Capa Red

Andrés Arcia-Moret

(Referencia: Computer Networking. 5th edition. Kurose & Ross)

Capa de Red: Introducción

- Responde a la pregunta ¿Cómo comunicar un par de nodos?
- Existe al menos una parte de la capa de red dentro de cada host de la Inet
- Veremos datagramas vs. virtual circuit model
- ¿Qué papel juega el direccionamiento?
- Aprenderemos a separar el *forwarding* del *routing*
  - *Forwarding —› paso simple de la entrada a la salida*
  - *Routing —› se toma en cuenta el camino completo*

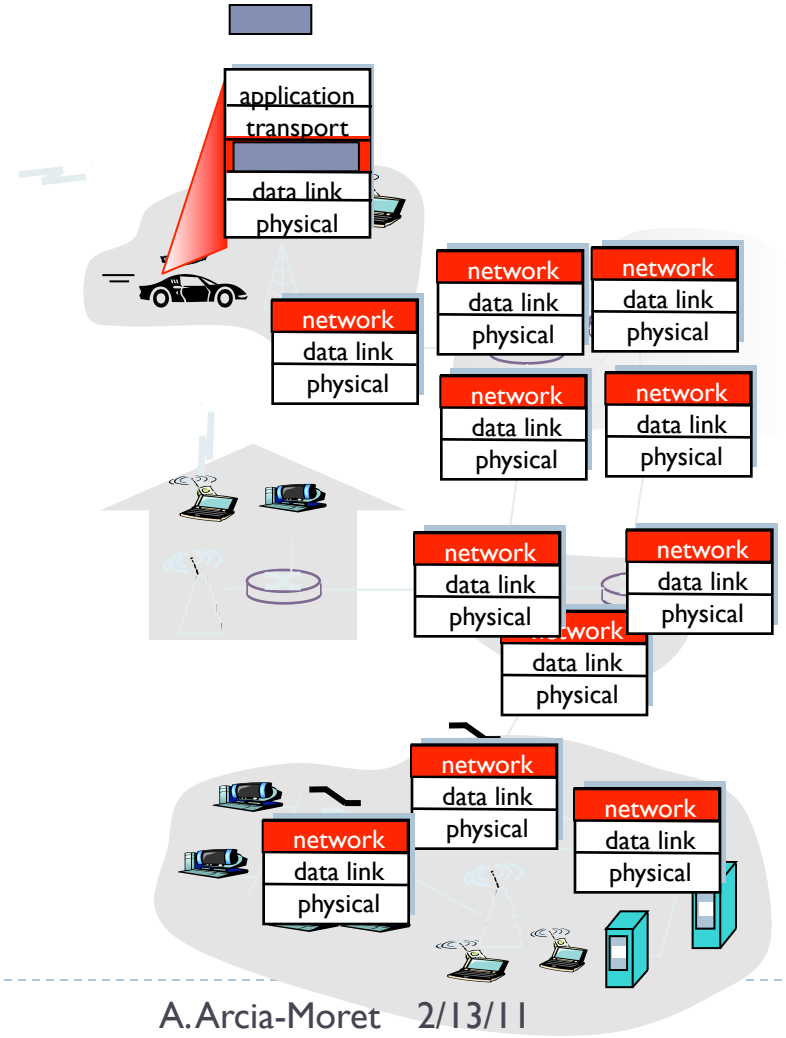
Objetivos

- Comprender los principios detras de los servicios de la capa de red:
  - Modelos de los servicios de la capa red
  - Forwarding versus Routing
  - ¿Cómo funciona un ruteador?
  - Selección de caminos en ruteo.
  - Manejo de la escala.
  - IPv6, mobilidad.
- Instanciación e implementación en la Internet.

Viaje de un Segmento en una Red

- Vea las pilas truncadas en los nodos intermedios:

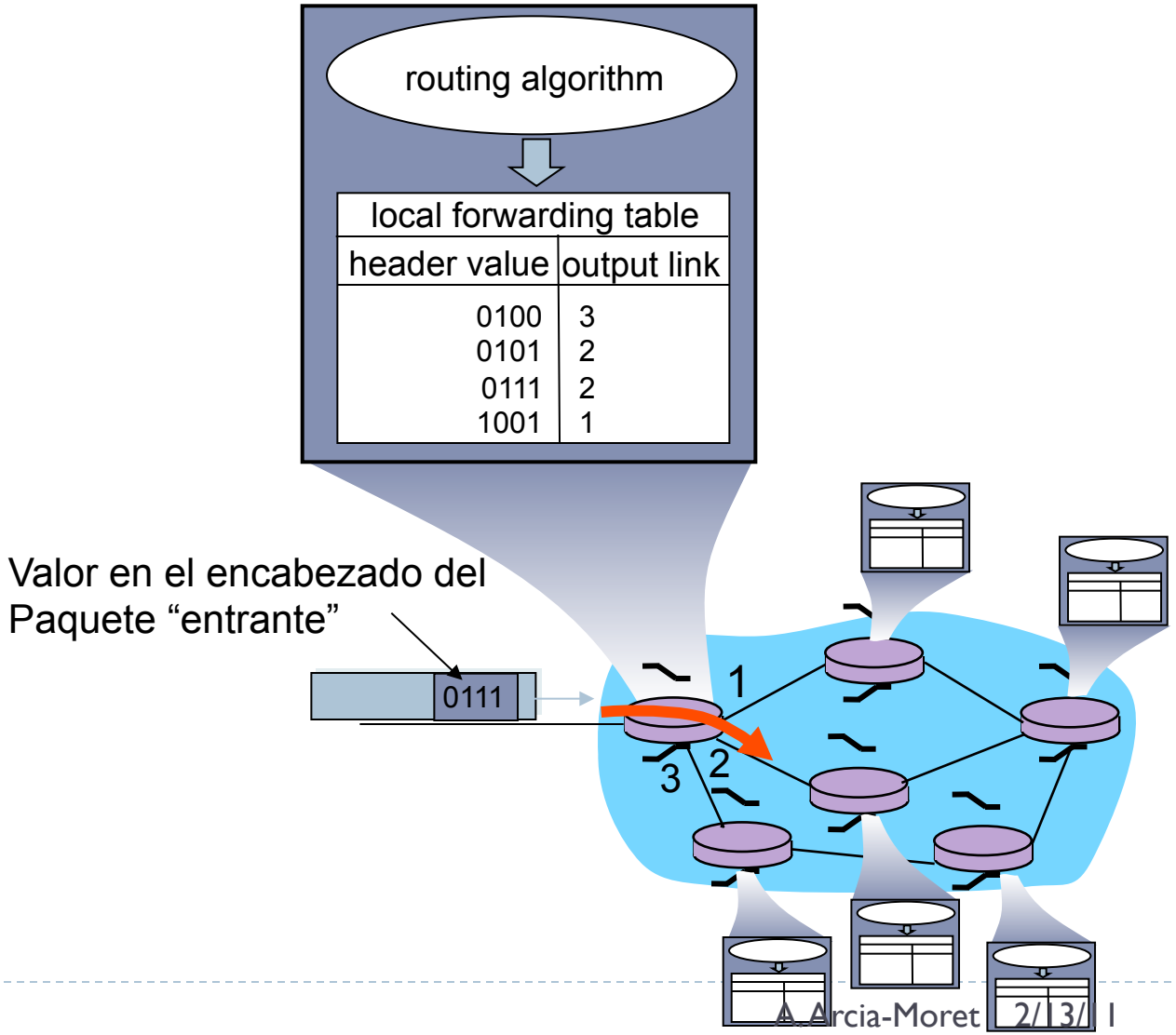
  - En los nodos intermedios hay pilas truncadas.
  - Se encapsulan los segmentos en datagramas
  - Cada enrutador procesa el encabezado IP de cada paquete



Funciones principales de la Capa Red

- Forwarding
  - Mover el paquete al próximo ruteador cercano al destino
  - Se tiene una tabla de forwarding por nodo
  - Mediante un ID (destino o ID del flujo) en el paquete se ubica la puerta de salida en la tabla.
- Routing
  - Explorar el camino desde el que envía hasta el que recibe
  - Considere el Routing como un viaje donde se toman decisiones en cada punto de intersección

Forwarding versus Routing



Configuración de las Tablas de Forwarding

- Diferencias entre forwarding y routing
  - Centralizado: Un sitio que provee la información a todos los ruteadores
  - Descentralizado: Una parte del algoritmo corre en cada host
- Situación extrema: Tablas de enrutamiento configuradas por seres humanos (lento y propenso a errores).
- Diferencia entre packet switch y routing
  - **Packet Switch**: Lleva paquetes de una entrada a la salida (en capa enlace)
  - **Router**: Examina la información de la capa de red para mover el paquete.

Establecimiento de la Conexión

- Función de la capa de red
- Para hacer forwarding y ruteo debe previamente haber una conexión virtual (como TCP). Ej: ATM, Frame Relay
- Antes de transmitir paquetes se establece una conexión entre los ruteadores
- Recuerde
  - Red: comunica 2 hosts
  - Transporte: comunica 2 procesos

Modelos de Servicio de la Capa Red

- El modelo de servicio responde a las siguientes preguntas:
  - ¿La capa transporte puede contar con la capa Red para entregar paquetes?
  - ¿En que orden se entregan los paquetes al destino?
  - ¿Cuál será el tiempo entre paquetes a la recepción?
  - ¿La red transporta señales sobre de congestión?

Los Servicios

Servicios ofrecidos a la capa transporte:

- Para un paquete
  - Entrega garantizada: el paquete llegará (eventualmente) al destino
  - Entrega garantizada con retardo acotado
- Para un flujo:
  - Entrega en orden: la secuencia de envio es la secuencia de recepción
  - Mínimo ancho de banda: Servicio de red emula un enlace con caracteristicas propias (ej: minimo bandwidth) —› exceso de tasa critica resulta en perdidas
  - Máximo Jitter: El tiempo de entrega entre 2 paquetes no excede un límite.

Los Servicios (2)

- Servicio de Seguridad: encriptado usando clave para encriptamiento por sesión.
- Pero la **capa de red de Internet** provee solamente el servicio "best-effort" (eufemismo para decir, "no hay servicio alguno").
  - No se garantiza timing entre paquetes, orden, fiabilidad

Modelos de Servicio

Network Architecture	Service Model	Guarantees ?				Congestion feedback
		Bandwidth	Loss	Order	Timing	
Internet	best effort	none	no	no	no	no (inferred via loss)
ATM	CBR	constant rate	yes	yes	yes	no congestion
ATM	VBR	guaranteed rate	yes	yes	yes	no congestion
ATM	ABR	guaranteed minimum	no	yes	no	yes
ATM	UBR	none	no	yes	no	no

Servicios Con y Sin Conexión de la Capa Red

- Red de datagramas —› servicio sin conexión
- Red de circuitos virtuales —› servicio CON conexión
- Servicio Red se comporta como servicio Transporte
  - Paralelo:
    - Setup y Handshake de la conexión
  - Diferencias:
    - Servicio host-to-host vs process-to-process
    - Se provee el uno o el otro (no los dos)
    - Se implementa en el CORE de la red (transporte en los extremos)

Circuitos Virtuales: Caso ATM

- Constant Bit Rate (CBR)
  - Primer servicio estandarizado para llevar a cabo tiempo real: audio/video
  - Objetivo: proveer celdas ATM que emulen un canal con ancho de banda fijo —› garantías BW, delay y perdidas.
- Available Bit Rate (ABR)
  - Un poco mejor que "Best Effort"
  - Las celdas pueden perderse
  - NO hay reordenamiento
  - Si hay recursos se puede transmitir más rápido que el Minimal Cell Rate —› aprovecha lo que hay.

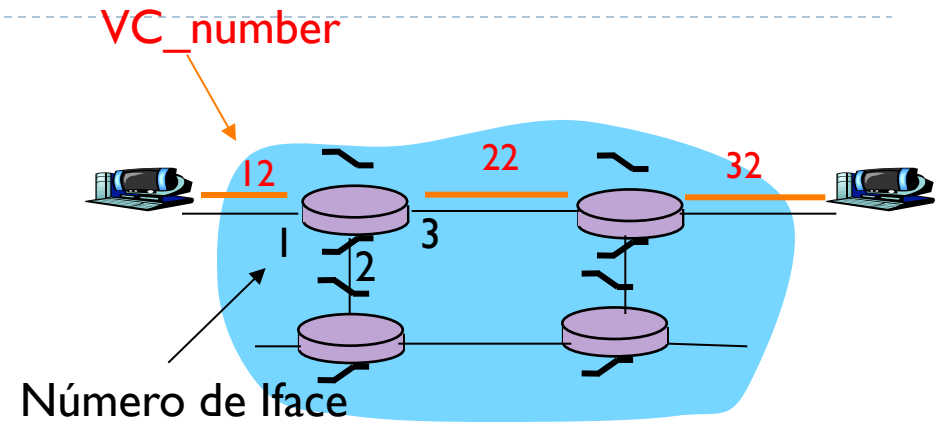
Circuito Virtual

- ¿Cómo se implementa?
  - Un camino (path) = links + routers
    - Se comporta como un circuito telefónico
  - Identificación de los paquetes y enlaces (un ID por cada uno)
  - Registros en la tabla de ruteo a lo largo del camino (almacenamiento de estados)
  - Se obtiene cada número de VC en las tablas de ruteo
  - Recursos de BW y buffer se puede reservar a un VC (para tener servicios predecibles).

Circuito Virtual

- ¿Por qué no el mismo número de celula (ATM cell)?
  - Habría que ponerse de acuerdo para preservar números únicos a lo largo del camino
  - El cambio reduce (simplifica) el tamaño del campo VC_number
  - Cada vez que se crea una conexión se añade una entrada a la tabla.
    - Forwarding Table mantiene información de la conexión <VC_number, OUT_iface> —› Aquí la celda cambia de VC_number.
- Un VC está compuesto de
  - Un camino
  - Numeros de VC por cada camino
  - Entradas en la Forwarding Table

Ejemplo



Forwarding table en ruteador noroeste:

Incoming interface	Incoming VC #	Outgoing interface	Outgoing VC #
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
...	...	...	...

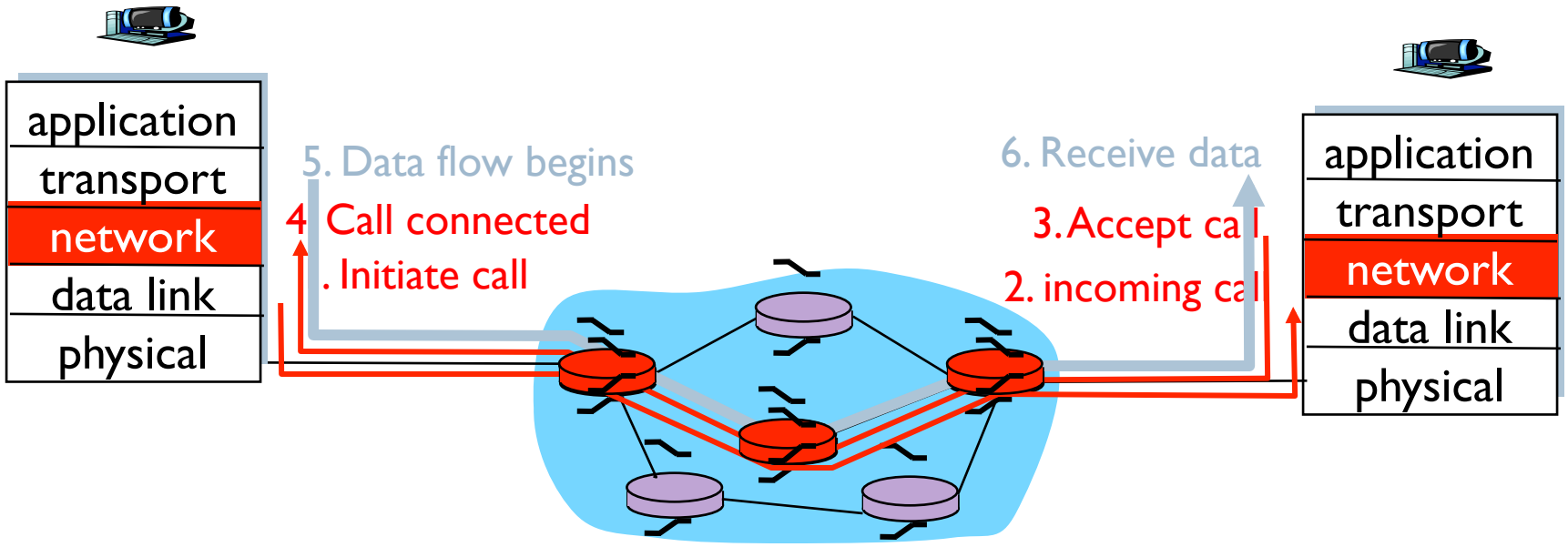
Los ruteadores mantienen información del estado de conexión!

Fases en un VC

- Establecimiento de VC
  - Capa transporte contacta la capa red y espera que se haga el VC.
  - Se reservan recursos: entradas en la tabla, ancho de banda.
- Transferencia de Datos
- Rompimiento del VC
  - Ejecutado por el emisor o receptor

Circuito Virtual (ejemplo)

- Ya no es usado hoy en día:



Pregunta

- ¿Recuerda cual es la desventaja principal de los circuitos virtuales?
- ¿Cómo se resolvión esta desventaja?

Redes de Datagramas

- No se necesita apertura de conexión
- Se marca el paquete con la dirección destino y se envía a la red
- No hay información por flujo almacenada en routers
- Se usa la dirección destino para buscar en la Tabla de Ruteo

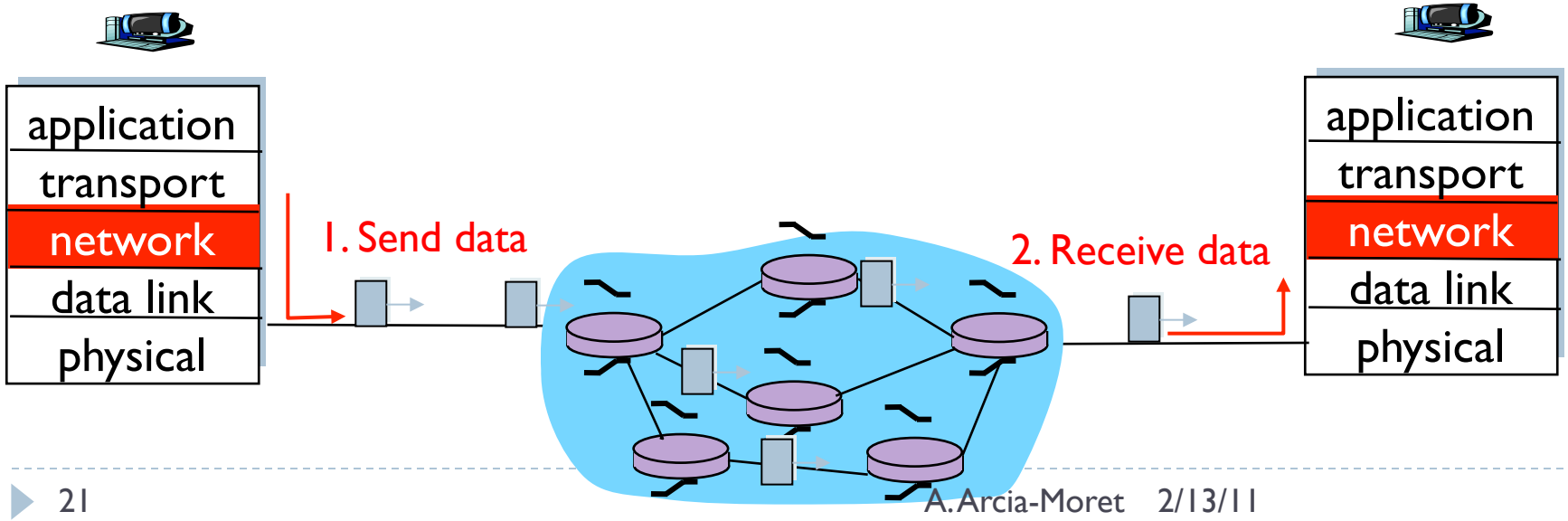


Tabla de Forwarding

4 mil millones de Entradas posibles

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

Busqueda del prefijo más largo

Prefix Match	Interface
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
otherwise	3

Ejemplos

DA: 11001000 00010111 00010110 10100001 Cual interface ?

DA: 11001000 00010111 00011000 10101010 Cual interface ?

Busqueda del prefijo más largo

- Puede haber más de una "posible" entrada en la tabla (resuelto con la regla del más largo)
- Tablas actualizadas cada 5 min
  - Puede haber reordenamiento cada 5 min

Datagrama versus VC

Datagramas:

- Servicio elastico
- Sistemas finales son "inteligentes"
- Heterogeneidad de Enlaces

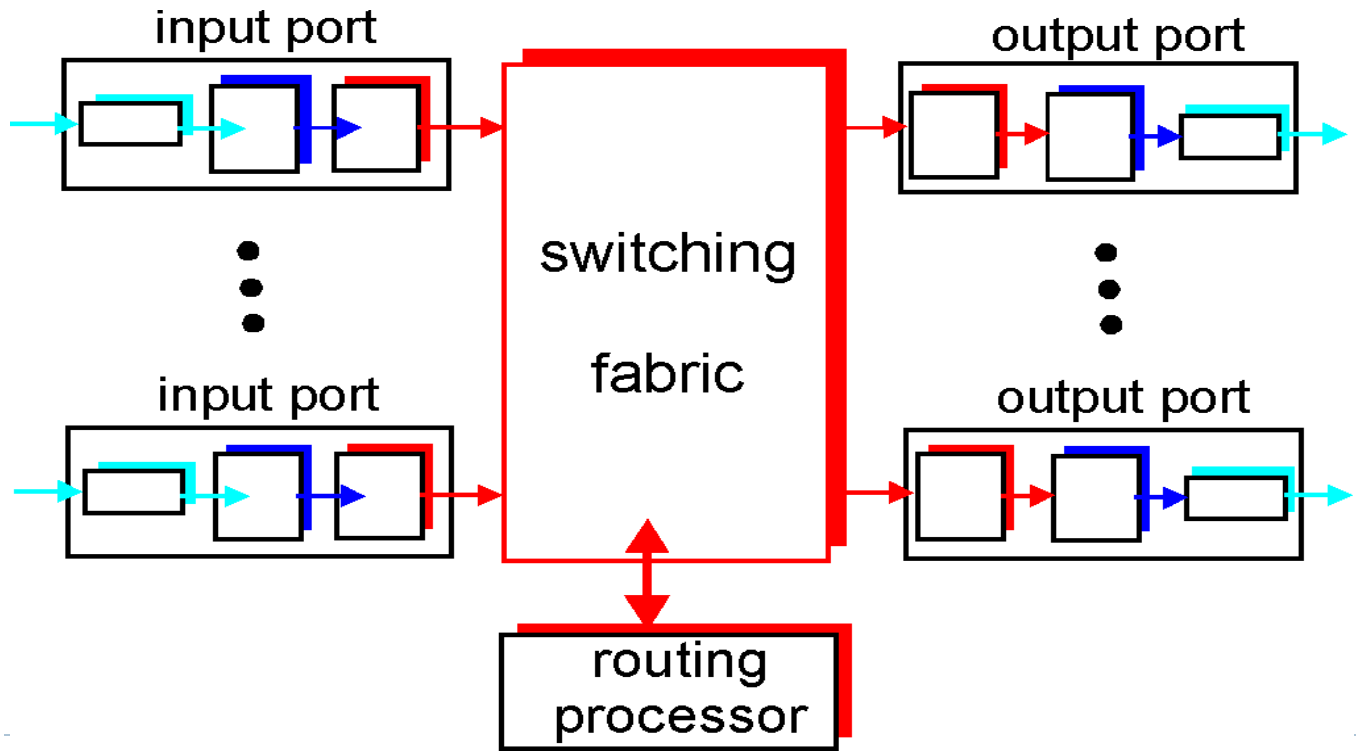
Circuitos Virtuales:

- Evolucionó de la telefonía
- Terminales tontos
- Heterogeneidad de Enlaces

¿Qué hay dentro de un ruteador?

Dos funciones principales de un ruteador:

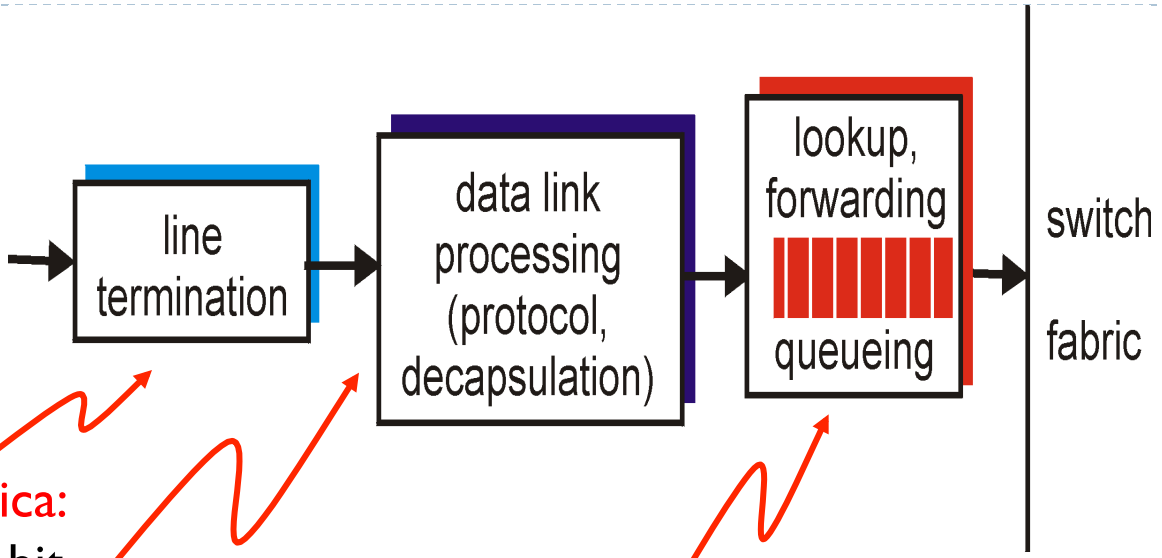
- Ejecuta algoritmos de ruteo (RIP, OSPF, BGP)
- *forwarding* datagrams de la entrada a la salida.



¿Qué hay dentro de un ruteador?

- Puertos de entrada
  - Establecen conexión con los enlaces físicos
  - Se hace una clasificación de paquetes (que van a la salida o al procesador)
- Fabrica de switches
  - Conecta puertas de entrada y salida
- Puertos de salida
  - Almacena los paquetes que van de salida (previamente clasificados)
- Procesador de ruteo
  - Ejecuta protocolos de ruteo

Los Puertos de Entrada



Capa física:
Recepción a nivel bit

Capa Enlace:
Ej: Ethernet

Switching descentralizado:

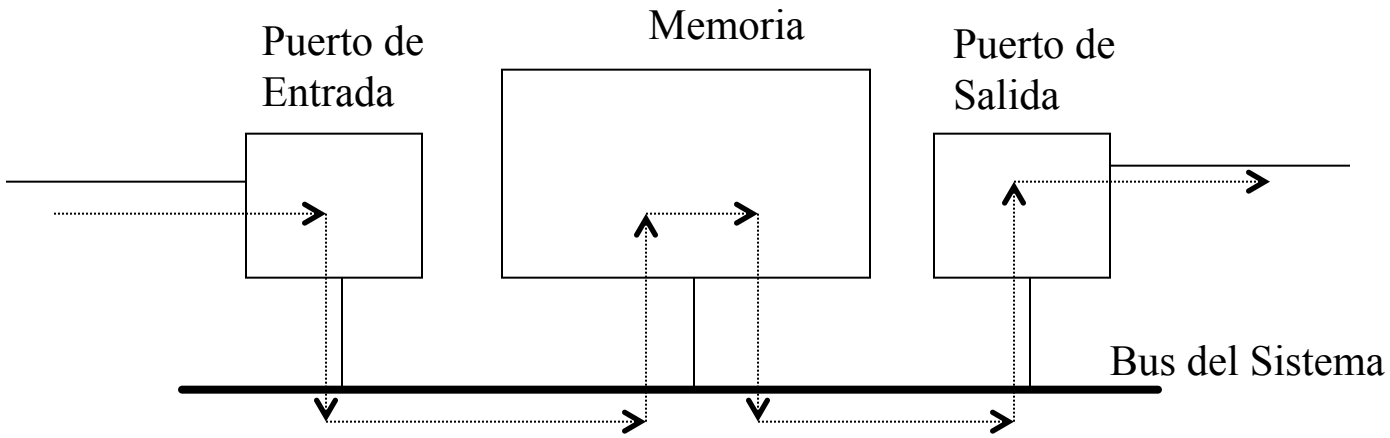
- Dado el destino, buscar el puerto de salida usando la "forwarding table" en la memoria del puerto de entrada
- **Objetivo**: Completar el procesamiento en el puerto de entrada a la velocidad de la conexión.
- **Encolamiento**: Si los datagramas llegan más rápidamente que la tasa de reenvio dentro de la fabrica de switches.

Los Puertos de Entrada

- **Implementa** la capa física y de enlace.
- **Determina** el puerto de salida a través de la fabrica de switches (usando la Forwarding Table 'FT')
- Hay una copia de la FT en cada puerto de entrada para descentralizar la decisión —› **no ocupa** el procesador
- Se hacen millones de **lookups** por segundo (longest prefix matching) —› más rápido que la velocidad de recepción de un paquete.
  - Ej: Optical Carrier OC-48 @ 2.5 Gbps. Para paquetes 256 bytes = 1 Mega lookups/secs.
- **Búsqueda** binaria para encontrar el puerto de salida.

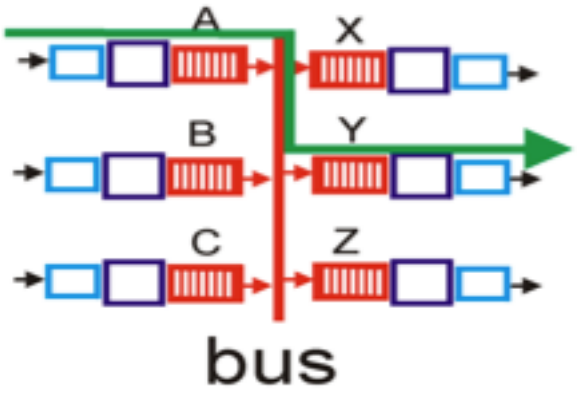
Fábrica de Switches

- Es donde ocurren los envíos de paquetes de los puertos de entrada a los puertos de salida
- Tipos de Intercambios (switching):
  - Via Memoria: primeros enrutadores a través del CPU. Paquetes de entrada interrumpen (señalizan) CPU. —› **Si** tasa de lectura/ escritura de un paquete es **B**, entonces tasa total de transmisión de 1 paquete es menos de **B/2**

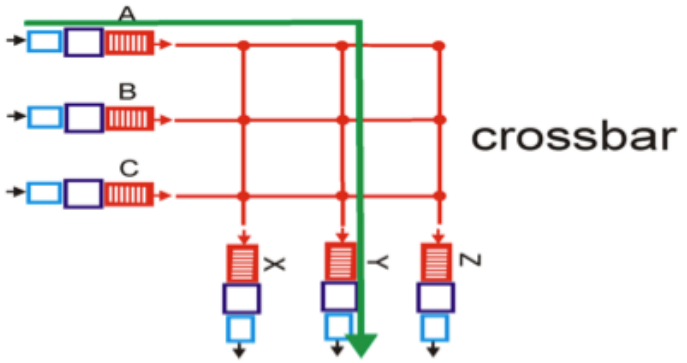


Tipos de Intercambio

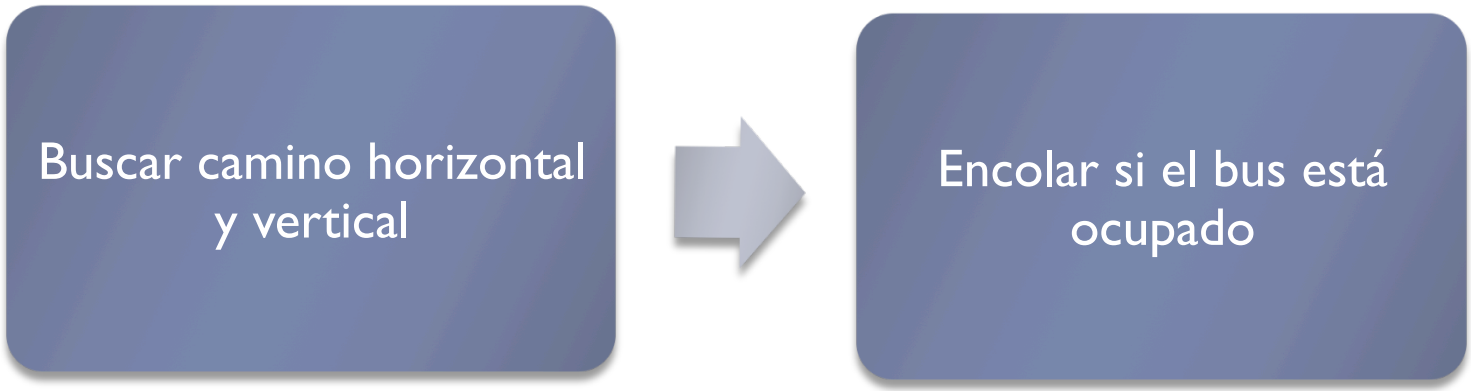
- Via Bus: Se transfiere de la entrada a la salida directamente via un *bus compartido* —› No interviene el procesador.
  - Si el bus está ocupado, se encola el paquete
  - Limite de procesamiento es la velocidad del bus (contención del bus)
  - Usado en Access Networks y Redes Empresariales (Cisco MXE 5600)



Fabrica de Switches



- Tipos de Intercambios (switching):
  - Via Red Interconectada:
    - Arquitectura usada para multiprocesadores.
    - Resuelve el problema de limitación de ancho de banda
    - Fragmenta los datagramas en celulas de tamaño fijo.
    - 2N buses que conectan N entradas con N salidas.
    - Ej: Cisco 12000, intercambia a 60 Gbps a través de su red interna.



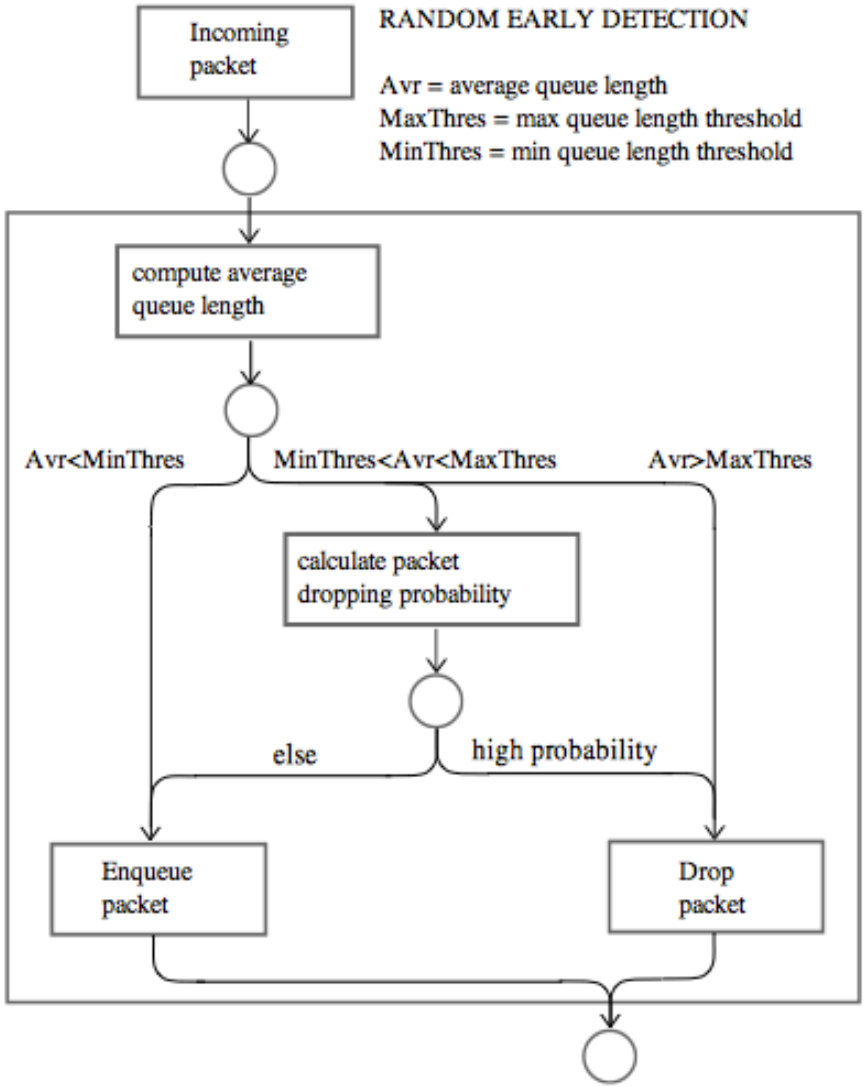
¿Dónde ocurre el encolamiento?

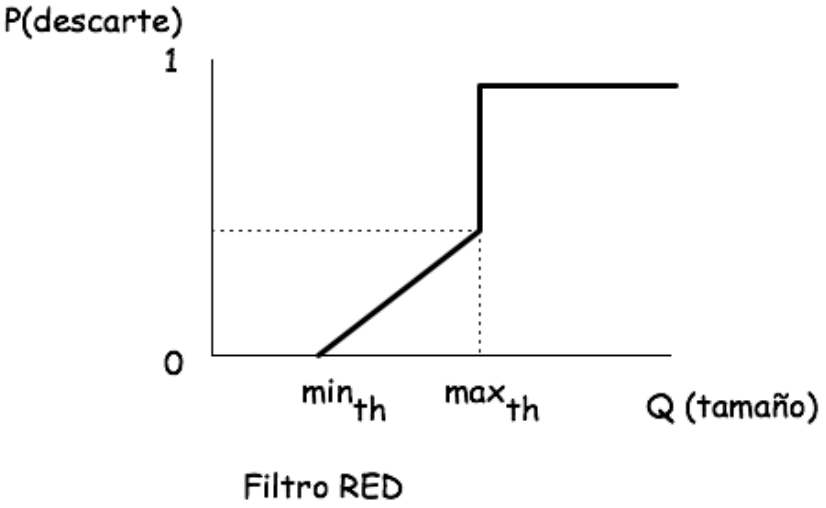
- Se puede formar en los **puertos de entrada** (espera por bus) **y salida** (por concurrencia)
- Si las colas crecen —› hay pérdidas de paquetes
- Velocidad de la fabrica de switch es la velocidad de mover paquetes de la entrada a la salida —› depende del tipo!
- Recordar que la Fabrica es N veces más rápida que la velocidad de entrada de los paquetes.
  - Tiempo de llevar N paquetes desde las N entradas a los buffers, es inferior al tiempo de recibir los N paquetes en las entradas
- Si los N paquetes de entrada van a un solo punto de la salida —› N-1 paquetes hacen cola.

Entonamiento

- Pero, ¿Cuanto buffer se necesita entonces?
  - Regla clásica: llenar el tubo → RTT x C. (RFC 3439)
    - Ejemplo: para 10 Gb x 250 ms = 2.5 GB de buffer !
  - Recientemente: para un número N grande de flujos TCP
    - B = RTT x C / sqrt(N)
- Planificación: para ofrecer calidad de servicio (QoS)
  - FCFS (first come first served)
  - Weighted Fair Queuing
- Politica de descarte de paquetes:
  - Descarte del último en llegar (droptail)
  - Descarte del primero de la cola (dropfront)
  - Marcar paquetes antes de botarlos (AQM). Ej: RED.

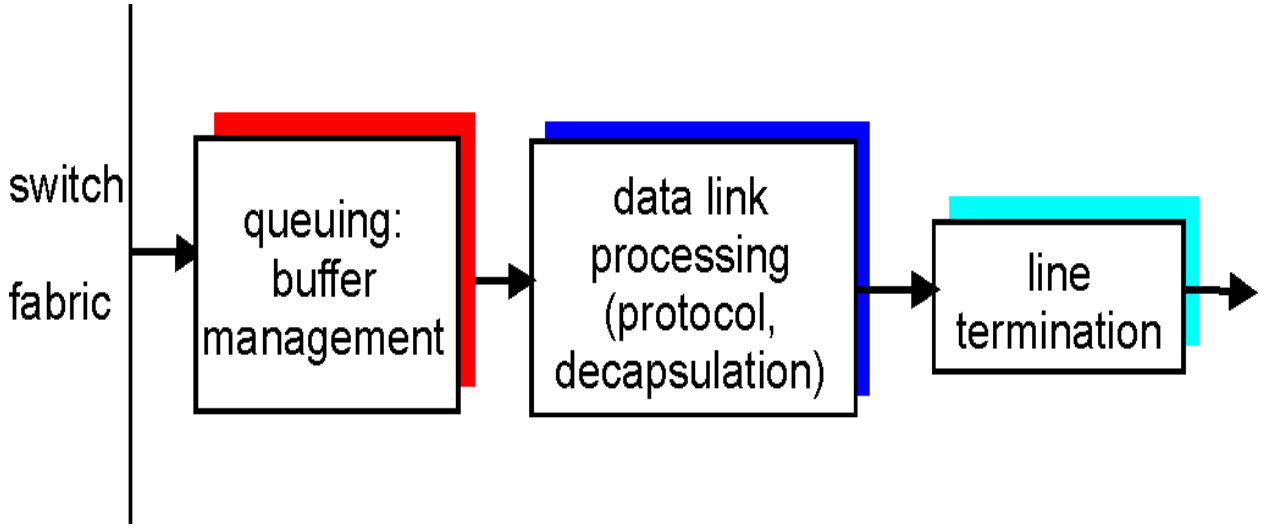
Algoritmo Random Early Detection





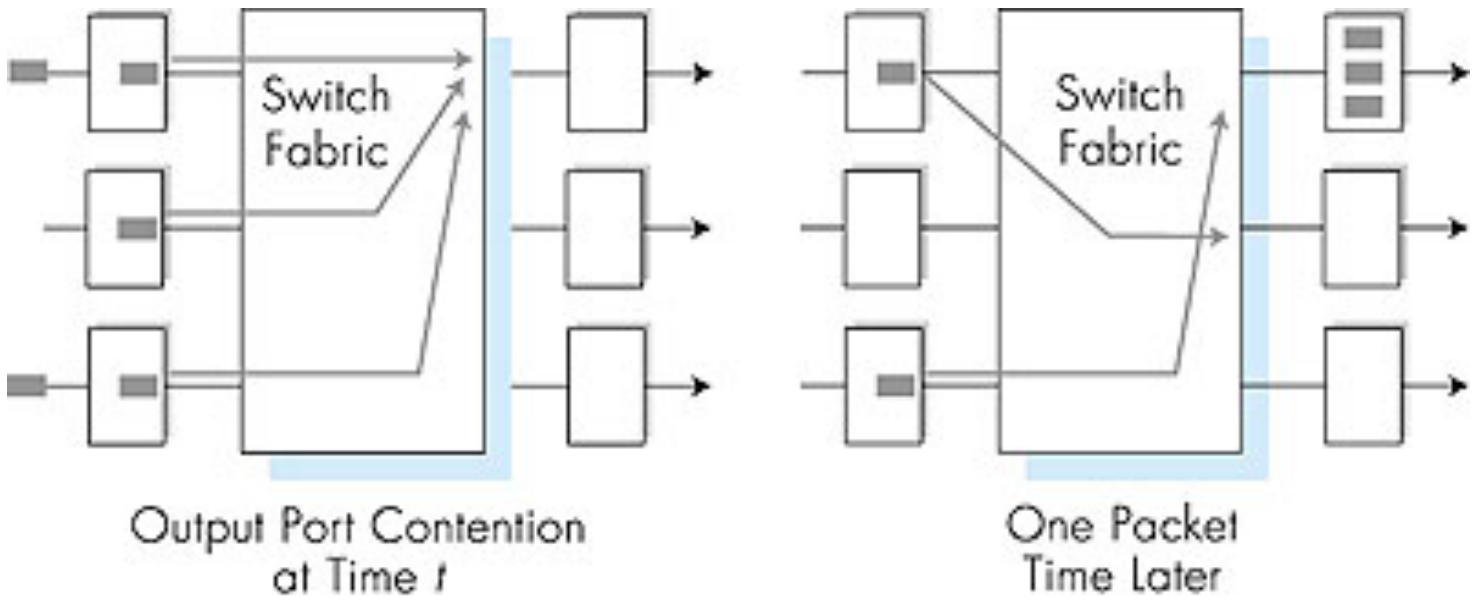
Filtro RED

Puertos de Salida



- *Buffering* se hace cuando los datagramas llegan más rápidamente que la velocidad de procesamiento
- *Diciplina de Scheduling* selecciona entre los datagramas que se tienen para ser despachados a la salida.

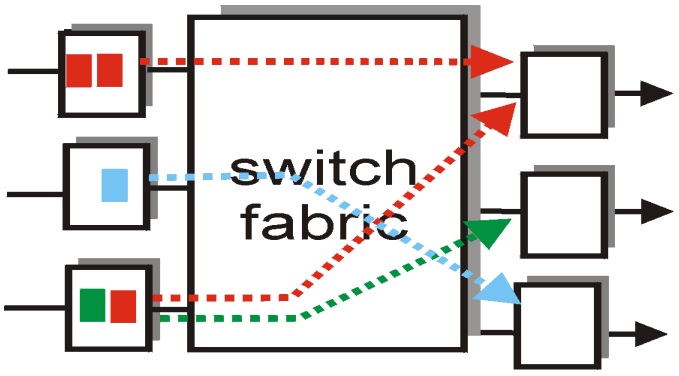
Detalles del encolamiento



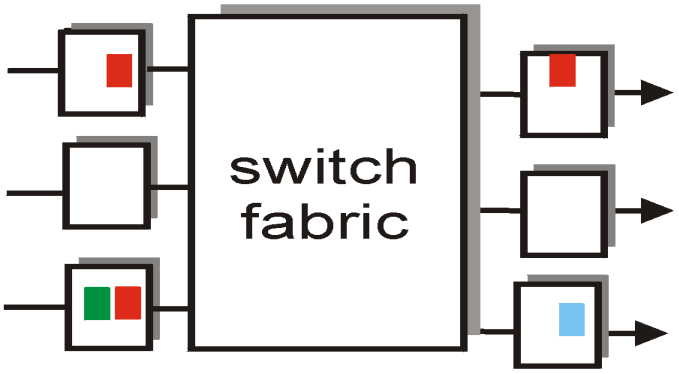
- Buffering cuando la tasa de entrada es más grande que la de salida
- *Retardo y perdidas cuando el puerto de salida esta ocupado!*

Colas en los Puertos de Entrada

- Fabrica es mas lenta cuando los puertos de entrada hacen concurrencia
- Head-of-the-line (HOL) blocking: un datagrama al principio de la cola bloquea los datagramas previos
- Perdidas y demora en los puertos de entrada!



output port contention at time t - only one red packet can be transferred



green packet experiences HOL blocking

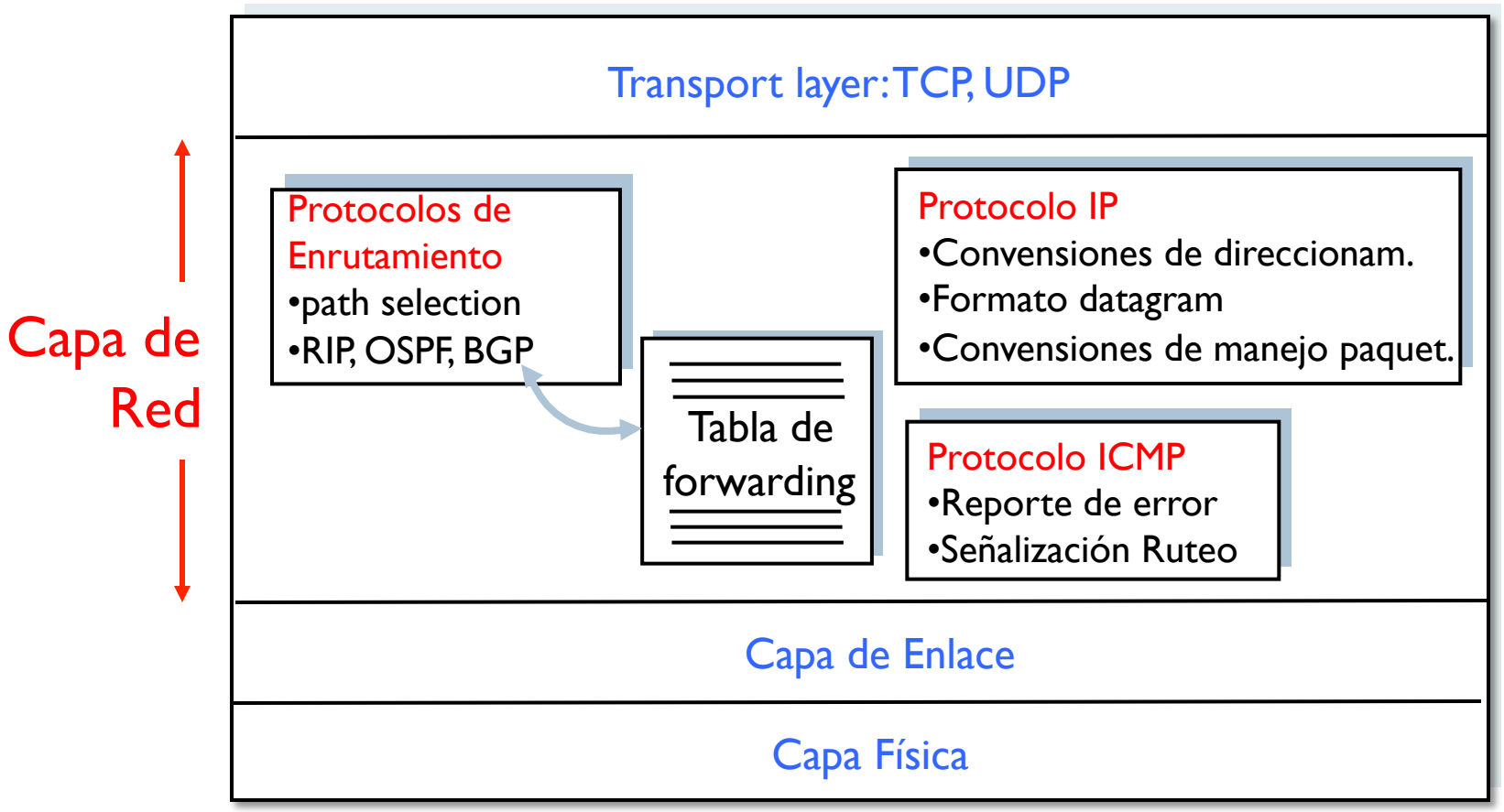
El Protocolo IP (Internet Protocol)

Descripción del protocolo de red.

El procolo IP: forwarding y direccionamiento en Internet

- Dos funciones importantes de IP: reenvío (forwarding) y direccionamiento (addressing).
- Dos versiones mayores
  - IPv4 (RFC 791)
  - IPv6 (RFC 2460 y RFC 4291)
- Componentes principales de la capa Red:
  - Protocolo IP
  - Enrutamiento (ej: Tablas de Forwarding, protocolos)
  - Mecanismos para saber el estado de la Red (ICMP).

La Capa de Red (funciones)



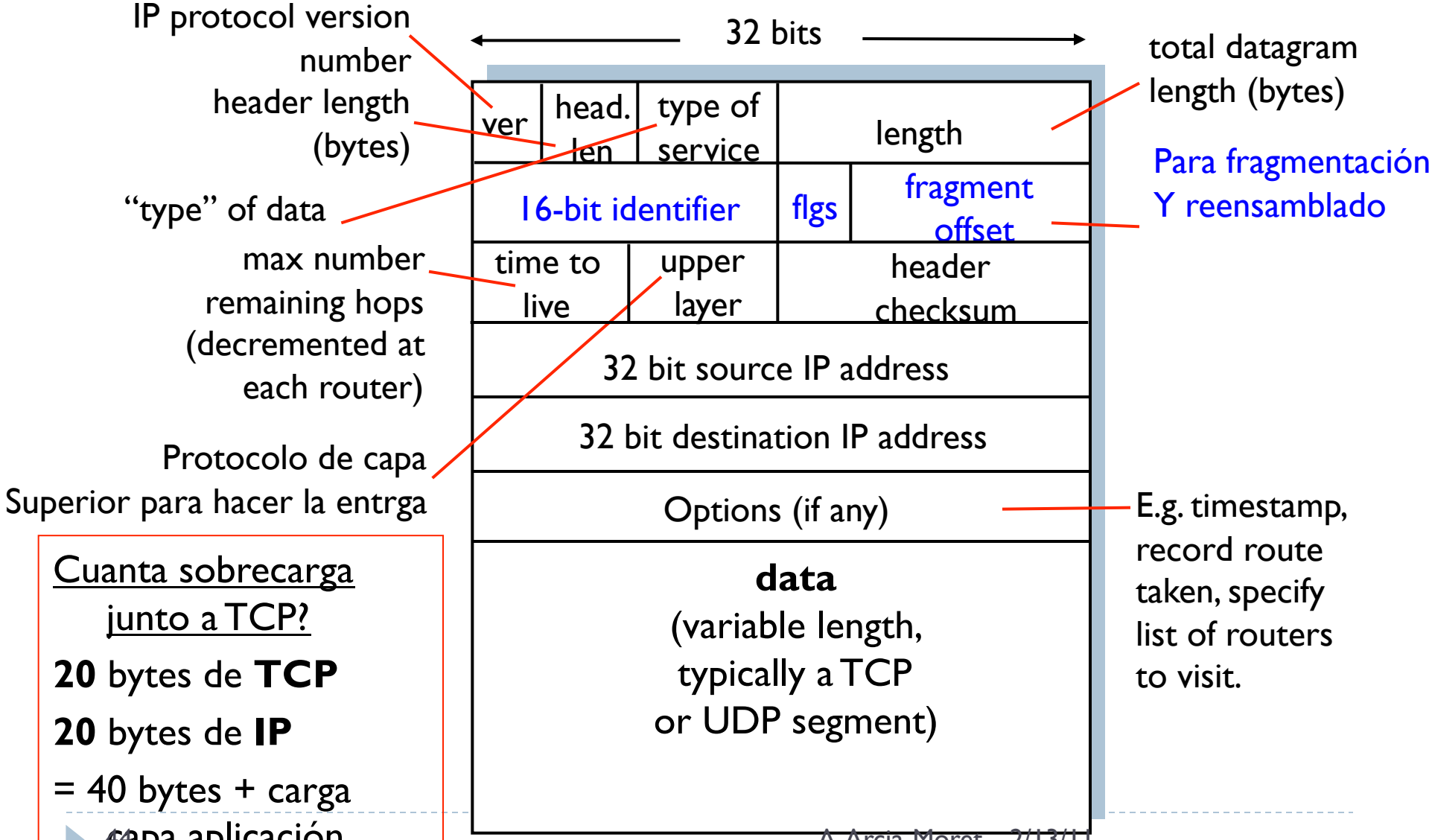
Formato de un Datagrama

- Estructura de un paquete IP es dificil de digerir! Pero, muy importante comprenderla… para todo ing. de redes.
- **Version:** IPv4 o IPv6 indica la estructura del resto del paquete
- **Header Length:** determina donde comienza la data (payload).
- **Type of Service:** diferencia la preferencia de los paquetes (ej: ***prioridad*** *a paquetes de* ***voz*** *sobre los* ***datos***).
- **Datagram Length**: 16 bits de longitud para marcar un tamaño máximo de 65535 bytes (por lo gnral son de 1500 bytes).

Formato de un Datagrama

- **Identifier, flags & fragmentation offset:** para hacer fragmentación IP (solo en IPv4)
- **Time to Live (TTL):** para que un paquete no circule para siempre. Contador se decrementa en 1 por router atravezado.
- **Protocolo:** indica el protocolo de transporte usado (6 —› TCP, 17 —› UDP) . Es la "goma" entre Red y Transporte (así como los puertos para Transporte y App).
- **Dirección Fuente y Destino:** 32 bits (4 octetos).
- **Opciones**: como en TCP para extender la funcionalidad pero complica el procesamiento.
- **Data**: La Razón de la Existencia del protocolo (y del paquete!)

Formato de un datagrama



Cuanta sobrecarga junto a TCP?

20 bytes de **TCP**

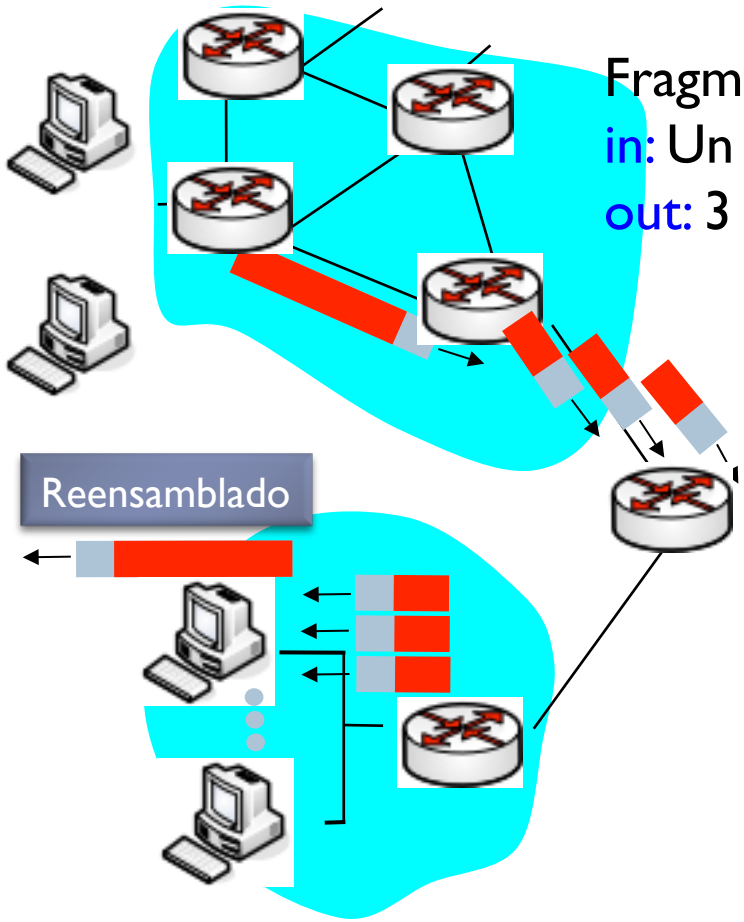
20 bytes de **IP**

= 40 bytes + carga capa aplicación

Fragmentación de un Paquete IP

- No todas las capas enlance pueden llevar paquetes del mismo tamaño.
  - Ej: Ethernet 1500 bytes vs. Wireless (WAN) 576 bytes
- Cada protocolo enlace pone límites duros
  - IP se "encapsula" en el protocolo enlace
  - Se usa el tamaño de datagrama más pequeño en la cadena.
- Si se recibe un paquete IP de talla X y debe redireccionarse a una salida con MTU Y < X entonces se fragmenta.
- Se reensambla en el **extremo final** (no en los routers, lo que simplifica el procesamiento).

Fragmentación y Reensamblado IP

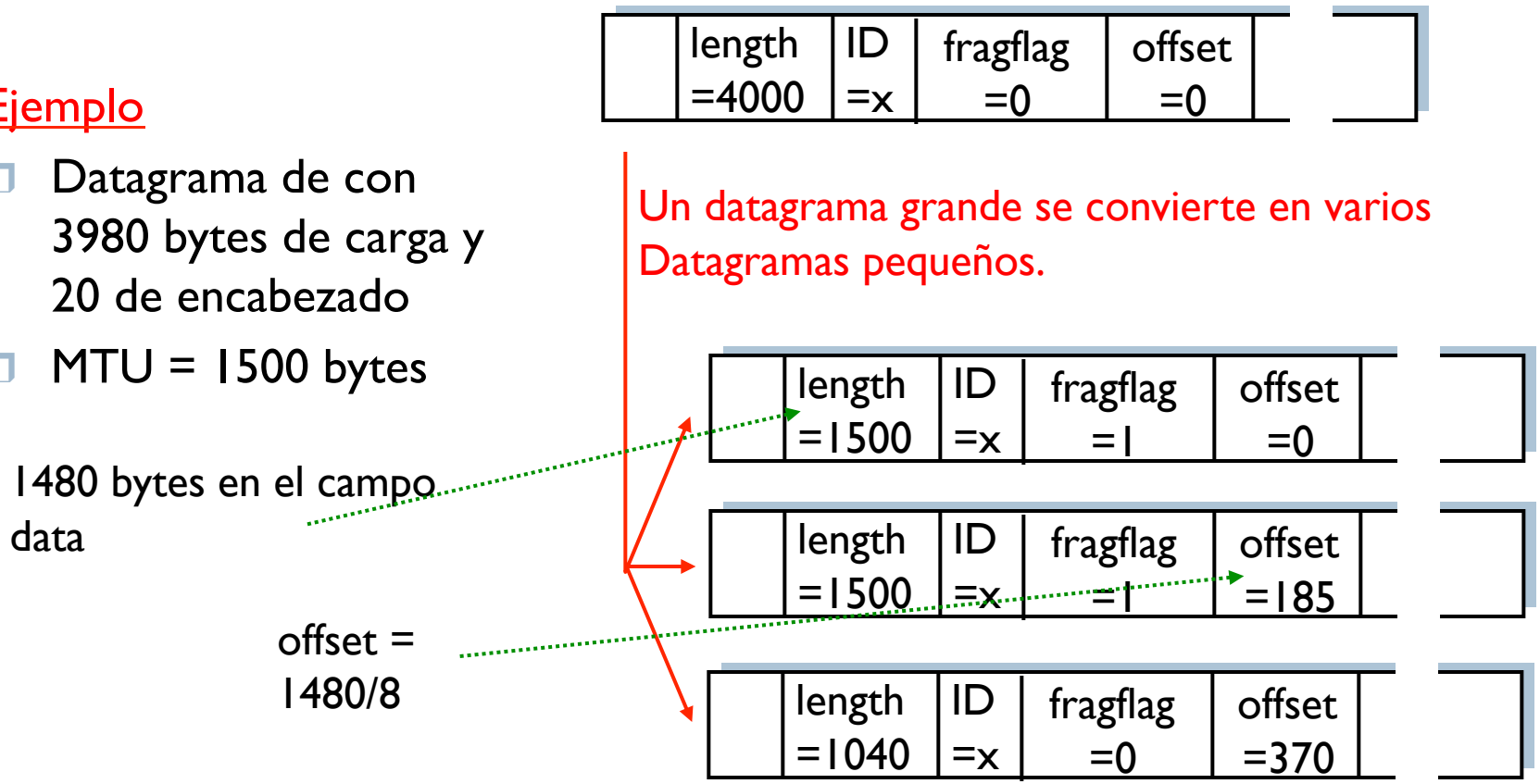


Fragmentación:
in: Un datagrama largo
out: 3 datagramas más pequeños

Ejemplo de Fragmentación/Reensamblaje

Ejemplo

- Datagrama de con 3980 bytes de carga y 20 de encabezado
- MTU = 1500 bytes



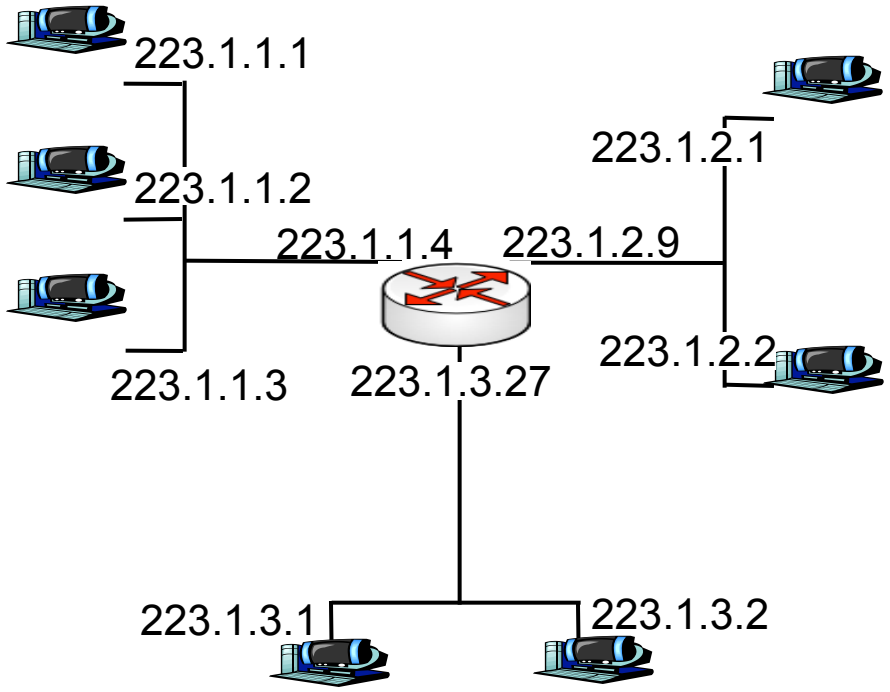
Fragmentación de un Paquete IP

- Cada fragmento mantiene la misma IP fuente, IP destino e identificador para que el destino sepa que pertenecen al mismo paquete.
  - El fragmento final se marca con ID 0.
  - El offset sirve para ordenar los fragmentos en el destino.
- Costos:
  - Complica los ruteadores y sistemas finales.
  - Puede ser usado para crear DoS letales enviando muchos fragmentos.

Direccionamiento IPv4

- El direccionamiento es tema central de la Internet. (¿Por qué?)
  - Escala
  - Servicio básico para transmisión de datos.
- Un host se conecta a la red a través de un solo link y se envían datagramas IP (con direcciones de 32 bits)
- Límite entre el host y el enlace físico es "la Interfaz"
- Como se envían y se reciben mensajes cada interfaz tiene su propio IP (IP ‹—› Interfaz)
  - Host —› tipicamente tiene una interfaz
  - Router —› varias interfaces

Introducción: Ejemplo.



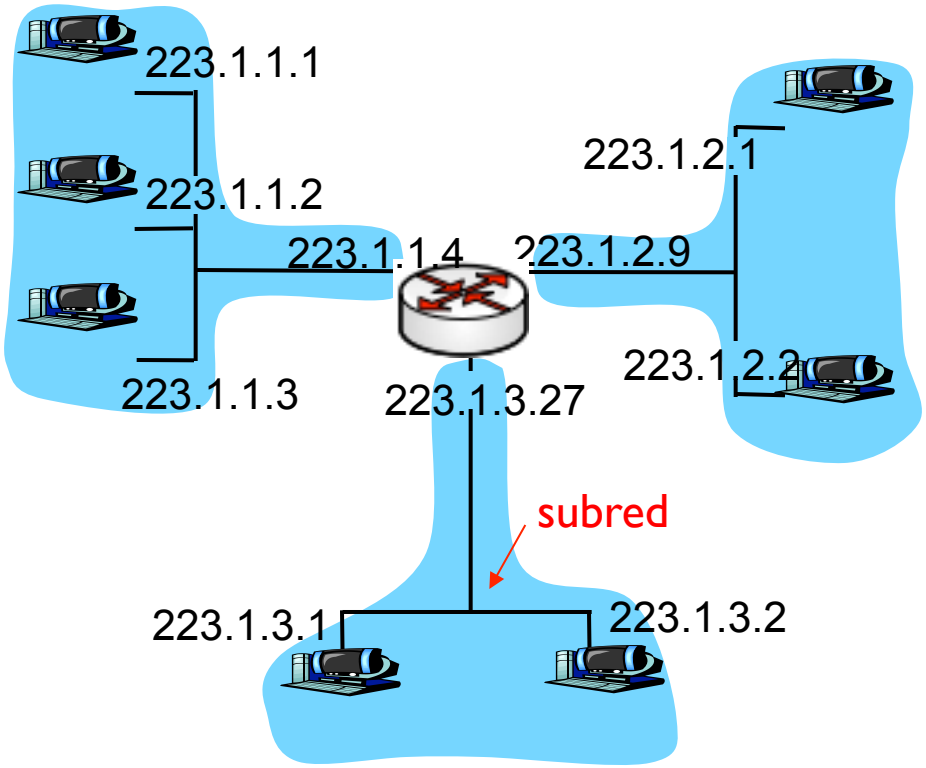
223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

Características de una Interfaz

- 32 bits de longitud → 2^{32} direcciones posibles
  - 4000 millones de direcciones
  - IP tiene una parte de la subred (izquierda)
  - Y una parte del host (derecha)
- Se escriben en decimales separados por puntos
  - Ej: 150.185.130.10 (10010110 10111001 10000010 00001010)
- Son direcciones únicas excepto las que están detrás de un NAT.
- Un host alcanza a otro SIN ruteador si están dentro de la misma subred.

Ejemplo: Subredes



Red conformada por 3 subredes

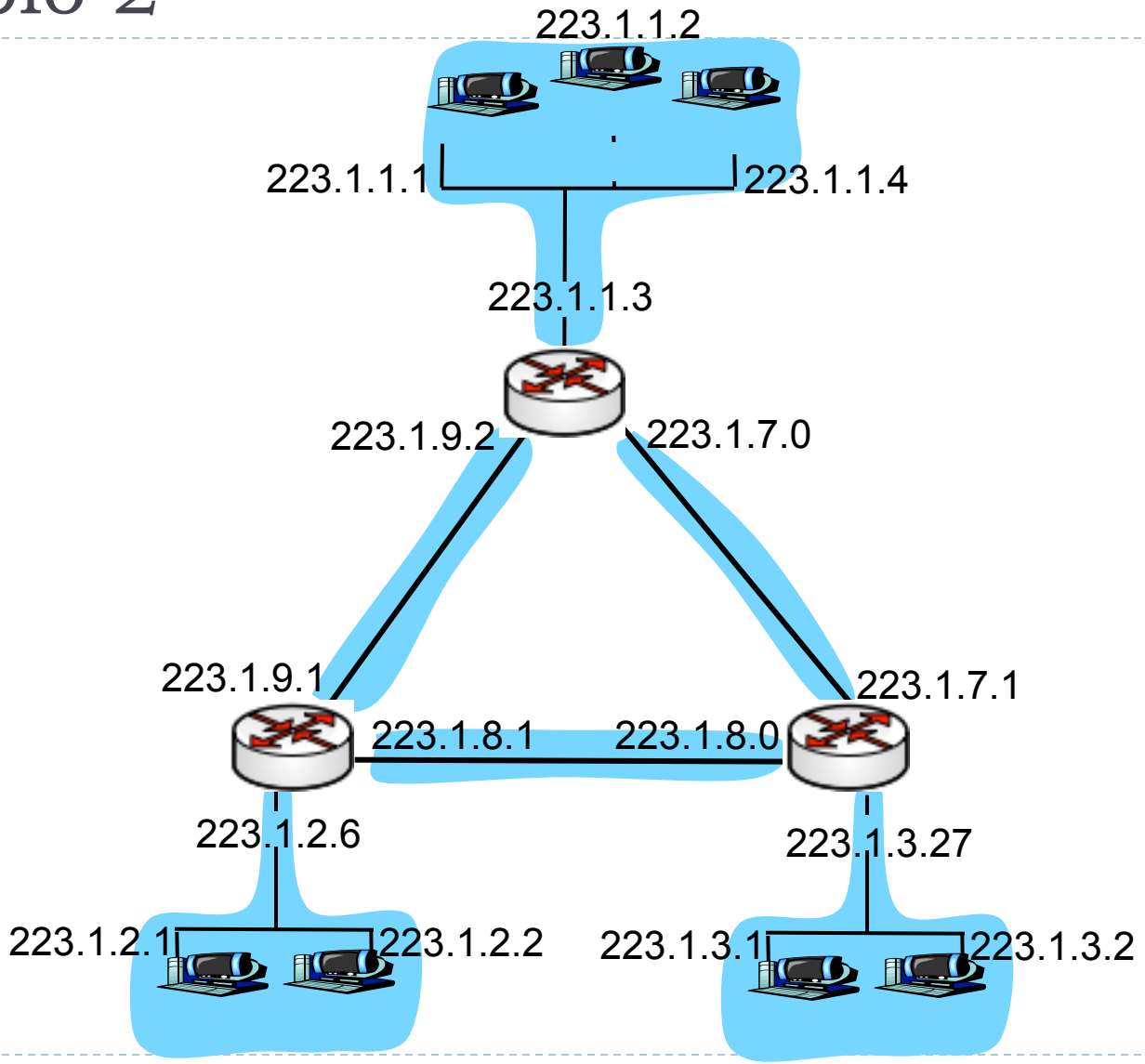
Ejemplo: Subred

- Observe los IP con prefijo común 223.1.1.xxx
- La subred de la izquierda está interconectada por una red "sin ruteadores"
  - Ej: Ethernet switch LAN (se llama oficialmente subred según **RFC 950**)
- La máscara de una Red se denota así: 223.1.1.0/24
  - —› /24 mascara de red
  - Indica que los /X bits más a la izquierda forman la subred

Ejemplo: Composición de la Subred

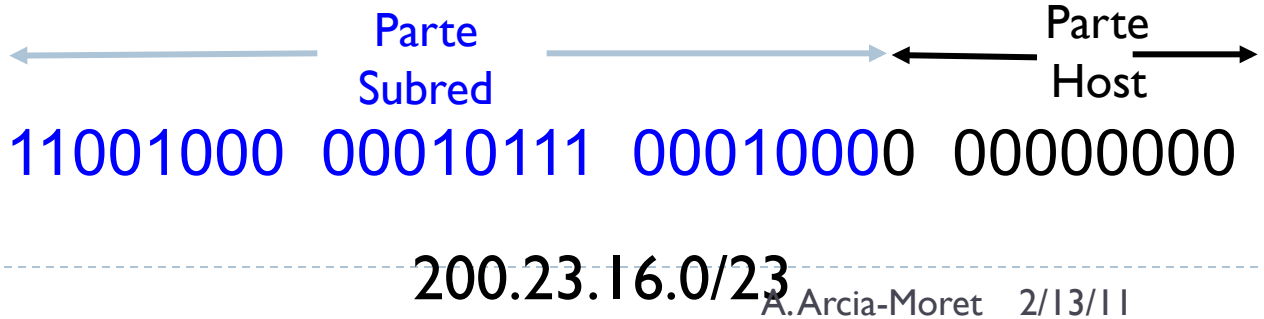
- Cada ruteador tiene 3 interfaces:
  - 2 para cada conexión punto a punto
  - 1 para la red interna
- ¿Cuantas subredes entonces?
  - 3 externas (subred local)
  - 3 internas (punto a punto)
- Regla para contar subredes
  - Desconecte los cables de cada **nodo** y **enrutador** y cree islas de redes aisladas. Cada una de estas redes "es" una subred.

Ejemplo 2


223.1.1.2
223.1.1.1
223.1.1.4
223.1.1.3
223.1.9.2
223.1.7.0
223.1.9.1
223.1.7.1
223.1.8.1
223.1.8.0
223.1.2.6
223.1.3.27
223.1.2.1
223.1.2.2
223.1.3.1
223.1.3.2

Enrutamiento: Classless Interdomain Routing (CIDR).

- Dividir el dominio en 2 partes: fijo y variable
  - Ej: a.b.c.d/x (x: número de bits del prefijo)
- Se asigna por lo general una porción contigua de direcciones.
- Ejemplo:
  - Organización de 200.13.16.0/20
  - Dentro puede haber 8 organizaciones —› agregación de direcciones
  - ISP más pequeño que anuncia 200.23.18.0/23
  - Regla del prefijo más largo para enrutar



Enrutamiento: Classless InterDomain Routing (CIDR)

- Antes se consideraban redes de 8, 16, 24 bits para hacer las distintas clases (A —› 1 byte, B —› 2 bytes, C—› 3 bytes).
- Una Clase C podría alojar 256-2 = 254 hosts
  - ¿Cómo se haría con 2000 hosts? Clase B (65535 es mucho)
  - Un mensaje a 255.255.255.255 —› broadcast
  - El primer IP del conjunto se usa como enrutador
- Observe que en las tablas de enrutamiento solamente se necesita una entrada (la del prefijo) para hacer forwarding hacia la red interna.

Desaparición de las Redes de Clase A

Ver animación GIF:

http://www.zone-internet.ch/2010/10/06/visualisez-la-disparition-des-adresses-ipv4/

¿Cómo Obtener una dirección IP?

- Pídala a su administrador y colóquela en el sitio apropiado del OS.
  - Windows : control-panel->network->configuration->tcp/ip->properties
  - UNIX: /etc/rc.config
- ¿Cómo obtener bloques de IP?
  - ISPs sub arriendan bloques de IP's (como en el ejemplo)
- Globalmente manejado por el ICANN basado en RFC 2050.
  - ICANN : Internet Corporation of Assigned Names & Numbers
  - Controles regionales: ARIN (America), RIPE (Europa), APNIC (Asia), LACNIC
    - Latinamerican and Caribean Internet Addresses Registry.

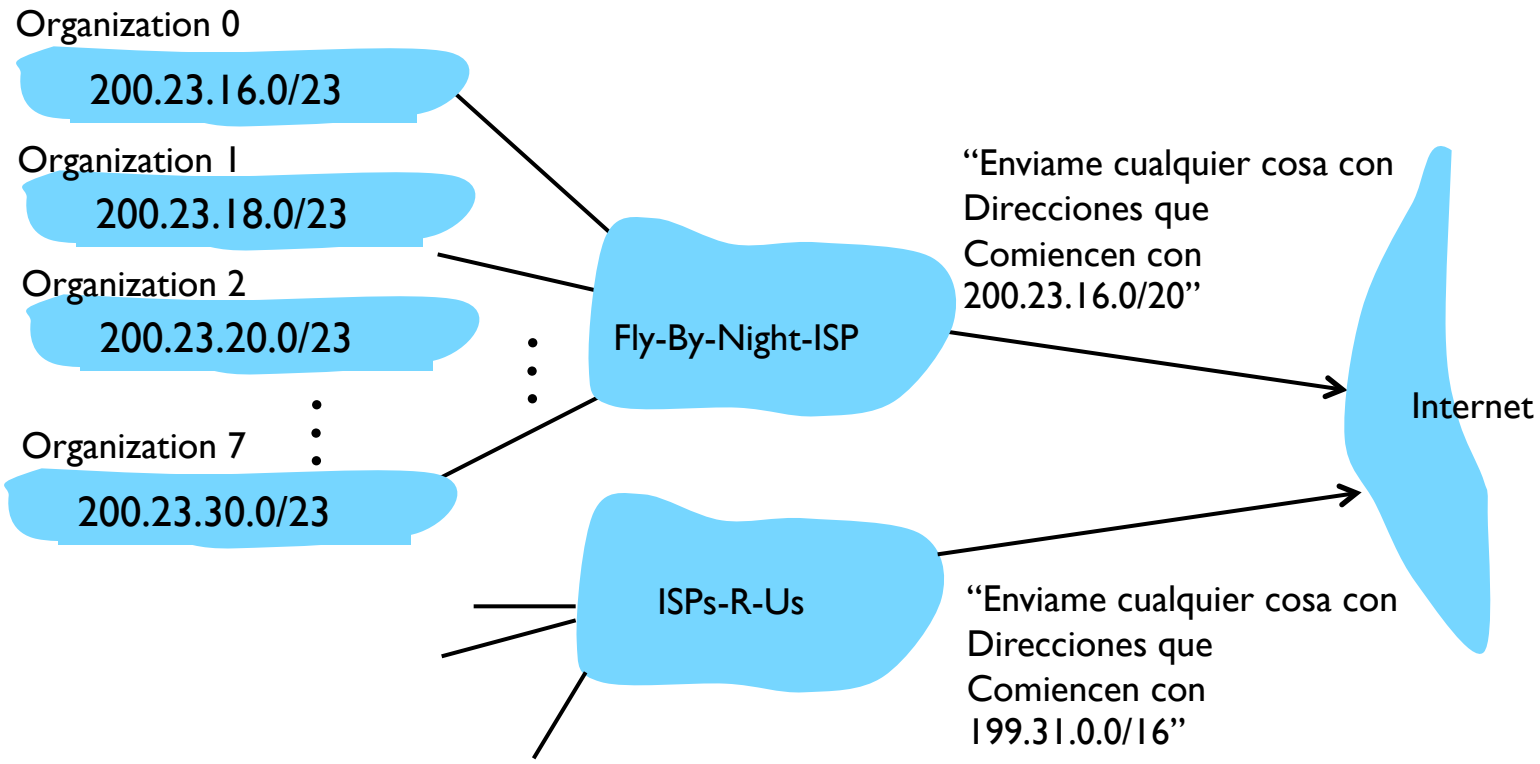
Pero, ¿Cómo obtengo un IP?

- Se obtiene a una porción del conjunto otorgado por el proveedor.

Bloque del ISP's	11001000 00010111 00010000	00000000	200.23.16.0/20
Organización 0	11001000 00010111 00010000	00000000	200.23.16.0/23
Organización 1	11001000 00010111 00010010	00000000	200.23.18.0/23
Organización 2	11001000 00010111 00010100	00000000	200.23.20.0/23
...	.....	....	....
Organización 7	11001000 00010111 00011110	00000000	200.23.30.0/23

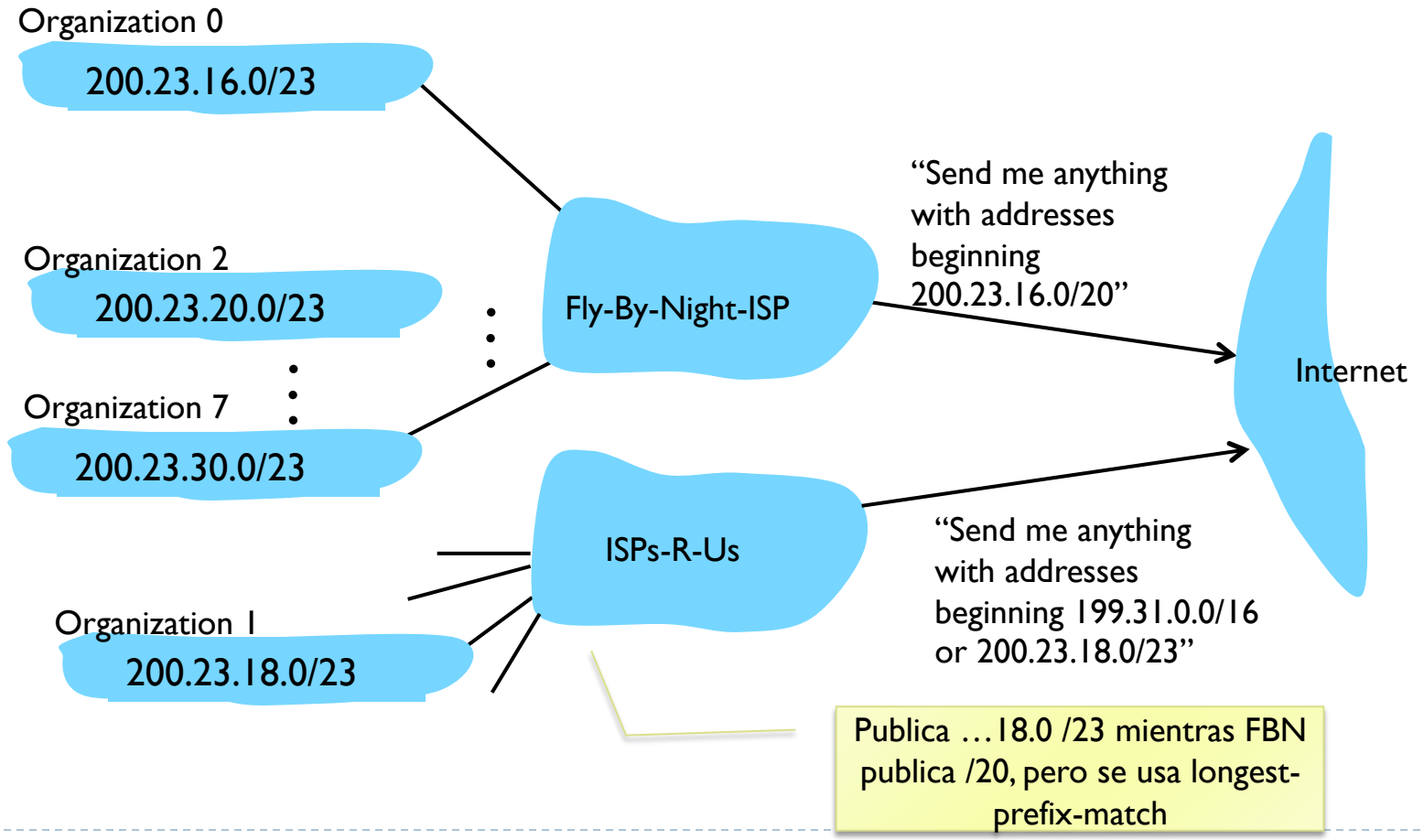
Pero, ¿Cómo obtengo un IP?

- El enrutamiento jerárquico permite una diseminación de la información de ruteo de forma eficiente.



Pero, ¿Cómo obtengo un IP?

ISPs-R-Us ofrece una ruta específica hacia la **Organización 1**

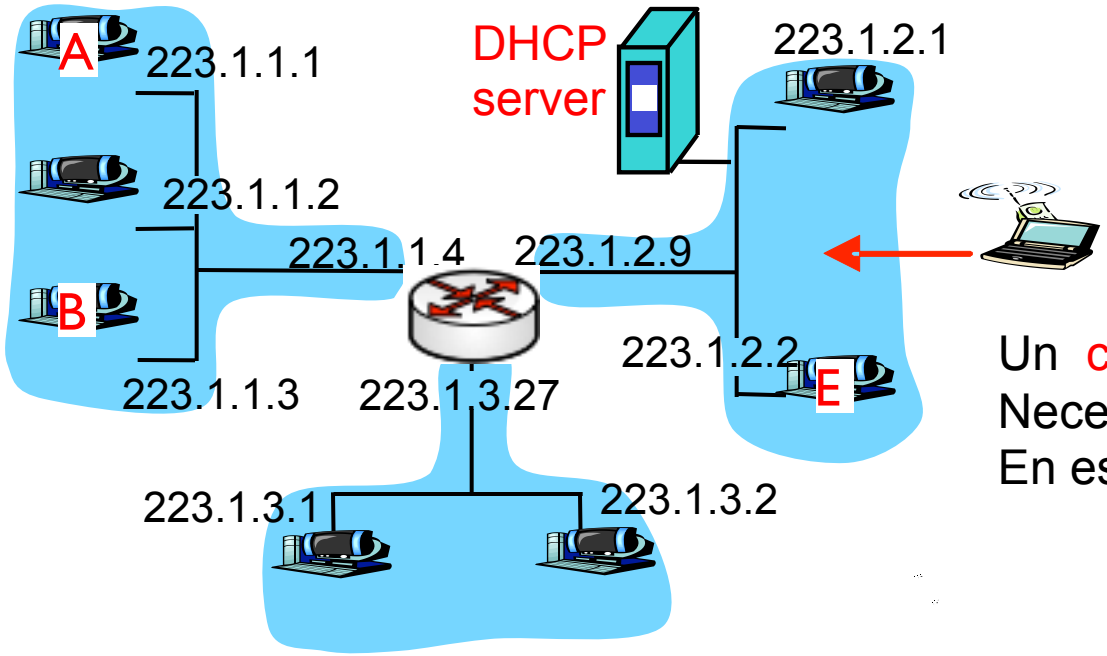


DHCP: Dynamic Host Configuration Protocol

Dynamic Host Configuration Protocol

- Especificado en el RFC 2131 (protocolo PnP)
- Da IPs de forma automática
- Se puede configurar para tener siempre el mismo IP.
- El protocolo otorga: **mascara, DNS, gateway**
- Se usa para lidiar con la dinamica de usuarios y el número de ellos.
  - En una red residencial de 2000 usuarios hay 400 activos.
- Se actualiza la tabla cada vez que un cliente entra (pide) o sale (devuelve) de la red.
- Cada subred tiene un servidor DHCP.

Escenario de un Cliente DHCP que llega

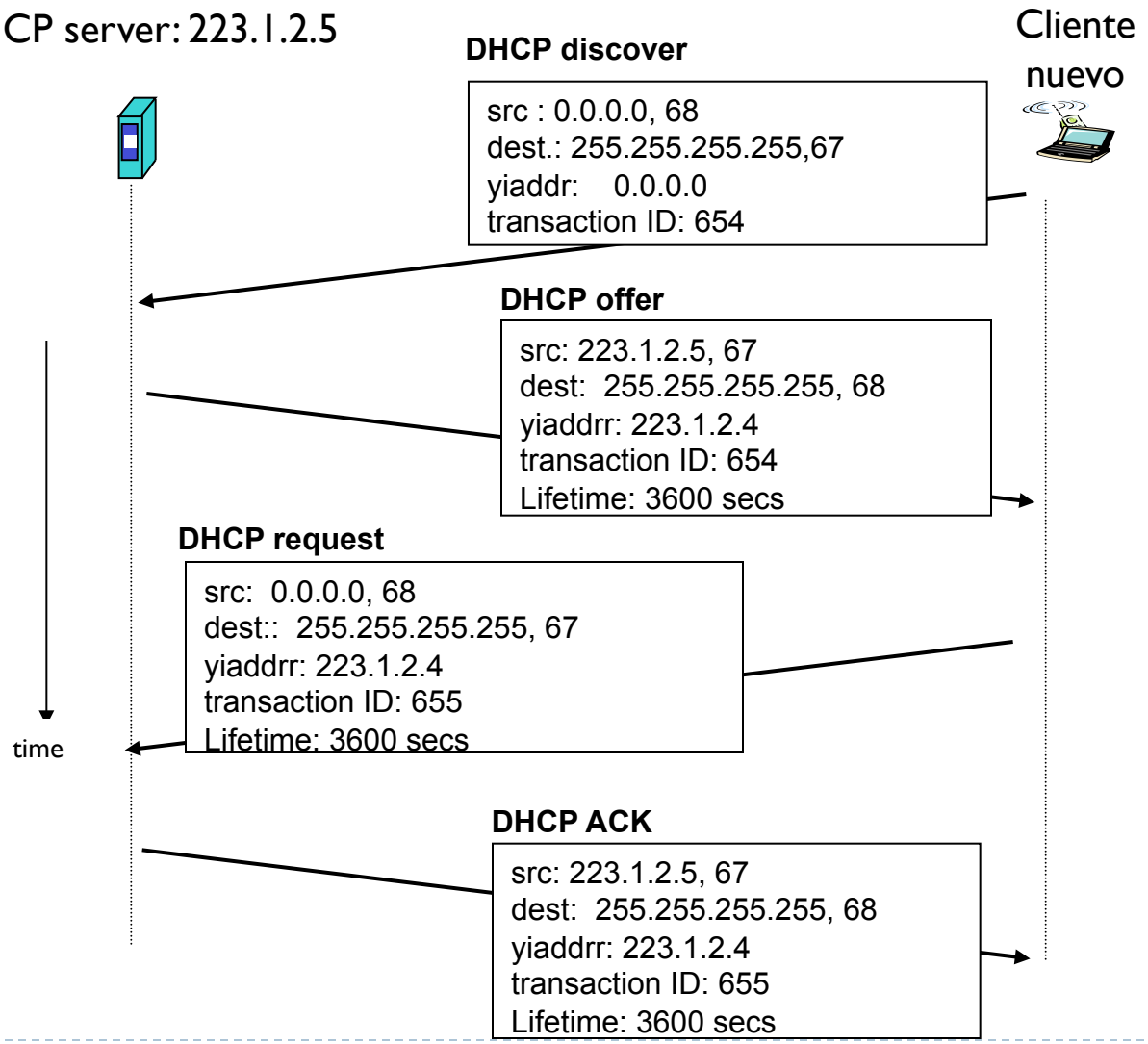


Un cliente DHCP
Necesita una dirección
En esta subred

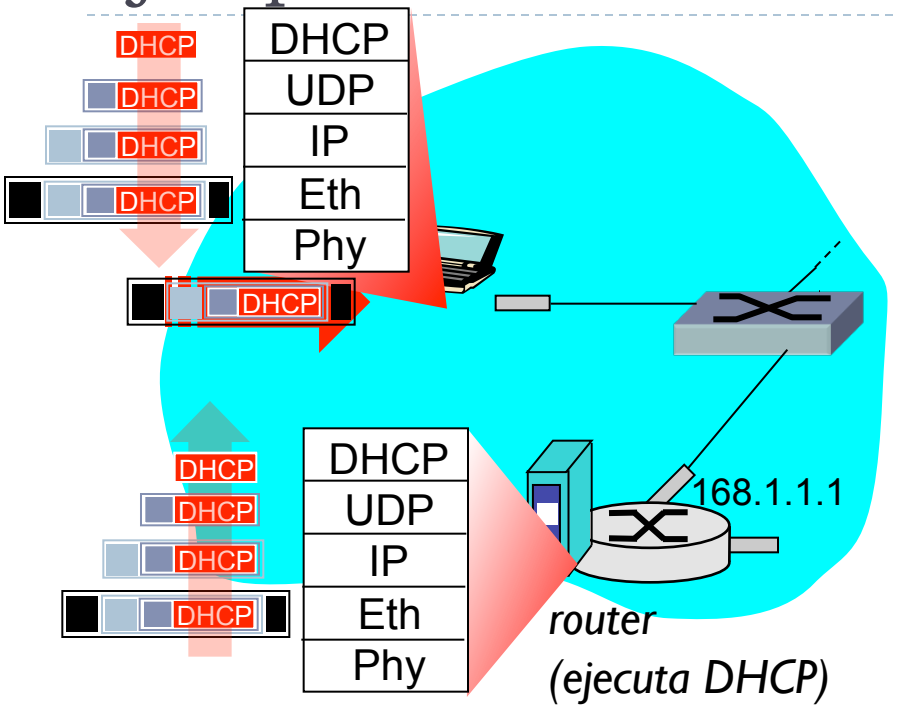
¿Cómo conectarse a una red?

1) DHCP Server Discovery: Envío de mensaje de descubierta paquete UDP al puerto 67.
2) DHCP server offer(s): envía en mensaje broadcast con 255.255.255.255 con su dirección IP.
    1) ID de la transacción
    2) IP propuesto para el cliente
    3) Mascara de red
    4) Tiempo del préstamo de la dirección
3) DHCP request: se envía pidiendo formalmente el ingreso a la red.
4) DHCP ACK: confirma los parámetros de la red.

Escenario a nivel protocolar



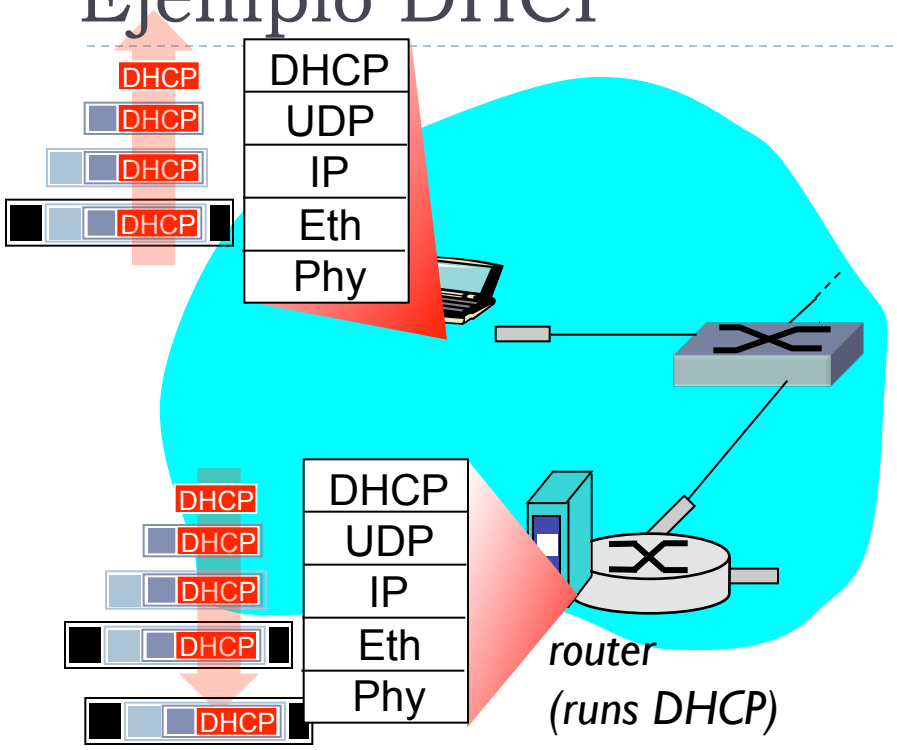
Ejemplo DHCP



- Necesidades del cliente:
- IP address, dir. del primer hop router, dir. DNS
  - usar DHCP

- Solicitud DHCP encapsulada en UDP, encapsulada en IP, encapsulada en in 802.1 Ethernet
- Broadcast de trama ethernet (dest: FFFFFFFFFFFF) en LAN, recibida por el ruteador que ejecuta servidor DHCP
- Demultiplexado Ethernet hacia IP, demulitplexado IP hacia UDP a su vez demultiplexado va hacia DHCP

Ejemplo DHCP



- Servidor DHCP formula DHCP ACK que contiene: dir. IP address, dir. IP del primer-hop router del cliente, nombre y dir. IP del servidor DNS.

- Encapsulado del servidor DHCP, trama llevada al cliente, demultiplexado al cliente DHCP
- El cliente conoce todos los parámetros de conexión a la red.

Traza DHCP por Wireshark

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 cliente

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."

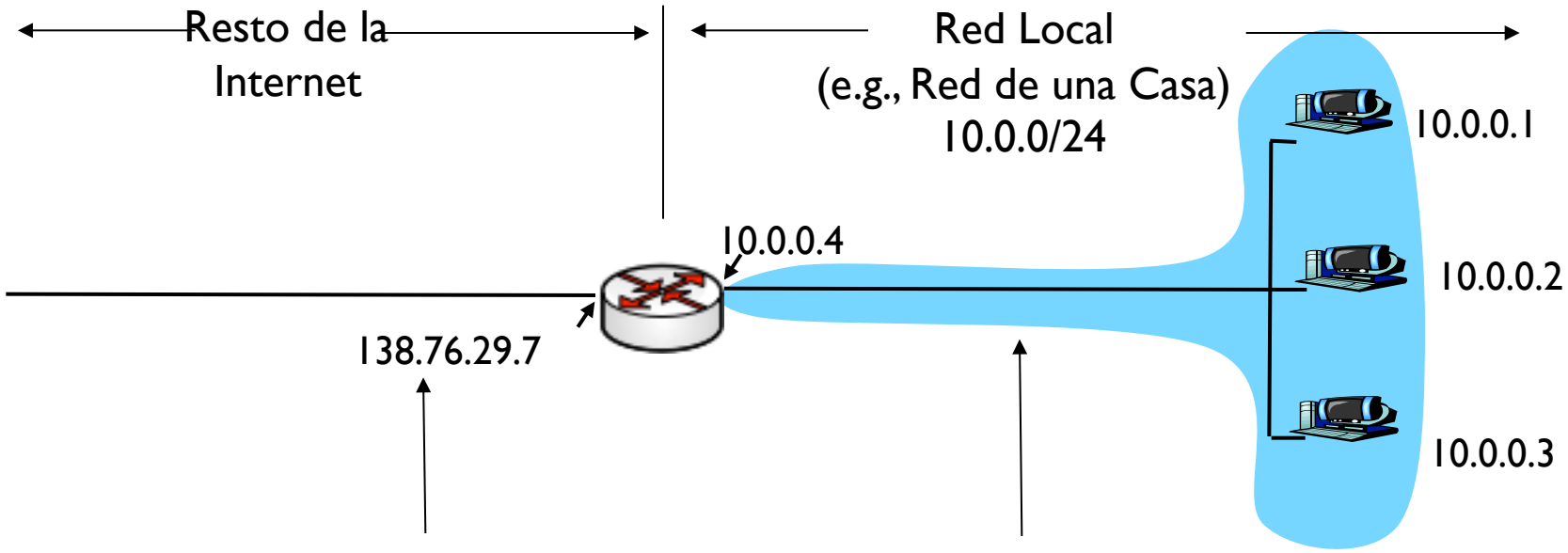
Respuesta Srv DHCP

NAT: Network Address Translation

NAT: Network Address Translation

- Hoy en día se necesitan muchisimas direcciones para conectar "todos" los dispositivos de red: PDA, consolas video, laptops, etc.
- Motivado por SOHO (Small Office Home Office) networks.
- Como las direcciones IPv4 no alcanzan se usa RFC 2663 y 3022 (NATs)
- Hay un espacio de direcciones IP reservado para ello: 10.0.0.0/8, 192.168.0.0/16 son dos de ellos (RFC 1918).
  - Es necesario para tener repetidos los mismos IP en diferentes subredes.
- Operación simple: Cuando un paquete llega al NAT, hay una tabla que traduce la dirección al host final.

NAT esquematizado



TODOS los datagramas que dejan el NAT tienen el mismo IP: 138.76.29.7, pero puertos diferentes.

Datagramas con fuente destino esta Red tienen direcciones10.0.0/24

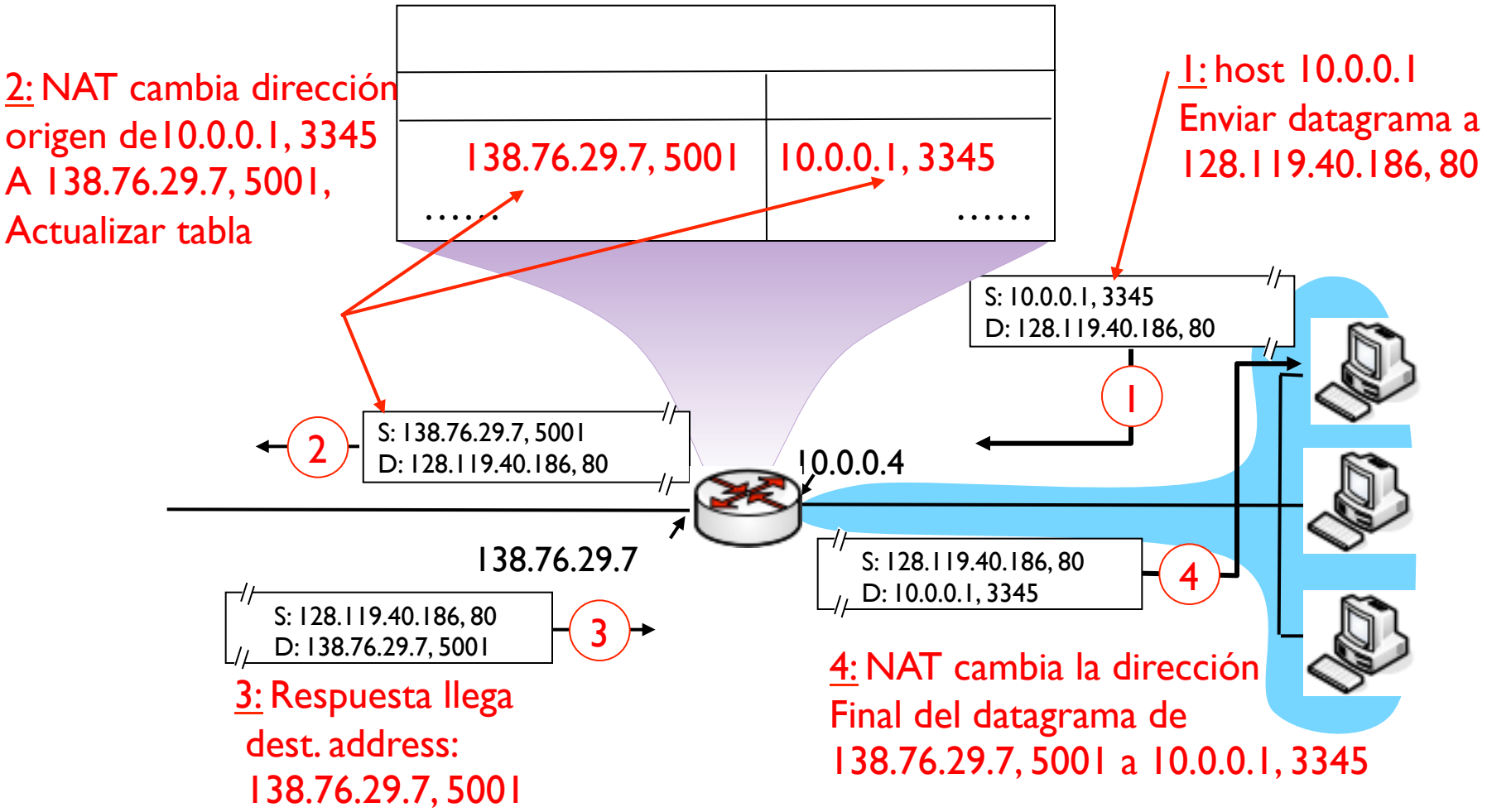
NAT

- Motivación: La red local utiliza una sola dirección IP (desde la perspectiva del mundo exterior):
  - Rango de direcciones no son necesarias saberlas para el ISP: solamente 1 es necesaria.
  - Se pueden cambiar las direcciones IP locales sin advertir al mundo exterior
  - Se puede cambiar de ISP sin cambiar de direcciones
  - Los dispositivos locales no son directamente direccionables (bueno para la seguridad).

NAT

- ¿Qué debe hacer un enrutador NAT?
  - Datagramas de salida: reemplazar la direccion IP/puerto del paquete por el del NAT.
  - Registro: deben almacenarse los pares <IP+puerto, NAT_IP +nuevo puerto>
  - Datagramas de entrada: hacer operación reversa <NAT_IP +nuevo puerto> por el IP interno y puerto encontrado en la tabla.

NAT: Resumen

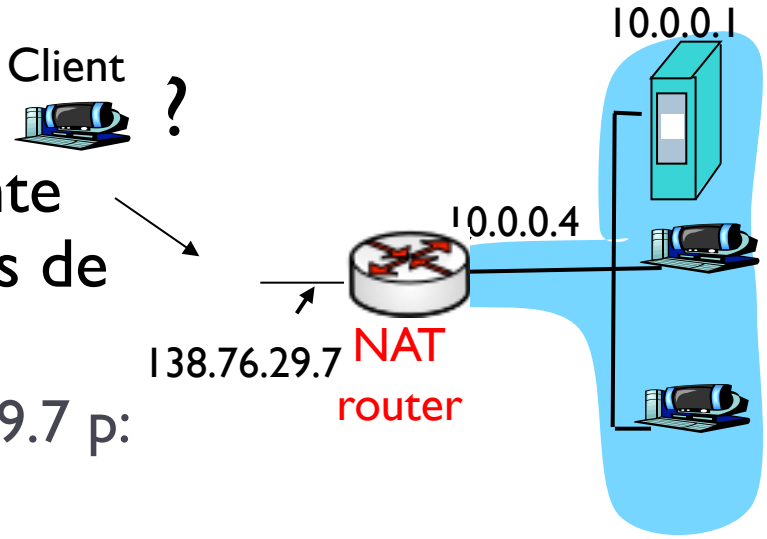

2: NAT cambia dirección origen de10.0.0.1, 3345 A 138.76.29.7, 5001, Actualizar tabla
138.76.29.7, 5001
10.0.0.1, 3345
1: host 10.0.0.1 Enviar datagrama a 128.119.40.186, 80
S: 10.0.0.1, 3345
D: 128.119.40.186, 80
1
2
S: 138.76.29.7, 5001
D: 128.119.40.186, 80
10.0.0.4
138.76.29.7
S: 128.119.40.186, 80
D: 10.0.0.1, 3345
4
S: 128.119.40.186, 80
D: 138.76.29.7, 5001
3
3: Respuesta llega dest. address: 138.76.29.7, 5001
4: NAT cambia la dirección Final del datagrama de 138.76.29.7, 5001 a 10.0.0.1, 3345

¿Cómo se hace el cambio en el NAT?

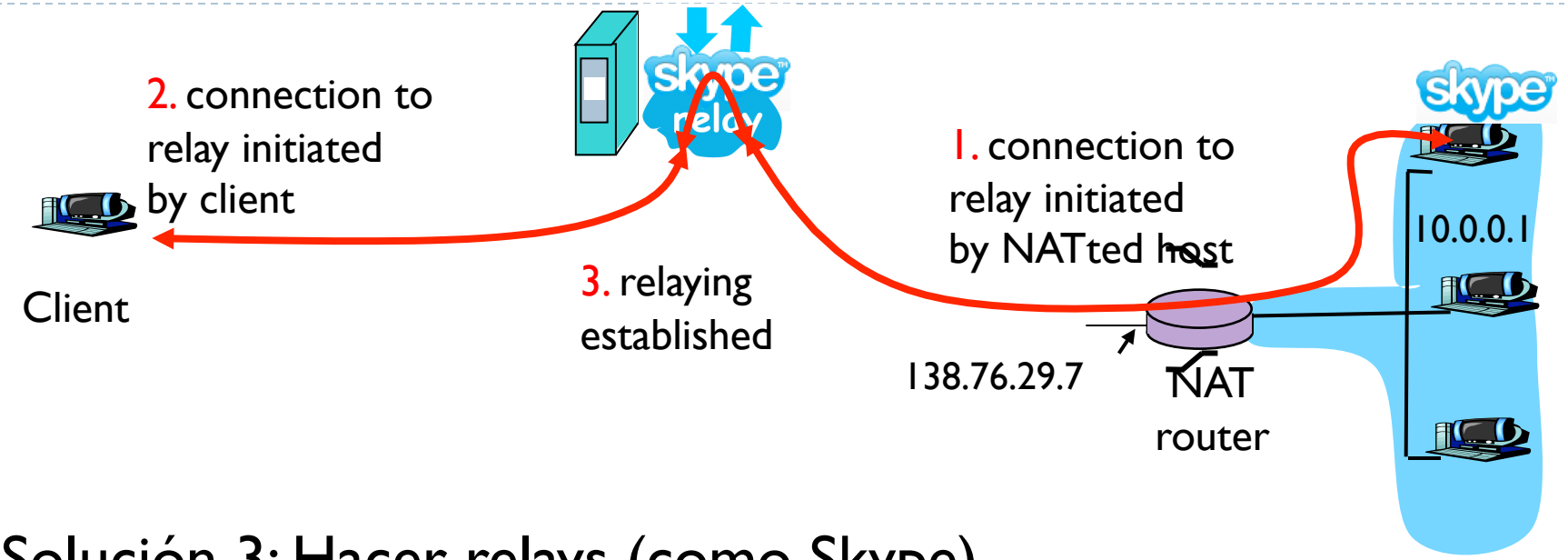
- Hasta 60000 conexiones simultaneas desde el NAT pues se tienen puertos de 16-bits.
- Costos (controversia; puristas del IETF):
  - Se viola el principio e2e.
  - Los puntos intermedios deberían procesar solo hasta capa 3.
  - Puertos son para direccionar procesor y NO hosts.
  - Debería usarse ya el IPv6
  - Interfiere con las aplicaciones P2P
    - Dificilmente se puede actuar como servidor detras de un NAT.
      - Si 1 servidor detras de un NAT —› connection reversed
      - Si los 2 detras de NAT —› aplication relays como Skype.

Problema de Acceso al NAT

- Problema: Se quiere acceso al 10.0.0.1 pero solamente se puede a la interfaz externa: 138.76.29.7
  - 10.0.0.1 es local
- Solución 1: Configurar estaticamente el NAT para que se acceda a través de un puerto del NAT-router.
  - Ej: todos los paquetes hacia 138.76.29.7 p: 5020 hacia 10.0.0.1 p: 1204
- Solución 2: UPnP para aprender direcciones públicas y registrar puertos con tiempos de expiración.



Problema de Acceso al NAT



- Solución 3: Hacer relays (como Skype)
  - Cliente NAT establece conexión con el Relay
  - Cliente externo contacta al Relay
  - Relay hace el puente entre los dos

UPnP

- Si PnP corre en un host local, se puede pedir una dirección **pública** desde una **privada**.
- Nodos externos pueden pedir una conexión "TCP" a un nodo interno a través de un llamado, "hueco"
  - Ej: BitTorrent corriendo en dirección privada (10.0.0.1) y puerto 3345
- NAT hace forwarding de los paquetes de data y control
- NATs han sido una gran dificultad para aplicaciones P2P y para protocolos como SCTP.

Internet Control Message Protocol

Internet Control Message Protocol (ICMP)

- Protocolo de excepciones
- Uno de los tres grandes componente de la capa Red (+IP + Routing)
- Mensajes sobre el estado de la capa Red.
- Está encima de IP (en realidad —› ver encabezados)
- Ejemplos de Aplicaiones
  - Ping
  - Traceroute

Tabla resúmen de mensajes

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

Traceroute

- La fuente envía una serie de segmentos UDP al destino
  - El primero TTL = 1
  - El segundo TTL = 2
  - Se envía número de puerto no utilizado
- Cuando el N-esimo datagrama llega al N-esimo router
  - Enrutador descarta el datagrama
  - Se envia un msg ICMP (tipo 11, codigo 0): TTL expiro
  - Mensaje vuelta incluye nombre del enrutador & dirección IP.
- Cuando msg ICMP llega la fuente calcula el RTT.
  - Traceroute hace esto 3 veces
- ICMP para cuando obtiene un Tipo 3, codigo 3: puerto inalcanzable

IPv6

Nueva generación del protocolo IP

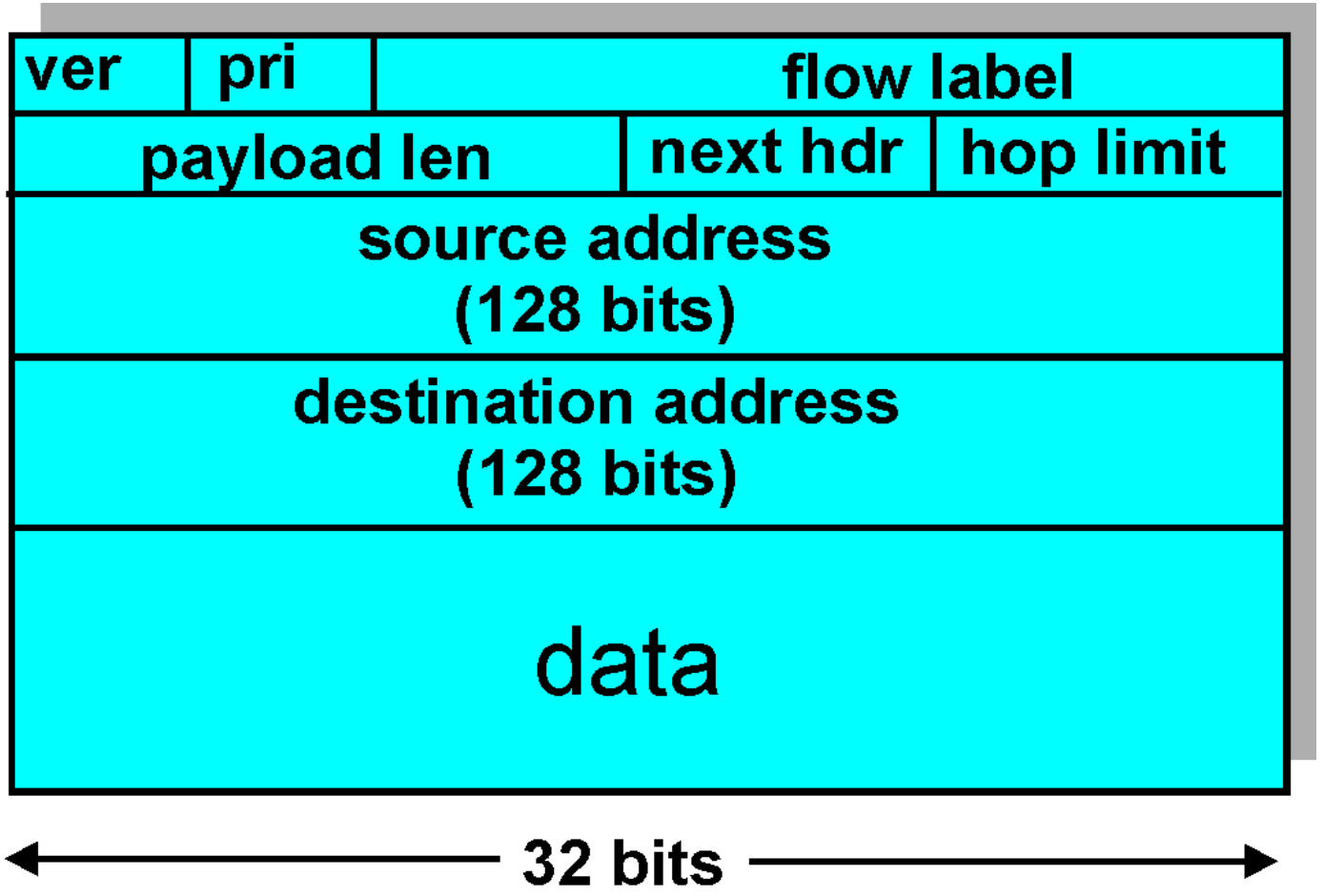
IPv6

- Mejora en escala y basado en experiencia operacional: 32 bits es insuficiente.
- Cálculos originales decian que las direcciones IP se acabarían en 2008 y 2018. Calculos más realistas dicen que es este año.
  - Recuerde: Ud. vió Redes durante el año del juicio
  - http://www.potaroo.net/tools/ipv4/
- Asignaciones actuales de las redes: Clase A (100%), Clase B (62%) y Clase C (37%)

Características

- Capacidad de direccionamiento extendida
  - IP para cada grano de arena en el planeta.
  - Añade *anycast* para enviar mensajes a todos los miembros de un grupo.
  - **Ej**: Un get para un conjunto de hosts (para que responda el más cercano).
- Encabezado de 40 bytes para procesamiento rápido.
- Nueva codificación de opciones más eficiente.
- Etiquetado de flujo y prioridad
  - Voz y audio son flujos, Mail y FTP **no**.
  - Prioridad a los paquetes de control.

Encabezado IPv6



Los Campos de IPv6

- **Version**: versión del protocolo.
- **Clase**: Tipo de servicio IPv4, IPv6.
- **Flow ID**: Etiquetado del flujo.
- **Longitud de la Carga**: Cuanto pesa la carga (sin header).
- **Next Header**: Apuntador a opciones
- **Hop Limit**: Como TTL.

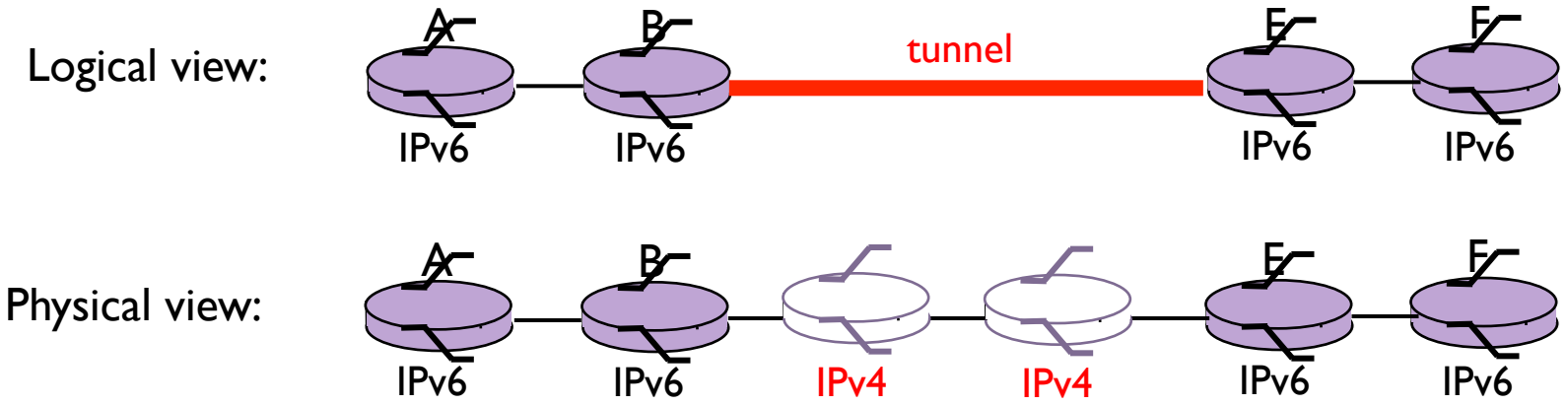
Diferencias entre IPv4 e IPv6

- **Fragmentación/Reensamblaje**: No permitidos en IPv6. Acelera el protocolo.
  - Si no se puede enviar paquete se recibe un mensaje "*Packet too big*".
- **Checksum del encabezado**: aplicando el principio e2e, se removió esta funcionalidad.
- **Opciones**: Se hace con un puntero al próximo encabezado para acelerar el procesamiento.
- Lleva consigo un cambio en ICMP —› ICMPv6: "packet too big".

Transición IPv4 a IPv6

- Hacer un "**flag day**": Todas las maquinas se reinician en IPv6.
  - Ya hubo uno para hacer la transición de NCP a TCP. No sirvió aun cuando la escala era muchisimo menor.
- Hacer el cambio **gradualmente**.
  - Se crea un tunel poniendo el paquete entero IPv6 en un IPv4.
  - A partir de 2008 es mandatorio en los routers backbone de los USs sean capaces de entender IPv6.
- Implementar un nodo "**dual stack**" IPv4/IPv6.
  - ¿Cómo determinar si un nodo es v4 o v6? R: DNS.
- Conclusión: muy **lento** de implementar, ¡15 años según los expertos!

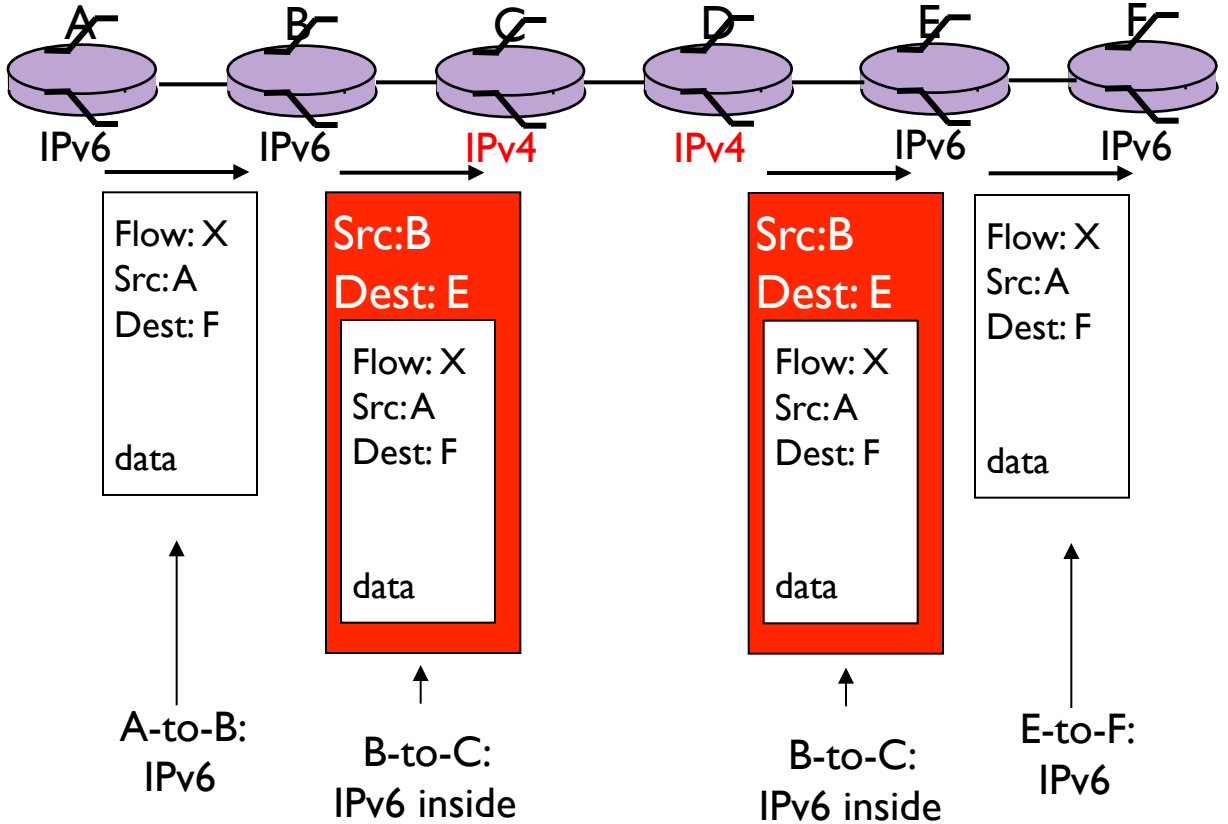
Tunel


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

Tunel

Logical view:

Physical view:



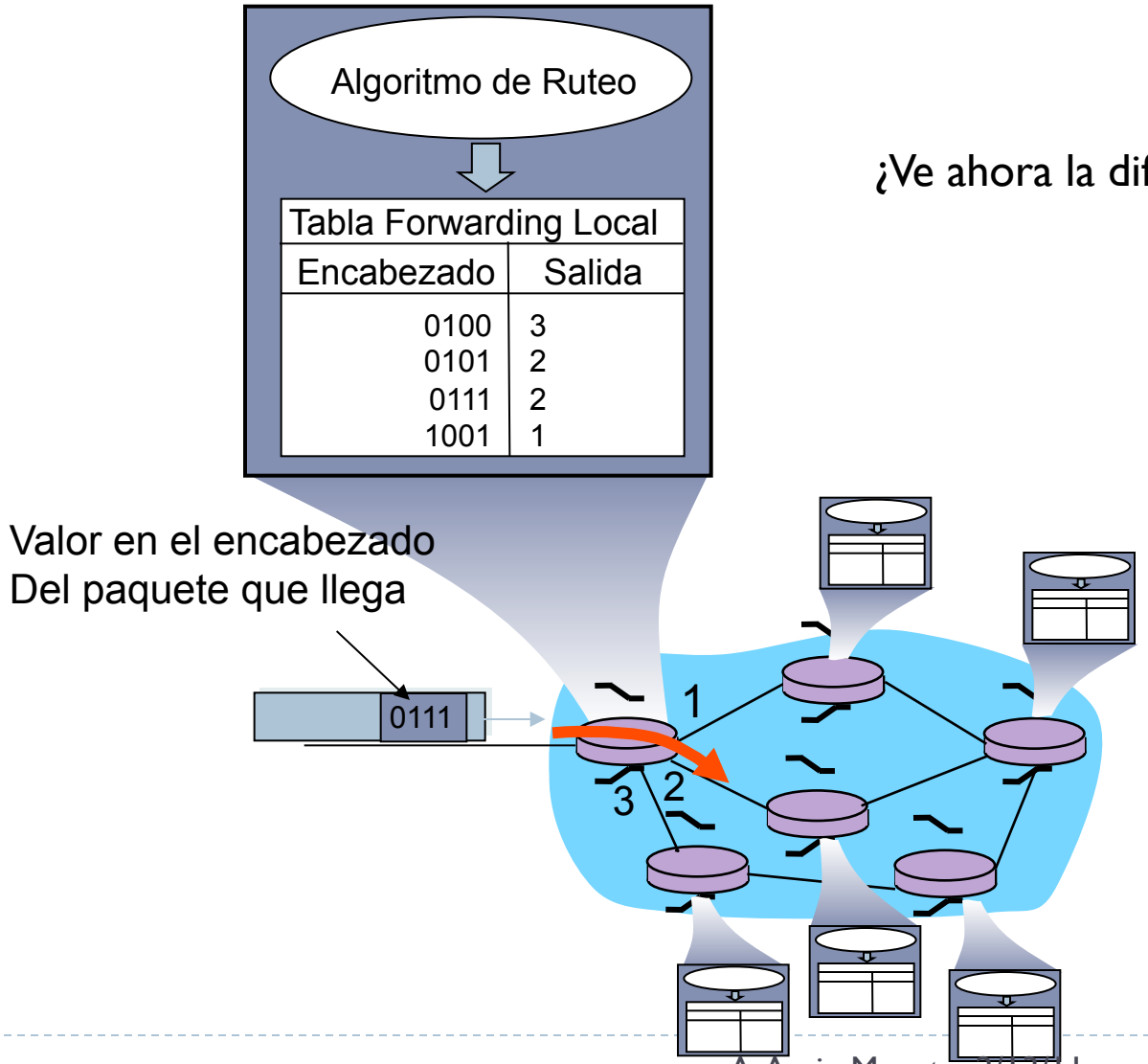
Algoritmos de Enrutamiento

Casos de Estudio

Algoritmos de Ruteo

- El rol del ruteo es buscar buenos *caminos o rutas* entre un par de nodos (emisor —› receptor) dentro de la red de enrutadores.
- Primer enrutador siempre es por omisión.
- Dos enrutadores importantes:
  - *Source Router:* Router por omisión en el emisor.
  - *Destination Router:* Router por omisión en el receptor.
- Problema: ¿Cómo enrutar de la fuente al destino?
- Respuesta General: Un buen camino es aquel de mínimo costo.
  - **No hay** solución única
  - El Ruteo entre proveedores (ISP) no es óptimo.

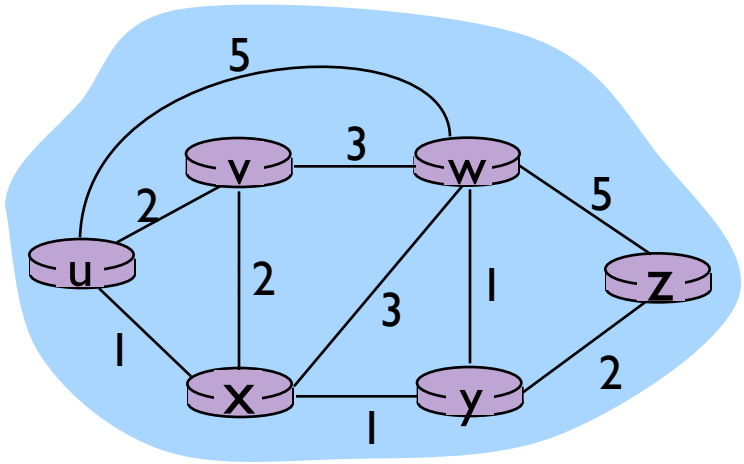
Ruteo versus Forwarding



¿Ve ahora la diferencia?

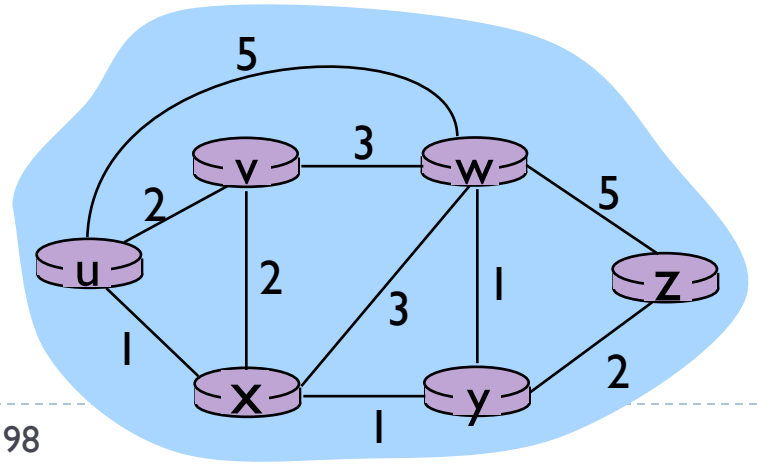
Simplificación del Problema (1)

- Grafo: colección de nodos N unidos por un grupo de enlaces E.
  - N = { u, v, w, x, y, z }
  - E = { (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }
- Se utilizan grafos **no** dirigidos
- Router decide el próximo salto



Simplificación del Problema (2)

- Costos: reflejan la longitud física del enlace y/o el tráfico que circula. También puede representar "conexión" si *c=1*.
  - Enlace trasatlántico no tiene el mismo costo que uno local.
  - El costo se denota: c(x,x') es el costo del enlace (x,x')
  - Si no hay conexión entre dos hosts —› c(x,x') = ∞
- Costo de un camino $(x_1, x_2, x_3, \ldots, x_p)$
  - $C = c(x_1,x_2) + c(x_2,x_3) + \ldots + c(x_{p-1},x_p)$



¿Cuál es el costo mínimo entre u y z?

Reflexiones

- Note que si todos los nodos tienen el mismo costo, el camino de menor costo es el mas corto también.
- Si determinó el camino más corto en la lamina anterior:
  - ¿Cómo lo hizo?
  - ¿Probó las 17 rutas posibles?
  - Es un ejemplo de enrutamiento centralizado:
    - toda la información la tiene usted.

Tipos de Algoritmo

- Algoritmo Global
  - Se usa el conocimiento de toda la topología de la red
  - Se les llama algoritmos "link state" (LS)
- Descentralizado
  - Se calcula de forma iterativa y distribuida.
  - Los nodos no tienen información completa de la red.
  - Se conocen como "distance vector" (DV), cada nodo contiene un vector de costos aproximados de sus vecinos.
  - Responde a cambios en la red
- Estáticos
  - Cambios lentos en el tiempo
- Dinámicos
  - Cambios más rápidos, actualizaciones periódicas, responden a los cambios en costos.

Algoritmo Link State (LS)

- Todos los costos y la topología están disponibles
  - Hay mensajes broadcast que comunican los estados de los enlaces.
  - Todos los nodos tienen la misma información

Algoritmo de Dijkstra

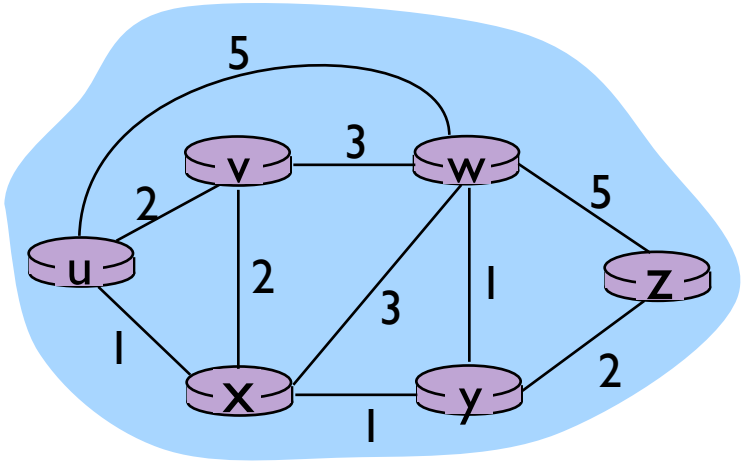
- Se computa el camino más corto de un nodo (origen) a todos los otros nodos.
  - Calcula la tabla de forwarding para dicho nodo.
- Propiedad fundamental: Luego de K iteraciones, se sabe la distancia mínima a K nodos destino.
- Notación
  - c(x,y): costo del enlace x a y; = ∞ si no hay camino
  - D(v): costo "actual" del camino mínimo del **origen** al destino "v" (en la iteración presente).
  - p(v): nodo previo (vecino) de v en la ruta critica.
  - N': subconjunto de nodos cuyo camino más corto es conocido

¿Cómo funciona?

```
1  Inicio:
2    N' = {u}  // nodo origen
3    Para todo nodo v
4      SI v es vecino de u
5          ENTONCES D(v) = c(u,v)
6      SI NO D(v) = ∞
7
8  Repita
9     Encontrar w que no esté N' tal que D(w) es mínimo
10    Añadir w a N'
11    Acutalizar D(v) para todo v vecino de w y no esté en N' :
12       D(v) = min( D(v), D(w) + c(w,v) )
13    /*el nuevo costo a v puede ser el viejo costo v o el
14     Camino más corto a w más el costo de w a v */
15 Hasta que todos los nodos esten en N'
```

Ejemplo

Step	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					



El nuevo costo **V'** puede ser:

* viejo costo **V**
* camino más corto a **W** + costo de **W** a **V**

W: nodo nuevo en **N'**

V: nodo vecino de W

Grafo Resultante de Caminos Cortos

Arbol de caminos más cortos resultante:

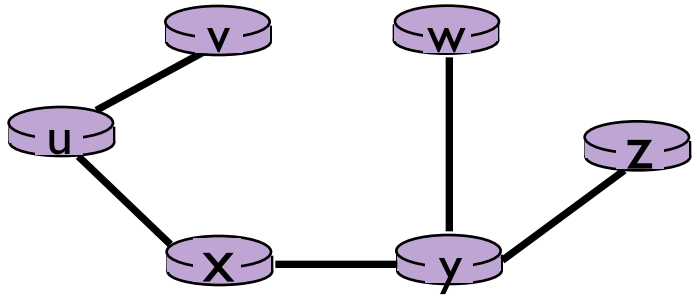


Tabla de forwarding en **u**

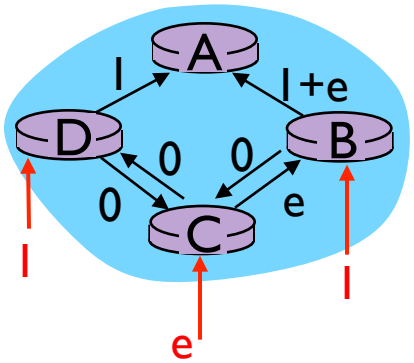
destino	enlace
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

Discusión

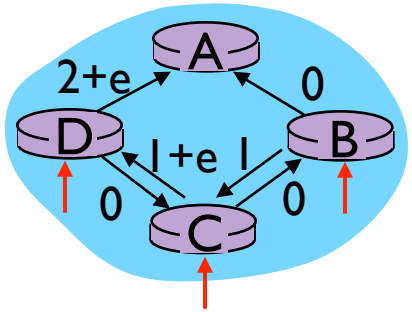
- Complejidad de Algoritmo: N nodos.
  - Cada iteración se chequean todos los nodos que no están en N'
  - Se comparan potencialmente: N nodos en la primera iteración, N-1 nodos en la segunda … n(n+1)/2 en total!.
  - Complejidad $O(N^2)$
    - La más eficiente en O(nlogn) usando un heap para la búsqueda del mínimo D(w)

Caso Patológico: Oscilaciones

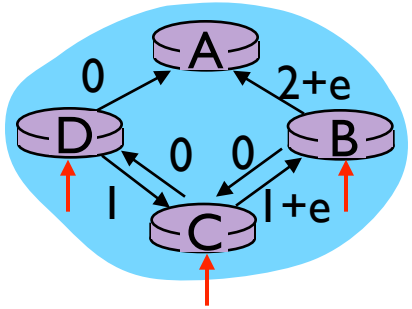
- Si los costos de los enlaces corresponden al tráfico:



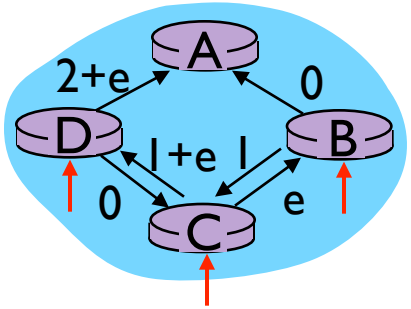
INICIO



… Recalcular la ruta



… Recalcular



… Recalcular

Caso Patológico: Oscilaciones

- Costo del enlace = delay del tráfico
- Costos en enlaces no simétricos: $c(u,v) \neq c(v,u)$
- Se encuentra siempre un mejor camino pues se hace oscilar la carga.

Solución a las oscilaciones

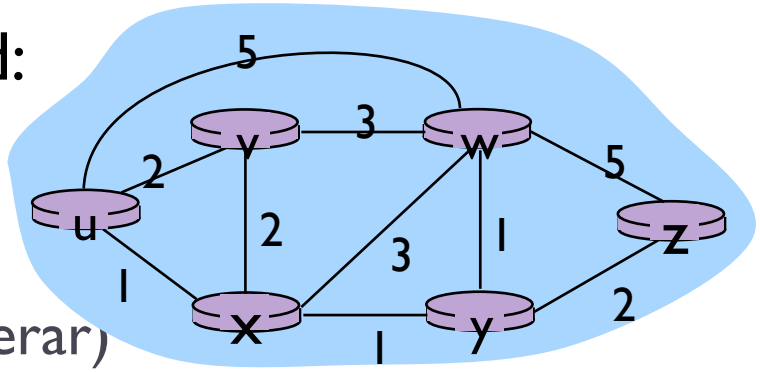
- No depender de la cantidad de tráfico (poco realista).
- No todos los ruteadores corren el algoritmo en un mismo momento.
  - Ruteadores se pueden sincronizar —› aleatorizar las advertencias del enrutador.

Algoritmo Distance Vector

- Distribuido
  - Recibe información de los vecinos. Recalcula distancia mínima y comparte el cálculo con sus vecinos (si aplica).
- Iterativo
  - Continúa hasta no haber más información para intercambiar. Se autolimita —› no hay señal de parada.
- Asíncrono
  - Cada nodo funciona independientemente. Puede intercambiarse información de forma simultanea.

Algoritmo Distance Vector

- Sea $d_x(y)$ el costo del camino mínimo entre **x** e **y**
- Se usa la ecuación de Bellman-Ford:
  - $d_x(y) = \min_v \{ c(x,v) + d_v(y) \}$
- Observe que:
  - $d_v(z)=5$, $d_x(z)=3$, $d_w(z)=3$ (luego de iterar)
- El Algoritmo B-F dice
  - $d_u(z) = \min \{ c(u,v) + d_v(z), c(u,x) + d_x(z), c(u,w) + d_w(z) \}$
  - $d_u(z) = \min \{ 2 + 5, 1 + 3, 5 + 3 \} = 4$
- Provee las entradas de la tabla de forwarding
  - Se sabe que si "x" está en el camino mínimo, los paquetes deben pasar por allí.



Algoritmo Distance Vector

- $D_x(y)$: estimado del costo de x hasta y
- Cada nodo mantiene:
  - El costo de ir a cada vecino c(x,v)
  - El vector de estimación a todos los destinos $\mathbf{D}_x = [D_x(y): y \in N]$
  - Los vectores de estimación de sus vecinos, para cada vecino **v**, x mantiene:
    - $\mathbf{D}_v = [D_v(y): y \in N]$

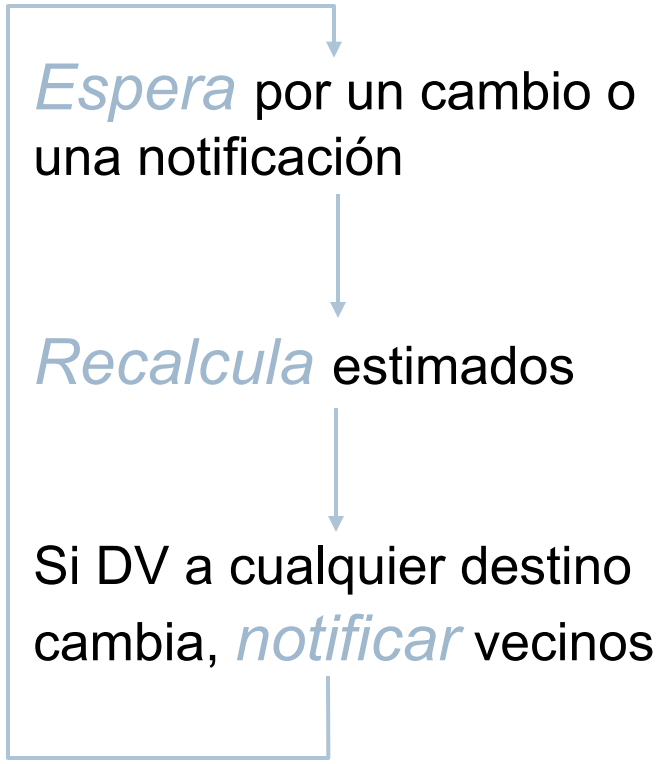
Algoritmo Distance Vector

- Idea principal:
  - Se envia una copia periódica de su vector a cada vecino.
  - Envíos asíncronos
  - Cuando se recibe una copia, se actualiza el DV utilizando la formula B-F
    - $D_x(y) = \min_v\{ c(x,v) + D_v(y) \}$ para cada nodo $y \in N$
  - Si se continúan recibiendo actualizaciones, el algoritmo converge a $d_x(y)$
- Usado en muchos protocolos: RIP, BGP, ISO IDRP, inclusive fue utilizado en ARPANet.

Algoritmo Distance Vector

- Iteraciones asíncronas
  - Cuando hay un cambio en costo
  - Luego se informa a los vecinos
- Distribuido
  - Un nodo informa a su vecino solo cuando cambia su DV
  - Notificación en cadena si es necesario

Cada Nodo:



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

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

Tabla nodo x

Costo a

desde	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

Costo a

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

Tabla nodo y

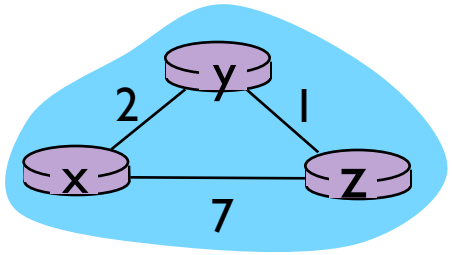
Costo a

desde	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

Tabla nodo z

Costo a

desde	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0



time

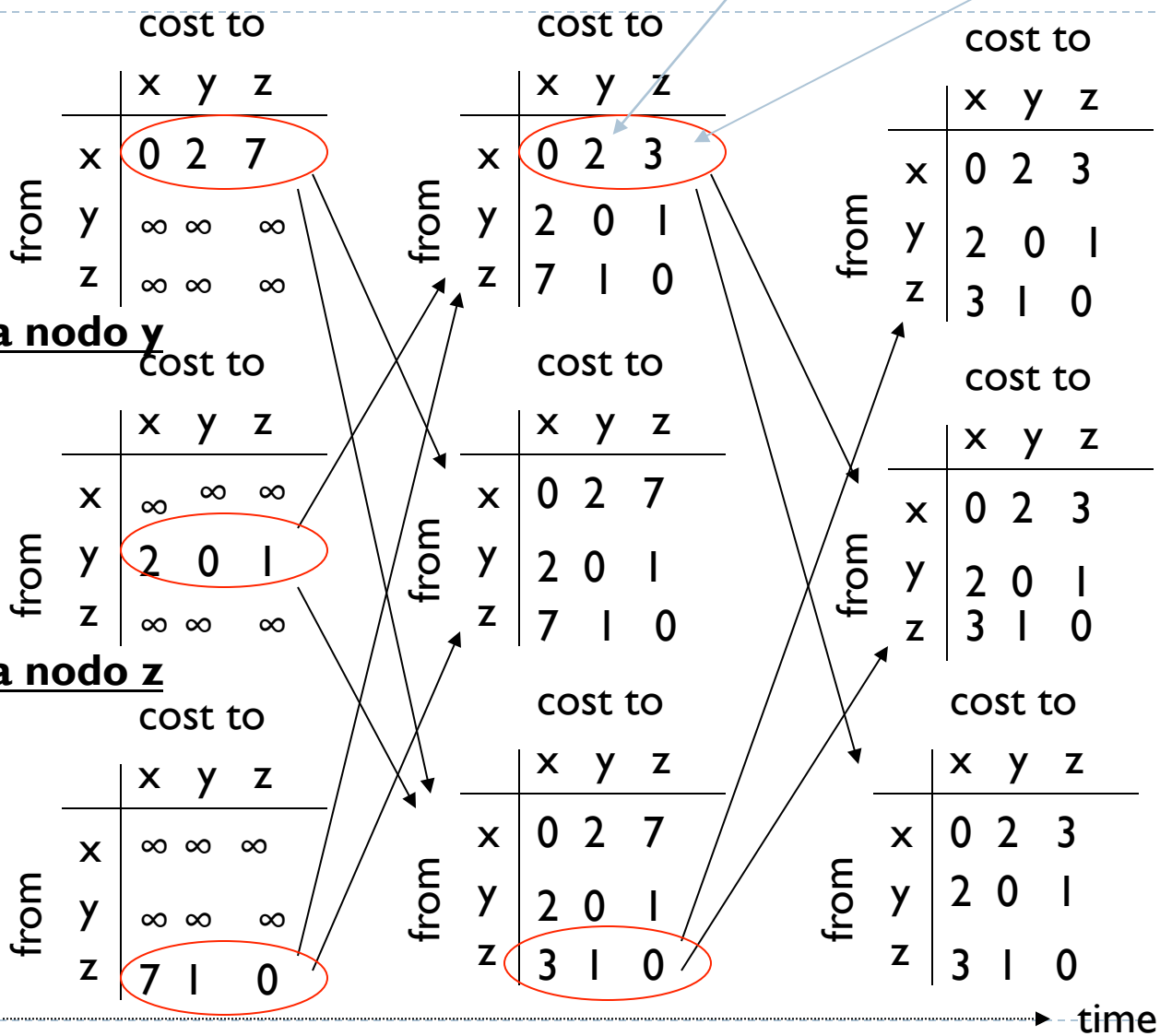
$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\}$
$= \min\{2+0 , 7+1\} = 2$

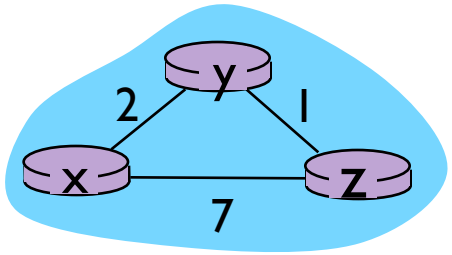
$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$
$= \min\{2+1 , 7+0\} = 3$

Tabla nodo x

Tabla nodo y

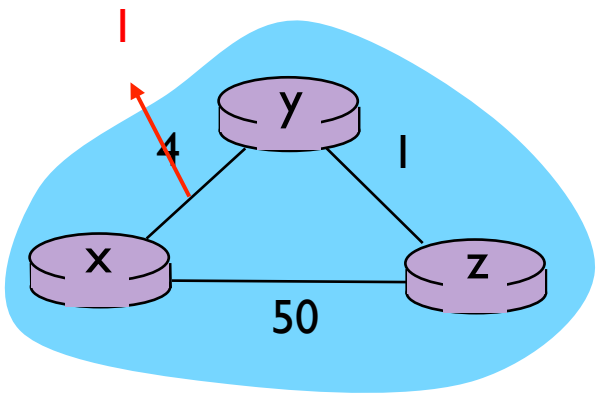
Tabla nodo z





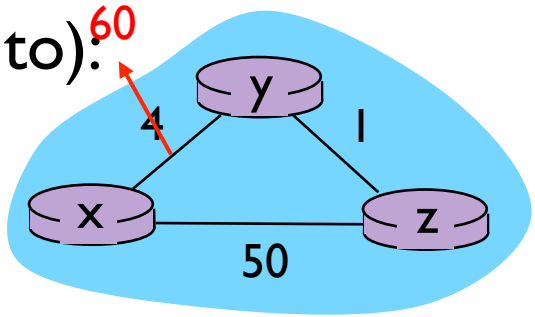
Cambio en los costos de los enlaces

- El costo decrece: (buenas noticias viajan rápido)
  - (t_0) El cambio es detectado por "y"
  - (t_1) "z" recibe una actualización de "y"…
    - Computa el costo mínimo a "x" y envía D_x a sus vecinos
  - (t_2) "z" envía su computo a "y" y éste no hace nada pues su costo mínimo **no cambia**… (termina el algoritmo)



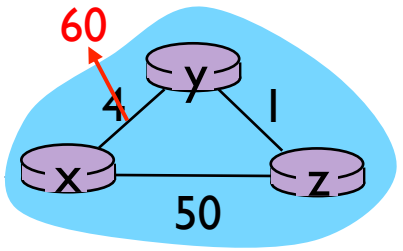
Cambio en los costos de los enlaces

- El costo aumenta (malas noticias viajan lento):
  - Problema de suma al infinito.



- Vector antes del cambio:
  - $D_y(x) = 4, D_y(z) = 1, D_z(y) = 1$ y $D_z(x) = 5$
  - Luego del cambio en el enlace: $\mathbf{D_y(x) = \min\{60 + 0, 1 + 5\} = 6}$
  - Porque "z" dijo a "y" que tiene un camino de 5 (¡Que despelote!).
  - Cuando "y" envíe paquetes a "z" se enviaran de vuelta a "y" hasta que el paquete muera… —› *lazo de enrutamiento*.
- Luego "y" informa su nuevo cálculo a "z"; con mentiras refinadas ☺, es decir que tiene un camino de costo 6 basado en el camino fantasma de costo 5 de "z"… (*suma al infinito*).

Retorno Envenenado



- Para evitar el lazo infinito se hace un retorno envenenado.
- "z" que conoce la dificultad, envía un nuevo peso de ruteo a "x" —› $D_z(x) = \infty$
- Esperar que "y" enrute (forzado) el tráfico a través de "x"
  - Tabla de distancia de "y" indica $D_z(x) = \infty$.
  - Ante el cambio 4—›60 "y" actualiza tabla y envia directo a "x"
  - Se envía el calculo a "z" y éste se dá cuenta que puede enviar más rápido por $D_z(x) = 50$.
- Esta solución NO sirve para más de 3 nodos ☹

Comparación entre los LS y DV

- Complejidad de los mensajes
  - LS: con **n** nodos, **E** enlaces —› **O(nE)** mensajes
    - Transmite info de los nodos vecinos
  - DV: intercambio entre los vecinos solamente
    - Transmite info de todos los caminos
    - Tiempo de convergencia variable
- Velocidad de convergencia
  - LS: Algoritmo $O(n^2)$, mensajes O(nE)
    - Pueden haber oscilaciones
  - DV: Tiempo de convergencia varía
    - Círculos de enrutamiento
    - Conteo al infinito

Comparación LS y DV

- Robustez: ¿Qué sucede si el enrutador no funciona o falla parcialmente?
  - LS:
    - Se pueden difundir costos incorrectos de enlaces
    - Cada nodo calcula solo su propia tabla
  - DV:
    - Puede difundir costos incorrectos de caminos
    - Cada tabla de nodo es usado por los otros ➔ error de propagación

Ruteo Jerárquico: “divide and conquer”

- En LS y DV los ruteadores son tratados todos como iguales para computar la ruta.

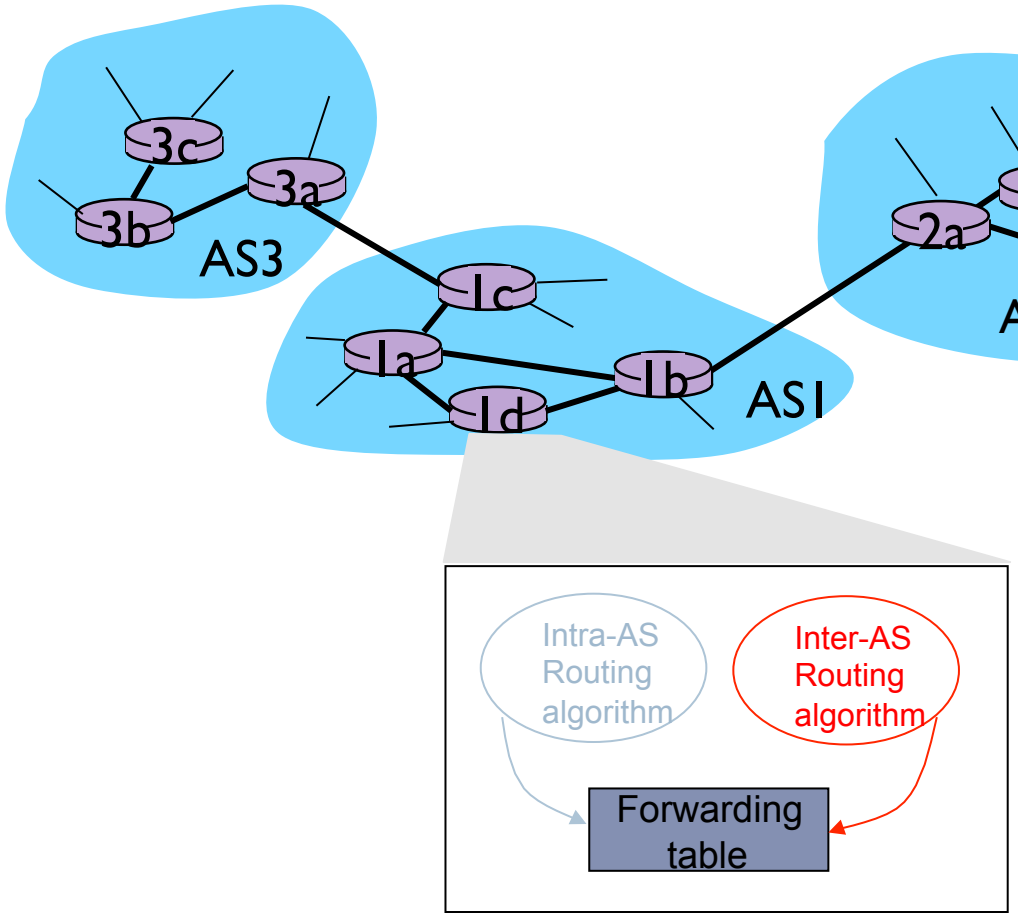
  Visión **simplista criticable** debido a:

- Escala:
  - Número de mensajes prohibitivo para cierto número de routers (100tos de miles)
  - Se requieren muchísimos recursos: memoria y ancho de banda.
  - ¿Intuye el costo de convergencia?
- Autonomía administrativa
  - Una organización debería ser autónoma en la administración de su propia red y al mismo tiempo poderse comunicar con otras organizaciones.

Solución: "Autonomous Systems"

- Recuerde la definición de Internet: "red de redes".
- Se agrupan enrutadores en regiones llamadas "Autonomous Systems" (AS).
  - Mismo control administrativo (ISP, Compañía de Red)
- Routers en el mismo AS ejecutan el mismo algoritmo (DV, LS → "intra AS routing protocol").
  - Pero en diferentes AS pueden correr diferentes algoritmos
- Hay routers especiales que llevan la información hacia fuera. → "gateway routers".

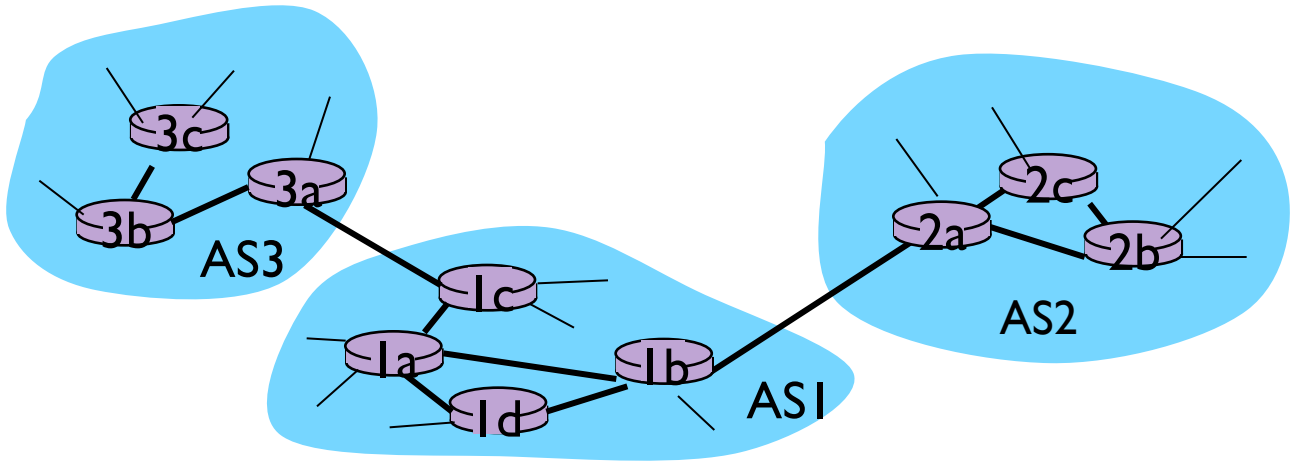
Ejemplo: AS interconectados.



- La tabla de forwarding es configurada por las dos clases de algoritmos
  - intra-AS para los destinos internos
  - inter-AS & intra-As para los destinos externos

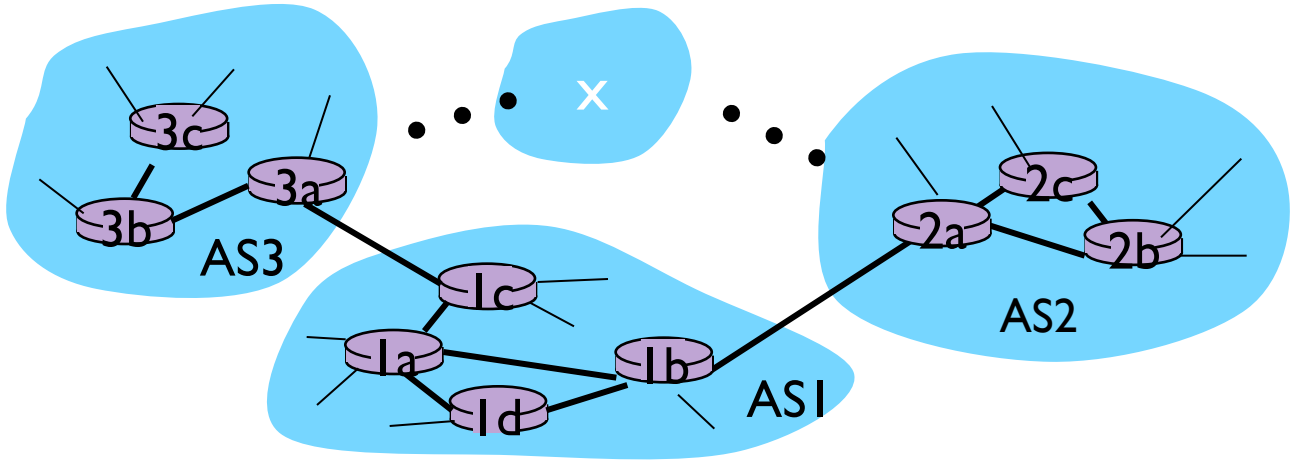
Ruteo Inter-AS

- ¿Cómo enrutamos paquetes fuera del AS?
  - Sencillo de hacer si hay un solo router "gateway" dentro del AS
- Paquetes llegan al gateway y éste pasa al siguiente AS.
- Para 2 o más enlaces que llevan el tráfico fuera del AS.
  - Se usa un protocolo inter-AS para informar a los ruteadores los destinos alcanzables por cada salida.
    - Primero saber los destinos, luego repartirlos a todos lo ruteadores.



Ejemplo: Selección entre múltiples AS

- Suponga que AS1 conoce del protocolo inter-AS que la subred **x** se alcanza desde AS3 y AS2.
- Para configurar la tabla de forwarding, el enrutador 1d debe determinar el "gateway" para llevar los paquetes al destino **x**.



Ejemplo: Selección entre multiples AS

- Se usa el protocolo de "ruteo de papa caliente".
  - Conocer a partir del protocolo inter-AS que la subred x es alcanzable a partir de varios gateways
  - Determinar el camino de menor costo a cada uno de ellos
  - "ruteo de papa caliente": seleccione la ruta de mínimo costo hacia el "gateway".
  - Determinar la interfaz "I" que llevará al "gateway"
  - Añadir una ruta en la tabla que diga (x, "I").

Ruteo en la Internet

Ruteo en la Internet

- Determinar el camino de los datagramas desde el origen hasta el destino.
- Involucra muchos de los conceptos vistos hasta ahora.
  - Autonomous Systems
  - Algoritmos Intra-AS
  - Algoritmos Inter-AS

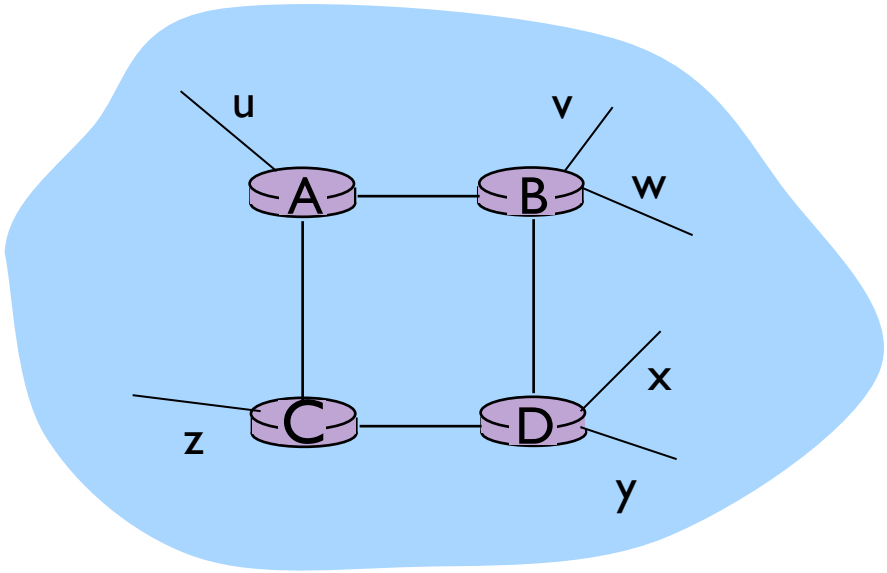
Ruteo Intra-AS en la Internet (RIP)

- Son conocidos también como Interior Gateway Protocols
- Hay dos protocolos Típicos:
  - Routing Information Protocol (RIP)
  - Open Shortest Path First (OSPF)
  - Propietario: Interior Gateway Routing Protocol (CISCO)

RIP (Routing Information Protocol)

- Uno de los los precursores "Intra-AS"
  - Arquitectura basada en Xerox Network System
- Se incorpora como parte de BSD en 1982 (gran puesta en marcha)
  - V1 —› RFC 1058
  - V2 —› RFC 2453
- Funciona como DV pero usa *hop-count* como métrica de costo… cada enlace tiene costo 1.
  - Limitado a 15 hops (saltos)
- Se intercambian mensajes cada 30 segs (RIP mensajes de advertencias)
  - Contiene hasta 25 subredes destino dentro del AS

RIP: un ejemplo



Desde router A a las subredes:

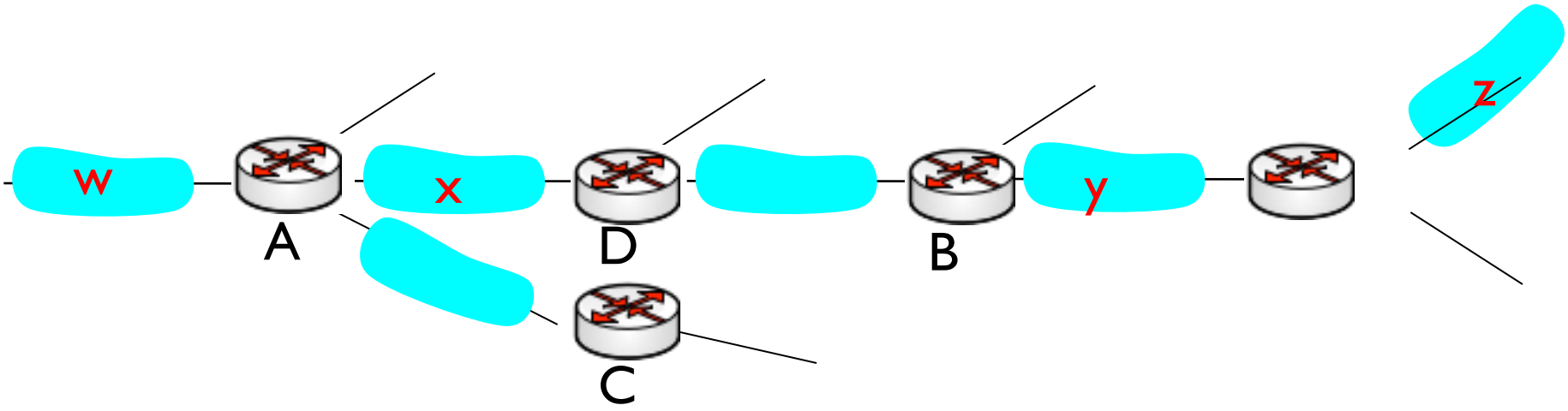
Subred destino	saltos
u	1
v	2
w	2
x	3
y	3
z	2

- Un AS con 6 subredes direccionables.
- Costo máximo 15
- Se intercambian msg cada 30 segs.

Avisos RIP

- En el próximo ejemplo, solo se consideran routers y subredes etiquetadas.
- Líneas punteadas dicen que el AS continúa.
- Cada enrutador tiene una "routing table" (RT).
  - RT = routing + forwarding
- Cada 30 segs se puede tener una nueva tabla.
- Router D actualiza su tabla
  - ¿Cómo se resolvió el camino más corto?

Ejemplo



Destination Network	Next Router	Num. of hops to dest.
w	A	2
y	B	2
z	B	7
x	--	1
....	....	....

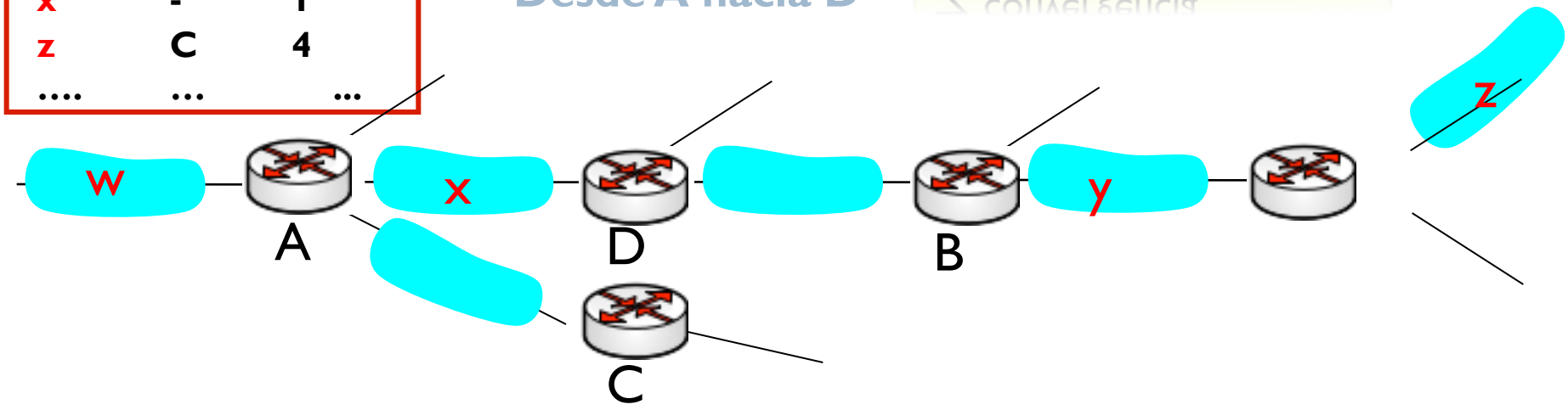
Tabla de Routing/Forwarding en el router D

Ejemplo de RIP

Dest	Next	hops
w	-	1
x	-	1
z	C	4
....	...	...

Aviso periodico
Desde A hacia D

Pero por qué es posible?
→ convergencia



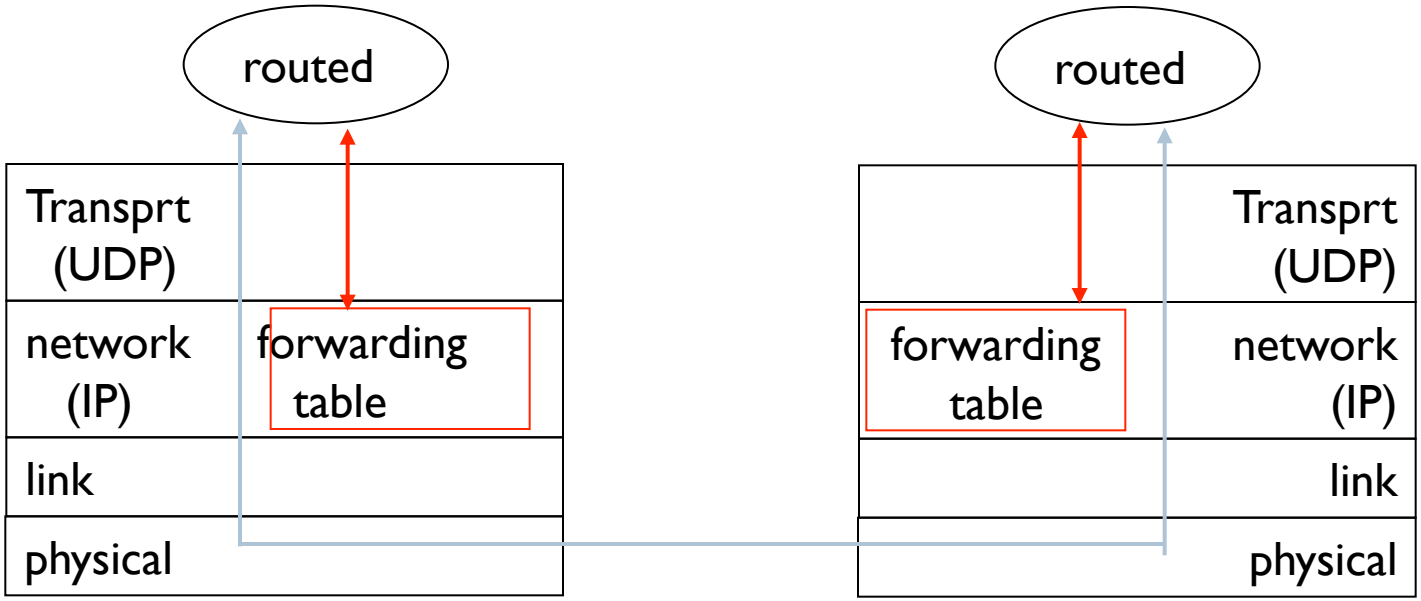
Destination Network	Next Router	Num. of hops to dest.
w	A	2
y	B	2
z	~~B~~ A	~~7~~ 5
x	--	1
....	....	....

Detalles de Implementación: *Ruta inalcanzable*

- Si no se escucha al vecino por 180 segs se da por inalcanzable
  - Las rutas a través de los vecinos se invalidan
  - Se reenvían nuevos avisos indicando lo sucedido
  - Se propaga rápidamente el mensaje y la falla a través de msg. UDP (… de capa 4 ☺)
  - Si esto sucede se reenvían avisos a los vecinos cercanos (puerto 520).
  - Reverso envenenado se usa para prevenir ping-pongs

Tabla de procesamiento de RIP

- Usa UDP encima de IP para implementar el protocolo.
- Implementación
  - Se usa UDP para pasar información al demonio "routed" a nivel app.



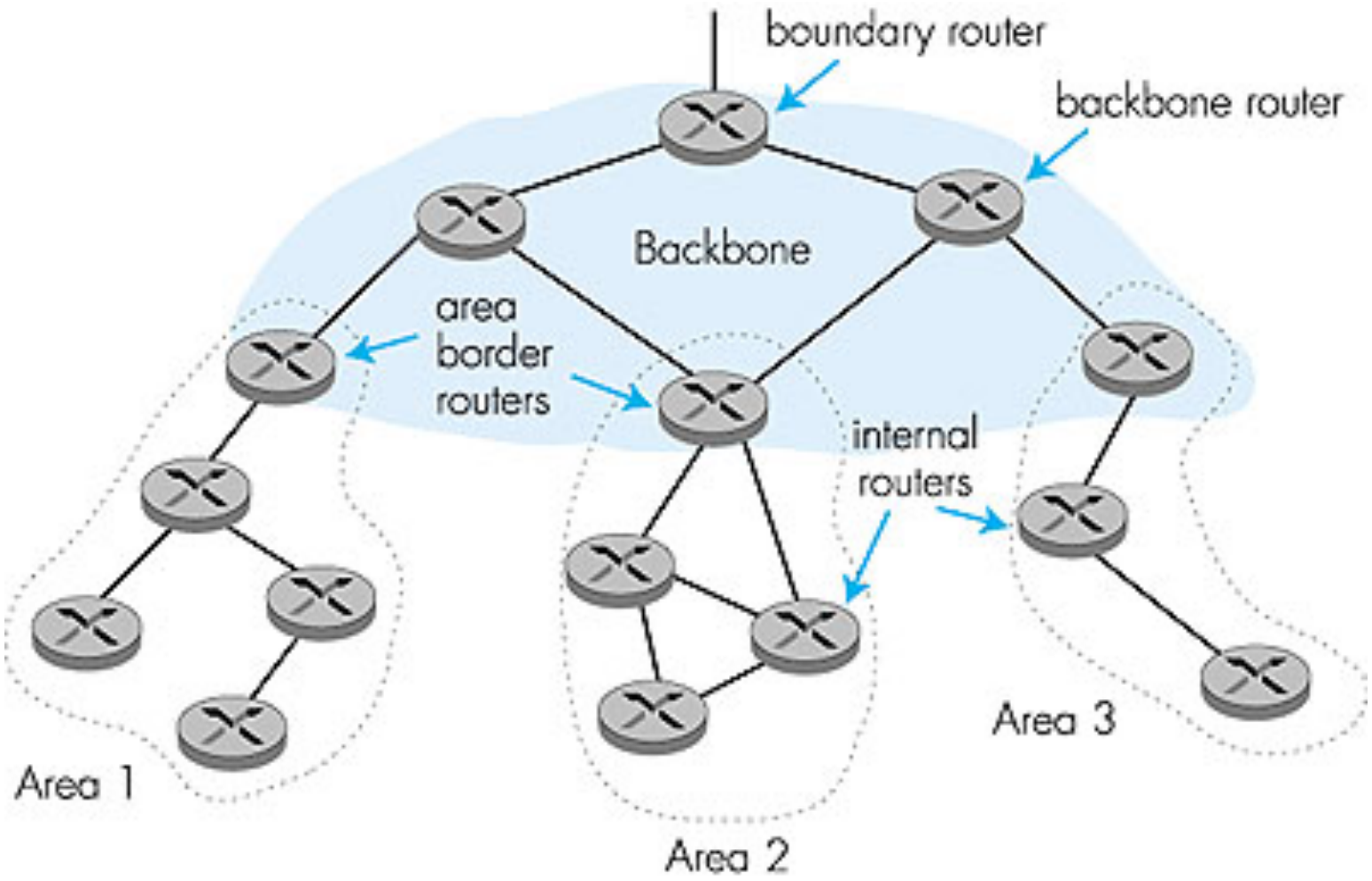
Ruteo Intra-AS en la Inet: OSPF

- OSPF se implementa en "tiers altos", RIP en tiers "bajos"
- Conocido como sucesor de RIP —› protocolo LS
  - Sistema abierto: disponible publicamente
  - Mapa topologico de toda la red en cada nodo
  - Se computa la ruta usando Dijsktra
- Costos individuales fijados por el administrador
  - Si son igual a 1—› minimo camino (# hubs)
  - 1/BW descartar ruteadores de bajo ancho de banda
- Muestra información de un nodo cuando hay cambios
  - Las advertencias son diseminadas al AS entero (via innundación o broadcasting)

Caracteristicas avanzadas de OSPF (no estan en RIP)

- **Seguridad**: intercambios (msgs OSPF) asegurados entre enrutadores. Se intercambia una clave con la que se encripta y decripta el paquete.
- **Múltiples caminos con el mismo costo**: Se pueden buscar varios caminos con el mismo costo para repartir la carga —› uno solo en RIP.
- **Diferentes métricas de costo**: distinto Tipos de Servicio (TOS) (ej: enlaces satelite, BE —› Baja prioridad, Real Time —› Alta)
- **Unicast/Multicast routing**: MOSPF provee el soporte se usa un nuevo aviso "link-state advertisement" para m-cast.
- **Soporte jerárquico de un mismo dominio de ruteo**: puede estructurar un AS "grande" jerárquicamente.

Jerarquización de un AS



Jerarquización de un AS

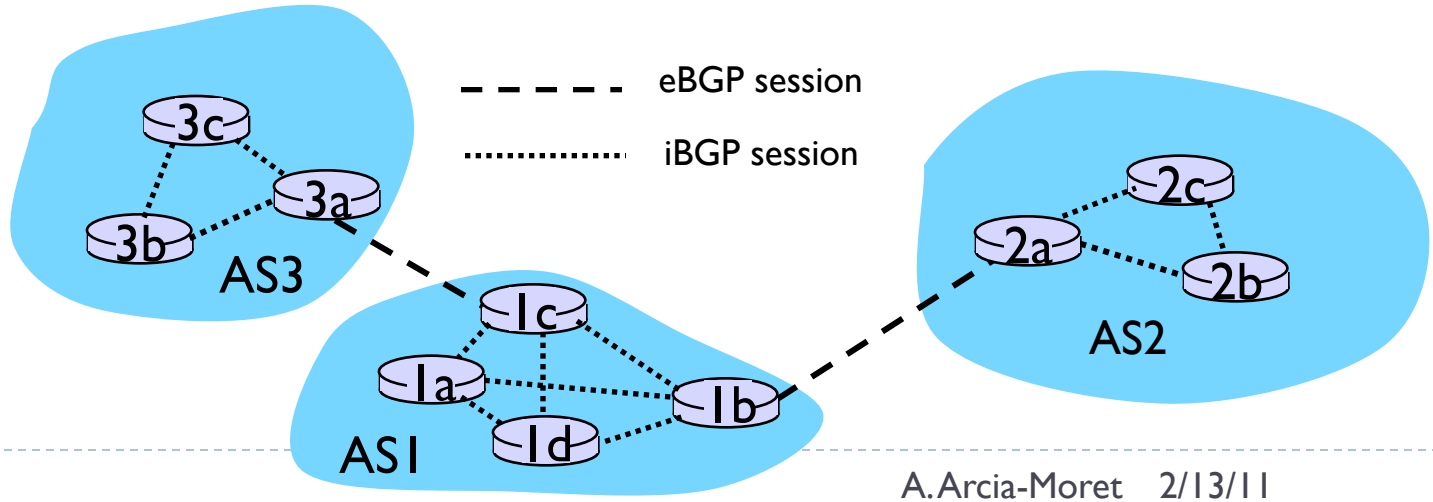
- Jerarquización por Areas: Area local y Backbone.
  - Avisos LS en el área
  - Cada nodo conoce direcciones caminos cortos a otras redes en otras áreas.
- Cada área ejecuta un algoritmo OSPF para ruteo
  - Se hace broadcast del estado de cada nodo
- Tres tipos de enrutadores
  - Enrutadores de borde de área: "resume" las distancias a redes en su propia área
  - Ruteadores Backbone: Ejecutan OSPF limitado al backbone
  - Ruteadores de Borde: Conectan con otros AS.
- Se enruta en **intra-Area** y luego se pasa al **backbone**.

Ruteo Inter-AS: BGP (Border Gateway Protocol)

- BGP es el estándar de facto
- RIP y OSPF determinan rutas críticas internamente
- ¿Qué sucede a través de múltiples AS? BGP provee:
  1. Dice si una red es alcanzable desde un AS vecino
  2. Propaga información de alcanzabilidad a otros ruteadores
  3. Determina buenas rutas a subredes en base a la información (2)
- Una red se anuncia (¡Aquí estoy!), BGP la toma en cuenta y hace que sea alcanzable desde cualquier AS.

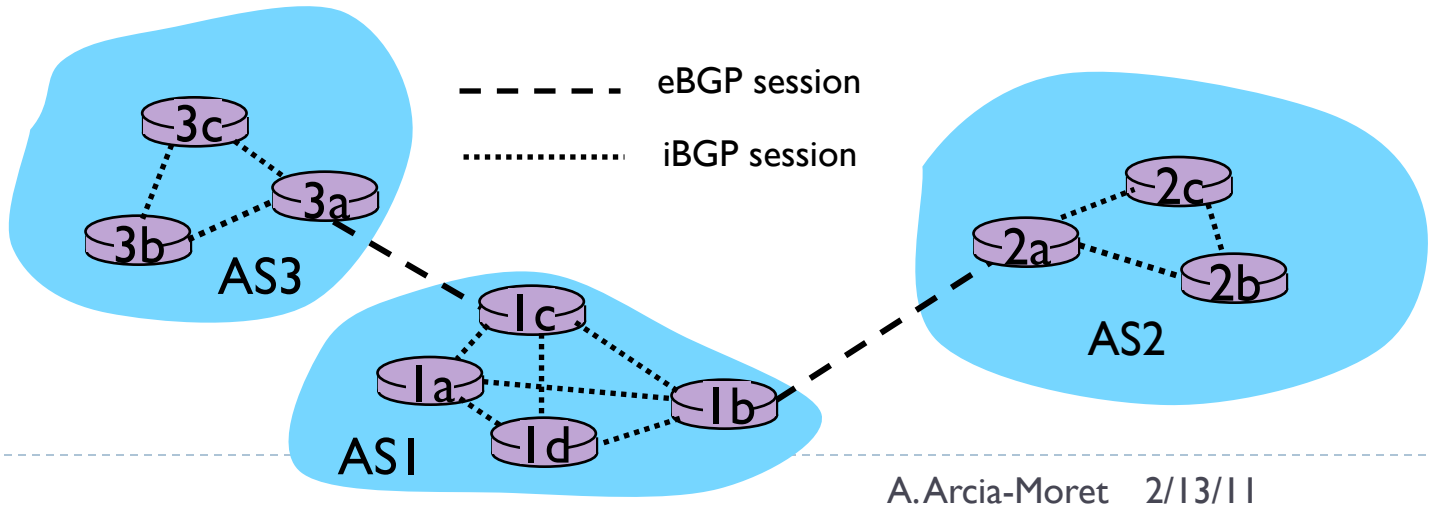
Principios de BGP

- Es el protocolo que enlaza a toda la Internet
  - Par de ruteadores (BGP-peers) intercambian información de ruteo a través de conexiones TCP semi-permanentes.
- Conjunto msgs + conexión = Sesión BGP
  - Hay sesiones externas (e-BGP) e internas (i-BGP)



Principios de BGP

- Vea que cuando AS2 publica un prefijo hacia AS1:
  - AS2 hace promesa de reenviar los datagramas a dicho prefijo
  - AS2 puede agrupar (*agragate*) prefijos en sus publicaciones
- BGP permite saber qué redes son alcanzables a través de AS vecinos:
  - AS3 envía su prefijo a AS1 a través de **3a-1c**. 1c usa iBGP. 1b redistribuye la información a AS2 con sesión eBGP.
  - Luego se actualizan las tablas de forwarding.



Atributos de Caminos y Rutas BGP

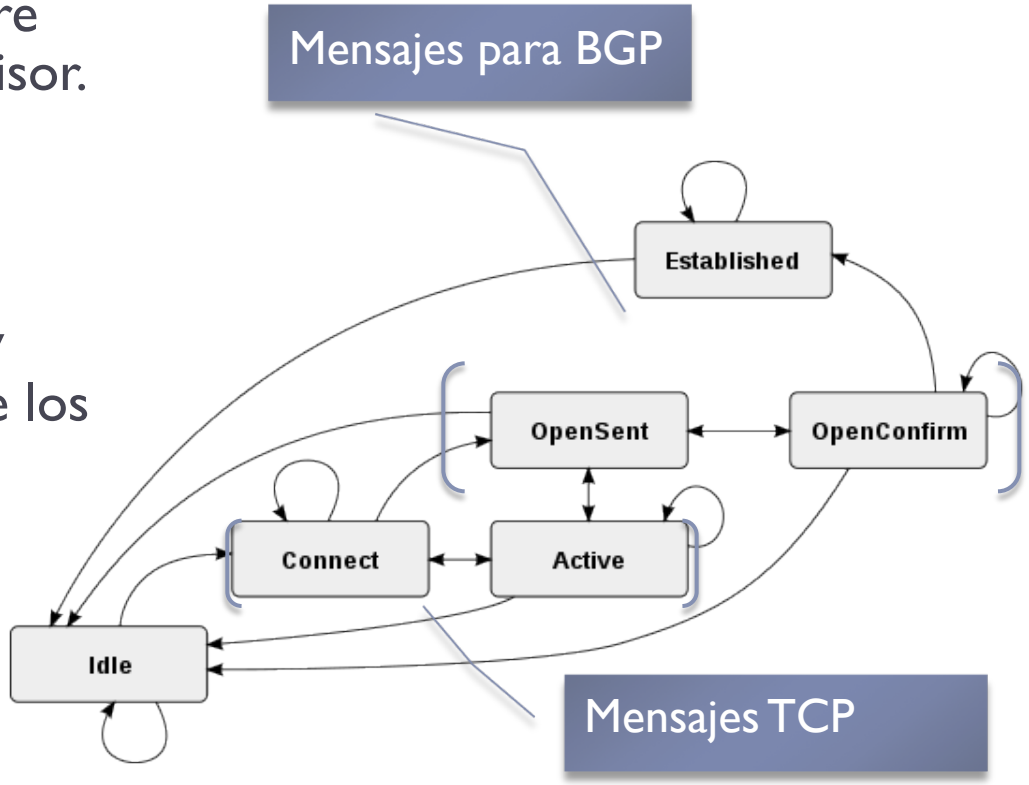
- Identificado con un número-AS (RFC 1930)
  - Asignado por el ICANN
- Se difunden algunos atributos BGP:
  - prefijo + atributos = ruta.
- Dos atributos importantes:
  - AS-PATH: Contiene los prefijos a través de los cuales ha pasado. Ej: AS-PATH = { AS2, AS1 }
    - Evita lazos, se usa para configurar la tablas de forwarding
  - NEXT-HOP: indica la interfaz en el router que comienza el AS-PATH.
- Cuando un ruteador recibe avisos de rutas, utiliza una política de importación (diferentes métricas) para aceptar/declinar.

Selección de la ruta BGP

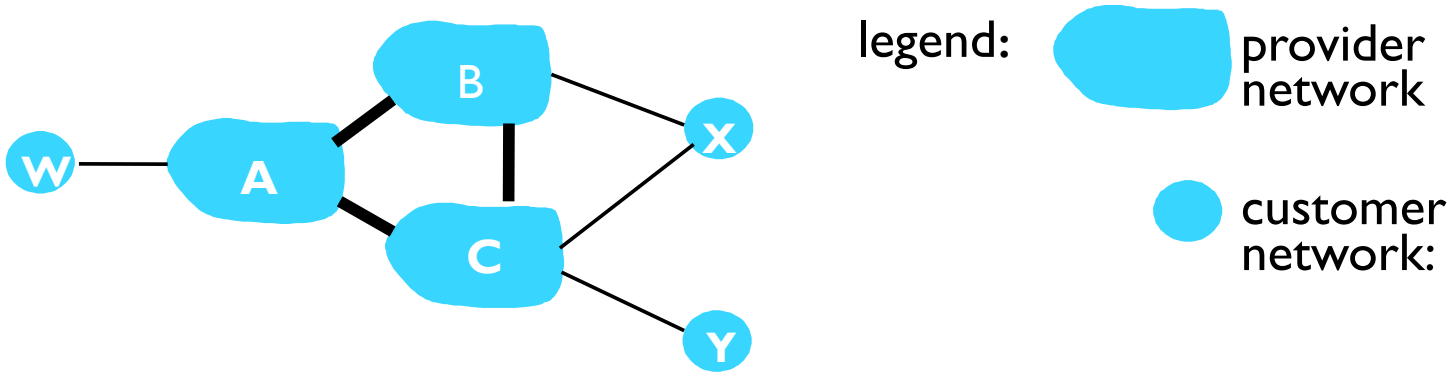
- BGP = eBGP + iBGP
- Puede haber más de una ruta por prefijo
  - Se utilizan las siguientes reglas hasta que una ruta "decante"
  - Parar cuando se encuentra ruta única, si no aplicar hasta el final
- Reglas de eliminación:
  1. Ruta de preferencia o por omisión (por el administrador)
  2. Camino mas corto (AS-PATH)
  3. (hay varios pref. local y varios cortos) —› NEXT-HOP más cercano: enrutamiento "papa-caliente".
  4. Criterios adicionales (ej: ID de BGP).

Mensajes BGP

- Los msg BGP se intercambian usando TCP
- Mensajes BGP:
  - **OPEN**: abre la conexión entre dos "peers" y autentica al emisor.
  - **UPDATE**: muestra un nuevo camino (o apaga algún otro)
  - **KEEPALIVE**: mantiene la conexión viva (cuando no hay UPDATES), también reconoce los request OPEN.
  - **NOTIFICATION**: reporta errores, y también cierra una conexión.

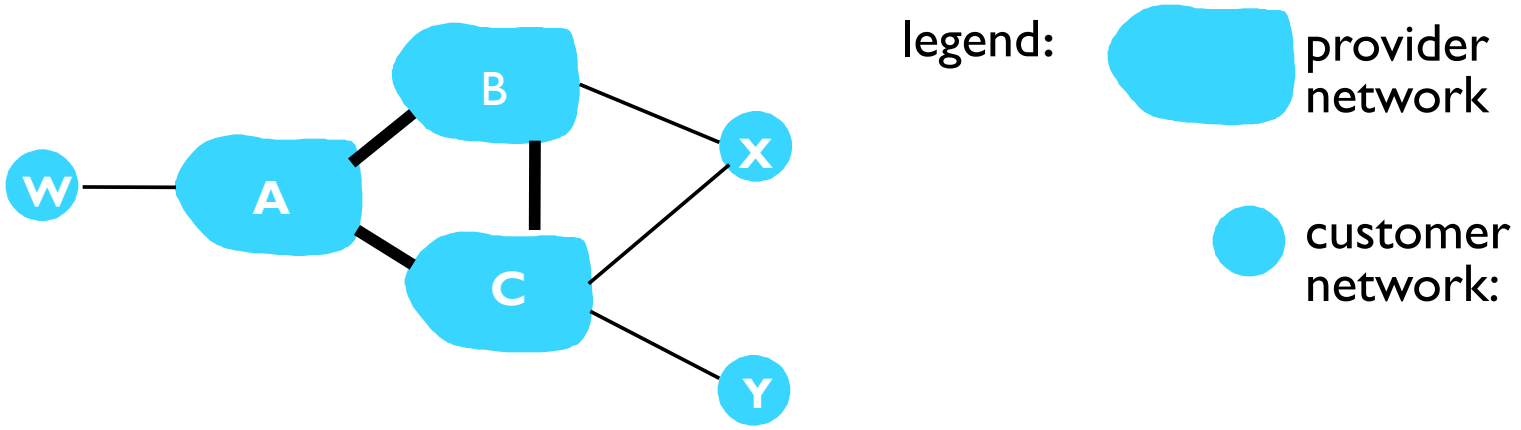


Políticas de Enrutamiento BGP



- A, B y C son redes de backbone
- w, x, e y son redes de clientes (de A, B y C)
  - Stub: recibe u origina información
  - Backbone: forwarding
- x es "multihomed" pegada a dos redes
  - x no quiere enrutar tráfico de B hacia C.
  - x no anuncia a B su ruta hacia a C.

Políticas de Enrutamiento BGP (2)



- ¿Debe B enrutar tráfico hacia C?
  - Si lo hiciese, se tendría una ruta CBAW
- B **no** debería hacer el trabajo de C.
  - No hay ganancia en hacerlo (ni C ni w son sus clientes!)
  - B fuerza a C enrutar via A para llegar a **w**
  - B solo atiende (enruta) a sus clientes.
  - Las reglas entre los ISP son confidenciales

¿Por qué se usan diferentes protocolos de ruteo Inter e Intra-AS ?

- POLITICA:
  - **Inter**: Entre AS dominan los aspectos políticos.
  - **Intra**: El administrador controla el tráfico enrutado
- ESCALA:
  - Algoritmos de ruteo / Estructuras de datos son un problema en el ruteo Inter-AS.
  - Intra-AS es menor problema, pues es una restricción de diseño.
- PERFORMACE:
  - Intra AS: se focalizan esfuerzos en la optimización de las rutas (performance)
  - Inter AS: Las políticas resaltan sobre el performance.

Broadcast/Multicast Routing

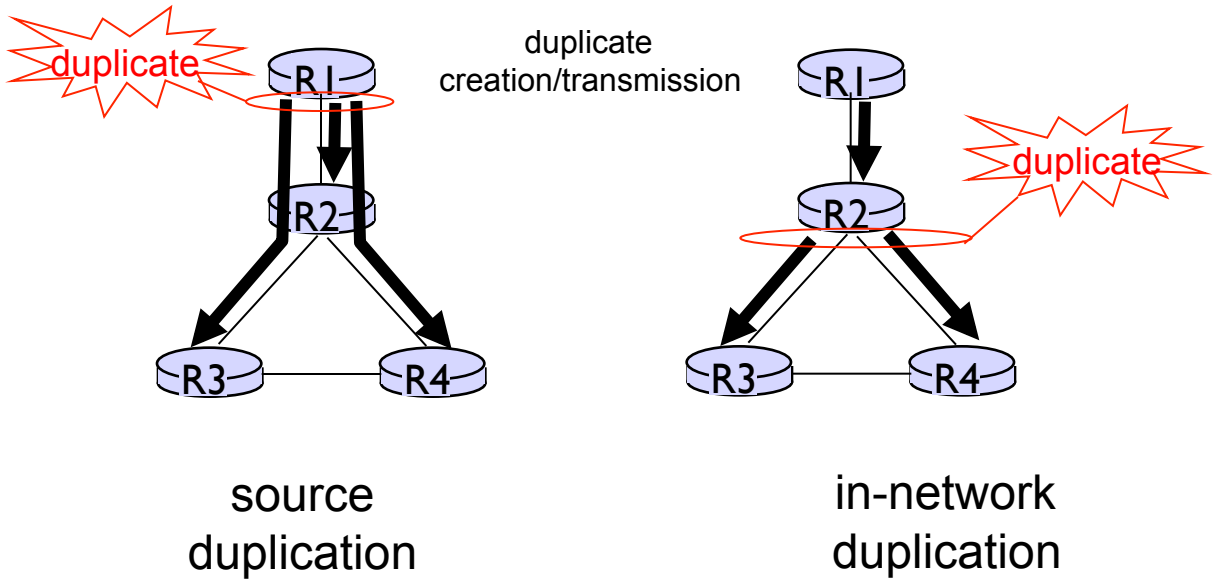
- Hasta ahora se ha estudiado ruteo "unicast" (comunicación p2p)
  - Ruteo broadcast: Servicio de entrega de un nodo a todos los nodos de la subred.
  - Ruteo multicast: Servicio de entrega a un subconjunto de "La Red".

Algoritmos Broadcast

- Podría implementarse enviando una copia del paquete a cada destino:
  - Hacer N copias
  - Enviar a N destinos con mensajes unicast
  - Simple y no necesitaría protocolo, pero ineficiente O(N)
- Hacen falta grupos para comunicarse
  - Pero, más overhead y más complejidad.
- Finalmente, unicast de N mensajes es competencia exclusiva de "broadcast".

Enrutamiento Broadcast

- Envía paquetes de la fuente a todos los nodos
- Duplicado de la fuente es ineficiente:



- ¿Cómo se determinan las direcciones de los destinos?

Inundación Descontrolada

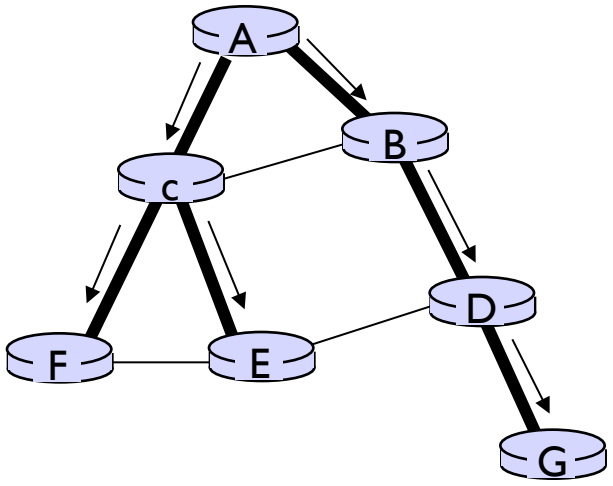
- Enviar una copia del paquete a todos los vecinos
  - Cada nodo hace la misma operación (excepto hacia la fuente de envío).
- Simple y elegante pero:
  - Si hay ciclos en el grafo los paquetes ruedan por siempre
  - Lluvias (tormentas) de broadcast se multiplica por el número de vecinos se tengan → puede hacer la red inutilizable.

Inundación controlada

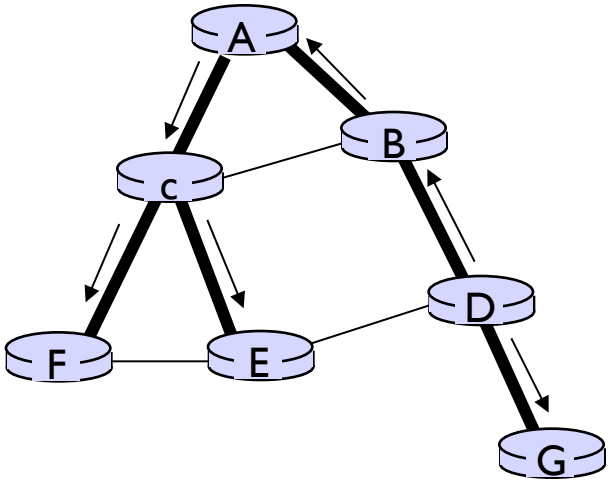
- Hay que escoger bien cuando inundar o **no** inundar…
- Evitan tormentas de broadcast (pero no redundancia)
- Varias maneras:
  - Inundación con # seq controlado
    - Se coloca # seq al broadcast
    - Se envía el paquete
    - Cada nodo mantiene una lista de dirección fuente y # de secuencia de cada paquete, recibido, reenviado o guardado.
    - Descartar si el paquete está en la lista (ej: Gnutella).
  - Reenvío de camino inverso
    - Simple y elegante
    - Retransmitir a todos sus vecinos excepto al padre si los nodos están en la ruta crítica
    - No looping y No broadcast
    - Solo necesita el conocimiento del vecino en la ruta del camino crítico

Spanning Tree para Broadcast

- Elimina por completo la redundancia.
- Es un árbol de formación cuasi-espontánea.
- Solo se hace forwarding a lo largo del árbol de spanning.



(a) Broadcast iniciado en A



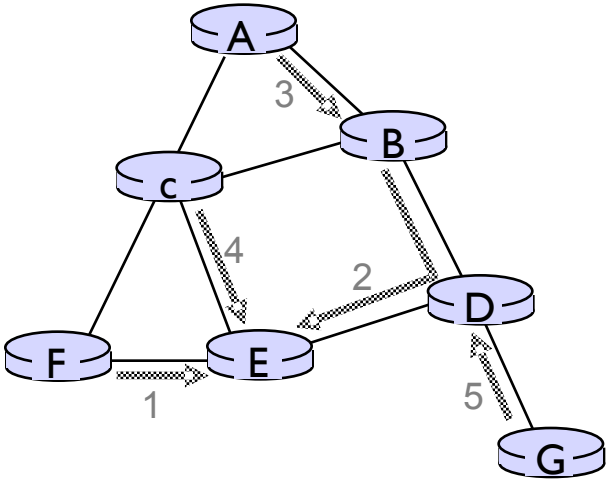
(b) Broadcast iniciado en D

Spanning Tree

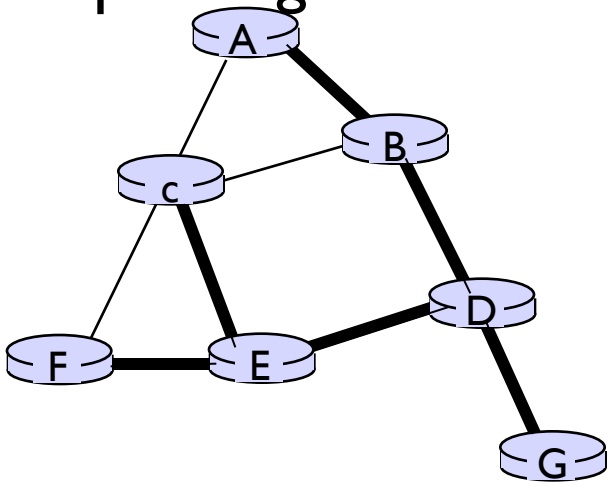
- Spanning Tree de un grafo G=(N,E) es un grafo G'=(N,E') tal que E' es subconjunto de E, G' es conexo y no tiene ciclos. G' tiene todos los nodos de G.
- El costo es la suma de todos los enlaces, cuando la suma es mínima, se tiene un **spanning tree mínimo**.
- Envíos
  - Se envía un mensaje por el **spanning tree** calculado paso a paso sin enviar por el nodo precedente
  - Se puede enviar en ambos sentidos de la ruta
  - Observe que cada nodo "debe" conocer a sus vecinos

Creación del Spanning Tree

- Buscar el nodo central
- Cada nodo envía mensajes para unirse al arbol (tree-join msg) a través de una ruta crítica.
- Msg llega al "centro" o a un nodo del "spanning tree"
- Una ruta nueva se considera del spanning tree



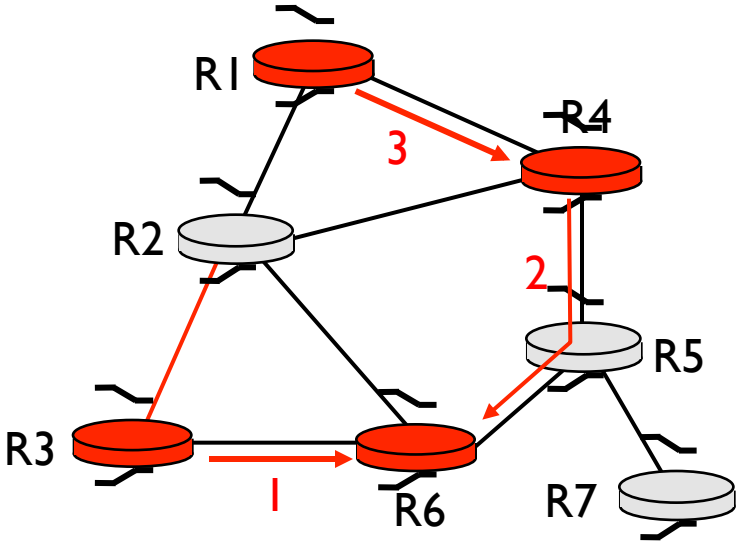
(a) Stepwise construction of spanning tree



(b) Constructed spanning tree

Ejemplo de Nodo Central

- Suponga que R6 es el nodo central



LEGEND

router with attached group member

router with no attached group member

1 path order in which join messages generated

Ruteo Multicast

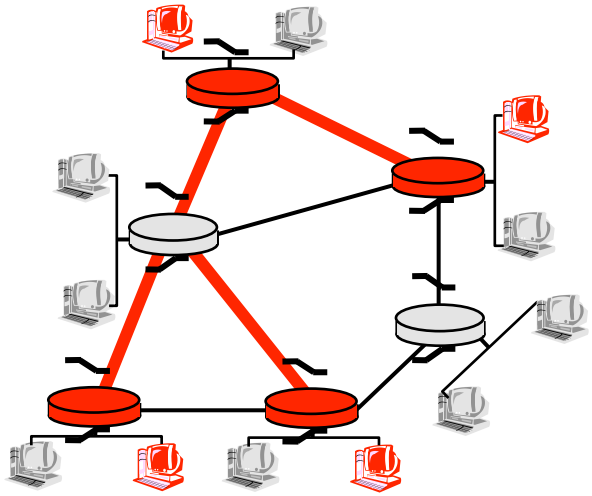
- Los paquetes son entregados a un subconjunto de los nodos.
- Aplicaciones posibles:
  - Transferencia de datos (updates de programas)
  - Streaming (audio, video, clases en vivo)
  - Data compartida (pizarra o teleconferencia)
  - Data especifica (valor de las acciones en la bolsa)
  - Juegos interactivos
- Problemas enfrentados:
  - ¿Cómo se identifica al receptor del paquete multicast?
  - ¿Cómo direccionar un paquete enviado a estos receptores?

Direccionamiento Multicast

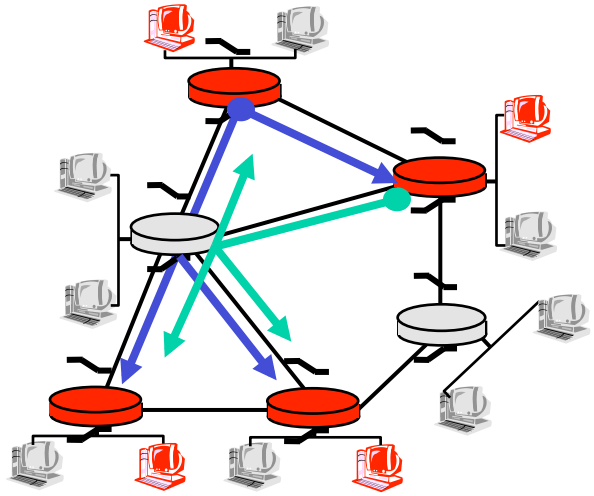
- Definitivamente no se puede llevar en cada paquete la dirección de cada receptor (sobrepasaría el peso de la data).
- Se utilizan "indirecciones" (address indirections) o direcciones "intermedias".
  - Una sola IP sirve para todo el grupo multicast

Ruteo Multicast: ¿Cuál es el problema?

- Objetivo: Encontrar uno o más árboles que conecten a los enrutadores que tengan miembros a **grupos locales multicast**.
  - **Arbol:** No todos los caminos entre enrutadores se usan.
  - **Basado en la fuente:** árboles diferentes desde cada emisor a receptor.
  - **Basado en árbol compartido:** por todos los miembros del grupo.



Shared tree



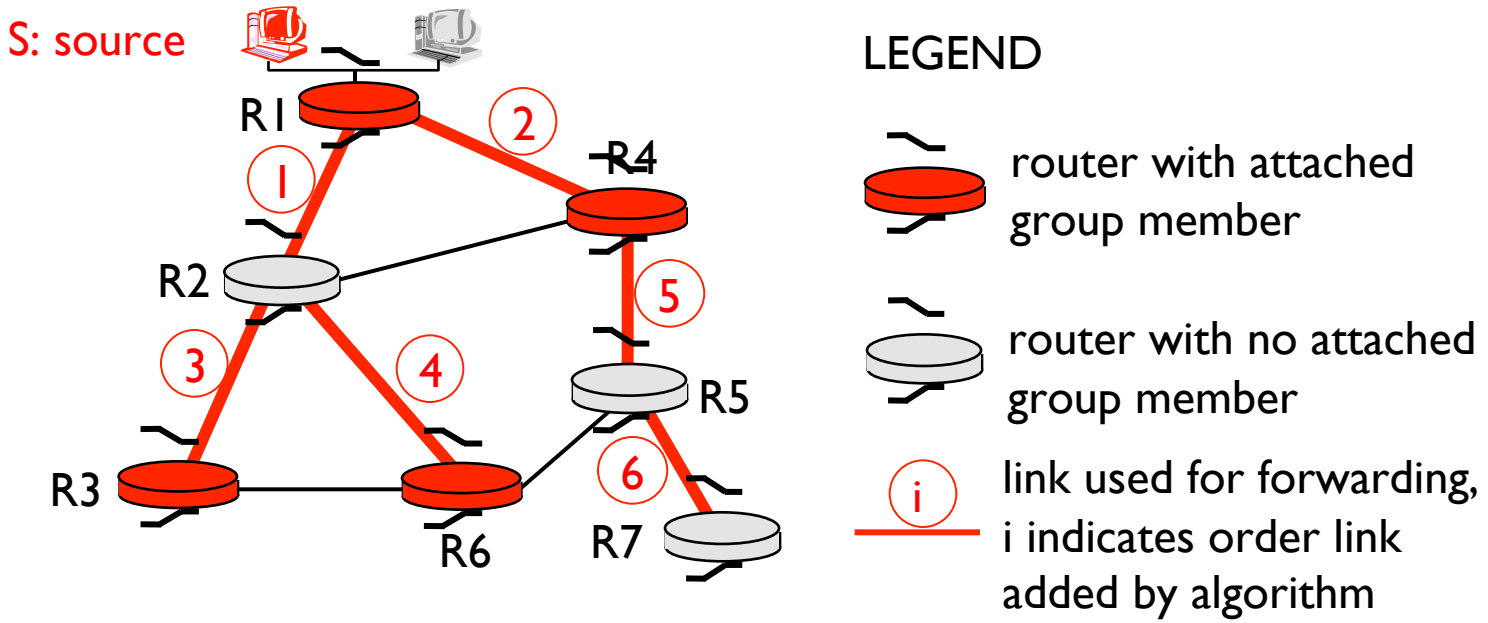
Source-based trees

Enfoques para construir los arboles mcast

- Basados en la fuente: un árbol por fuente
  - Árboles de caminos mínimos
  - Forwarding en la ruta reversa
- Árboles de grupos compartidos: los grupos usan un solo árbol
  - Árbol de "spanning" mínimo (Steiner)
  - Árboles basados en determinar el centro

Arbol de Camino Mínimo

- Arbol de forwarding mcast: Arbol con el camino más corto desde la fuente a todos los receptores
  - Se usa el algoritmo de Dijkstra.



Forwarding con Camino Reverso

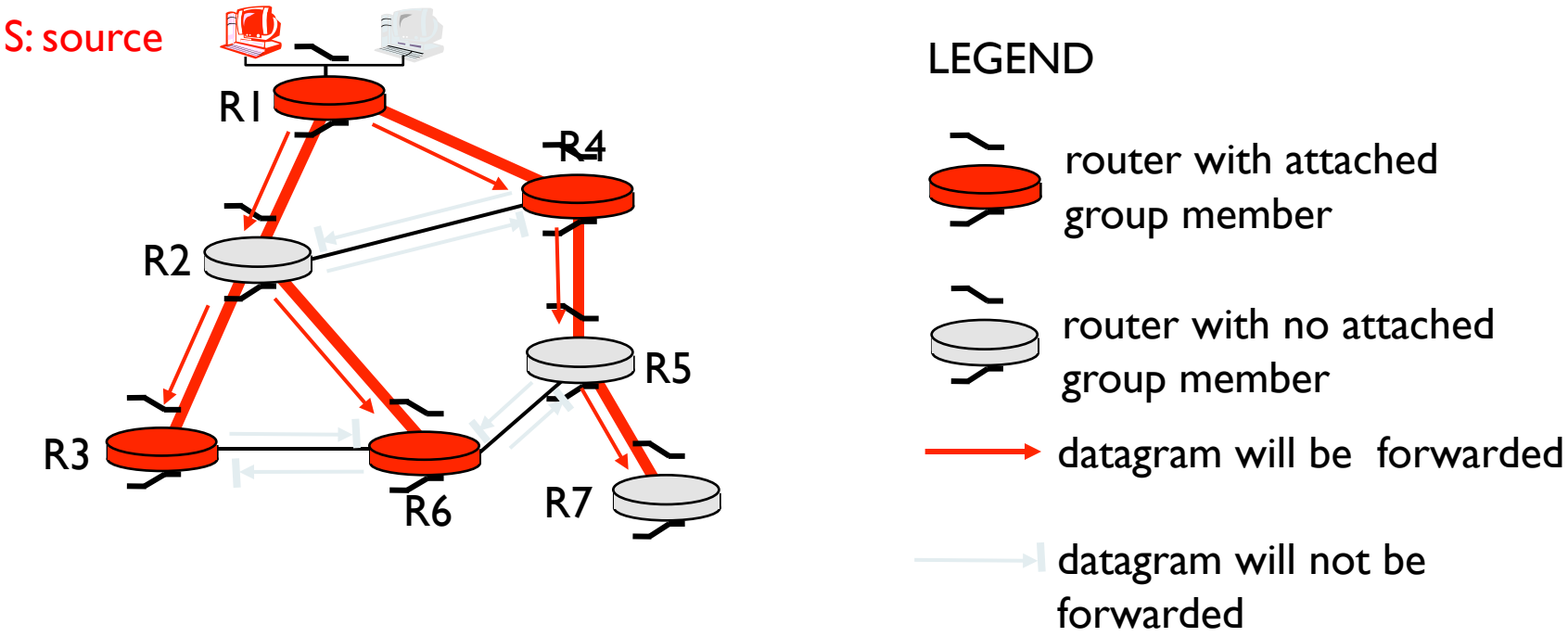
- Confía en el conocimiento del router del camino más corto desde él hasta el emisor
- Cada router tiene políticas simples de forwarding

Si (datagrama mcast se recibe por el camino critico)

entonces reenviar (inundar) datagrama en todos sus enlaces

si no ignorar datagrama

Ejemplo



- Resulta en un reverso con fuente específica
- Puede ser mala elección para rutas asimétricas

Forwarding en camino reverso: podar

- El árbol de forwarding contiene sub-árboles con miembros que no son del grupo multicast
  - Estos nodos **no** tienen porque reenviar datagramas en en árbol
  - Se poda en el camino de regreso para no hacer bajar más mensajes.