

Network Layer

Broadcast & Multicast

Broadcasting

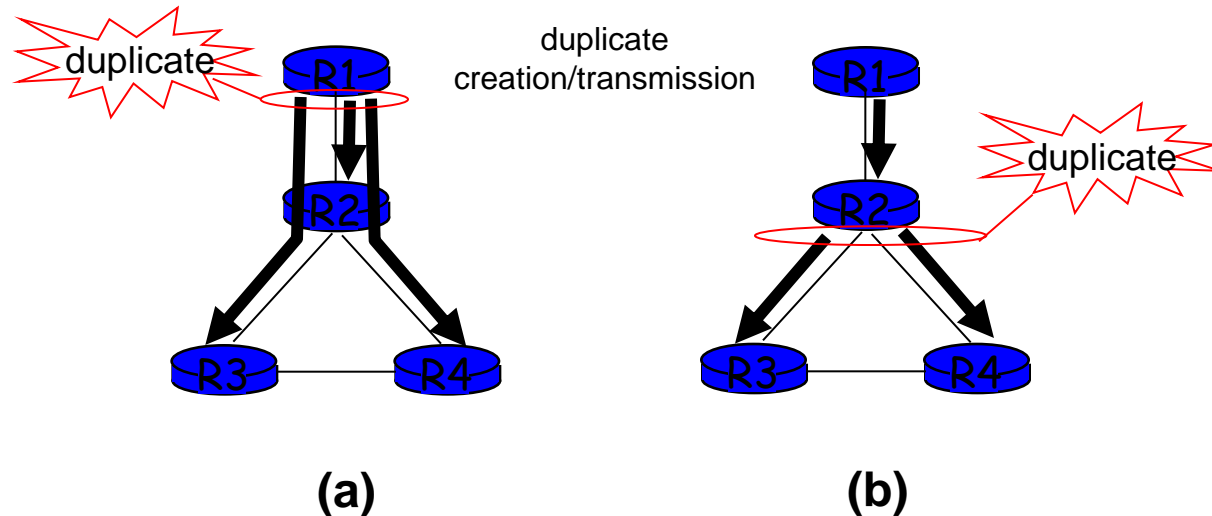
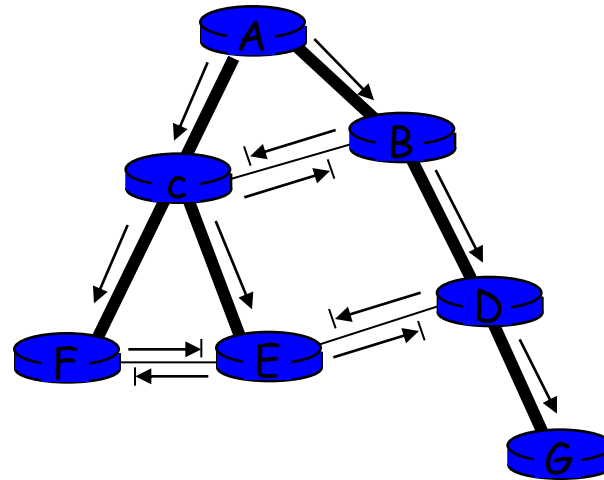


Figure 4.39 Source-duplication versus in-network duplication.
(a) source duplication, (b) in-network duplication

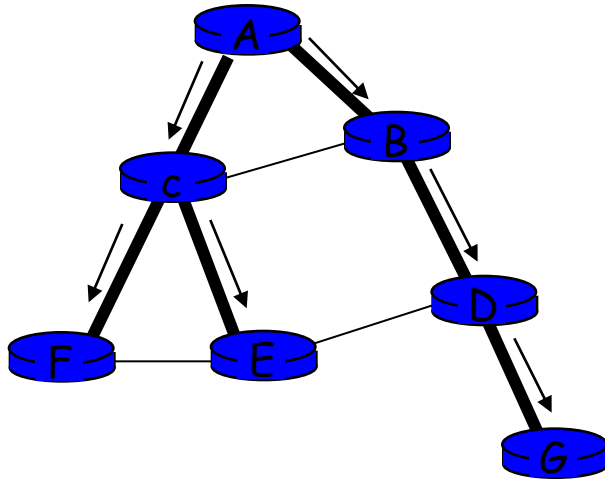
Broadcasting Problem



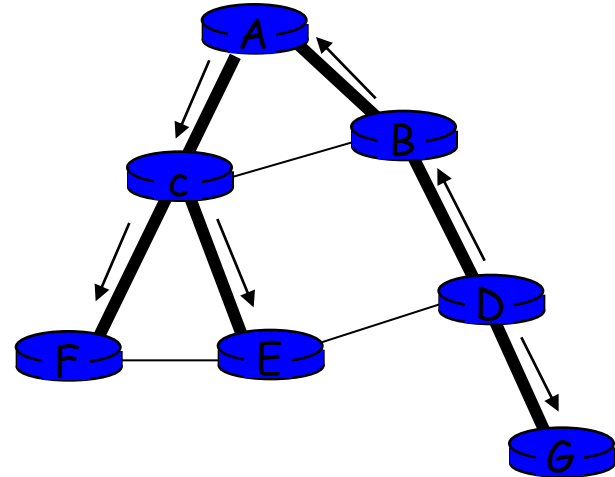
How to fix this?

Figure 4.40: Reverse path forwarding

Broadcasting Along a Spanning Tree

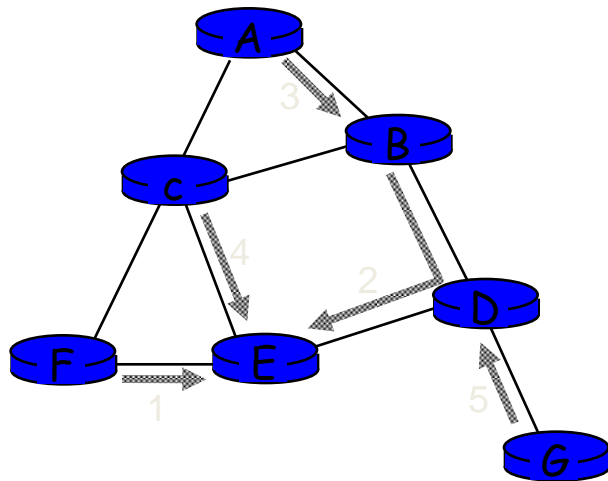


(a) Broadcast initiated at A

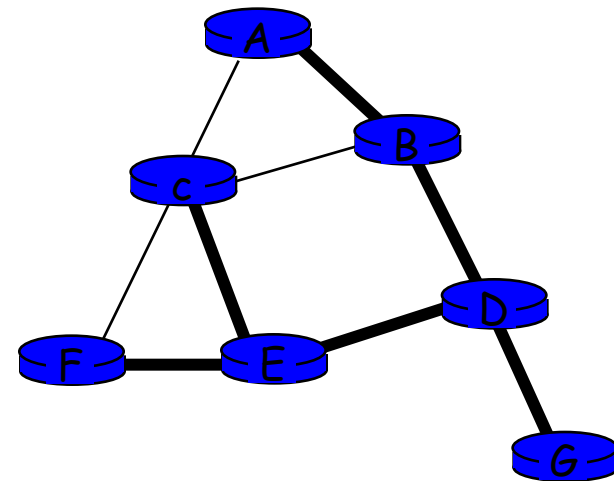


(b) Broadcast initiated at D

Figure 4.41: Broadcast along a spanning tree



(a) Stepwise construction of spanning tree

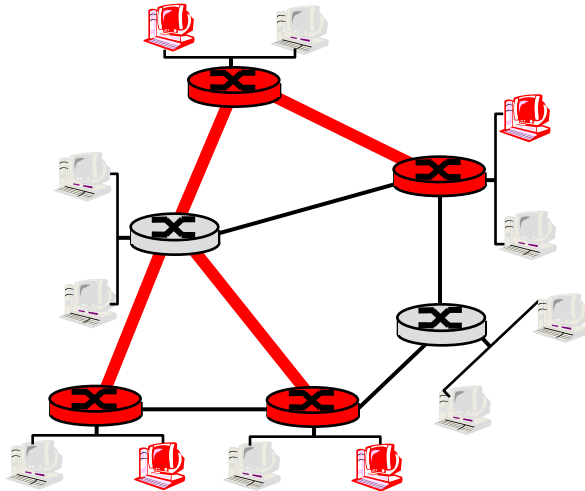


(b) Constructed spanning tree

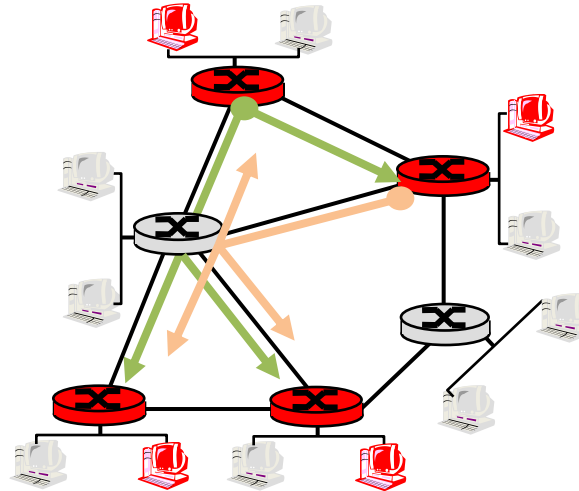
Figure 4.42: Center-based construction of a spanning tree

Multicast Routing: Problem Statement

- **Goal:** find a tree (or trees) connecting routers having local mcast group members
 - tree: not all paths between routers used
 - source-based: different tree from each sender to rcvrs
 - shared-tree: same tree used by all group members



Shared tree



Source-based trees

Approaches for building mcast trees

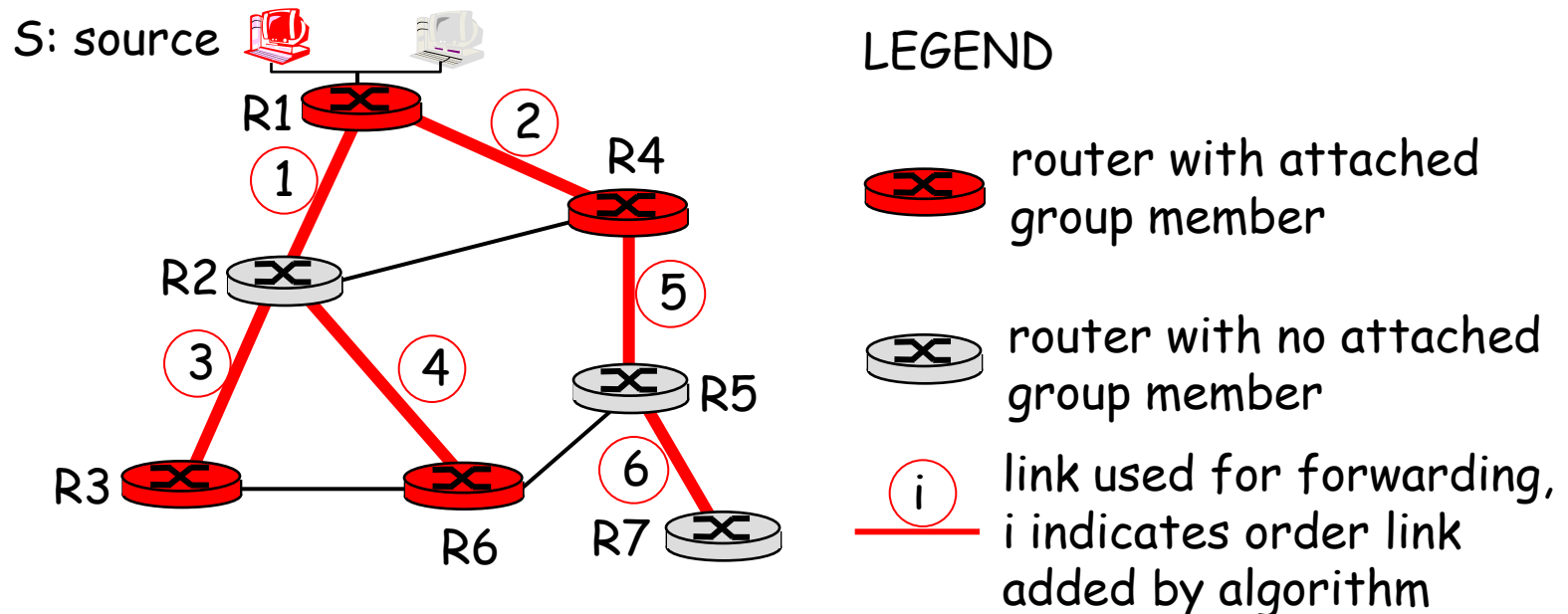
Approaches:

- source-based tree: one tree per source
 - shortest path trees
 - reverse path forwarding
- group-shared tree: group uses one tree
 - minimal spanning (Steiner)
 - center-based trees

...we first look at basic approaches, then specific protocols adopting these approaches

Shortest Path Tree

- mcast forwarding tree: tree of shortest path routes from source to all receivers
 - Dijkstra's algorithm

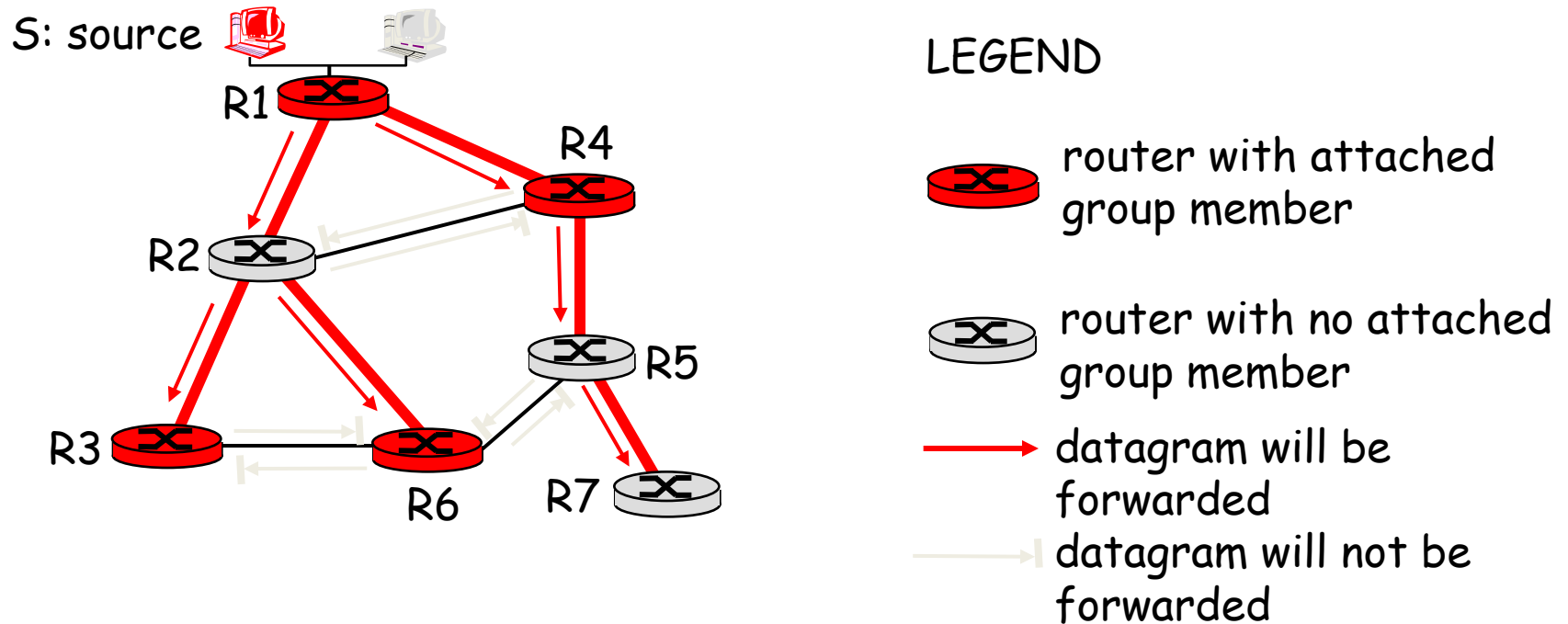


Reverse Path Forwarding

- ❑ rely on router's knowledge of unicast shortest path from it to sender
- ❑ each router has simple forwarding behavior:

if (mcast datagram received on incoming link on shortest path back to center)
then flood datagram onto all outgoing links
else ignore datagram

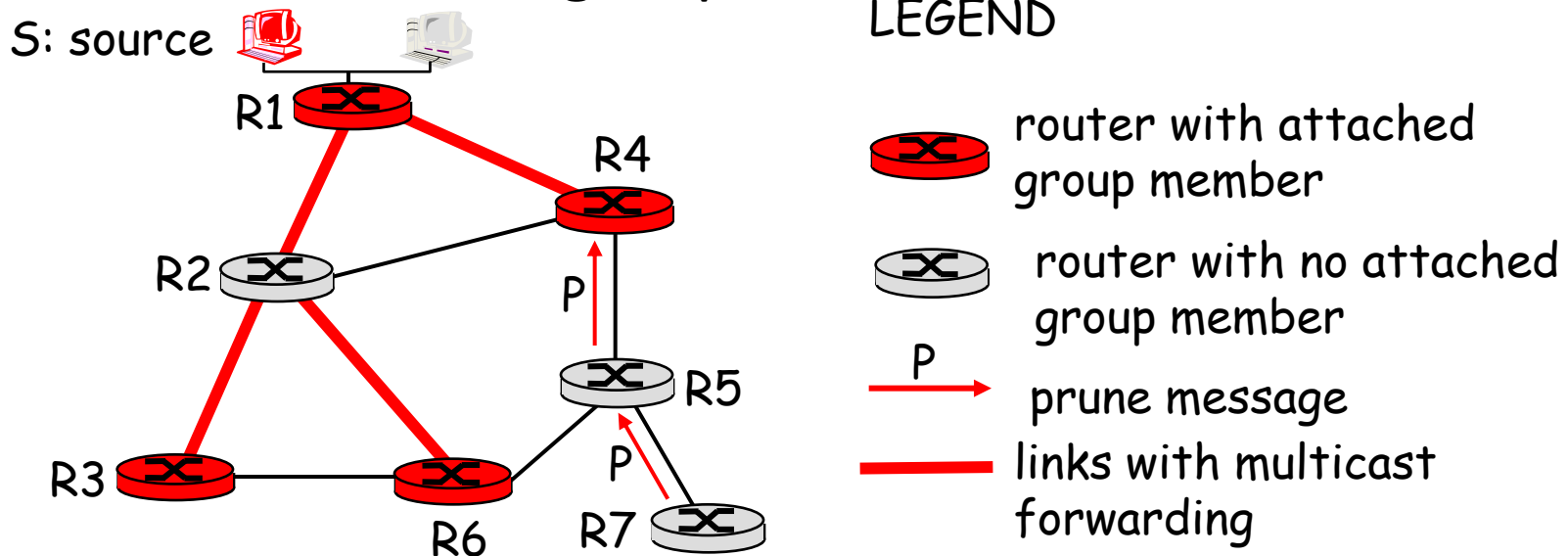
Reverse Path Forwarding: example



- result is a source-specific *reverse* SPT
 - may be a bad choice with asymmetric links

Reverse Path Forwarding: pruning

- forwarding tree contains subtrees with no mcast group members
 - no need to forward datagrams down subtree
 - “prune” msgs sent upstream by router with no downstream group members



Shared-Tree: Steiner Tree

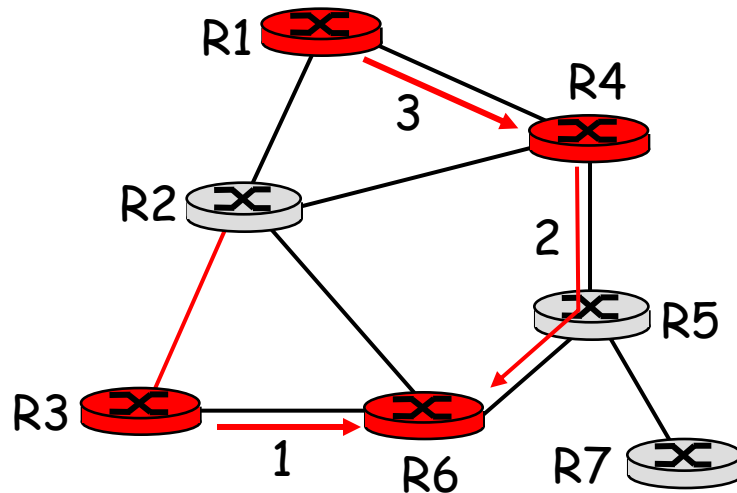
- Steiner Tree: minimum cost tree connecting all routers with attached group members
- problem is NP-complete
- excellent heuristics exists
- not used in practice:
 - computational complexity
 - information about entire network needed
 - monolithic: rerun whenever a router needs to join/leave

Center-based trees




- single delivery tree shared by all
- one router identified as “*center*” of tree
- to join:
 - edge router sends unicast *join-msg* addressed to center router
 - *join-msg* “processed” by intermediate routers and forwarded towards center
 - *join-msg* either hits existing tree branch for this center, or arrives at center
 - path taken by *join-msg* becomes new branch of tree for this router

Center-based trees: an example

Suppose R6 chosen as center:



LEGEND

-  router with attached group member
-  router with no attached group member
-  path order in which join messages generated

Internet Multicasting Routing: DVMRP

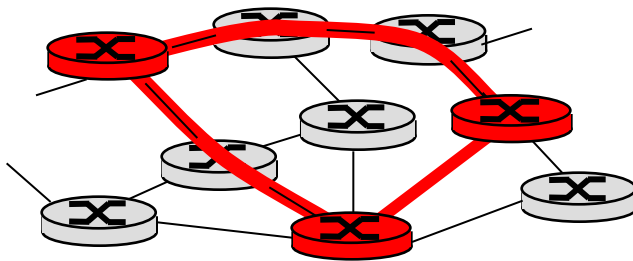
- DVMRP: distance vector multicast routing protocol, RFC1075
- *flood and prune*: reverse path forwarding, source-based tree
 - RPF tree based on DVMRP's own routing tables constructed by communicating DVMRP routers
 - no assumptions about underlying unicast
 - initial datagram to mcast group flooded everywhere via RPF
 - routers not wanting group: send upstream prune msgs

DVMRP: continued...

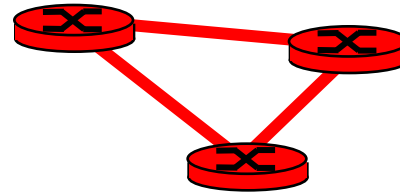
- soft state: DVMRP router periodically (1 min.) “forgets” branches are pruned:
 - mcast data again flows down unpruned branch
 - downstream router: reprune or else continue to receive data
- routers can quickly regraft to tree
 - following IGMP join at leaf
- odds and ends
 - commonly implemented in commercial routers
 - Mbone routing done using DVMRP

Tunneling

Q: How to connect “islands” of multicast routers in a “sea” of unicast routers?



physical topology



logical topology

- ❑ mcast datagram encapsulated inside “normal” (non-multicast-addressed) datagram
- ❑ normal IP datagram sent thru “tunnel” via regular IP unicast to receiving mcast router
- ❑ receiving mcast router unencapsulates to get mcast datagram

PIM: Protocol Independent Multicast

- not dependent on any specific underlying unicast routing algorithm (works with all)
- two different multicast distribution scenarios :

Dense:

- ❑ group members densely packed, in "close" proximity.
- ❑ bandwidth more plentiful

Sparse:

- ❑ # networks with group members small wrt # interconnected networks
- ❑ group members "widely dispersed"
- ❑ bandwidth not plentiful

Consequences of Sparse-Dense Dichotomy:

Dense

- group membership by routers *assumed* until routers explicitly prune
- *data-driven* construction on mcast tree (e.g., RPF)
- bandwidth and non-group-router processing *profligate*

Sparse:

- no membership until routers explicitly join
- *receiver-driven* construction of mcast tree (e.g., center-based)
- bandwidth and non-group-router processing *conservative*

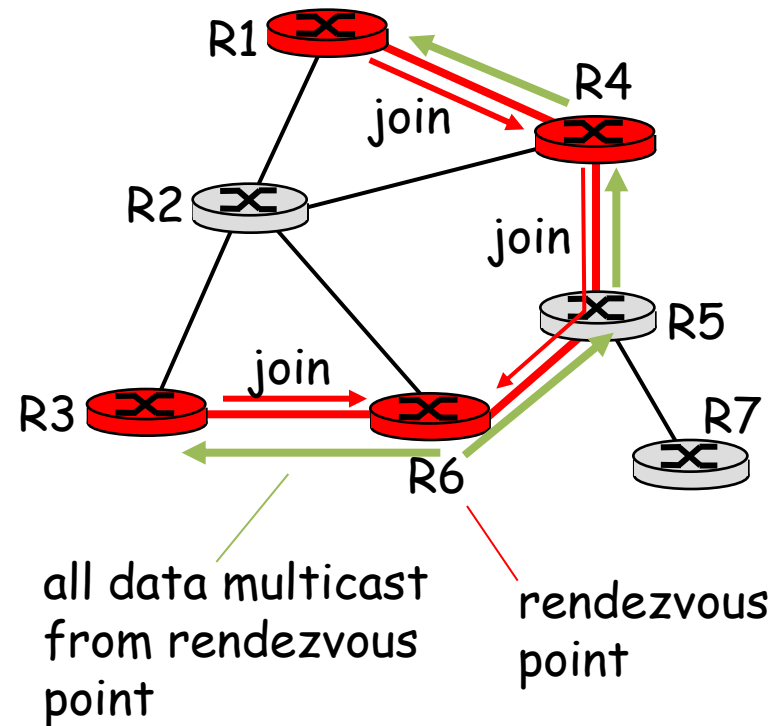
PIM- Dense Mode

flood-and-prune RPF, similar to DVMRP but

- ❑ underlying unicast protocol provides RPF info for incoming datagram
- ❑ less complicated (less efficient) downstream flood than DVMRP reduces reliance on underlying routing algorithm
- ❑ has protocol mechanism for router to detect it is a leaf-node router

PIM - Sparse Mode

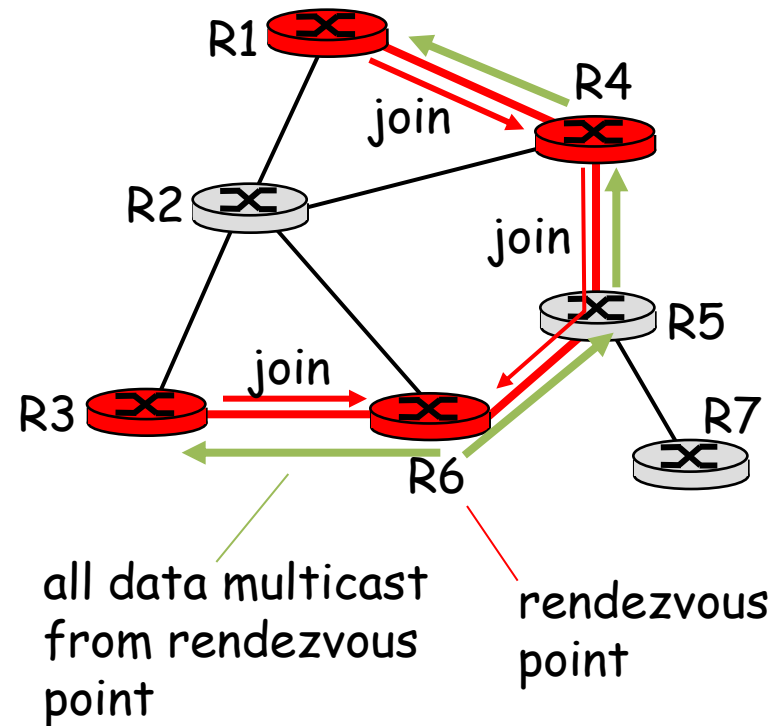
- center-based approach
- router sends *join* msg to rendezvous point (RP)
 - intermediate routers update state and forward *join*
- after joining via RP, router can switch to source-specific tree
 - increased performance: less concentration, shorter paths



PIM - Sparse Mode

sender(s):

- unicast data to RP, which distributes down RP-rooted tree
- RP can extend mcast tree upstream to source
- RP can send *stop* msg if no attached receivers
 - “no one is listening!”



Network Layer: summary

What we've covered:

- network layer services
- routing principles: link state and distance vector
- hierarchical routing
- IP
- Internet routing protocols RIP, OSPF, BGP
- what's inside a router?
- IPv6

Next stop:

the Data
link layer!