Multicast

Overview

➤ multicast applications

➤ multicast membership: IGMP

➤ multicast routing: DVMRP

➤ multicast routing: PIM

➤ the MBONE

Additional references:

- Stephen A. Thomas, *IPng and the TCP/IP protocols*, Wiley, 1996.
- Christian Huitema, *Routing in the Internet*, Prentice Hall, 1995.

Partially drawn from http://www-scf.usc.edu/˜dbyrne/960223.txt (D. Estrin)

Broadcast and multicast

broadcast: all hosts on (small, local) network

directed broadcast: all hosts on remote network

multicast: multiple recipients (group)

Applications for Multicast

- audio-video distribution (1-to-many) and symmetric (all-to-all)
- distributed simulation (war gaming, multi-player Doom, . . .)
- resource discovery (where's the next time server?)
- file distribution (stock market quotes, new software, . . .)
- network news (Usenet)

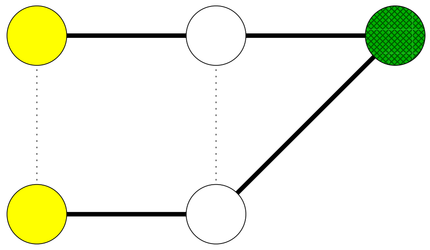
Multicast trees

spanning tree $\equiv$ tree that connects all the vertices (hosts/routers)

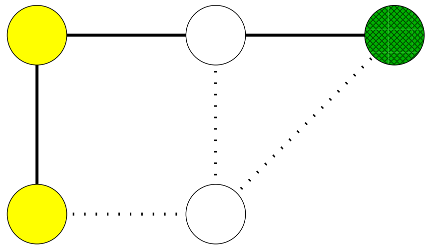
shared tree: single tree for *all* sources S

- minimum-cost spanning (MST) tree (where cost = hops, delay, \$, ...)
- does not minimize length of S to individual destination
- all traffic concentrated on tree ➠ reservation failures

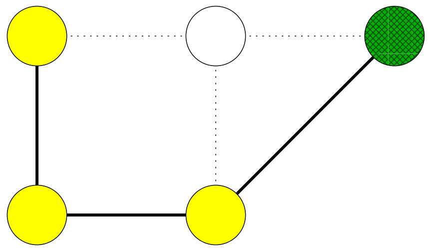
per-source tree: build independently for each source ➠ many variations!



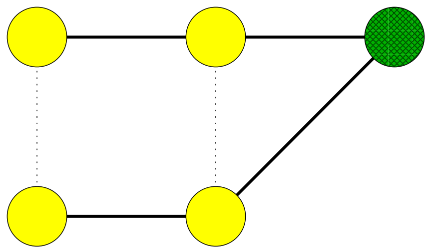
Steiner tree: minimize the total number of links for all sinks



N-P complete (travelling salesman), unstable: small additions → large changes in traffic flows



Add one node:



Problems:

- construct trees one node at a time

- may be
  - periodic group membership flooding ➠ link state
    choose common root for tree and compute individually ➠ MOSPF
  - backward-learning ➠ distance-vector: send membership reports towards S

Connection-oriented multicast

- enumerate sources explicitly ➠ source-based trees
- examples:
  - ATM ➠ explicitly add each end point
  - ATM: end nodes attach themselves to tree
  - enumeration of end points in packet
- only connection-oriented (packet header size!)
- source needs to know destinations $\leftrightarrow$ resource discovery, dynamic groups difficult
- but: natural transition from unicast to multicast

Host group model

Deering, 1991:

- senders need not be members, anyone may send to any group at any time;
- groups may have any number of members;
- there are no topological restrictions on group membership;
- membership is dynamic and autonomous;
- host groups may be transient or permanent.

Local multicast

Some local networks are by nature multi/broadcast: Ethernet, Token Ring, FDDI, . . .

Ethernet, Tokenring:

- broadcast: all ones
- multicast: 01.xx.xx.xx.xx.xx
- adapter hardware can filter dynamic list of addresses

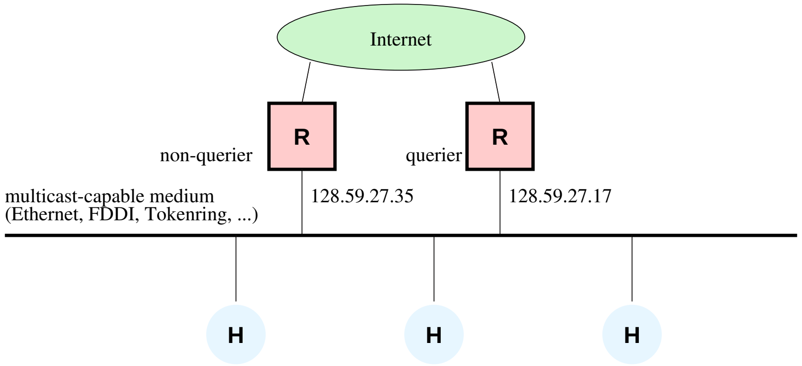
ATM: point-to-multipoint links ➠ need ATM multicast server (MARS)

IP multicast

- host-group model
- network-level; data packets same, only address changes
- need help of routers
- special IP addresses (class D): 224.0.0.0 through 239.255.255.255
- 28 bits ➠ 268 million groups (plus scope)
- 224.0.0.x: local network only ➠ 224.0.0.1: all hosts; 224.0.0.2: all routers
- some pre-assigned (224.0.1.2: SGI Dogfight)
- others dynamic (224.2.x.x for multimedia conferencing)
- map into Ethernet: 01.00.5E.00.00.00 + lower 23 bits
- ttl value limits distribution: 0=host, 1=network

IGMP

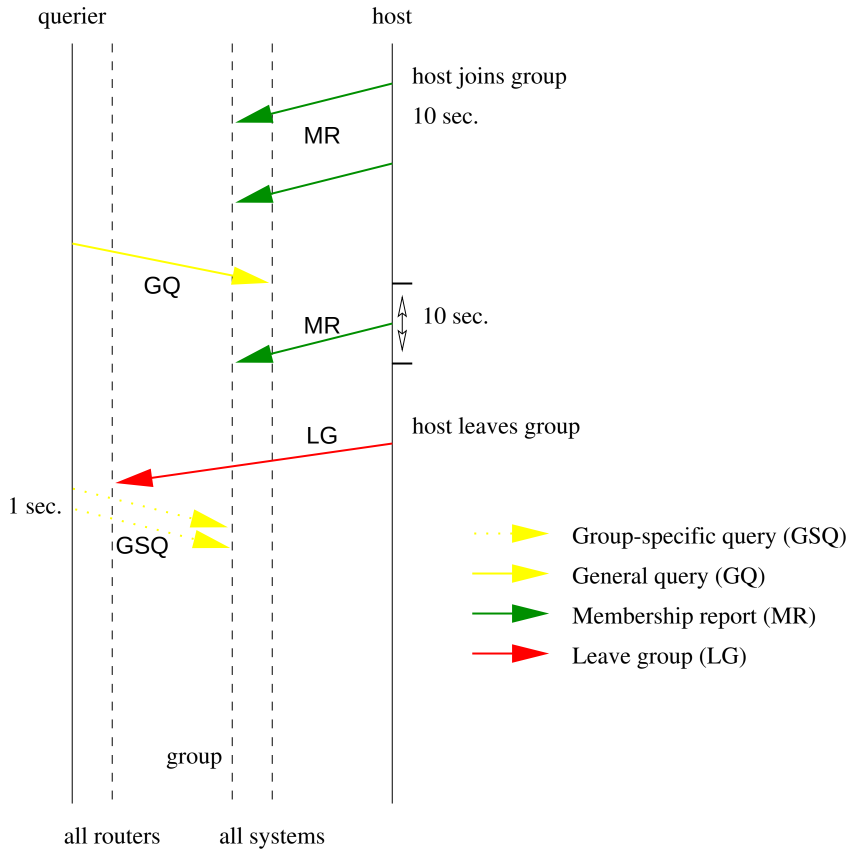
Multicast for local (*broadcast*) networks, between router and hosts



- router listens to all multicast packets on all interfaces
- hosts sends IGMP report for first process to join group to that multicast group (ttl=1), maybe repeat
- router multicasts query to all hosts (224.0.0.2) ≈ every 125 seconds or on start-up
- host waits and listens for others; if nobody else, send response for groups it's in

- if “responsible” for group, notify “all router” group ➠ querier sends group-specific query ➠ reduce bandwidth consumption
- random interval determined by router ($<$ 10 seconds)
- really appropriate for today’s switched Ethernet?

IGMPv2 timing



IGMPv2 packet

0	4	8	12–16	16–32
version (2)	IGMP type (1,6,7)	response time		16-bit checksum
32-bit group address (class D IP address)				

8 bytes

```
$ netstat -g
Group Memberships
Interface Group                RefCnt
--------- -------------------- ------
lo0       ALL-SYSTEMS.MCAST.NET     1
le0       224.2.127.255             1
le0       ALL-SYSTEMS.MCAST.NET     1
```

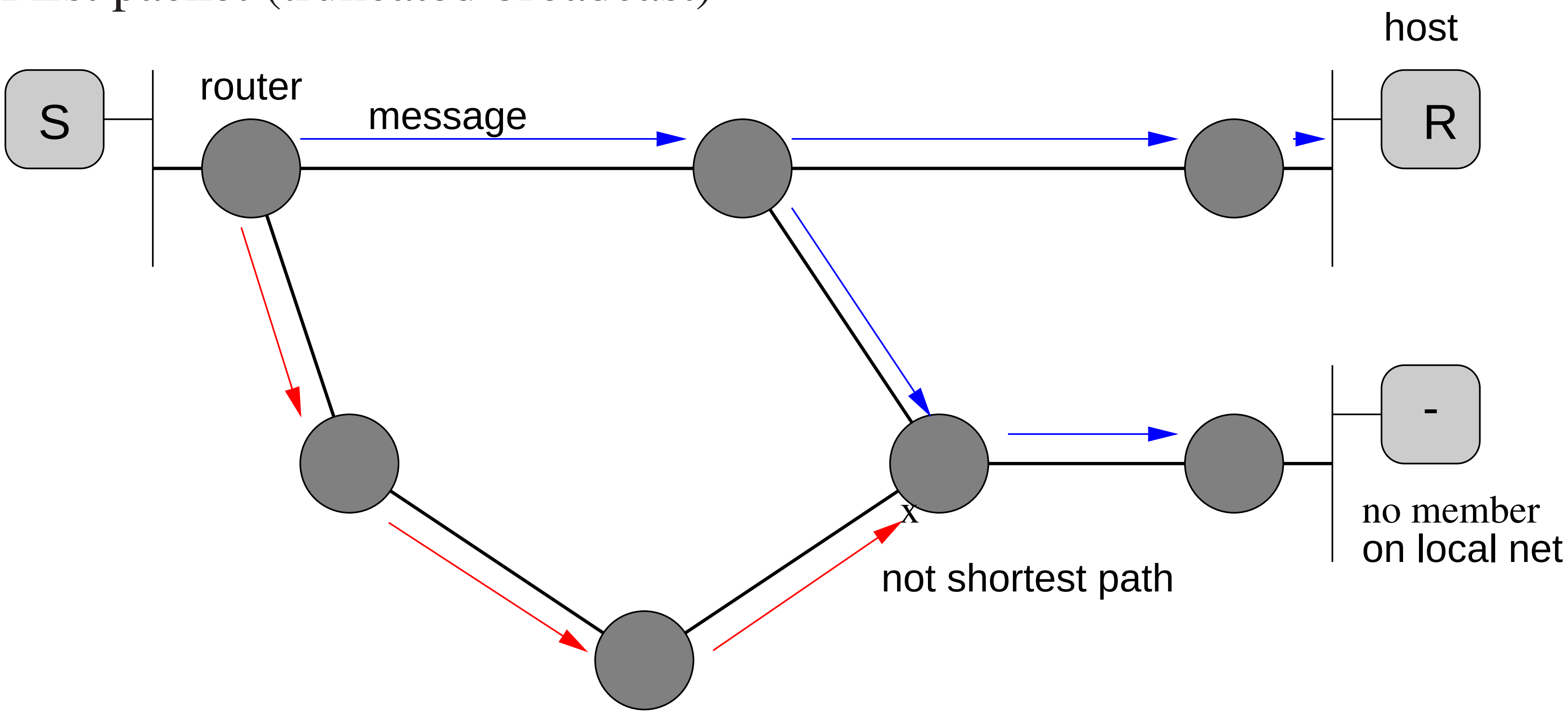
Reverse path flooding

iif: incoming interface; oif: outgoing interface

- if iif is on shortest path to source S
- forward to all other oifs (*RPF check*) towards receivers R in group G
- avoids forwarding duplicates
- result is a source based tree

Multicast forwarding

First packet (truncated broadcast)



Reverse path broadcasting

- do RPF check as before
- exchange unicast routing info to establish “parentage”
- restrict oifs to child nodes

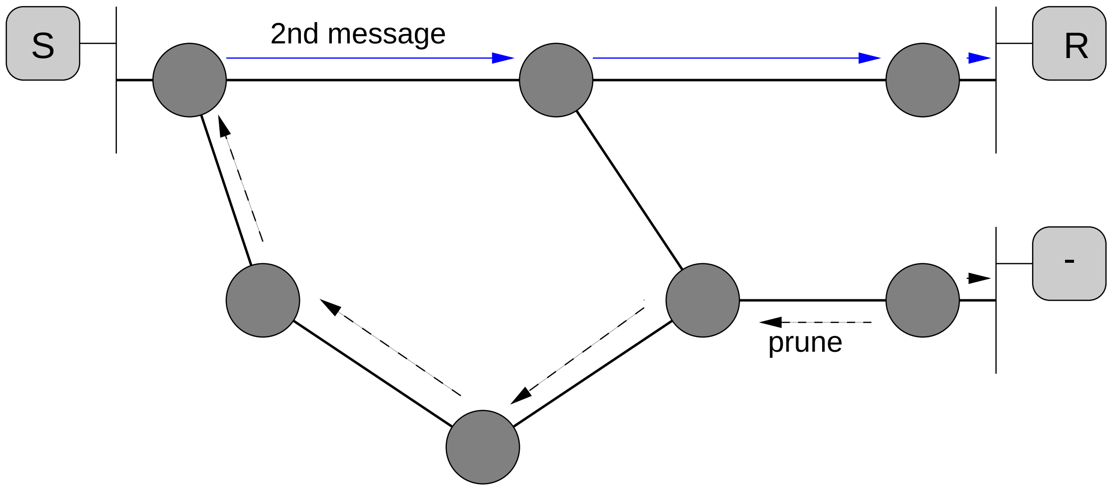
➠ reduce duplicates

Multicast forwarding with truncation

- flood with RPF check
- pruning: leaves of tree send “prune” if no members below
- receivers tell routers of membership
- routers know whether to forward to LAN or prune
- prune state must time out ➠ periodic broadcast
- trade-off: join latency $\leftrightarrow$ bandwidth
- add: explicit “graft” to cancel prune: ➠ join latency $\downarrow$
- still need occasional broadcast for topology changes

Multicast forwarding

With pruning:



➠ router needs to keep “negative” list for groups

Distance Vector Multicast Routing Protocol (DVMRP)

- flood + RPF check
- pruning: time out 1 minute
- routers may send *grafts* upstream
- only send to children
- maintain routing information (DV)
- used in $\rightarrow$ MBone

Multicast Open Shortest Path First (MOSPF)

- link-state based
- include membership info in link-state advertisements
- compute tree for each S, G pair ➠ oifs
- can create shortest-path trees even with asymmetric links
- cannot afford to recompute trees with each LS change

Problems

- “multicast storms”
- MOSPF: broadcast of membership to off-tree areas
- DVMRP: occasional broadcast of packets ➠ bad for WANs
- prune state in routers for sparse groups
- multicast routing vs. unicast routing: reverse path with asymmetric links
- hierarchical routing?
- few “big” senders, lots of background mumbling

➠ compromise on optimal trees

Interdomain multicast routing: CBT

- core-based trees: center-based shared trees with *core*

  $+$**:** no source specific state

  $-$**:** path lengths, traffic concentration

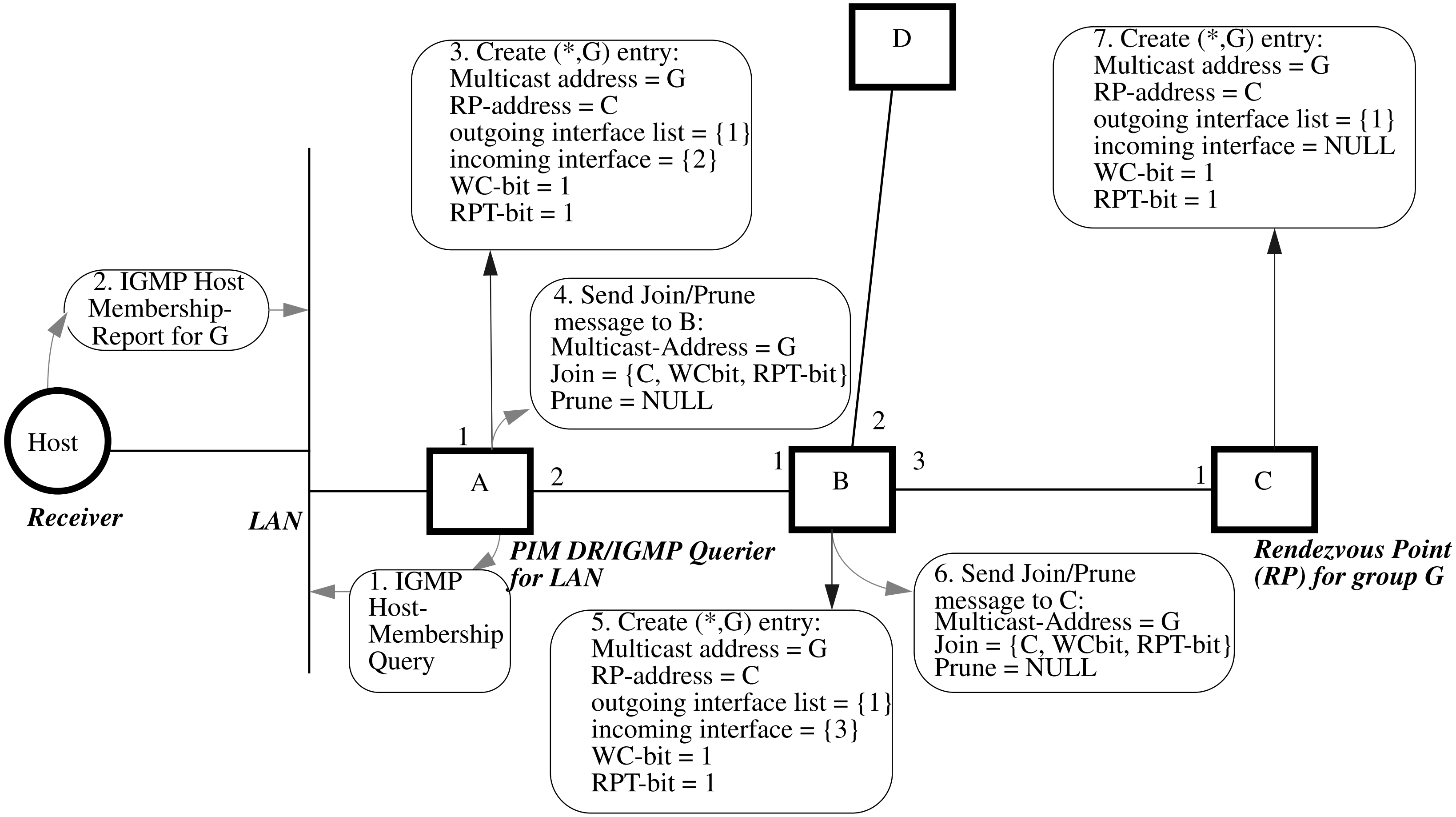
- explicit joining (vs. implicit join and explicit prune)
  join messages from R's router to root of tree

Protocol Independent Multicast (PIM)

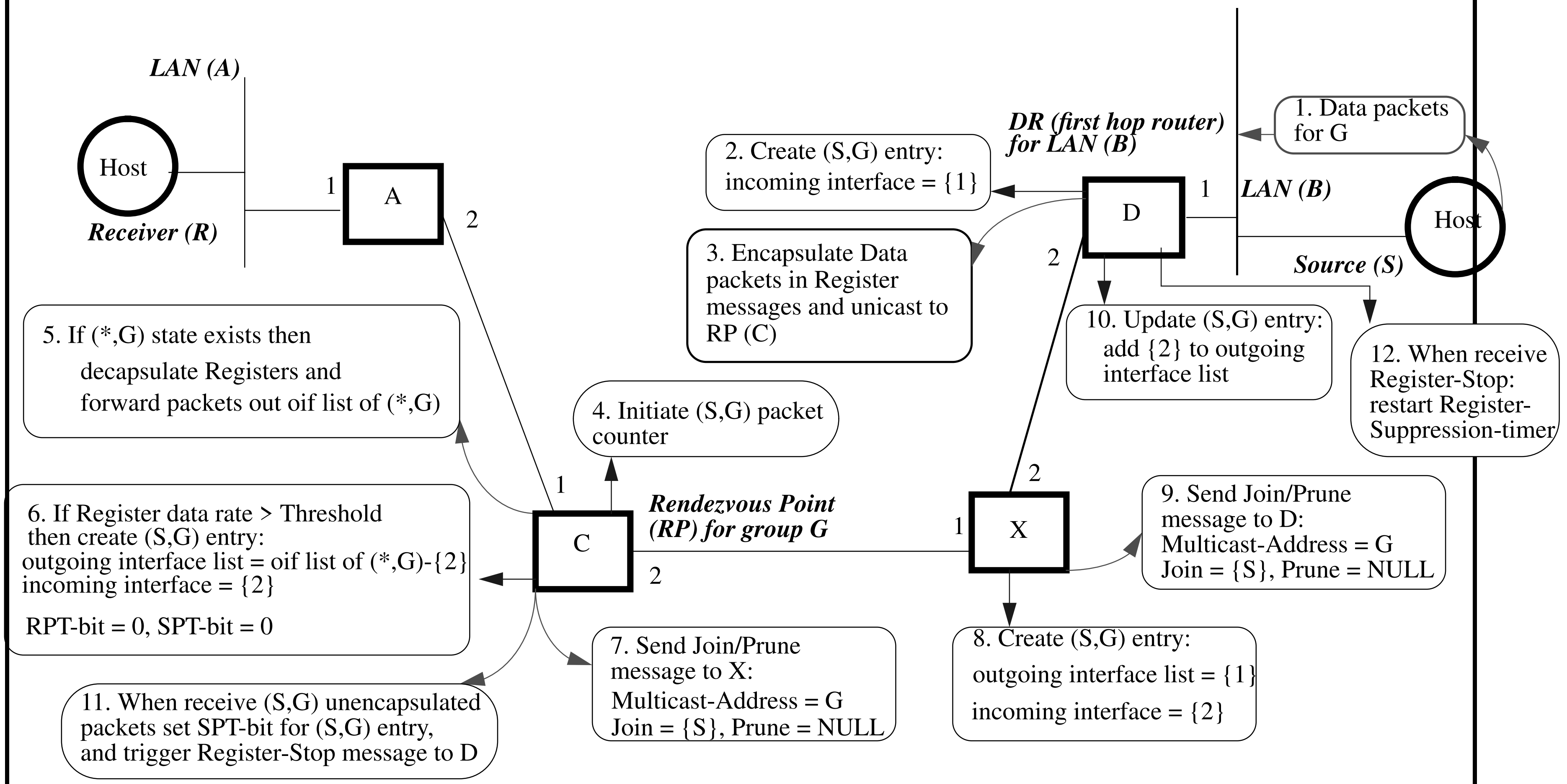
- DVMRP: own routing ➠ PIM: use unicast routing
- "generalization" of CBT: core ➠ rendezvous point (RP)
- Rs and Ss need to know group address and RP ➠ hash function
- supports SPTs and shared trees, depending on traffic
- dense mode: most networks are members ➠ $\approx$ DVMRP $-$ routing: only prune and graft
- sparse mode ➠ several modes:
  1. group-specific RP-rooted shared tree
  2. source-based tree

  RP-tree:
  Rs hear of new S
  new R hear from all S

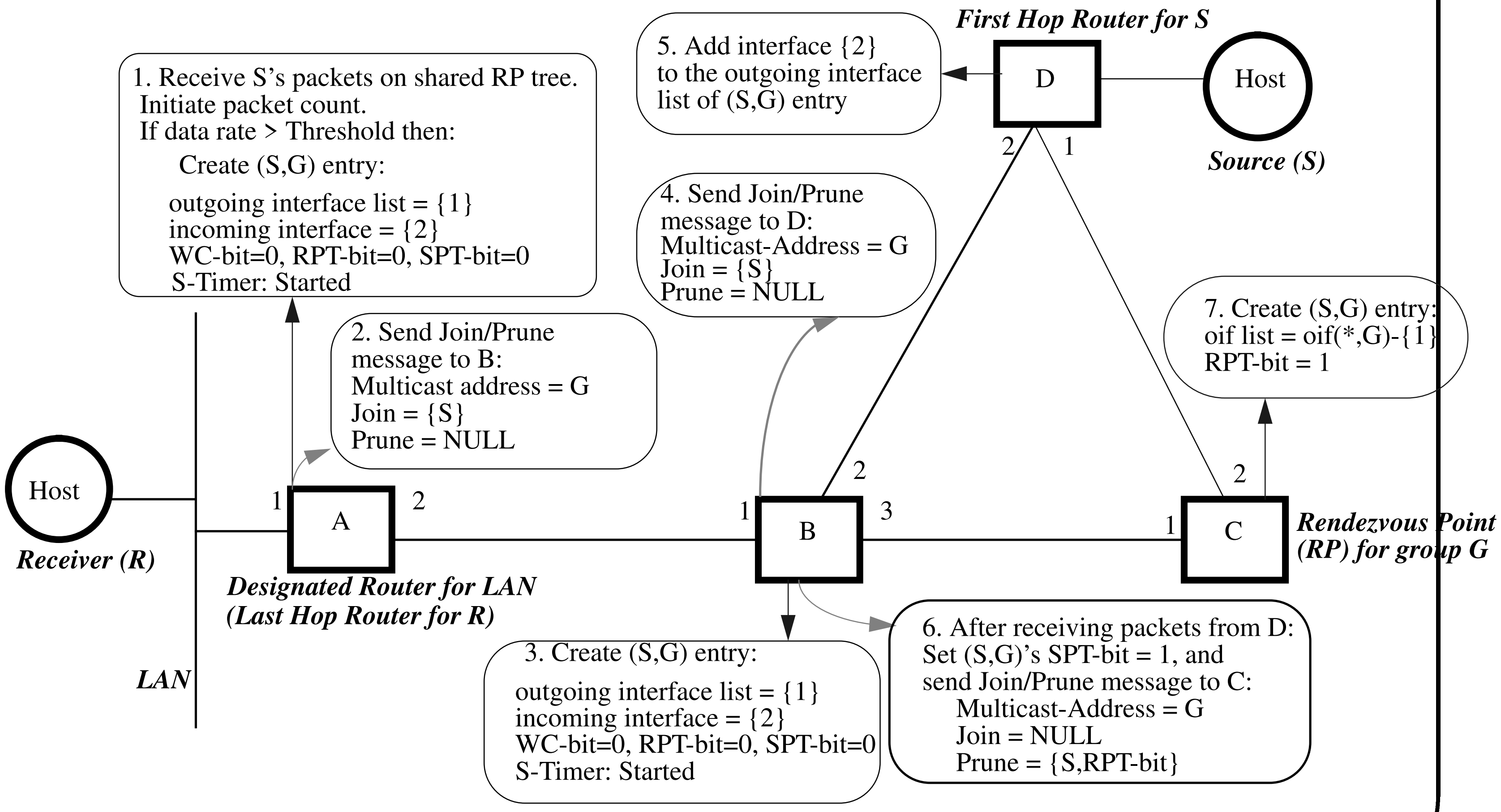
PIM: receiver joins group



PIM: host sends to group

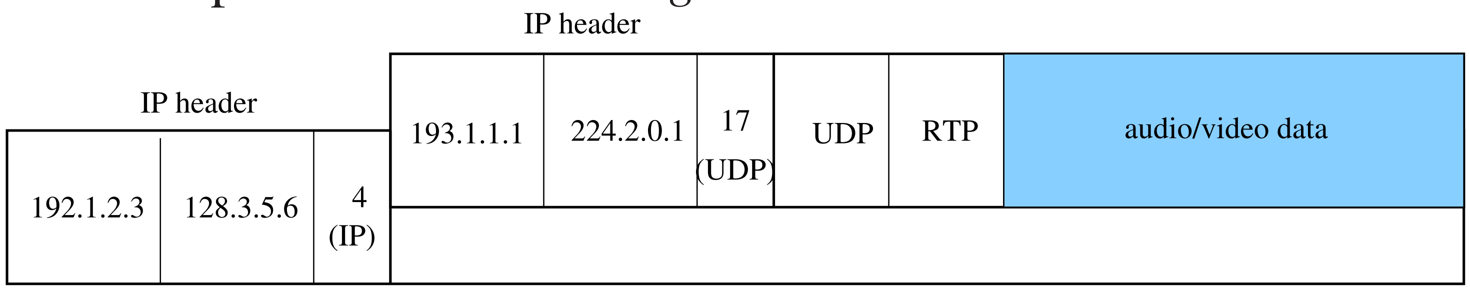


PIM: switching from shared to source tree



MBONE

- MBONE ≡ multicast backbone
- overlay network over Internet
- needed until deployment of multicast-capable backbone routers
- IP-in-IP encapsulation ➠ *tunneling*:



source: 193.1.1.1; group: 224.2.0.1; MBONE tunnel: 192.1.2.3 to 128.3.5.6

- limited capacity, resilience

Multicast: Further Information

- S. Deering, RFC 1112
- D. Comer, *Internetworking with TCP/IP*
- R. Perlman, *Interconnections - Bridges and Routers* (“IP multicast is bad”)
- C. Huitema, *Routing in the Internet*
- PIM: `draft-ietf-idmr-pim-{s,d}m=spec-*.txt`