

Redes de computadoras

Capa de Red – Algoritmos de enrutamiento

Las diapositivas están basadas en en libro:
“Redes de Computadoras – Un enfoque descendente”
de James F. Kurose & Keith W. Ross

Algoritmos de enrutamiento

Reenvío:

Llega un paquete a un router y en base a una tabla de reenvío determina la interfaz de enlace a la que tiene que dirigir al paquete.

Algoritmos de enrutamiento:

Se ejecutan en los routers los cuales intercambian información y mediante cálculos determinan las tablas de reenvío.

El objetivo del enrutamiento es encontrar buenas rutas desde el emisor hasta el receptor a través de la red de routers.

Algoritmos de enrutamiento

El router puerta de enlace predeterminado para el host, router del primer salto, que envía un paquete será el que consideremos **router de origen**.

Del mismo modo al router puerta de enlace predeterminado para el host destino le llamaremos **router de destino**.

Enrutar un paquete de host a host pasa a ser un problema de enrutar un paquete de router a router.

Abstracción de grafos

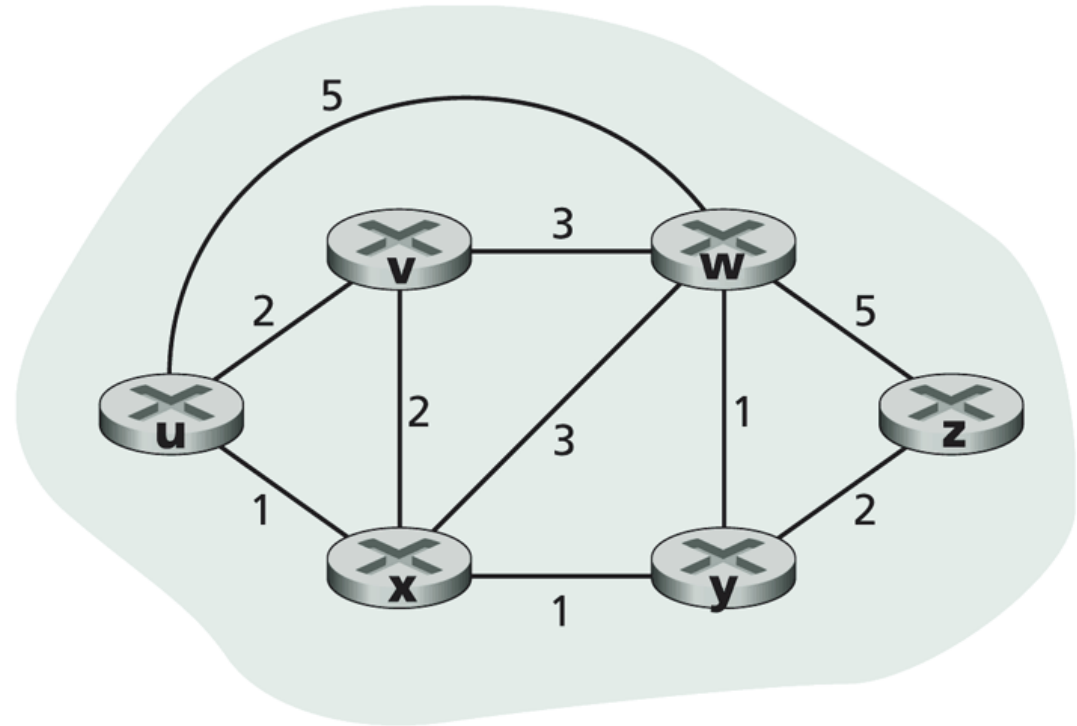
Grafo: $G = (N, E)$

N = Conjunto de enrutadores

$\{u, v, w, x, y, z\}$

E = Conjunto de enlaces

$\{(u,v), (u,x), (v,x), (v,w), (x,w),$
 $(x,y), (w,y), (w,z), (y,z)\}$



- Grafo no dirigido $(x,y) == (y,x)$

- Nodo "y" vecino de "x" si (y,x) pertenece a E

Cada enlace (arista) tiene un costo asignado, por lo que el coste de la ruta será la suma del coste de las aristas recorridas.

Algoritmos de enrutamiento

Distintos tipos:

- Algoritmo de enrutamiento global
Utiliza el conocimiento global y completo de la red.
 - **LS** Por estado de enlaces (Link status)
- Algoritmo de enrutamiento descentralizado
El costo se calcula de forma iterativa y distribuída.
 - **DV** Por vector de distancias (Distance Vector)

Enrutamiento estático y dinámico

- **Estático**, las rutas cambian muy poco. (intervención humana)
- **Dinámico**, respuesta directa a cambios en la topología o costos.
- **Sensibles a la carga**, costes varían dinámicamente en función a la congestión.

Algoritmo de enrutamiento de estado de enlaces (Link State)

La topología de la red y los costos de todos los enlaces son conocidos.

- Cada nodo difunde paquete de estado de enlace conteniendo identidades y costes de sus enlaces conectados.

Protocolo de enrutamiento OSPF de Internet

Algoritmo de Dijkstra

- Iterativo: Una iteración por cada nodo en la red
- Obtiene el costo mínimo desde un nodo a los demás

Notación:

$c(x,y)$: costo del enlace. ∞ si los nodos no son vecinos.

$D(v)$: mínimo valor calculado hasta el momento desde el origen al nodo v .

$p(v)$: Nodo predecesor en el camino del origen a v .

N' : Conjunto de nodos cuyo costo mínimo es conocido hasta el momento.

Algoritmo de Dijkstra

Inizialización:

$N' = \{u\}$

Para cada nodo v

Si v es vecino de u

entonces $D(v) = c(u,v)$

Sino $D(v) = \infty$

Loop

Encontrar w que no esté en N' tal que $D(w)$ es mínimo

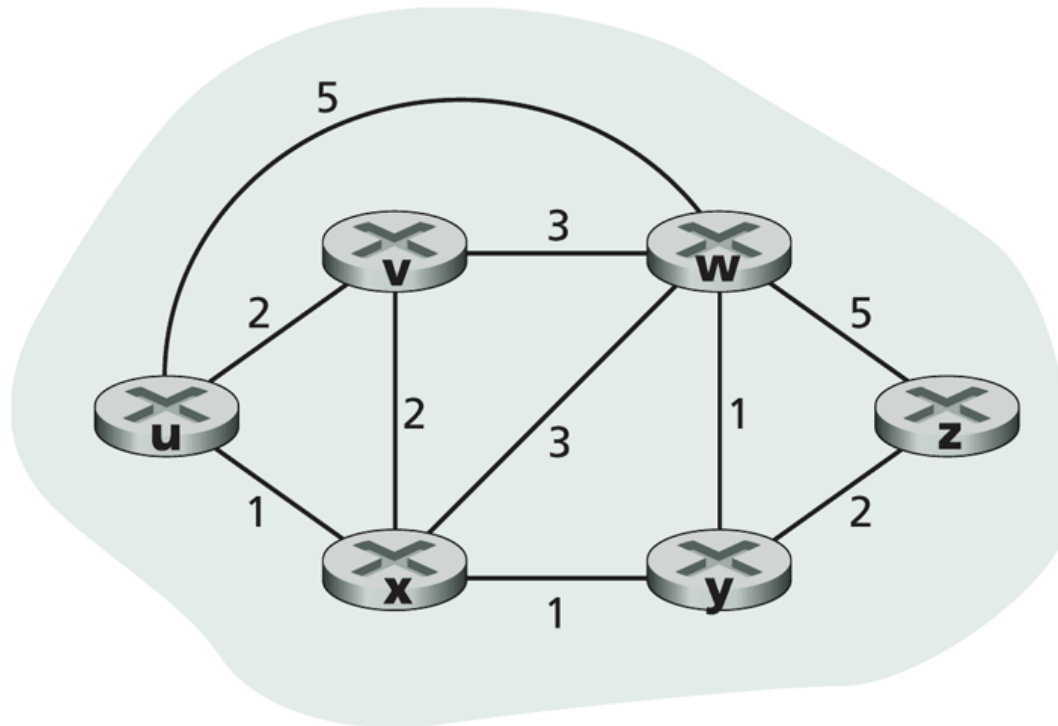
Agregar w a N'

actualizar $D(v)$ para cada v adyacente a w y que no esté en N' : $D(v) = \min(D(v), D(w) + c(w,v))$

Hasta que todos los nodos estén en N'

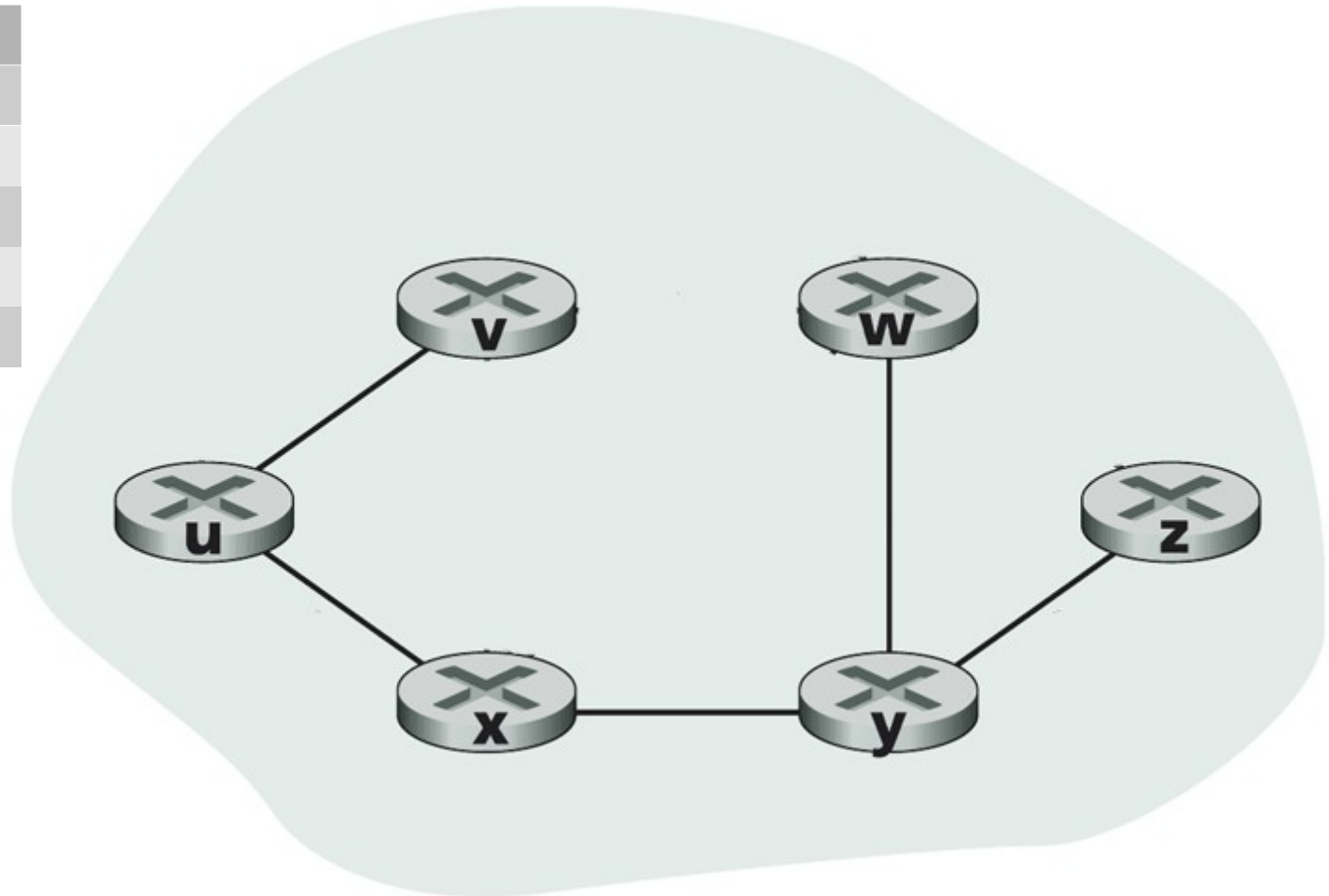
Algoritmo de Dijkstra

Rutas de costo mínimo desde u a todos los destinos posibles



Algoritmo de Dijkstra

Destino	Enlace
v	(u,v)
w	(u,x)
x	(u,x)
y	(u,x)
z	(u,x)



Algoritmo de Dijkstra

Oscilaciones de enrutamiento

Considerando la carga del enlace como el único factor que afecte el costo se podría generar una oscilación constante del enrutamiento.

Luego de el cálculo y estimación de la mejor ruta se podría volcar gran parte del volumen de tráfico hacia esta.

De este modo se iría oscilando entre una ruta y otra.

“Link Status”

Oscilaciones de enrutamiento

Ejemplo con costes de enlace no simétricos

Cada vez que se ejecuta el algoritmo de LS cambia el sentido del flujo de datos.

Para evitarlo se deja que cada router elija aleatoriamente el momento en que se envíe el paquete de estado.

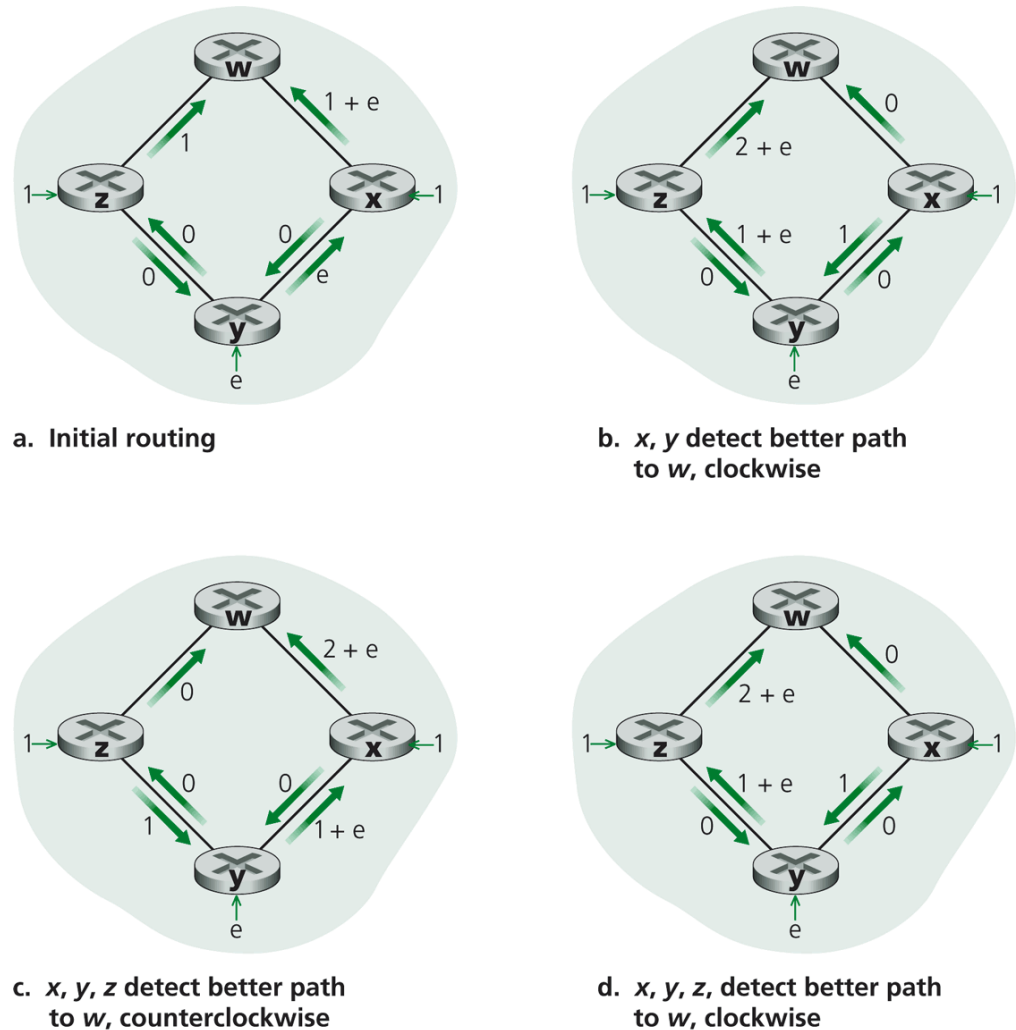


Figure 4.26 ♦ Oscillations with link-state routing

Algoritmo por Vector de Distancias

El algoritmo por vector de distancias es:

- Distribuido

Cada nodo recibe información de sus vecinos, realiza un cálculo y luego distribuye los resultados a sus vecinos.

- Iterativo

Este proceso se repite hasta que no hay más información a intercambiar.

- Asíncrono

No se requiere que todos los nodos operen sincronizados entre sí.

Ecuación de Bellman-Ford

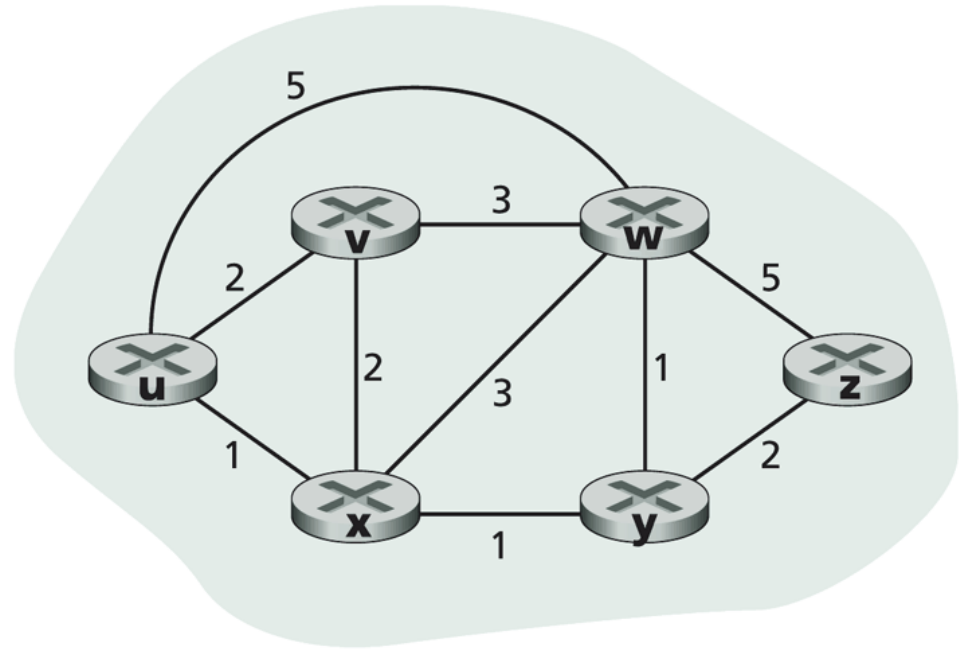
$dx(y) :=$ costo del camino de menor costo entre “x” e “y”

$$dx(y) = \min \{ c(x,v) + dv(y) \}$$

Algoritmo por Vector de Distancias

Ecuación de Bellman-Ford

$$dv(z) = 5, dx(z) = 3, dw(z) = 3$$



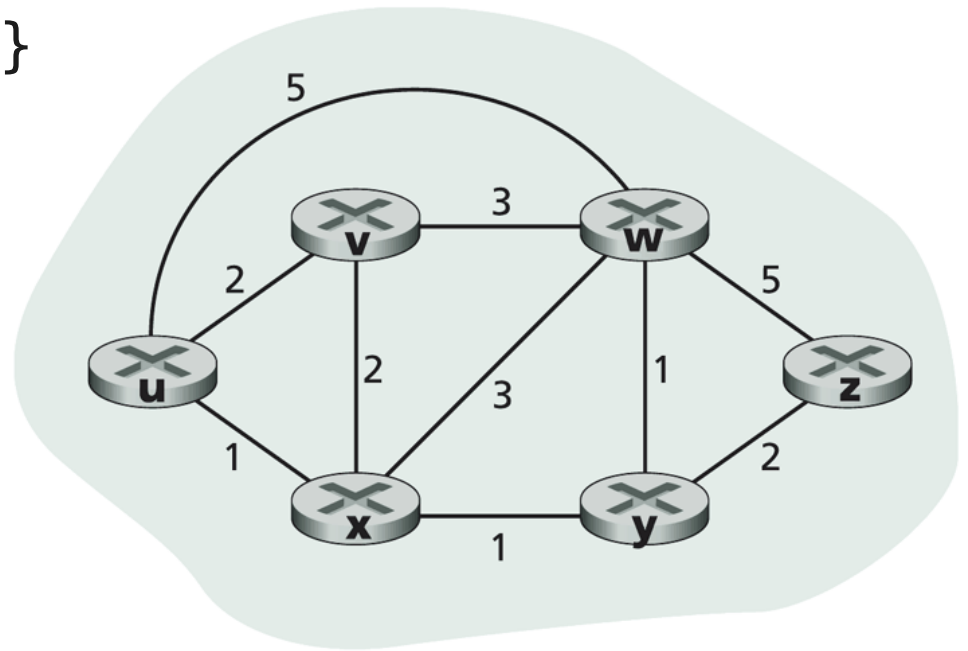
Algoritmo por Vector de Distancias

Ecuación de Bellman-Ford

$$dv(z) = 5, dx(z) = 3, dw(z) = 3$$

La ecuación de B-F dice:

$$du(z) = \min \{ c(u,v) + dv(z), \\ c(u,x) + dx(z), \\ c(u,w) + dw(z) \}$$



Algoritmo por Vector de Distancias

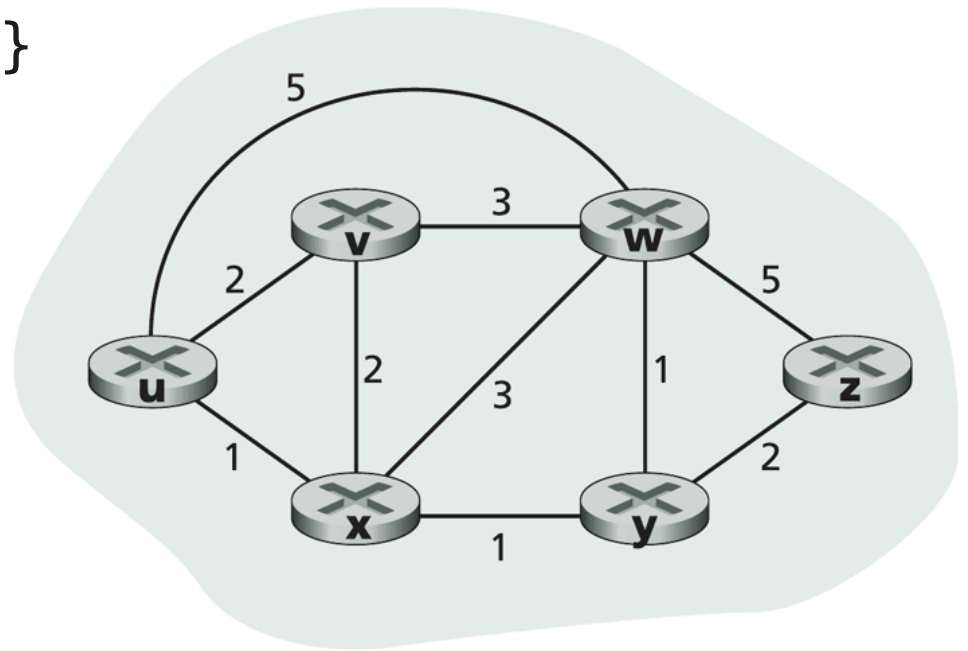
Ecuación de Bellman-Ford

$$dv(z) = 5, dx(z) = 3, dw(z) = 3$$

La ecuación de B-F dice:

$$du(z) = \min \{ c(u,v) + dv(z), \\ c(u,x) + dx(z), \\ c(u,w) + dw(z) \}$$

$$du(z) = \min \{ 2 + 5, \\ 1 + 3, \\ 5 + 3 \} = 4$$



Algoritmo por Vector de Distancias

$D_x(y)$ = Estimación del menor costo de x a y

x mantiene vector de distancia $D_x = [D_x(y): y \in N]$

Nodo x :

Conoce el costo a cada vecino v : $c(x,v)$

Mantiene los vectores distancia de los vecinos.

para cada vecino v , x mantiene $D_v = [D_v(y): y \in N]$

Algoritmo por Vector de Distancias

Idea básica:

- Cada un determinado tiempo, cada nodo envía la estimación de su vector de distancias a sus vecinos.
- Cuando x recibe un nuevo vector de distancias de un vecino, actualiza el propio con 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 ciertas condiciones no muy restrictivas, la estimación $D_x(y)$ converge al camino de menor costo $d_x(y)$

Algoritmo por Vector de Distancias

Iterativo, Asíncrono:

Cada iteración local causada por:

- Cambio en el costo local de un enlace
- Mensaje de actualización de V.D. de un vecino

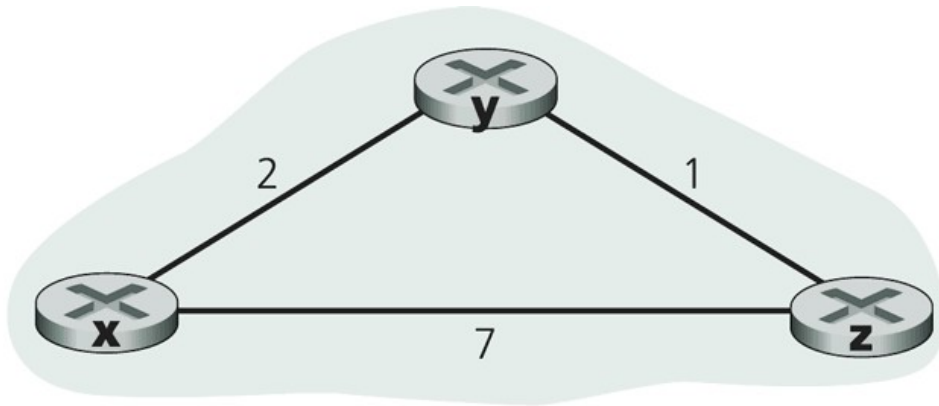
Distribuido:

- Cada nodo notifica a sus vecinos sólo cuando su V.D. cambia
- Los vecinos notifican a sus propios vecinos si es necesario

Cada nodo:

- Espera cambios en los enlaces locales o mensaje de alguno de sus vecinos
- Recalcula estimaciones
- Si algún V.D. cambia, notifica a sus vecinos

Algoritmo por Vector de Distancias



Node x table

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

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

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

Node y table

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

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

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

Node z table

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

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

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

Algoritmo por Vector de Distancias

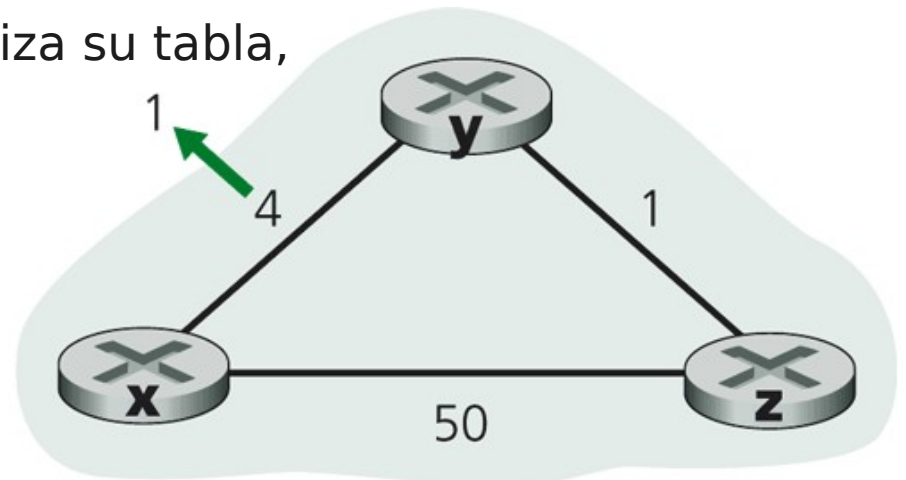
Cambio en el costo del enlace:

- Un nodo detecta un cambio en el costo del enlace
- Actualiza su información de enrutamiento y actualiza su V.D.
- Si el V.D. cambia, notifica a sus vecinos.

T0: y detecta un cambio en el costo de su enlace a x actualiza su V.D. e informa a sus vecinos.

T1: z recibe la actualización de y, actualiza su tabla, computa su nuevo costo hacia x, envía su V.D. a los vecinos.

T2: y recibe la actualización de z, actualiza su V.D. Como los costos mínimos no cambian, termina el algoritmo.



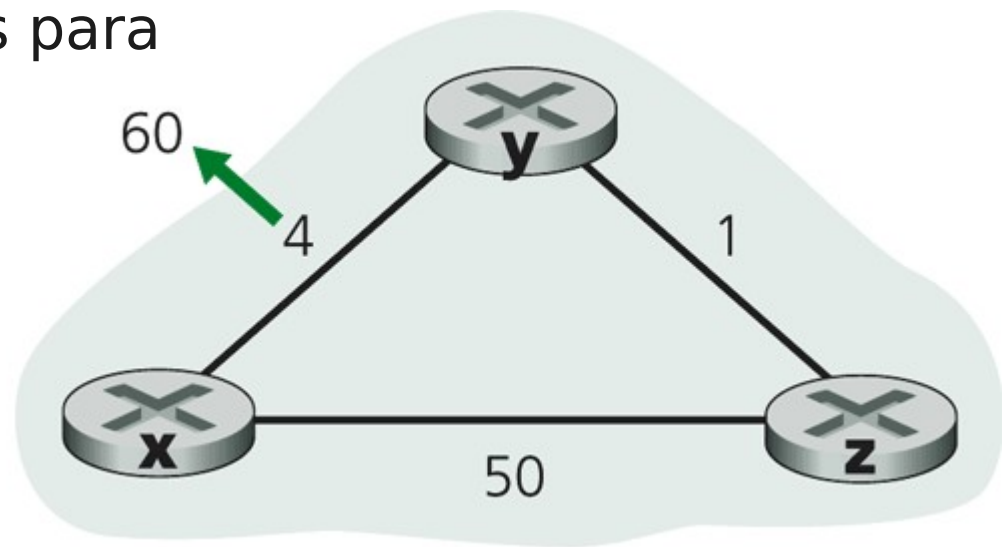
Algoritmo por Vector de Distancias

Cambio en el costo del enlace:

- Las “buenas noticias” viajan rápido
- Las “malas noticias” viajan lento

Problema del conteo infinito

- Se necesitan 44 iteraciones para que el algoritmo termine



Comparación de algoritmos LS y DV

Complejidad en mensajes:

LS: con n nodos, E enlaces, se envían $O(n * E)$ mensajes.

VD: intercambio entre vecinos

- Aunque tiempo de convergencia varía

Velocidad de convergencia:

LS: Algoritmo $O(n^2)$

VD: varía el tiempo de convergencia

(Problema del conteo infinito)

Robustés:

LS:

- Un nodo puede informar costo de enlace incorrecto
- Cada nodo computa su tabla únicamente.

VD:

- Los nodos pueden informar costo de camino incorrecto
- La tabla de cada nodo se usa por otros
(posible error de propagación)

Enrutamiento Jerárquico

Hasta ahora el estudio fue idealizado

- Todos los routers eran idénticos
- Pocos routers

Escala:

Con 200 millones de destinos:

No se puede guardar todos los destinos en tablas

El intercambio de tablas de enrutamiento sobrecargaría los enlaces

Autonomía administrativa:

Internet: red de redes

Cada administrador de la red puede querer controlar el enrutamiento en su propia red.

Enrutamiento Jerárquico

Separar enrutadores en regiones:

“Sistemas autónomos” (AS)

Los enrutadores en el mismo AS corren el mismo protocolo de enrutamiento

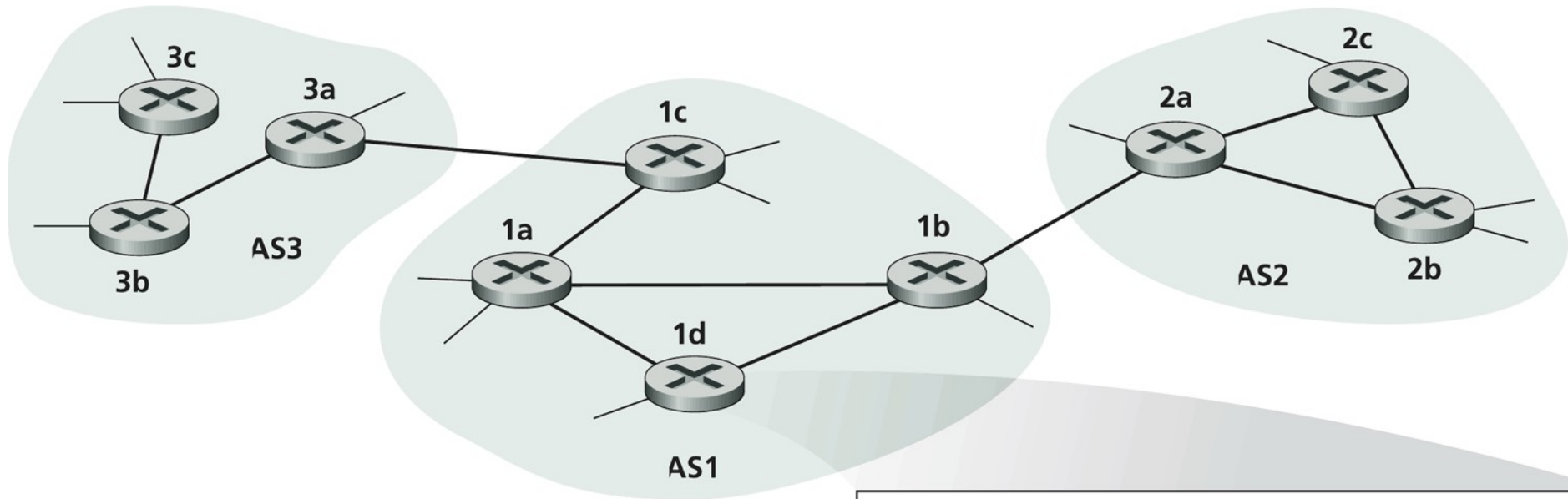
- Protocolo de enrutamiento “intra-AS”
- Routers en diferentes AS pueden correr diferentes protocolos de enrutamiento

Router de puerta de enlace

- Routers en el borde de su AS
- Tiene un enlace hacia un enrutador en otro AS

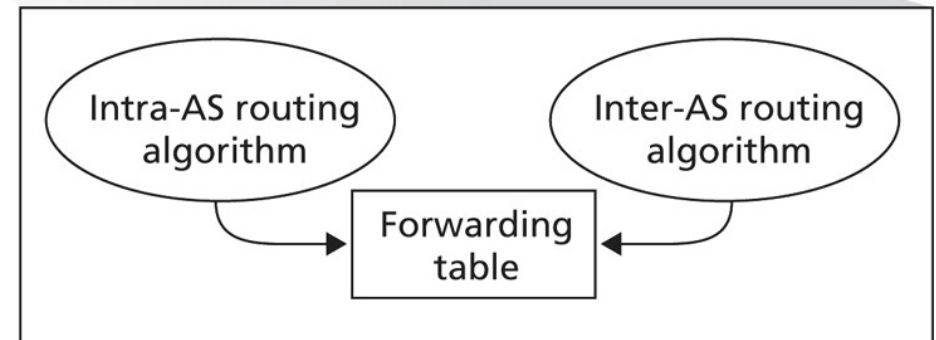
Enrutamiento Jerárquico

Sistemas autónomos conectados



Las tablas de reenvío son configuradas por protocolos de enrutamiento intra-AS e inter-AS

- Intra-AS entradas internas
- Intra-AS e Inter-AS para entradas externas



Enrutamiento Jerárquico

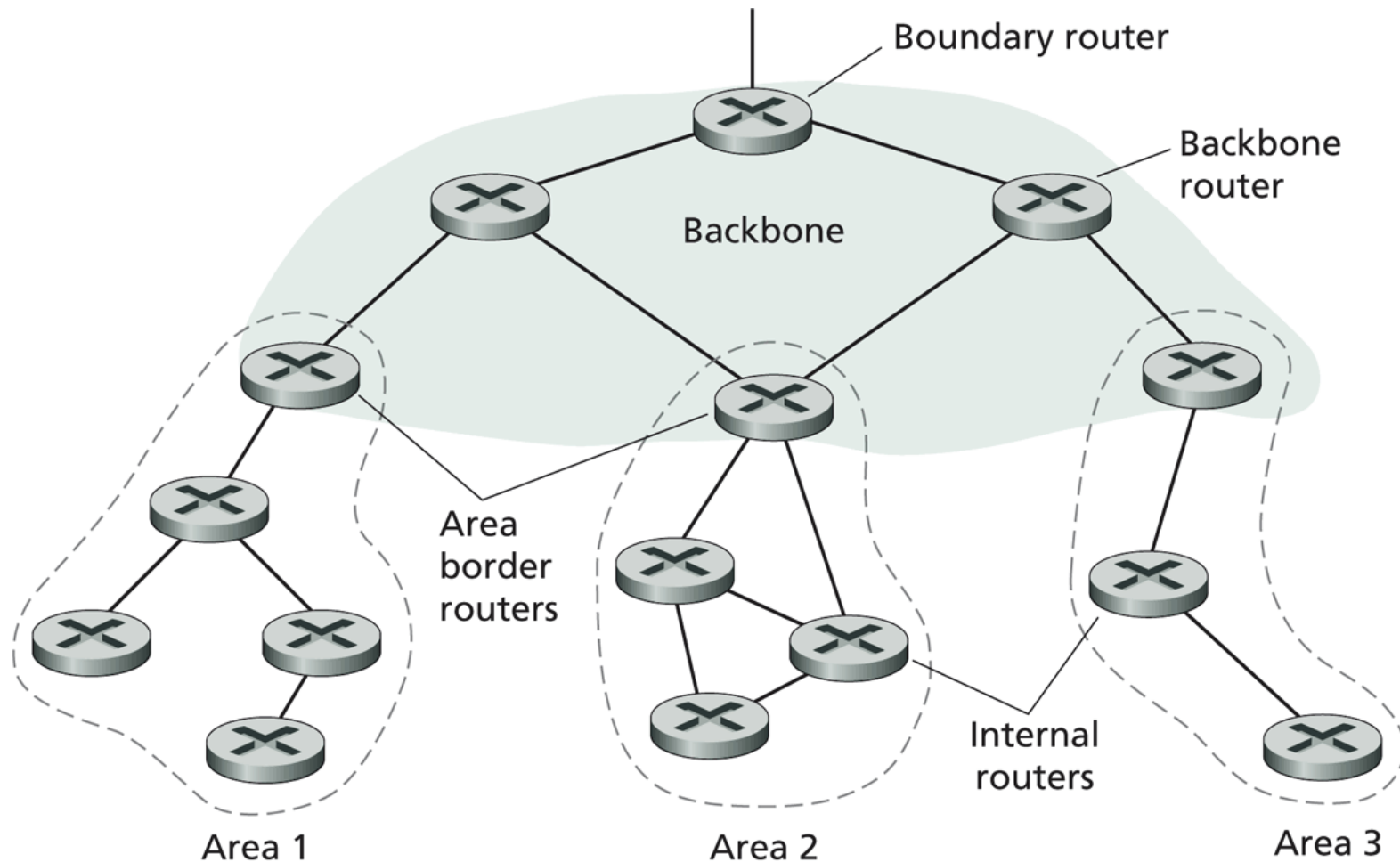


Figure 4.37 ♦ Hierarchically structured OSPF AS with four areas

Enrutamiento Jerárquico

Algoritmos Inter-AS

Un router en AS1 recibe un datagrama destinado hacia afuera de AS1.

- En enrutador debe reenviar el paquete hacia el router de puerta de enlace, ¿pero a cuál?

AS1 debe:

- Aprender qué destinos son alcanzables desde AS2 y cuáles desde AS3
- Propagar esta información a todos los enrutadores en AS1

Enrutamiento Intra-AS

También conocidos como Interior Gateway Protocols (IGP)

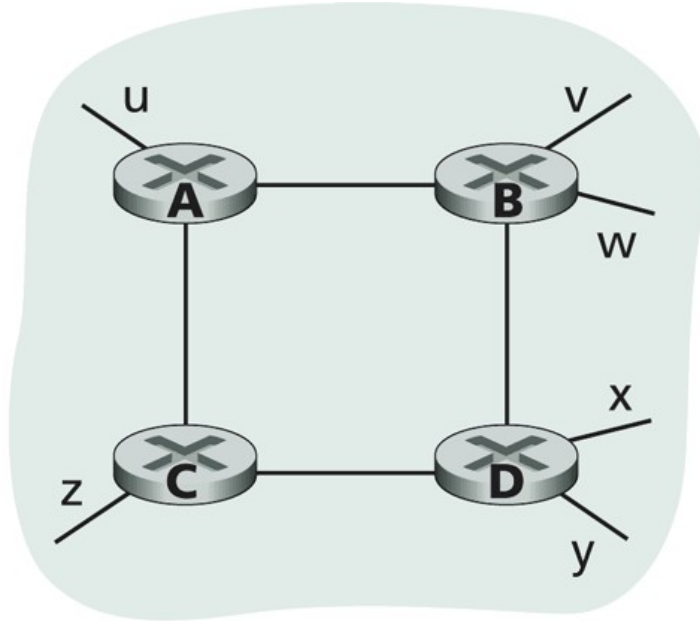
Los más conocidos son:

- RIP: Routing Information Protocol
- OSPF: Open Shortest Path First

RIP (Routing Information Protocol)

Algoritmo de vector de distancias

- Métrica: #hops (máximo = 15 hops), cada enlace tiene costo 1
- Cada mensaje tiene hasta 25 subredes destino (en sentido de direccionamiento IP)



Destination	Hops
u	1
v	2
w	2
x	3
y	3
z	2

OSPF (Open Shortest Path First)

Open: Abierto públicamente

- Usa algoritmo Link State
- Mapa de la red en cada nodo
- Computa la información de routing con algoritmo de Dijkstra

Los avisos se envían a todo el AS (Realizando un inundado de paquetes)

- Se envían en mensajes OSPF directamente sobre IP
(En lugar de sobre TCP o UDP)

Enrutamiento inter-AS en Internet

BGP (Border Gateway Protocol)

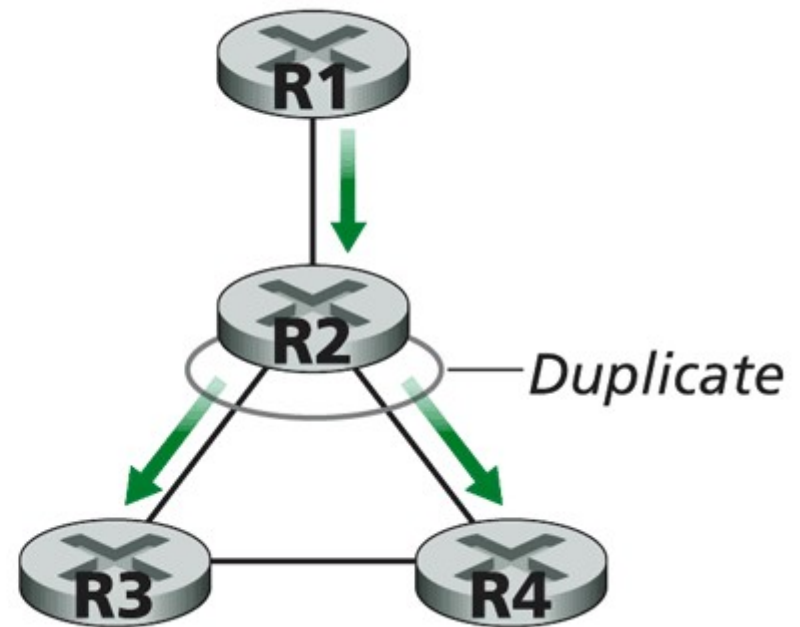
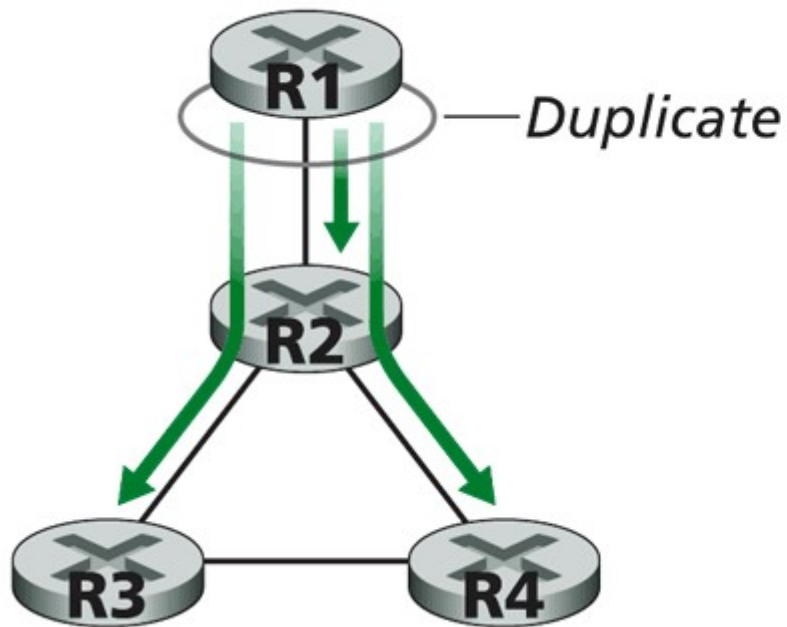
El algoritmo usado en Internet para routing inter-AS

Provee a cada AS:

- Buenas rutas hacia otras redes basado en la topología de los AS's y la política utilizada (por contrato algunos AS no pueden reenviar paquetes hacia otros)
- Permite a una subred avisar de su existencia al resto de Internet.

Enrutamiento por difusión

Entregar paquetes de un nodo origen a todos los demás nodos
Duplicación en el origen es ineficiente:



Duplicación en la red

Inundación:

Cuando un nodo recibe un paquete de difusión, envía una copia a cada vecino

- Problema de ciclo

Inundación controlada:

Un nodo sólo reenvía un paquete si no ha reenviado el mismo paquete antes

- El nodo debe recordar qué paquetes ya ha difundido
- Reverse path forwarding (RPF): sólo reenviar paquetes si llegó desde el camino más corto entre el nodo y el destino.

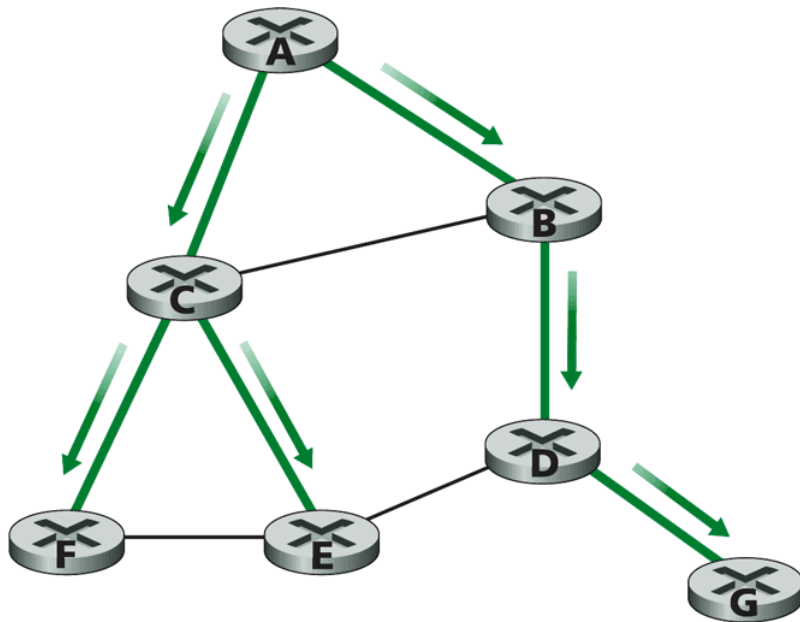
Árbol de recubrimiento

- No se reciben paquetes redundantes.

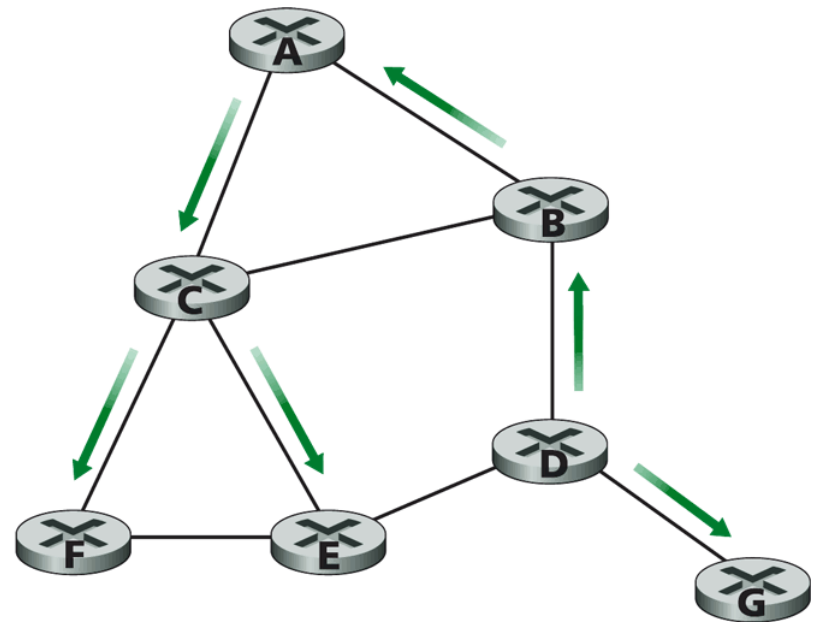
Árbol de recubrimiento

Primero construir un árbol recubridor

Los nodos envían copias sólo por las aristas del árbol



a. Broadcast initiated at A



b. Broadcast initiated at D

Figure 4.42 ♦ Broadcast along a spanning tree