

# Routing: Routing Multicast

Area de Ingeniería Telemática  
<http://www.tlm.unavarra.es>

Grado en Ingeniería en Tecnologías de  
Telecomunicación, 3º

# Temas de teoría

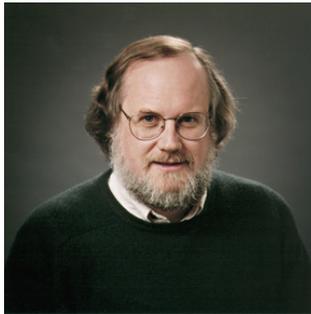
1. Introducción
2. QoS
3. Encaminamiento dinámico en redes IP
4. Tecnologías móviles
5. Otros temas

# Objetivo

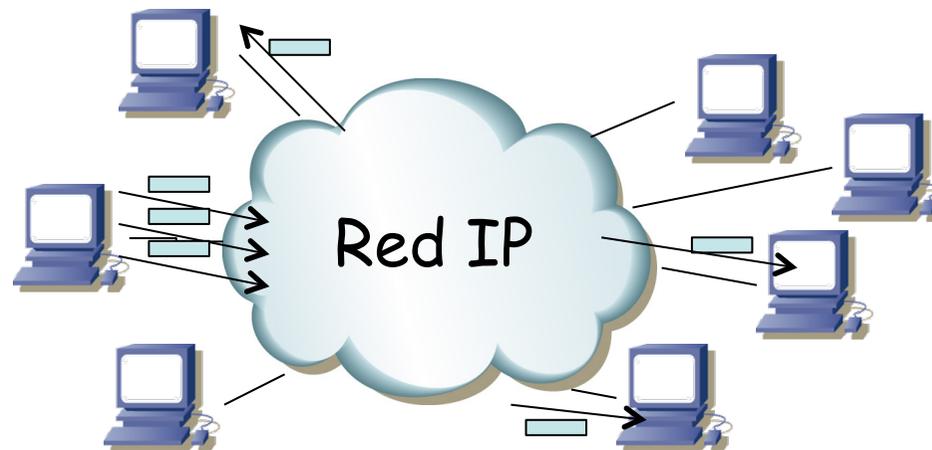
- Conocer el funcionamiento de los protocolos de encaminamiento multicast más comunes

# Multicast

- Permite mandar paquetes a un subconjunto de hosts de la red
- En unicast habría que enviar un paquete a cada destino y habría que conocer las direcciones de todos ellos (...)

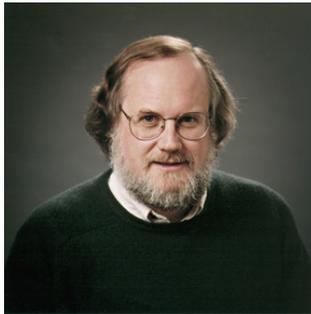


Dr. Steve Deering

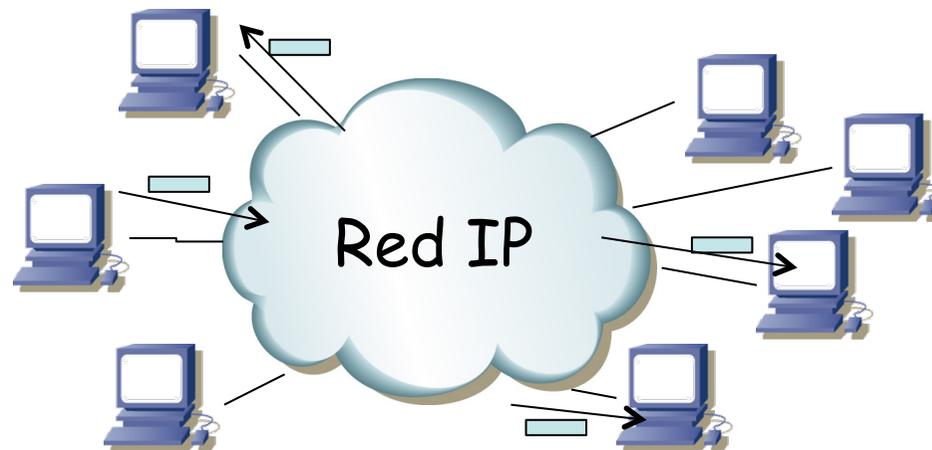


# Multicast

- Permite mandar paquetes a un subconjunto de hosts de la red
- En unicast habría que enviar un paquete a cada destino y habría que conocer las direcciones de todos ellos
- En multicast origen envía un solo paquete y la red se encarga de replicarlo y hacerlo llegar a todos los interesados (...)



Dr. Steve Deering



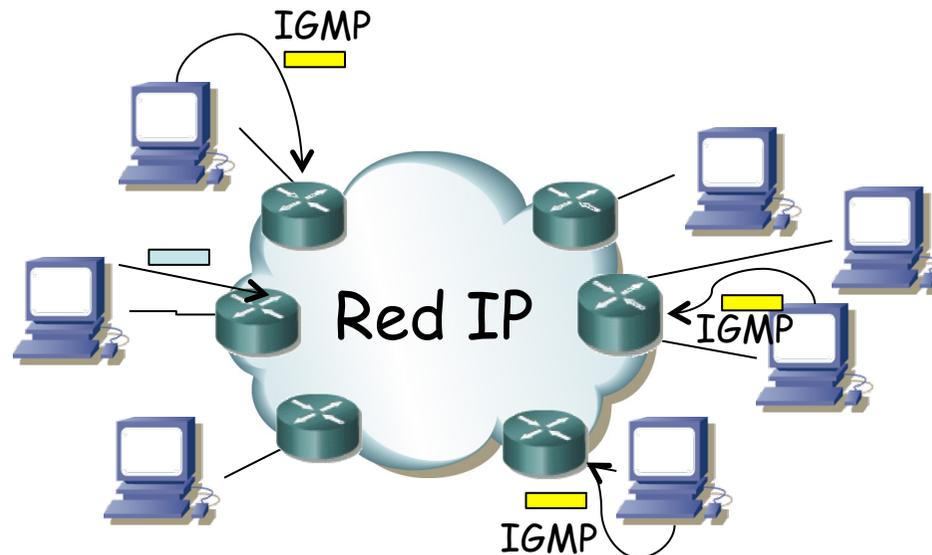
# Multicast

- Permite mandar paquetes a un subconjunto de hosts de la red
- En unicast habría que enviar un paquete a cada destino y habría que conocer las direcciones de todos ellos
- En multicast origen envía un solo paquete y la red se encarga de replicarlo y hacerlo llegar a todos los interesados
- ¿Qué tiene que hacer la aplicación del host origen?
  - Nada especial (solo lo que requiera el API para enviar a destinos multicast)
- ¿Sabe el origen quiénes van a recibirlo? No
- ¿Cómo sabe la red que el paquete puede tener que ir a varios hosts?
  - Bloque 224.0.0.0/4
  - Se habla de “grupo multicast” pues la dirección hace referencia a un grupo de hosts



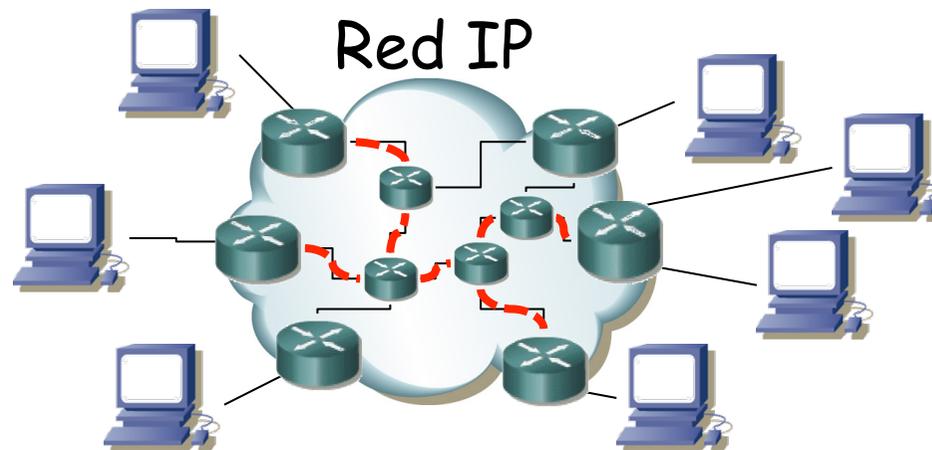
# Multicast

- ¿Cómo sabe la red a qué hosts debe hacerlo llegar?
  - La aplicación receptora indica al S.O. que quiere recibir los paquetes de un grupo multicast
  - El host indica a la red que quiere recibir los paquetes que van a ese grupo multicast mediante IGMP (se “une al grupo”)
  - El que envía el tráfico multicast no necesita unirse al grupo
- ¿Cómo sabe la red cómo hacer llegar esos paquetes?
  - (...)



# Multicast

- ¿Cómo sabe la red a qué hosts debe hacerlo llegar?
  - La aplicación receptora indica al S.O. que quiere recibir los paquetes de un grupo multicast
  - El host indica a la red que quiere recibir los paquetes que van a ese grupo multicast mediante IGMP (se “une al grupo”)
  - El que envía el tráfico multicast no necesita unirse al grupo
- ¿Cómo sabe la red cómo hacer llegar esos paquetes?
  - Protocolo de encaminamiento multicast
  - Calcula árboles para comunicar orígenes con destinos
- Se controla el “alcance” de los paquetes con el TTL



# Grupos Multicast

- Los grupos pueden ser permanente o transitorios
- Lo que es *permanente* es la dirección del grupo; pertenencia dinámica
- No hay límites de localización o número de miembros
- Un interfaz puede pertenecer a varios grupos
- No hace falta ser miembro para enviar datagramas al grupo
- Routers → Mrouter (Multicast Router)
- Direcciones controladas por IANA (RFC 5771, BCP 51)  
<http://www.iana.org/assignments/multicast-addresses/multicast-addresses.xml>
- **Local Network Control Block (224.0.0.0/24)**
  - Para tráfico de control que no sale del enlace
- **Internetwork Control Block (224.0.1.0/24)**
  - Tráfico de control que puede ser reenviado por Internet
- **Ad-hoc Blocks (I, II y III)**
  - Pueden llegar a Internet
  - Aplicaciones que necesitan bloques pequeños de direcciones
- **SDP/SAP Block (224.2.0.0/16)**
- **Source-Specific Multicast Block (232.0.0.0/8): RFC 4607**

# Grupos Multicast

- **GLOP Block (233.0.0.0/8)**
  - 233.[ASN16bits].0/24
  - ASs que tengan un ASN de 32 bits pueden pedir un bloque dentro del Ad-hock Block III
- **Administratively Scoped Block (239.0.0.0/8)**
  - RFC 2365
  - No sale del dominio y su asignación es gestionada por el mismo (pueden duplicarse de un dominio a otro)
  - IPv4 Local Scope (239.255.0.0/16)
  - IPv4 Organizacion Local Scope (239.192.0.0/14)

# Ejemplos

## Local Network Control Block

- 224.0.0.1 *All Systems on this Subnet*
- 224.0.0.2 *All Routers on this Subnet*
- 224.0.0.4 *All DVMRP Routers*
- 224.0.0.5 *OSPF/IGMP All Routers*
- 224.0.0.6 *OSPF/IGMP Designated Routers*
- 224.0.0.9 *RIP2 Routers*
- 224.0.0.13 *All PIM routers*
- 224.0.0.18 *VRRP*
- 224.0.0.22 *IGMP*
- 224.0.0.102 *HSRP*

## Internet Control Block

- 224.0.1.1 *NTP Network Time Protocol*

# Aplicaciones

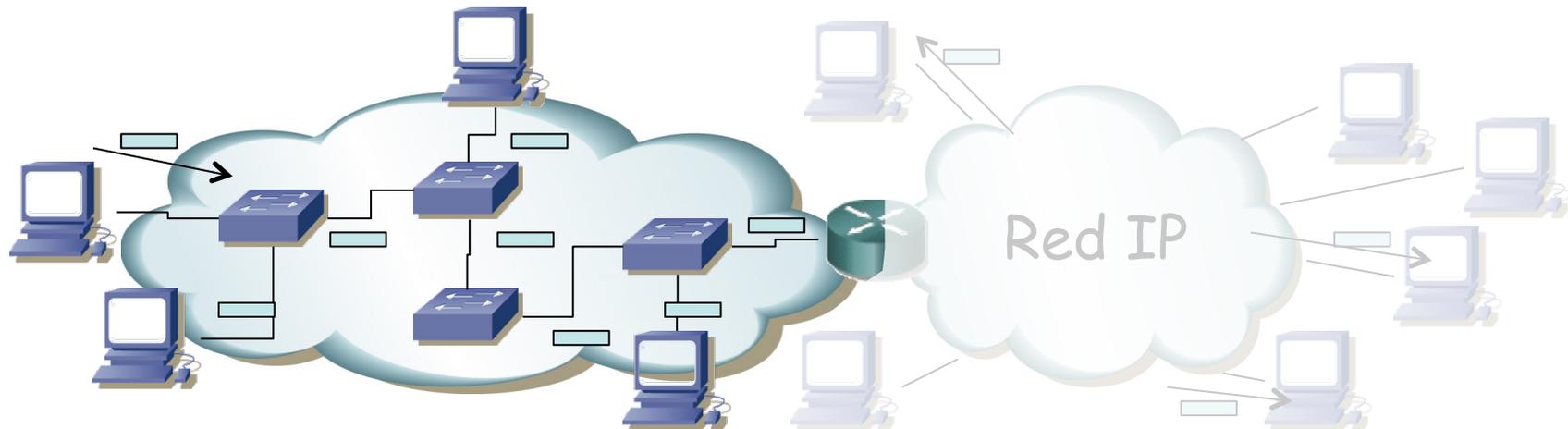
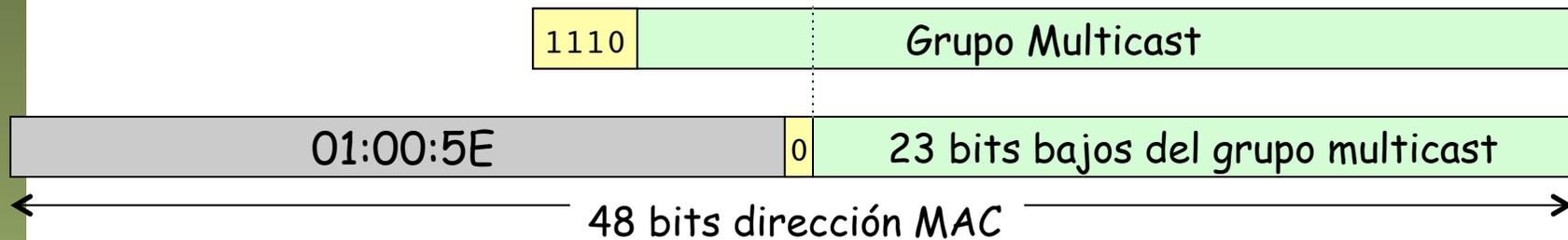
- Monitorización: bolsa, sensores, seguridad, etc
- Anuncios: noticias, hora, horarios, etc
- Distribución de ficheros y cachés
- Sincronización de bases de datos
- Teleconferencia, *distance learning*, difusión de eventos *live* (deportes, conciertos...)
- IPTV
- Edición compartida, colaboración
- Juegos online multijugador
- Chats de grupo
- Descubrimiento de servicios
- Protocolos de encaminamiento

# Problemas

- Adaptarse al receptor
  - En unicast el flujo puede adaptarse a los requisitos de un receptor
  - En multicast es común a todos
  - Ej: bitrate de un vídeo, tiempos de respuesta por retardo en la red, recuperación ante errores, etc
- Introduce gran cantidad de tráfico en la red
  - Un solo host con un flujo puede utilizar gran cantidad de enlaces
  - Si le forzáramos a enviar en unicast a cada destino estaría limitado por su enlace de acceso el agregado que introduce en la red
- UDP
  - No reacciona ante congestión en la red
  - Vale, sí, pero la alternativa es un unicast a cada destino... de flujos también UDP
- Seguridad
  - Más complicado que en unicast la distribución de claves, la autenticación del otro extremo, etc.

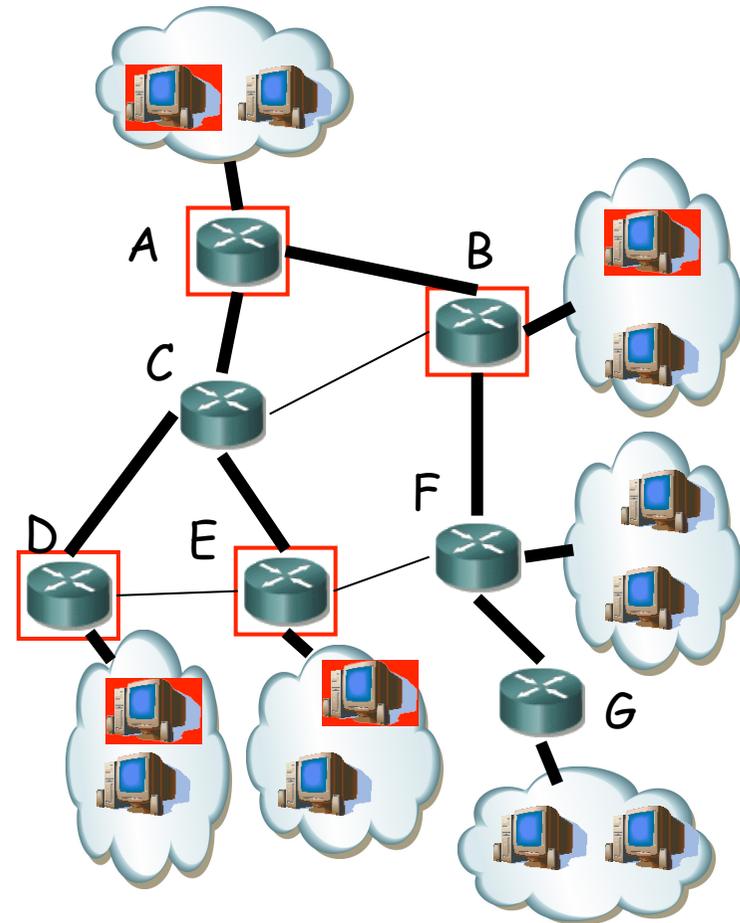
# Multicast en LAN Ethernet

- Se puede enviar un paquete a un conjunto de hosts de la LAN
- Direcciones MAC en Ethernet tienen el primer bit (en transmisión) a 1
- Los conmutadores tendrán que hacer inundación
- Llega a los interesados pero también al resto de hosts de la LAN
- Paquete IP multicast se mapea en trama a MAC multicast (32 a 1)
- Al indicar una aplicación que le interesa un grupo multicast la NIC entregará al nivel IP todas las tramas que correspondan a esa MAC



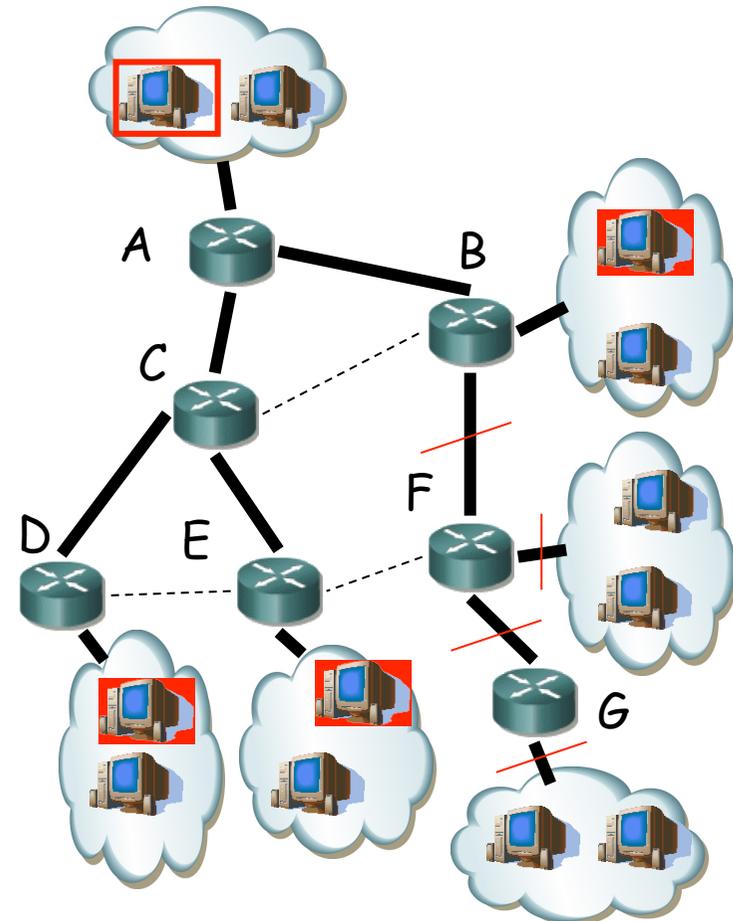
# Shortest Paths

- Encontrar un árbol de enlaces que conecte con el origen a todos los routers que sirven a hosts del grupo
- Árboles de expansión (*spanning trees*)
- Mínimos respecto a una métrica (*shortest path spanning trees*)
- Puede implicar a otros routers



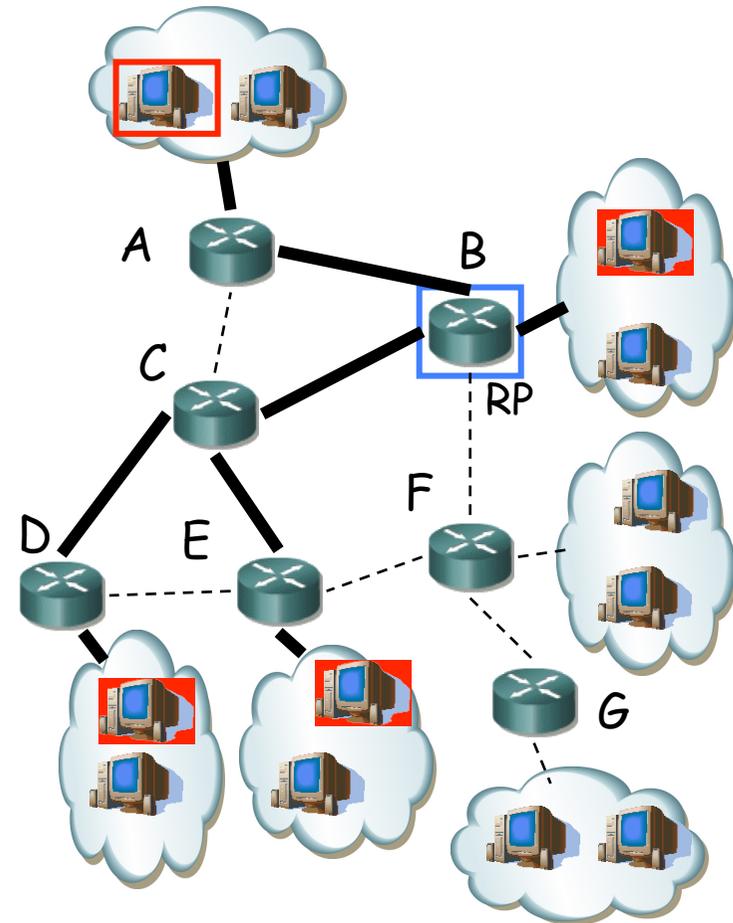
# Source-Based Trees

- Un árbol en el que la raíz es el router de la fuente del flujo
- Un árbol para cada fuente S en cada grupo G, árbol (S,G)
- N° Árboles = Grupos x Fuentes
- Los emplean protocolos en *modo denso*
- Cuando en todas las subredes hay receptor
- **Pruning (poda)**
  - MRouter sin hosts adyacentes ni MRouters downstream en el grupo manda mensaje *prune upstream* (...)
  - Caduca el estar *pruned* y se tiene que renovar
- **Grafting (injerto)**
  - MRouter *pruned* descubre un nuevo cliente interesado en el grupo
  - Manda mensaje para reunirse al árbol
- Protocolos: DVMRP, PIM-DM



# Group-Shared Tree

- Construir un solo árbol para cada grupo (\*,G)
- Incluye a todos los routers adyacentes a hosts en el grupo
- Los emplean protocolos en *modo disperso (sparse)*
- Supone que el flujo no es deseado a menos que se indique explícitamente
- Se suele construir el árbol empleando un **Rendezvous-Point Tree**:
  - Se escoge un nodo central que será la raíz
  - Los demás envían mensajes unicast a él para unirse al árbol
  - Reenviado hasta que el mensaje llega a él o a uno en el árbol
  - Suele tener que refrescarse la rama
- Suelen soportar *pruning*
- Protocolos: CBT, PIM-SM
- Algunos protocolos lo combinan con árboles hasta la raíz

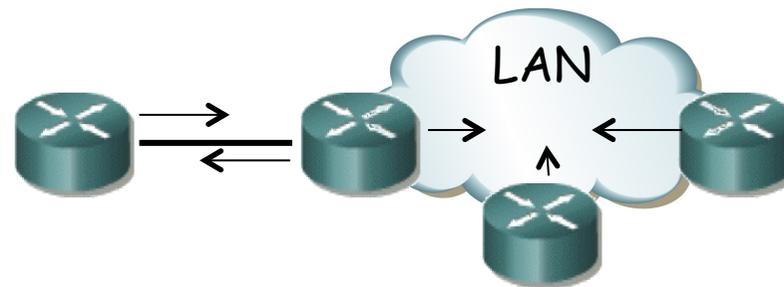


# DVMRP

- Distance Vector Multicast Routing Protocol (RFC 1075)
- El primer protocolo de enrutamiento multicast
- Distance Vector, inspirado en RIP
- Métrica número de saltos (infinito = 32) así que no sirve para la Internet, solo como IGP
- Classless
- Lenta convergencia
- Periódicamente cada mrouter anuncia a sus vecinos las redes origen y el coste hasta ellas
- Emplea un tipo de mensajes IGMP

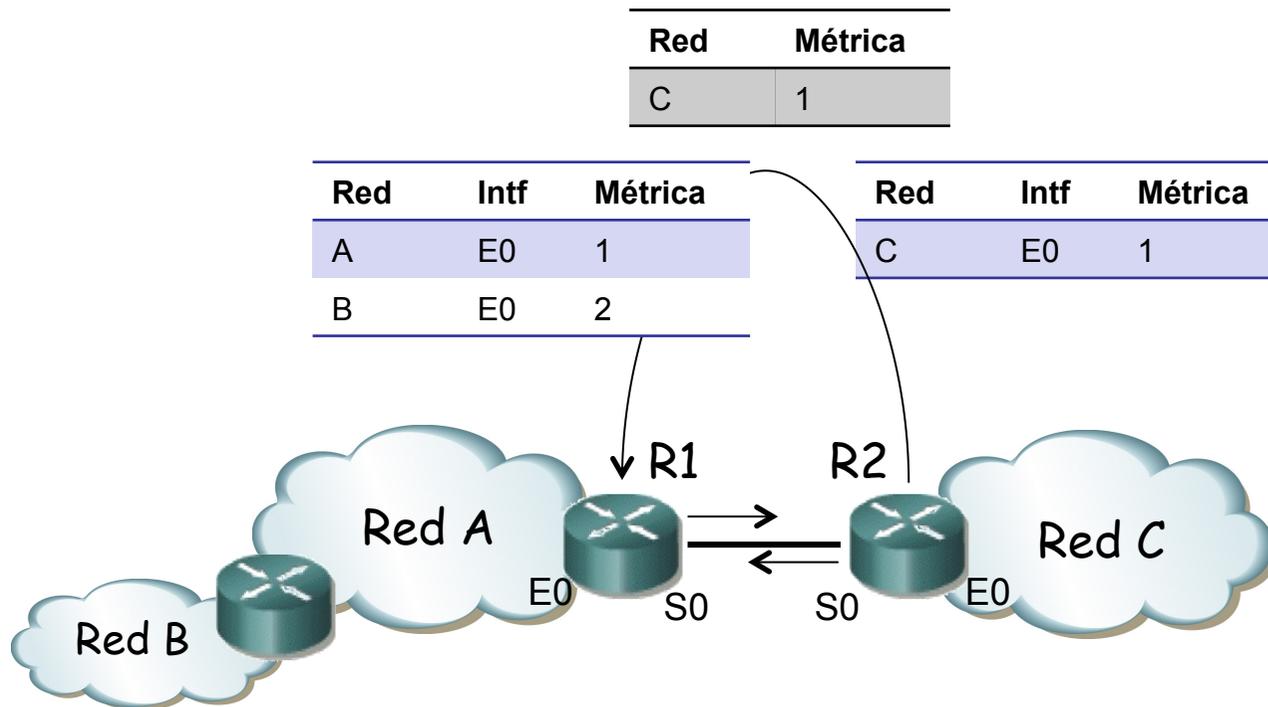
# Neighbor Discovery

- Mantiene adyacencias con routers DVMRP vecinos
- Envía periódicamente mensajes a 224.0.0.4 (*DVMRP Routers*)
- Cada uno envía en el mensaje la lista de routers que ha visto por ese interfaz
- Se verifica la bidireccionalidad del enlace viéndose listado en el anuncio del vecino



# Route Reports

- Cada mrouter tiene una lista de redes con el coste hasta ellas y el interfaz por el cual las alcanza
- Envía periódicamente a los vecinos vector de distancias a esas redes
- (... ..)



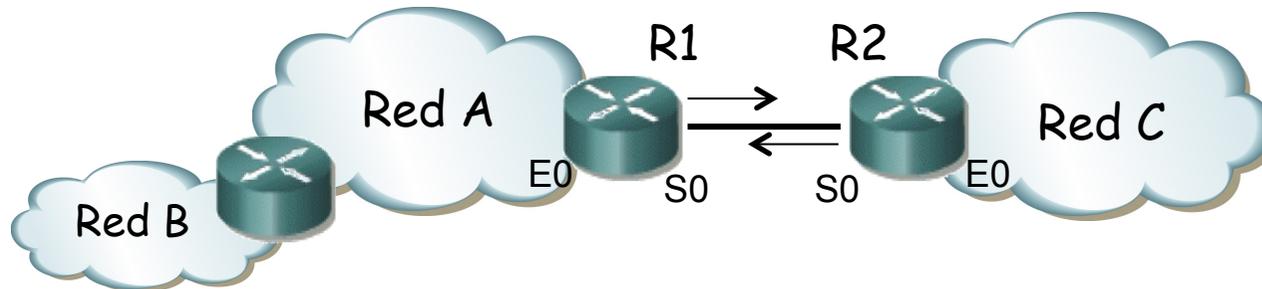
# Route Reports

- Cada mrouter tiene una lista de redes con el coste hasta ellas y el interfaz por el cual las alcanza
- Envía periódicamente a los vecinos vector de distancias a esas redes
- Actualiza al recibir vector de forma similar a RIP (ecuación de BF)
- Destinos a los que se llega por un interfaz los anuncia con la métrica incrementada en 32 (es una especie de *poison reverse*) (. . .)

Red	Métrica
C	1

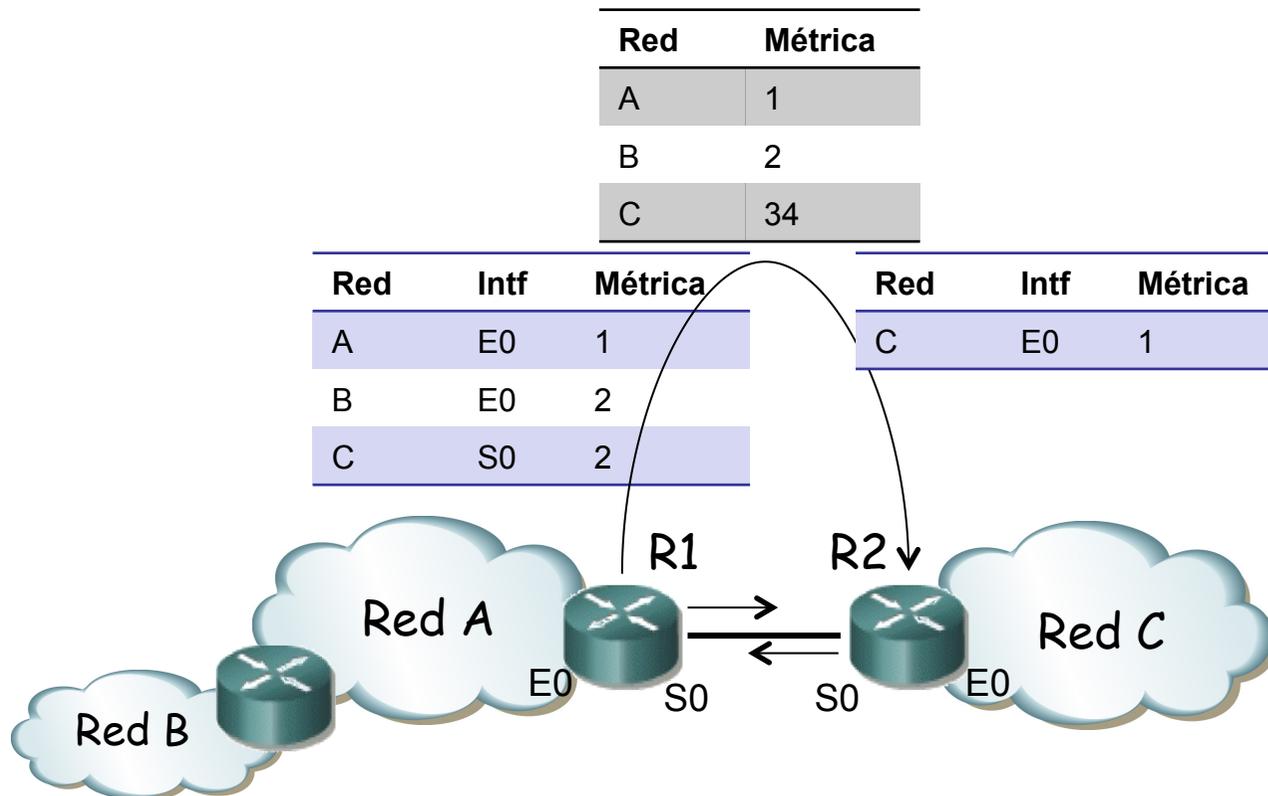
Red	Intf	Métrica
A	E0	1
B	E0	2
C	S0	2

Red	Intf	Métrica
C	E0	1



# Route Reports

- Cada mrouter tiene una lista de redes con el coste hasta ellas y el interfaz por el cual las alcanza
- Envía periódicamente a los vecinos vector de distancias a esas redes
- Actualiza al recibir vector de forma similar a RIP (ecuación de BF)
- Destinos a los que se llega por un interfaz los anuncia con la métrica incrementada en 32 (es una especie de *poison reverse*) (. . .)

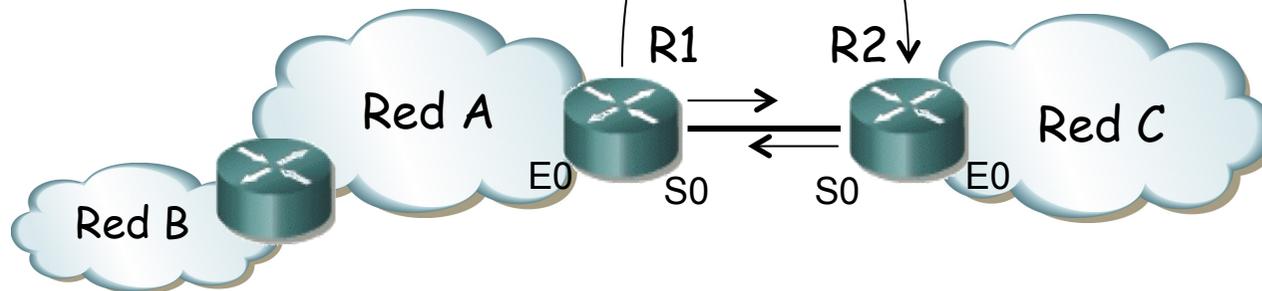


# Route Reports

- Cada mrouter tiene una lista de redes con el coste hasta ellas y el interfaz por el cual las alcanza
- Envía periódicamente a los vecinos vector de distancias a esas redes
- Actualiza al recibir vector de forma similar a RIP (ecuación de BF)
- Destinos a los que se llega por un interfaz los anuncia con la métrica incrementada en 32 (es una especie de *poison reverse*) (. . .)

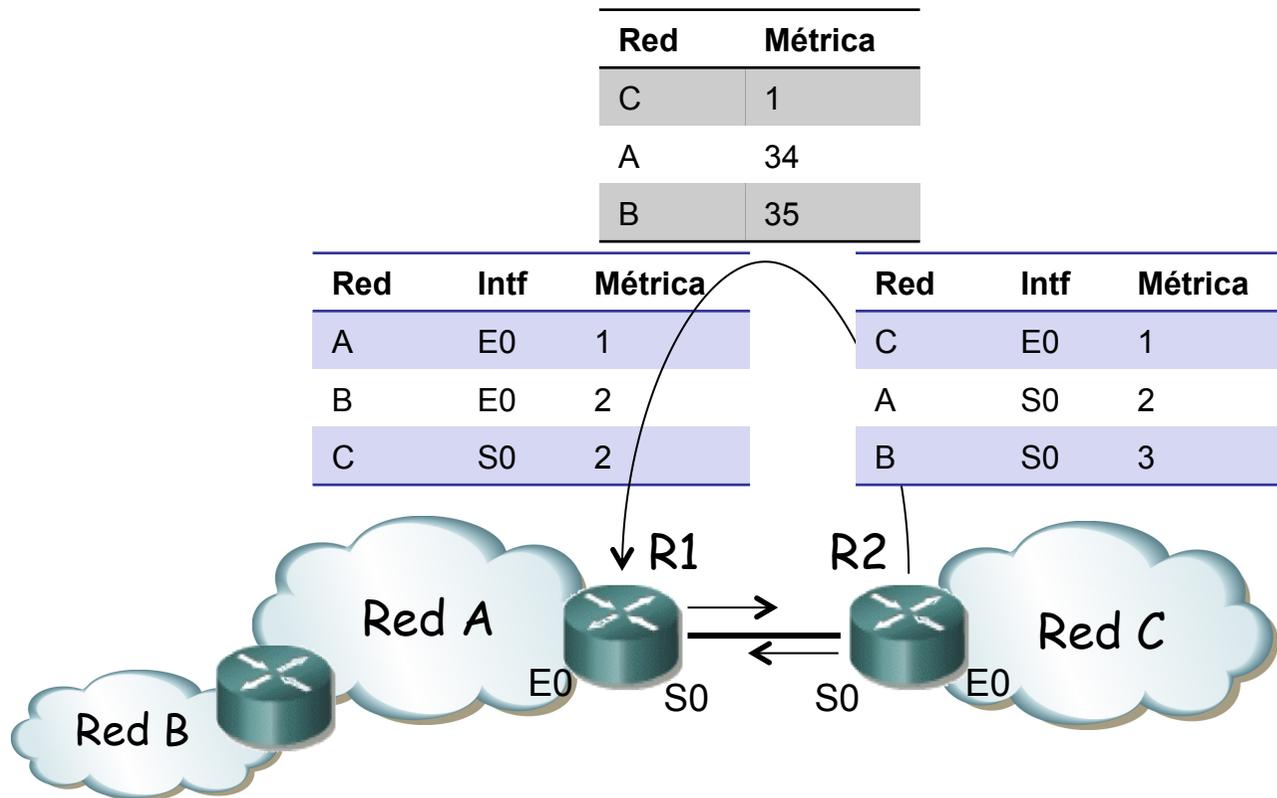
Red	Métrica
A	1
B	2
C	34

Red	Intf	Métrica	Red	Intf	Métrica
A	E0	1	C	E0	1
B	E0	2	A	S0	2
C	S0	2	B	S0	3



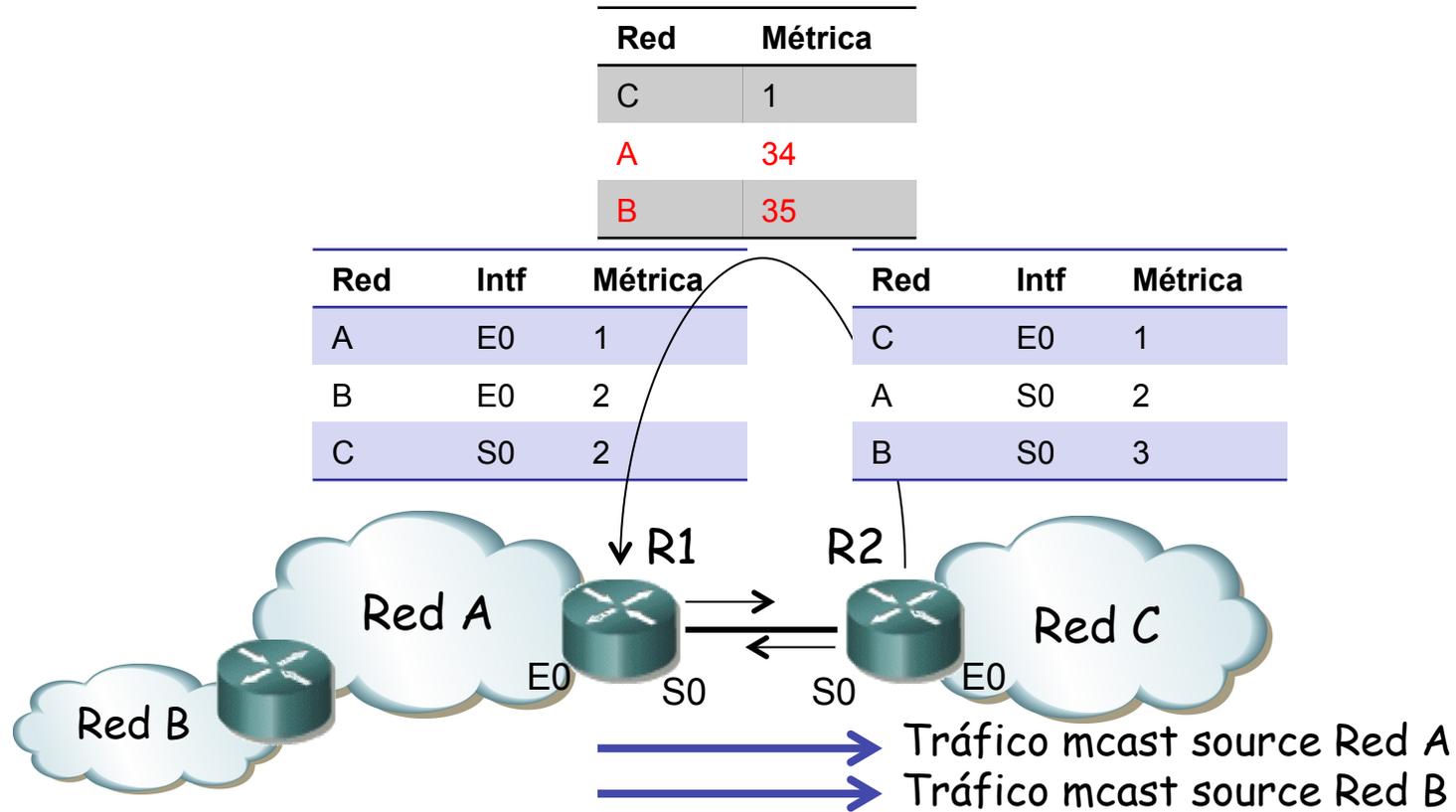
# Route Reports

- Cada mrouter tiene una lista de redes con el coste hasta ellas y el interfaz por el cual las alcanza
- Envía periódicamente a los vecinos vector de distancias a esas redes
- Actualiza al recibir vector de forma similar a RIP (ecuación de BF)
- Destinos a los que se llega por un interfaz los anuncia con la métrica incrementada en 32 (es una especie de *poison reverse*) (. . .)

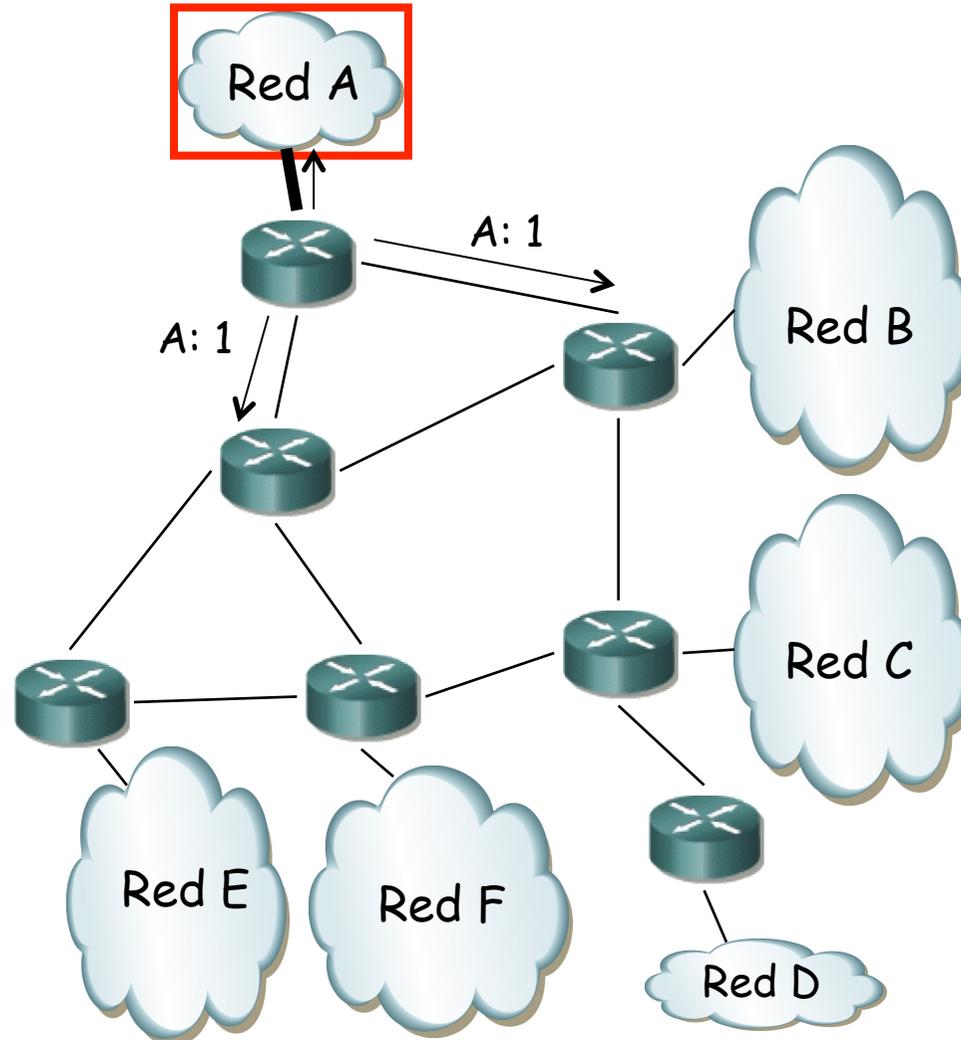


# Truncated Broadcast Trees

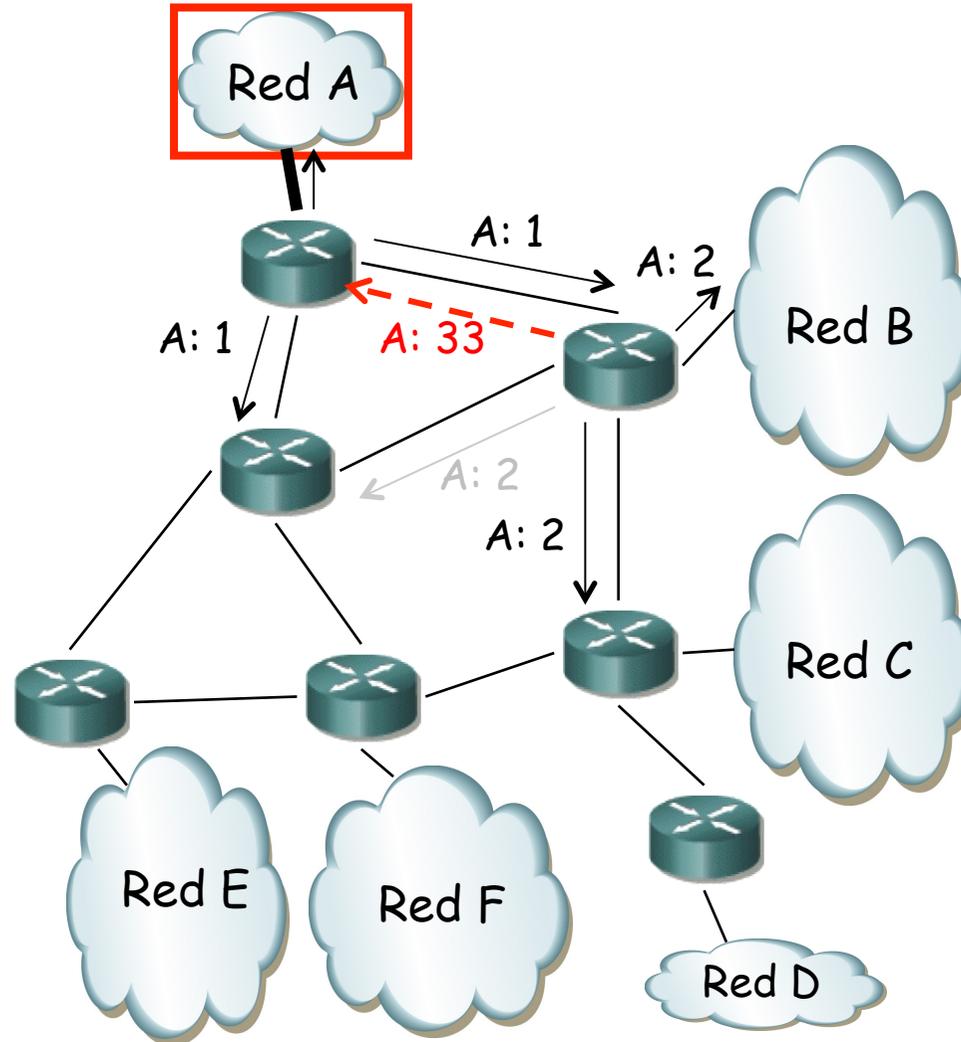
- DVMRP es de tipo denso, calculando *source trees*
- Los calcula en base a los anuncios *poison reversed*
- Un mrouter envía al router *upstream* un anuncio con las redes para las que está *downstream* con la métrica incrementada en 32
- El mrouter *upstream* aprende que debe reenviar a ese mrouter el tráfico que venga de esos orígenes



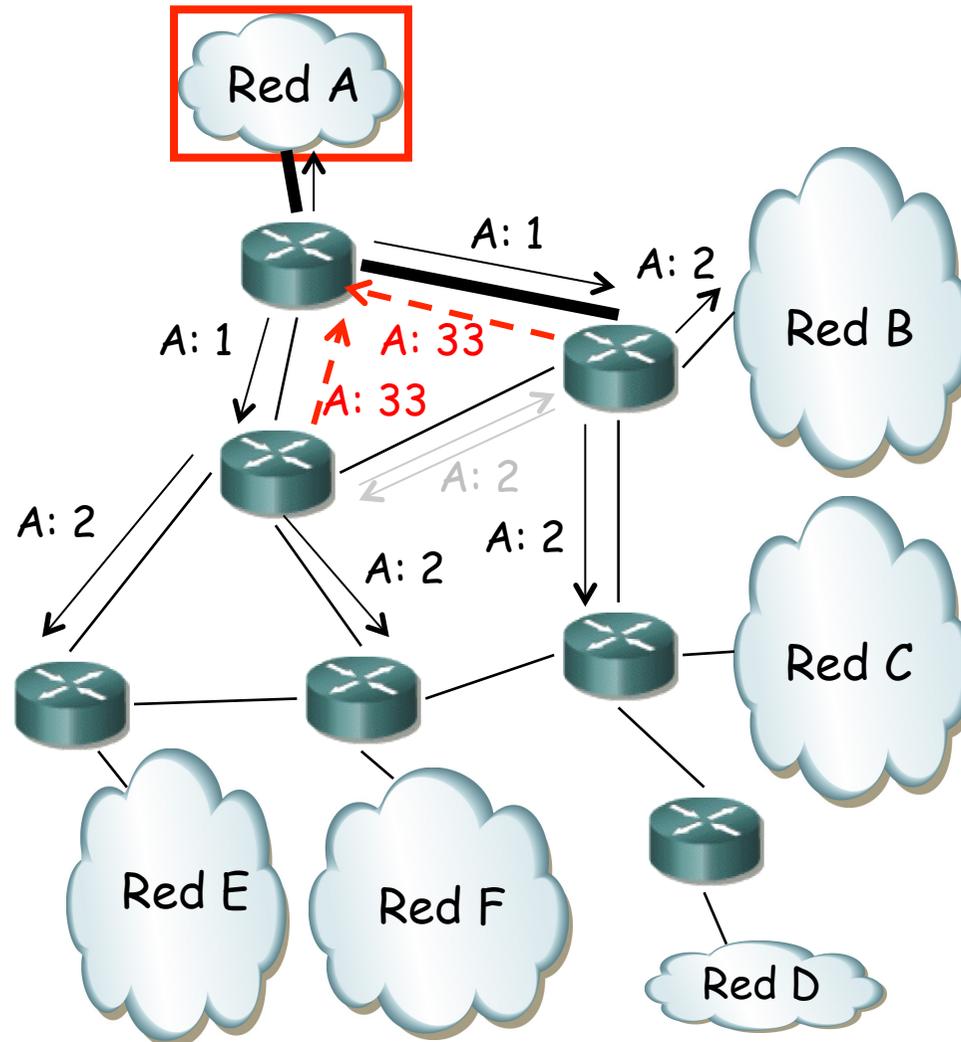
# Ejemplo



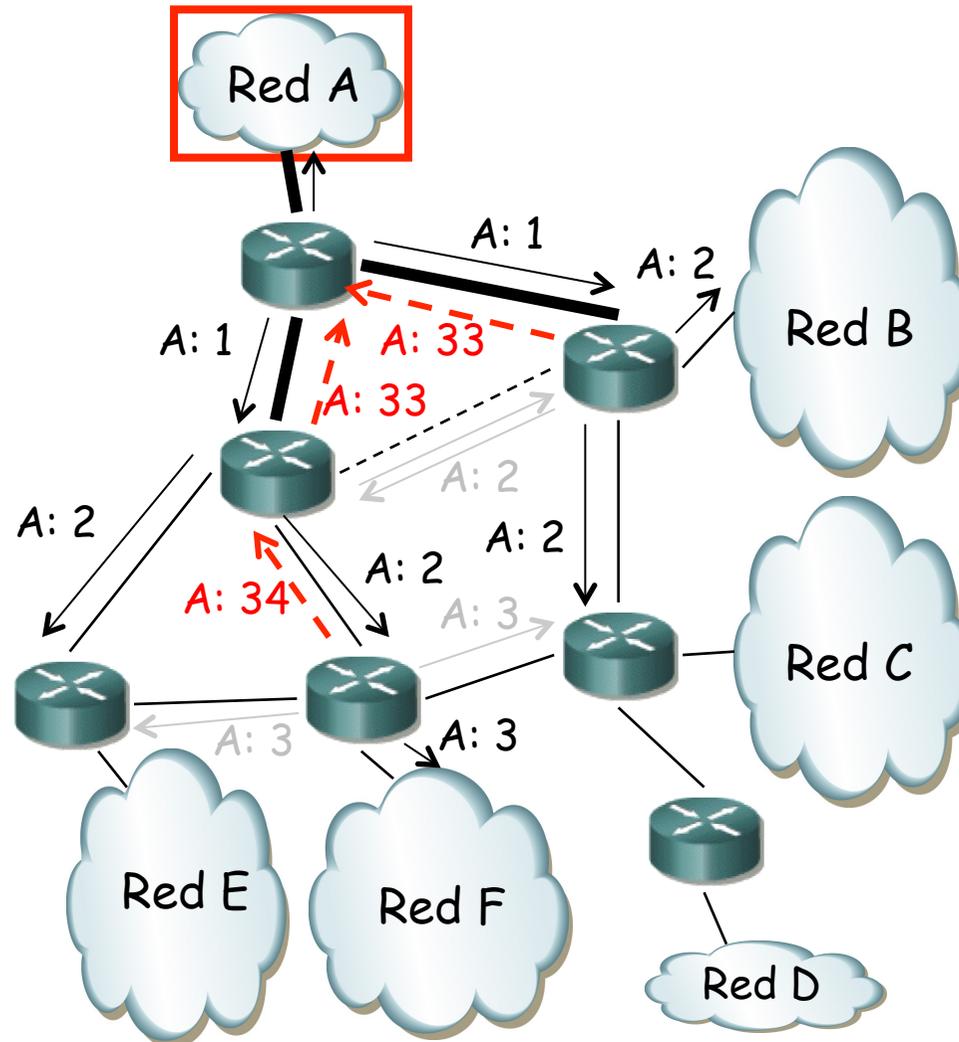
# Ejemplo



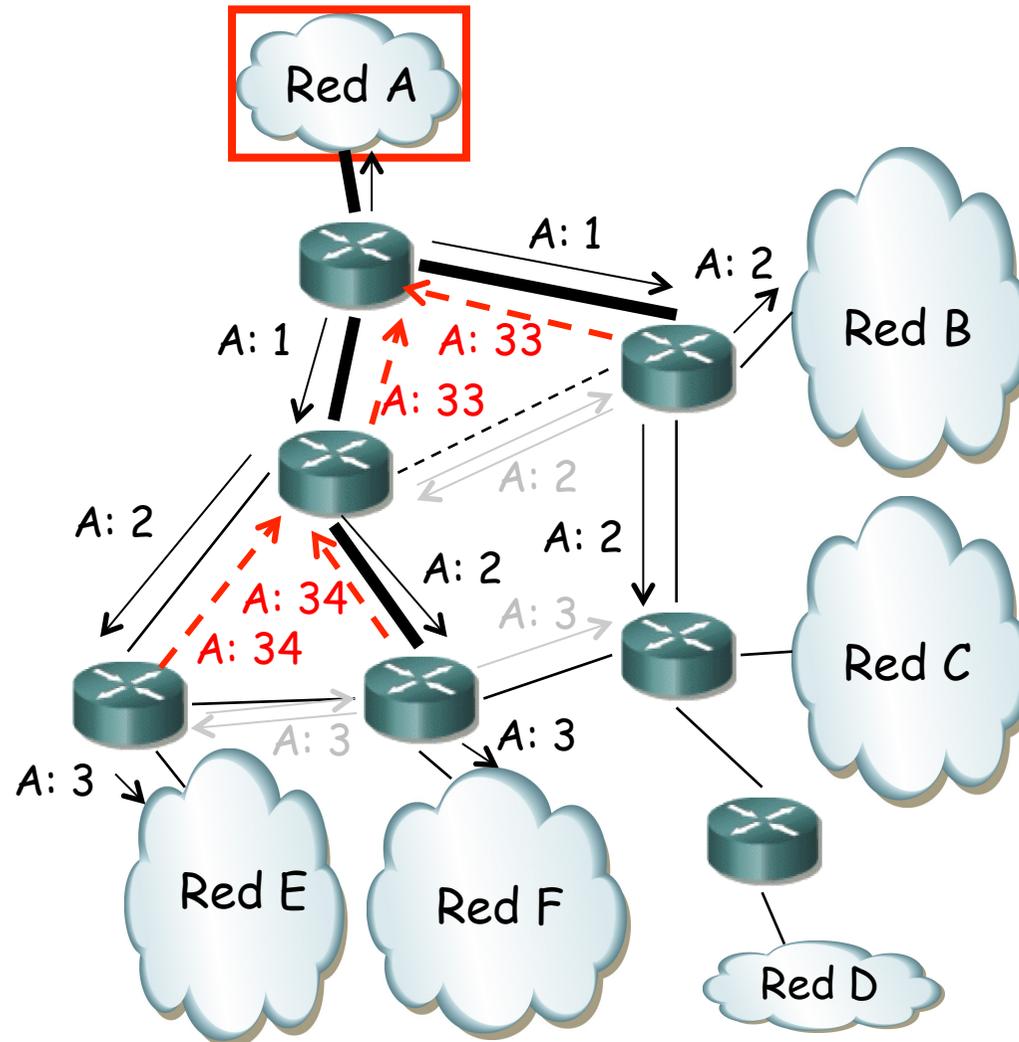
# Ejemplo



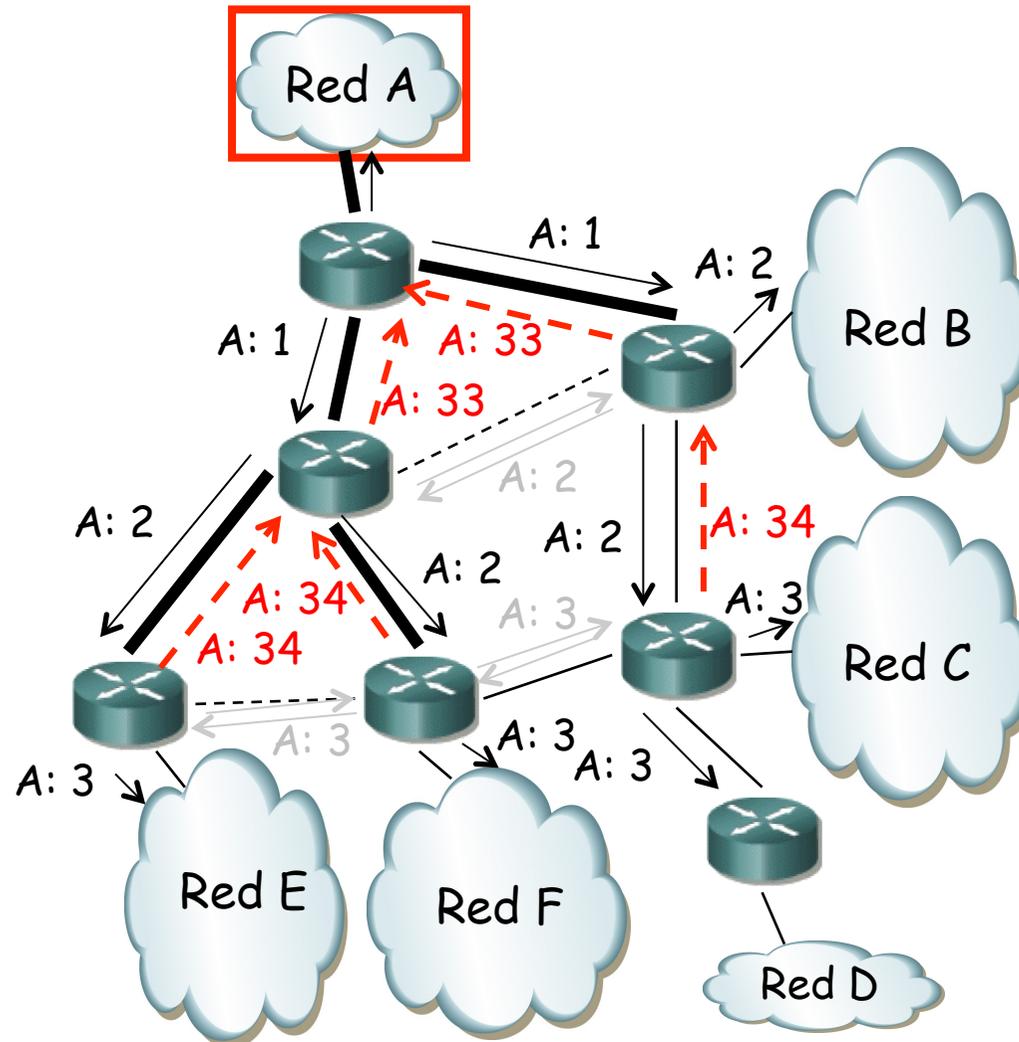
# Ejemplo



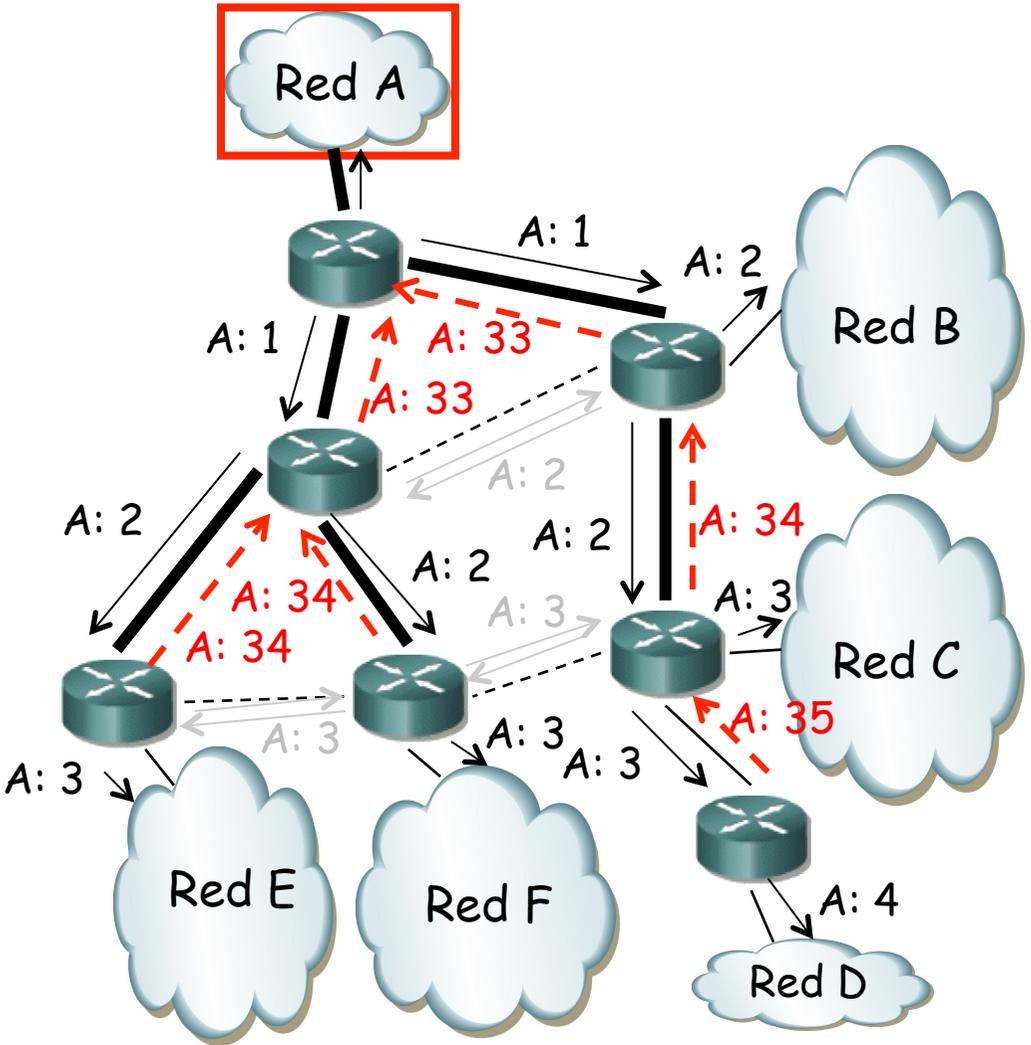
# Ejemplo



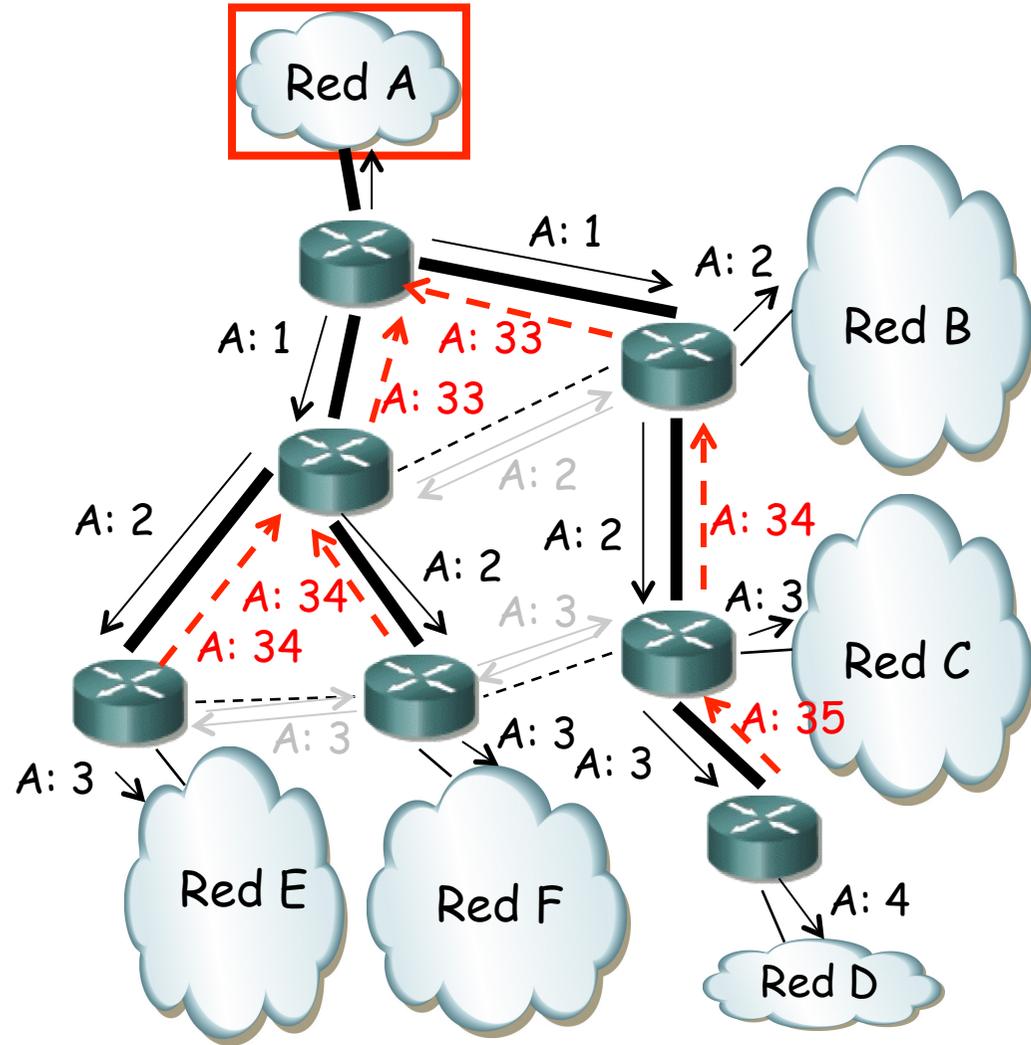
# Ejemplo



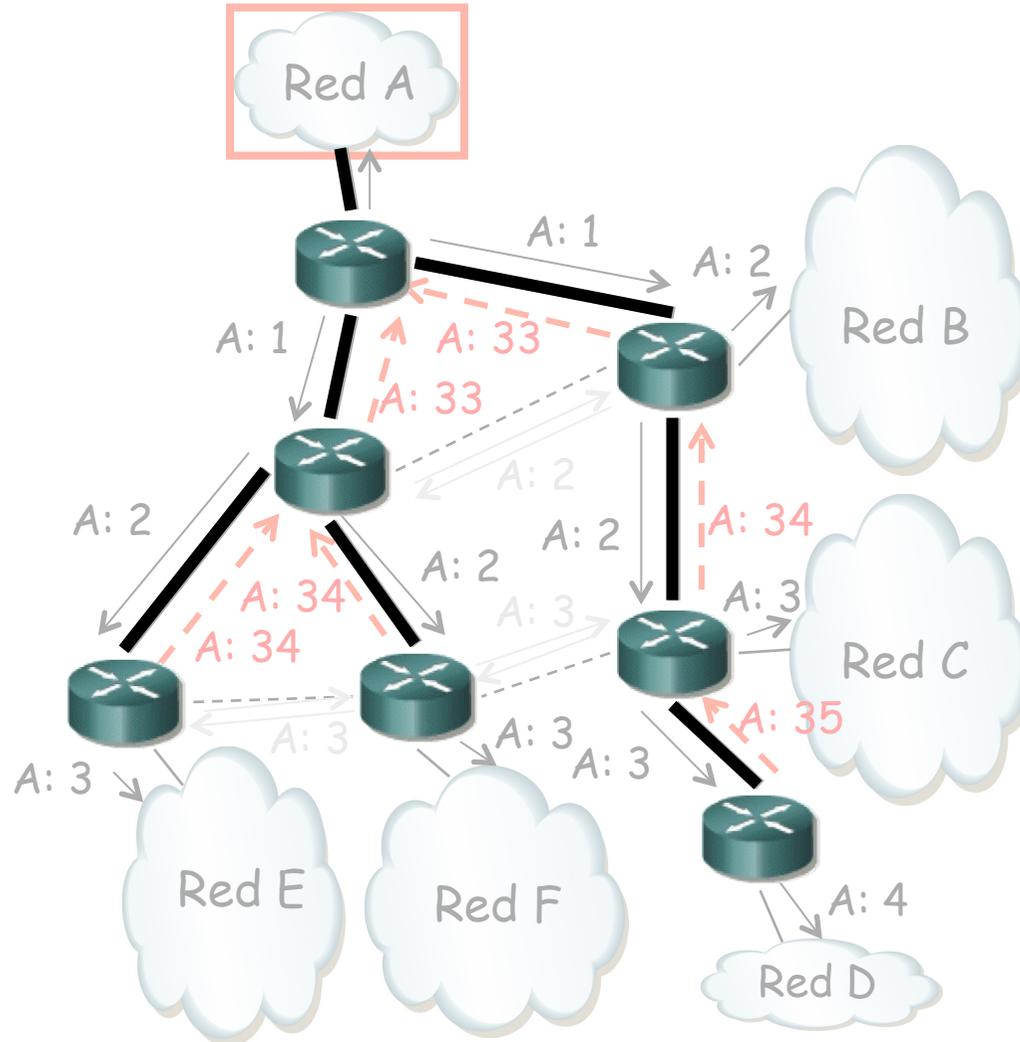
# Ejemplo



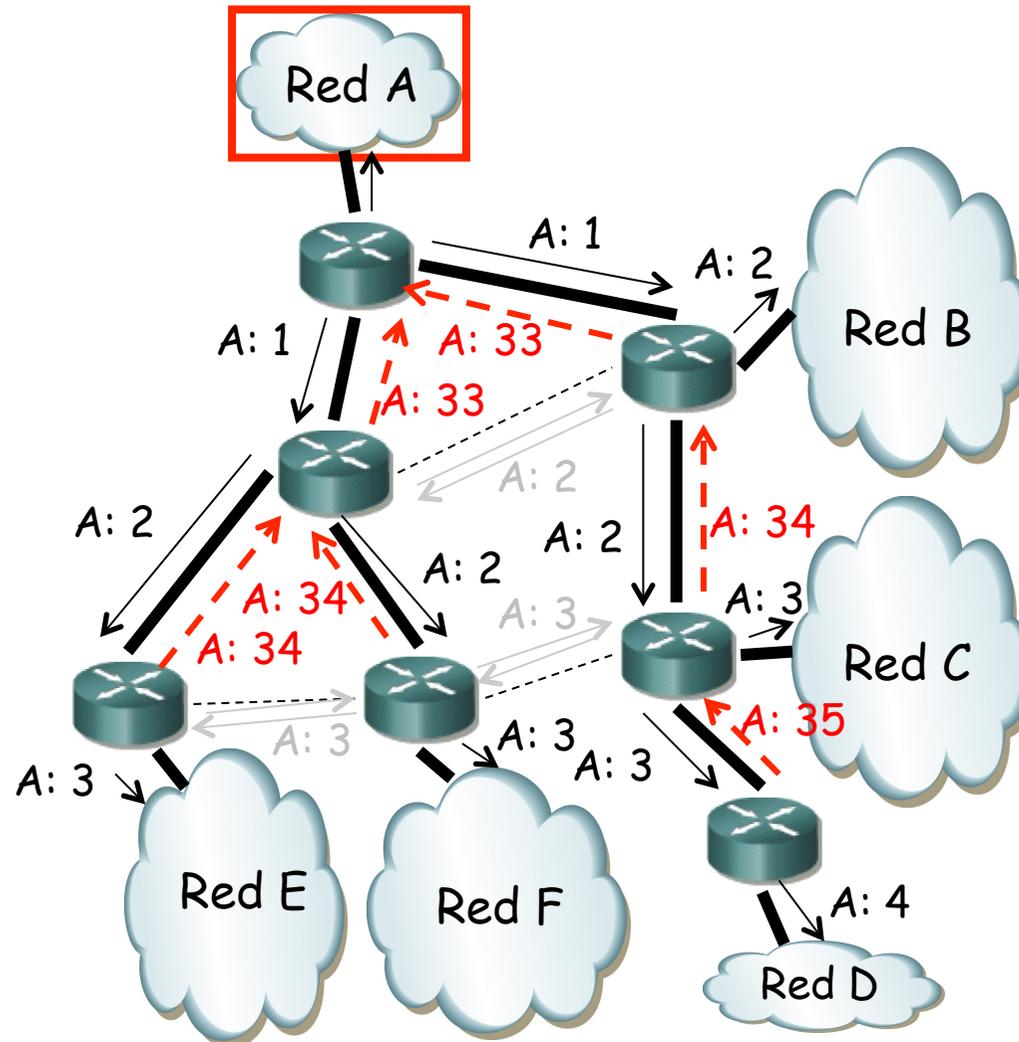
# Ejemplo



# Ejemplo

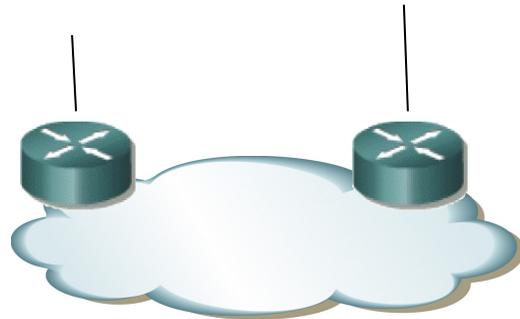


# Ejemplo



# Truncated Broadcast Trees

- Es un árbol para cada red origen
- En una LAN con varios m routers
  - Solo uno introducirá el tráfico (*designated forwarder*)
  - El que tenga el menor coste a la fuente
  - En caso de igual coste el que tenga menor dirección IP

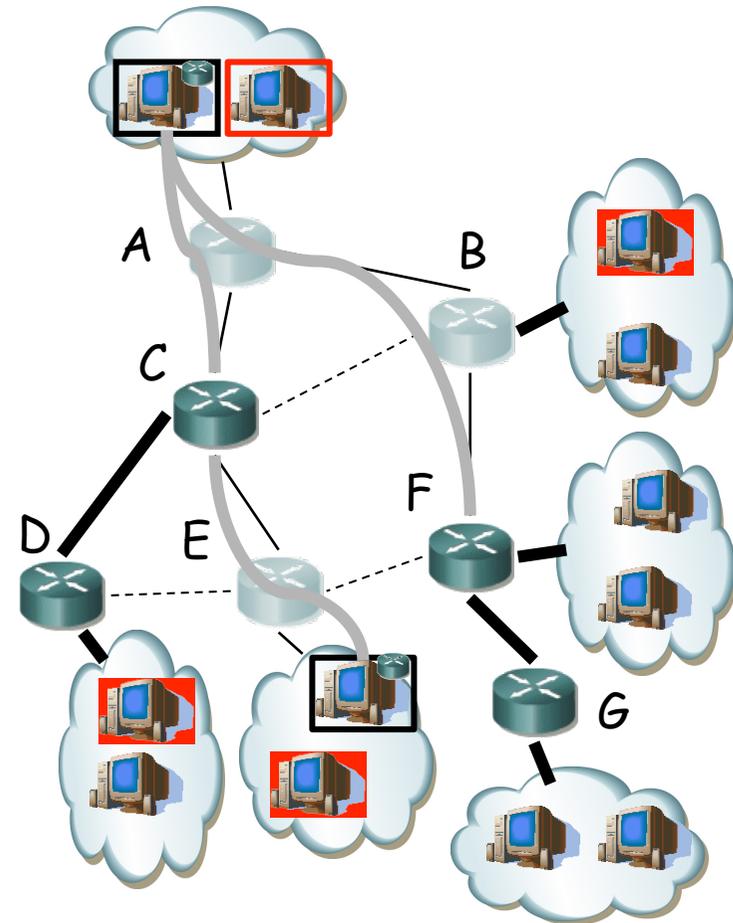


# RPF

- **Reverse Path Forwarding**
- Empleado por DVMRP para determinar si reenviar el paquete
- Examina la dirección origen del paquete multicast para determinar si ha llegado por el interfaz por el cual enviaría tráfico a ese host
- Si llega por ese interfaz (el *reverse path*) lo reenvía por el árbol
- Se emplea la tabla de rutas de DVMRP
- En otros protocolos emplea la tabla de rutas unicast (PIM, CBT)
- En funcionamiento estable no deberían llegar paquetes multicast por un interfaz incorrecto
- **Pruning**
  - Mrouters extremo sin receptores envían mensaje de poda *upstream*
  - Ascende por el árbol mientras los mrouter no tengan clientes *downstream*
  - La poda caduca y se reactivaría el *flooding* si no se retransmite la poda
- **Grafting**
  - Enviado *upstream* para reactivar *flooding* sin esperar a que caduque poda
  - Los mensajes son confirmados por el nodo *upstream*

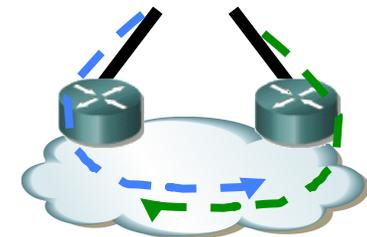
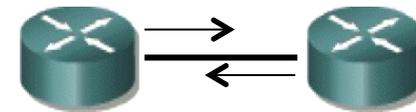
# MBONE

- Multicast Backbone
- Partiendo de una red en la que muchos routers no soportan multicast
- MRouters conectados entre sí con **túneles**
- Los mrouter han sido muchas veces estaciones UNIX
- DVMRP
- Session Description Protocol (SDP) para describir las sesiones
- Session Announcement Protocol (SAP) el protocolo de transporte empleado por SDP



# PIM

- *Protocol Independent Multicast* (independiente del protocolo de enrutamiento unicast que se emplee)
- Emplea la tabla de rutas del protocolo unicast para el RPF check
- Classless (si lo es el enrutamiento unicast)
- PIMv1 sobre IGMP, PIMv2 sobre IP (protocolo 103, TTL=1)
- *Dense Mode (PIM-DM, RFC 3973)*
  - *Source Distribution Trees (flood-and-prune)*
- *Sparse Mode (PIM-SM, RFC 4601)*
  - *Source or Shared Trees*
- **Neighbor Discovery**
  - Mensaje PIM Hello cada 30s dirigido a 224.0.0.13 (*All PIM Routers*)
  - Mantiene adyacencia durante 90s si no recibe de nuevo el Hello
  - En redes multiacceso sirve para seleccionar un *Designated Router* que es el de mayor prioridad (o mayor dirección IP)
- **Asserts**
  - Cuando un router recibe tráfico multicast por interfaz de salida
  - Envían mensaje de *Assert* para decidir quién sigue reenviando
  - Será el de menor coste al origen (o mayor IP si empatan)



# PIM-DM

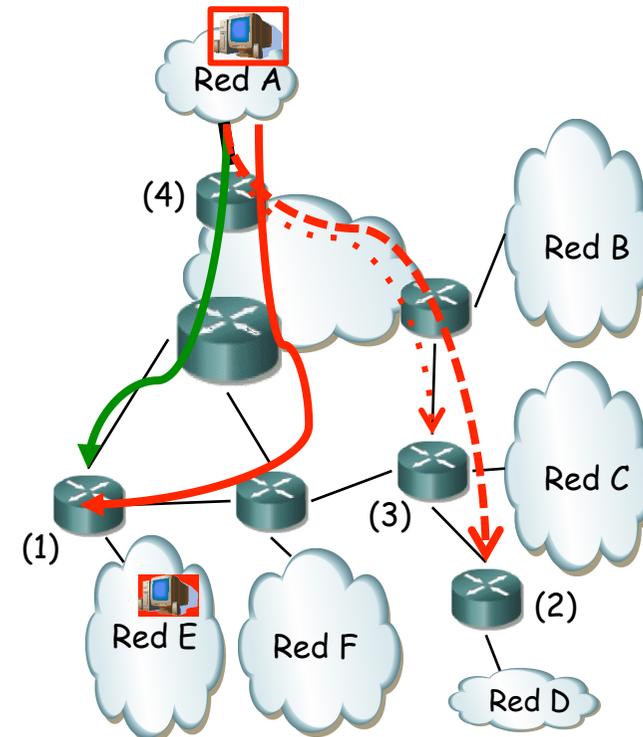
- Inicialmente inunda la red, usando RPF
- Si hay más de una entrada en la tabla de rutas para ese destino (ECMP) comprueba con la de mayor dirección de next-hop

## Pruning

- Para tráfico que llega por un interfaz p2p no RPF (1)
- Cuando es un router extremo sin receptores conectados (2)
- Router sin receptores conectados y que ha recibido un *Prune* de sus vecinos no RPF (3)
- Router que ha recibido *Prune* de LAN, mientras otro no lo anule (*override*) con un *Join* en menos de 3s (4)
- El estado *pruned* caduca
- Se puede evitar con mensajes de *State Refresh* que envía el router adyacente a la fuente por el árbol refrescando las podas

## Grafting

- Para reinjertarse en el árbol
- Ante mensajes IGMP de receptores o ante *Grafts* de routers downstream
- Se confirman (*Graft* y *Graft-ACK* son unicast)

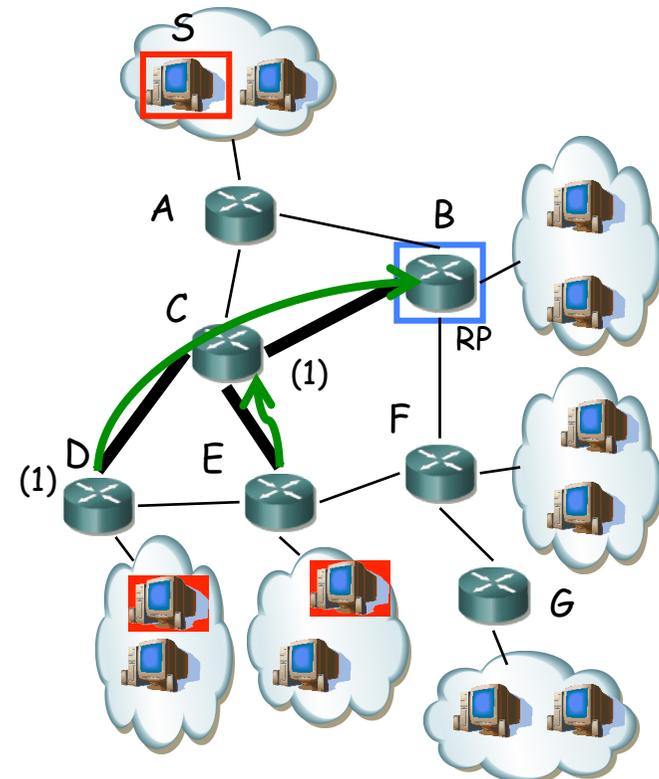




# PIM-SM

## Rendezvous Point Tree (RPT)

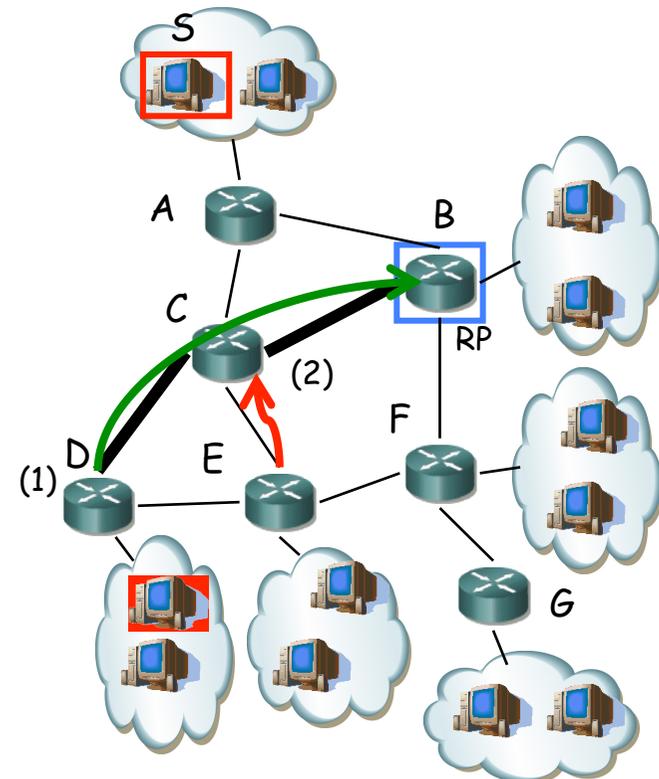
- Solo envía a las zonas de la red que lo han solicitado
- El *rendezvous point* (RP) es la raíz del *shared tree* y el árbol un RPT
- Mrouter con un receptor unido al grupo por IGMP envía *Join* hacia el RP
- Mrouters en el camino ponen al interfaz por el que reciben el *Join* en la lista de interfaces por donde reenviar para (\*,G) y envían hacia el RP (1)
- Si mrouter ya tiene estado para (\*,G) no envía el *Join* hacia el RP (2)
- (...)



# PIM-SM

## Rendezvous Point Tree (RPT)

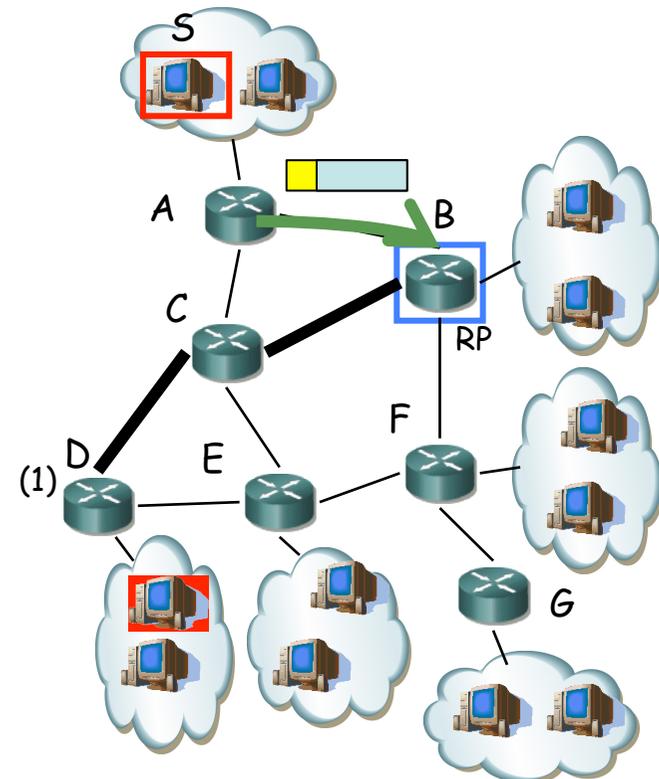
- Solo envía a las zonas de la red que lo han solicitado
- El *rendezvous point* (RP) es la raíz del *shared tree* y el árbol un RPT
- Mrouter con un receptor unido al grupo por IGMP envía *Join* hacia el RP
- M routers en el camino ponen al interfaz por el que reciben el *Join* en la lista de interfaces por donde reenviar para (\*,G) y envían hacia el RP (1)
- Si mrouter ya tiene estado para (\*,G) no envía el *Join* hacia el RP (2)
- Si un router queda sin receptores envía *Prune*
- El *Join/Prune* se reenvía periódicamente
- En LANs emplea el mismo mecanismo con *override* que PIM-DM



# PIM-SM

## Rendezvous Point Tree (RPT)

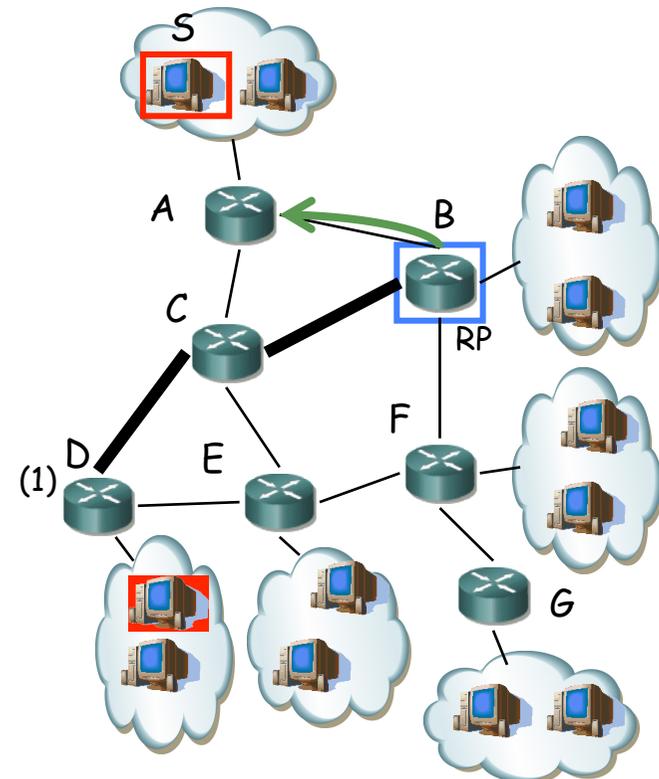
- La fuente envía los paquetes al RP encapsulados en paquetes unicast a él
- El RP distribuye desde él el paquete nativo multicast por el RPT
- (...)



# PIM-SM

## Rendezvous Point Tree (RPT)

- La fuente envía los paquetes al RP encapsulados en paquetes unicast a él
- El RP distribuye desde él el paquete nativo multicast por el RPT
- RP envía *Join* a (S,G) upstream
- (...)



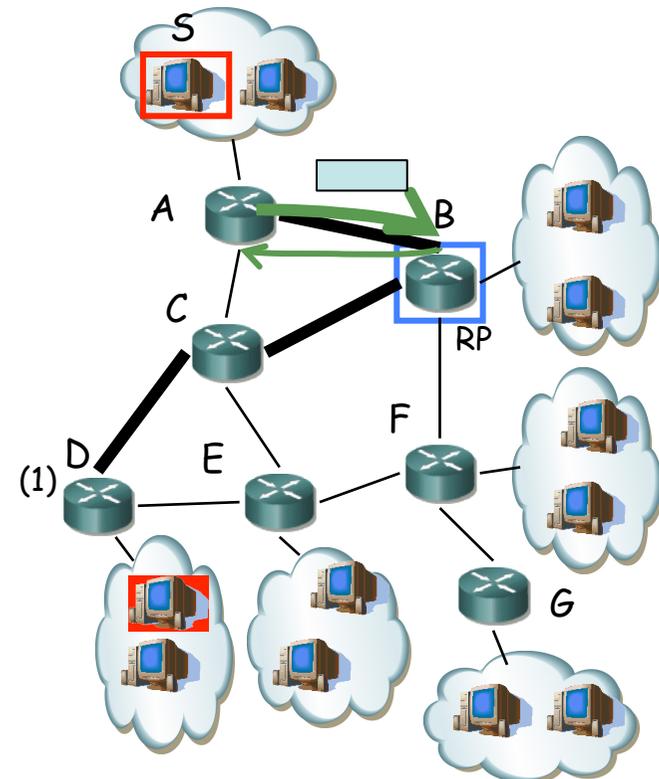
# PIM-SM

## Rendezvous Point Tree (RPT)

- La fuente envía los paquetes al RP encapsulados en paquetes unicast a él
- El RP distribuye desde él el paquete nativo multicast por el RPT
- RP envía *Join* a (S,G) upstream
- Fuente empieza a enviar los paquetes en multicast por ese árbol (S,G) hasta el RP
- Al recibirlos, el RP envía un *Register-Stop* para dejar de recibir los encapsulados unicast
- Desde S al RP fluye el tráfico por un (S,G) y desde el RP fluye por un (\*,G)

## RP discovery

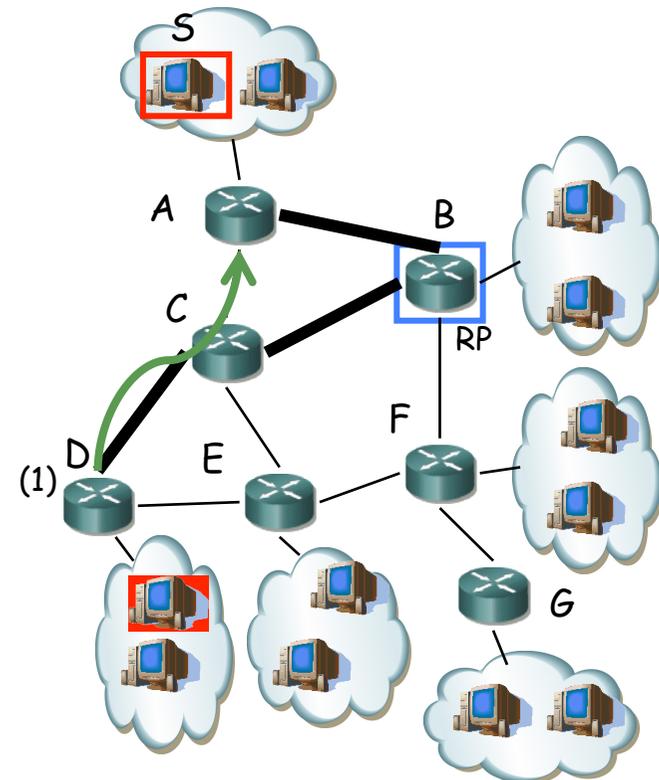
- El RP para un grupo puede estar configurado estáticamente en el resto de m routers
- O se puede descubrir automáticamente



# PIM-SM

## Shortest Path Tree (SPT)

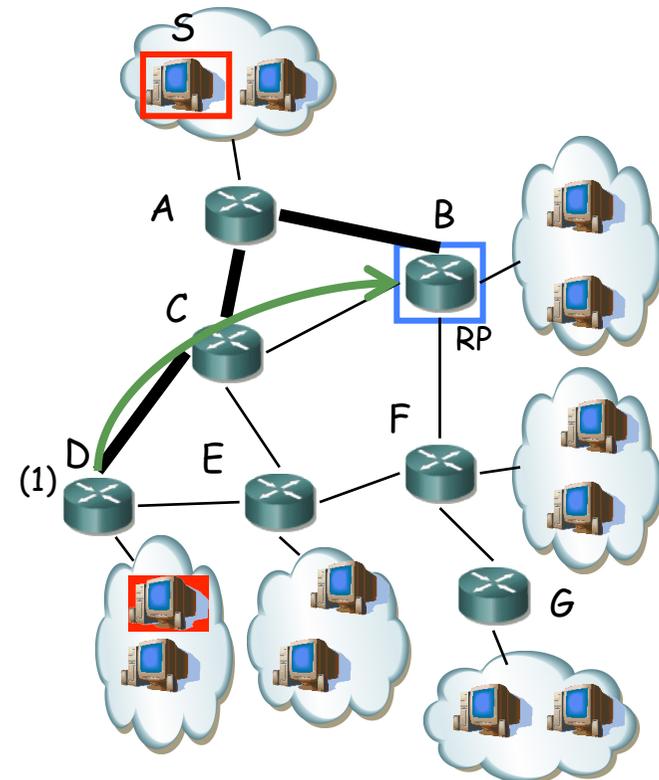
- El camino por el *shared tree* puede no ser rentable
- Mrouter junto a cliente puede enviar un *Join* a (S,G)
- Eso implica un árbol desde la fuente
- (...)



# PIM-SM

## Shortest Path Tree (SPT)

- El camino por el *shared tree* puede no ser rentable
- Mrouter junto a cliente puede enviar un *Join* a (S,G)
- Eso implica un árbol desde la fuente
- Cuando le llega el tráfico por los dos árboles envía un *Prune* del (\*,G)
- Si el RP está *pruned* por el resto de interfaces puede enviar *Prune* hacia S



# Otros protocolos

## MOSPF

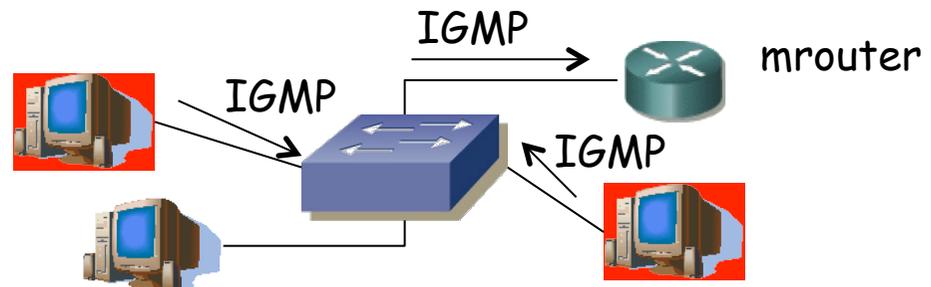
- Multicast Extensions to OSFP (RFC 1584)
- LSAs de pertenencia a un grupo
- Crea *source trees*
- Dijkstra para cada (S,G) y cada vez que se une un nuevo receptor

## MBGP

- Multiprotocol Extensios for BGP-4 (RFC 4760)
- Para transportar información para más protocolos que IPv4 (IPv6, IPX...), unicast y multicast
- Distribuye la información que se puede emplear para el RPF de forma independiente para el tráfico multicast que para el unicast
- No calcula árboles, sigue haciendo falta otro protocolo

# IGMP snooping

- Switches hacen inundación del tráfico multicast
- Un switch que hace *IGMP snooping* procesa los mensajes IGMP para decidir por dónde no hace falta que reenvíe el tráfico multicast
- RFC 4541 “*Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches*”
- RFC sobre implementaciones y recomendaciones
- El switch procesa los *IGMP Report* de los hosts para saber por dónde reenviar (siempre *flooding* para paquetes no IGMP a 224.0.0.0/24)
- En la CAM mantendrá información indicando por qué puertos reenviar los paquetes que van a una MAC multicast
- Reenvía los *IGMP Report* solo por los puertos donde haya mrouter
  - Evita así que otros hosts del grupo supriman sus *Reports*
  - Si los suprimen no sabrá si siguen perteneciendo al grupo
  - El mrouter debe anunciarse como tal, por ejemplo con *IGMP Queries*



# Resumen

- De uno a muchos
- Rango de direcciones reservado
- Unirse y abandonar grupos
- Reverse Path Forwarding
- Enrutamiento empleando árboles
- Raíz en origen o en un *rendezvous point*
- En LANs reducir tráfico por inundación mediante IGMP *snooping*