**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 packet 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

**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

**Multicast Routing: Problem Statement**
- **Goal**: find a tree (or trees) connecting routers having local multicast group members
  - tree: not all paths between routers used
  - source-based: different tree from each sender to receiver
  - shared tree: one tree used by all group members

**Approaches for building multicast 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

![Diagram of Shortest Path Tree]

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

![Diagram of Reverse Path Forwarding: example]

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

![Diagram of Reverse Path Forwarding: pruning]

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

![Diagram of Steiner Tree]

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

![Diagram of Center-based Trees]
Center-based trees: an example

Suppose R6 chosen as center:

![Diagram showing router connections and paths]

**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: re-prune 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?

![Diagram showing tunneling]

- 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!"