

## 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

Int. Redes de Computadores-Capa de Red 4-80

---

---

---

---

---

---

---

---

## 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  $x$

Int. Redes de Computadores-Capa de Red 4-81

---

---

---

---

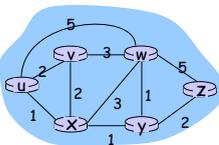
---

---

---

---

## Ejemplo Bellman-Ford



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

La ecuación de B-F dice:

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

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

Int. Redes de Computadores-Capa de Red 4-82

---

---

---

---

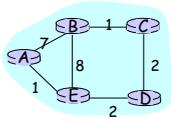
---

---

---

---

## Tabla (vector) de distancias: ejemplo



$$D_E(C,D) = c(E,D) + \min_w \{D_E(C,w)\} = 2+2 = 4$$

$$D_E(A,D) = c(E,D) + \min_w \{D_E(A,w)\} = 2+3 = 5$$

$$D_E(A,B) = c(E,B) + \min_w \{D_E(A,w)\} = 8+6 = 14$$

loop!

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

---

---

---

---

---

---

---

---

## Tabla de distancias -> forwarding

		cost to destination via			Outgoing link to use, cost	
		A	B	D		
destination	A	1	14	5	A	A,1
	B	7	8	5	B	D,5
	C	6	9	4	C	D,4
	D	4	11	2	D	D,4

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"  $D_x = [D_x(y): y \in N]$
- Nodo x también mantiene el vector de distancia hacia sus vecinos
  - Para cada vecino v, x mantiene  $D_v = [D_v(y): y \in N]$

---

---

---

---

---

---

---

---

## Algoritmo Distance Vector

### Idea básica:

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

$$D_x(y) \leftarrow \min_v \{c(x,v) + D_v(y)\} \text{ para cada nodo } y \in N$$
- Bajo condiciones "naturales", la estimación  $D_x(y)$  converge al menor costo  $d_x(y)$

Int. Redes de Computadores-Capa de Red 4-86

---

---

---

---

---

---

---

---

## Algoritmo Distance Vector

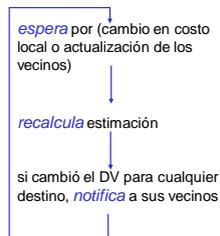
### Iterativo, asíncrono:

- 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:



Int. Redes de Computadores-Capa de Red 4-87

---

---

---

---

---

---

---

---

## 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  $c(X,y) = \infty$  if
- 4 for each neighbor  $w$
- 5  $D_x(y) = \infty$  for all destinations  $y$  in  $N$
- 5 for each neighbor  $w$
- 6 send distance vector  $D_x = [D_x(y): y \text{ in } N]$  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  $D_x(y) = \min \{c(X,v) + D_v(y)\}$
- 15
- 16 if  $D_x(y)$  changed for any destination  $y$
- 17 send distance vector  $D_x = [D_x(y): y \text{ in } N]$  to  $w$
- 18
- 19 forever

Int. Redes de Computadores-Capa de Red 4-88

---

---

---

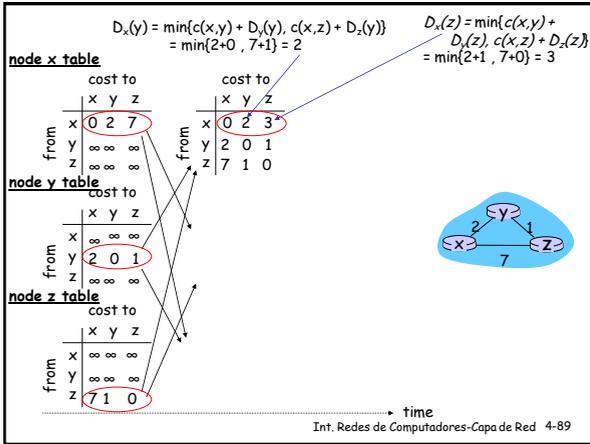
---

---

---

---

---




---

---

---

---

---

---

---

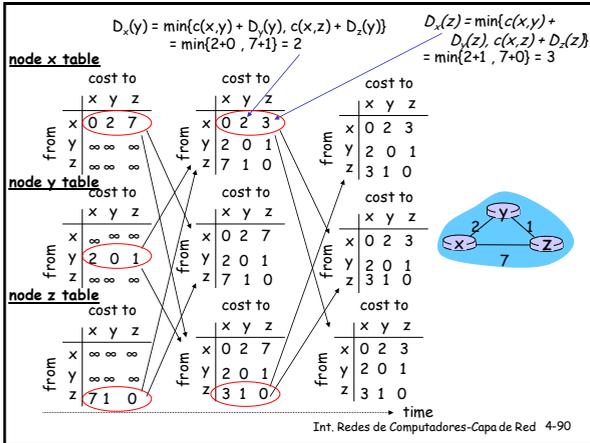
---

---

---

---

---




---

---

---

---

---

---

---

---

---

---

---

---

### 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

Int. Redes de Computadores-Capa de Red 4-91

---

---

---

---

---

---

---

---

---

---

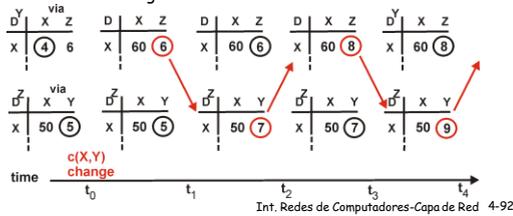
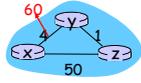
---

---

## 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




---

---

---

---

---

---

---

---

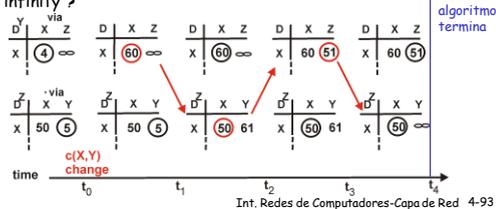
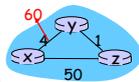
---

---

## 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

### Complejidad 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

Int. Redes de Computadores-Capa de Red 4-94

---

---

---

---

---

---

---

---

---

---

## 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

Int. Redes de Computadores-Capa de Red 4-95

---

---

---

---

---

---

---

---

## 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

Int. Redes de Computadores-Capa de Red 4-96

---

---

---

---

---

---

---

---

## 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)

Int. Redes de Computadores-Capa de Red 4-97

---

---

---

---

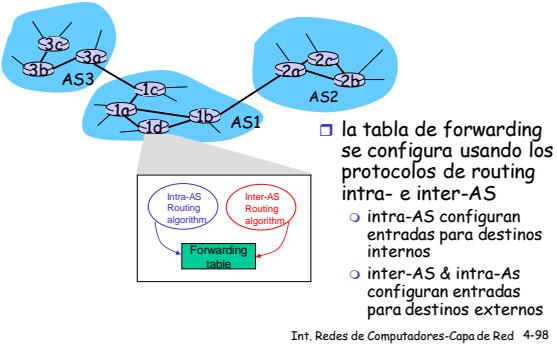
---

---

---

---

## Interconexión de ASs




---

---

---

---

---

---

---

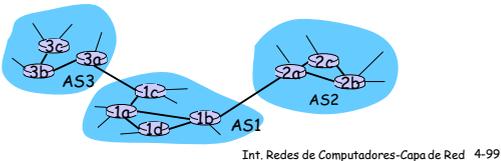
---

---

---

## 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!**




---

---

---

---

---

---

---

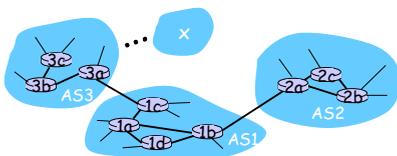
---

---

---

## 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 *x*.
  - instala la entrada (*x,I*) en su tabla de forwarding




---

---

---

---

---

---

---

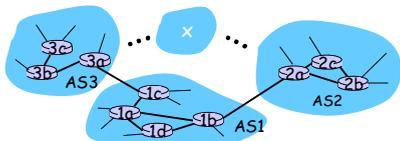
---

---

---

### Ejemplo: elegir entre múltiples ASs

- ❑ supongamos ahora que AS1 sabe que la subred *x* es alcanzable 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!



Int. Redes de Computadores-Capa de Red 4-101

---

---

---

---

---

---

---

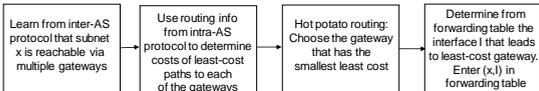
---

---

---

### Ejemplo: elegir entre múltiples ASs

- ❑ supongamos ahora que AS1 sabe que la subred *x* es alcanzable 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.



Int. Redes de Computadores-Capa de Red 4-102

---

---

---

---

---

---

---

---

---

---

## 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

Int. Redes de Computadores-Capa de Red 4-103

---

---

---

---

---

---

---

---

---

---

## 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)

Int. Redes de Computadores-Capa de Red 4-104

---

---

---

---

---

---

---

---

## 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

Int. Redes de Computadores-Capa de Red 4-105

---

---

---

---

---

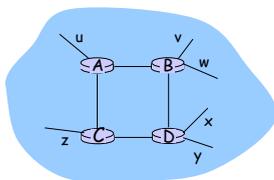
---

---

---

## 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

Int. Redes de Computadores-Capa de Red 4-106

---

---

---

---

---

---

---

---

## 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

Int. Redes de Computadores-Capa de Red 4-107

---

---

---

---

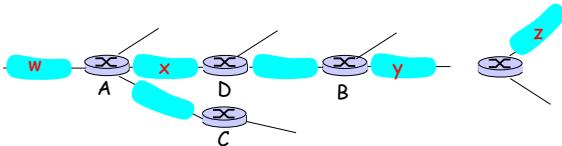
---

---

---

---

## 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

Int. Redes de Computadores-Capa de Red 4-108

---

---

---

---

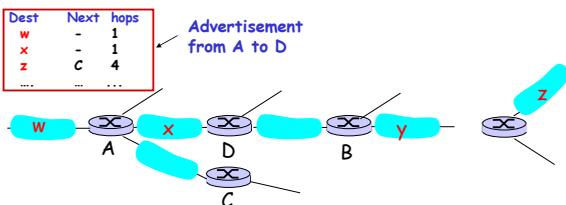
---

---

---

---

## 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	<del>B</del> A	<del>7</del> 5
X	--	1
...	...	...

Routing/Forwarding table in D

Int. Redes de Computadores-Capa de Red 4-109

---

---

---

---

---

---

---

---

## 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); distancia infinita = 16 hops)

Int. Redes de Computadores-Capa de Red 4-110

---

---

---

---

---

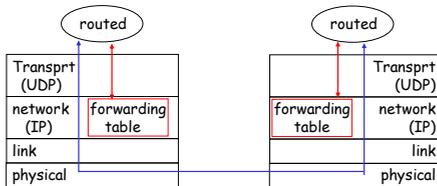
---

---

---

## 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



Int. Redes de Computadores-Capa de Red 4-111

---

---

---

---

---

---

---

---

## 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

Int. Redes de Computadores-Capa de Red 4-112

---

---

---

---

---

---

---

---

## 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)

Int. Redes de Computadores-Capa de Red 4-113

---

---

---

---

---

---

---

---

## Características "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.

Int. Redes de Computadores-Capa de Red 4-114

---

---

---

---

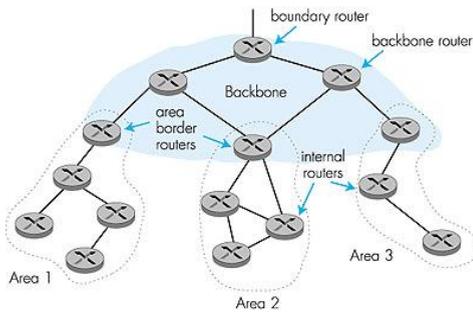
---

---

---

---

## OSPF Jerárquico



Int. Redes de Computadores-Capa de Red 4-115

---

---

---

---

---

---

---

---

## 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).

Int. Redes de Computadores-Capa de Red 4-116

---

---

---

---

---

---

---

---

## 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

Int. Redes de Computadores-Capa de Red 4-117

---

---

---

---

---

---

---

---

## 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*"

Int. Redes de Computadores-Capa de Red 4-118

---

---

---

---

---

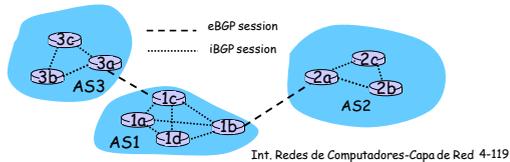
---

---

---

## 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



---

---

---

---

---

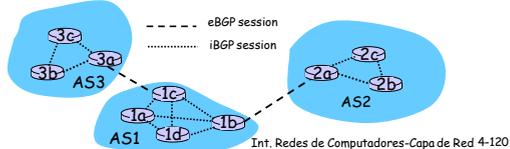
---

---

---

## 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.
  - prefijo + atributos = "rutas"
- dos atributos importantes:
  - **AS-PATH**: contiene la lista de ASs que ha atravesado la publicación de un prefijo: e.j., 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.

Int. Redes de Computadores-Capa de Red 4-121

---

---

---

---

---

---

---

---

## 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

Int. Redes de Computadores-Capa de Red 4-122

---

---

---

---

---

---

---

---

## 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

Int. Redes de Computadores-Capa de Red 4-123

---

---

---

---

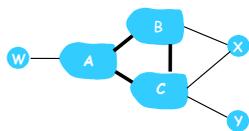
---

---

---

---

## BGP routing policy



referencia:  
● red del proveedor  
● red del cliente

- 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

Int. Redes de Computadores-Capa de Red 4-124

---

---

---

---

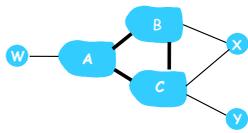
---

---

---

---

## BGP routing policy



referencia:  
red del proveedor  
cliente

- 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!

Int. Redes de Computadores-Capa de Red 4-125

---

---

---

---

---

---

---

---

## 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

Int. Redes de Computadores-Capa de Red 4-126

---

---

---

---

---

---

---

---