas por el campo "siguiente cabecera"
- ❖ **ICMPv6:** nueva versión de ICMP
  - tipos de mensajes adicionales, por ejemplo, "Paquete demasiado grande"
  - funciones para gestión de grupos multidifusión

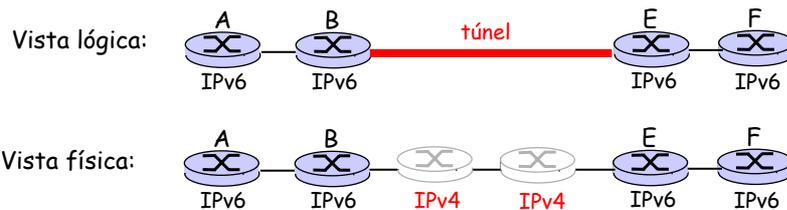
Capa de red 4-70

## Transición de IPv4 a IPv6

- ❖ No se puede cambiar todos los routers a la vez
  - no hay "día d"
  - mejor idea: cómo hacer que la red opere con una mezcla de routers IPv4 e IPv6
- ❖ **"Tunelar"**: IPv6 transportado como carga en un datagrama IPv4 estándar entre los routers IPv4

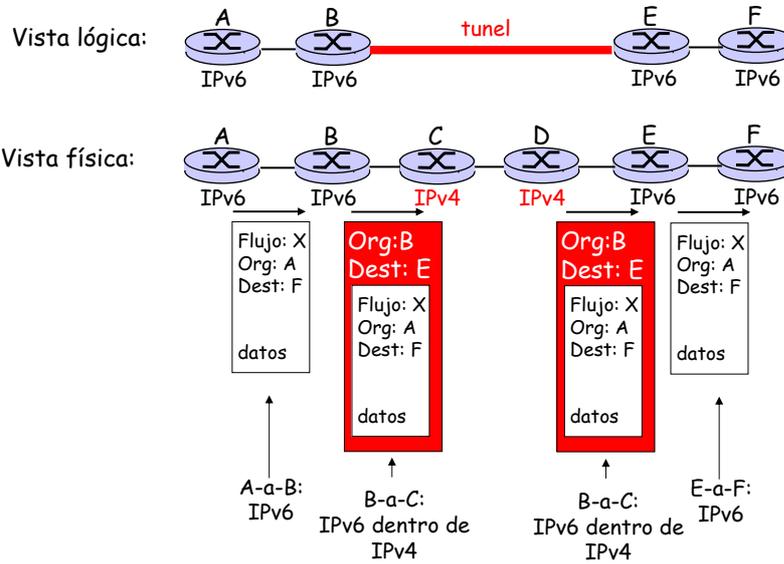
Capa de red 4-71

## Tunelar



Capa de red 4-72

# Tunelar



Capa de red 4-73

# Capítulo 4: la capa de red

## 4.1 Introducción

## 4.2 Redes de circuitos virtuales y de datagramas

## 4.3 Las tripas de un router

## 4.4 IP: Internet Protocol

- Formato de datagramas
- Direccionamiento IPv4
- ICMP
- IPv6

## 4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector de distancias
- Enrutamiento jerárquico

## 4.6 Enrutamiento en Internet

- RIP
- OSPF
- BGP

## 4.7 Enrutamiento por difusión y multidifusión

Capa de red 4-74

## Capítulo 4: la capa de red

### 4.1 Introducción

### 4.2 Redes de circuitos virtuales y de datagramas

### 4.3 Las tripas de un router

### 4.4 IP: Internet Protocol

- Formato de datagramas
- Direccionamiento IPv4
- ICMP
- IPv6

### 4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector de distancias
- Enrutamiento jerárquico

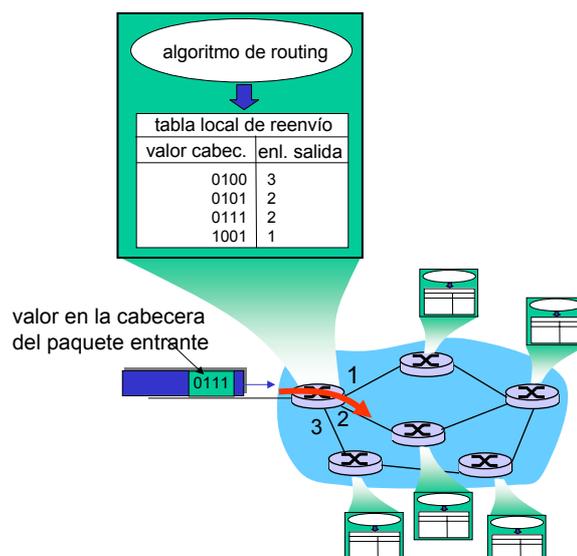
### 4.6 Enrutamiento en Internet

- RIP
- OSPF
- BGP

### 4.7 Enrutamiento por difusión y multidifusión

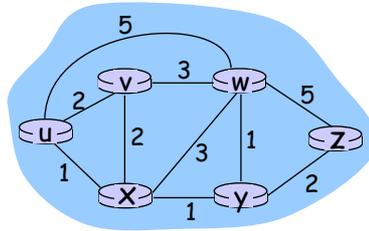
Capa de red 4-75

## Interacción routing-reenvío



Capa de red 4-76

## Modelo de grafo



Grafo:  $G = (N, E)$

$N$  = conjunto de routers =  $\{ u, v, w, x, y, z \}$

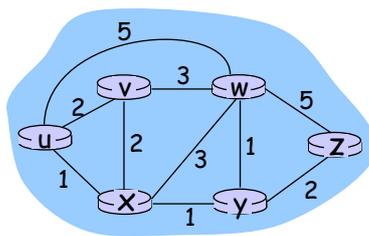
$E$  = conjunto de enlaces =  $\{ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) \}$

Observación: modelar mediante grafos es útil en otros contextos

Ejemplo: P2P, donde  $N$  es el set de peers y  $E$  es el set de conexiones TCP

Capa de red 4-77

## Modelo de grafo: costes



•  $c(x,x')$  = coste de enlace  $(x,x')$

- p.ej.,  $c(w,z) = 5$

• el coste puede ser siempre 1, o inversamente prop. al ancho de banda, o inversamente prop. a la congestión

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

Pregunta: ¿Cuál es la ruta más barata entre  $u$  y  $z$  ?

Algoritmo de routing: el que encuentra la ruta más barata

Capa de red 4-78

## Clasificación de los algoritmos de routing

### ¿Información global o descentralizada?

#### Global:

- ❖ todos los routers tienen completa info sobre la topología y costes de enlace

- ❖ algoritmos "estado del enlace"

#### Descentralizada:

- ❖ el router conoce a los vecinos físicamente conectados y el coste de los enlaces a ellos
- ❖ proceso iterativo de computación e intercambio de info con los vecinos
- ❖ algoritmos "vector distancia"

### ¿Estática o dinámica?

#### Estática:

- ❖ las rutas cambian poco con el tiempo

#### Dinámica:

- ❖ las rutas cambian rápidamente
  - actualización periódica
  - en respuesta a cambios en el coste de enlaces

## Capítulo 4: la capa de red

### 4.1 Introducción

### 4.2 Redes de circuitos virtuales y de datagramas

### 4.3 Las tripas de un router

### 4.4 IP: Internet Protocol

- Formato de datagramas
- Direccionamiento IPv4
- ICMP
- IPv6

### 4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector de distancias
- Enrutamiento jerárquico

### 4.6 Enrutamiento en Internet

- RIP
- OSPF
- BGP

### 4.7 Enrutamiento por difusión y multidifusión

## Algoritmo estado de enlaces

### algoritmo de Dijkstra

- ❖ todos los nodos conocen la topología y costes de enlace
  - vía "difusión del estado del enlace"
  - todos los nodos, misma info
- ❖ computa rutas de mínimo coste desde un nodo origen a todos los demás
  - da la *tabla de reenvío* para ese nodo
- ❖ iterativo: tras  $k$  iteraciones, se conocen las rutas de coste mínimo a  $k$  destinos.

### Notación:

- ❖  $c(x,y)$ : coste de enlace del nodo  $x$  al  $y$ ;  $= \infty$  si no son vecinos inmediatos
- ❖  $D(v)$ : valor actual del coste de la ruta desde el origen al destino  $v$
- ❖  $p(v)$ : predecesor en la ruta desde el origen a  $v$
- ❖  $N'$ : conjunto de nodos cuya ruta de mínimo coste está ya clara

## Algoritmo de Dijkstra

### 1 Inicialización:

- 2  $N' = \{u\}$
- 3 para todo nodo  $v$
- 4 si  $v$  adyacente a  $u$
- 5 entonces  $D(v) = c(u,v)$
- 6 si no  $D(v) = \infty$
- 7

### 8 Lazo

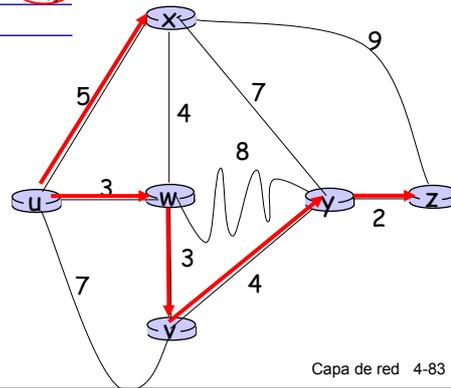
- 9 buscar  $w$  no en  $N'$  tal que  $D(w)$  sea el mínimo
- 10 añadir  $w$  a  $N'$
- 11 para todo  $v$  adyacente a  $w$  y no en  $N'$ , actualiza  $D(v)$ :
- 12  $D(v) = \min( D(v), D(w) + c(w,v) )$
- 13 /\* el nuevo coste a  $v$  es: o viejo coste a  $v$ , o coste de ruta
- 14 más barata conocida a  $w$  más coste de  $w$  a  $v$  \*/
- 15 **hasta que todos los nodos están en  $N'$**

## Algoritmo de Dijkstra: ejemplo

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

### Notas:

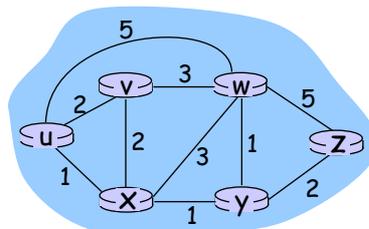
- ❖ construir árbol de camino más corto rastreando los predecesores
- ❖ si hay algún empate, se desempata eligiendo arbitrariamente



Capa de red 4-83

## 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	$\infty$	$\infty$
1	ux	2,u	4,x		2,x	$\infty$
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



Capa de red 4-84

## Algoritmo de Dijkstra: ejemplo (2)

Camino más corto resultante desde u:

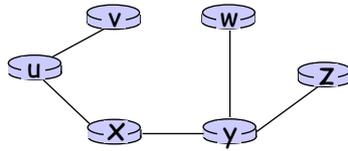


Tabla de reenvío resultante en u:

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

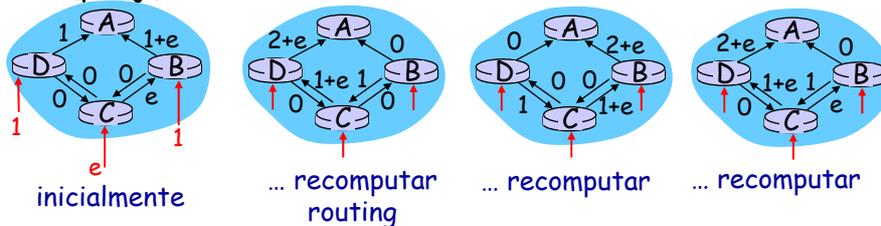
## Algoritmo de Dijkstra: discusión

**Complejidad del algoritmo:**  $n$  nodos

- ❖ cada iteración: ver qué nodos,  $w$ , no están en  $N$
- ❖  $n(n+1)/2$  comparaciones:  $O(n^2)$
- ❖ otras implementaciones más eficientes:  $O(n \cdot \log n)$

**Posibles oscilaciones:**

- ❖ coste de enlace = cantidad de tráfico transportado (p. ej.)



## Capítulo 4: la capa de red

### 4.1 Introducción

### 4.2 Redes de circuitos virtuales y de datagramas

### 4.3 Las tripas de un router

### 4.4 IP: Internet Protocol

- Formato de datagramas
- Direccionamiento IPv4
- ICMP
- IPv6

### 4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector de distancias
- Enrutamiento jerárquico

### 4.6 Enrutamiento en Internet

- RIP
- OSPF
- BGP

### 4.7 Enrutamiento por difusión y multidifusión

Capa de red 4-87

## Algoritmo vector de distancias

### Ecuación Bellman-Ford (programación dinámica)

Definir

$d_x(y) :=$  coste del camino más barato desde  $x$  a  $y$

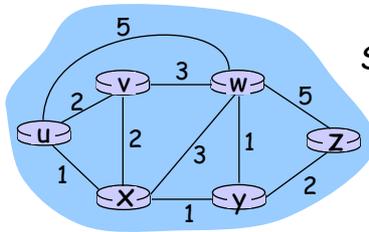
Tenemos

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

donde el mín se toma sobre todos los vecinos  $v$  de  $x$

Capa de red 4-88

## Ejemplo de Bellman-Ford



Se ve que  $d_v(z) = 5$ ,  $d_x(z) = 3$ ,  $d_w(z) = 3$

Mediante B-F:

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

El nodo que alcanza el mínimo se convierte en el siguiente salto del camino más barato -> tabla de reenvío

## Algoritmo vector de distancias

- ❖  $D_x(y)$  = estimación del mínimo coste desde  $x$  a  $y$ 
  - $x$  mantiene el vector distancia  $D_x = [D_x(y): y \in N]$
- ❖ nodo  $x$ :
  - sabe el coste hacia cada vecino  $v$ :  $c(x,v)$
  - mantiene los vectores distancia de sus vecinos. Para cada vecino  $v$ ,  $x$  mantiene  $D_v = [D_v(y): y \in N]$

## Algoritmo vector de distancias (4)

### Idea básica:

- ❖ de vez en cuando, cada nodo manda la estimación de su propio vector de distancias a sus vecinos
- ❖ cuando  $x$  recibe una nueva estimación del DV, actualiza su propio DV mediante la ecuación B-F:

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

- ❖ en condiciones normales, la estimación  $D_x(y)$  converge hacia el coste mínimo real  $d_x(y)$

## Algoritmo vector de distancias (5)

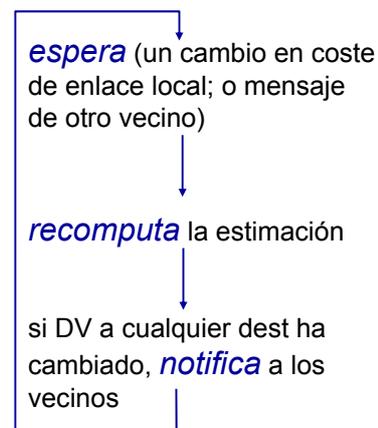
**Iterativo, asíncrono:** una iteración local viene causada por:

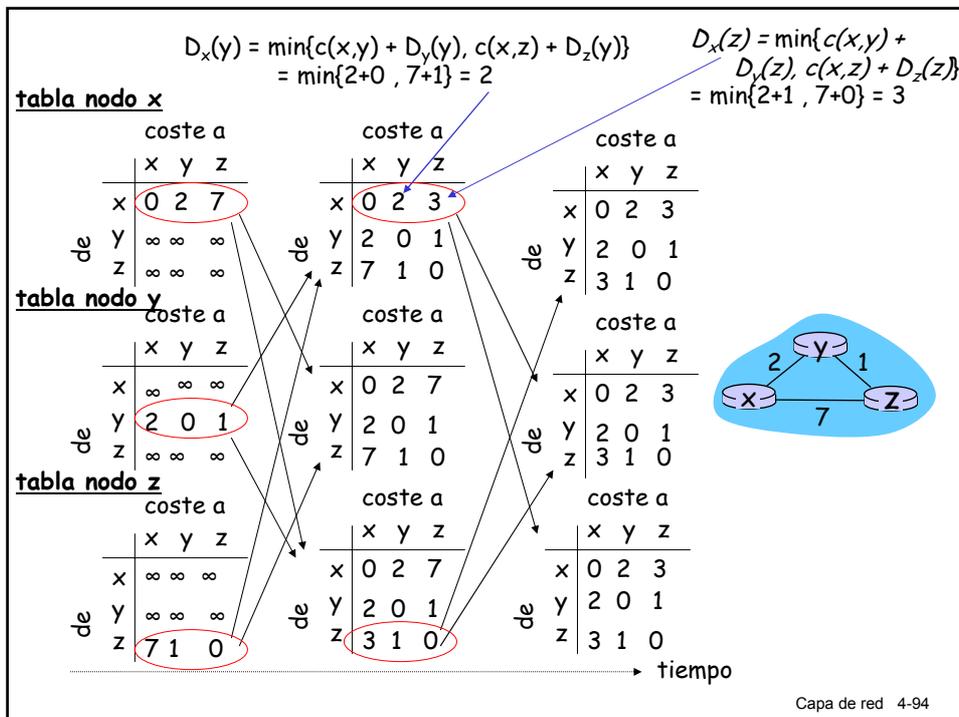
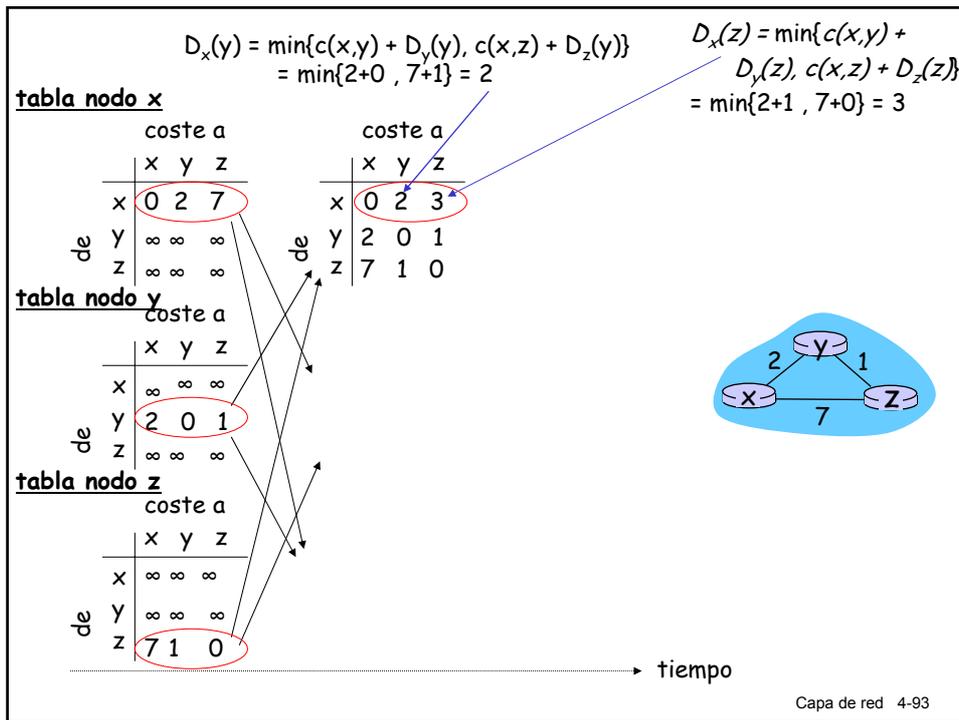
- ❖ cambio de coste en el enlace local
- ❖ mensaje de actualización de DV desde un vecino

### **Distribuido:**

- ❖ cada nodo notifica a los vecinos *solo* si su propio DV cambia:
  - y los vecinos a sus vecinos, si es necesario

**Cada nodo:**

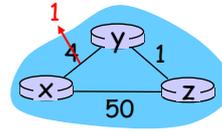




## Vector Distancia: cambios en el coste del enlace

### Cambios en el coste del enlace:

- ❖ el nodo detecta cambios en el coste de un enlace
- ❖ actualiza info de routing, recalcula el vector de distancias
- ❖ si el DV cambia, notifica a los vecinos



"las buenas noticias viajan deprisa"

$t_0$ : y detecta cambio en el coste del enlace, actualiza su DV, informa a los vecinos.

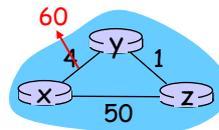
$t_1$ : z recibe actualización de y, actualiza su tabla, computa nuevo coste mínimo a x, envía a sus vecinos su DV.

$t_2$ : y recibe actualización de z, actualiza su tabla de distancias. La ruta de coste mínimo *no cambia*, así que y *no envía* mensaje de vuelta a z.

## Vector Distancia: cambios en el coste del enlace

### Cambios en el coste del enlace :

- ❖ las buenas noticias viajan rápido
- ❖ las malas, despacio - ¡problema de la "cuenta hasta infinito"!
- ❖ 44 iteraciones hasta que se estabiliza: iuf!



### Inversa envenenada:

- ❖ Si Z enruta X a través de Y:
  - Z le dice a Y que su distancia (la de Z) a X es infinito (para que Y no enrute a X vía Z)
- ❖ ¿resuelve esto completamente el problema de la "cuenta hasta infinito"?

## Comparación de los algoritmos LS y DV

### Complejidad del mensaje

- ❖ **LS:** con  $n$  nodos,  $E$  enlaces, se envían  $O(nE)$  mensajes
- ❖ **DV:** intercambio solo entre vecinos
  - tiempo de convergencia variable

### Velocidad de Convergencia

- ❖ **LS:** algoritmo  $O(n^2)$ , necesita  $O(nE)$  mensajes
  - puede oscilar
- ❖ **DV:** tiempo de convergencia variable
  - bucles de enrutado posbl.
  - prob. "cuenta hasta  $\infty$ "

### Robustez: ¿qué pasa si falla un router?

#### **LS:**

- el nodo puede difundir un coste de *enlace* erróneo
- cada nodo computa solo *su propia tabla*

#### **DV:**

- el nodo DV puede difundir un coste de *ruta* erróneo
- la tabla de cada nodo la usan otros...
  - el error se propaga por la red

Capa de red 4-97

## Capítulo 4: la capa de red

### 4.1 Introducción

### 4.2 Redes de circuitos virtuales y de datagramas

### 4.3 Las tripas de un router

### 4.4 IP: Internet Protocol

- Formato de datagramas
- Direccionamiento IPv4
- ICMP
- IPv6

### 4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector de distancias
- **Enrutamiento jerárquico**

### 4.6 Enrutamiento en Internet

- RIP
- OSPF
- BGP

### 4.7 Enrutamiento por difusión y multidifusión

Capa de red 4-98

## Enrutamiento jerárquico

Nuestro estudio, hasta ahora, era ideal:

- ❖ todos los routers idénticos,
  - ❖ red "plana",
- ... *ino es válido en la práctica!*

**escala:** con 200 millones de destinos:

- ❖ no se pueden meter todos en las tablas de routing!
- ❖ con solo los intercambios de tablas se saturarían los enlaces!

**autonomía administrativa**

- ❖ internet = red de redes
- ❖ cada administrador de red querrá controlar su propia porción de la red

## Enrutamiento jerárquico

- ❖ agregación de routers en regiones, llamadas "sistemas autónomos" (AS)
- ❖ los routers en el mismo AS ejecutan el mismo protocolo de enrutamiento
  - hay un enrutamiento "intra-AS"...
  - mientras routers en diferentes AS pueden ejecutar diferentes protocolos intra-AS

router pasarela (gateway)

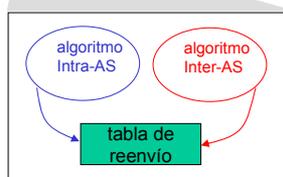
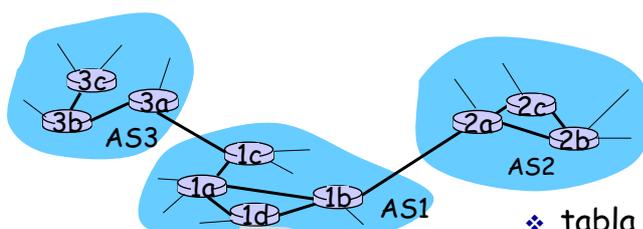
- ❖ en la "frontera" de su propio AS
- ❖ enlaza con el router pasarela de otro AS

## Tipos de ASes

- ❖ **AS terminal**
  - tiene una sola conexión con el mundo exterior.
  - el tráfico que soporta es solo el que se origina o tiene como destino ese AS.
- ❖ **AS terminal con múltiples conexiones**
  - igual al anterior, pero se conecta al mundo exterior a través de más de una conexión.
- ❖ **AS de tránsito**
  - tiene varias conexiones con el mundo exterior.
  - permite transportar tráfico en tránsito, es decir, que no se origina ni tiene como destino ese AS.

Capa de red 4-101

## ASes interconectados



- ❖ tabla de reenvío configurada tanto con el algoritmo intra-AS como el inter-AS
  - entradas intra-AS para destinos internos...
  - entradas inter-AS e intra-AS para destinos externos

Capa de red 4-102

## Tareas Inter-AS

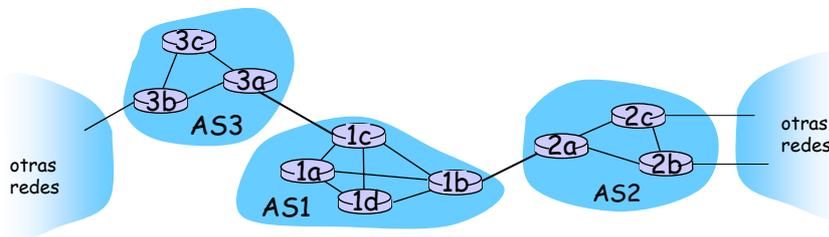
- ❖ si un router en AS1 recibe un datagrama destinado al exterior de AS1:

- debe reenviarlo al router pasarela, pero... ¿a cuál?

### AS1 debe:

1. conocer qué destinos son alcanzables vía AS2 y cuáles vía AS3
2. propagar esta info a *todos* los routers en AS1

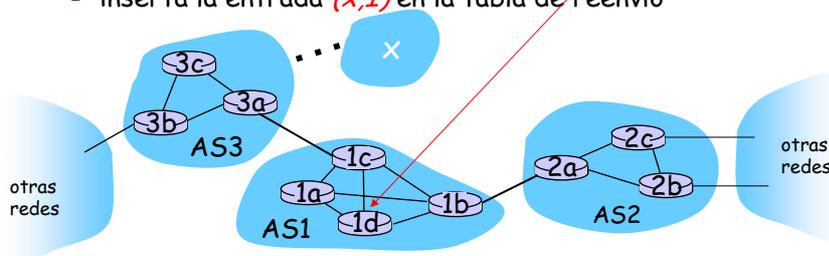
**una tarea para el routing inter-AS!**



Capa de red 4-103

## Ejemplo: generar tabla de reenvío en router 1d

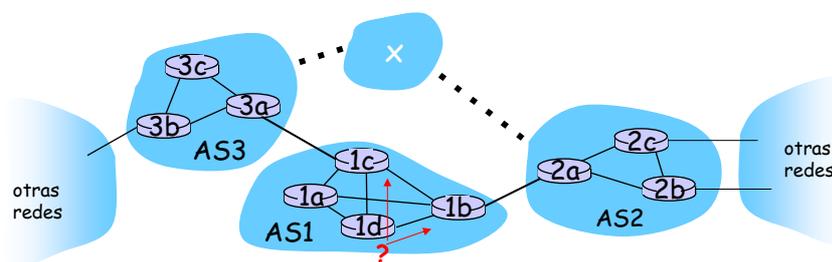
- ❖ supongamos que AS1 aprende (vía protocolo inter-AS) que la subred  $x$  es alcanzable vía AS3 (pasarela 1c) pero no vía AS2.
  - el protocolo inter-AS propaga esta info a todos los routers internos
- ❖ el router 1d determina a partir de la info del routing intra-AS que su interfaz  $I$  está sobre la ruta de mínimo coste a 1c.
  - inserta la entrada  $(x, I)$  en la tabla de reenvío



Capa de red 4-104

## Ejemplo: elección entre múltiples ASes

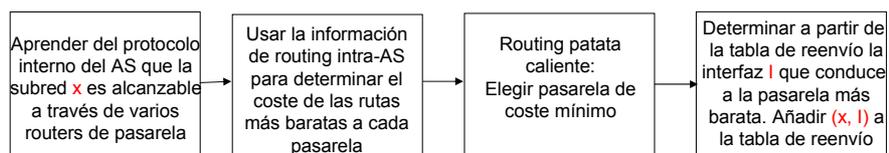
- ❖ supongamos ahora que AS1 aprende vía protocolo inter-AS que la subred **x** es alcanzable vía AS3 y vía AS2.
- ❖ para configurar su tabla de reenvío, el router 1d debe determinar a cuál pasarela debe enviar los paquetes con destino a **x**
  - itarea también para el protocolo inter-AS!



Capa de red 4-105

## Ejemplo: elección entre múltiples ASes

- ❖ supongamos ahora que AS1 aprende vía protocolo inter-AS que la subred **x** es alcanzable vía AS3 y vía AS2.
- ❖ para configurar su tabla de reenvío, el router 1d debe determinar a cuál pasarela debe enviar los paquetes con destino a **x**.
  - itarea también para el protocolo inter-AS!
- ❖ **enrutamiento de la patata caliente:** enviar el paquete al router más cercano.



Capa de red 4-106

## Capítulo 4: la capa de red

### 4.1 Introducción

### 4.2 Redes de circuitos virtuales y de datagramas

### 4.3 Las tripas de un router

### 4.4 IP: Internet Protocol

- Formato de datagramas
- Direccionamiento IPv4
- ICMP
- IPv6

### 4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector de distancias
- Enrutamiento jerárquico

### 4.6 Enrutamiento en Internet

- RIP
- OSPF
- BGP

### 4.7 Enrutamiento por difusión y multidifusión

Capa de red 4-107

## Enrutamiento Intra-AS

❖ conocidos como protocolos de pasarela interior  
**Interior Gateway Protocols (IGP)**

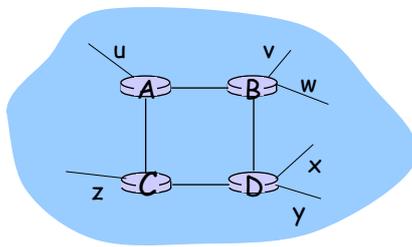
❖ protocolos Intra-AS más típicos:

- RIP: Routing Information Protocol (protocolo de información de enrutamiento)
- OSPF: Open Shortest Path First (en primer lugar, la ruta más corta).
  - "Open": se refiere a que está a disposición pública.
- IGRP: Interior Gateway Routing Protocol (propiedad de Cisco)

Capa de red 4-108

## RIP ( Routing Information Protocol)

- ❖ incluido en la distribución BSD-UNIX en 1982
- ❖ algoritmo vector de distancias
  - métrica de distancia : n° saltos (max = 15), cada enlace cuesta 1
  - DVs intercambiados con los vecinos cada 30 s en un mensaje de respuesta RIP (llamado *anuncio RIP*)
  - en cada anuncio: lista de hasta 25 *subredes* (en sentido IP) destino



del router A hacia *subredes* destino

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

Capa de red 4-109

## RIP: Ejemplo

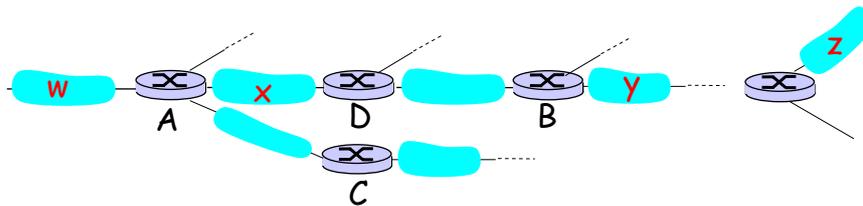


tabla de reenvío en router D

subred destino	siguiente router	n° saltos a dest
w	A	2
y	B	2
x	--	1
....	....	....

Capa de red 4-110

## RIP: Ejemplo

D recibe un anuncio de A

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

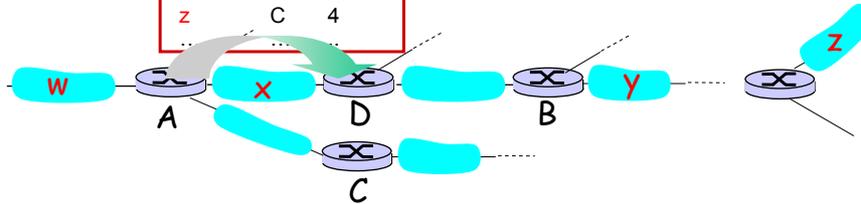


tabla de reenvío en router D

subred dest	siguiente router	nº saltos a dest
w	A	2
y	B	2
z	<del>B</del> A	<del>7</del> 5
x	--	1
....	....	....

Capa de red 4-111

## RIP: Fallo y recuperación de enlace

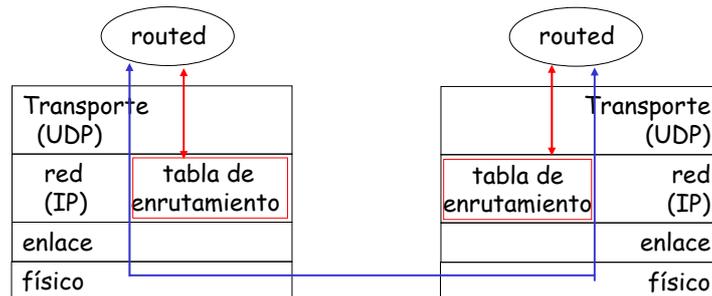
Si tras 180 s no recibo anuncio de mi vecino --> entiendo que el vecino o el enlace "han muerto"

- invalido rutas a través del "muerto"
- envío anuncios a los demás vecinos
- los vecinos por su parte envían más anuncios (si sus tablas cambian)
- la info sobre un enlace caído se propaga rápidamente (?) a toda la red
- *inversa envenenada*: se usa para evitar bucles ping-pong (se considera distancia "infinita" = 16 saltos)

Capa de red 4-112

## Procesamiento de tablas de enrutado RIP

- ❖ Las tablas de enrutado RIP se manejan a nivel de **aplicación** por el daemon *routed*
- ❖ los anuncios se mandan en paquetes UDP, periódicamente repetidos



Capa de red 4-113

## OSPF (Open Shortest Path First)

- ❖ "abierto": disponible públicamente
- ❖ usa algoritmo de *estado de enlace* (LS)
  - "inundación" de paquetes LS
  - cada nodo arma el "mapa topológico" completo
  - se computan las rutas mediante el algoritmo de Dijkstra
- ❖ el anuncio OSPF lleva una entrada por cada vecino
- ❖ los anuncios se difunden **a todos los demás** routers del AS (vía "inundación")
  - los mensajes OSPF se transportan directamente sobre IP (no sobre TCP o UDP)
- ❖ anuncios periódicos (haya o no cambios)

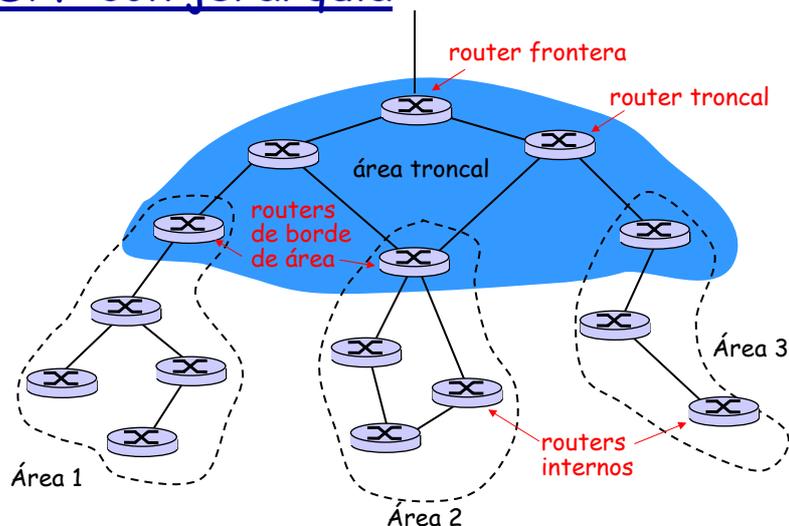
Capa de red 4-114

## Funcionalidades "avanzadas" de OSPF (que no estaban en RIP)

- ❖ **seguridad**: todos los mensajes OSPF van autenticados (para evitar intrusiones maliciosas)
- ❖ se permiten **múltiples rutas** con el **mismo coste** (sólo se permitía una en RIP)
- ❖ en cada enlace, múltiples métricas de coste, en función del tipo de servicio (ToS).
  - Ejemplos: enlace vía satélite, coste bajo; ToS de tiempo real, coste alto
- ❖ soporte integrado uni- y multi-difusión:
  - Multidifusión OSPF (MOSPF): usa la misma "base de datos topológica" que OSPF
- ❖ OSPF **jerárquico** en dominios grandes.

Capa de red 4-115

## OSPF con jerarquía



Capa de red 4-116

## OSPF con jerarquía

- ❖ **jerarquía a dos niveles:** área local y troncal.
  - anuncios de estado de enlace solo dentro del área
  - cada nodo tiene el detalle de la topología de área; solo sabe el camino más corto a redes de otras áreas.
- ❖ ***routers de borde de área:*** "resumen" las distancias a redes dentro de su propia área, mandan anuncios a otros routers de borde de área.
- ❖ ***routers troncales:*** ejecutan el enrutado OSPF dentro de los límites del área troncal.
- ❖ ***routers de frontera:*** conectan a otros ASs.

Capa de red 4-117

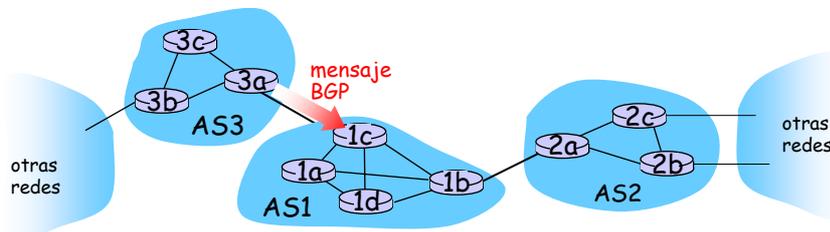
## Enrutado Internet inter-AS: BGP

- ❖ **BGP (Border Gateway Protocol):** *el protocolo* por antonomasia para el enrutamiento inter-dominios
  - "el engrudo que mantiene Internet bien cohesionada"
- ❖ BGP ofrece a cada AS un medio de:
  - **eBGP:** obtiene información de alcanzabilidad enviada por los ASs.
  - **iBGP:** propaga la información de alcanzabilidad a todos los routers internos al AS.
  - determina las rutas "buenas" a otras redes, según la información de alcanzabilidad y las políticas del AS.
- ❖ permite que la subred anuncie su existencia al resto de Internet: *"¡Eh! ¡Que estoy aquí!"*

Capa de red 4-118

## Fundamentos de BGP

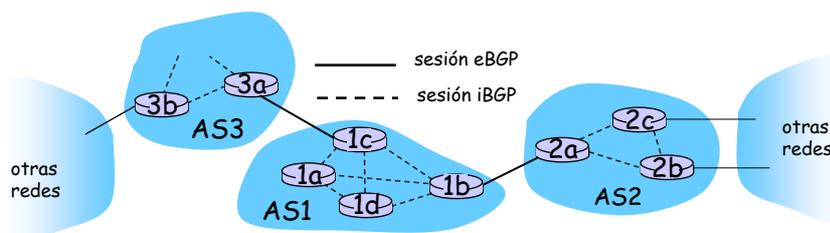
- ❖ **Sesión BGP:** dos routers BGP ("pares") intercambian mensajes:
  - anunciando rutas a diferentes subredes (= prefijo de red) destino (protocolo "vector de ruta")
  - intercambiados mediante conexiones TCP semi-permanentes
- ❖ cuando AS3 anuncia un prefijo a AS1:
  - AS3 *promete* que reenviará los datagramas hacia ese prefijo
  - AS3 puede agregar prefijos en su anuncio



Capa de red 4-119

## BGP: distribución de info de rutas

- ❖ usando una sesión eBGP entre 3a y 1c, AS3 le envía info de alcanzabilidad a AS1.
  - 1c puede entonces usar iBGP para distribuir la nueva info de prefijo a todos los routers en AS1
  - 1b puede a su vez re-anunciar la nueva info de alcanzabilidad a AS2 sobre la sesión eBGP 1b-2a
- ❖ cuando un router se entera del nuevo prefijo, crea un entrada para él en su tabla de reenvío.



Capa de red 4-120

## Atributos de ruta & rutas BGP

- ❖ los prefijos anunciados incluyen atributos BGP
  - prefijo + atributos = "ruta"
- ❖ dos atributos importantes:
  - **AS\_PATH**: contiene la lista de ASs atravesados en su camino por el anuncio de prefijo: ej., AS 67, AS 17
  - **NEXT\_HOP**: indica el router interno del AS por el que se llega al siguiente AS. (Pueden necesitarse varios enlaces para llegar desde el AS actual al siguiente)
- ❖ el router pasarela que recibe el anuncio de ruta usa una **política de importación** para aceptar/declinar el anuncio
  - ej., "nunca enrutar a través del AS fulano"
  - enrutado *basado en políticas*

Capa de red 4-121

## Selección de la ruta BGP

- ❖ si el router se entera de que hay más de una ruta para llegar al destino AS, selecciona una basándose en:
  1. el valor de preferencia local, asignado vía atributo: es una decisión sobre la política
  2. AS\_PATH más corto
  3. router NEXT\_HOP más próximo: enrutamiento "patata caliente"
  4. otros criterios

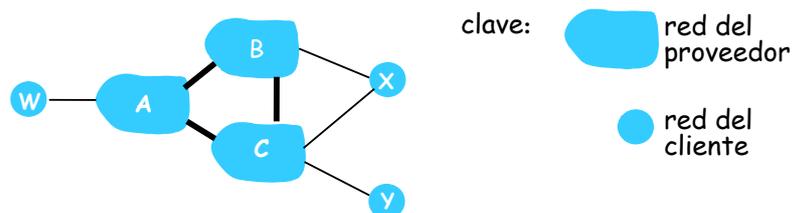
Capa de red 4-122

## Mensajes BGP

- ❖ Los "pares" intercambian mensajes BGP sobre conexiones TCP
- ❖ tipos de mensajes BGP:
  - **OPEN**: abre una conexión TCP con un par y autentifica al remitente
  - **UPDATE**: anuncia nueva ruta (o retira una vieja)
  - **KEEPALIVE**: mantiene "viva" la conexión, incluso en ausencia de UPDATES; también si hay peticiones de tipo ACKs OPEN
  - **NOTIFICATION**: informa de errores en mensajes previos; también se usa para cerrar la conexión

Capa de red 4-123

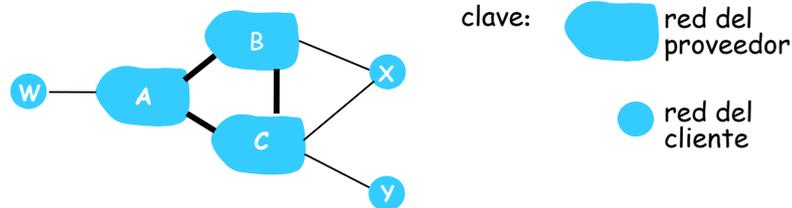
## Política de enrutamiento BGP



- ❖ A,B,C son **redes del proveedor**
- ❖ X,W,Y son clientes (de las redes del proveedor)
- ❖ X está **multi-conectado**: enganchado a dos redes
  - ¡X no quiere enrutar paquetes de B a C vía su propia red!
  - ...así que X no anunciará a B que hay una ruta a C

Capa de red 4-124

## Política de enrutamiento BGP (2)



- ❖ A anuncia ruta AW a B
- ❖ B anuncia ruta BAW a X
- ❖ ¿Debe B anunciar la ruta BAW a C?
  - ¡Para nada! B no "gana" nada enrutando CBAW, pues ni W ni C son clientes suyos (de B)
  - B quiere forzar a C a enrutar hacia W vía A
  - ¡Es normal! B quiere enrutar (= dar servicio) *solo* a sus clientes

Capa de red 4-125

## ¿Por qué diferentes enrutados,

### Intra- e Inter-AS?

#### Políticas:

- ❖ Inter-AS: el administrador quiere control sobre cómo enrutar su tráfico, y quién enruta a través de su red.
- ❖ Intra-AS: administrador único, no hay que negociar...

#### Escala:

- ❖ el enrutado jerárquico reduce la tabla de reenvío y el tráfico de "actualizaciones"

#### Rendimiento:

- ❖ Intra-AS: focalizado en rendimiento
- ❖ Inter-AS: las políticas pueden resultar más importantes que el rendimiento

Capa de red 4-126

## Capítulo 4: la capa de red

### 4.1 Introducción

### 4.2 Redes de circuitos virtuales y de datagramas

### 4.3 Las tripas de un router

### 4.4 IP: Internet Protocol

- Formato de datagramas
- Direccionamiento IPv4
- ICMP
- IPv6

### 4.5 Algoritmos de enrutamiento

- Estado de enlaces
- Vector de distancias
- Enrutamiento jerárquico

### 4.6 Enrutamiento en Internet

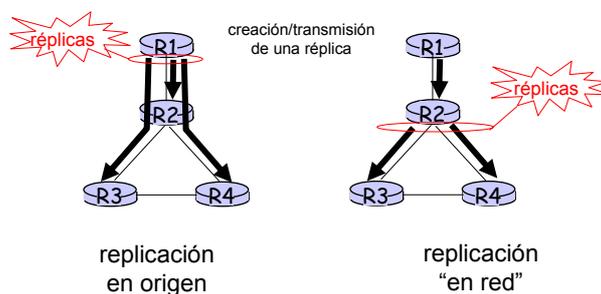
- RIP
- OSPF
- BGP

### 4.7 Enrutamiento por difusión y multidifusión

Capa de red 4-127

## Enrutamiento por difusión

- ❖ difusión: mandar paquetes de un origen a todos los demás nodos
- ❖ replicar desde el origen no es eficiente:



- ❖ replicación en origen: ¿cómo determina el origen las direcciones de los receptores?

Capa de red 4-128

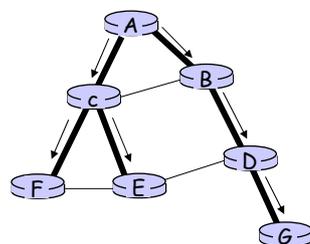
## Replicación en red

- ❖ **inundación:** cuando un nodo recibe un paquete de difusión, envía copias a todos sus vecinos
  - problemas: ciclos y "tormentas" de difusión
- ❖ **inundación controlada:** el nodo sólo difunde los paquetes de difusión si no ha difundido el mismo paquete previamente
  - el nodo guarda memoria de la identidad de los paquetes ya difundidos
  - o bien: reenvío por el camino inverso *reverse path forwarding* (RPF): sólo se reenvía un paquete si ha llegado por el camino más corto desde su origen
- ❖ **árbol de recubrimiento:**
  - ningún nodo recibe paquetes redundantes

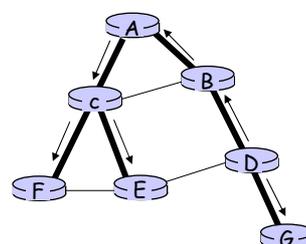
Capa de red 4-129

## Árbol de recubrimiento

- ❖ Primero, contruir árbol de recubrimiento
- ❖ Los nodos mandan copias solo a lo largo del árbol



(a) Difusión iniciada en A

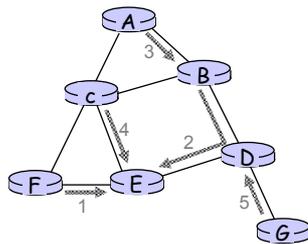


(b) Difusión iniciada en D

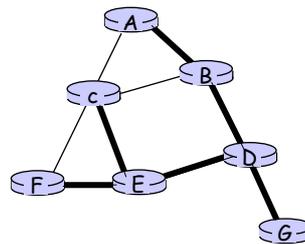
Capa de red 4-130

## Árbol de recubrimiento: creación

- ❖ método basado en *nodo central*
- ❖ cada nodo envía un mensaje de "adhesión" al *nodo central* vía unidifusión
  - el mensaje es retransmitido constantemente hasta que llegue a un nodo que ya pertenezca al árbol de recubrimiento



(a) Construcción paso a paso del árbol de recubrimiento (E es el nodo central)

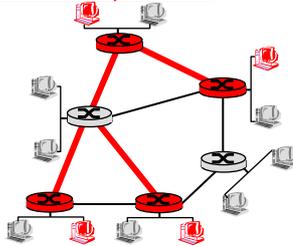


(b) Árbol de recubrimiento generado

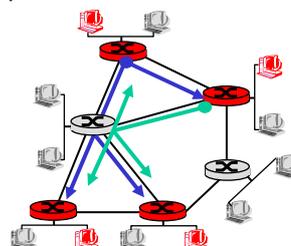
Capa de red 4-131

## Enrutado multi-difusión: ¿cuál es el problema?

- ❖ **Objetivo:** encontrar un árbol (o árboles) que conecten routers que tengan miembros del grupo multidifusión
  - **árbol:** no se usan todas las rutas entre routers
  - **basado en el origen:** árboles diferentes para orígenes diferentes
  - **árbol compartido:** mismo árbol para todos los miembros del grupo



Árbol compartido



Árboles basados en el origen

Capa de red 4-132

## Posibilidades para construir árboles multi-difusión

Posibilidades:

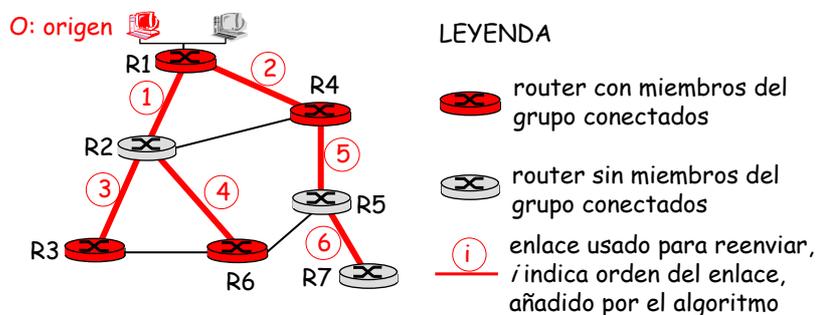
- ❖ **árbol basado en el origen:** un árbol por cada origen
  - árboles de camino mínimo
  - reenvío por el camino inverso
- ❖ **árbol compartido en el grupo :** el grupo usa un solo árbol
  - recubrimiento mínimo (Steiner)
  - árboles "centrados"

...miramos primero posibilidades básicas; luego los protocolos que adoptan esas posibilidades

Capa de red 4-133

## Árbol de camino mínimo

- ❖ **árbol de reenvío multi-difusión:** árbol con las rutas más cortas desde un origen a todos los destinos
  - algoritmo Dijkstra



Capa de red 4-134

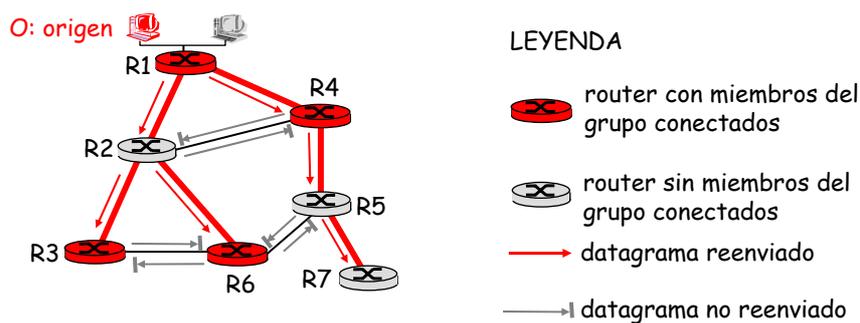
## Reenvío por el camino inverso

- ❖ se basa en que el router conoce el camino unidifusión más corto desde el remitente
- ❖ cada router se comporta de este sencillo modo:

*si* (recibido un datagrama de multidifusión que ha venido desde su origen por el camino más)  
*entonces* inundar todos los enlaces de salida con ese datagrama  
*y si no* ignorar datagrama

Capa de red 4-135

## Reenvío por el camino inverso: ejemplo

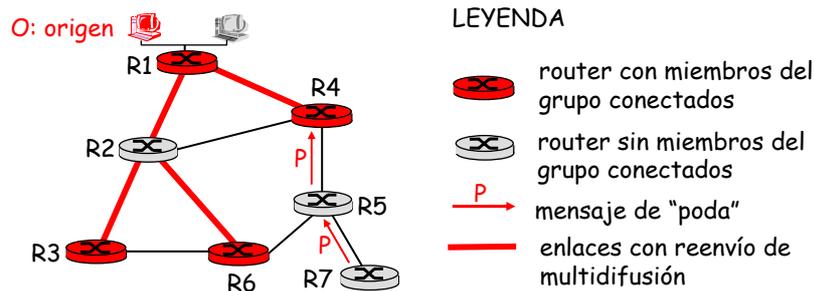


- ❖ el resultado es un árbol de camino mínimo *inverso*, específico para cada posible origen
  - puede ser mala idea en enlaces asimétricos

Capa de red 4-136

## Reenvío por el camino inverso: poda

- ❖ el árbol de reenvío contiene sub-árboles sin miembros de grupo de multidifusión
  - no se necesita reenviar datagramas por ellos
  - el router sin miembros conectados larga hacia arriba un mensaje de "poda"



Capa de red 4-137

## Enrutado por multidifusión en Internet: DVMRP

- ❖ **DVMRP**: *distance vector multicast routing protocol*, RFC1075
- ❖ **inundación y poda**: árbol basado en el origen, reenvío por camino inverso y poda
  - árbol RPF basado en las propias tablas DVMRP, construidas por comunicación entre routers DVMRP
  - sin hipótesis sobre la unidifusión subyacente
  - datagrama inicial al grupo de multidifusión inundado por todas partes vía RPF
  - router que no quiera grupo: que envíe hacia arriba un mensaje de poda

Capa de red 4-138

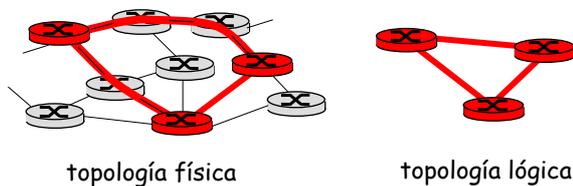
## DVMRP: seguimos...

- ❖ ***estado soft***: router DVMRP se "olvida" periódicamente (1 min.) de que las ramas están podadas:
  - de nuevo se envían datagramas de multidifusión hacia abajo por la rama sin podar
  - router que esté "por abajo": que vuelva a podar, o que siga recibiendo datos
- ❖ routers se pueden enganchar rápidamente al árbol
  - siguiendo IGMP se juntan en la hoja
- ❖ detallitos...
  - habitualmente implementado en routers comerciales
  - el routing "mega-troncal" usa DVMRP

Capa de red 4-139

## Tunelado

***P:*** ¿Cómo conectar "islas" de routers multidifusión en un "mar" de routers unidifusión?



- ❖ datagrama multidifusión encapsulado en un datagrama "normal" (sin dirección de multidifusión)
- ❖ datagrama IP normal enviado "en modo túnel" vía unidifusión IP normal al router multidifusión de destino
- ❖ router destino desencapsula y obtiene el datagrama multidifusión

Capa de red 4-140

## PIM: Protocol Independent Multicast

- ❖ independiente del algoritmo de routing unidifusión subyacente (funciona con todos)
- ❖ dos posibles escenarios para la multidifusión:

### Denso:

- ❖ miembros del grupo muy "próximos" entre sí
- ❖ ancho de banda a saco

### Disperso:

- ❖ nº de redes con miembros del grupo pequeño en relación al nº de redes interconectadas
- ❖ miembros muy "alejados"
- ❖ ancho de banda limitado

Capa de red 4-141

## Consecuencias de la dicotomía denso-disperso:

### Denso

- ❖ *se asume* pertenencia al grupo, hasta que los routers podan explícitamente
- ❖ construcción del árbol de multidifusión basándose en datos (p.e., RPF)
- ❖ *derrochando* ancho de banda y procesamiento de routers sin grupo

### Disperso:

- ❖ no pertenencia hasta que los routers se unen explícitamente
- ❖ construcción del árbol de multidifusión basándose en el receptor (p.e., árbol central)
- ❖ *conservando* ancho de banda y procesamiento de routers sin grupo

Capa de red 4-142

## PIM- Modo Denso

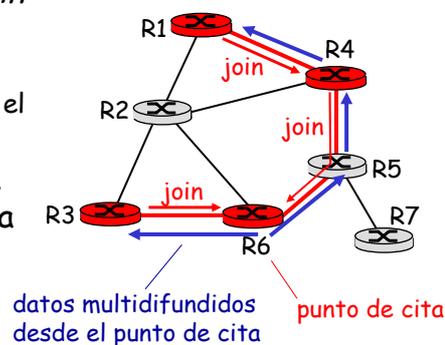
**RPF inundación y poda**, similar a DVMRP pero...

- ❖ el protocolo subyacente le da a RPF info sobre el datagrama entrante
- ❖ la inundación es menos complicada (menos eficiente) que el DVMRP, reduce el soporte necesario del algoritmo de routing subyacente
- ❖ tiene mecanismos para detectar si un router es una "hoja" del árbol

Capa de red 4-143

## PIM - Modo Disperso

- ❖ solución central
- ❖ router envía mensajes *join* a un punto de cita (RP)
  - routers intermedios actualizan estado, reenvían el *join*
- ❖ después de unirse vía RP, el router puede cambiar a "árbol con origen específico"
  - así aumenta rendimiento: menos concentración, caminos más cortos

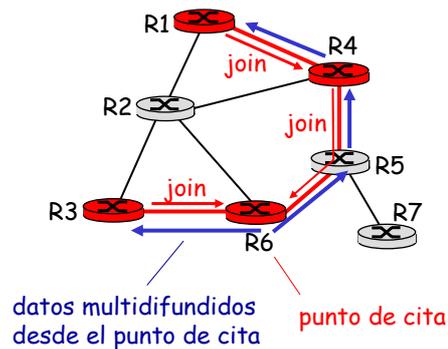


Capa de red 4-144

## PIM - Modo Disperso

### remitente(s):

- ❖ datos unidifusión a RP, quien los distribuye por un árbol enraizado en RP
- ❖ RP puede extender el árbol multidifusión hacia arriba, hacia el origen
- ❖ RP puede enviar mensaje de *stop* si no hay receptores
  - "nadie está escuchando!"



Capa de red 4-145

## Capítulo 4: resumen

- 4.1 Introducción
- 4.2 Redes de circuitos virtuales y de datagramas
- 4.3 Las tripas de un router
- 4.4 IP: Internet Protocol
  - Formato de datagramas
  - Direccionamiento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de enrutamiento
  - Estado de enlaces
  - Vector de distancias
  - Enrutamiento jerárquico
- 4.6 Enrutamiento en Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Enrutamiento por difusión y multidifusión

Capa de red 4-146