An Architecture for Wide-Area Multicast Routing (Protocol Independent Multicast - PIM)

Deering, Estrin, Farinacci, Jacobson, Liu, Wei

SIGCOMM 94

Motivation

- Receivers and senders of are often sparsely populated over a very wide area.
- An internetwork often contains many possible paths between a source *S* and any receiver.
- Flood and prune protocols discover receivers by sending packets everywhere, and pruning back when there are no receivers
  - *inefficient in the wide area*

UTD

Shared and Shortest-Path (source) distribution trees

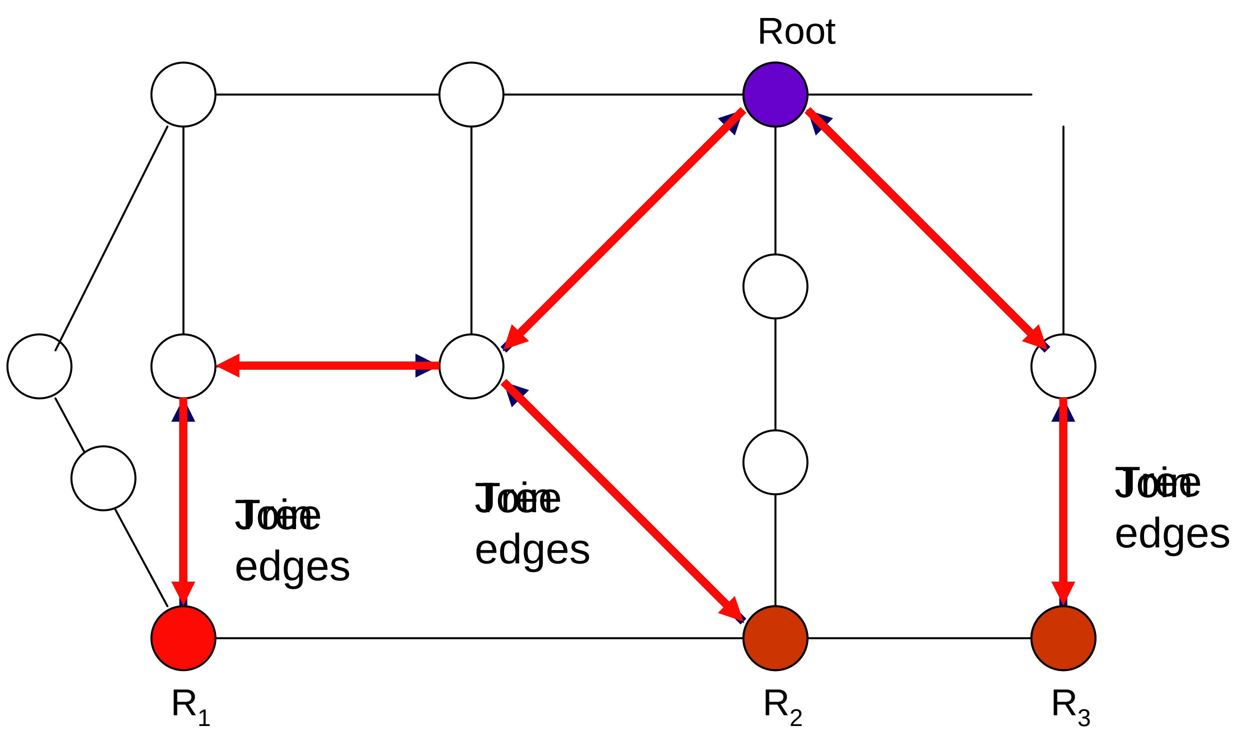
- Shared tree is used by all receivers and senders of a group.
  - Allows very large scale.
  - All sources can send data via this tree
  - All receivers receive data via this tree
  - Routers only need to keep track of information for one tree
- Shortest-Path (source) trees are possible
  - A specific tree can be built for a source with much traffic
  - Last hop routers can:
    - change to the shortest path tree for certain sources
      - switch to shortest-path tree usually if the data rate of the source is high
    - receive from the shared tree for other sources.

Receiver Driven

- Trees are built using a form of reverse-path forwarding
  - Your parent on the tree is the next-hop to the root.
- Explicit join/prune tree management.
  - Routers with local (same LAN) receivers send an explicit join along the path to the root
  - All routers along this path will join the multicast tree (if not on it already)
  - Prune messages remove tree branches if receivers are no longer on their subtree
- We say that PIM-SM is thus ***receiver driven***.

Tree Construction

- Designated routers send join messages along the next hop to root
- Each router along the way
  - records where the message is received from (i.e. who is the child)
  - Forwards the join to the next hop to the root (its parent)
- Tree edges are maintained by periodic refresh messages



Routing protocol independence

- Makes use of existing unicast routing functionality to guide tree construction.
- It is independent of the particular protocol used.
- Only knowledge of the next-hop to a destination is needed.

Multicast forwarding states at routers

- used to determine how a multicast packet will be forwarded
- consists of several elements:
  - source address – S (S = * for the shared tree)
  - group address - G
  - incoming interface – iif (parent on the tree)
    - Determined by the reverse-path forwarding check
      - i.e., my parent is the next hop to S
  - outgoing interface list – oif list (children on the tree)
    - Determined by receiving join messages from them
- Only packets arriving on the iif are accepted

Deconstruction

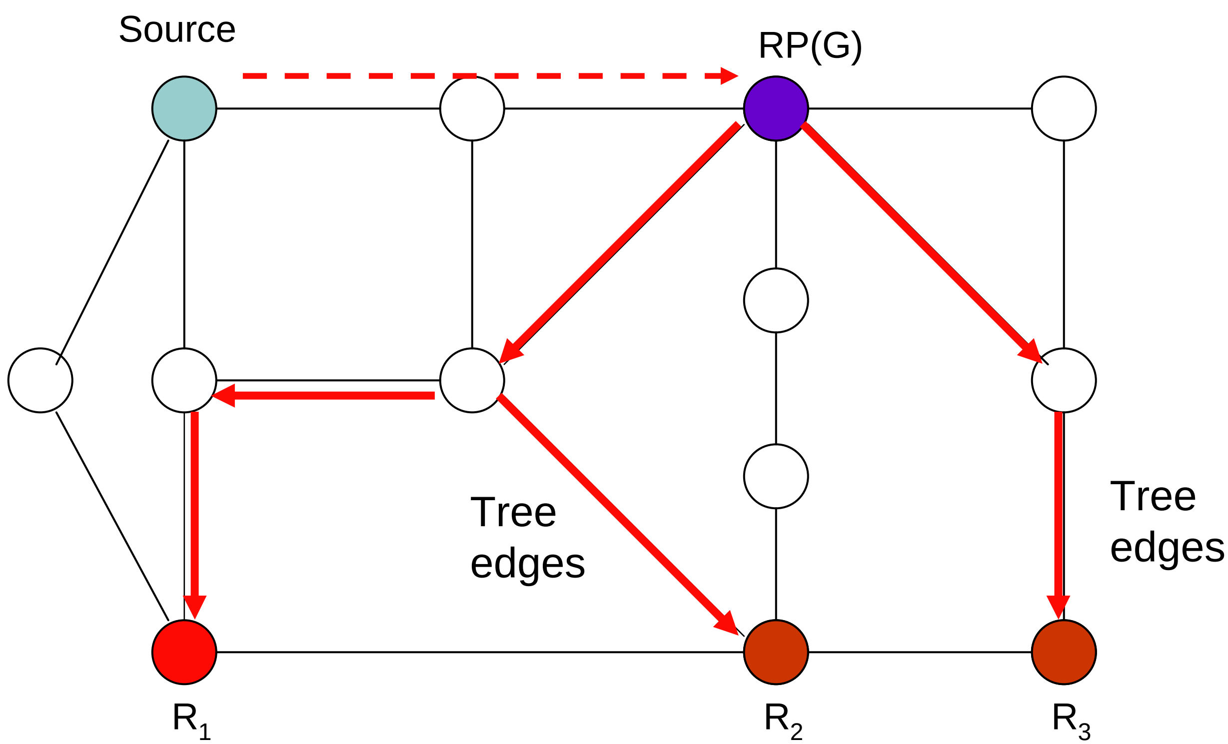
- Prune messages are used to remove edges from the tree
- A router sends a prune message to its parent if it no longer receives IGMP messages from its LANs and no refresh messages from other routers.
- Timeouts also remove edges from the tree if a periodic join is not received from a child.

Rendezvous Point (RP)

- Each multicast group *G* has a router known as the *rendezvous point (RP)*.
  - The RP is the root of the *shared tree* for group *G*.
  - Thus, the shared tree is also known as the *RP tree (RPT).*
- Every router must know the RP for every group *G*.
  - In some cases, the same router is used as the RP for every group.
  - There is only one RP for any group *G* at any time*.*
- *RP(G)* denotes the unicast IP address of the RP of G

Overall Steps

- A receiver sends an IGMP message to its designated router (DR) wishing to join group G
- DR joins the shared tree rooted at RP(G)
  - Its join message indicates (*,G), i.e., wishes data from all sources
- Sources send data messages to RP(G) via encapsulation
  - Data is encapsulated and routed to RP(G)
    - The message is called “register”
  - RP(G) then decapsulates it and forwards it along the tree


Source
RP(G)
Tree
edges
Tree
edges
R1
R2
R3

Shortest Path Trees

- The ***shortest path tree*** (*SPT*) for a source *S* of a group *G*
  - is a tree rooted at the designated router for *S*
  - for every receiving host *R,*
    - the path between *R* and *S* is the shortest path from *R* to *S.*
    - if we use cost, the optimum path from S to R may not be the optimum path from R to S.

- The SPT is built as before (root now is S)
- BOTH the SPT of S and the Shared Tree will co-exist.
- Who should join the SPT of S?

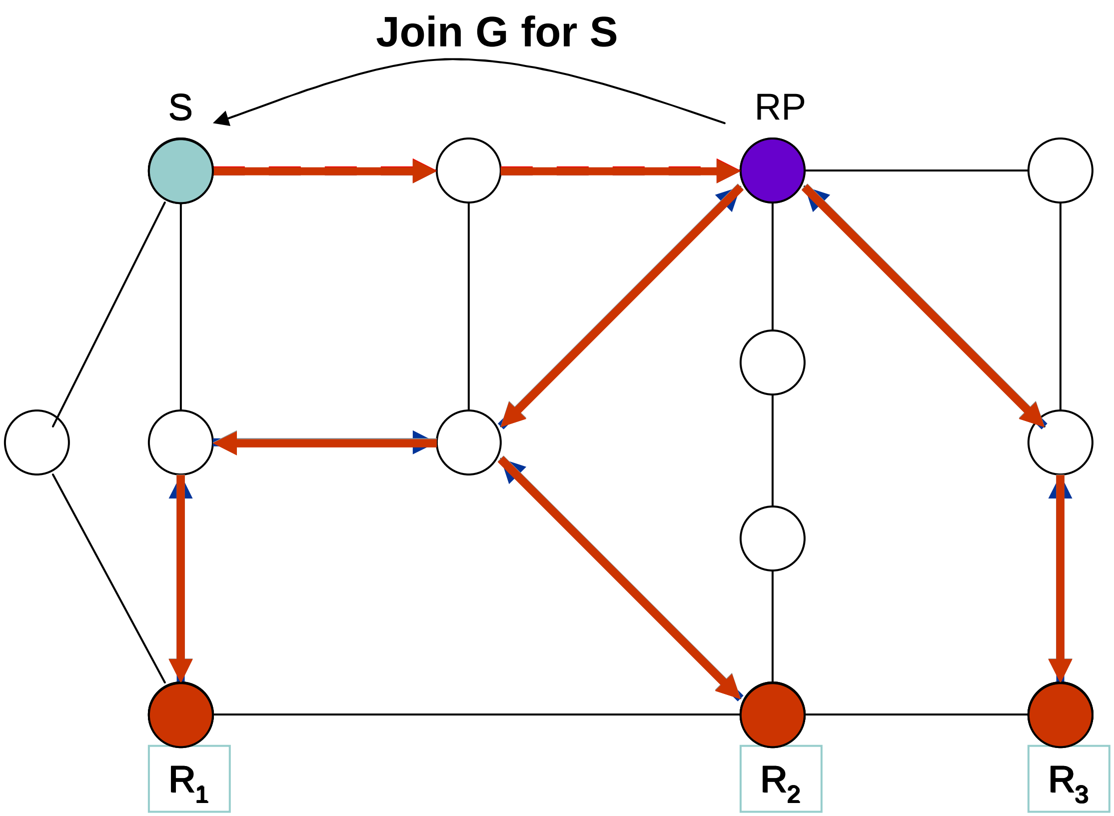
RP(G) joins SPT of S

- If the traffic from S is high, the RP(G) may wish to avoid encapsulation/decapsulation.

- To do so, RP(G) joins the SPT of S
  - I.e., RP(G) simply becomes a receiver of SPT of S
  - If SPT of S does not exist, it is built on the fly, i.e. RP(G) is the first receiver of the SPT of S.

- Thus, RP(G) receives data messages from S through SPT of S
  - RP(G) then forwards these messages along the shared tree
  - RP(G) then sends a “stop register” message to S
    - Otherwise it would get two copies of every message.

PIM-SM Example



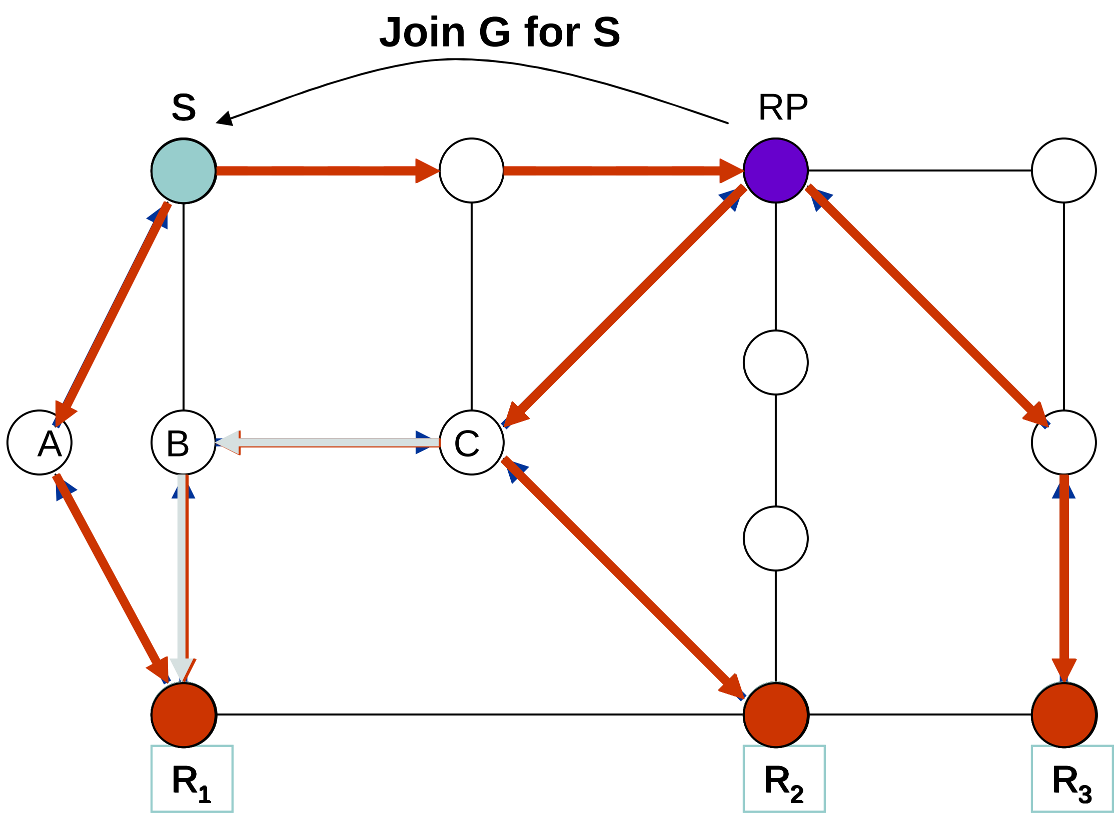
Step 1: Building shared tree
Step 2: Source sending to RP
Step 3: Stop encapsulation

Receiver Joins the SPT of S

- Due to traffic load, the DR of a receiver may decide to join the SPT of S.
  - I.e., S → DR will have the optimum path from DR to S
- Thus, the DR will be connected to two trees.
  - Shared Tree
  - SPT of S
- For any router R
  - Once messages arrive along the parent of the SPT of S, the router PRUNES itself from the Shared Tree
  - This prune is selective only for the source S
  - Data from other sources will continue to be received via the shared tree

UTD

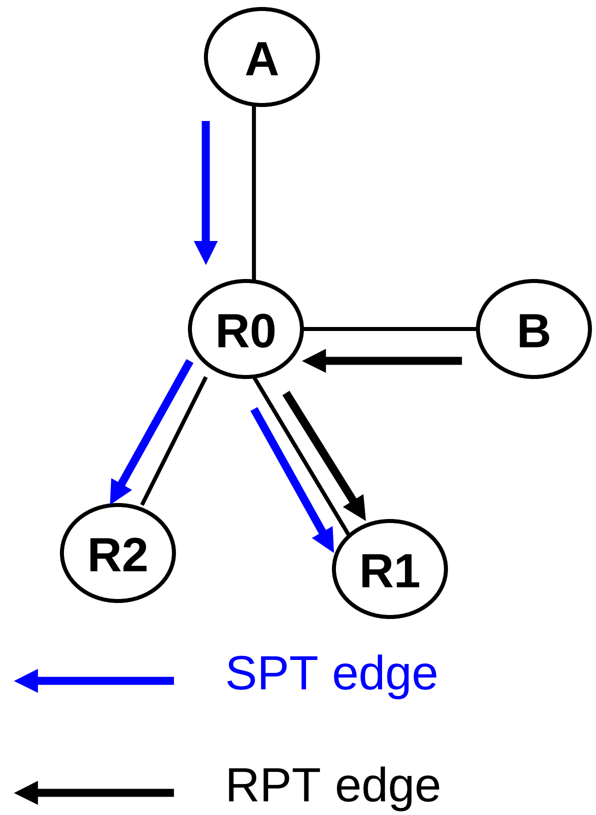
PIM-SM Example



Step 1: Building shared tree
Step 2: Source sending to RP
Step 3: Stop encapsulation
Step 4: Switch to SPT
Step 5: Prune shared tree

Overlap of SPT and RPT

- What if SPT and RPT overlap?
- R1 sends join(S,G) to R0, R0 sends join(S,G) to A.
- As long as data from S is not received along the SPT(S,G), R0 considers SPT(S,G) to be **inactive**
  - R0 forwards data from S only to RPT(G) children (i.e. R1)
- When data from S is received along SPT(S,G), R1 considers SPT(S,G) to be **active**
  - R0 forwards data from S to children on **both** SPT(S,G) and RPT(G)
  - R0 sends a prune(S,G) to B
- Note: R1 does not send prune(S,G) to R0, why?



A = parent on SPT(S,G)

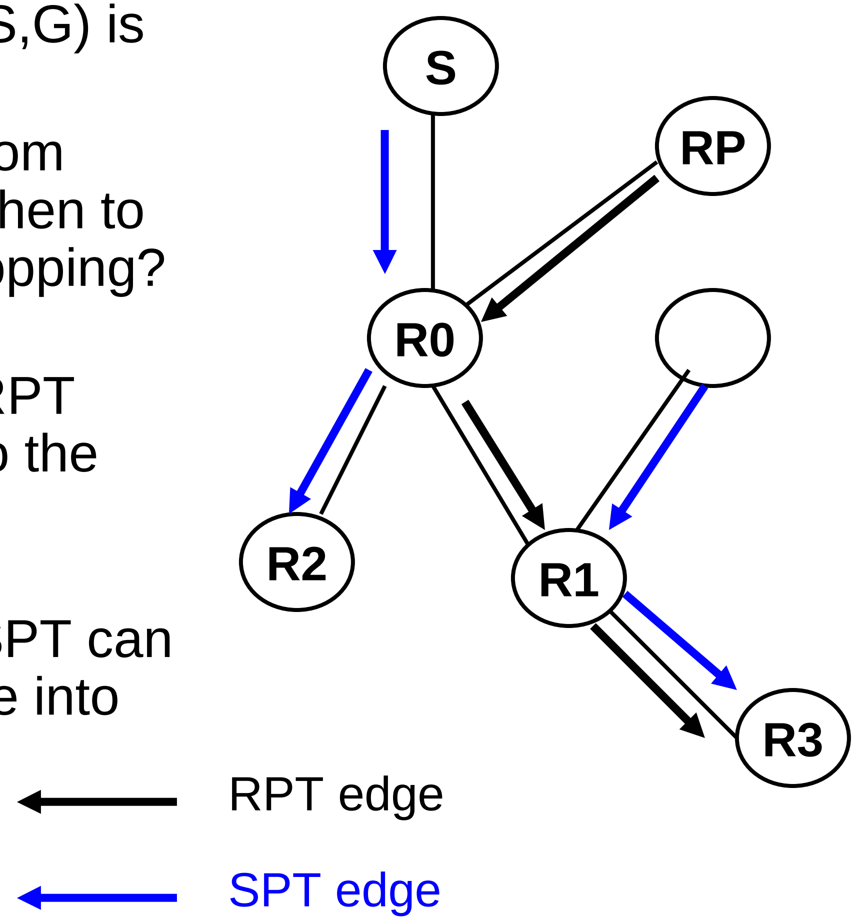
B = parent on RPT(G)

Why the pruning?

- You don’t want data to be sent more than once along the tree
- Also, depending on how the SPT and RPT tree are shaped, data could go back-and-forth between them forever!
- Note that if your parent on SPT(S,G) and on RPT(G) are different
  - If you receive multicast from S along the RPT(G) you throw it away even if it arrives first than on SPT(S,G)

Data crossing from one tree to another?

- Assume the SPT(S,G) is active.
- Can the data go from SPT then to RPT then to SPT … without stopping?
  - A message originating at RPT cannot cross to the SPT
  - A message originating at SPT can cross only once into RPT



How do routers discover sources?

- A source *S* for group *G* is "discovered" by a router when the router receives a packet from *S* addressed to group *G.*
  - A router discovers a **directly connected source S** when the router receives a packet from *S* on the same LAN and addressed to group *G*.
    - The router must encapsulate and register with RP(G)
  - A router discovers a **distant (non-connected) source** when the router receives a packet from the source along the shared tree for group *G.*
    - The router may choose to join the SPT of S.

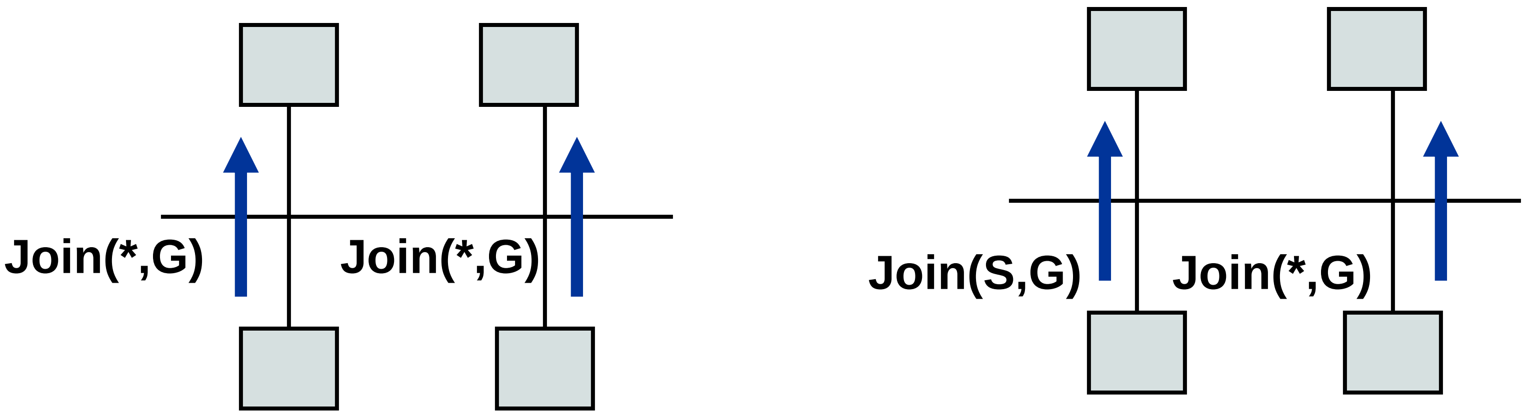
Pros and Cons

- Pros:
  - Good in sparsely populated networks (few receivers)
  - Only one tree is necessary (other trees for efficiency if desired)

- Cons
  - If no sources are sending data right now, the routers still need to maintain information about the shared tree
  - The SPT is shortest from receiver to source, not from source to receiver.

What about multi-access links

- The following scenarios are possible



- In both cases, the top two routers send G messages to the LAN (duplicating effort)

Solution

- Routers hear all multicast messages
- The top routers realize there are multiple parents on this link, and they broadcast “assert” messages on the link
- For a source S
  - The router that is a member of the SPT of S wins (puts (S,G) multicasts on this LAN)
  - If both are on the SPT of S, then the one “closest to S”
  - If neither are, then the one closest to RP(G).

What about the children

- Assert messages are also heard by the children
- Their subsequent join messages are sent to the router who “won” the assert election process.