Cap. 4: Capa de red

- 4. 1 Introducción
- 4.2 circuitos virtuales y datagramas
- 4.3 dentro de un router
- 4.4 IP: Internet Protocol
  - formato de datagramas
  - direccionamiento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de enrutamiento
  - Link state
  - Distance Vector
  - Enrutamiento jerárquico
- 4.6 Enrutamiento en Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast y multicast

Algoritmo Distance Vector

Ecuación de Bellman-Ford (programación dinámica)

Se define

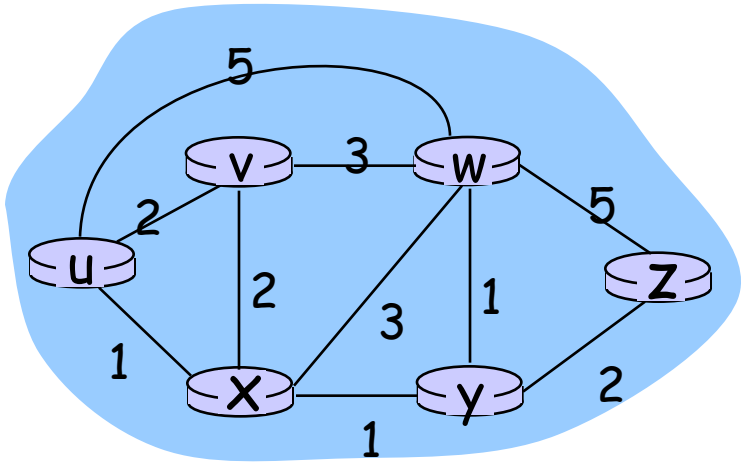
$d_x(y)$:= costo del camino de menor costo de x a y

Luego

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

donde min se calcula entre todos los vecinos v de x

Ejemplo Bellman-Ford



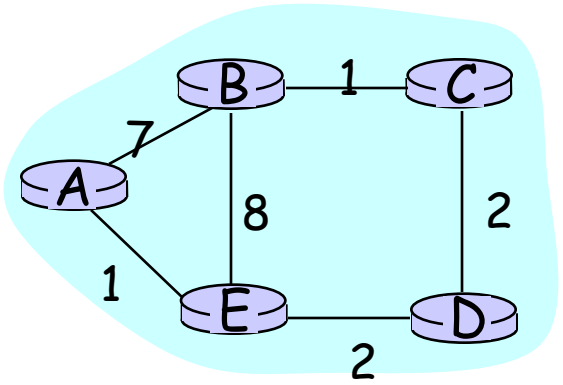
Se cumple,
$d_v(z) = 5$, $d_x(z) = 3$, $d_w(z) = 3$

La ecuación de B-F dice:

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

El nodo que logra el mínimo es el "next hop" en el camino más corto ➜ tabla de forwarding

Tabla (vector) de distancias: ejemplo



$D_E(C,D) = c(E,D) + \min_w \{D_D(C,w)\}$

$= 2+2 = 4$

$D_E(A,D) = c(E,D) + \min_w \{D_D(A,w)\}$

$= 2+3 = 5$

loop!

$D_E(A,B) = c(E,B) + \min_w \{D_B(A,w)\}$

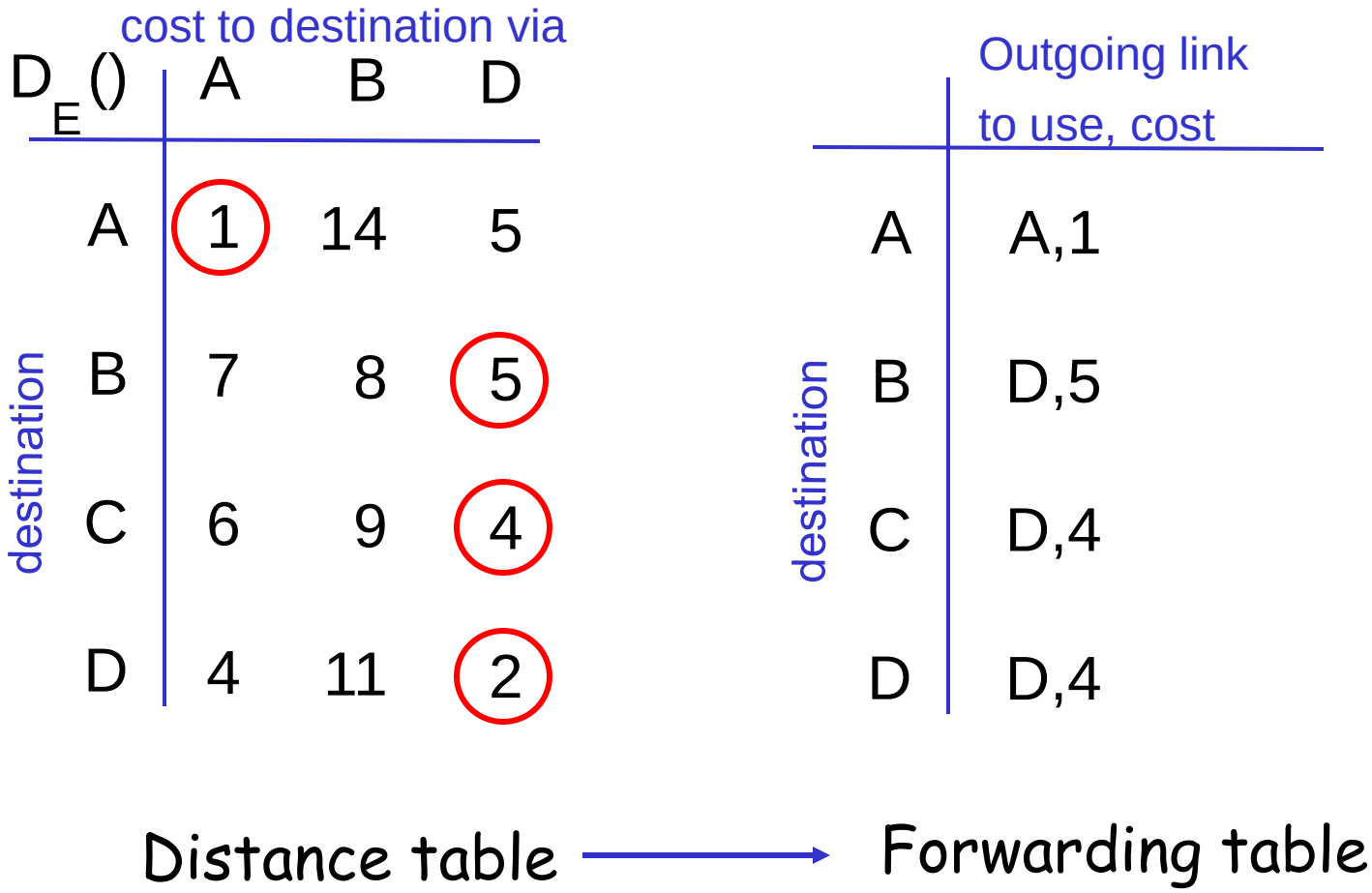
$= 8+6 = 14$

loop!

$D_E()$ — cost to destination via

destination	A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

Tabla de distancias -> forwarding



Distance table → Forwarding table

Algoritmo Distance Vector

- $D_x(y)$ = estimación del menor costo de x a y
- Nodo x conoce el costo a cada vecino v: $c(x,v)$
- Nodo x mantiene el "distance vector" $\mathbf{D_x} = [D_x(y): y \in N]$
- Nodo x también mantiene el vector de distancia hacia sus vecinos
  - Para cada vecino v, x mantiene $\mathbf{D_v} = [D_v(y): y \in N]$

Algoritmo Distance Vector

Idea básica:

- Cada cierto tiempo, cada nodo envía su estimación de "distance vector" a sus vecinos
- Asincrónico
- Cuando un nodo x recibe una nueva estimación del DV de su vecino, actualiza su propio DV usando la ecuación de B-F:

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

- Bajo condiciones "naturales", la estimaación $D_x(y)$ *converge al menor costo* $d_x(y)$

Algoritmo Distance Vector

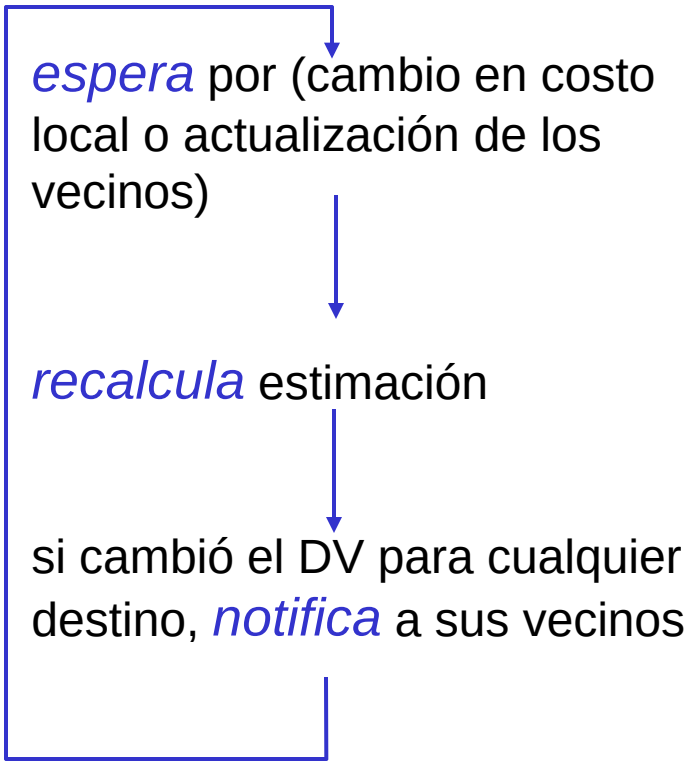
Iterativo, asincrónico:
cada iteración local causada por:

- cambio en el costo de enlaces local
- mensaje de actualización del DV de un vecino

Distribuido:

- cada nodo notifica a sus vecinos *solo* cuando cambia su DV
  - vecinos notifican luego a sus vecinos si es necesario

Cada nodo:



Algoritmo Distance Vector

en todos los nodos, X:

```
1  Initialization:
2   for all destinations y in N:
3      D_X(y) = c(X,y) /* if y is not a neighbor c(X,y) = ∞ */
4   for each neighbor w
5      D_X(y) = ∞ for all destinations y in N
5   for each neighbor w
6      send distance vector D_X = [D_X(y): y in N] to w
7
9  loop
10    wait (until I see a link cost change to some neighbor w
11          or until I receive DV update from neighbor w)
12
13   for each y in N:
14     D_X(y) = min {c(X,v) + D_X(y)}
15
16   if D_X(y) changed for any destination y
17     send distance vector D_X = [D_X(y): y in N] to w
18
19  forever
```

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

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

node x table

from \ cost to	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

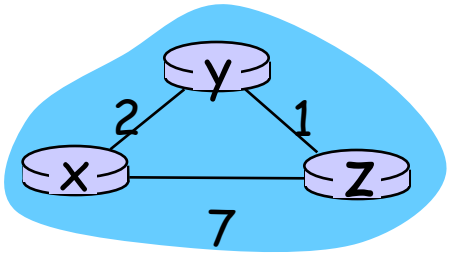
from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

node y table

from \ cost to	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

node z table

from \ cost to	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0



time

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

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

node x table

from	cost to x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

from	cost to x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

from	cost to x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

node y table

from	cost to x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

from	cost to x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

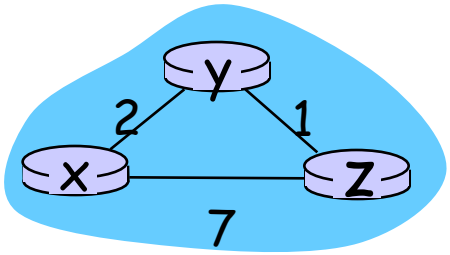
from	cost to x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

node z table

from	cost to x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

from	cost to x	y	z
x	0	2	7
y	2	0	1
z	3	1	0

from	cost to x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

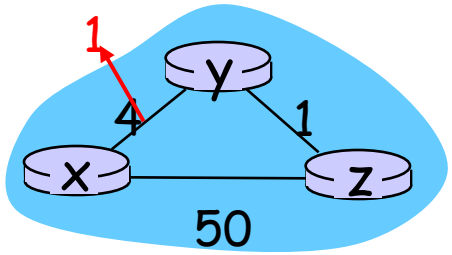


time

Distance Vector: cambios en los costos

Cambia el costo de un enlace:

- nodo detecta el cambio
- actualiza información de routing, recalcula distance vector
- si cambia DV, notifica a vecinos



"las buenas noticias viajan rápido"

En tiempo t_0, y detecta el cambio de costo, actualiza su DV, e informa a sus vecinos.

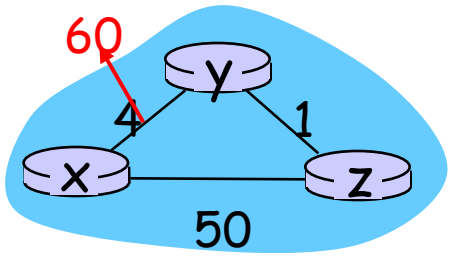
En tiempo t_1, z recibe la actualización desde y, actualiza su tabla. Calcula el nuevo cost mínimo a x, envía su DV a los vecinos.

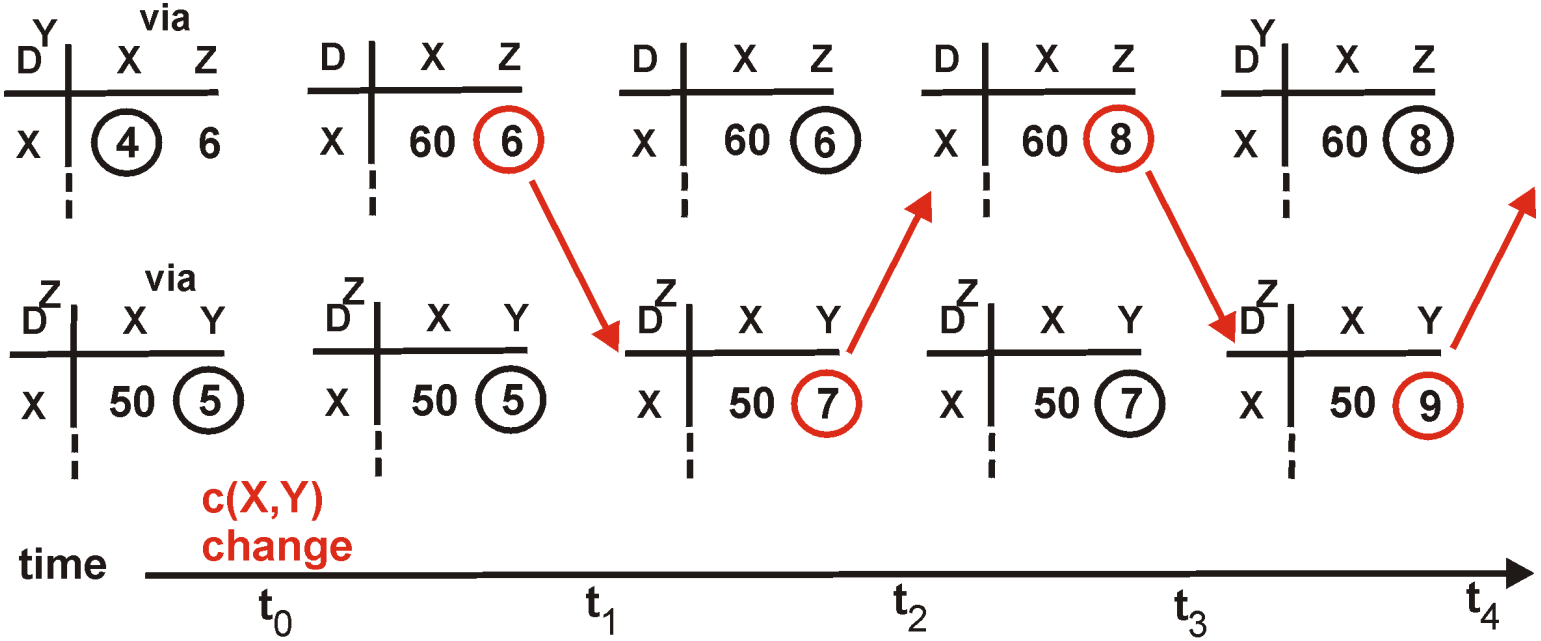
En tiempo t_2, y recibe la actualización desde z, actualiza su tabla de distancias.
El costo mínimo de y no cambia, entonces *no* envía ninguna actualización

Distance Vector: cambios en los costos

Cambia el costo de un enlace:

- buenas noticias viajan rápido
- malas noticias viajan lento - problema "count to infinity"!
- 44 iteraciones para que se estabilice el algoritmo

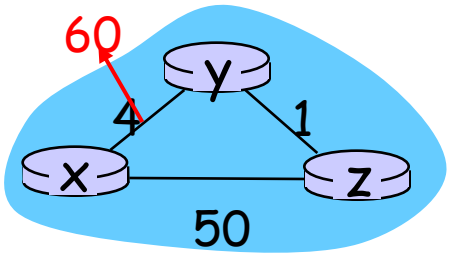


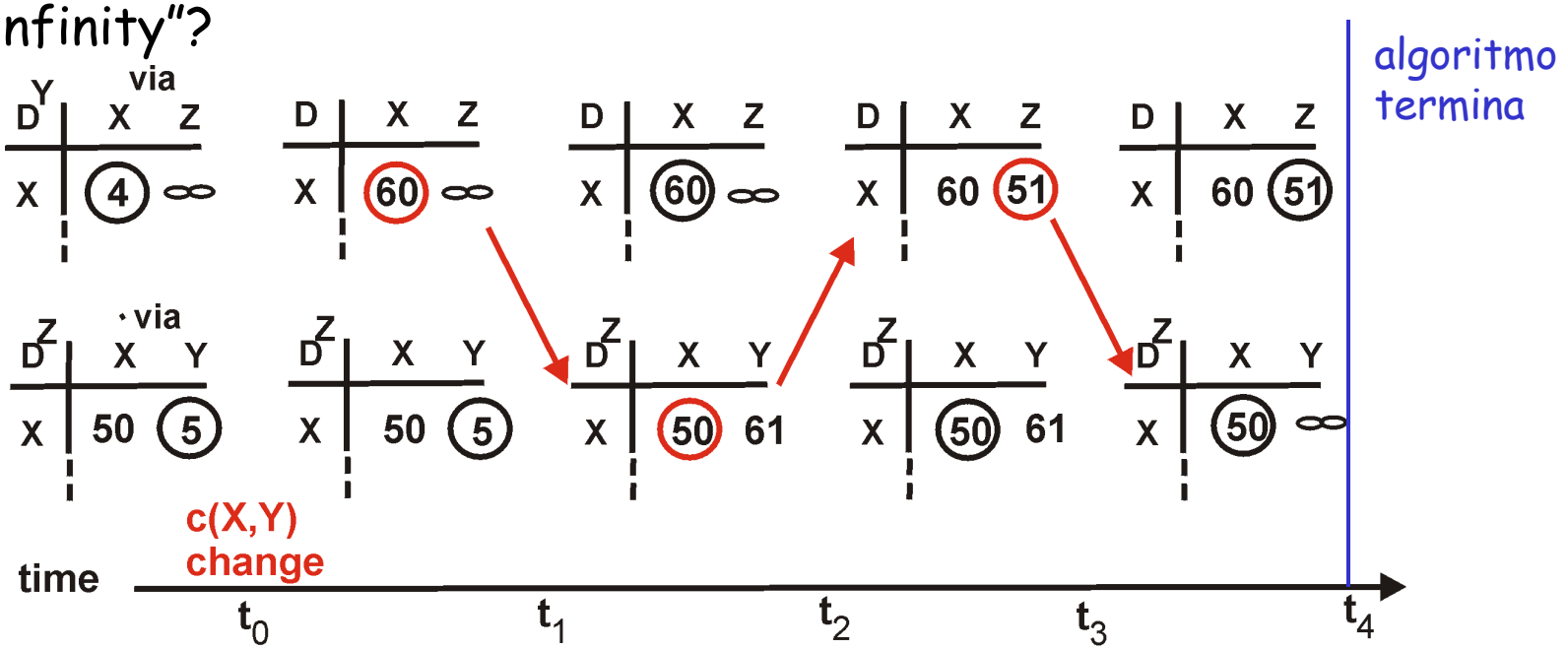


Distance Vector: cambios en los costos

"Poisoned reverse":

- Si Z encamina a través de Y para llegar a X :
  - Z avisa a Y que su distancia a X es infinito (luego Y no encaminará a X via Z)
- esto resuelve el problema de "count to infinity"?





Comparación entre algoritmos LS y DV

Compejidad de mensajes

- LS: con n nodos, E enlaces, O(nE) mensajes enviados
- DV: intercambio exclusivamente entre vecinos
  - tiempo de convergencia variable

Convergencia

- LS: algoritmo $O(n^2)$ y requiere O(nE) mensajes
  - puede tener oscilaciones
- DV: el tiempo de convergencia varía
  - puede haber "routing loops"
  - problema de "count-to-infinity"

Robustez: que pasa si un router funciona mal?

LS:

- cada nodo puede publicar costo de *enlace* incorrecto
- cada nodo calcula solamente *su propia* tabla

DV:

- Un nodo puede publicar incorrectamente el costo del *camino*
- Cada nodo usa las tablas de los otros
  - el error se propaga en la red

Cap. 4: Capa de red

- 4. 1 Introducción
- 4.2 circuitos virtuales y datagramas
- 4.3 dentro de un router
- 4.4 IP: Internet Protocol
  - formato de datagramas
  - direccionamiento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de enrutamiento
  - Link state
  - Distance Vector
  - Enrutamiento jerárquico
- 4.6 Enrutamiento en Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast y multicast

Enrutamiento Jerárquico

Hasta ahora hemos visto una idealización

- routers idénticos
- red "plana"

… *no* sucede en la práctica

escala: 100+ millones de destinos:

- no es posible almacenar todos los destinos en las tablas de routing!
- Los intercambios de información inundarían los enlaces!

autonomía administrativa

- internet = red de redes
- cada administrador quiere tener control de su propia red

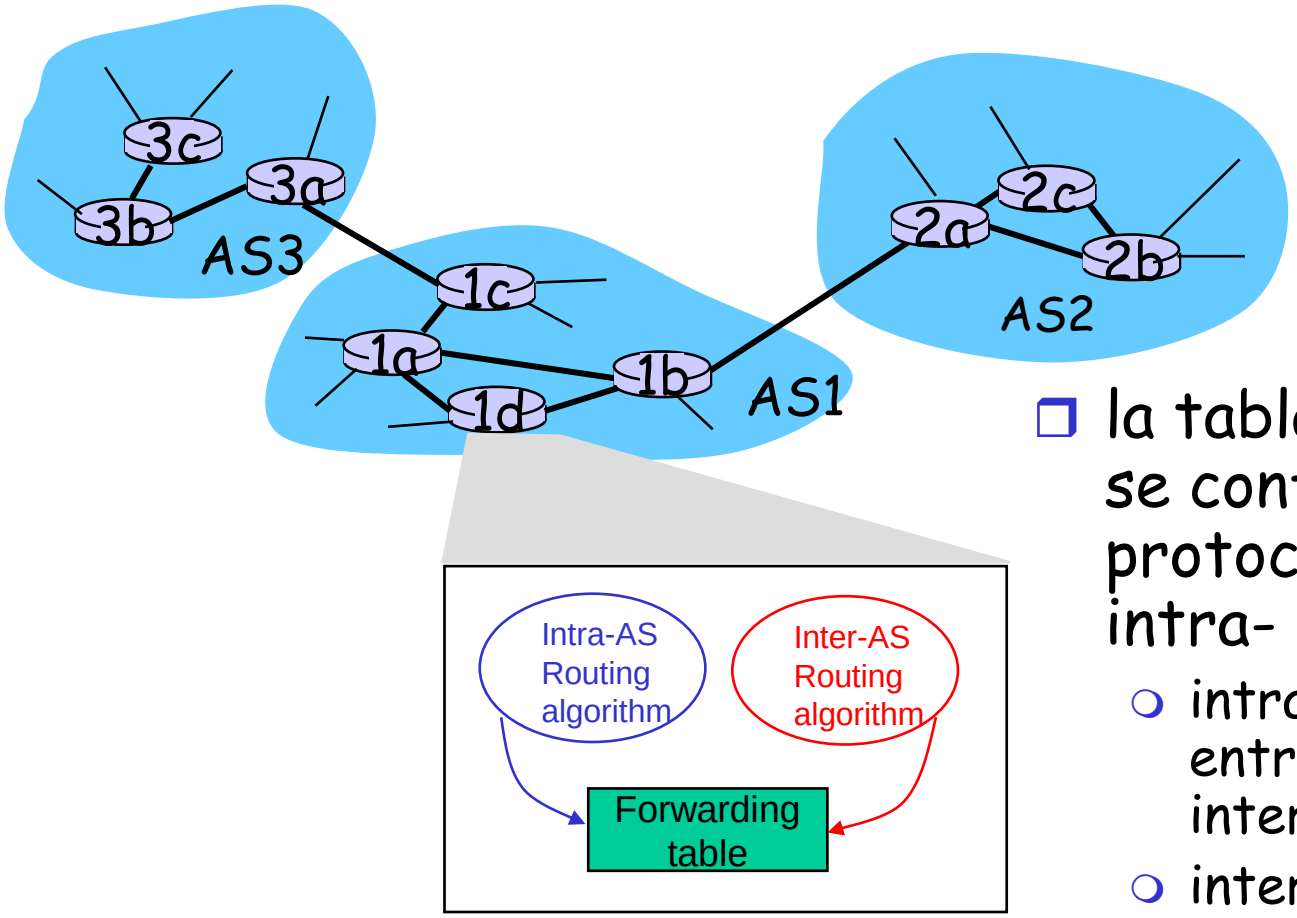
Enrutamiento Jerárquico

- agrupamiento de routers en regiones, "autonomous systems" (AS)
- routers dentro de un AS corren el mismo protocolo de routing
  - "intra-AS" routing protocol
  - routers en ASs diferentes pueden usar protocolos de routing diferentes

"gateway" router

- Enlace(s) directo(s) a router(s) en otro(s) AS(s)

Interconexión de ASs



- la tabla de forwarding se configura usando los protocolos de routing intra- e inter-AS
  - intra-AS configuran entradas para destinos internos
  - inter-AS & intra-As configuran entradas para destinos externos

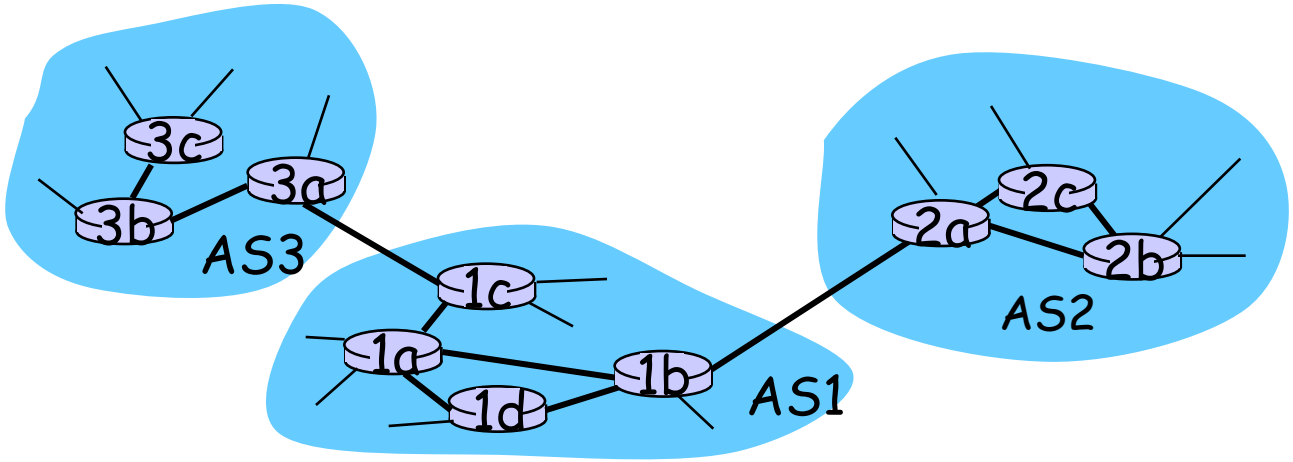
Tareas inter-AS

- supongamos que un router en AS1 recibe un datagrama destinado fuera de AS1:
  - el router debe encaminar el paquete a un "gateway", pero cual?

AS1 debe:

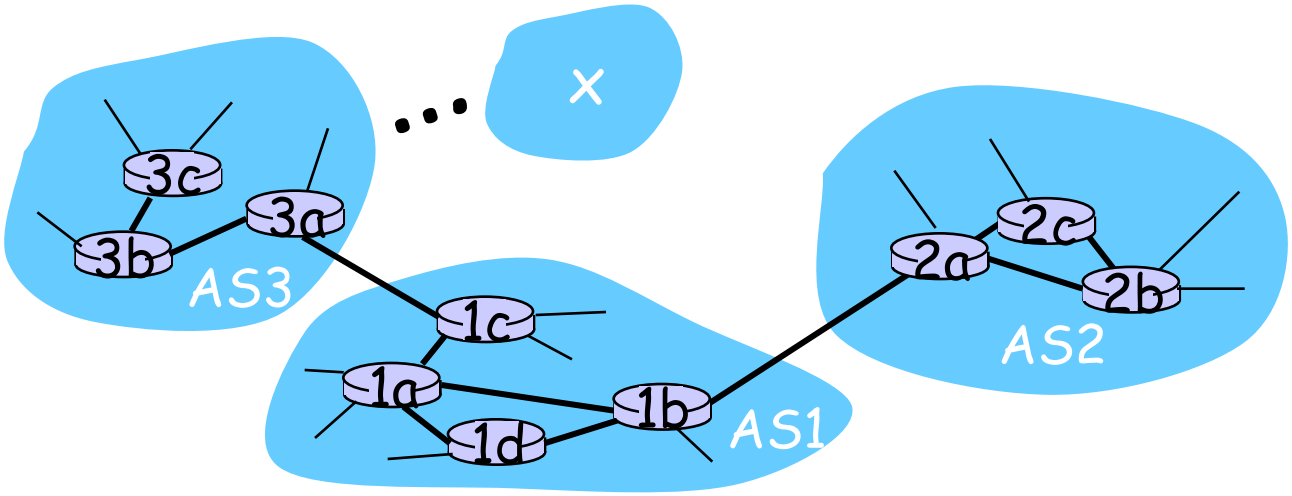
1. aprender que destinos son alcanzables por AS2, y cuales por AS3
2. Propagar esta información en AS1

Trabajo del routing inter-AS!



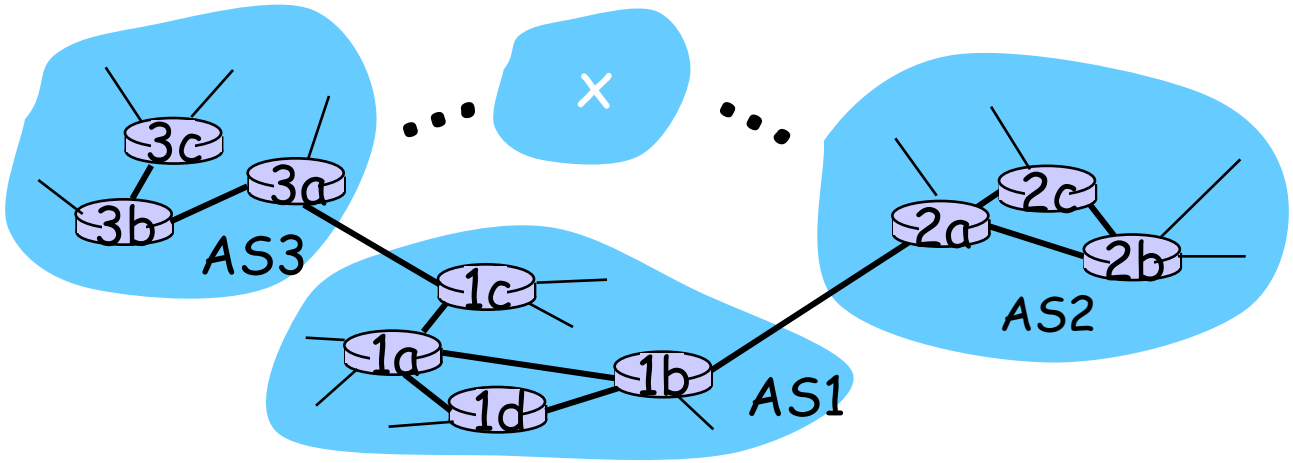
Ejemplo:
configuración de la tabla de forwarding en router 1d

- supongamos que AS1 sabe (via protocolo inter-AS) que la subred *x* es alcanzable via AS3 (gateway 1c) pero no via AS2.
- el protocolo inter-AS protocol propaga información de alcanzabilidad a los routers internos.
- Usando esta información, el router 1d determina que su interfaz *I* está en el camino de menor costo a 1c.
  - instala la entrada *(x,I)* en su tabla de forwarding



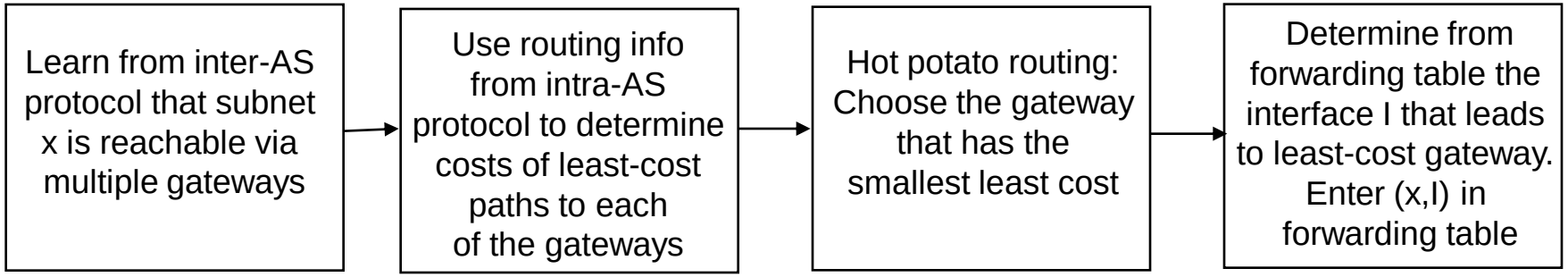
Ejemplo: elegir entre múltiples ASs

- supongamos ahora que AS1 sabe que la subred *x* es alacanzable via AS3 *y* via AS2, usando el protocolo inter-AS.
- para configurar la tabla de forwarding, el router 1d debe determinar que gateway usar para encaminar paquetes destinados a x.
  - Esto también es trabajo del routing inter-AS!



Ejemplo: elegir entre múltiples ASs

- supongamos ahora que AS1 sabe que la subred *x* es alacanzable via AS3 *y* via AS2, usando el protocolo inter-AS.
- para configurar la tabla de forwarding, el router 1d debe determinar que gateway usar para encaminar paquetes destinados a x.
  - Esto también es trabajo del routing inter-AS!
- "hot potato routing": envía el paquete por el más cercano de los dos gateways.



Cap. 4: Capa de red

- 4. 1 Introducción
- 4.2 circuitos virtuales y datagramas
- 4.3 dentro de un router
- 4.4 IP: Internet Protocol
  - formato de datagramas
  - direccionamiento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de enrutamiento
  - Link state
  - Distance Vector
  - Enrutamiento jerárquico
- 4.6 Enrutamiento en Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast y multicast

Routing intra-AS

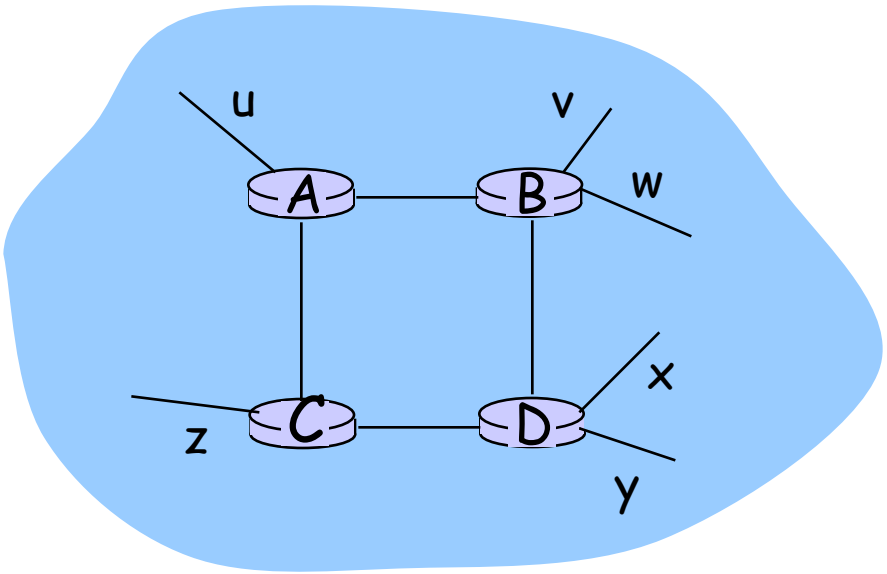
- a.k.a. Interior Gateway Protocols (IGP)
- los más comunes:
  - RIP: Routing Information Protocol
  - OSPF: Open Shortest Path First
  - IGRP: Interior Gateway Routing Protocol (propietario de Cisco)

Cap. 4: Capa de red

- 4. 1 Introducción
- 4.2 circuitos virtuales y datagramas
- 4.3 dentro de un router
- 4.4 IP: Internet Protocol
  - formato de datagramas
  - direccionamiento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de enrutamiento
  - Link state
  - Distance Vector
  - Enrutamiento jerárquico
- 4.6 Enrutamiento en Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast y multicast

RIP (Routing Information Protocol)

- algoritmo distance vector
- incluido en la distribución de BSD-UNIX en 1982
- métrica de distance: # of hops (máx = 15 hops)



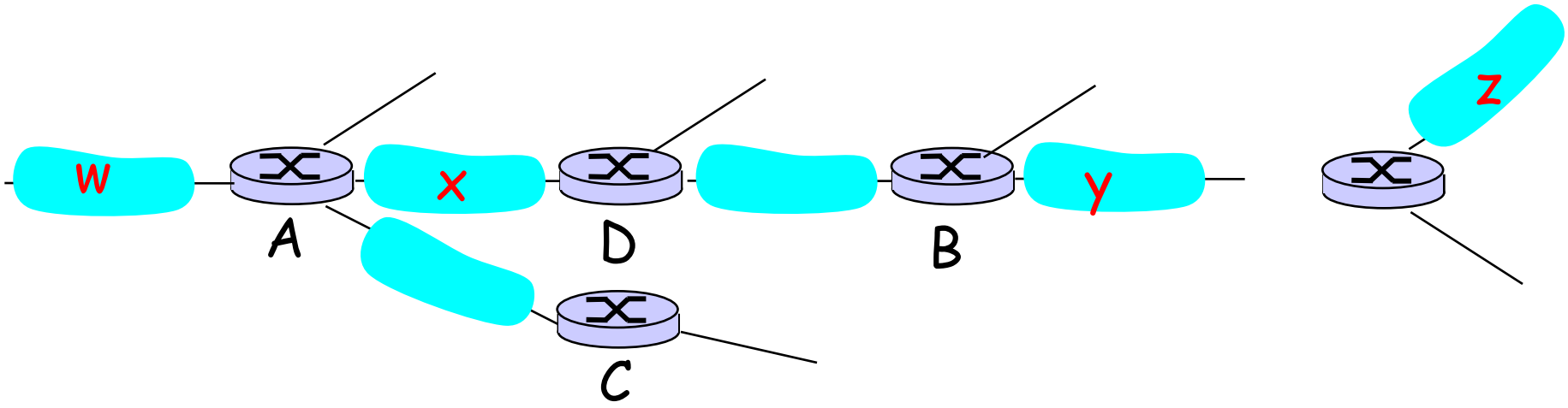
del router A a subredes:

destino	hops
u	1
v	2
w	2
x	3
y	3
z	2

RIP advertisements

- *distance vectors:* intercambiados entre vecinos cada 30 segs. via Response Message (también llamado advertisement)
- cada advertisement: lista de hasta 25 subredes destino dentro del AS

RIP: Ejemplo



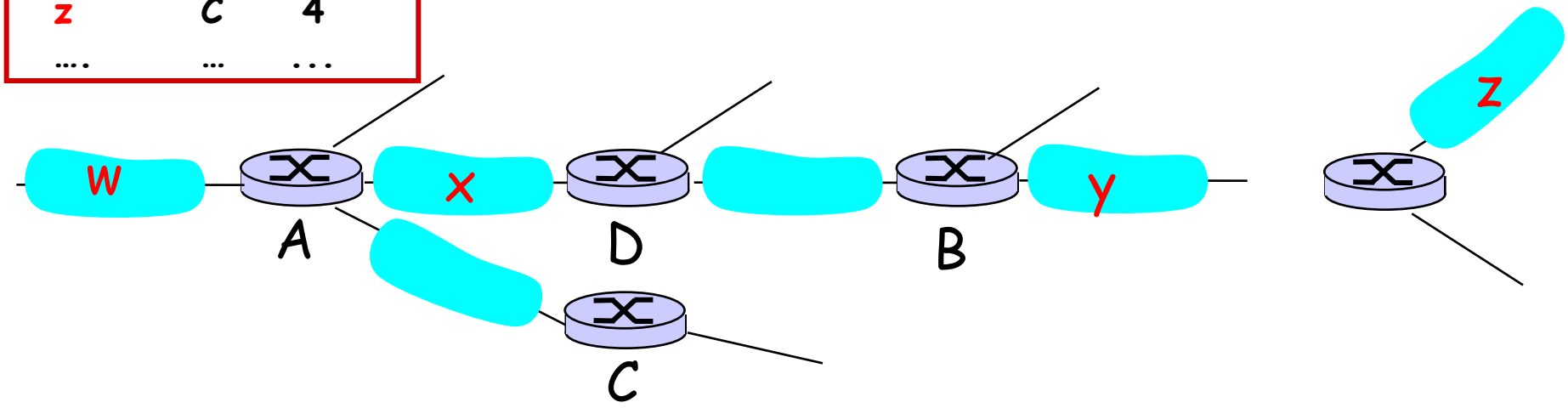
Destination Network	Next Router	Num. of hops to dest.
w	A	2
y	B	2
z	B	7
x	--	1
….	….	….

Routing/Forwarding table in D

RIP: Ejemplo

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

Advertisement from A to D



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

Routing/Forwarding table in D

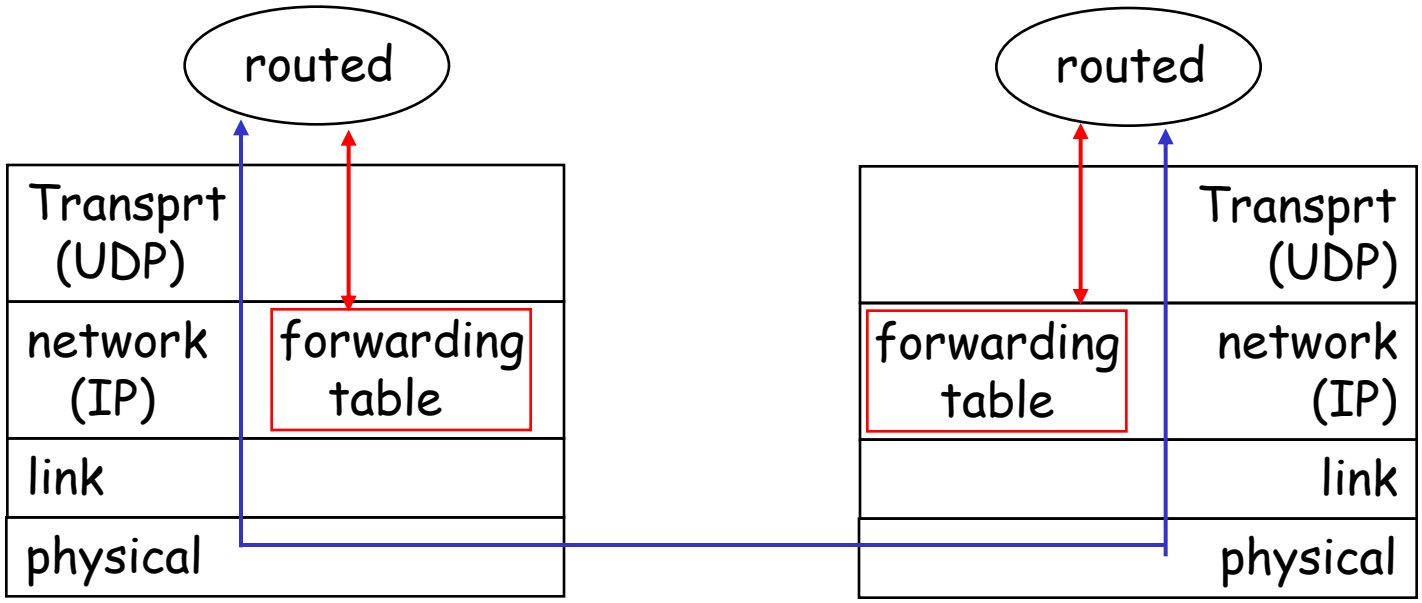
RIP: Link Failure & Recovery

Si no se recibe un advertisement pasados 180 segs. --> vecino/enlace es declarado "muerto"

- se invalidan las rutas via este vecino
- se envían nuevos advertisements a vecinos...
- ...que a su vez envían nuevos advertisements (si hay cambios en las tablas)
- Fallo en enlace se propaga a toda la red, rápidamente (?)
- *poison reverse* usado para prevenir loops (ping-pong); distamcia infinita = 16 hops)

RIP: procesamiento de la tabla

- La tabla de enrutamiento de RIP es gestionada por un proceso de capa de aplicación, llamado route-d (daemon)
- los advertisements se envían en paquetes UDP



Cap. 4: Capa de red

- 4. 1 Introducción
- 4.2 circuitos virtuales y datagramas
- 4.3 dentro de un router
- 4.4 IP: Internet Protocol
  - formato de datagramas
  - direccionamiento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de enrutamiento
  - Link state
  - Distance Vector
  - Enrutamiento jerárquico
- 4.6 Enrutamiento en Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast y multicast

OSPF (Open Shortest Path First)

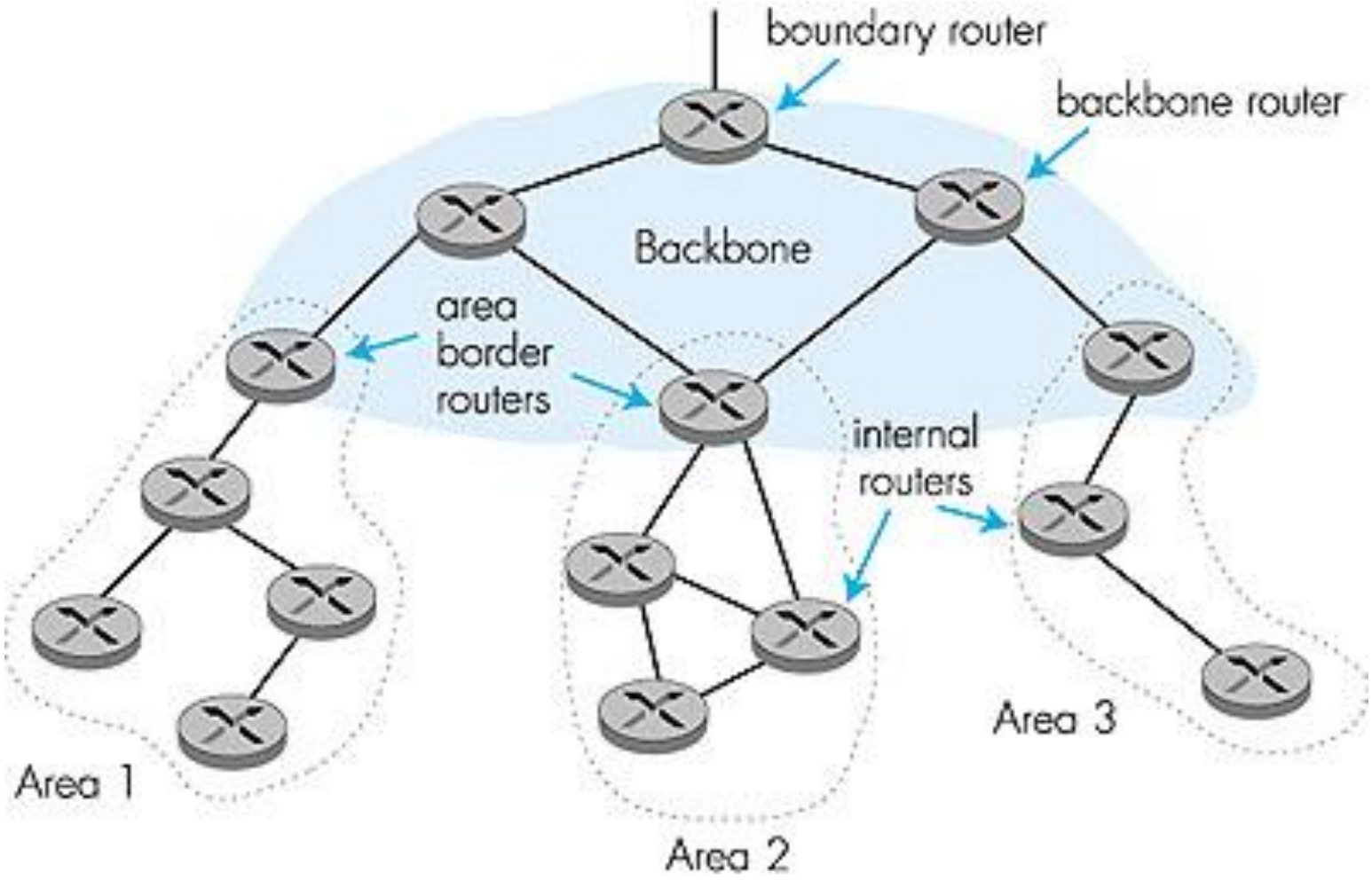
- "open": disponible públicamente
- usa algoritmo Link State
  - diseminación de paquetes LS
  - mapa de la topología en cada nodo
  - Cómputo de rutas usando el algoritmo de Dijkstra

- advertisement de OSPF transporta una entrada para cada router vecino
- los advertisements son diseminados a todo el AS (via flooding)
  - los mensajes OSPF son transportados directamente sobre IP (en lugar de TCP or UDP)

Carácterísticas "avanzadas" de OSPF (no en RIP)

- seguridad: todos los mensajes OSPF son autenticados (para prevenir intrusiones maliciosas)
- se admiten múltiples caminos de igual costo (solo uno en RIP)
- para cada enlace, métricas de costo diferentes según TOS (ej., el costo de un enlace satelital se configura "bjo" para best effort; "alto" para tiempo real)
- Soporte integrado uni y multicast:
  - Multicast OSPF (MOSPF) usa la misma base de datos de topología que OSPF
- OSPF jerárquico en dominios grandes.

OSPF Jerárquico



OSPF Jerárquico

- jerarquía de dos niveles: área local, backbone.
  - Link-state advertisements solo en el área
  - cada nodo conoce la topología detallada del área, pero solo resúmenes de las subredes en otras áreas.
- *area border routers:* "sumarizan" distancias a redes en el área propia, y lo publican hacia los otros Area Border routers.
- *backbone routers:* OSPF limitado al backbone.
- *boundary routers:* conectan con otros ASs (gateways o routers de borde).

Cap. 4: Capa de red

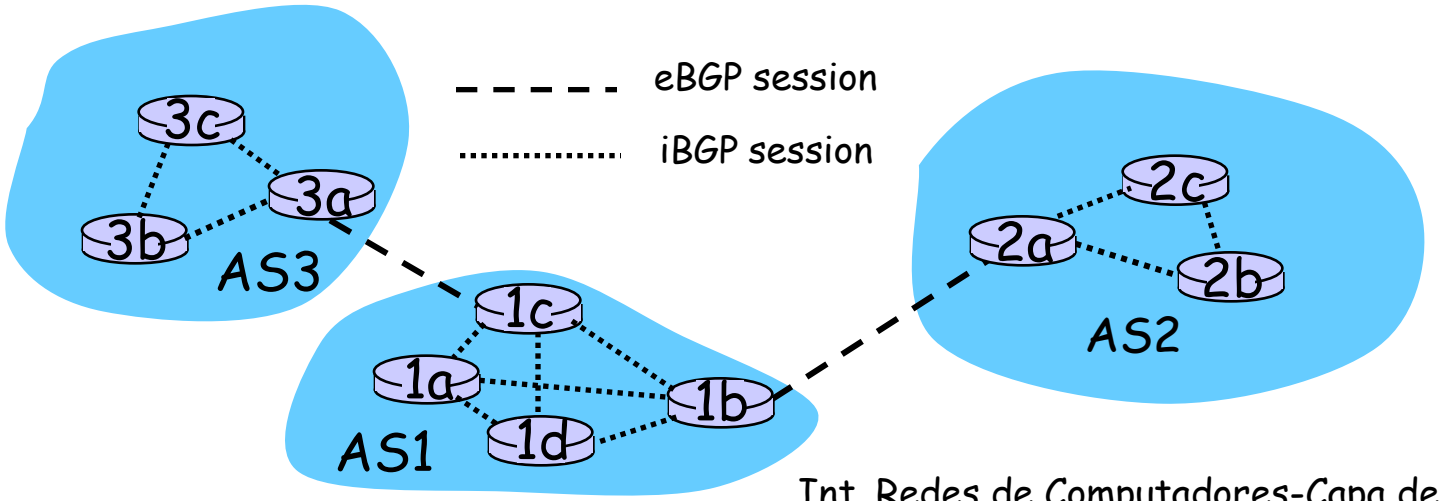
- 4. 1 Introducción
- 4.2 circuitos virtuales y datagramas
- 4.3 dentro de un router
- 4.4 IP: Internet Protocol
  - formato de datagramas
  - direccionamiento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de enrutamiento
  - Link state
  - Distance Vector
  - Enrutamiento jerárquico
- 4.6 Enrutamiento en Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast y multicast

Inter-AS routing en Internet: BGP

- BGP (Border Gateway Protocol): estándar *de facto*
- BGP provee mecanismos para:
  1. Obtener información de alcanzabilidad de subredes de los ASs vecinos.
  2. Propaga información de alcanzabilidad a los routers internos del AS.
  3. Determina que rutas son "buenas" basadas en la información de alcanzabilidad y las políticas de enrutamiento.
- permite informar la alcanzabilidad de subredes al resto de Internet: *"aquí estoy"*

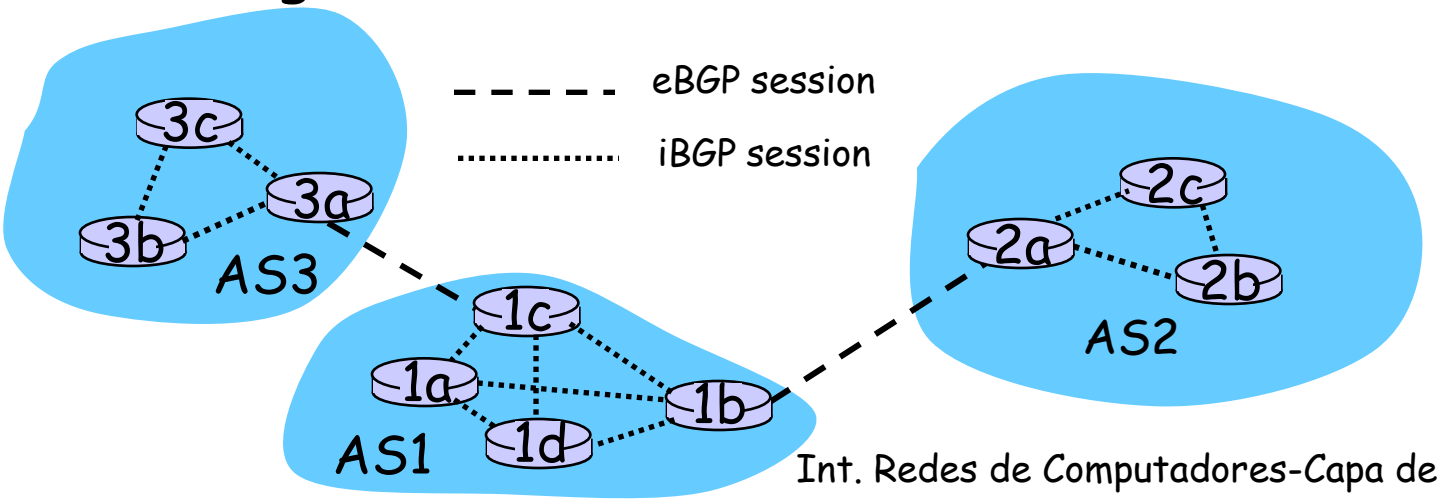
BGP: conceptos básicos

- pares de routers (BGP peers) intercambian información de routing sobre conexiones TCP semi-permanentes: sesiones BGP
  - Las sesiones BGP no se corresponden necesariamente con enlaces físicos.
- cuando AS2 publica un prefijo a AS1:
  - AS2 *"promete"* encaminar datagramas para ese prefijo.
  - AS2 puede agregar prefijos es sus publicaciones



Distribución de la información de alcanzabilidad

- AS3 envía la información de alcanzabilidad a AS1 usando una sesión eBGP entre 3a y 1c.
  - 1c puede usar iBGP para distribuir la información de prefijos a los routers en AS1
  - 1b puede re-publicar la información hacia AS2 usando la sesión eBGP 1b-2a
- Cuando un router aprende un prefijo nuevo, crea una entrada para este prefijo en la tabla de forwarding.



Path attributes & rutas BGP

- Las publicaciones de prefijos incluyen atributos BGP.
  - prefjo + atributos = "rutas"
- dos atributos importantes:
  - AS-PATH: contiene la lista de ASs que ha atravesado la publicación de un prefijo: ej., AS 67, AS 17
  - NEXT-HOP: indica el router específico en el próximo AS (pues puede haber múltiples enlaces entre ASs).
- Cuando un router de borde recibe una publicación, usa su "import policy" para aceptar/rechazar.

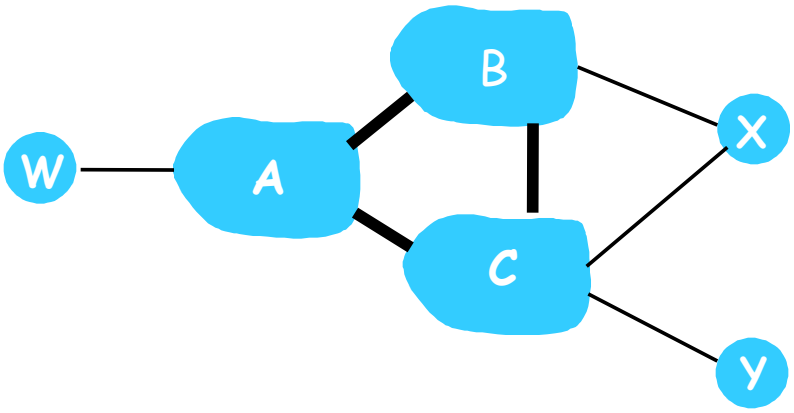
BGP route selection

- un router puede aprender más de una ruta para un prefijo dado: se necesita un proceso de selección.
- reglas de eliminación:
  1. atributo "local preference": política de decisión
  2. shortest AS-PATH
  3. closest NEXT-HOP router: hot potato routing
  4. criterios adicionales

Mensajes BGP

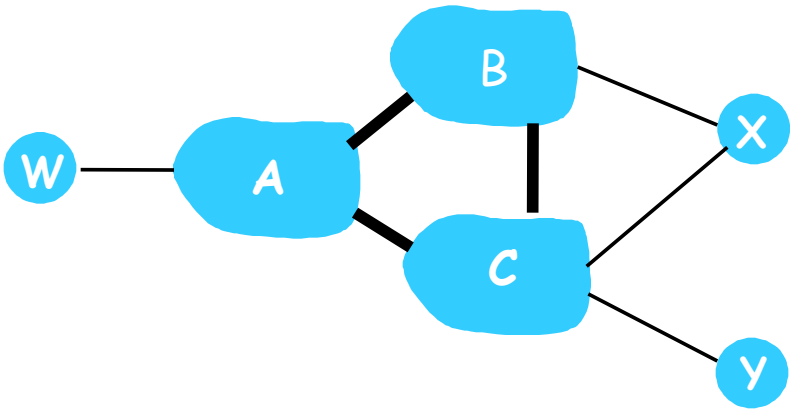
- los mensajes BGP se intercambian usando TCP.
- mensajes BGP:
  - OPEN: abre conexión TCP con "peer" y autentica al que envía
  - UPDATE: publica nuevos caminos (o da de baja otros)
  - KEEPALIVE: mantiene la conexión viva en ausencia de UPDATES; se usa también como ACK del OPEN
  - NOTIFICATION: reporta errores en mensaje previo; también se usa para cerrar conexión

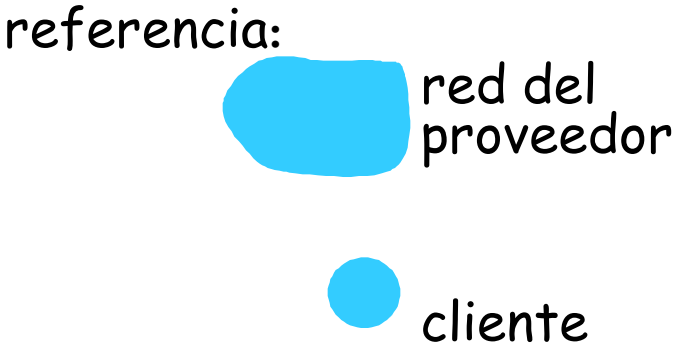
BGP routing policy



- A,B,C son redes de proveedores
- X,W,Y son clientes
- X es dual-homed: conectado a dos proveedores
  - X no permite enrutar desde B via X hacia C…
  - … luego X no va a publicar a B un ruta hacia C

BGP routing policy





- A publica camino AW a B
- B publica camino BAW a X
- debería B publicar camino BAW a C?
  - No! B no tiene "retorno" por enrutar CBAW dado que ni W ni C son sus clientes
  - B quiere forzar que C enrute hacia w via A
  - B quiere enrutar *solo* desde/hacia sus clientes!

Por qué Intra- e Inter-AS routing ?

Policy:

- Inter-AS: los administradores quieren controlar como se enruta su tráfico, y quien usa el AS como tránsito.
- Intra-AS: administración única, no se necesitan políticas

Escala:

- enrutamiento jerárquico reduce el tamaño de las tablas y de la información de actualización de enrutamiento

Performance:

- Intra-AS: enfocado en performance
- Inter-AS: políticas son más importantes que performance