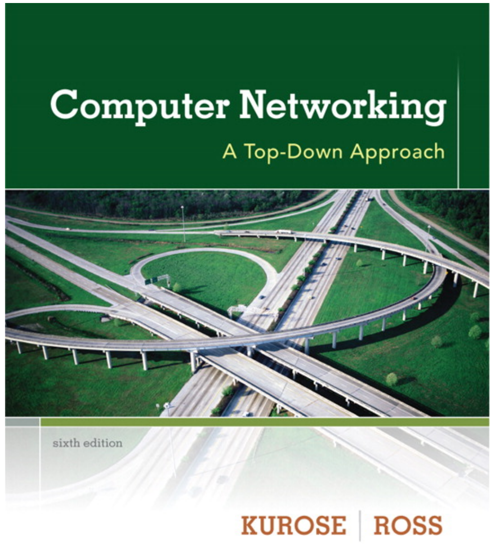
Chapter 4
Network Layer



Reti degli Elaboratori
Canale AL
Prof.ssa Chiara Petrioli
a.a. 2013/2014

Computer Networking: A Top Down Approach
6th edition
Jim Kurose, Keith Ross
Addison-Wesley
March 2012

We thank for the support material Prof. Kurose-Ross

Chapter 4: outline

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

4.4 IP: Internet Protocol

- datagram format
- IPv4 addressing
- ICMP
- IPv6

4.5 routing algorithms

- link state
- distance vector
- hierarchical routing

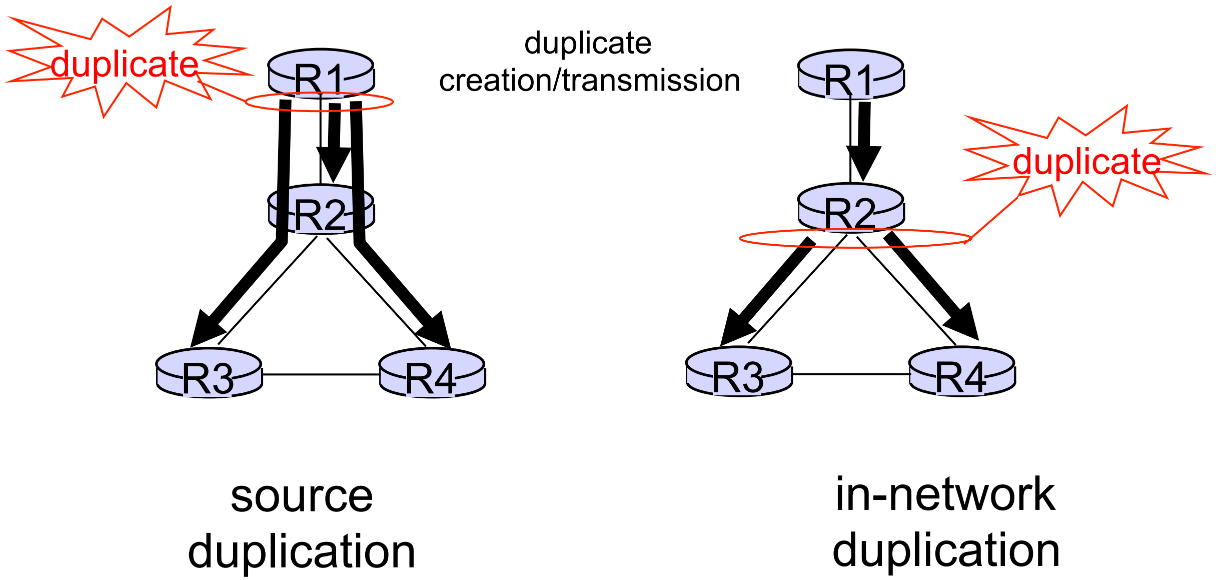
4.6 routing in the Internet

- RIP
- OSPF
- BGP

4.7 broadcast and multicast routing

Broadcast routing

- deliver packets from source to all other nodes
- source duplication is inefficient:

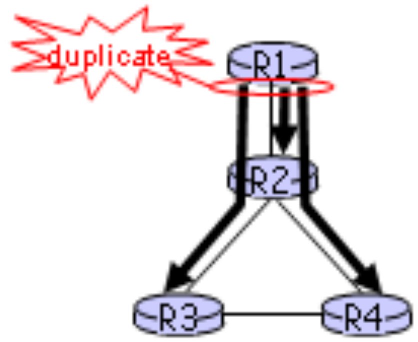


- source duplication: how does source determine recipient addresses?

In-network duplication

- *flooding:* when node receives broadcast packet, sends copy to all neighbors
  - problems: cycles & broadcast storm
- *controlled flooding:* node only broadcasts pkt if it hasn't broadcast same packet before
  - node keeps track of packet ids already broadcasted
  - or reverse path forwarding (RPF): only forward packet if it arrived on shortest path between node and source
- *spanning tree:*
  - no redundant packets received by any node

Unicast ad N vie

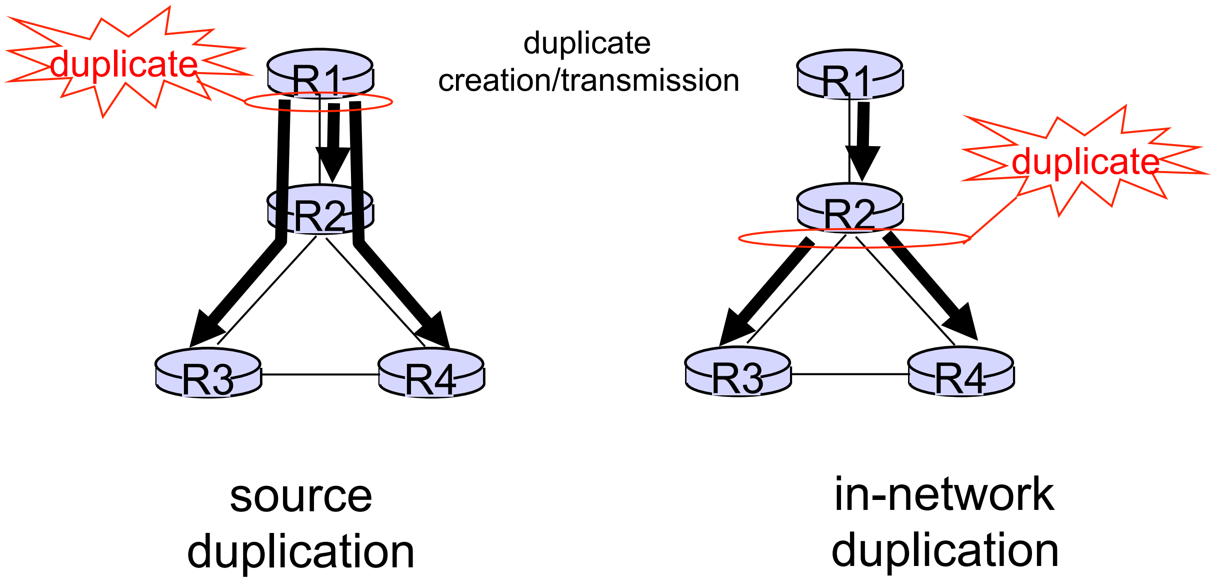


source duplication

- Inefficiente
  - Un singolo collegamento attraversato da N copie del messaggio se il nodo origine è connesso al resto della rete tramite un unico collegamento
- Indirizzi di tutte le destinazioni devono essere noti al mittente
  - altri meccanismi protocollari sono richiesti
- Broadcast può essere usato per inoltrare informazioni di topologia in una situazione in cui le rotte non sono ancora note
  - es. OSPF

Broadcast Routing

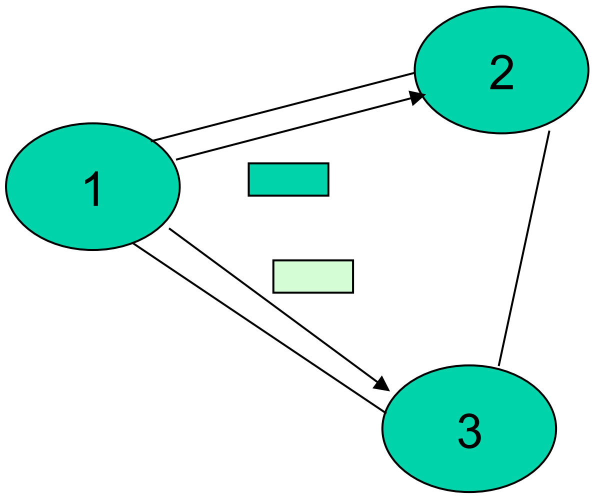
- deliver packets from source to all other nodes
- source duplication is inefficient:



- source duplication: how does source determine recipient addresses?

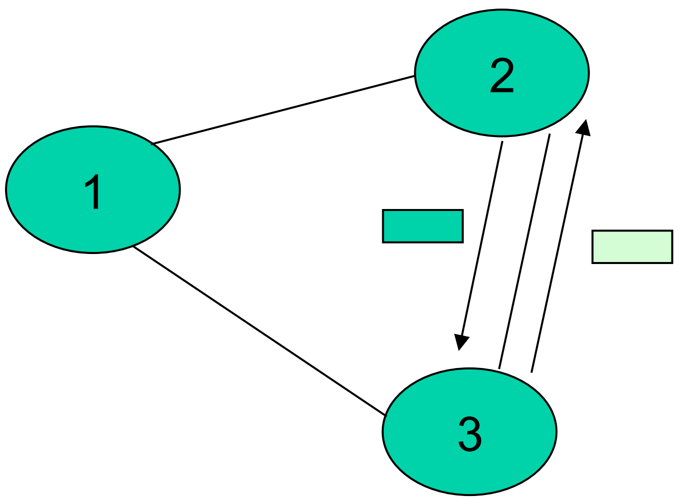
In-network duplication

- flooding: when node receives brdcst pckt, sends copy to all neighbors EXCEPT the one from which the pckt was received
  - Problems: cycles & broadcast storm



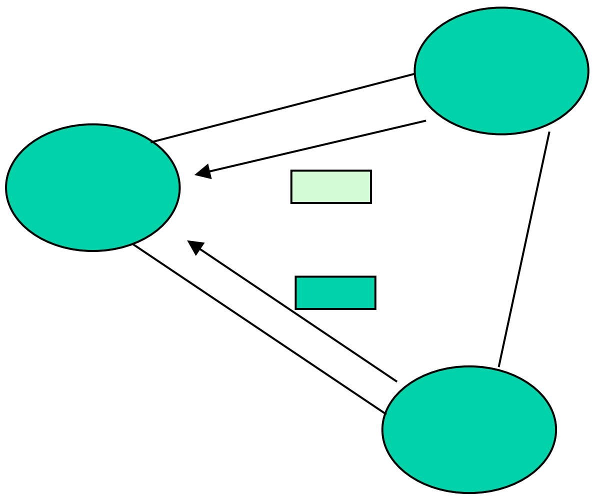
In-network duplication

- flooding: when node receives brdcst pckt, sends copy to all neighbors
  - Problems: cycles & broadcast storm



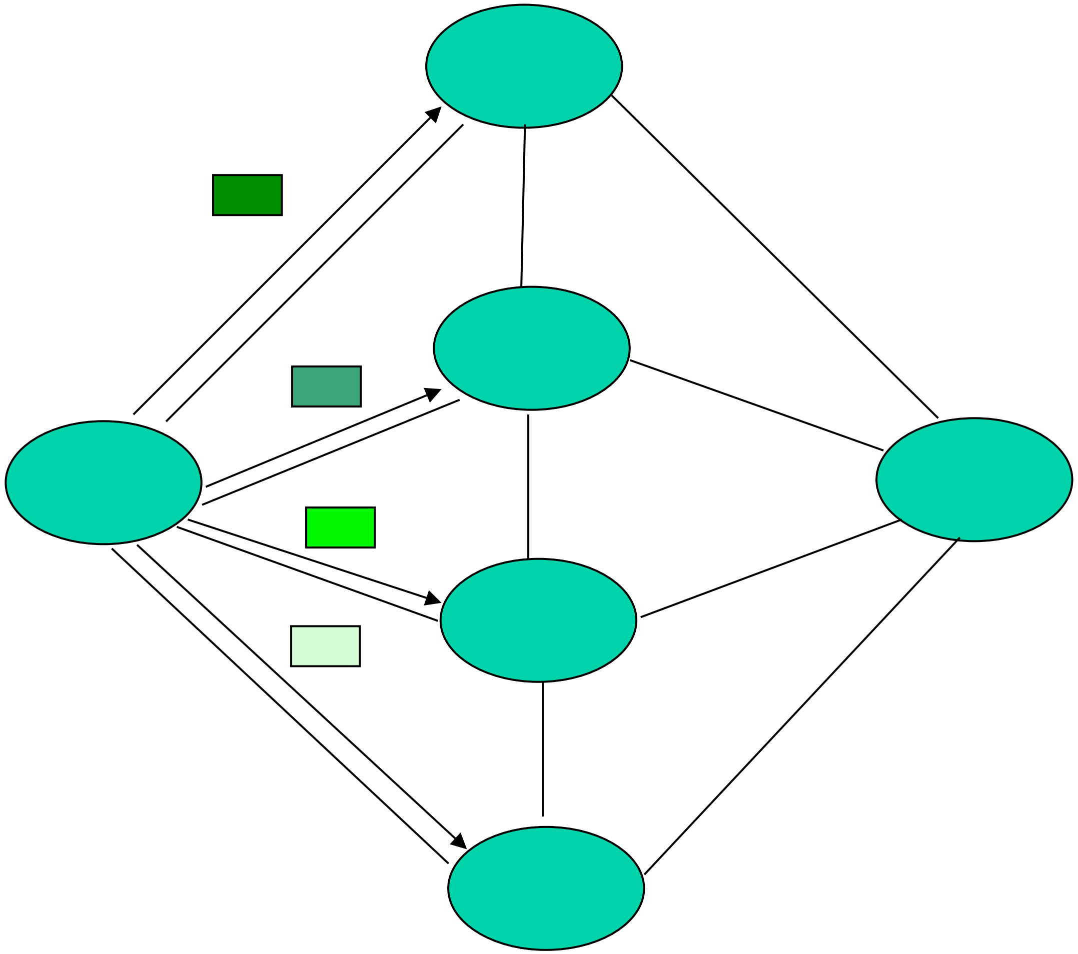
In-network duplication

- flooding: when node receives brdcst pckt, sends copy to all neighbors
  - Problems: cycles & broadcast storm

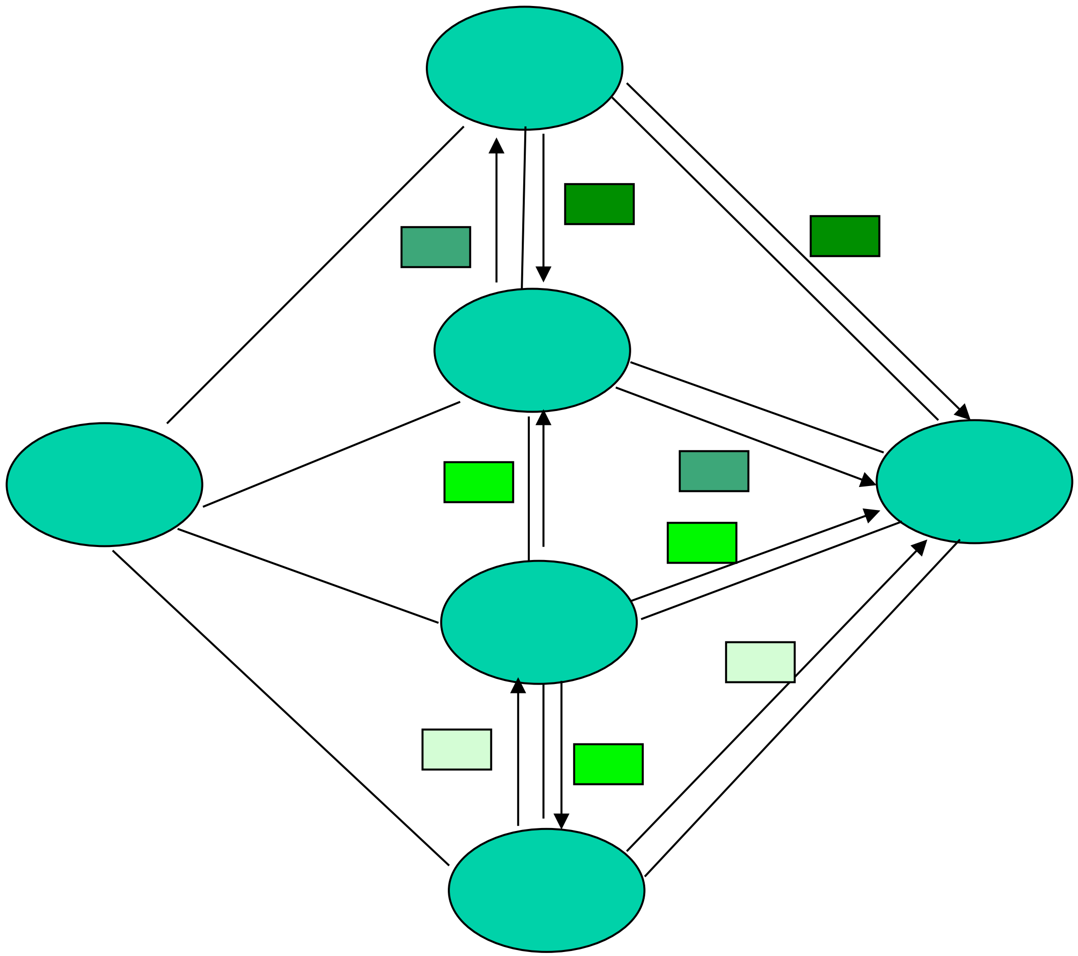


E ricominciamo come nella prima situazione
Bisogna saper distinguere tra quando mandiamo un nuovo messaggio e quando stiamo ritrasmettendo qualcosa che abbiamo già visto
→ Sequence numbers!

Broacast storm

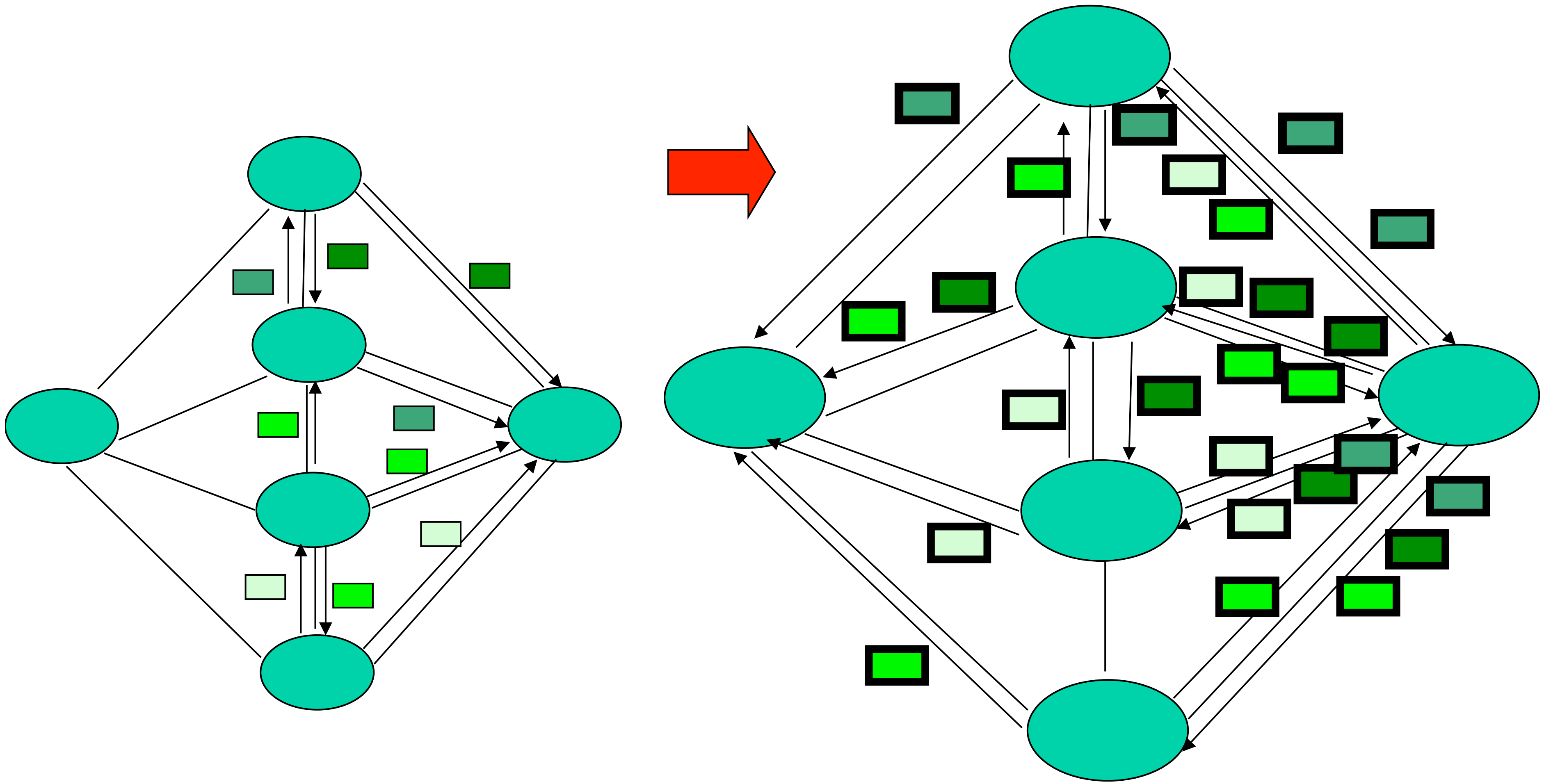


Broacast storm



Broacast storm

Il numero di pacchetti in rete cresce significativamente!!



Controlled flooding

- Il nodo origine pone il proprio indirizzo ed il numero di sequenza nei pacchetti che invia in broadcast
- Ciascun nodo mantiene una lista di ID origine, SEQN per i broadcast ricevuti, trasmesso o inoltrato
- Se riceve un pacchetto broadcast per prima cosa verifica se <ID, SEQN> compare nella lista dei pacchetti già gestiti
  - Se si scarta
  - Altrimenti riinvia su tutte le interfacce tranne quella da cui ha ricevuto

Controlled flooding, altre opzioni

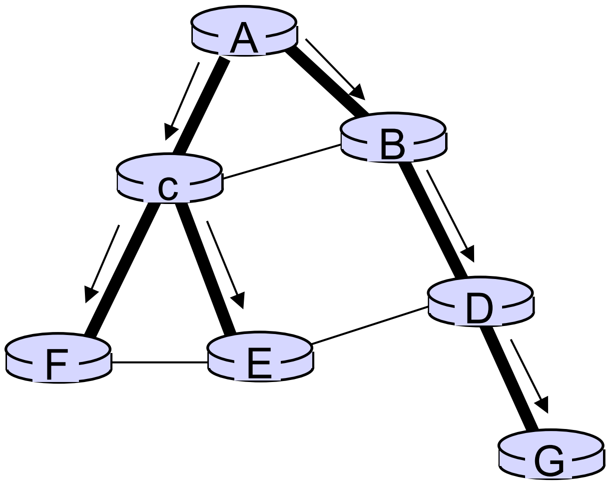
- Reverse path forwarding (RPF): only forward pckt (on all links but the one from which the packet was received) if it arrived on shortest path between node and source

In-network duplication

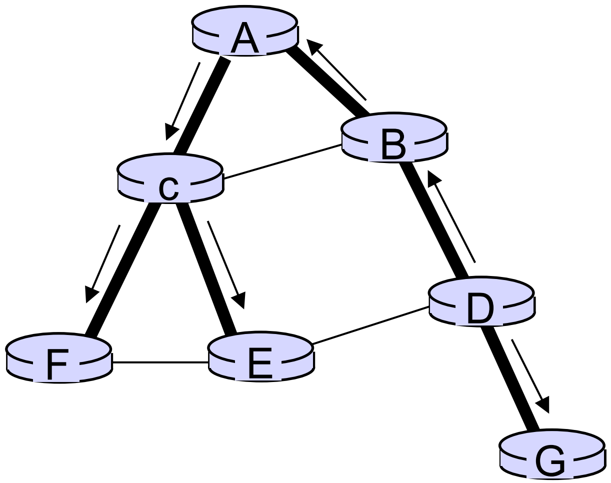
- *flooding:* when node receives broadcast packet, sends copy to all neighbors
  - problems: cycles & broadcast storm
- *controlled flooding:* node only broadcasts pkt if it hasn't broadcast same packet before
  - node keeps track of packet ids already broadcasted
  - or reverse path forwarding (RPF): only forward packet if it arrived on shortest path between node and source
- *spanning tree:*
  - no redundant packets received by any node

Spanning Tree

- First construct a spanning tree
- Nodes forward copies only along spanning tree



(a) Broadcast initiated at A



(b) Broadcast initiated at D

Minimum spanning tree- Prim’s Algorithm

- Prim's algorithm:

```
let T be a single vertex x
while (T has fewer than n vertices)  {
        Find the smallest edge connecting T to G-T
        Add it to T
}
```

Minimum spanning tree--Kruskal algorithm

- Kruskal's algorithm:

  Sort the edges of G in increasing order of weight

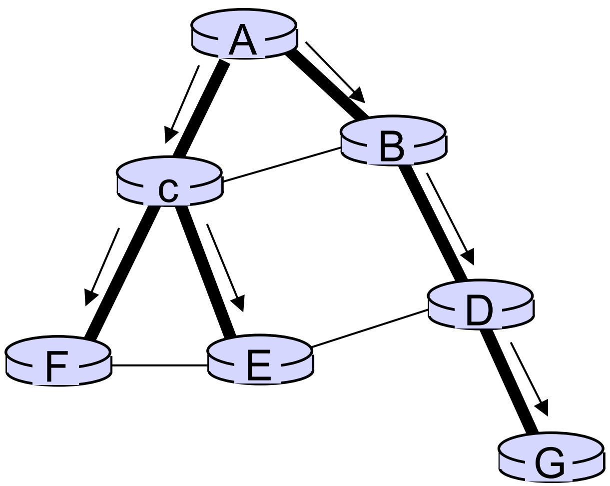
  Keep a subgraph S of G, initially empty
  For each edge e in sorted order

  If the endpoints of e are disconnected in S then add e to S

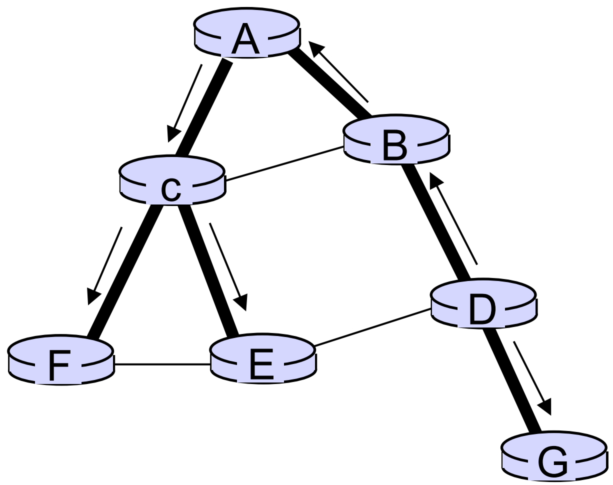
  Return S

Spanning tree

- first construct a spanning tree
- nodes then forward/make copies only along spanning tree



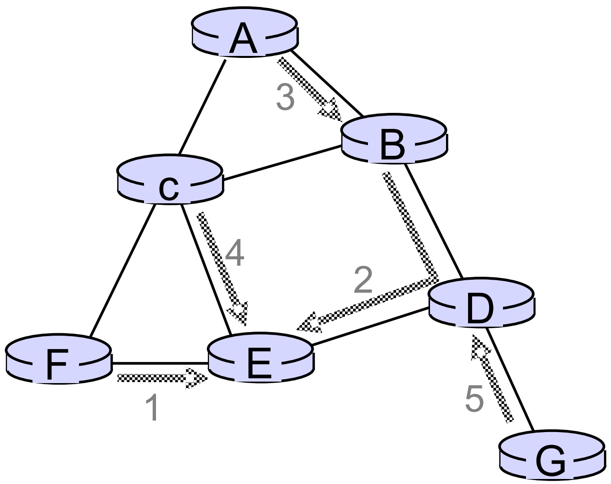
(a) broadcast initiated at A



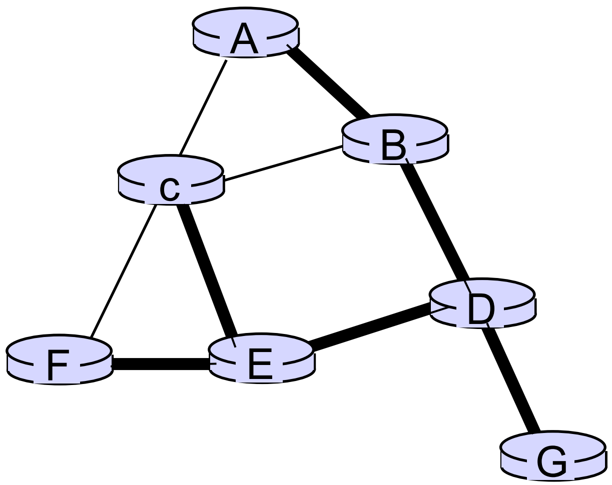
(b) broadcast initiated at D

Spanning tree: creation

- center node
- each node sends unicast join message to center node
  - message forwarded until it arrives at a node already belonging to spanning tree



(a) stepwise construction of spanning tree (center: E)



(b) constructed spanning tree

Multicasting

- Molte applicazioni richiedono il trasferimento di pacchetti da uno o più mittenti ad un gruppo di destinatari
  - trasferimento di un aggiornamento SW su un gruppo di macchine
  - streaming (audio/video) ad un gruppo di utenti o studenti
  - applicazioni con dati condivisi (lavagna elettronica condivisa da più utenti)
  - aggiornamento di dati (adnamento di borsa)
  - giochi multi-player interattivi
  - …

Indirizzamento Multicast

- L'identificatore che rappresenta un gruppo multicast è un indirizzo IP multicast di classe D
- Come ci si affilia ad un indirizzo multicast? Come vengono gestiti i cambiamenti dinamici (join/ remove) nel gruppo?
  - Gestione dinamica del gruppo OLTRE a
  - Algoritmi per la consegna delle informazioni ad un gruppo multicast

IGMP Internet Group Management Protocol

- Messaggi incapsulati in datagrammi IP, con IP protocol number 2
- Mandati con TTL a 1
- Messaggi IGMP
  - Type (8bit) Query (richiesta dal router)/ Membership Report (risposta dagli host)/ Leave group (ma anche possible timeout + mancata risposta alla richiesta del router→ soft state)
- Max Response Time (per rispondere ad una query)
- Checksum
- Group Address (0 se si manda una general query, indirizzo IP del gruppo nel caso di una group specific query con cui si richiede chi sia affiliato a quel gruppo)

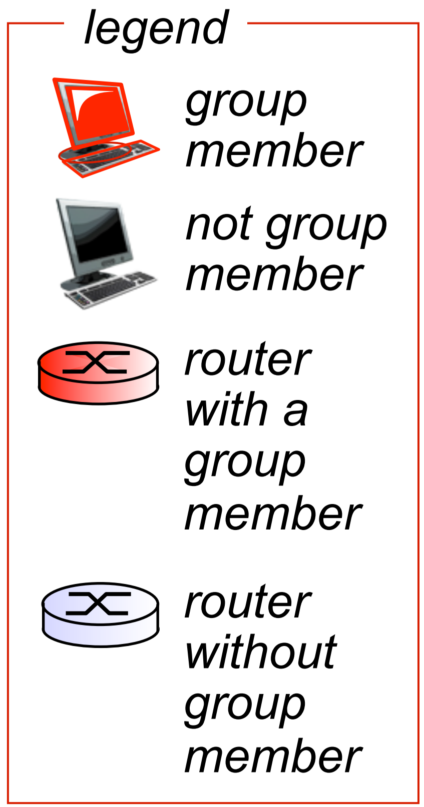
IGMP Internet Group Management Protocol

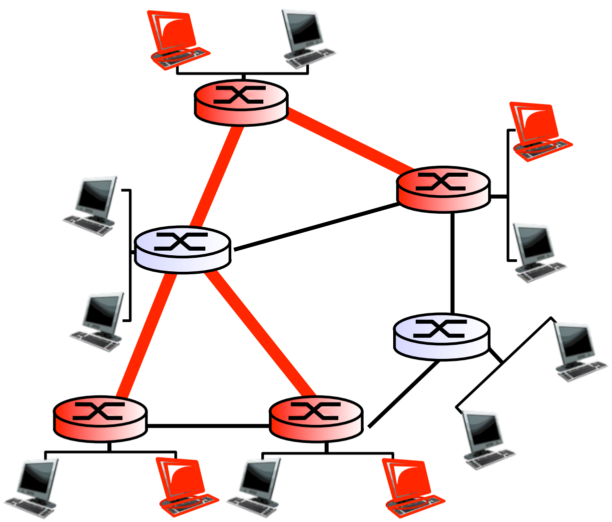
- IGMP consente ad un router di imparare quali gruppi multicast hanno affiliati sulle sottoreti connesse a ciascuna delle loro interfacce
- Un router multicast tiene una lista per ciascuna sottorete dei multicast group (multicast group membership→ almeno un elemento del gruppo fa parte della sottorete) con un timer per membership
  - la membership deve essere aggiornata da report inviati prima della scadenza del timer
  - può essere anche aggiornata tramite messaggi di leave espliciti

Multicast routing: problem statement

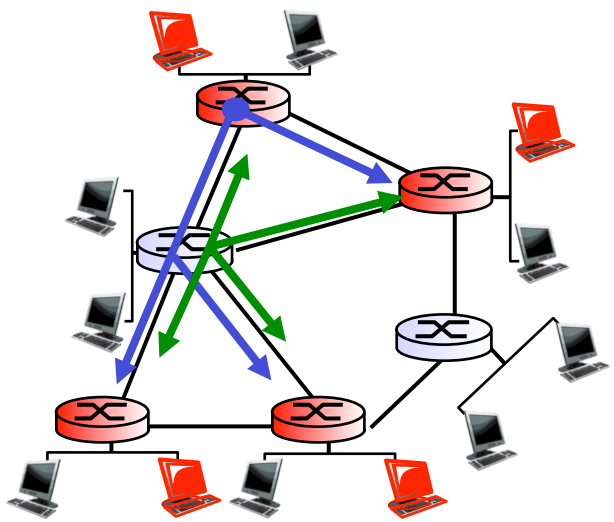
goal: find a tree (or trees) connecting routers having local mcast group members

- *tree:* not all paths between routers used
- *shared-tree:* same tree used by all group members
- *source-based:* different tree from each sender to rcvrs





shared tree



source-based trees

Approaches for building mcast trees

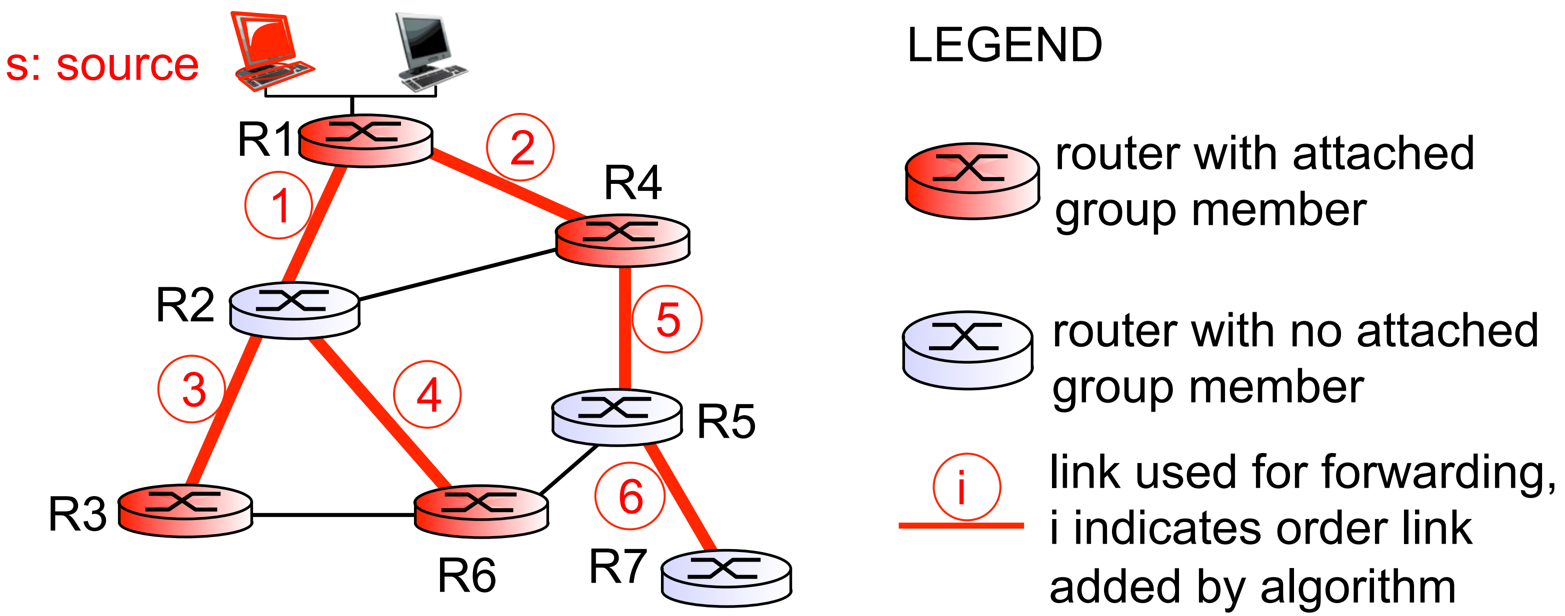
approaches:

- *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

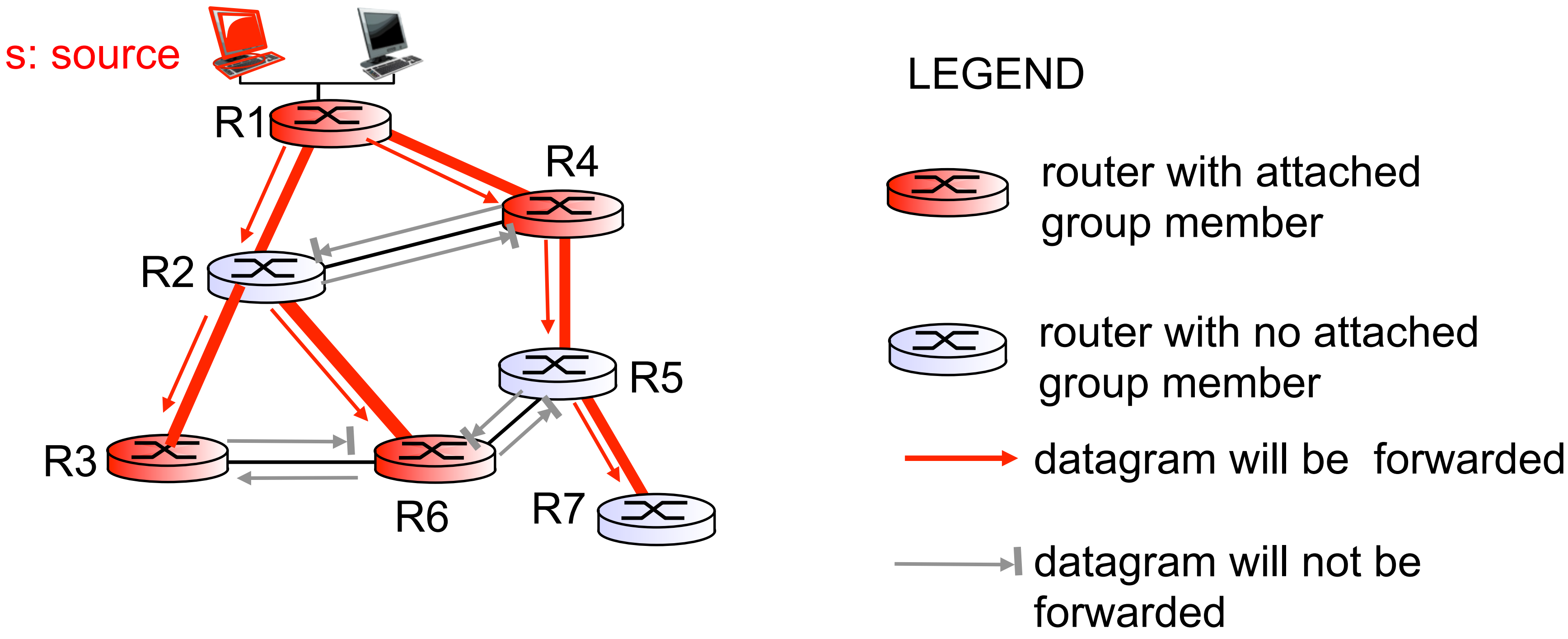


Reverse path forwarding

- 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

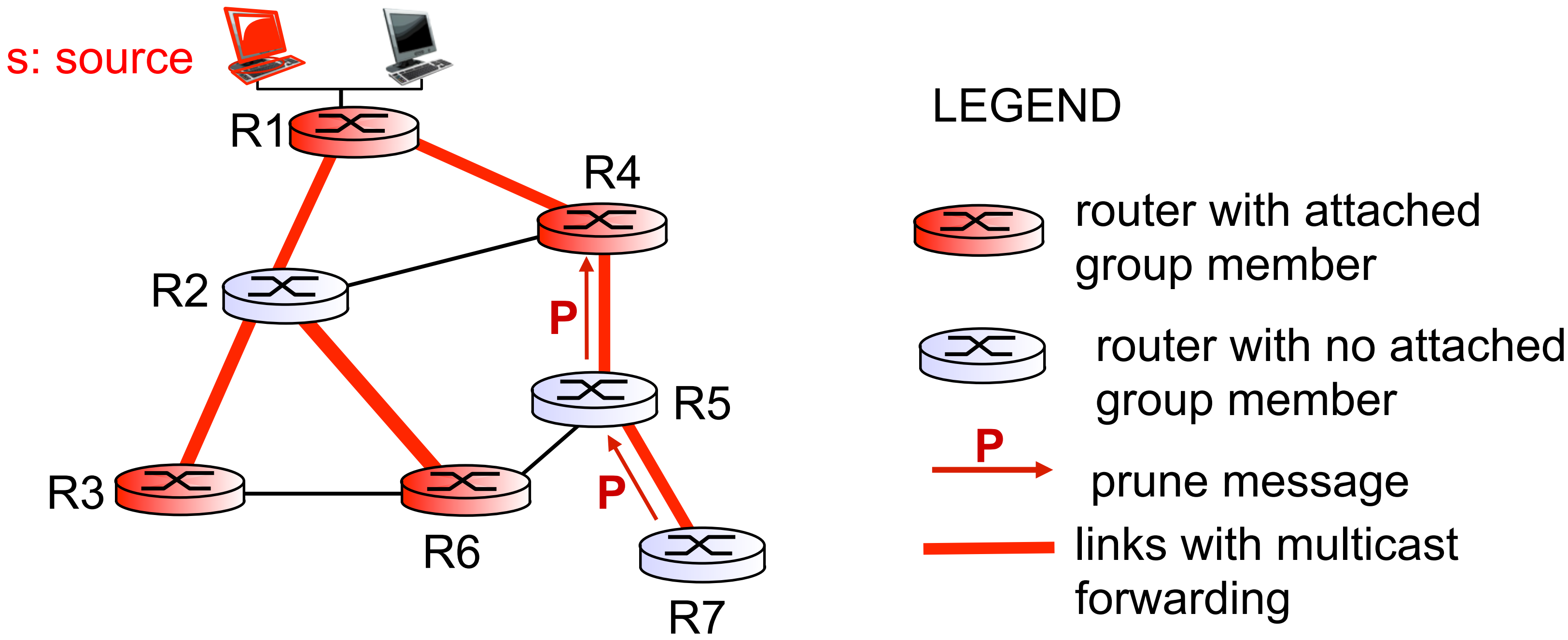
Reverse path forwarding: example



- 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

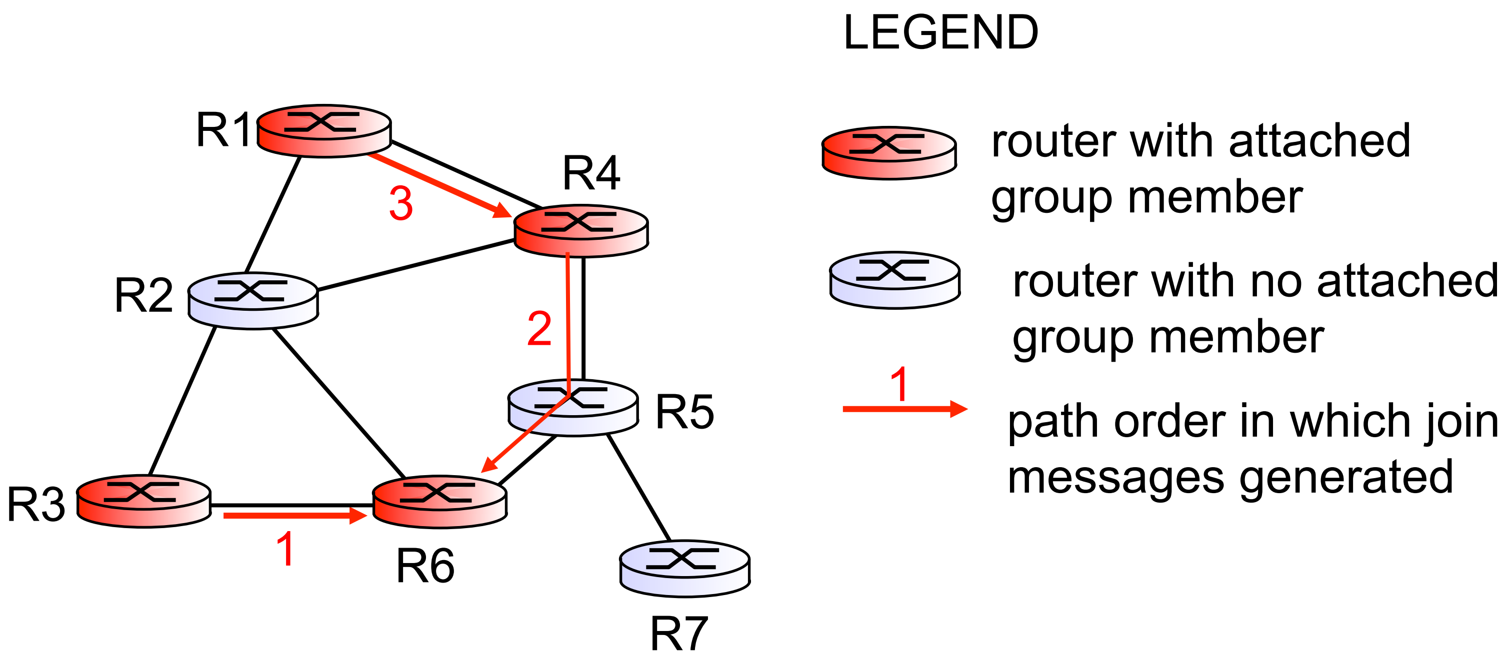
- *steiner tree:* minimum cost tree connecting all routers with attached group members
- problem is NP-complete
- excellent heuristics exists
- not used in practice:
  - computational complexity
  - information about entire network needed
  - monolithic: rerun whenever a router needs to join/ leave

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: example

suppose R6 chosen as center:



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 router

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*

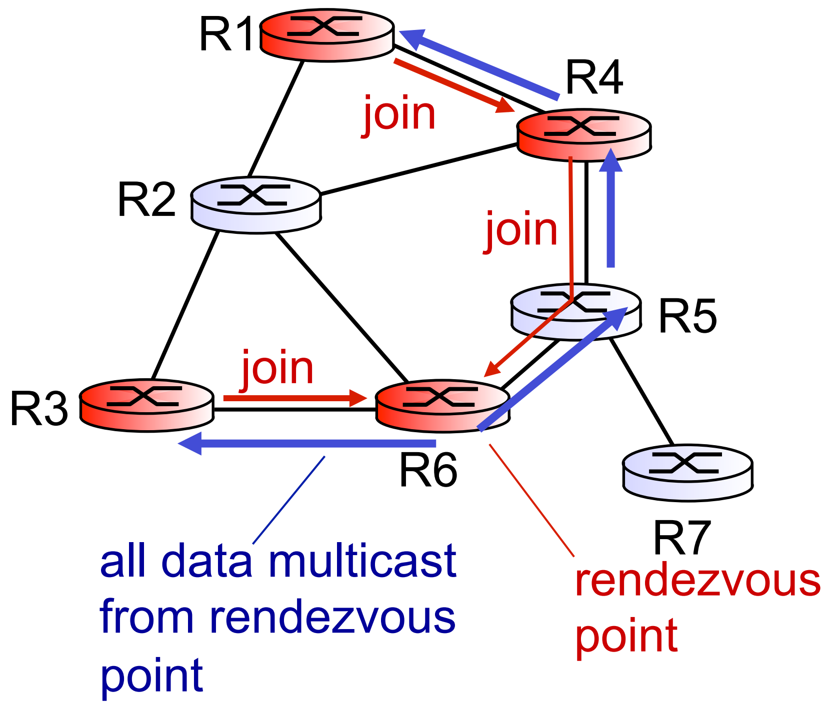
PIM- dense mode

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

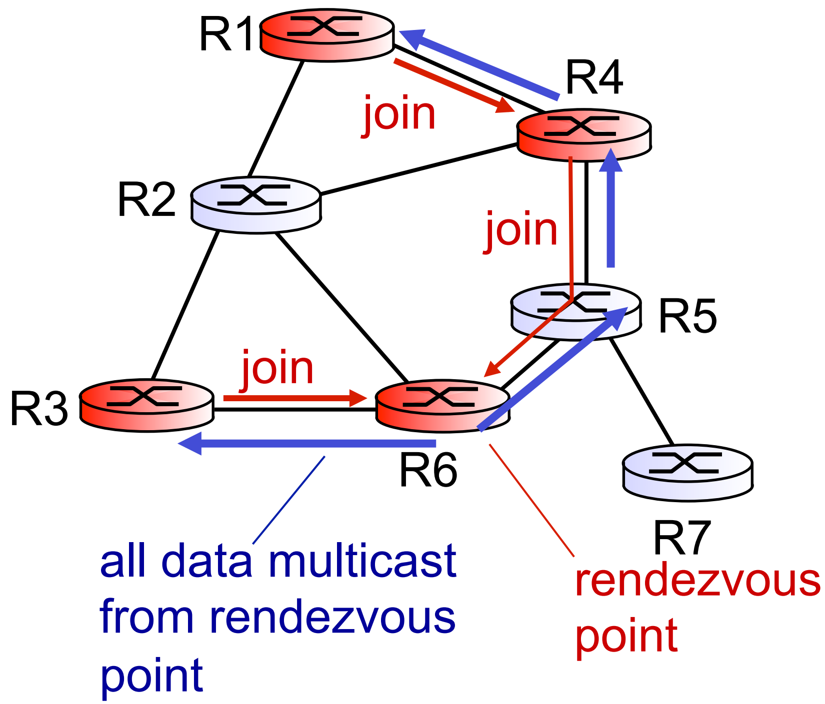
- center-based approach
- router sends *join* msg to rendezvous point (RP)
  - intermediate routers update state and forward *join*
- after joining via RP, router can switch to source-specific tree
  - increased performance: less concentration, shorter paths



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!"



Chapter 4: *done!*

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

4.4 IP: Internet Protocol
- datagram format, IPv4 addressing, ICMP, IPv6

4.5 routing algorithms
- link state, distance vector, hierarchical routing

4.6 routing in the Internet
- RIP, OSPF, BGP

4.7 broadcast and multicast routing

- understand principles behind network layer services:
  - network layer service models, forwarding versus routing how a router works, routing (path selection), broadcast, multicast
- instantiation, implementation in the Internet