

# Advanced Networking

## Multicast

Renato Lo Cigno

Renato.LoCigno@disi.unitn.it

Homepage:

[disi.unitn.it/locigno/index.php/teaching-duties/advanced-networking](http://disi.unitn.it/locigno/index.php/teaching-duties/advanced-networking)

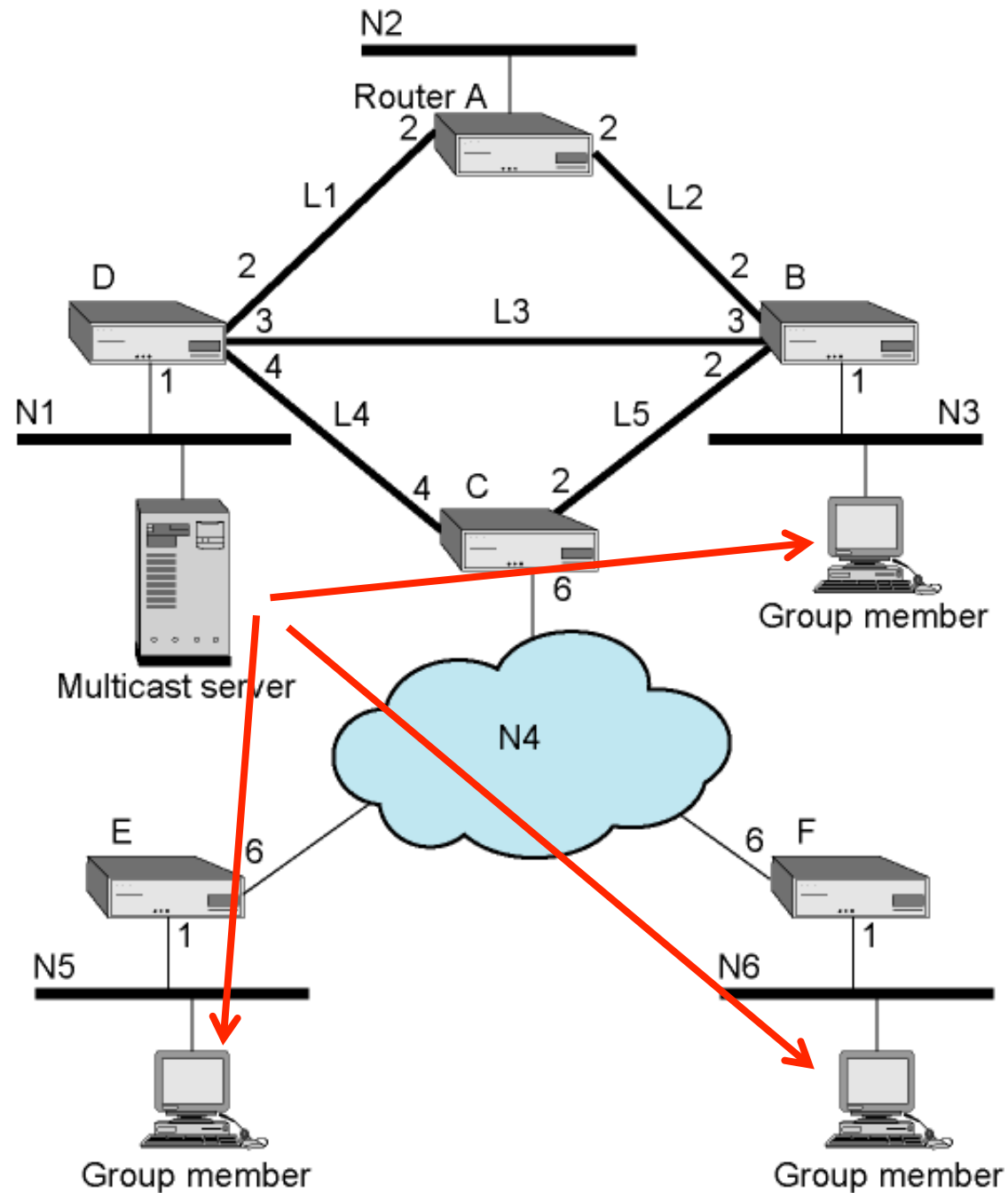
# Multicasting

- Addresses that refer to group of hosts on one or more networks
- Applications
  - Multimedia “broadcast” and streaming
  - Teleconferencing
  - Distributed Database
  
  - Distributed computing (Grids, Clouds, Crowdsourcing, ...)
  - Real time workgroups
  - File distribution

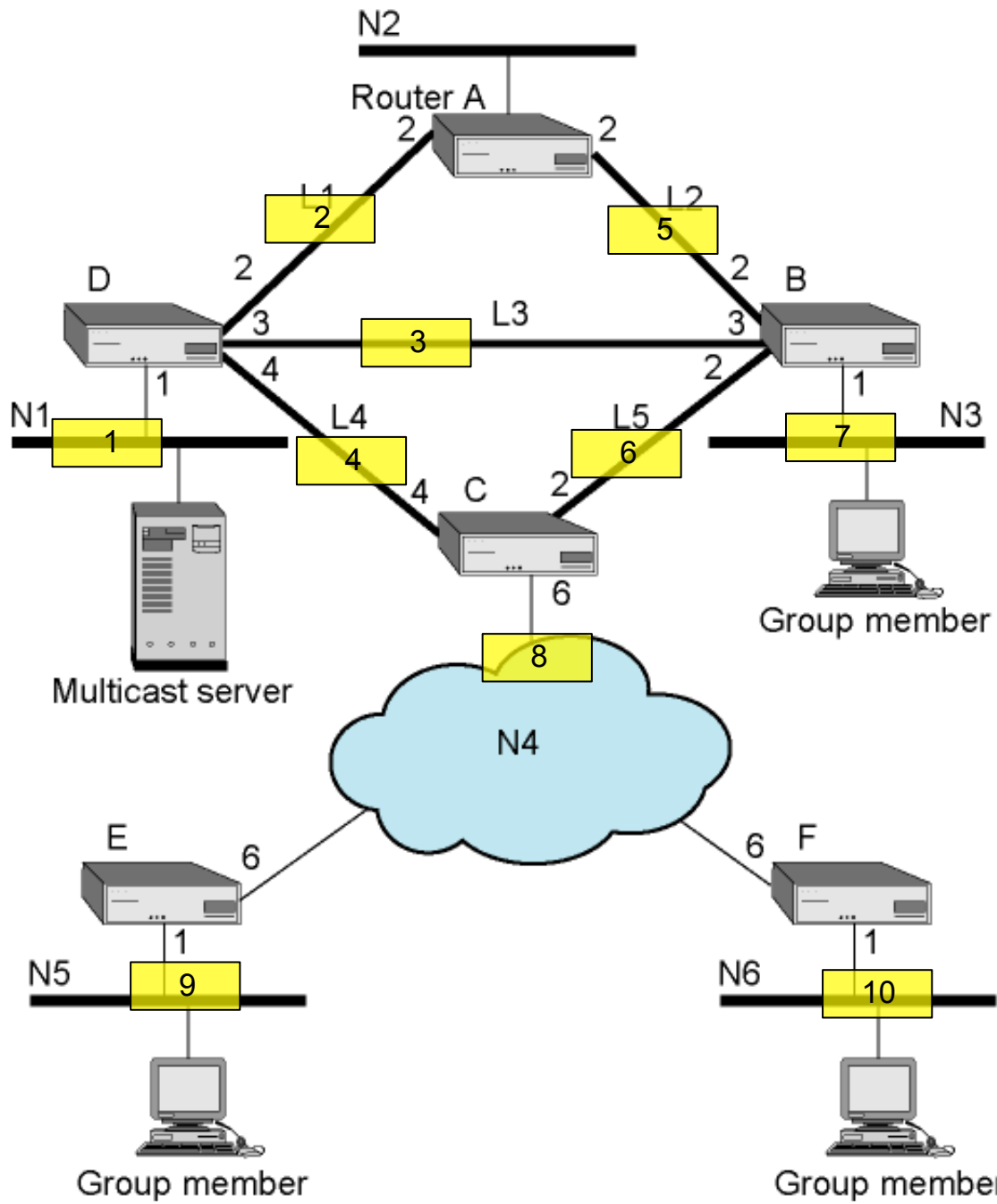


# Example of multicast configuration

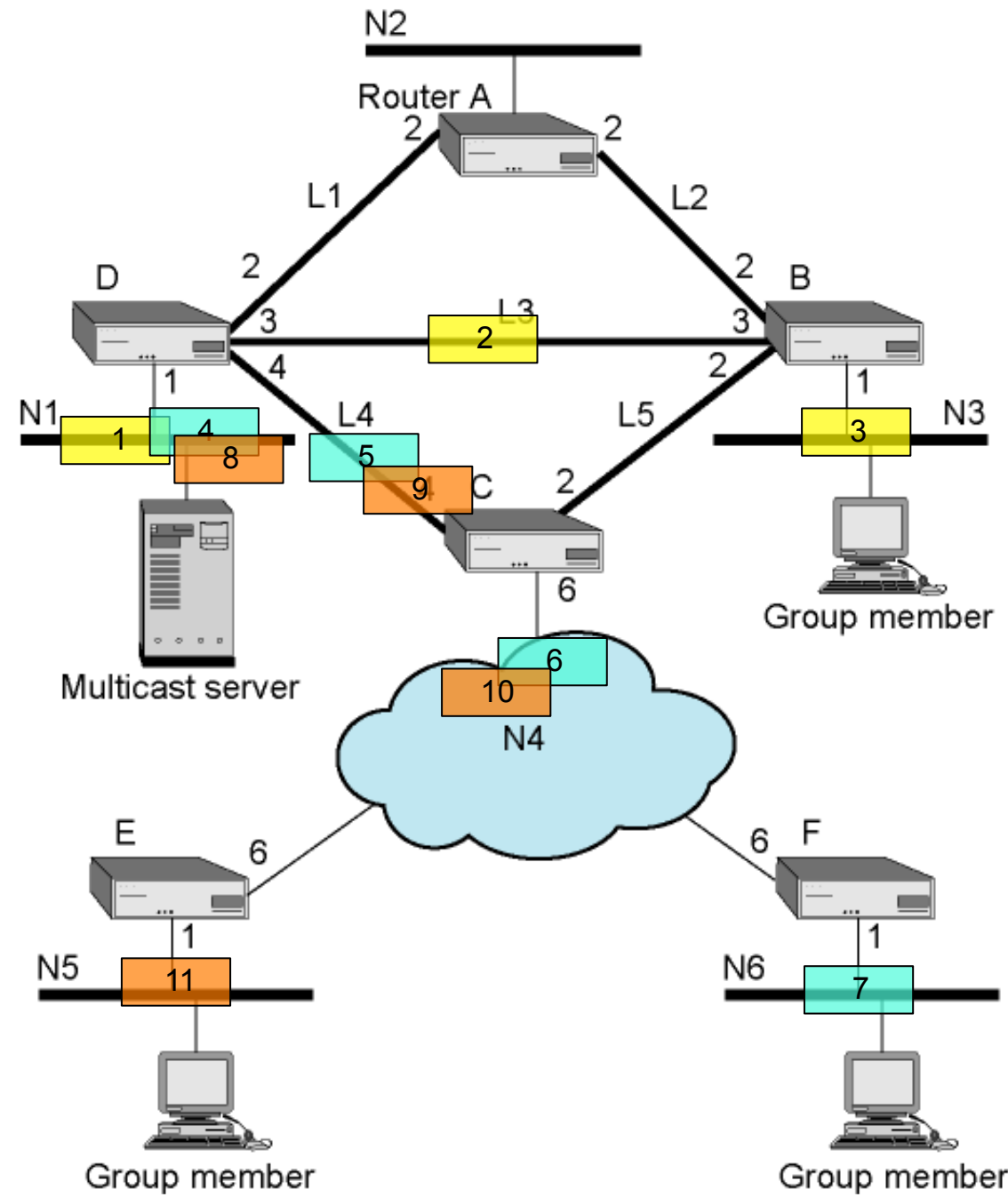
Performance can be measured as the total number of packet required



# Broadcast/ Flooding



# Multiple Unicast



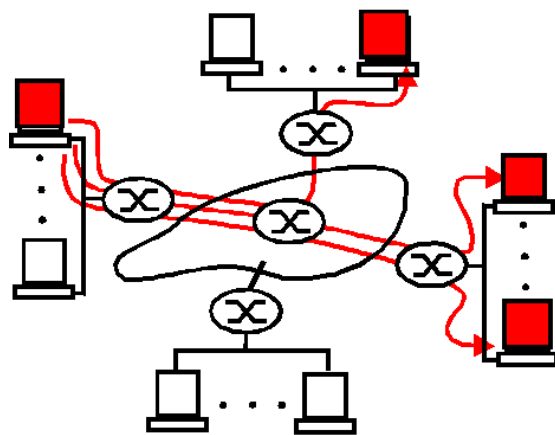
# Broadcast and Multiple Unicast

- Broadcast a copy of packet to each network
  - Requires 10 copies of packet
  - Enormous waste for sparse groups
- Multiple Unicast
  - Send packets directly to each host in the multicast group
  - 11 packets
  - Enormous waste for dense groups
- Can we do better than this?

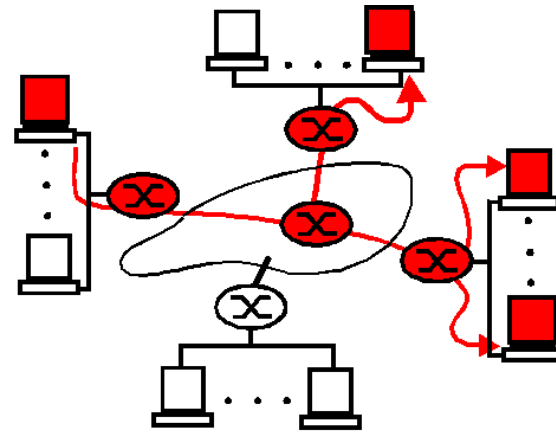


# Multicast Routing

- Multicast: delivery of same packet to a group of receivers with the minimum overhead
- Multiple unicast vs. multicast
  - Host based vs. network based



multicast via unicast



network multicast

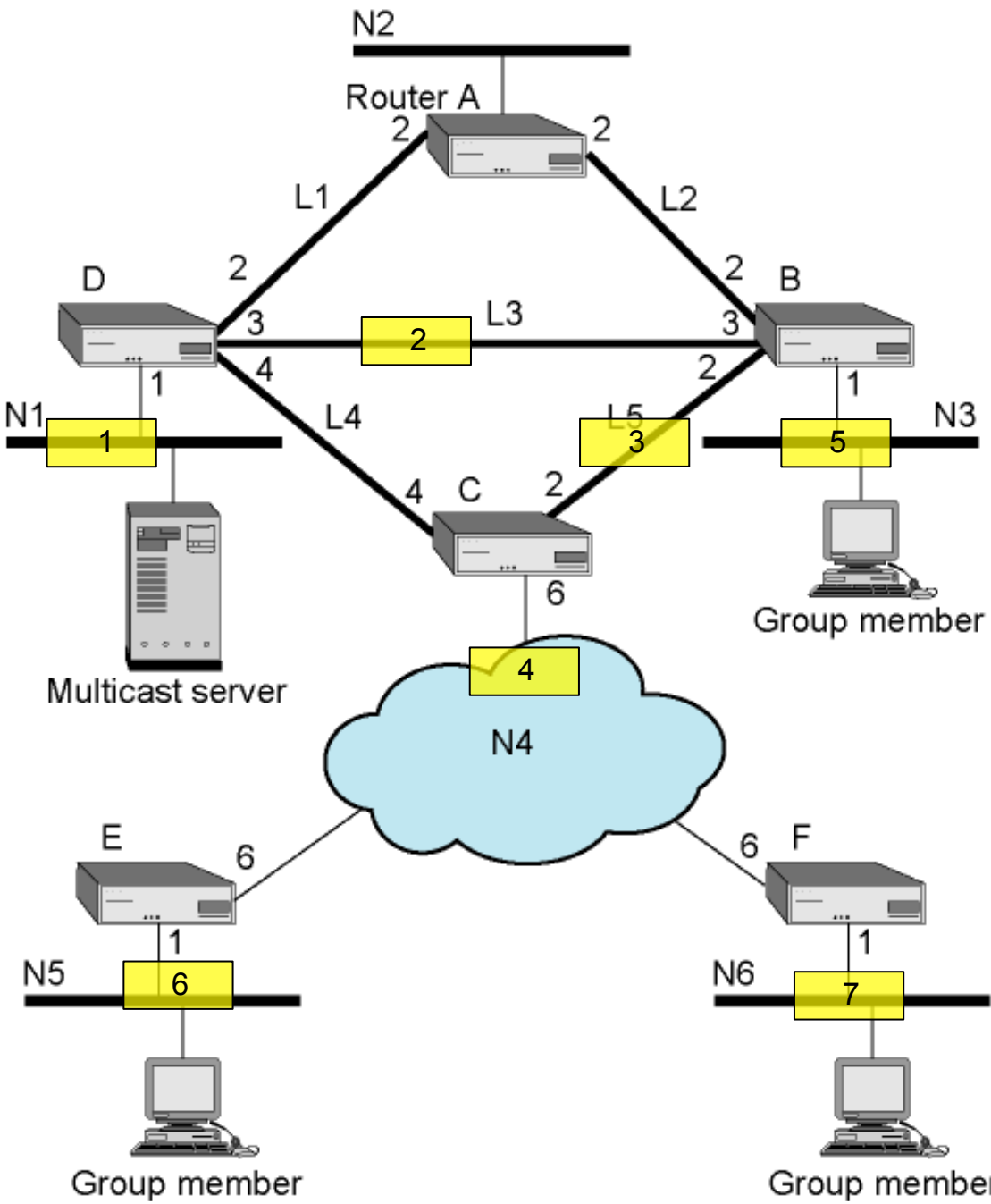
# “True” Multicast

- Determine least cost path to each network that has host in group
  - Find the minimum global spanning tree configuration containing networks with group members
- Transmit single packet along spanning tree
- Routers replicate packets at branch points of spanning tree
- 7 packets required ... let's see if it is true





# True Multicast



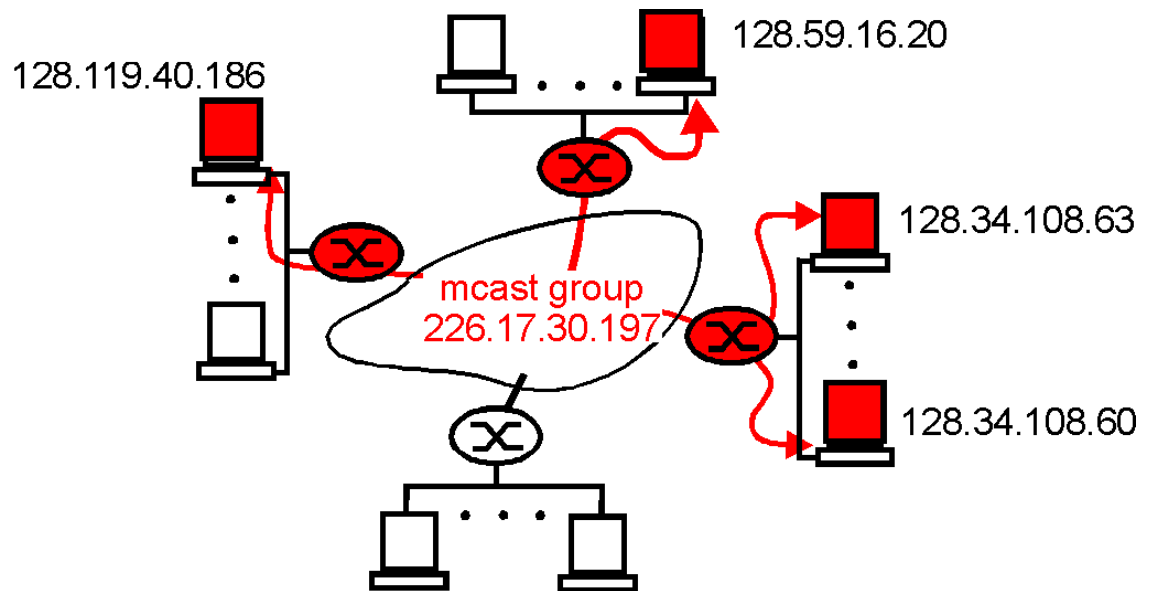
# Spanning Tree Problem

- Given a graph  $G=(V,E)$ 
  - nodes are vertices and links are edge
  - connected and undirected
- A Spanning Tree (ST) for  $G$  is a subgraph without cycles (i.e., a tree) which covers all vertices
- There are one or more STs for  $G$



# Multicast Group Address

- M-cast group address “delivered” to all receivers in the group
- Internet uses Class D for m-cast
- M-cast address distribution etc. managed by IGMP Protocol



# Requirements for Multicasting (1)

- Router may have to forward more than one copy of packet
- Convention needed to identify multicast addresses
  - IPv4 - Class D - start 1110
  - IPv6 - 8 bit prefix, all 1, 4 bit flags field, 4 bit scope field, 112 bit group identifier
- Nodes must translate between IP multicast addresses and list of networks containing group members
- Router must translate between IP multicast address and network multicast address



# Requirements for Multicasting (2)

- Mechanism required for hosts to join and leave multicast group
- Routers must exchange info
  - Which networks include members of given group
  - Sufficient info to work out shortest path to each network
  - Routing algorithm to work out shortest path
  - Routers must determine routing paths based on source and destination addresses



# Requirements for Multicasting (3)

- One local protocol (IGMP) for multicast membership
- Several protocols for Multicast tree management
  - Several standards
  - Little support in BGP
  - Rarely deployed by operators but for fully controlled networks, e.g., IPTV



# IPv4 Multicast Space

- Host Extension for IP Multicasting
  - Groups may be permanent or transient:
    - Permanent groups have well-known addresses
    - Transient groups receive address dynamically
  - The multicast addresses are in the range 224.0.0.0 through 239.255.255.255.
  - Which authority coordinates addresses assignments??
    - *<http://www.iana.org>*
    - *The Internet Assigned Numbers Authority (IANA) is the body responsible for coordinating some of the key elements that keep the Internet running smoothly;*



# IANA Activities

- From IANA websites:
- IANA's various activities can be broadly grouped in to three categories:
  - Domain Names: IANA manages the DNS root, the .int and .arpa domains, and an IDN practices resource.
  - Number Resources: IANA coordinates the global pool of IP and AS numbers, providing them to Regional Internet Registries
  - Protocol Assignments: Internet protocols' numbering systems are managed by IANA in conjunction with standards bodies





# Assigning Multicast Addresses

- How does IANA assign IPv4 multicast addresses:
  - Local Network Control Block: range 224.0.0.0 - 224.0.0.255 reserved for routing protocols and low-level topology discovery or maintenance protocols

224.0.0.0	Never assigned
224.0.0.1	All Hosts on this Subnet
224.0.0.2	All Routers on this Subnet
224.0.0.4	DVMRP Routers
24.0.0.13	All PIM Routers (hello messages...)
...	...



# Assigning Multicast Addresses

- There are several addresses blocks already assigned
  - Internetwork Control Block (224.0.1.0 - 224.0.1.255 (224.0.1/24))
  - AD-HOC Block I (224.0.2.0 - 224.0.255.255)
  - RESERVED (224.1.0.0-224.1.255.255 (224.1/16))
  - SDP/SAP Block (224.2.0.0-224.2.255.255 (224.2/16))



# Assigning Protocol Number

- IANA assigns IP protocol numbers
  - TCP has IP protocol number 6
  - UDP is 17
  - PIM messages have IP protocol number 103
- IANA port numbers
  - ssh is 22
  - echo is 7
  - telnet 23
  - PIM over Reliable Transport: is a “congestion-control” modification for JP messages
    - pim-port 8471 tcp
    - pim-port 8471 sctp



# Useful Links

- IETF Datatracker
  - The IETF Datatracker is a web-based system for managing information about Internet-Drafts (I-Ds), RFCs and several other important aspects of the IETF process
  - <http://datatracker.ietf.org/>
- IETF TOOLS team
  - The purpose of the TOOLS team is to provide IETF feedback and guidance during the development of software tools to support various parts of IETF activities
  - <http://tools.ietf.org/>



# Internet Group Management Protocol (IGMP)

- IGMP v3: RFC 3376 (2002)
- IGMP v2: RFC 2236 (1997)
- IGMP v1 alias Host Extensions for IP Multicasting v3: RFC 1112 (1989)
- Obsoletes: RFCs 988, 1054
  - Host Extensions for IP Multicasting v2: RFC 1054 (1988)
  - Host Extensions for IP Multicasting v1: RFC 988 (1986)
- Host and router exchange of multicast group info
- Use broadcast LAN to transfer info among multiple hosts and routers



# Principle of Operations

- Hosts send messages to routers to subscribe to and unsubscribe from multicast group
  - Group defined by multicast address
- Routers check which multicast groups of interest to which hosts
- IGMP currently version 3
- IGMPv1
  - Hosts could join group
  - Routers used timer to unsubscribe members



# Operation of IGMP v1 & v2

- Receivers have to subscribe to groups
- Sources do not have to subscribe to groups
- Any host can send traffic to any multicast group
- Problems:
  - Spamming of multicast groups
  - Even if application level filters drop unwanted packets, they consume valuable resources
  - Establishment of distribution trees is problematic
  - Location of sources is not known
  - Finding globally unique multicast addresses difficult

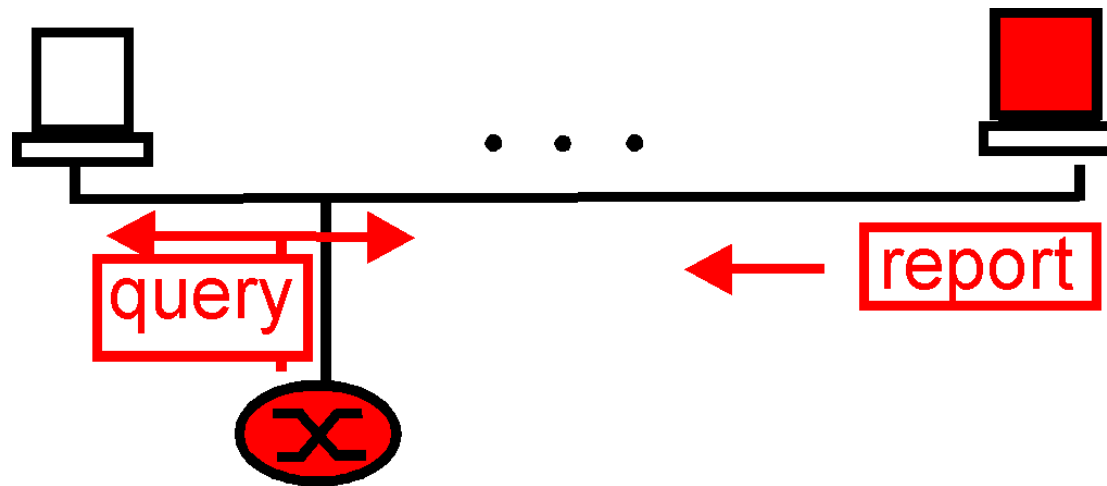


- Allows hosts to specify list from which they want to receive traffic
  - Traffic from other hosts blocked at routers
- Allows hosts to block packets from sources that send unwanted traffic
- IGMPv3 itself runs on multicast: address 224.0.0.22



# IGMP dialogues

- IGMP (Internet Group Management Protocol) operates between Router and local Hosts, typically attached via a LAN (e.g., Ethernet)
- Router queries the local Hosts for m-cast group membership info



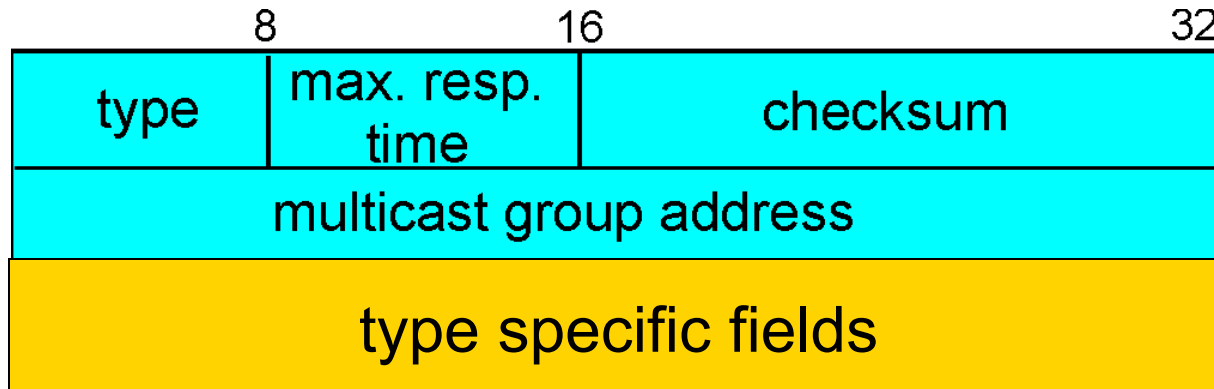
# IGMP Protocol

- Router “connects” active Hosts to m-cast tree via m-cast protocol
- Hosts respond with membership reports: actually, the first Host which responds (at random) speaks for all
- Host issues “leave-group” msg to leave; this is optional since router periodically polls anyway (soft state concept)

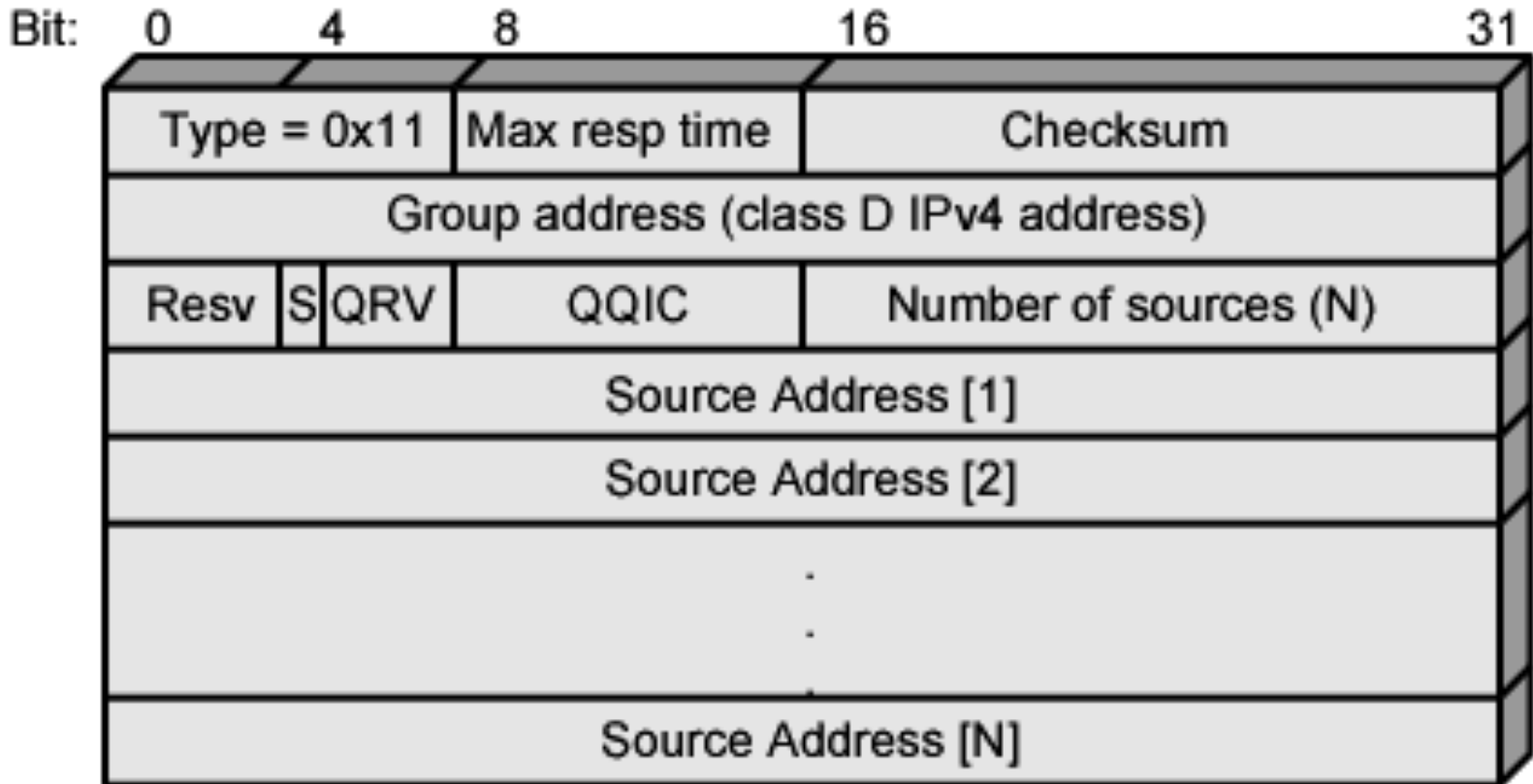


# IGMP message types

IGMP Message type	Sent by	Purpose
membership query: general	router	query for current active multicast groups
membership query: specific	router	query for specific m-cast group
membership report	host	host wants to join group
leave group	host	host leaves the group



# IGMP Message Formats: Membership Query



(a) Membership query message

# Membership Query

- Sent by multicast router
- General query
  - Which groups have members on attached network
- Group-specific query
  - Does group have members on an attached network
- Group-and-source specific query
  - Do attached device want packets sent to specified multicast address
  - From any of specified list of sources



# Membership Query Fields (1)

- Type
- Max Response Time
  - Max time before sending report in units of 1/10 second
- Checksum
  - Same algorithm as IPv4
- Group Address
  - Zero for general query message
  - Multicast group address for group-specific or group-and-source
- S Flag
  - 1 indicates that receiving routers should suppress normal timer updates done on hearing query

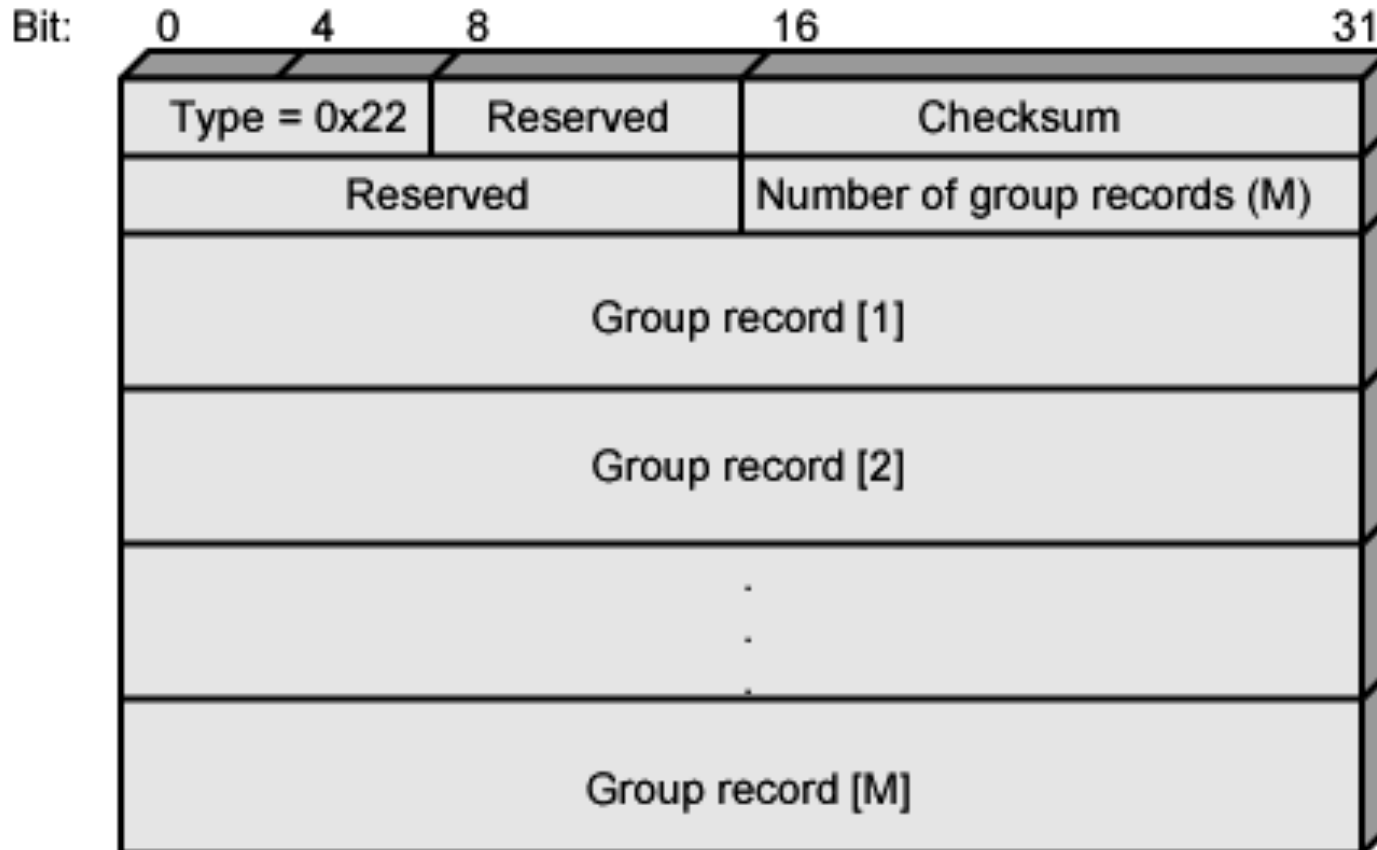


# Membership Query Fields (2)

- QRV (querier's robustness variable)
  - RV value used by sender of query
  - Routers adopt value from most recently received query
  - Unless RV was zero, when default or statically configured value used
  - RV dictates number of retransmissions to assure report not missed
- QQIC (querier's query interval code)
  - QI value used by querier
  - Timer for sending multiple queries
  - Routers not current querier adopt most recently received QI
  - Unless QI was zero, when default QI value used
- Number of Sources
- Source addresses
  - One 32 bit unicast address for each source



# IGMP Message Formats: Membership Report



(b) Membership report message

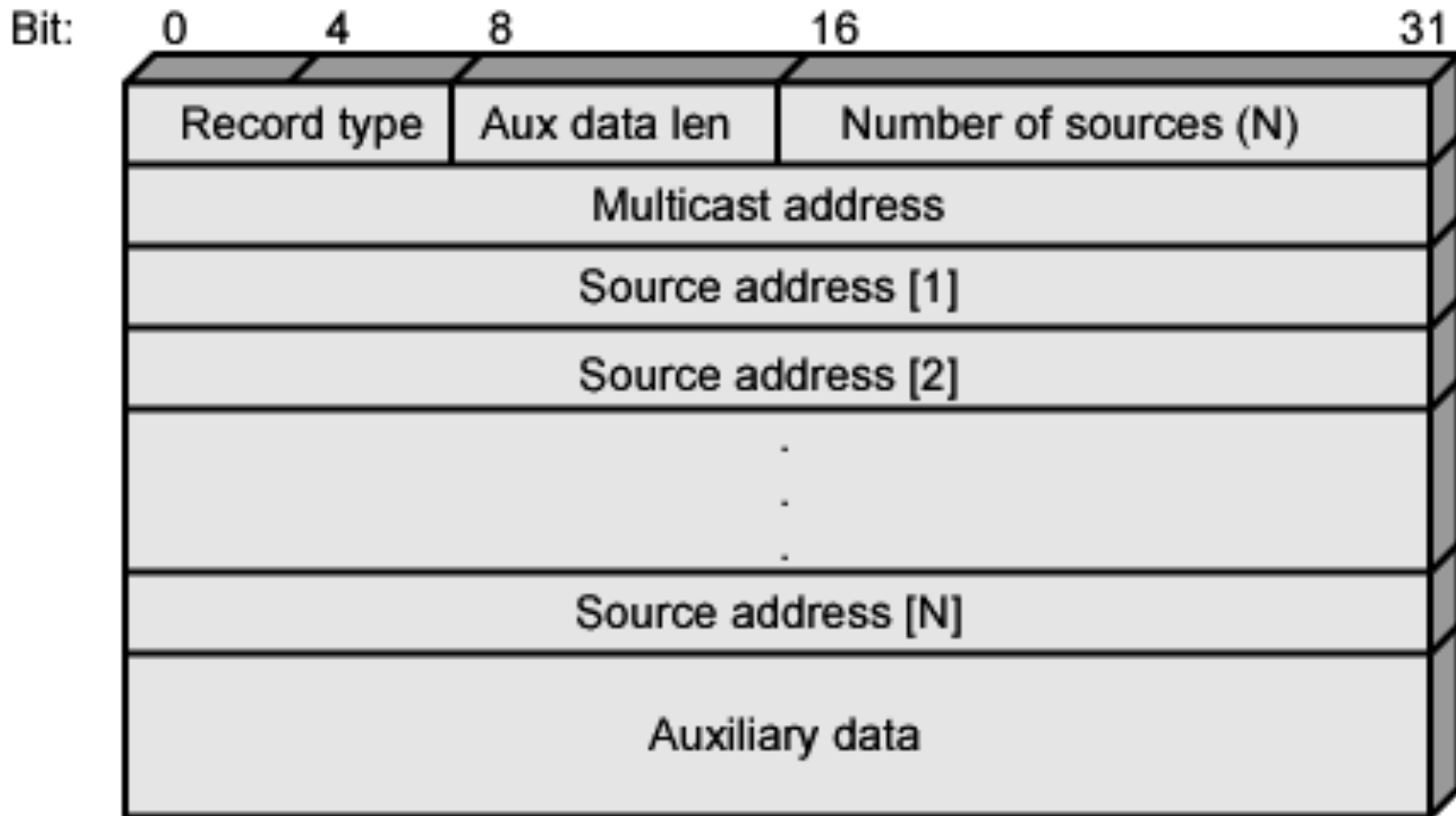


# Membership Reports

- Type
- Checksum
- Number of Group Records
- Group Records
  - One 32-bit unicast address per source



# IGMP Message Formats: Group Record



(c) Group record

# Group Record

- Record Type
  - **"Current-State Record"**
    - MODE\_IS\_INCLUDE (in response to a Query) INCLUDE()
    - MODE\_IS\_EXCLUDE EXCLUDE()
  - **"Filter-Mode-Change Record"**
    - CHANGE\_TO\_INCLUDE\_MODE (when the filter mode change) TO\_IN()
    - CHANGE\_TO\_EXCLUDE\_MODE TO\_EX()
  - **"Source-List-Change Record"**
    - ALLOW\_NEW\_SOURCES (when the source list change) ALLOW()
    - BLOCK\_OLD\_SOURCES BLOCK()
- Aux Data Length
  - In 32-bit words
- Number of Sources
- Multicast Address
- Source Addresses
  - One 32-bit unicast address per source
- Auxiliary Data
  - Currently, no auxiliary data values defined



# IGMP Operation - Joining

- Host using IGMP wants to make itself known as group member to other hosts and routers on LAN
- IGMPv3 can signal group membership with filtering capabilities with respect to sources
  - EXCLUDE mode – all group members except those listed
  - INCLUDE mode – Only from group members listed
- To join group, host sends IGMP membership report message
  - Address field contains the group multicast address
  - Sent in IP datagram with Group Address field of IGMP message and Destination Address encapsulating IP header same
  - Current members of group receive and learn about new member
  - Routers listen to all IP multicast addresses to hear all reports



# IGMP Operation – Keeping Lists Valid

- Routers periodically issue IGMP general query message
  - In datagram with all-hosts multicast address
  - Hosts that wish to remain in groups must read datagrams with this all-hosts address
  - Hosts respond with report message for each group to which it claims membership
- Router does not need to know every host in a group
  - Needs to know at least one group member still active
  - Each host in group sets timer with random delay
  - Host that hears another claim membership cancels own report
  - If timer expires, host sends report
  - Only one member of each group reports to router



# IGMP Operation - Leaving

- Host leaves group, by sending leave group message to all-routers static multicast address
- Send membership report message with EXCLUDE option and null list of source addresses
- Router determine if there are any remaining group members using group-specific query message



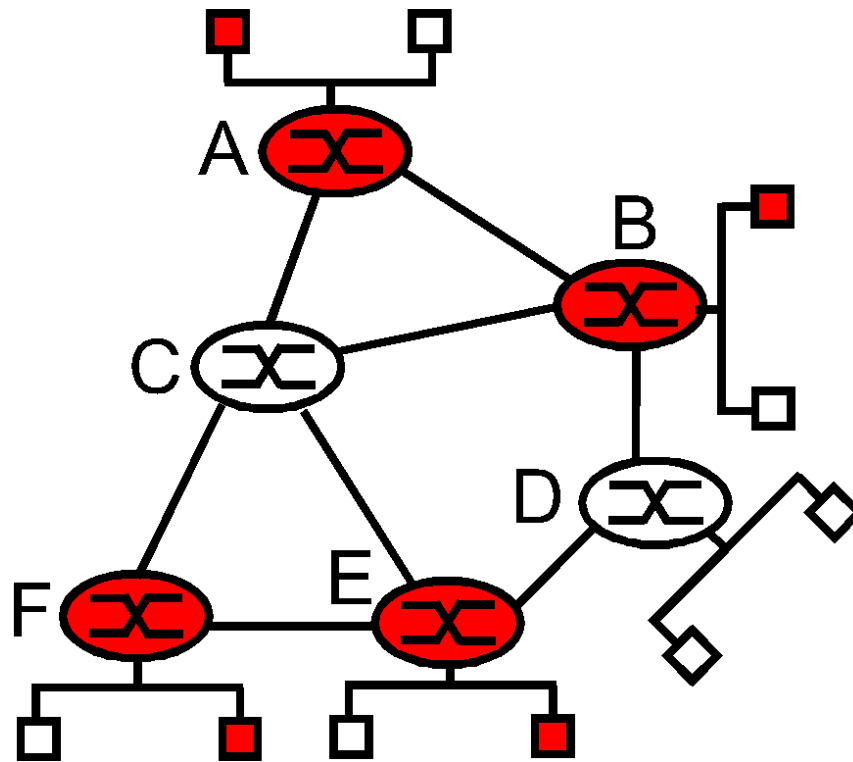
# Global distribution

- IGMP is complex, but solves simple problems:
  - local management of registrations
  - delivering of an IP multicast packet (normally UDP) to a number of hosts on a LAN via Ethernet
- The other problem is more complex and it is routing on the global infrastructure



# The Multicast Tree problem

- Problem: find the best (e.g., min cost) tree which interconnects all the members





# Steiner tree problem

- Given a graph  $G = (V, E, w)$ 
  - G connected and undirected
  - Weight function  $w: E \rightarrow R$
  - FIND  $G_{ST} = (V_{ST}, E_{ST}, w)$ 
    - $T \subseteq V_{ST}$  multicast set
    - $w(G_{ST}) = \sum_{(i,j) \in E_{ST}} w(i, j)$  is minimum
  - $T - V_{ST}$  are called Steiner nodes
  - The Steiner Tree (ST) Problem is NP-Complete, i.e.,
    - It is an NP Problem (solution can be verified in polynomial time)
    - And it is NP-hard (any NP problem may be converted into it)
  - Special case: when  $T=V \rightarrow$  Minimum Spanning Tree (MST), which can be solved in polynomial time



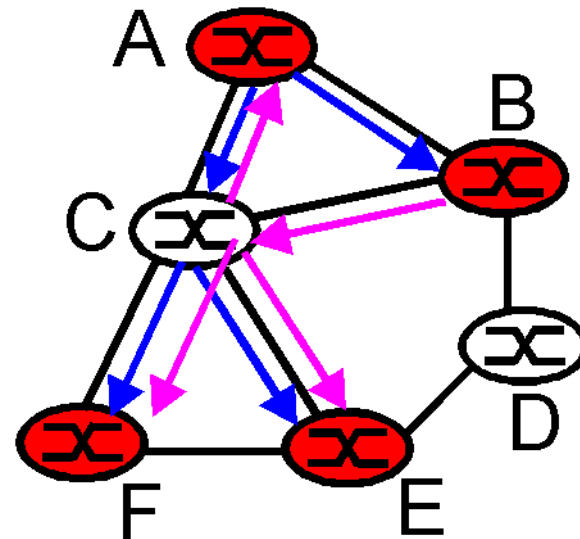
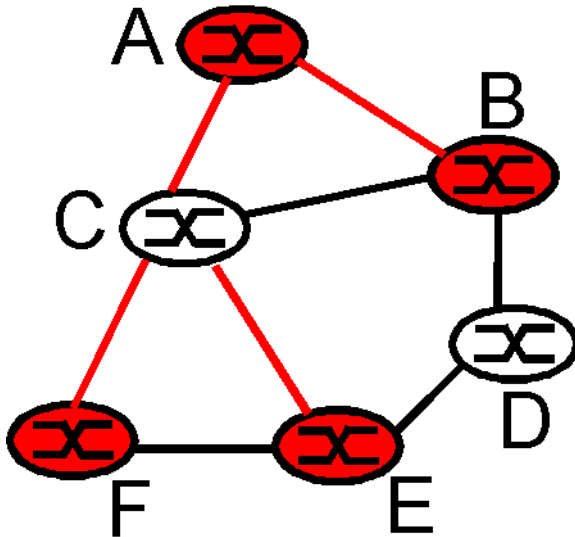
# Steiner Tree Problem

- The complexity comes from
  - The degree of freedom given by non-receiving routers
  - The requirement to find a single minimum tree for all possible sources in the m-cast group
- Limiting to 1 source the problem can be solved using inverse unicast routes
  - Routers must be able to understand the negotiation and duplicate packets accordingly
- With more sources one router can be elected as rendezvous point mapping the multi-source problem to the single source one



# Multicast Tree options

- **GROUP SHARED TREE:** just one spanning tree; the root is the “**CORE**” or the “**Rendez Vous**” point; all messages go through the **CORE**
- **SOURCE BASED TREE:** each source is the root of its own tree connecting to all the members; thus N separate trees



# Multicast Extension to OSPF (MOSPF)

- Enables routing of IP multicast datagrams within single AS
- Each router uses MOSPF to maintain local group membership information
- Each router periodically floods this to all routers in area
- Routers build shortest path spanning tree from a source to all networks containing members of group (Dijkstra)
  - Takes time, so on demand only



# Forwarding Multicast Packets

- If multicast address not recognised, discard
- If router attaches to a network containing a member of group, transmit copy to that network
- Consult spanning tree for this source-destination pair and forward to other routers if required



# Equal Cost Multipath Ambiguities

- Dijkstra's algorithm will include one of multiple equal cost paths
  - Which depends on order of processing nodes
- For multicast, all routers must have same spanning tree for given source node
- MOSPF has tiebreaker rule



# Inter-area Multicasting

- Multicast groups may contain members from more than one area
- Routers only know about multicast groups with members in its area
- Subset of area's border routers forward group membership information and multicast datagrams between areas
  - Inter-area multicast forwarders

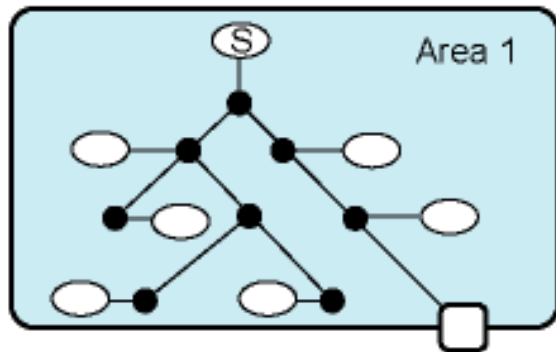


# Inter-AS Multicasting

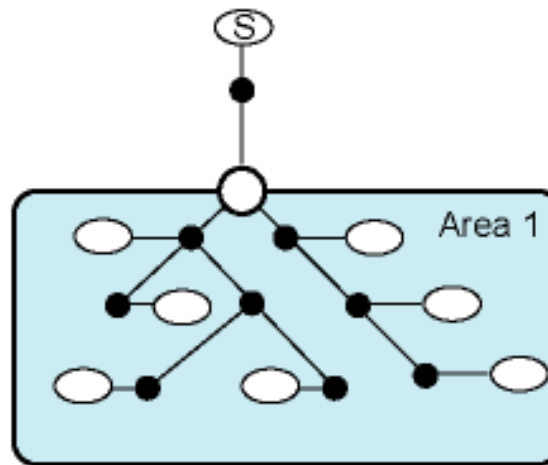
- Certain boundary routers act as inter-AS multicast forwarders
  - Run an inter-AS multicast routing protocol as well as MOSPF and OSPF
  - MOSPF makes sure they receive all multicast datagrams from within AS
  - Each such router forwards if required
  - Use reverse path routing to determine source
    - Assume datagram from X enters AS at point advertising shortest route back to X
    - Use this to determine path of datagram through MOSPF AS



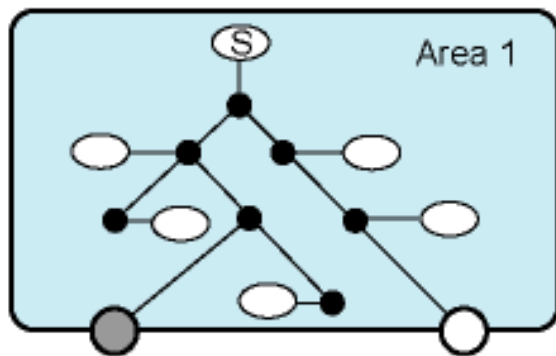




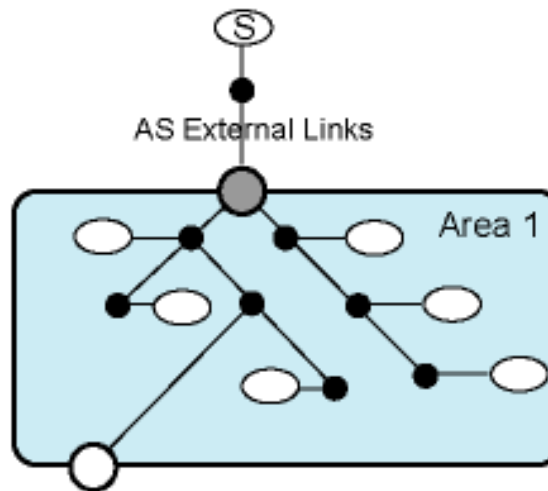
(a) Inter-Area Routing: Source in Same Area



(b) Inter-Area Routing: Source in Remote Area



(c) Inter-AS Routing: Source in Same Area



(d) Inter-AS Routing: Source in Different AS

- Ⓢ Source subnetwork
- Subnet containing group members
- Intra-area MOSPF router
- Inter-area multicast forwarder
- Inter-AS multicast forwarder
- Wild-card multicast receiver

# Multicast Routing Protocol Characteristics

- Extension to existing protocol
  - MOSPF v OSPF
- Designed to be efficient for high concentration of group members
- Appropriate with single AS
- Not for large internet

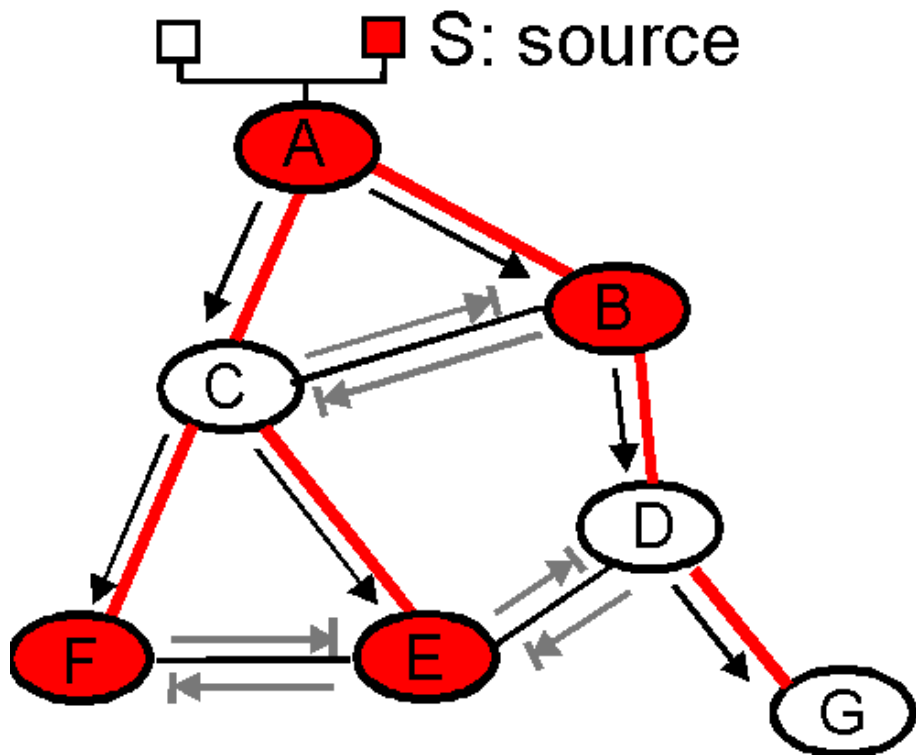


# Group Shared Tree





- Predefined **CORE** for given m-cast group (e.g., posted on web page)
- New members “join” and “leave” the tree with explicit join and leave control messages
- Tree grows as new branches are “grafted” onto the tree
- CBT (Core Based Tree) and PIM Sparse-Mode are Internet m-cast protocols based on GSTree
- All packets go through the **CORE**



# Group Shared Tree



## Legend

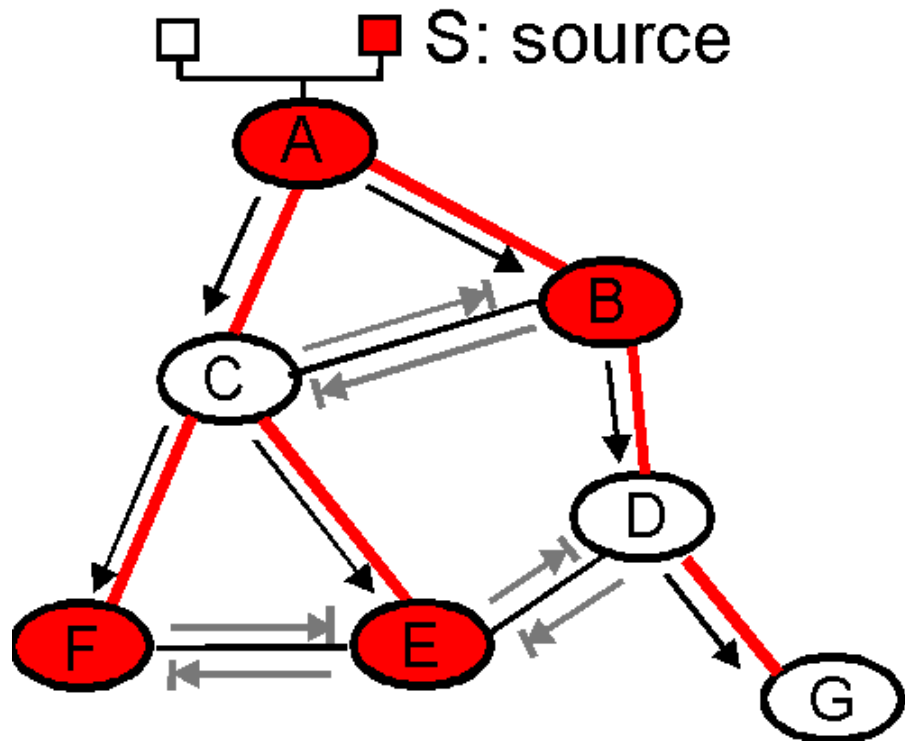
-  router with attached group member
-  router with no attached group member
-  pkt that will be forwarded
-  pkt not forwarded beyond receiving router

# Source Based Tree





- Each source is the root of its own tree: the tree of shortest paths
- Packets delivered on the tree using “reverse path forwarding” (RPF); i.e., a router accepts a packet originated by source  $S$  only if such packet is forwarded by the neighbor on the shortest path to  $S$
- In other words, m-cast packets are “forwarded” on paths which are the “reverse” of “shortest paths” to  $S$



# Source Based Tree



## Legend

-  router with attached group member
-  router with no attached group member
-  pkt that will be forwarded
-  pkt not forwarded beyond receiving router



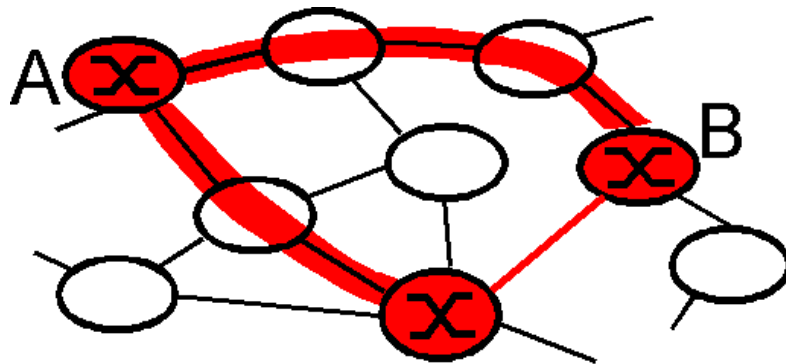
# Source-Based tree: DVMRP

- DVMRP was the first m-cast protocol deployed on the Internet; used in Mbone (Multicast Backbone)
- Initially, the source broadcasts the packet to ALL routers (using RPF)
- Routers with no active Hosts (in this m-cast group) “prune” the tree; i.e., they disconnect themselves from the tree

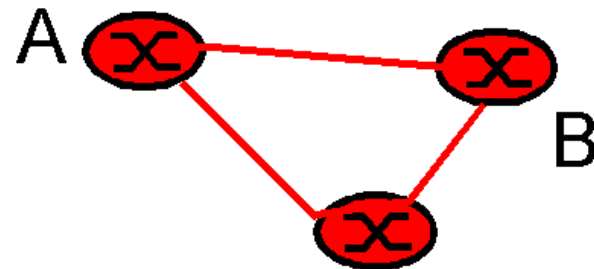


# Source-Based tree: DVMRP

- Recursively, interior routers with no active descendents self-prune. After timeout (2 hours in Internet) pruned branches “grow back”
- Problems: only few routers are mcast-able; solution: tunnels



physical topology



logical mcast topology



# PIM (Protocol Independent Multicast)

- Is becoming the de facto inter AS m-cast protocol standard
- “Protocol Independent” because it can operate on different routing infrastructures
  - Extract required routing information from any unicast routing protocol
  - Work across multiple AS with different unicast routing protocols
- PIM can operate in two modes:
  - Sparse Mode RFC 2362
  - Dense Mode RFC 3973



# PIM Strategy

- Flooding is inefficient over large sparse internet
- Little opportunity for shared spanning trees
- Focus on providing multiple shortest path unicast routes
- Dense mode
  - For intra-AS
  - Alternative to MOSPF
- Sparse mode
  - Inter-AS multicast routing



# PIM – Sparse Mode

- A sparse group:
  - Number of networks/domains with group members present significantly small than number of networks/domains in internet
  - Internet spanned by group not sufficiently resource rich to ignore overhead of current multicast schemes



# PIM – Sparse Mode

- For a group, one router designated rendezvous point (RP)
- Group destination router sends join message towards RP requesting its members be added to group
  - Use unicast shortest path route to send
  - Reverse path becomes part of distribution tree for this RP to listeners in this group
- Node sending to group sends towards RP using shortest path unicast route
- Destination router may replace group-shared tree with shortest path tree to any source
  - By sending a join back to source router along unicast shortest path
- Selection of RP dynamic
  - Not critical



# PIM – Sparse Mode

- Initially, members join the “Shared Tree” rooted on a Rendez Vous Point
  - Join messages are sent to the RP, and routers along the path learn about the “multicast” session, creating a tree from the RP to the receivers
  - The source send the packet to the router which encapsulate such a packet in a unicast message towards the RP. The RP unpack the message and send the packet on the tree.
- Later, once the “connection” to the shared tree has been established, opportunities to connect **DIRECTLY** to the source are explored (thus establishing a partial Source Based tree)
  - e.g., load exceeds the forwarding threshold



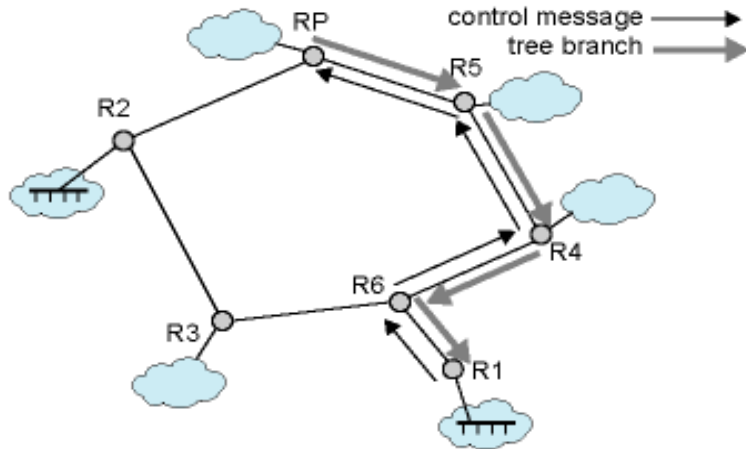
# Group Destination Router

# Group Source Router

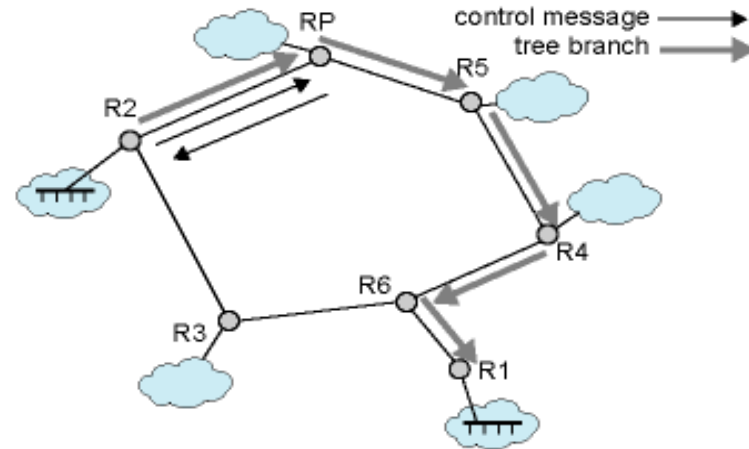
- Group Destination Router
  - Has local group members
  - Router becomes destination router for given group when at least one host joins group
    - Using IGMP or similar
- Group source router
  - Attaches to network with at least one host transmitting on multicast address via that router



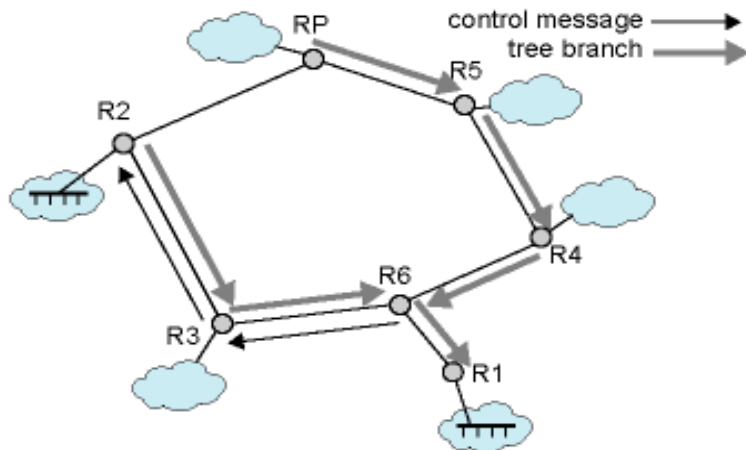
# Example of PIM Operation



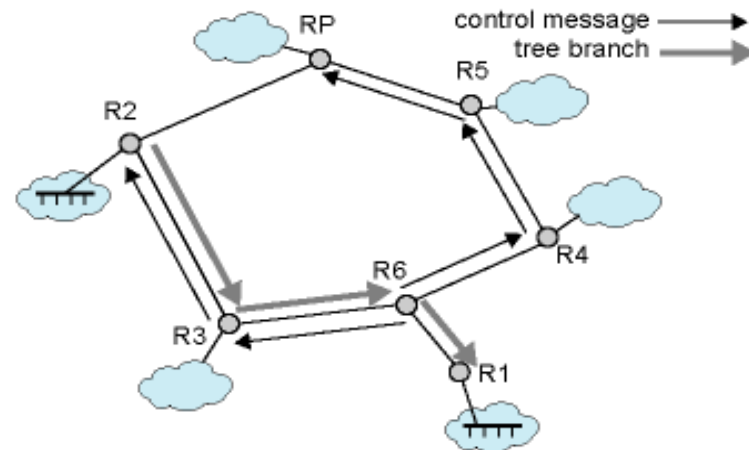
(a) R1 sends Join toward RP; RP adds path to distribution tree



(b) R2 sends Register to RP; RP returns Join; R2 builds path to RP



(c) R1 sends Join to R2; R2 prunes path to RP



(d) R6 sends Prune to RP; RP prunes path to R1

# PIM – Dense Mode

- A dense group:
  - Designed for wired-static networks
  - Routers are close to each other
  - Dense clients within an area (e.g., an organization)
  - Flood and Prune Protocol
  - Heavily use of Timers





# PIM – Dense Mode

- For a group there is one or many sources
- Each source floods data towards each interface
- Routers check whether there are nodes/routers on interfaces interested in that multicast group
  - Forward packets towards such interfaces except the RPF
  - Otherwise send a prune to the RPF' (i.e., next hop on the RPF), actually leaving the group



# PIM – Dense Mode

- Routers get/keep in touch through Hello msgs
- Routers send
  - **Prune** to leave a group: no clients on that group;
  - **Join** to receive again data from that group after a Prune;
  - **Graft** when the RPF' changes or a client joins a pruned group
  - **GraftAck** to ack a graft on downstream interfaces



# PIM – Dense Mode

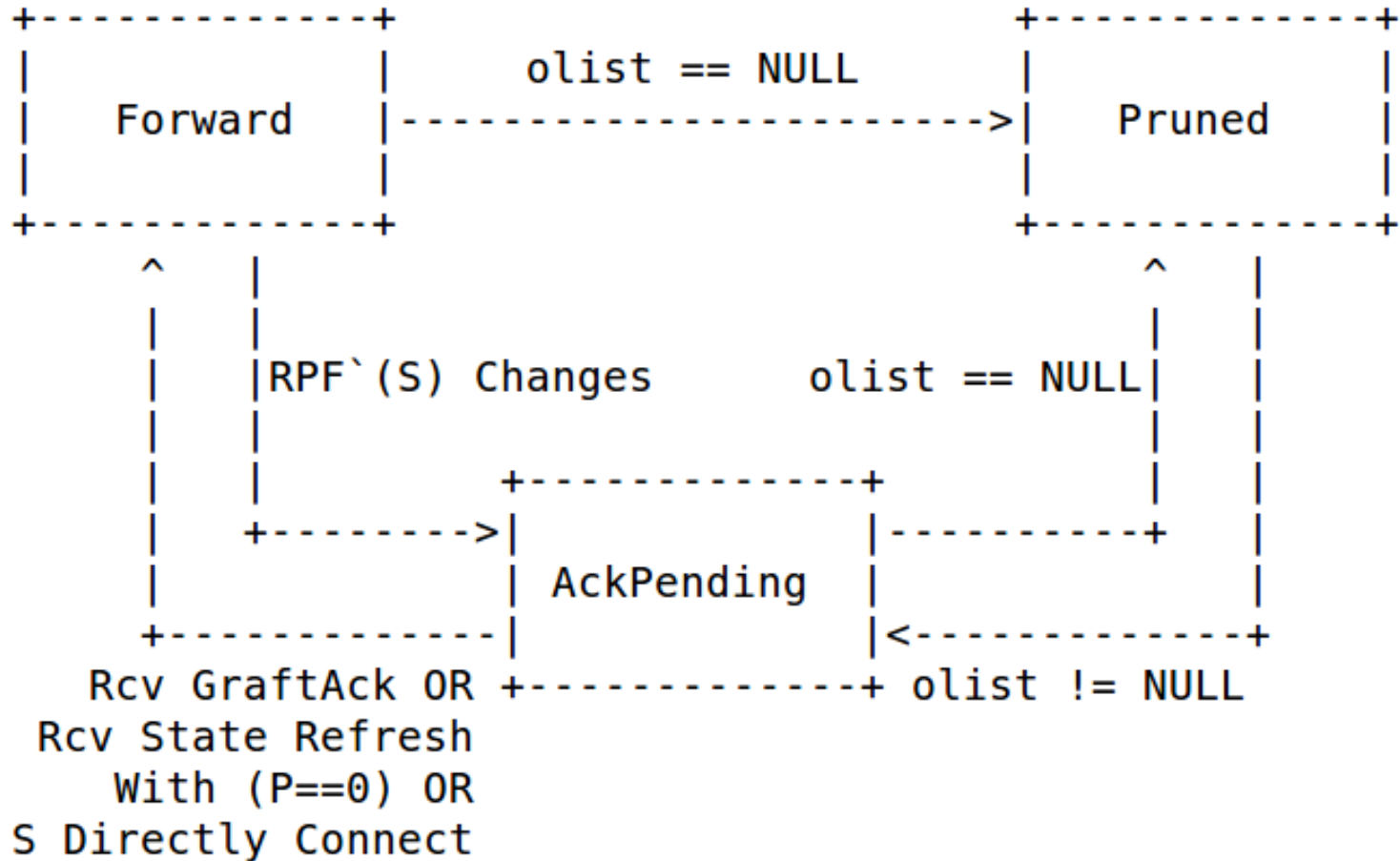


Figure 1: Upstream Interface State Machine

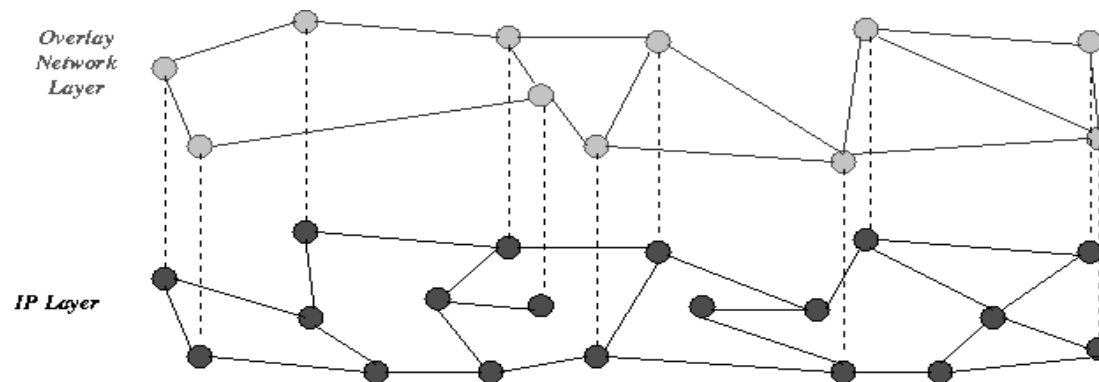
# Sparse vs. Dense Mode

- RP must be configured
- Explicit join
- Traffic flows to where it's needed
- Just routers along paths keep the state
- Scales better than DM
- Flood and Prune -> congestion?
- Routers must keep (S,G) state information
- Routers negotiate traffic forwarding: assert msgs
- More reliable on dynamic network:
  - Routers knows all (S,G)
  - No RP constraints



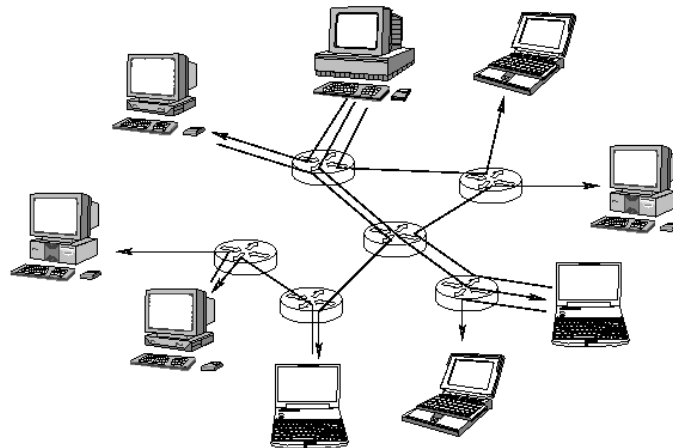
# Application-Level Multicast

- Multicast protocols are not supported by all routers
  - ISPs won't replace such routers
- Reproduce multicast at application level
  - Use Peer-to-peer paradigm among nodes interested in the multicast session
  - Build a logical multicast-overlay over the physical network
  - Nodes have a small view of the network: they know few nodes called neighbors



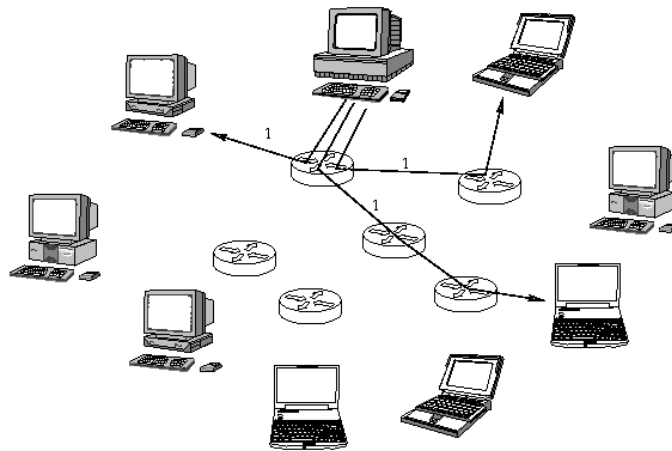
# Application-Level Multicast

- The source sends packets towards a few neighbors
- Nodes forward data received to their neighbors
  - Exchanging messages about data received, avoiding duplicates
  - Two main strategies: Push and Pull
    - PUSH: nodes send data towards neighbors → sender oriented
    - PULL: nodes request data from neighbors → receiver oriented



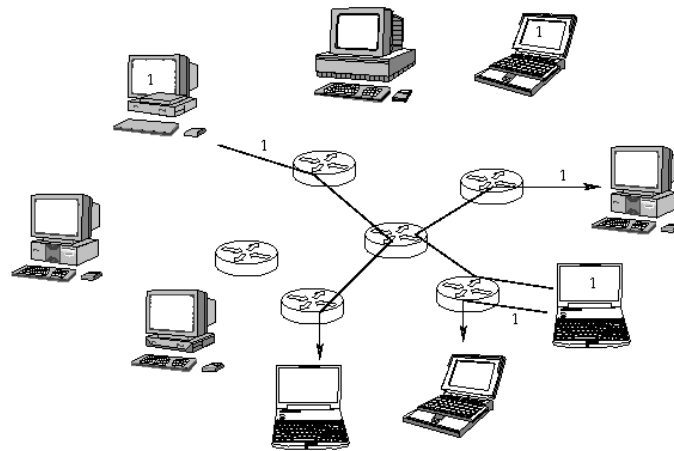
# Example: Following One Chunk

- Source encodes the first piece of data
- It issues three copies of chunk number 1 to three neighbors
- Three nodes have the chunk 1 at the end of cycle 1



# Example: Following One Chunk

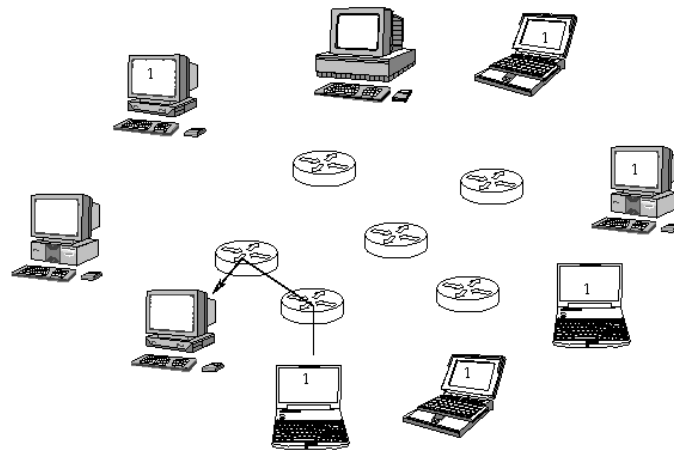
- Three node has the chunk number 1
- Each node tries to PUSH the chunk to some neighbor
- In cycle 2 there are  $3 + 3 = 6$  nodes with chunk 1





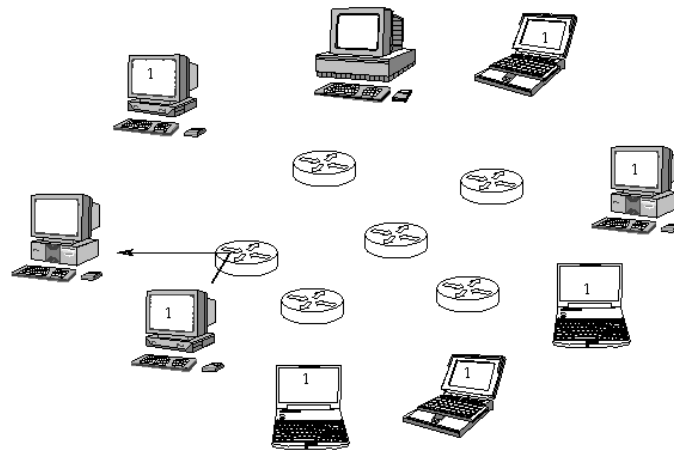
# Example: Following One Chunk

- The chunk number 1 is still forwarded towards some neighbors
- At the end of cycle three, there are 7 nodes have that chunk



# Example: Following One Chunk

- The forwarding process for chunk 1 ends
- The last node receives chunk number 1
- After 4 cycles all the nodes got the chunk
- Remember: it's an example! It's not a bound!!



# Example: Following One Chunk

- Some questions:
  - Are chunks path always the same?
  - Is the chunks receiving order correct?
  - Are links stable?
  - Is video streaming delay sensitive?
  - But that's another story...

