CS551
Multicast Routing: IGMP

Bill Cheng

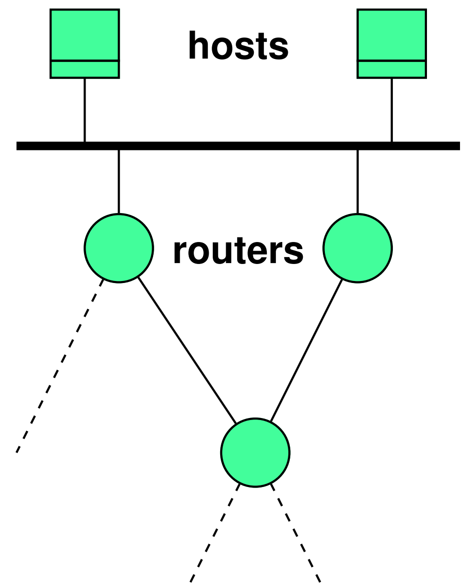
http://merlot.usc.edu/cs551-f12

Components of the IP Multicast Architecture

service model

host-to-router protocol (IGMP)

multicast routing protocols (various)



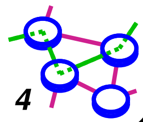
Internet Group Management Protocol (IGMP)

- the protocol by which hosts report their multicast group memberships to neighboring routers
- version 1, the current Internet Standard, is specified in RFC-1112
  - version 2: RFC 2236
- operates over broadcast LANs and point-to-point links
- occupies similar position and role as ICMP in the TCP/IP protocol stack

Link-layer Transmission/reception

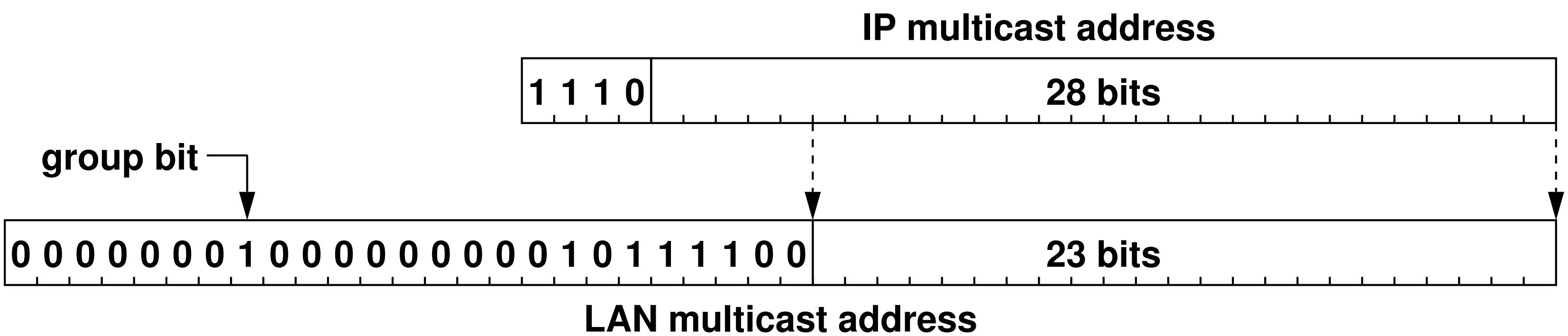
- **Transmission:**
  - an IP multicast packet is transmitted as a link-layer multicast, on those links that support multicast
  - the link-layer destination address is determined by an algorithm specific to the type of link (next slide)

- **Reception:**
  - the necessary steps are taken to receive desired multicasts on a particular link, such as modifying address reception filters on LAN interfaces
  - multicast routers must be able to receive all IP multicasts on a link, without knowing in advance which groups will be sent to



Mapping to Link-layer Multicast Addresses

- for Ethernet and other LANs using 802 addresses:



- for point-to-point links: no mapping needed

IGMP Version 1 Message Format

Vers	Type	Reserved	Checksum
Group Address			

Version : 1

Type : 1 = Membership Query
2 = Membership Report

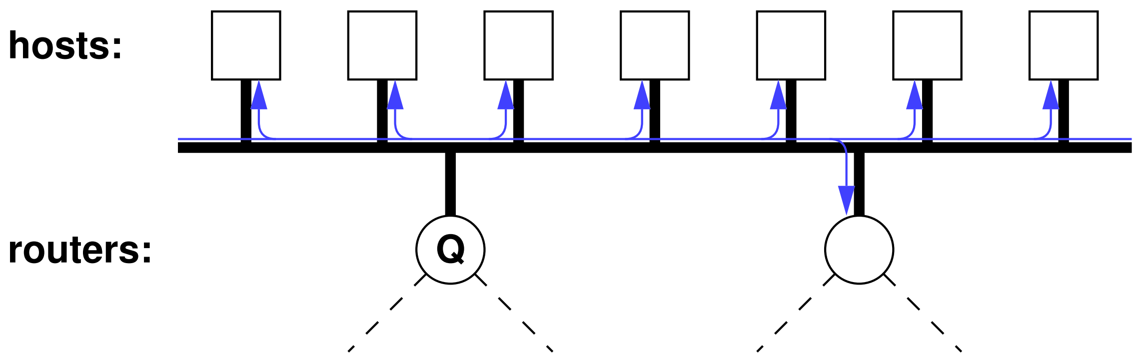
Checksum : standard IP-style checksum of the IGMP Message

Group Address : group being reported (zero in Queries)

IGMP Goal

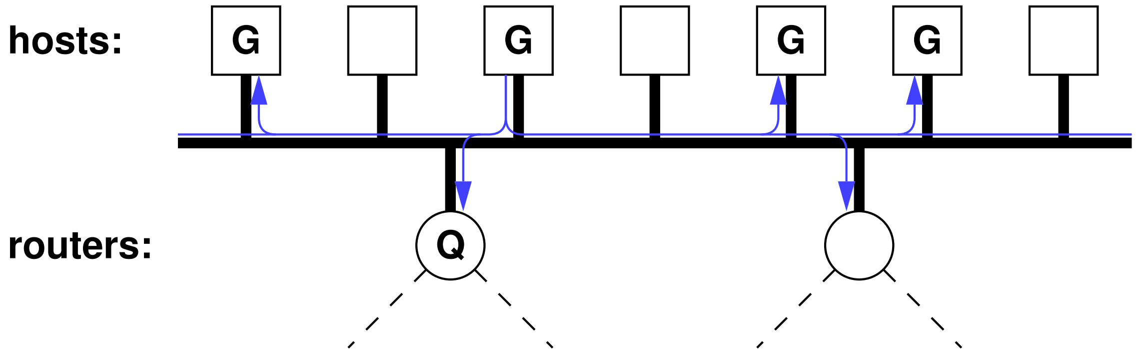
- Determine what IP multicast groups have receivers present on the LAN
  - just care about some vs. zero receivers, not how many
- Approach
  - designate one router as IGMP "querier"
  - it asks all hosts
  - get at least one response per active group
  - example of ***soft state*** (periodically query), so occasional losses are okay

How IGMP Works



- on each link, one router is elected the "querier"
- querier periodically sends a *Membership Query* message to the all-systems group (224.0.0.1), with TTL = 1
- on receipt, hosts start random timers (between 0 and 10 seconds) for each multicast group to which they belong

How IGMP Works (Cont...)



- when a host's timer for group G expires, it sends a *Membership Report* ***to group G***, with TTL = 1
- other members of G hear the report and stop their timers
- routers hear ***all*** reports, and time out nonresponding groups

IGMP Implications

- In normal case, only one report message per group present is sent in response to a query (routers need not know who all the members are, only that members exist)
- Query interval is typically 60 -- 90 seconds
  - IGMPv2 adds explicit leave messages
- To reduce *join latency*, when a host first joins a group, it sends one or two immediate reports (unsolicited responses), instead of waiting for a query

IGMP Version 2

- changes from version 1:
  - new message and procedures to reduce "leave latency"
  - standard querier election method specified
  - version and type fields merged into a single field
- backward-compatible with version 1
- soon to appear as a Proposed Standard RFC
- widely implemented already

CS551
Multicast Routing

Bill Cheng

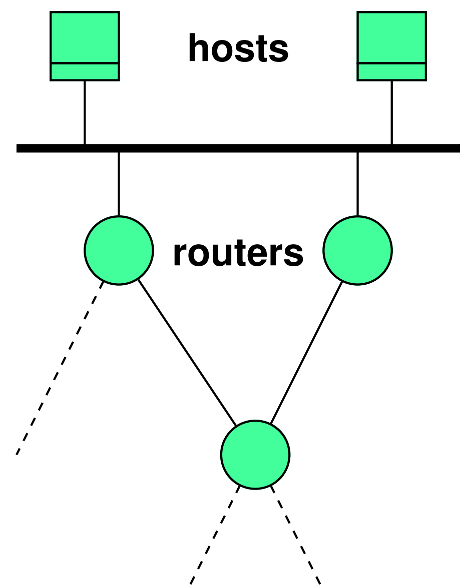
http://merlot.usc.edu/cs551-f12

Components of the IP Multicast Architecture

service model

host-to-router protocol (IGMP)

multicast routing protocols (various)



Multicast Routing

- Multicast service model makes it hard to locate receivers
  - anonymity
  - dynamic join/leave
- Options so far (not very efficient)
  - flood data packets to entire network, or
  - tell routers about all possible groups and receivers so they can create routes (trees)

Early Routing Techniques

- ***Flood and prune***
  - begin by flooding traffic to entire network
  - prune branches with no receivers
  - *unwanted state where there are no receivers*
  - examples: DVMRP, PIM-DM
- **Link-state multicast protocols**
  - routers advertise groups for which they have receivers to entire network
  - compute trees on demand
  - *unwanted state where there are no senders*
  - examples: MOSPF

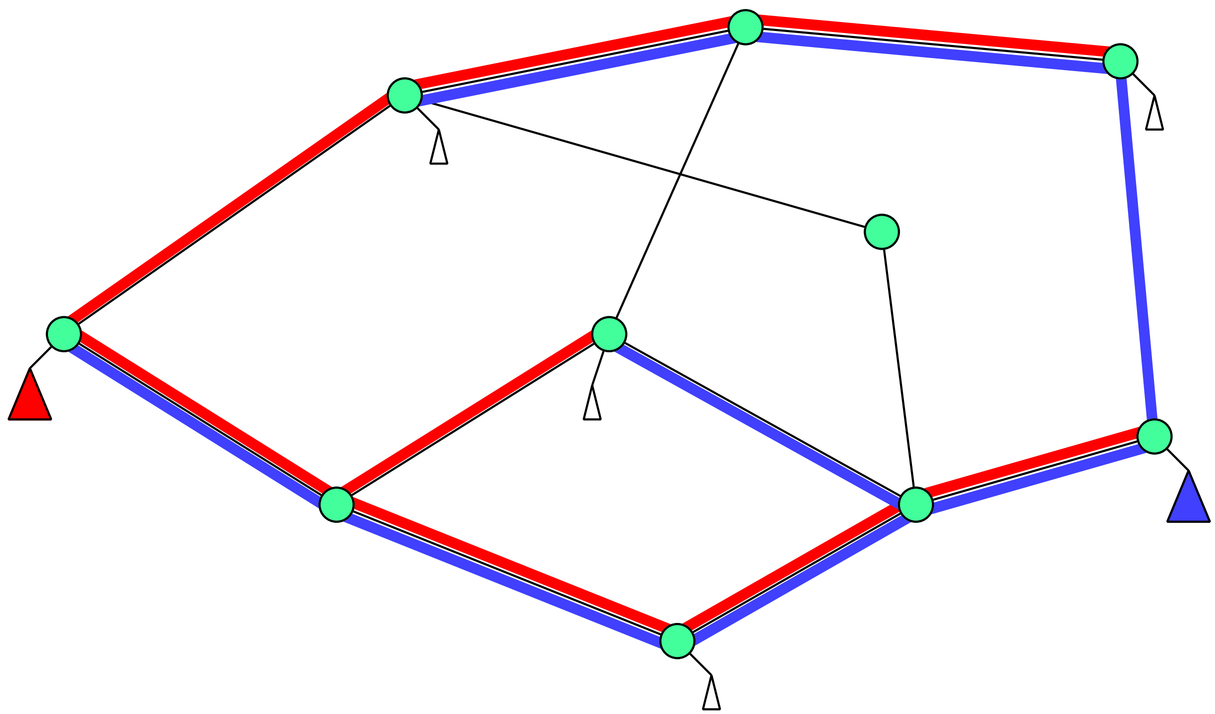
Rendezvous Options

- Specify *rendezvous* (or meeting place) to which sources send initial packets, and receivers join; requires mapping between multicast group address and meeting place
  - examples: CBT, PIM-SM

Multicast Tree Taxonomy

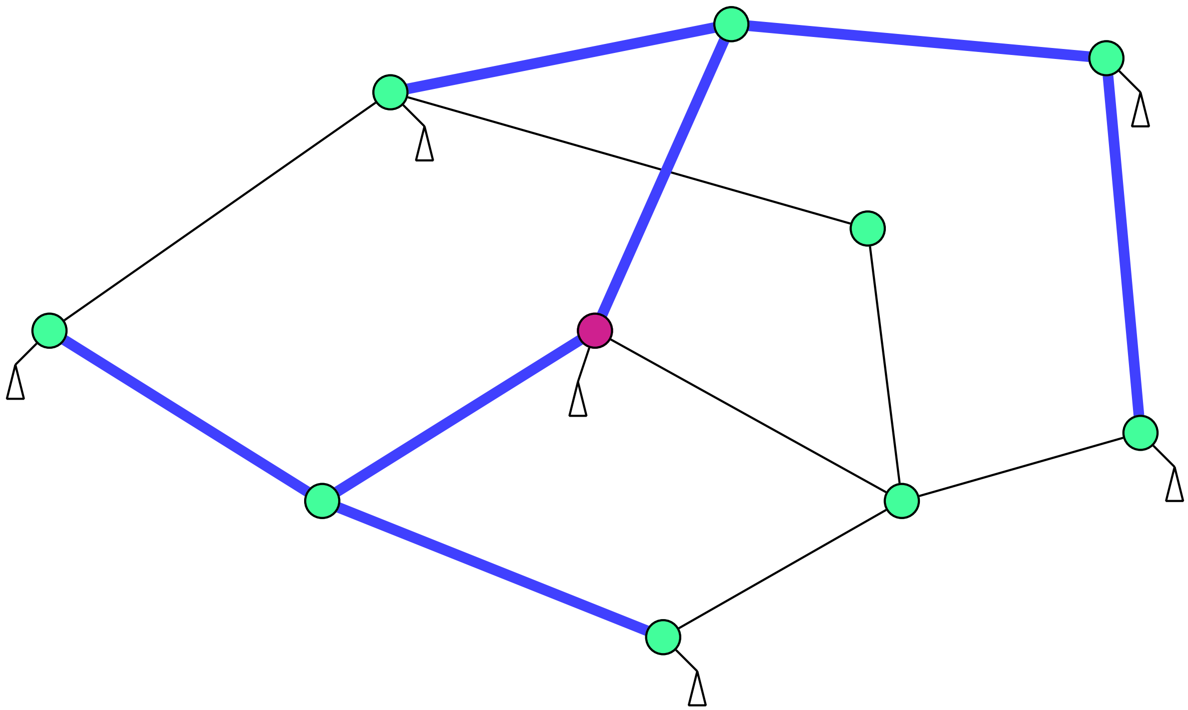
- **Multicast routing can build different types of distribution trees**
- ***Source-based trees***
  - **separate shortest path tree (SPT) *for each sender***
    - **can have multiple senders per group**
  - **examples: DVMRP, MOSPF, PIM-DM, PIM-SM**
- ***Shared trees***
  - **single tree shared by all members**
  - **shared tree rooted at group core/rendezvous point**
  - **examples: CBT, PIM-SM**

Source-based Trees



- output link determined from input link, multicast address, and *source* address

A Shared Tree



- output link determined from input link & multicast address

Shared v.s. Source-Based Trees

- Source-based trees
  - shortest path trees - low delay, better load distribution
  - more state at routers (per-source state)
  - efficient for dense-area multicast
- Shared trees
  - higher delay (bounded by factor of 2), traffic concentration
  - per-group state at routers
  - efficient for sparse-area multicast

Protocol Taxonomy

- **DVMRP - source-based trees**
- **MOSPF - source-based trees**
- **PIM - shared and source-based trees**

Who Can Send?

- Anyone (Deering's service model)
  - model used by most multicast applications
- Single-source
  - only one node can send (others must make their own group)
  - EXPRESS [Holbrook99a]

Multicast Status

- MBone exists
  - moderately widely used in research
  - but not always stable
    - multi-domain routing is hard, need to coordinate people and often people don't talk about experimental services
- Some commercial use (applications)
  - but very little ISP support
    - concerned about how to charge, and potential over-use
- Multicast widely used on LANs
  - e.g., Google, Inktomi use it for load balancing

CS551
DVMRP & MOSPF
[Deering88b]

Bill Cheng

http://merlot.usc.edu/cs551-f12

Key Ideas

- Lays foundation for IP multicast
  - defines IP multicast service model
    - e.g., best effort, packet based, anonymous groups
    - compare to ISIS with explicit group membership, guaranteed ordering (partial or total ordering)
- Several algorithms
  - extended/bridged LANs
  - distance-vector extensions (DVMRP)
  - link-state extensions (MOSPF)
- Cost analysis

Characterizing Groups

- **Pervasive or dense**
  - **most LANs have a receiver**
- **Sparse**
  - **few LANs have receivers**
- **Local**
  - **inside a single adminstrative domain**

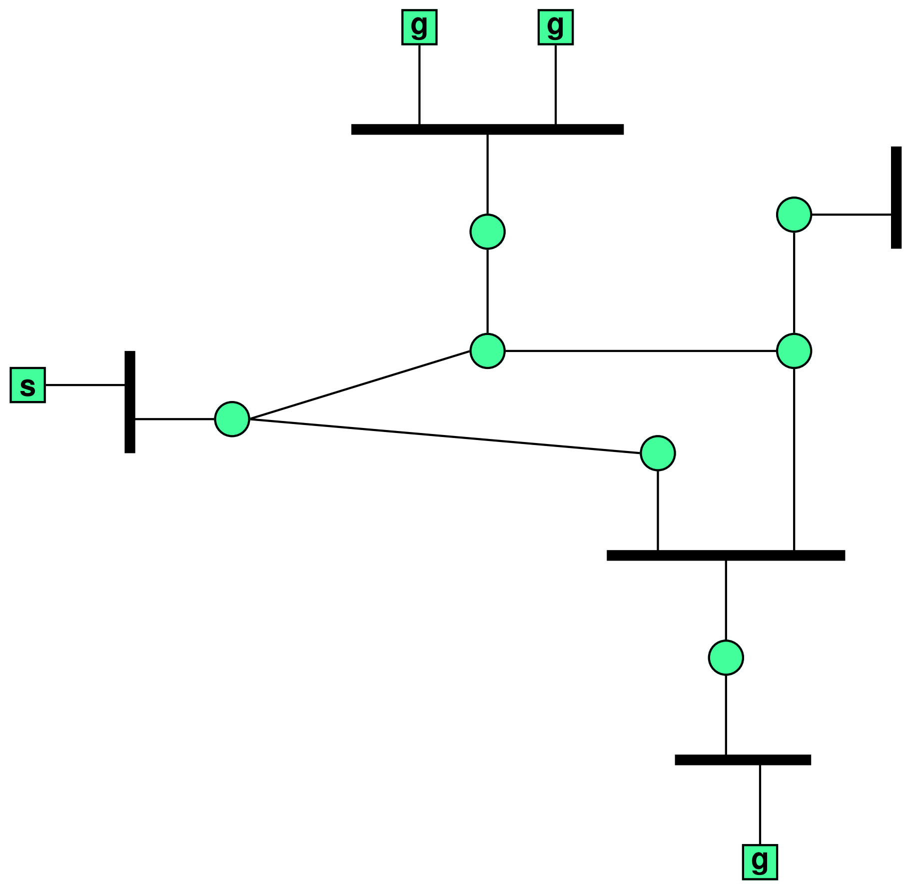
Distance-vector Multicast Routing Protocol (DVMRP)

- Basic idea: *flood and prune*
  - flood: send information about new *sources* everywhere
  - prune: routers will tell us if they don't have receivers
- Routing information is soft state; periodically reflood (and prune) to refresh this information
  - if no refresh, then the information goes away
    - ⇒ easy fault recovery
- DVMRP consists of two major components:
  - a conventional distance-vector routing protocol (like RIP)
  - a protocol for determining how to forward multicast packets, based on the routing table

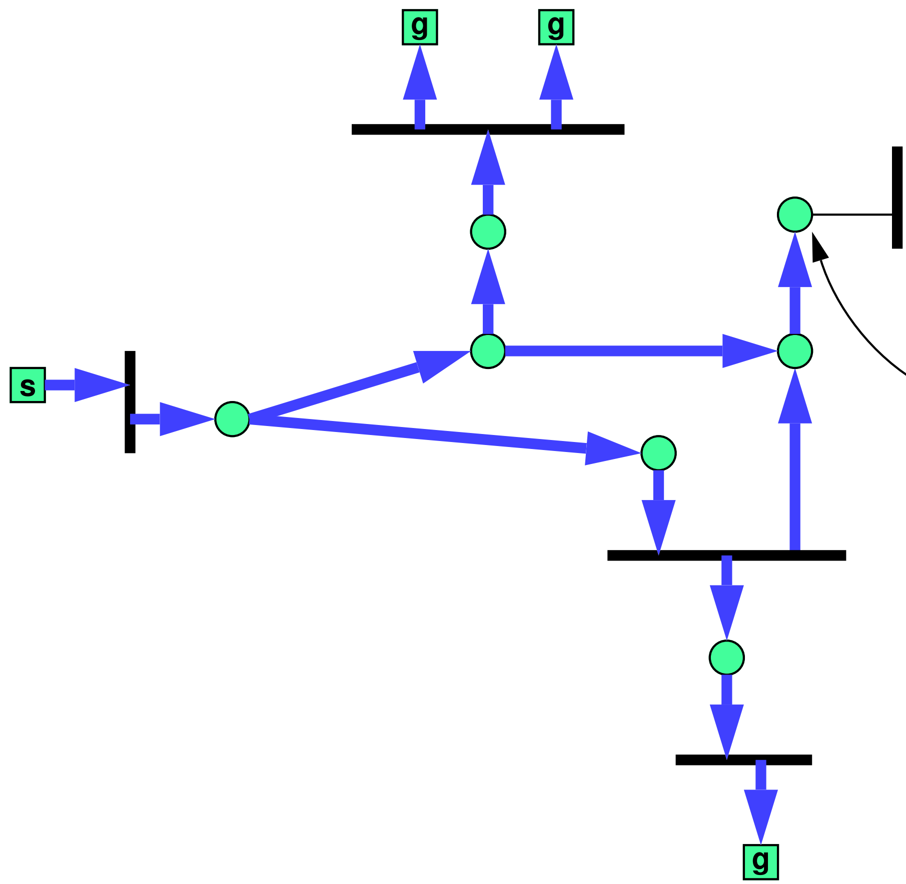
Multicast Forwarding

- A DVMRP router forwards a packet if
  - *Reverse Path Forwarding (RPF)*
    - the packet arrived from the link used to reach the source of the packet (in unicast routing)
    - take advantage of what is available from unicast
  - similar (but not quite the same) to flooding each packet once
    - if downstream links have not pruned the tree

Example Topology

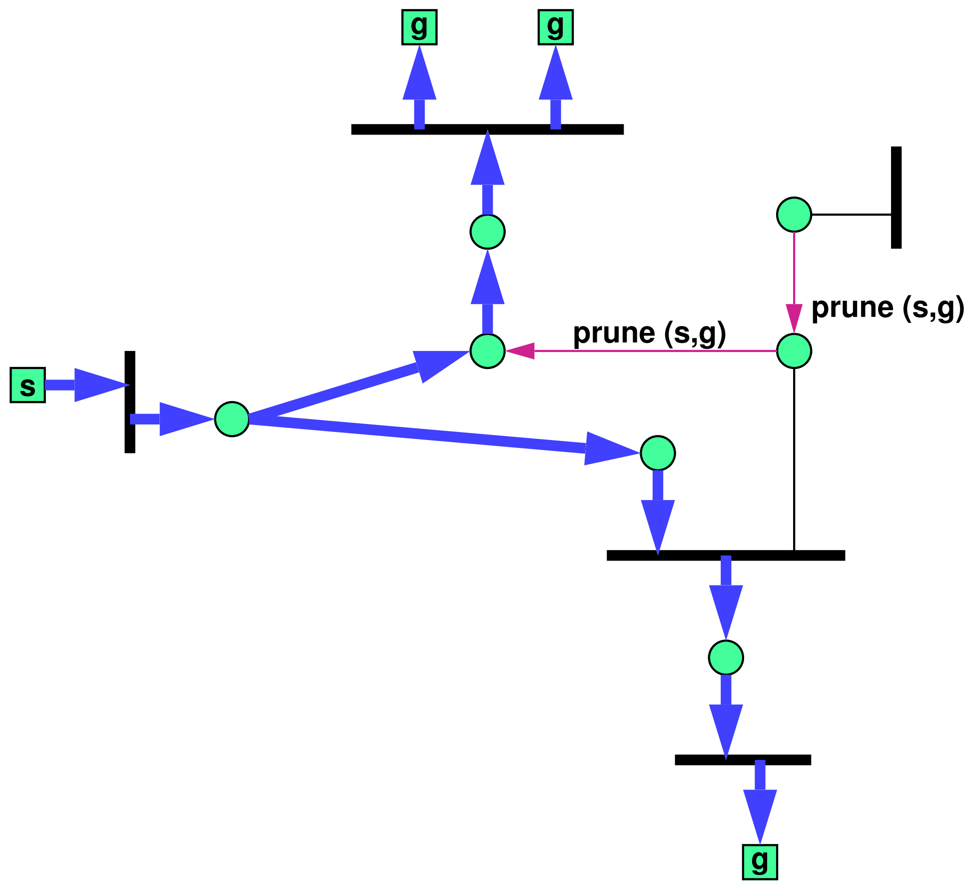


Phase 1: Flood Using Truncated Broadcast

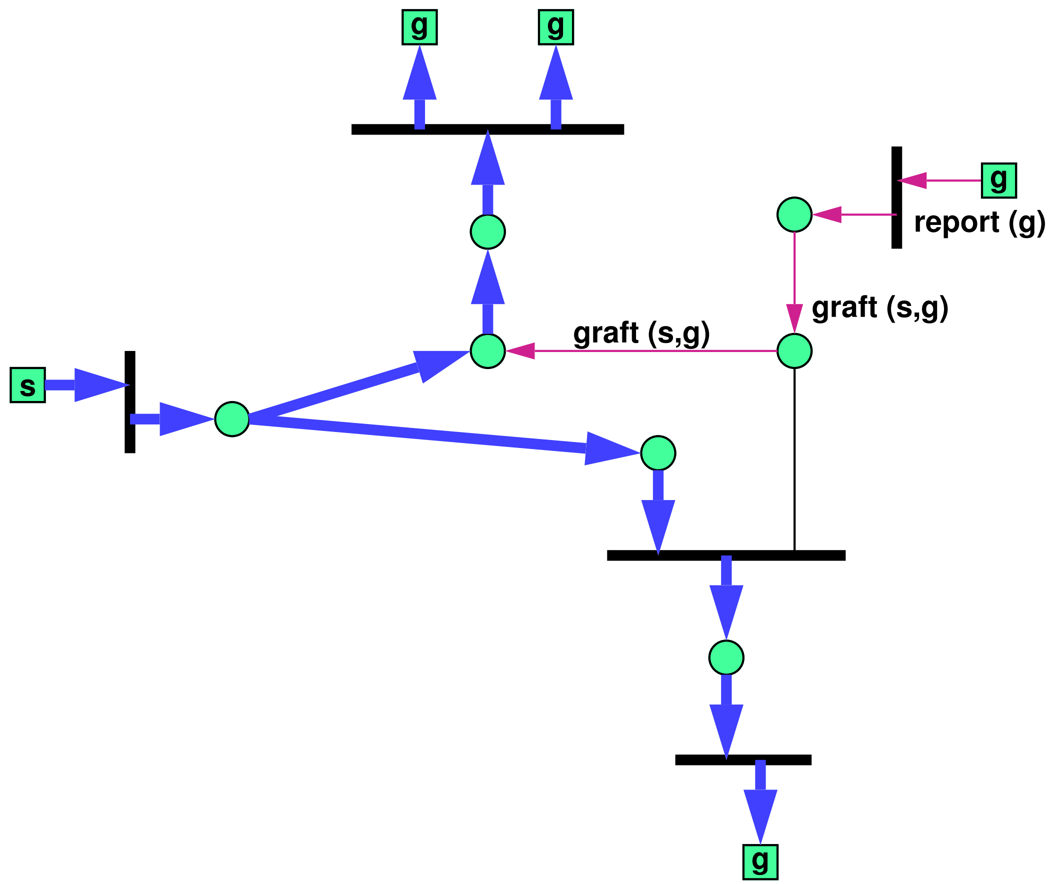


This router knows it has no group members on its LAN, so it does not broadcast over its LAN

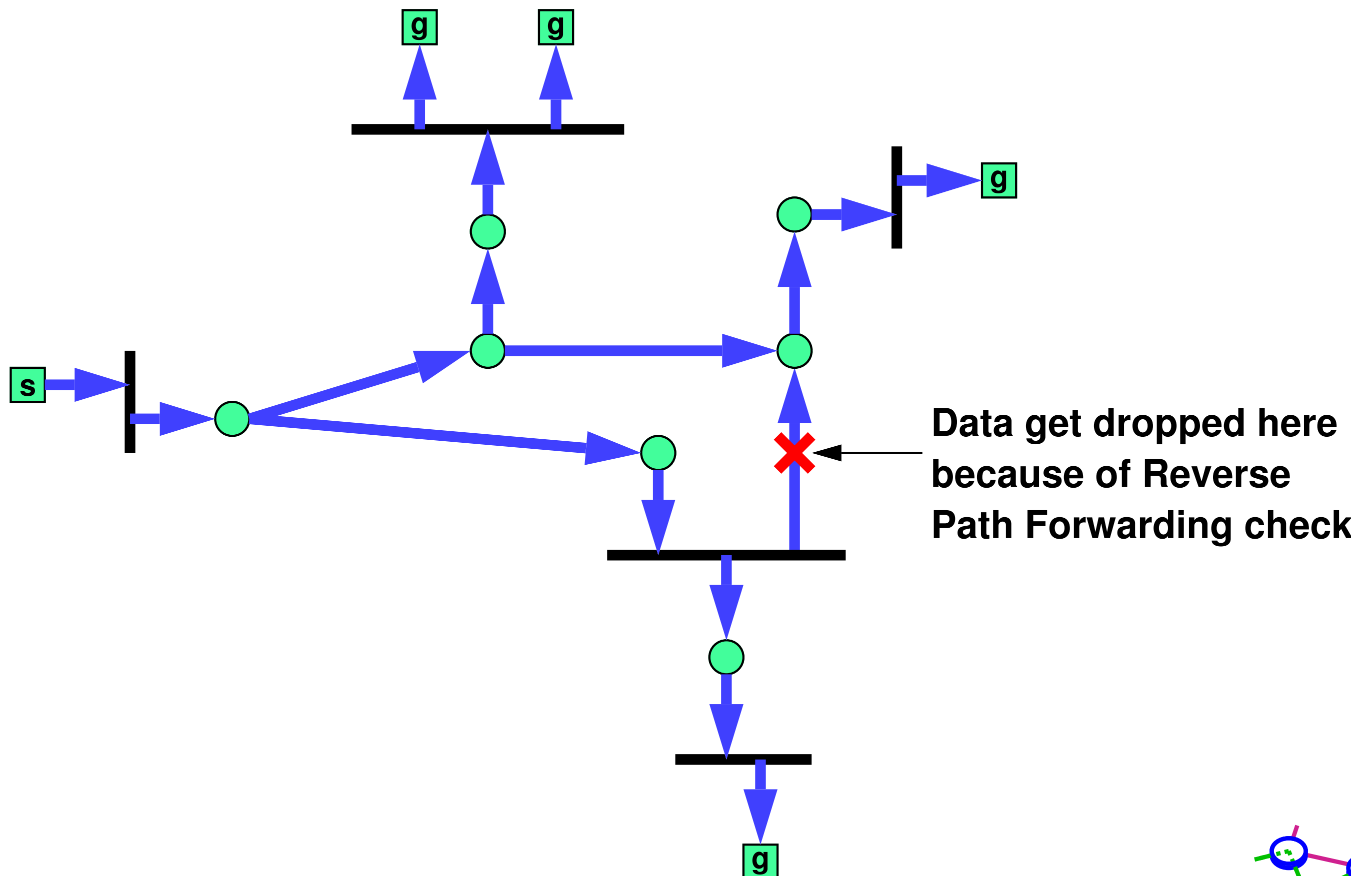
Phase 2: Prune



Phase 3: Graft



Phase 4: Steady State



Sending Data in DVMRP

- Data packets are sent on all branches of the tree
  - send on all interfaces except the one they came in on
- RPF (Reverse Path Forwarding) check:
  - drop packets that arrive on incorrect interfaces (i.e., not from the unicast direction to the sending host)
  - why? suppress errant packets

DVMRP Pros and Cons

- Pros
  - simple
  - works well with many receivers

    ⇒ *overhead is per-sender*, receivers are passive
- Cons
  - works poorly with many groups

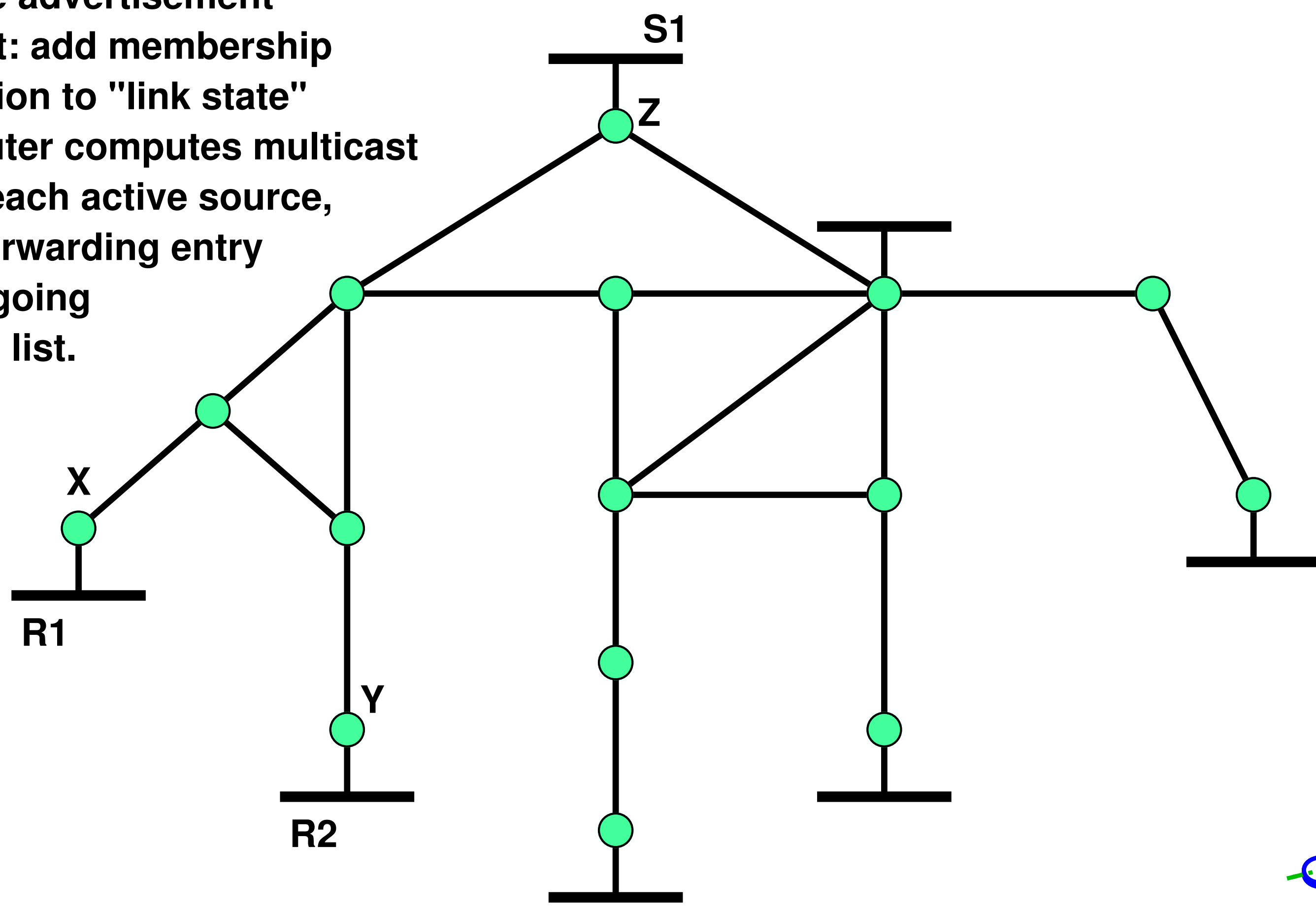
    ⇒ every send in every group floods the nets
  - works poorly with sparse groups

    ⇒ flood data everywhere and then prune back, expensive if only needed at some places

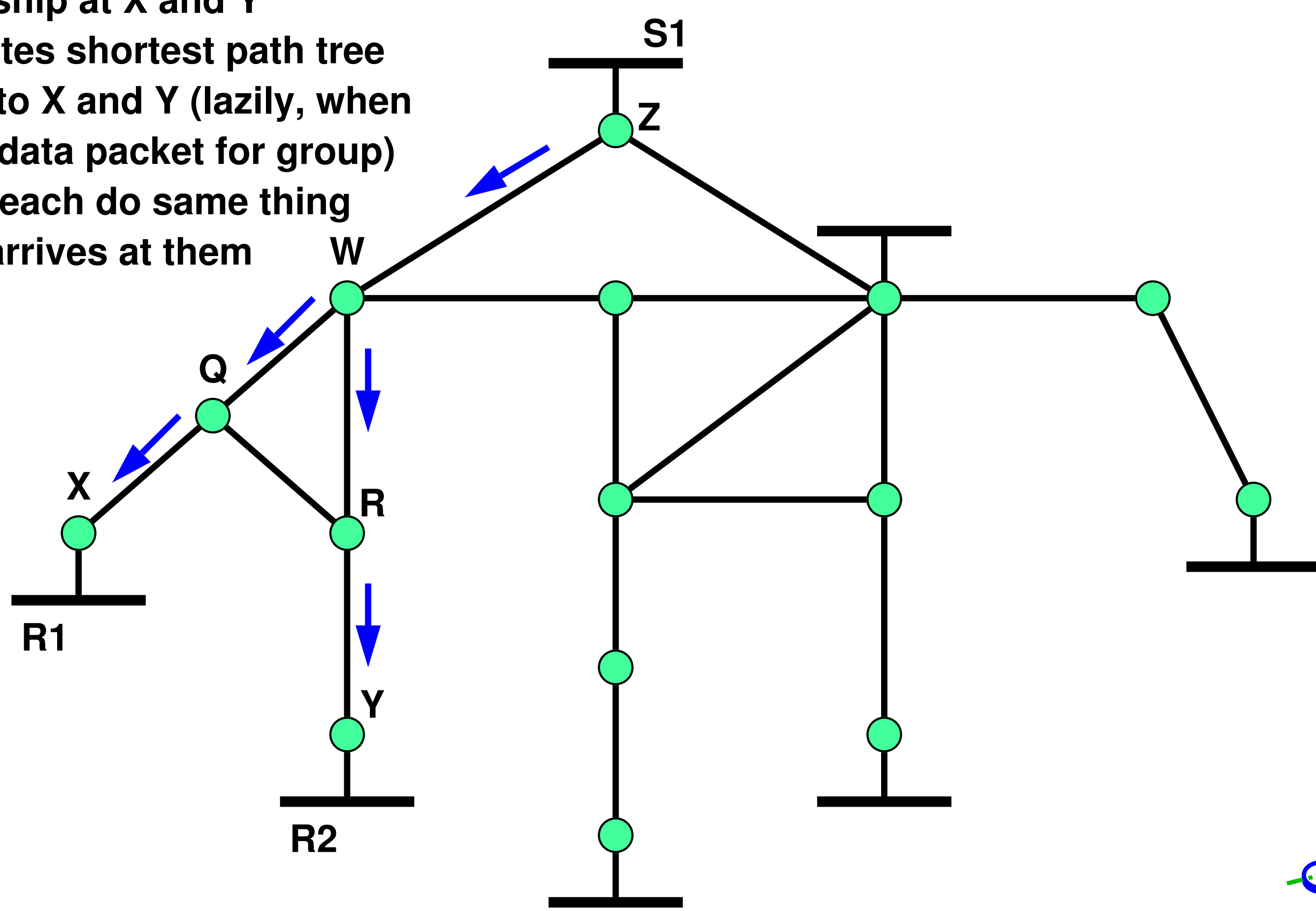
Link-state Multicast Routing

- Basic idea: treat group members (receivers) as new links
  - flood information about them to everyone in LSA message (just like LSA routing)
- Realized as MOSPF (Multicast Open Shortest-Path First)
  - add-on to OSPF
  - each router indicates groups for which there are directly-connected members
  - link-state advertisements augmented with multicast group addresses to which local members have joined
  - link-state routing algorithm augmented to compute shortest-path distribution tree from any source to any set of destinations

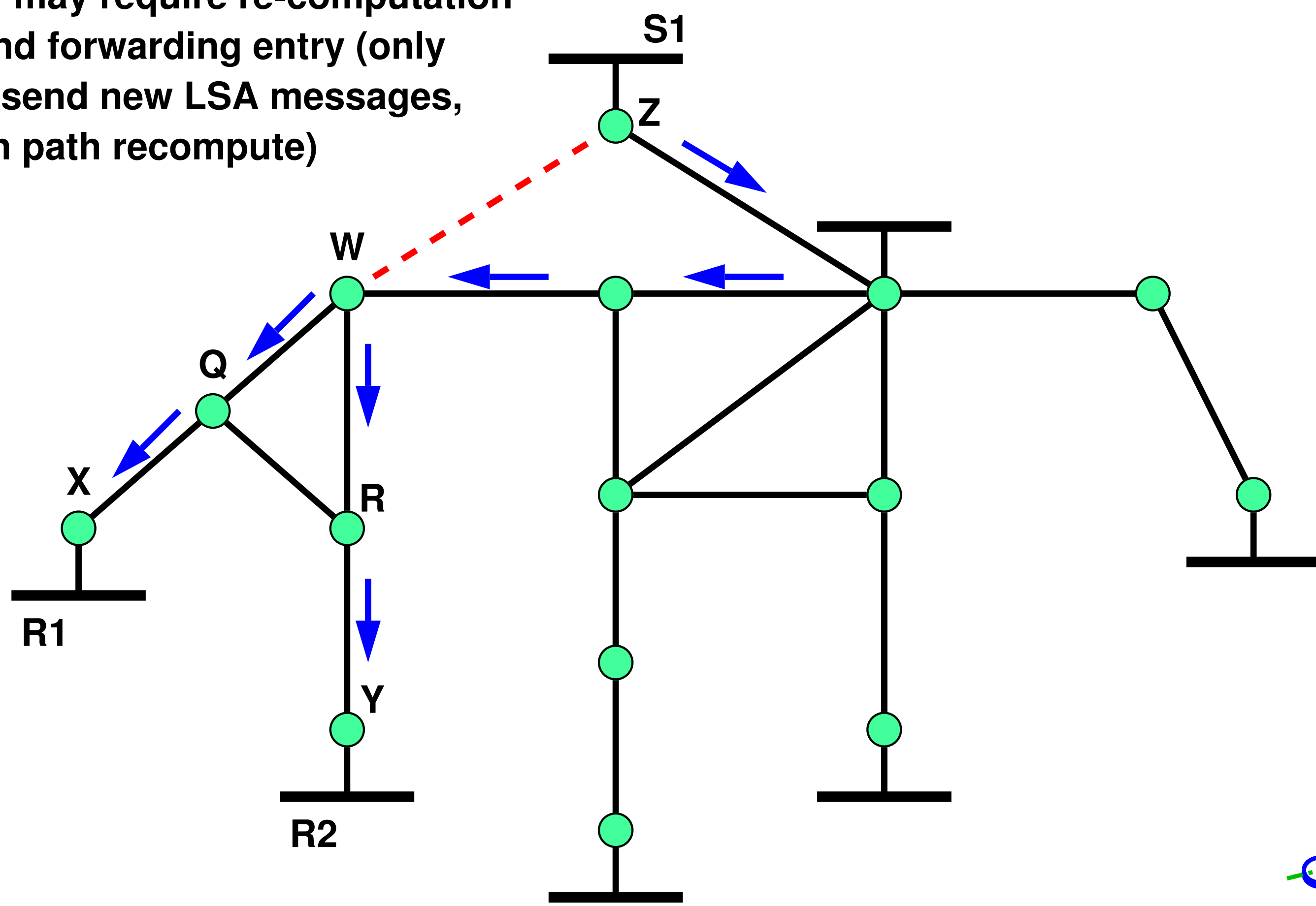
- Link state: Each router floods link state advertisement
- Multicast: add membership information to "link state"
- Each router computes multicast tree for each active source, builds forwarding entry with outgoing interface list.



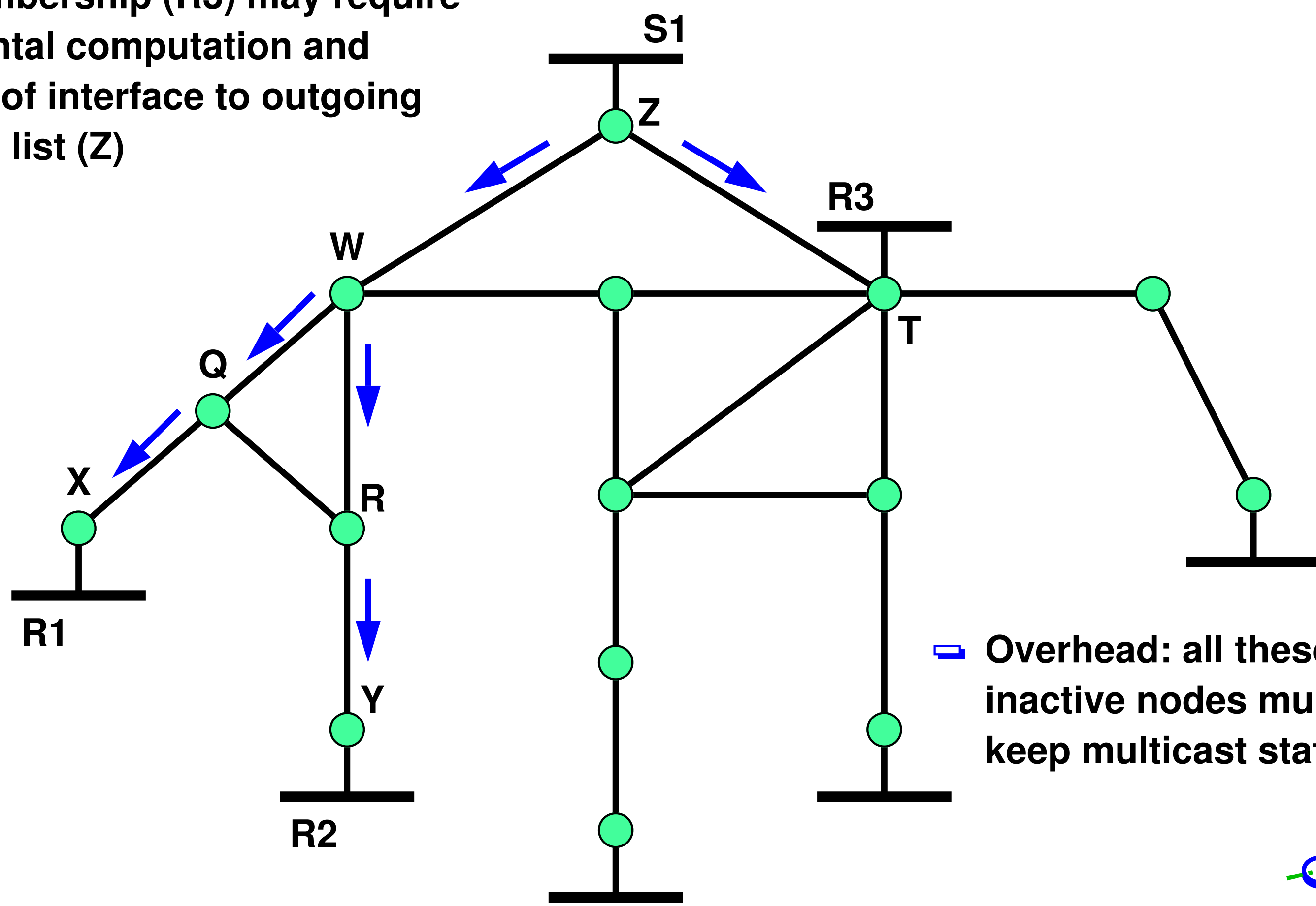
- Z has network map, including membership at X and Y
- Z computes shortest path tree from S1 to X and Y (lazily, when it gets a data packet for group)
- W, Q, R, each do same thing as data arrives at them



- Link state advertisement with new topology may require re-computation of tree and forwarding entry (only Z and W send new LSA messages, but all on path recompute)



- Link state advertisement (T) with new membership (R3) may require incremental computation and addition of interface to outgoing interface list (Z)



- Overhead: all these inactive nodes must keep multicast states

MOSPF Pros and Cons

- Pros
  - simple add on to OSPF
  - works well with many senders
    - ⇒ no per-sender state
- Cons
  - works poorly with many receivers
    - ⇒ per-receiver costs
  - works poorly with sparse groups
    - ⇒ lots of information goes places that don't want it
  - works poorly with large domains
    - ⇒ link-state scales with respect to number of links

      many links causes frequent changes

CS551
Multicast Routing: PIM
[Deering96a]

Bill Cheng

http://merlot.usc.edu/cs551-f12

Key Ideas

- Want a multicast routing protocol that works well with sparse users
- Use a single shared tree; fix one host as rendezvous point

Rendezvous

- With source-based trees senders and receivers meet by:
  - flooding and pruning
  - LS distribution of group and receiver state
- How do we solve the problem?
  - *shared trees*
  - establish a meeting place: center, core or rendezvous point
    - trade-off: shared trees can be inefficient

PIM Protocol Overview

- Basic protocol steps
  - routers with local members *Join* toward *Rendezvous Point (RP)* to join *Shared Tree*
  - routers with local sources encapsulate data in *Register* messages to RP
  - routers with local members may initiate data-driven switch to *source-specific shortest path trees*
- Soft state: periodic state-driven refreshes, time-out idle state
- See PIM v.2 Specification (RFC2362)

PIM Terminology

- ***incoming interface (iif):*** **interface from which multicast packet is accepted and forwarded**
- ***outgoing interface list (oif list):*** **interfaces out of which multicast packets are forwarded**
- ***Rendezvous Point (RP):*** **used in PIM as alternative to broadcast**
- ***Designated Router (DR):*** **one router per multi-access LAN elected to track group membership, and then Join/Prune accordingly**

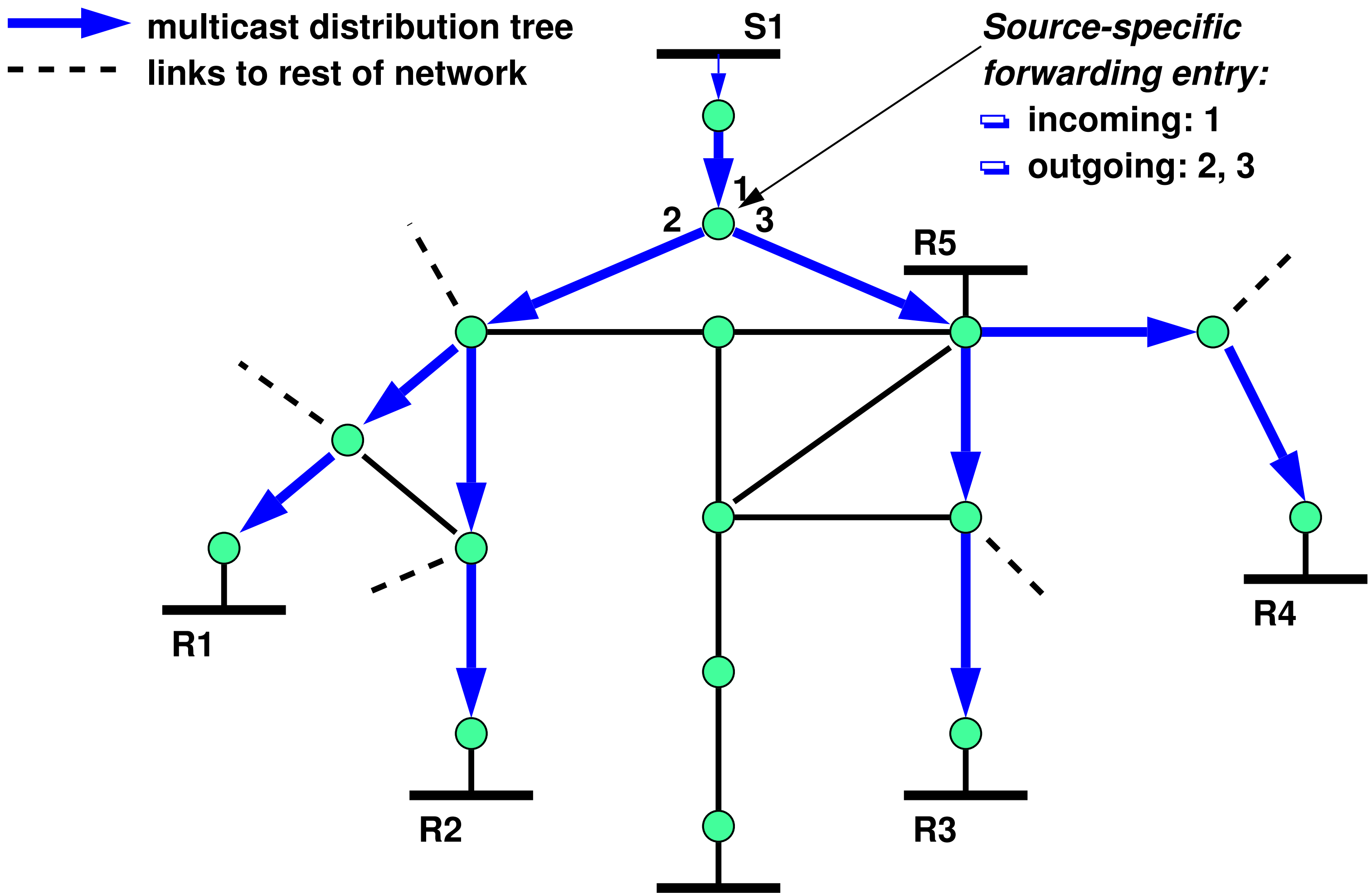
PIM Terminology (Cont...)

- ***Shared tree:*** **reverse-shortest-path tree rooted at RP**
- ***Source-specific tree:*** **reverse-shortest-path tree rooted at source. Also referred to as *Shortest Path Tree (SPT)***
- ***Entry:*** **Multicast forwarding state for a particular source-specific or Shared tree**
- ***Reverse-path forwarding (RPF) check:*** **checks if a packet arrived on the interface used to reach the source of the packet**

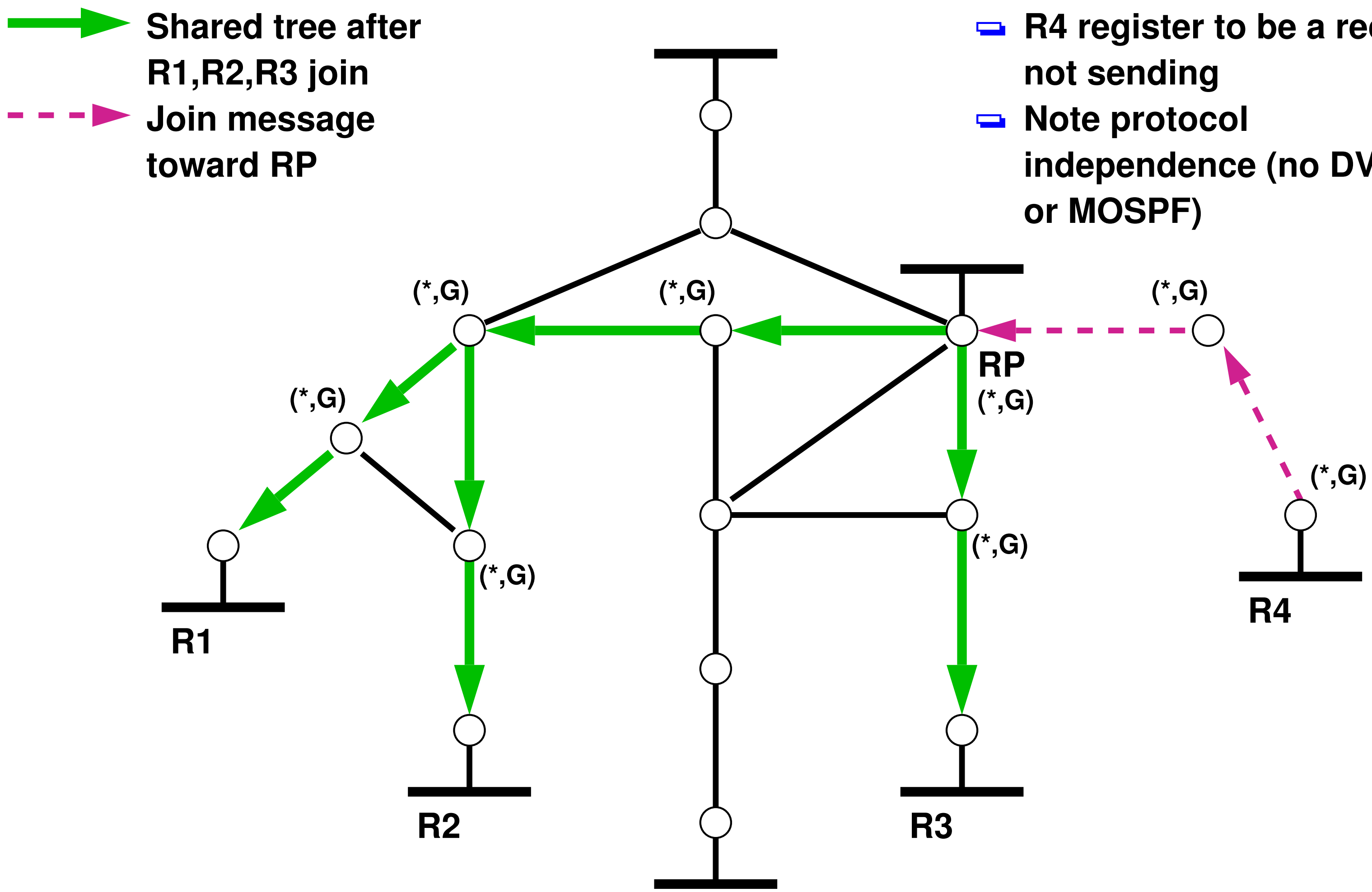
How to Build A Shared Tree

- Quite easy if you have a RP!
  - simply send a message towards the RP
    - use the *unicast* routing table to get there
  - add links to the tree as you go
  - stop if you get to a router that's already in the tree
  - get *reverse* shortest path to RP

Multicast Distribution Tree Example (DVMRP)



PIM Example: Build Shared Tree



- R4 register to be a receiver, not sending
- Note protocol independence (no DVMRP or MOSPF)

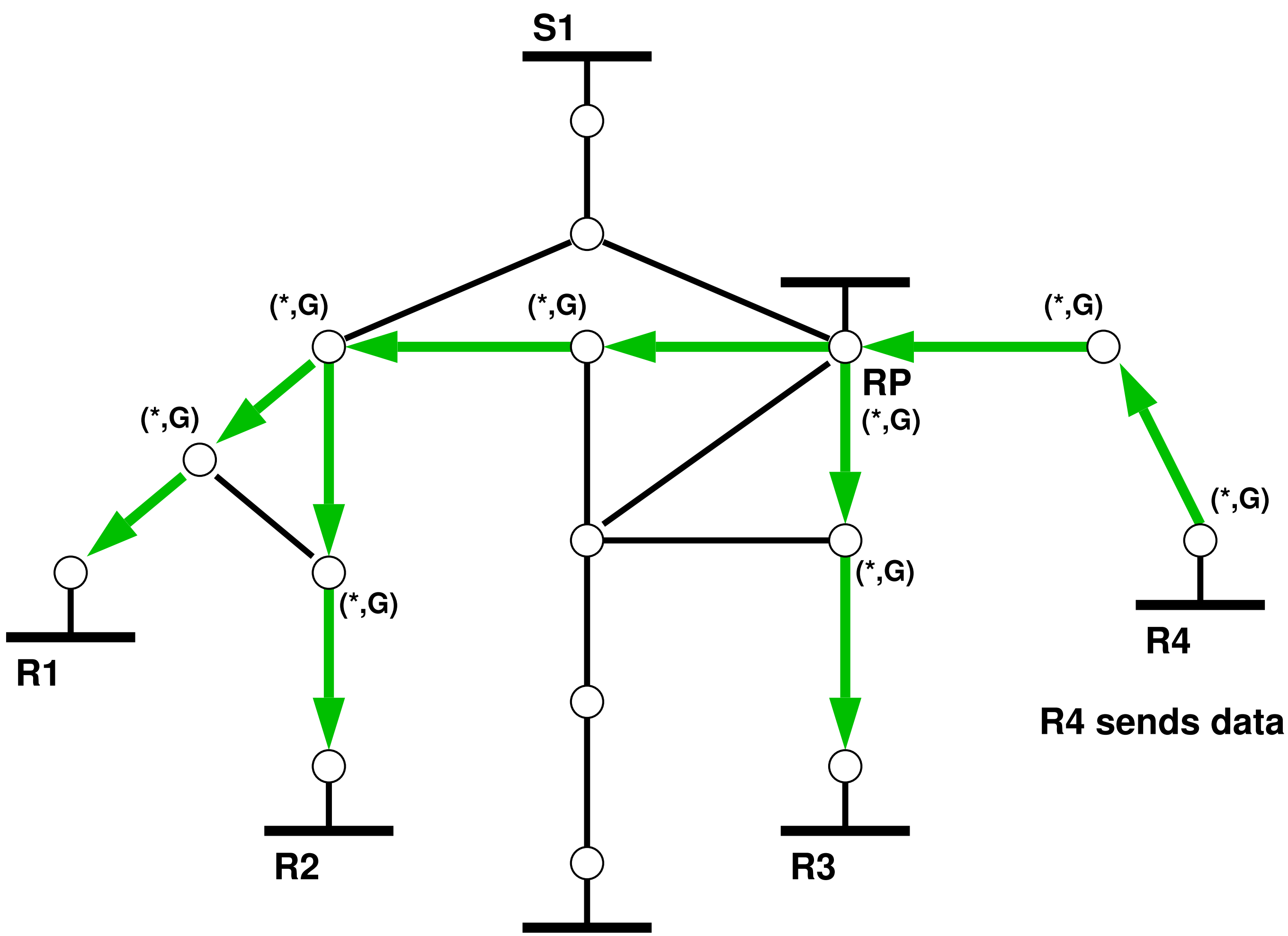
How Do Routers Know RPs?

- RP information is flooded through the network
  - cannot avoid flooding something!
  - but flooding control information is OK
- If there are multiple RPs, each router uses the same hash function to pick a unique RP for the group
  - hash based on group address

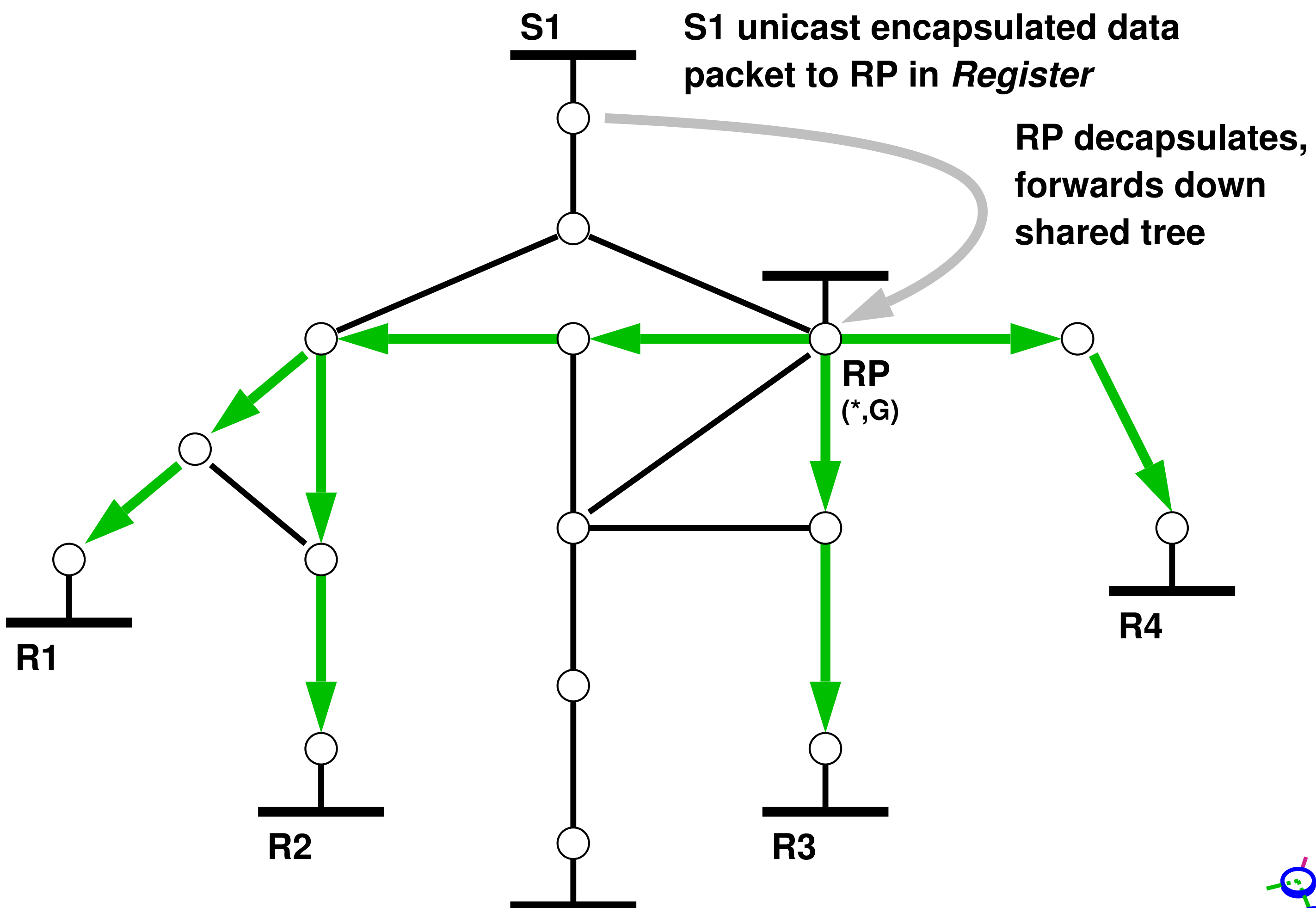
PIM: Sending Data

- If you are on the tree, you just send it as with other multicast protocols
  - it follows the multicast tree
- If you are not on the tree (say, you are a sender but not a group member), the packet is tunneled to the RP that sends it
  - this makes central placement of RP important

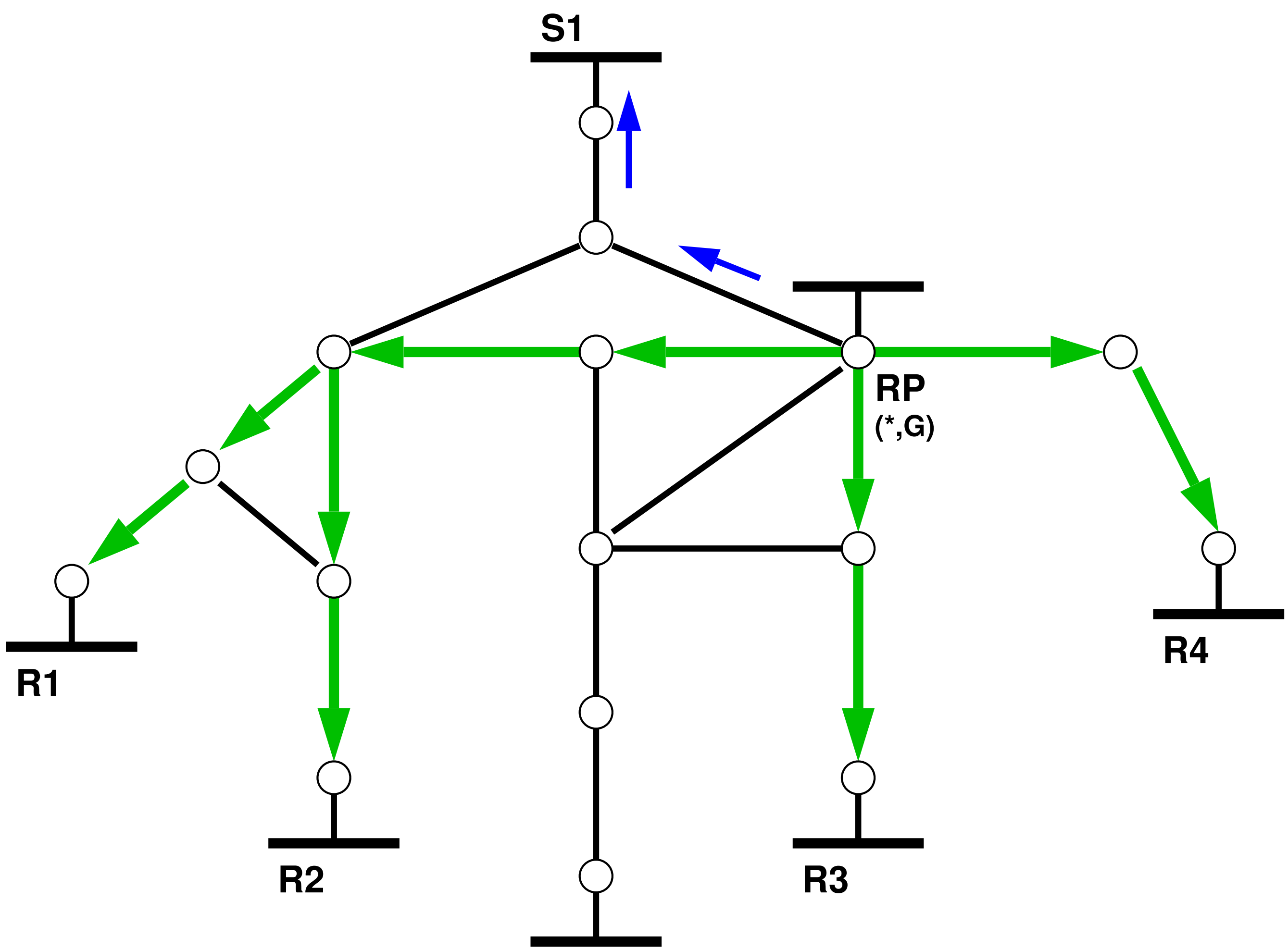
PIM Example: Sending Data On The Tree



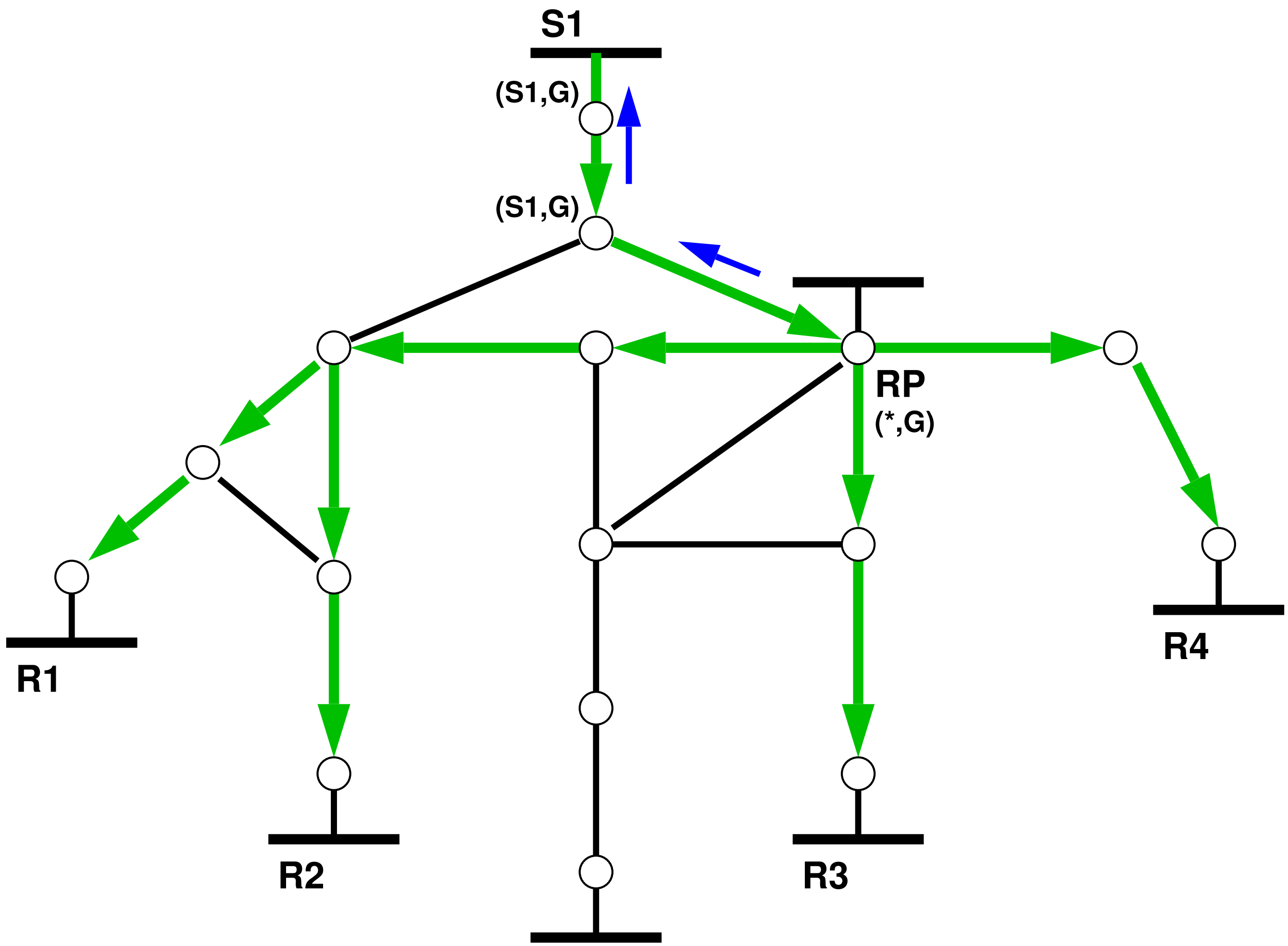
Data Encapsulated in Register



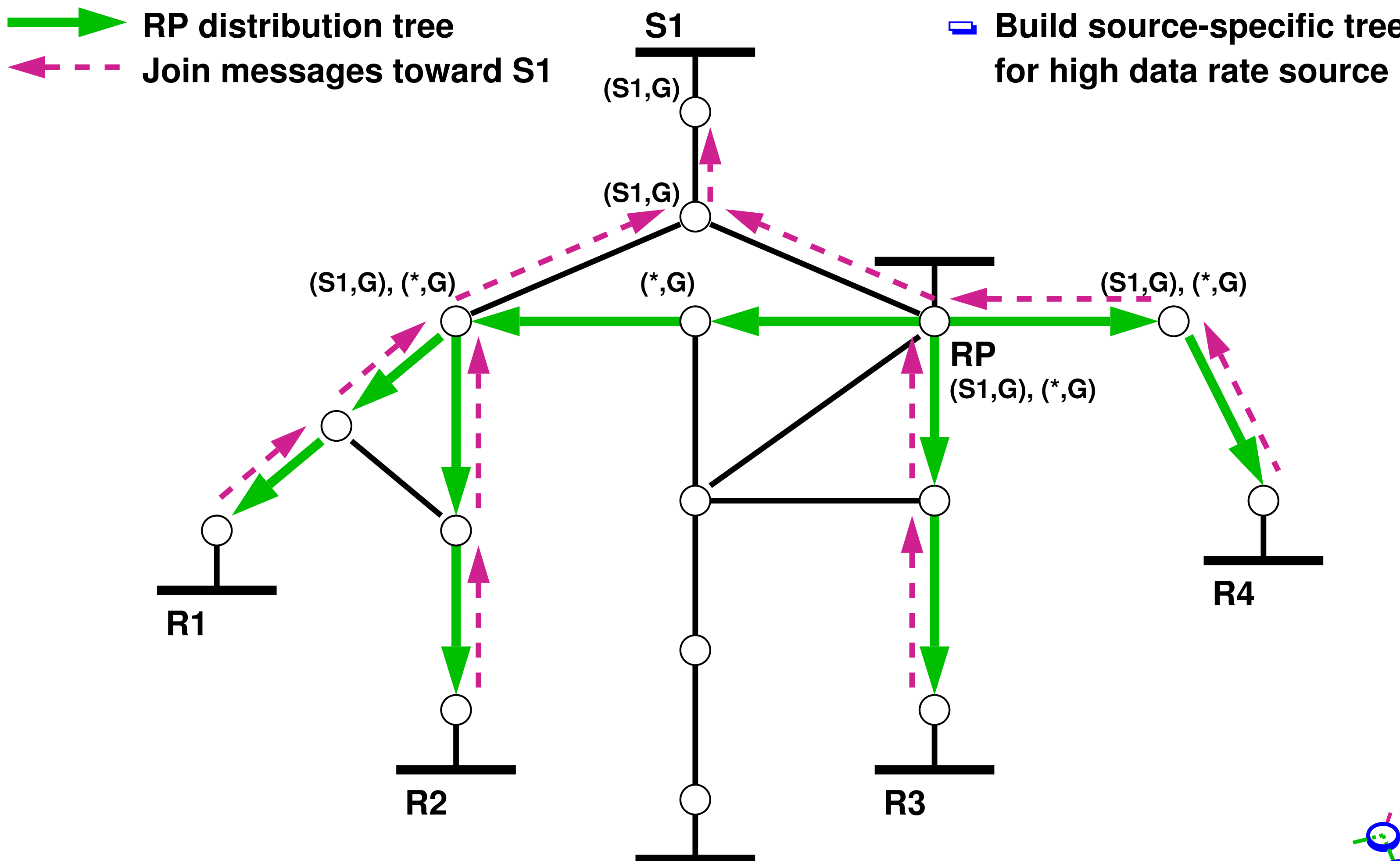
RP May Ask High-rate Src to Join



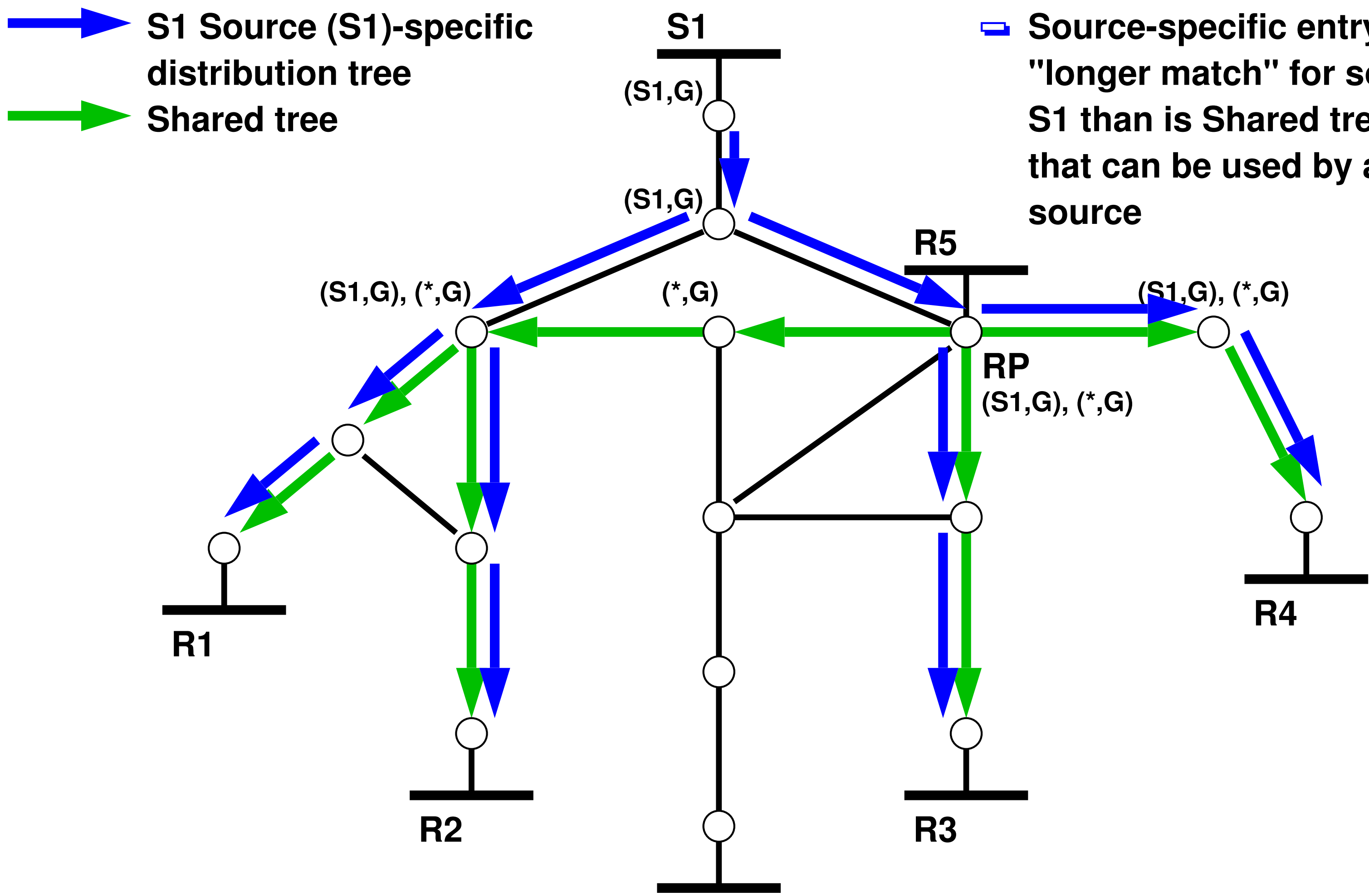
RP May Ask High-rate Src to Join (Cont...)



Build Source-specific Distribution Tree

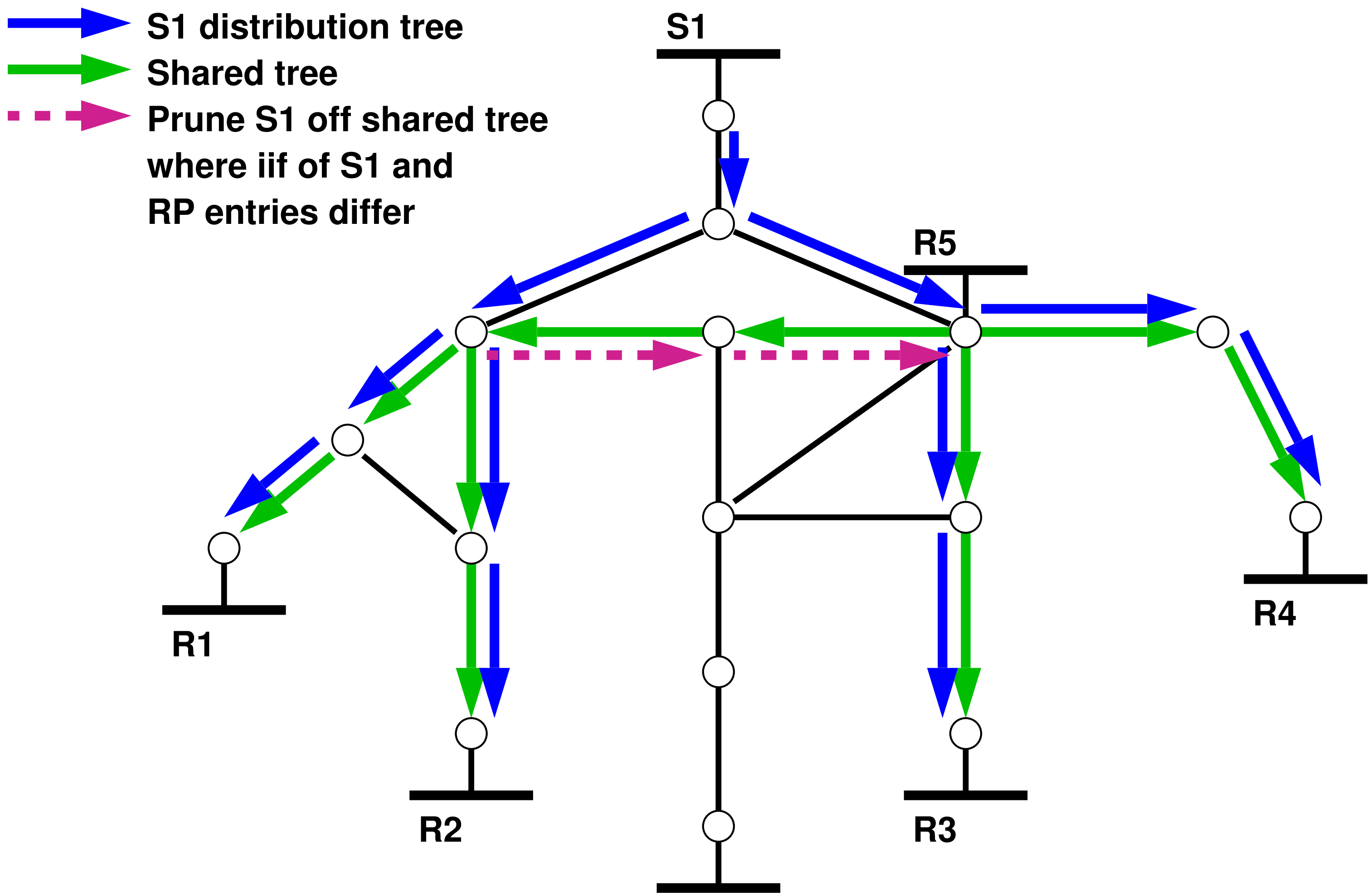


Forward Packets on "Longest Match" Entry



Source-specific entry is "longer match" for source S1 than is Shared tree entry that can be used by any source

Prune S1 off Shared Tree to Avoid Duplicates



Discussion

- Context
  - interest in multicast motivated by audio and video apps
  - PIM was part of a large body of work in multicast routing
- Impact
  - improved scalability compared to DVMRP and MOSPF
  - standardize and implemented
- Multicast status
  - PIM is an intra-domain routing protocol
    - RP flooding limits scalability
  - subsequent work developed inter-domain multicast protocols
    - BGMP & MSDP
  - multicast deployment deadlock
    - management of multicast is hard