

## 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 4-127

---

---

---

---

---

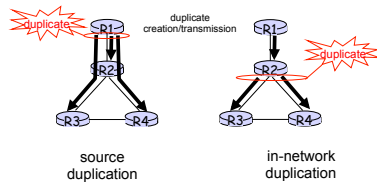
---

---

---

## Broadcast Routing

- entrega de paquetes desde la fuente a todos los nodos
- duplicación en la fuente es ineficiente:



- duplicación en la fuente: como determinar la dirección de los receptores?
  - registro?

Int. Redes de Computadores-Capa de Red 4-128

---

---

---

---

---

---

---

---

## Duplicación en la red

- flooding: cuando un nodo recibe un paquete, envía copias a todos sus vecinos
  - Problemas: ciclos & "tormenta" de broadcasts
- flooding controlado: el nodo solo hace broadcast de un paquete si no lo ha enviado antes
  - el nodo debe llevar la cuenta de los paquetes enviados recientemente
  - o "reverse path forwarding" (RPF): solo envía un paquete si llegó por el camino más corto entre el nodo y la fuente
- spanning tree
  - ningún nodo recibe paquetes redundantes

Int. Redes de Computadores-Capa de Red 4-129

---

---

---

---

---

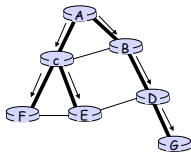
---

---

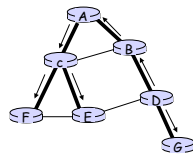
---

## Spanning Tree

- primero hay que contruir el spanning tree
- los nodos envían copias solamente sobre el spanning tree



(a) Broadcast iniciado en A



(b) Broadcast iniciado en D

Int. Redes de Computadores-Capa de Red 4-130

---

---

---

---

---

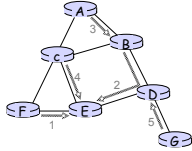
---

---

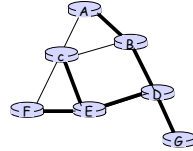
---

## Spanning Tree: creación

- nodo central o raíz
- cada nodo envía un mensaje unicast al nodo central para unirse al árbol
  - el mensaje es reenviado hasta que llega a un nodo que pertenece al spanning tree



(a) Construction paso a paso del spanning tree



(b) Spanning tree construido

Int. Redes de Computadores-Capa de Red 4-131

---

---

---

---

---

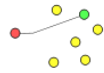
---

---

---

## Multicast

UNICAST: UN EMISOR, UN RECEPTOR.



BROADCAST: UN EMISOR, TODOS LOS RECEPTORES.



ANYCAST: VARIOS EMISORES, UN RECEPTOR (\*).

MULTICAST: VARIOS EMISORES, VARIOS RECEPTORES VOLUNTARIOS.



Int. Redes de Computadores-Capa de Red 4-132

---

---

---

---

---

---

---

---

## Grupos de multicast

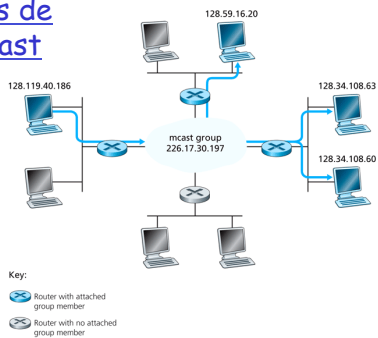


Figure 4.48 ♦ The multicast group: A datagram addressed to the group is delivered to all members of the multicast group.

Int. Redes de Computadores-Capa de Red 4-133

---

---

---

---

---

---

---

---

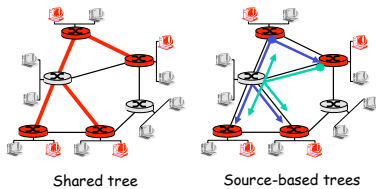
---

---

## Multicast Routing: el problema

□ **Objetivo:** encontrar un árbol (o árboles) que conecta routers que tienen miembros de grupos de mcast

- **árbol:** no se usan todos los caminos
- **source-based:** árboles diferentes desde cada fuente a receptores
- **shared-tree:** todos los miembros del grupo usan el mismo árbol



Int. Redes de Computadores-Capa de Red 4-134

---

---

---

---

---

---

---

---

---

---

## Como se construyen los árboles de multicast?

Posibilidades:

- **source-based tree:** un árbol por fuente
  - shortest path trees
  - reverse path forwarding
- **group-shared tree:** el grupo usa un único árbol
  - minimal spanning (Steiner)
  - center-based trees

...primero vamos a ver los enfoques básicos, y algunos protocolos que los implementan

Int. Redes de Computadores-Capa de Red 4-135

---

---

---

---

---

---

---

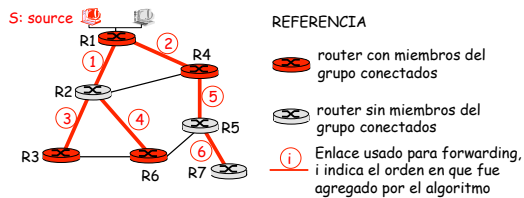
---

---

---

## Shortest Path Tree

- mcast forwarding tree: árbol de caminos más cortos desde la fuente a todos los receptores
  - Algoritmo de Dijkstra



Int. Redes de Computadores-Capa de Red 4-136

---

---

---

---

---

---

---

---

## Reverse Path Forwarding

- Se basa en el conocimiento que tiene el router del “unicast shortest path” desde si mismo a la fuente
- cada router se comporta de forma simple:

*if* (datagrama mcast recibido en enlace entrante en el “shortest path” hacia el centro)  
*then* “flooding” del datagrama en los enlaces de salida  
*else* ignorar datagrama

Int. Redes de Computadores-Capa de Red 4-137

---

---

---

---

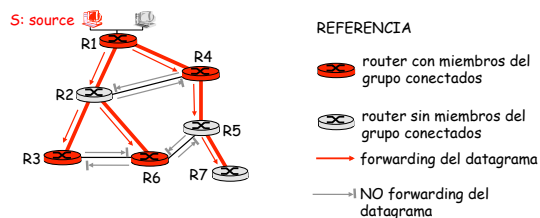
---

---

---

---

## Reverse Path Forwarding: ejemplo



- el resultado es un “reverse SPT” específico para la fuente
  - puede ser una mala opción con enlaces asimétricos

Int. Redes de Computadores-Capa de Red 4-138

---

---

---

---

---

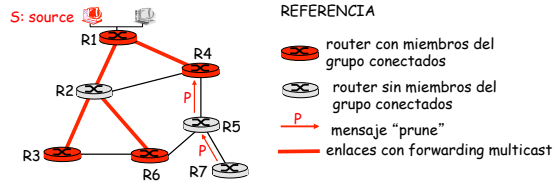
---

---

---

## Reverse Path Forwarding: pruning

- el árbol de forwarding contiene sub-árboles sin miembros del grupo de mcast conectados
  - no es necesario reenviar datagramas en estos sub-árboles
  - Los routers que no tienen miembros conectados envían mensaje "prune" (poda) "hacia atrás"



Int. Redes de Computadores-Capa de Red 4-139

---

---

---

---

---

---

---

---

## Shared-Tree: Steiner Tree

- **Steiner Tree**: árbol de costo mínimo que conecta todos los routers con miembros del grupo conectados
- problema NP-completo
  - existen buenas heurísticas para atacarlo
- no se usa en la práctica:
  - complejidad computacional
  - se necesita información de toda la red
  - monolítico: debe recalcularse cada vez que un router hace un "join"/"leave"

Int. Redes de Computadores-Capa de Red 4-140

---

---

---

---

---

---

---

---

## Center-based trees

- árbol de entrega único compartido
- un router identificado como "**centro**" del árbol
- proceso de unión (join):
  - Los routers de borde envían mensajes unicast *join* destinados al router central
  - el mensaje *join* es procesado por los routers intermedios y reenviados hacia el centro
  - el mensaje *join* se une a una rama existente, o llega al centro
  - el camino recorrido por el mensaje *join* se convierte en una nueva rama

Int. Redes de Computadores-Capa de Red 4-141

---

---

---

---

---

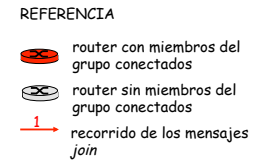
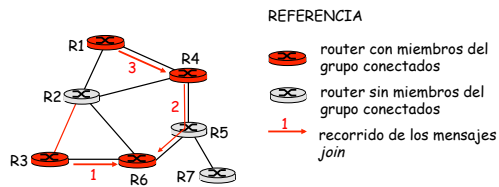
---

---

---

## Center-based trees: un ejemplo

Supongamos que R6 se elige como centro:



Int. Redes de Computadores-Capa de Red 4-142

---

---

---

---

---

---

---

---

## Internet Group Management Protocol: IGMP

- hemos visto como se arman los árboles de distribución...
- ...pero todavía no sabemos como hace un host para unirse a un grupo de multicast
- **IGMP**: protocolo usado por los routers locales (los default gateways) y los hosts

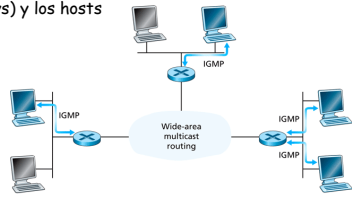


Figure 4.49 ♦ The two components of network-layer multicast in the Internet: IGMP and multicast routing protocols

Int. Redes de Computadores-Capa de Red 4-143

---

---

---

---

---

---

---

---

## IGMP

- mensajes:
  - query: desde el router a los hosts
  - membership report: desde los hosts a los routers
  - leave: desde los hosts a los routers
- el router puede hacer queries genéricos ("todos los grupos") o específicos
- los hosts pueden enviar mensajes de asociación sin esperar ser interrogados (no solicitados)
- soft state: el router mantiene una tabla de los hosts que pertenecen a cada grupo enviando queries periódicos y recibiendo las respuestas
- **IGMP**: protocolo de control análogo a ICMP para unicast

Int. Redes de Computadores-Capa de Red 4-144

---

---

---

---

---

---

---

---

## Internet Multicasting Routing: DVMRP

- **DVMRP**: distance vector multicast routing protocol, RFC1075
- **flood & prune**: reverse path forwarding, source-based tree
  - árbol RPF basado en las tablas de routing de DVMRP, construida por intercambio de mensajes nativos (no se basa en unicast)
  - el datagrama inicial se envía por flooding al grupo de mcast usando RPF
  - los routers que no participan del grupo envían mensajes *prune* hacia "arriba"

Int. Redes de Computadores-Capa de Red 4-145

---

---

---

---

---

---

---

---

## DVMRP

- **soft state**: los routers DVMRP "se olvidan" periódicamente (1 min.) que las ramas que están "pruned":
  - se vuelven a enviar flujos mcast por esas ramas
  - los routers "downstream" (hacia abajo) tienen que volver a mandar prune o seguirán recibiendo el flujo
- los routers pueden reconectarse
  - siguiendo los *join* de IGMP en las hojas (redes locales)
- "de la vida real"
  - usualmente implementado en routers comerciales
  - usado en Mbone (también MOSPF)

Int. Redes de Computadores-Capa de Red 4-146

---

---

---

---

---

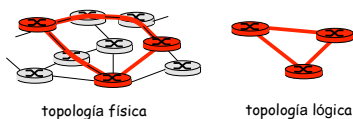
---

---

---

## Tunneling

**P:** cómo conectar "islas" de routers multicast en un "mar" unicast?



- datagramas mcast encapsulados en datagramas "normales"
- datagramas enviados a través de "túneles" usando unicast al receptor mcast
- receptor mcast de-encapsula los datagramas

Int. Redes de Computadores-Capa de Red 4-147

---

---

---

---

---

---

---

---

## PIM: Protocol Independent Multicast

- ❑ no depende en ningún algoritmo de routing unicast específico (funciona con todos)
- ❑ dos escenarios de distribución multicast :

### Dense:

- ❑ miembros del grupo “densamente empaquetados”, o cercanos
- ❑ mucho ancho de banda disponible

### Sparse:

- ❑ # de routers con miembros del grupo conectado es pequeño con respecto al # total de routers
- ❑ miembros “muy dispersos”
- ❑ menos ancho de banda disponible

Int. Redes de Computadores-Capa de Red 4-148

---

---

---

---

---

---

---

---

## Consecuencias de la dicotomía

### Sparse-Dense:

#### Dense

- ❑ membresía de los routers al grupo *se asume* hasta que se ejecuta un *prune* explícito
- ❑ la construcción del árbol mcast es *data-driven* (p. e.j., RPF)
- ❑ *derroche* de recursos (ancho de banda y membresía)

#### Sparse:

- ❑ membresía explícita via *join*
- ❑ la construcción del árbol mcast es *receiver-driven* (p.e.j., center-based)
- ❑ uso *conservador* de los recursos

Int. Redes de Computadores-Capa de Red 4-149

---

---

---

---

---

---

---

---

## PIM- Dense Mode

**flood-and-prune RPF**, similar a DVMRP, pero...

- ❑ el protocolo unicast de base provee la información RPF para el datagrama entrante
- ❑ el mecanismo de flooding “hacia abajo” (downstream) es más sencillo (y menos eficiente) que en DVMRP, y por lo tanto no depende tanto del protocolo de base
- ❑ tiene un mecanismo para detectar los nodos “hoja” (leaf-node router)

Int. Redes de Computadores-Capa de Red 4-150

---

---

---

---

---

---

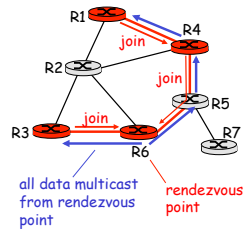
---

---



## PIM - Sparse Mode

- ❑ center-based
- ❑ router envía mensaje *join* a un rendezvous point (RP)
  - routers intermedios actualizan su estado y reenvían el *join*
- ❑ luego de asociarse via RP, un router puede cambiarse a un árbol source-specific
  - mejora la performance: menor concentración, caminos más cortos



Int. Redes de Computadores-Capa de Red 4-151

---

---

---

---

---

---

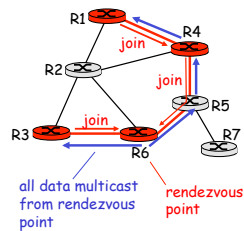
---

---

## PIM - Sparse Mode

### fuelle(s):

- ❑ datos unicast al RP, quien distribuye en el árbol del cual es raíz
- ❑ el RP puede extender los árboles mcast "hacia arriba" hasta la fuente
- ❑ el RP puede enviar mensaje *stop* si no hay receptores
  - "nadie está escuchando!"



Int. Redes de Computadores-Capa de Red 4-152

---

---

---

---

---

---

---

---

## 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 4-153

---

---

---

---

---

---

---

---