Multicast Protocols

IGMP – IP Group Membership Protocol

DVMRP – DV Multicast Routing Protocol

MOSPF – Multicast OSPF

PIM – Protocol Independent Multicast

Multicast in local area networks

Multicast addresses

IGMP – Internet Group Membership Protocol

Multicast addresses

32 bits

MSB(t)	network	host	Class
1110	**28 bits - multicast group address**		**D**
1111	**experiments**		**E**

224.0.0.0 - 239.255.255.255

224.0.0.1	All systems
224.0.0.2	All routers
224.0.0.4	All DVMRP routers
224.0.0.1 - 224.0.0.255	Local segment usage only
239.0.0.0 - 239.255.255.255	Admin scoped multicast (local significance)
239.192.0.0 - 239.195.255.255	Organization local scope

- Sender does not need to belong to G.
- Address space is flat.

Multicast in broadcast networks

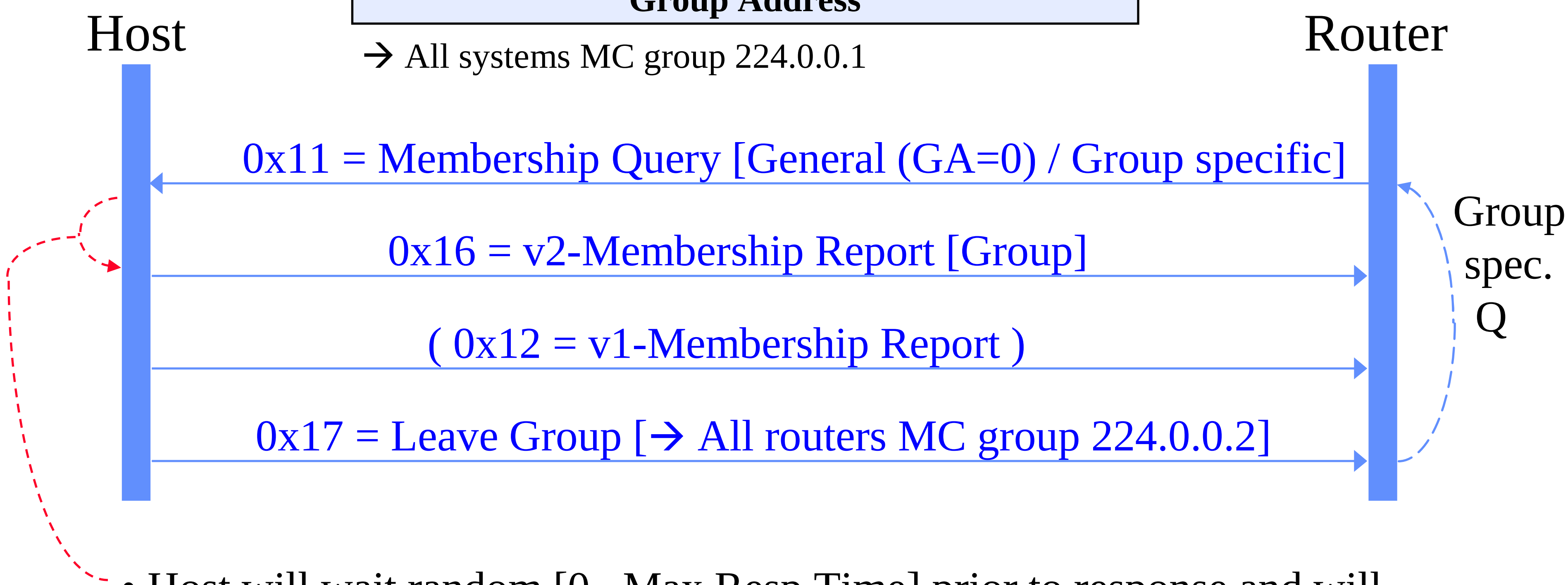
- In broadcast networks only one copy should be sent of a multicast packet
- Some broadcast network support group addresses
  - E.g. Ethernet
  - Group address is based on the IP address
    - Place low-order 23 bits of multicast address into low-order 23 bits of MAC address 01-00-5E-00-00-00
    - No ARP required
- Point-to-point links need no special arrangements

Routers discover multicast receivers using IGMP

- IGMP = Internet Group Membership Protocol
- Version 2 defined in RFC-2236
- Runs directly over IP (protocol type 2)
- Used locally within a network
  - TTL=1 in all IGMP messages
- Router with lowest IP address is active on a network
- Routers do not need to know the exact members, only whether there are members for a specific group

IGMPv2 - Internet Group Management Protocol implements Group Membership

Ver=2	Type	Max Resp Time	Checksum
Group Address			



- Host will wait random [0...Max Resp Time] prior to response and will suppress its response if it sees another response to the same group

IGMPv3 adds selective reception from sources within a group

Host

Router

Membership Query

Ver=3	0x11	Max Resp Time	Checksum
Group Address			
Reserved		Number of Sources (N)	
Source Address			
Source Address			
Source Address			

Variants:

- General query: GA=0 and number of sources=0.
- Group specific query: GA=/=0, number of sources=0
- Group and source specific Query

0x22 = V3 Membership Report

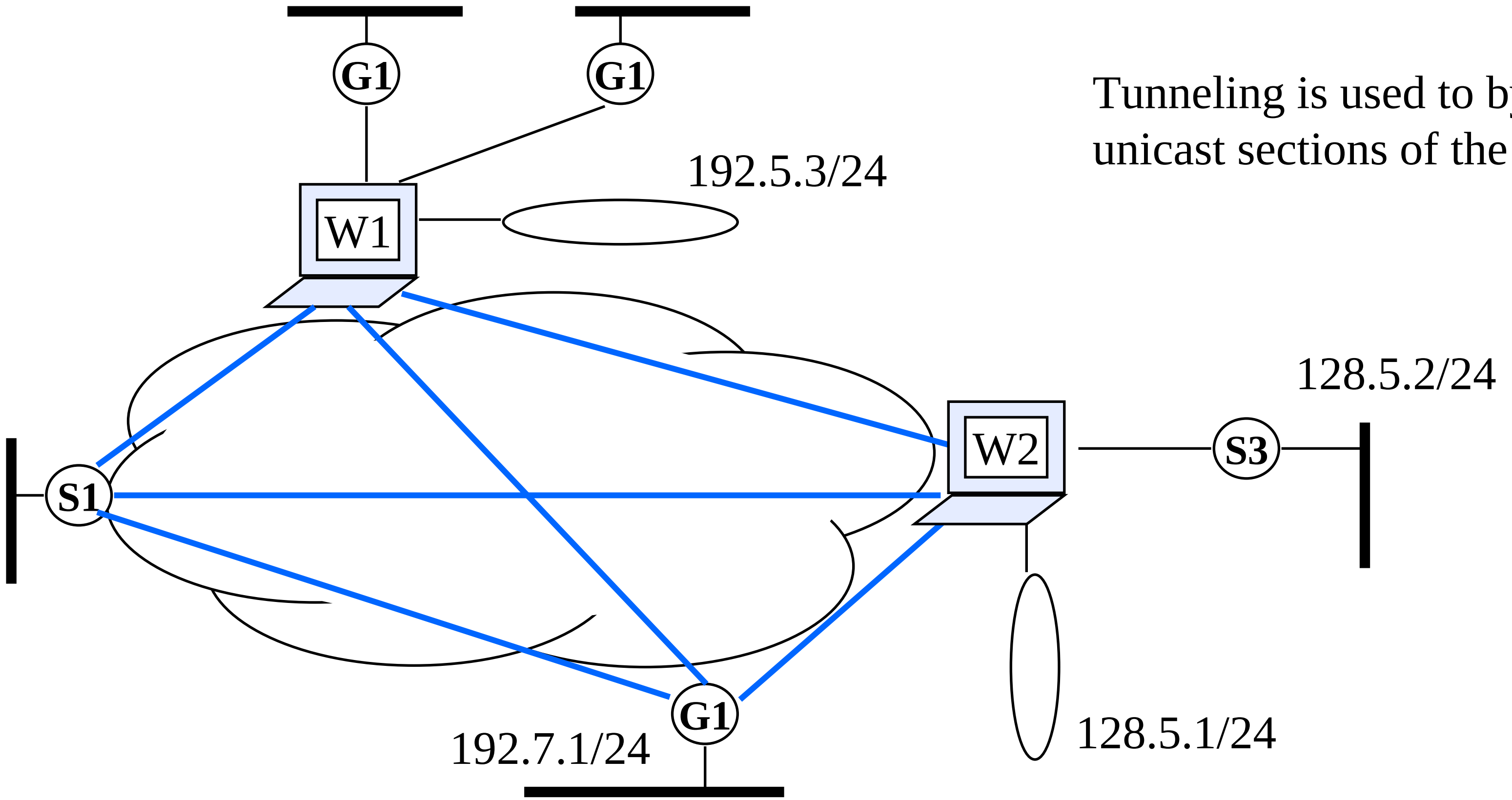
Can exclude listed sources within a group or include only listed sources within a group

MBone

MBone – an overlay multicast Internet

- Multicast backbone (MBone) was deployed to support research
  - Enable multicast applications without waiting for full availability of multicasting standards
- Started in 1992
- Uses tunnels to link multicast islands
  - Previously as source routed packet
  - Now with encapsulation
- Uses DVMRP and IGMP

MBone overlay is based on workstations running DVMRP



Tunneling is used to bypass unicast sections of the Internet

Experimental routing protocols have been developed for MBone

Tree type	Shared tree	Source based trees	
Algorithm	Center based tree	Flood and prune	Domain-wide reports
Protocols	PIM Sparse* Core Based Tree*	DVMRP PIM Dense*	MOSPF

* Relies on unicast routing protocol to locate multicast sources.

- Those that don’t, can route multicast on routes separate from unicast routes.

DVMRP – Distance Vector Multicast Routing Protocol

DVMRP – Distance Vector Multicast Routing Protocol

- First multicast protocol in the Internet (1988)
- Distance vector routing protocol similar to RIP
  - Except that sources are like destinations in RIP
- Routers maintains *separate multicast routing tables*
- Uses the *reverse-path-forwarding (RPF) algorithm*
- Nodes exchange
  - Distance in hops (reverse path distance)
  - IP address and mask of source
- Tunnels explicitly configured with
  - Destination router
  - Cost
  - Threshold

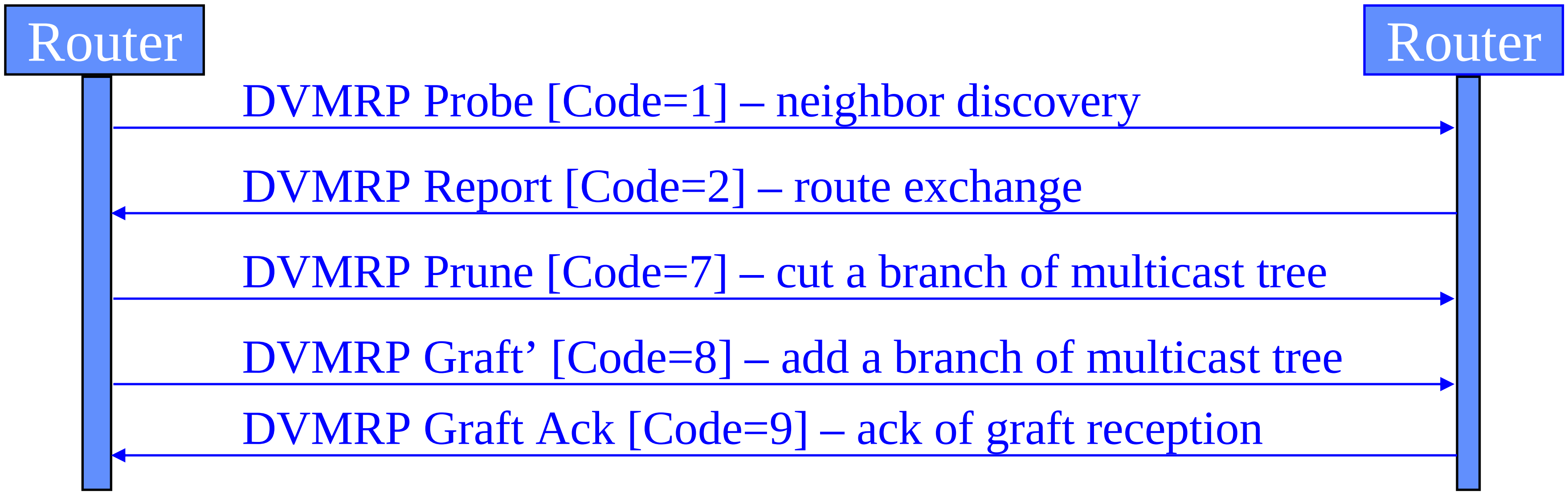
DVMRP is used for multicast routing in the MBone

- DVMRP messages are IGMP messages (IP protocol=2=IGMP, TTL=1)

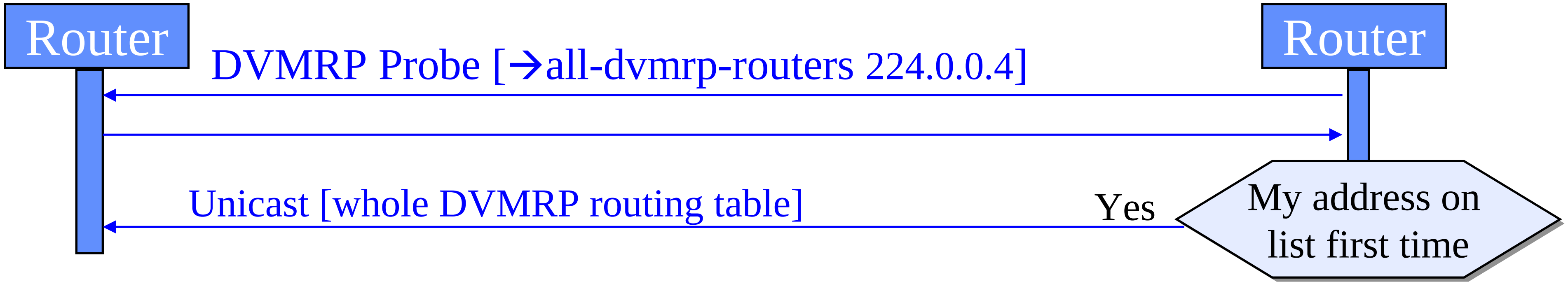
DVMRP header:

Type=0x13	Code	Checksum	
Reserved		Minor vers =0xff	Major vers = 3

Version 3 (1997) presented in this course



Probes are used for neighbor discovery



- Probes are exchanged on tunnel and physical interfaces
- Contains the list of neighbors on the interface
  - If empty, this is leaf network managed by IGMP
- Multicasts are not exchanged until two-way neighbor relationship is established
- Routers see each others versions and capability flags ⇒ compatibility
- Keepalive ⇒ fault detection, restart detection
  - sent each 10s, timeout set at 35s

DVMRP uses the concept of dependent downstream routers

- DVMRP uses the route exchange as a mechanism for upstream routers to determine if any downstream routers depend on them for forwarding from particular source networks
  - Implemented with ”poison reverse”
  - If a downstream router selects an upstream router as the best next hop to a source, it echoes back the route with a metric = original metric + inf

Route reports are used to build the source based trees

Router — DVMRP Report → Router

Each DVMRP router periodically (60s) broadcasts to its neighbors
- the list of pairs (source, metric)
- source aggregation according to CIDR may be used

Known neighbor — yes

- The receiving MC router calculates the previous hop on each sources multicast path = the DVMRP router that reports shortest distance *from the source*
- If equal distance ⇒ choose smallest IP address

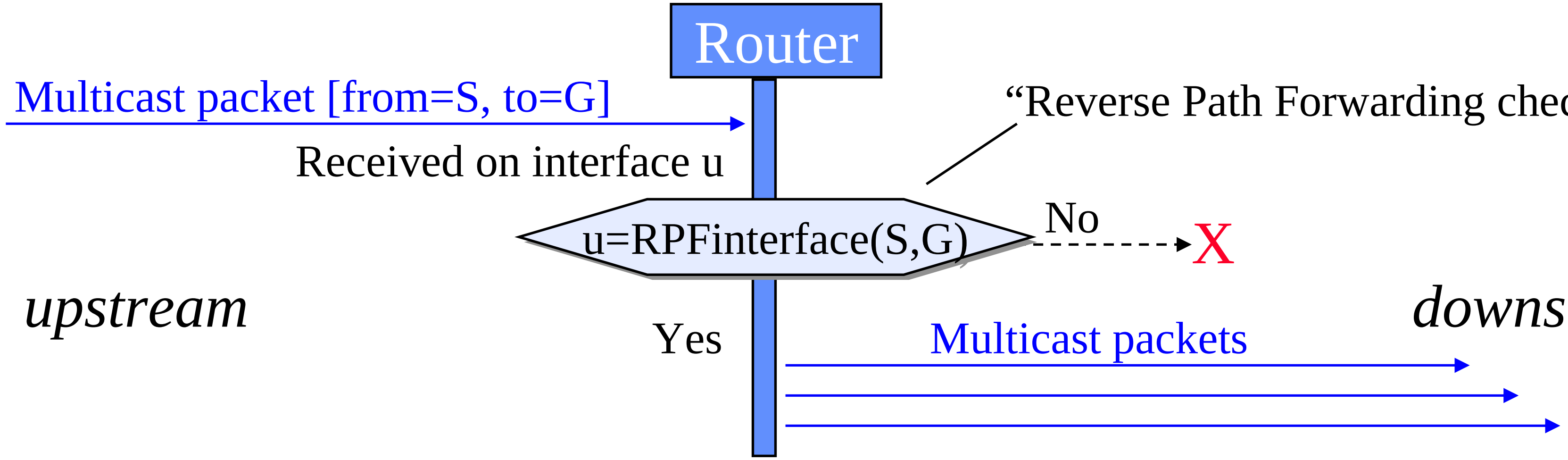
DVMRP Report [inf < metric < 2*inf]

(compare to poisonous reverse, inf=32)

Designated forwarder

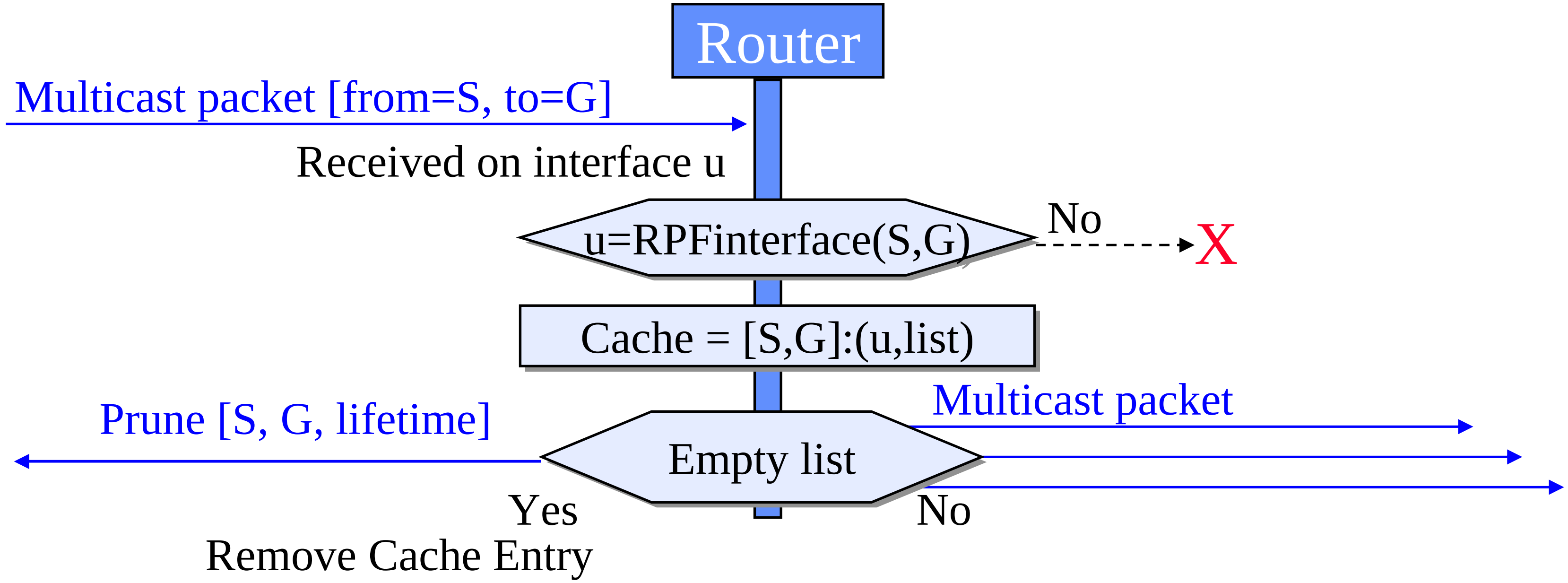
Downstream Dependent neighbor

The multicast algorithm of DVMRP is based on Reverse Path Forwarding (RPF)



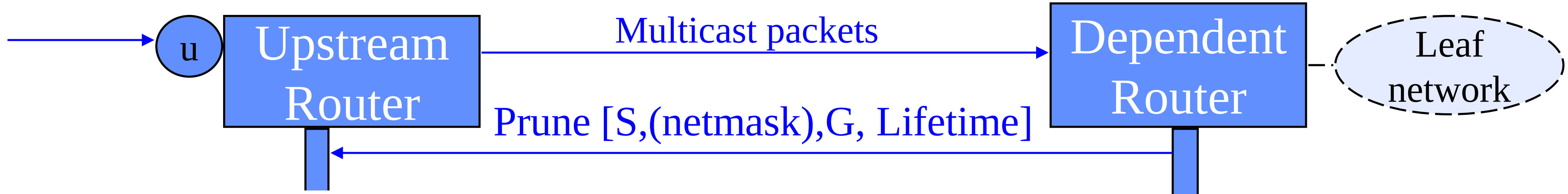
- At first multicast from RPF interface a Forwarding Cache Entry [S,G]:(u,list...) is created using the DVMRP routing table
  - The list contains all downstream routers that have reported dependency on S
- The router is designated forwarder for downstream nodes
- If the designated forwarder becomes unreachable, another router assumes the role of designated until it hears from a better candidate

List of dependent neighbors is used to minimize the multicast tree



- Initially list may contain all multicast interfaces but the upstream interface
- Downstream address is removed from list if
  - It is a leaf network and G is not in IGMP DB for this phys. network
  - Downstream node has selected another designated forwarder
  - Prune received from all dependent neighbors on this interface

Prunes minimize the multicast tree



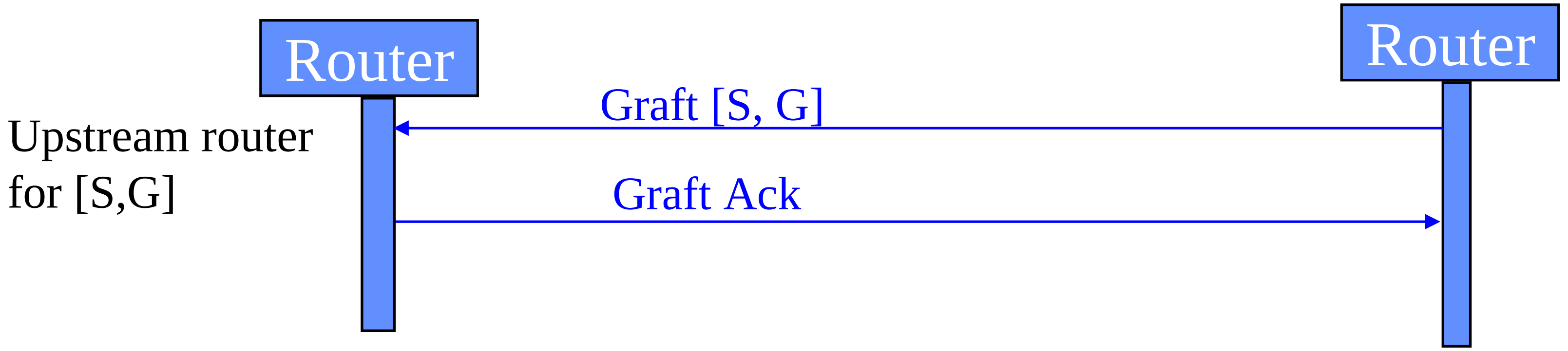
If known dependent neighbor
- If mask and mask=sent mask with (S,G)
  - Prune all sources in network (S, mask)
- If prune is already active
  - reset timeout to new value
- If all dependent neighbors have sent prunes
  - If no group members on the mc-interface
    - Remove u from all Forwarding Cache entries
    - If last u
      - Send prune

Prune[S,(m), G]

If Mcasts keep arriving (3s)
- Resend Prune with exponential backoff = double interval each time

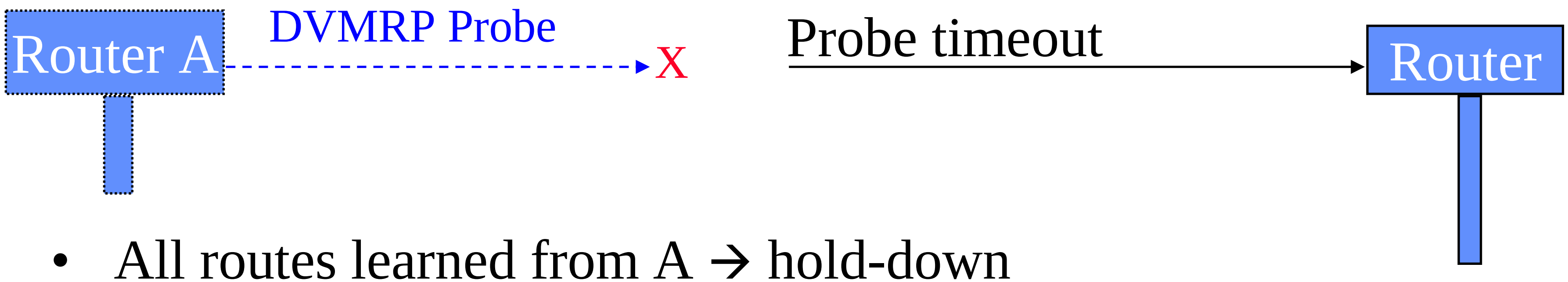
Remove Cache Entry

Grafts are used to grow the tree when a new member joins the group



- The graft is always acknowledged
  - if no multicast, nobody is sending
- If no ack is received, the graft is resent with exponential backoff retransmissions
- The graft is forwarded upstream if necessary

On probe timeout caches are flushed



- All routes learned from A → hold-down
- All downstream dependencies ON A are removed
- If A was designated forwarder, a new one is selected for each (source, group) pair
- Forwarding cache entries based on A are flushed
- Graft acks to A are flushed.
- Downstream dependencies are removed.
  - If last, send prune upstream

Route hold-down is a state prior to deleting the route

- Routes expire on report timeout or when an infinite metric is received
- An alternate route (that in RIP caused temporary loops) may exist
- Routers continue to advertise the route with inf metric for 2 report intervals – this is the hold-down period
- All forwarding cache entries for the route are flushed
- During hold-down, the route may be taken back, if
  - metric <inf, and
  - metric = SAME, and
  - received from SAME router

PIM – Protocol Independent Multicast

PIM – Protocol Independent Multicast

- Most popular multicast protocol
- Two modes of operation
  1. Dense mode
  2. Sparse mode
- Independent of any particular unicast routing protocol
- Uses unicast routing table

  ⇒ Simple protocol

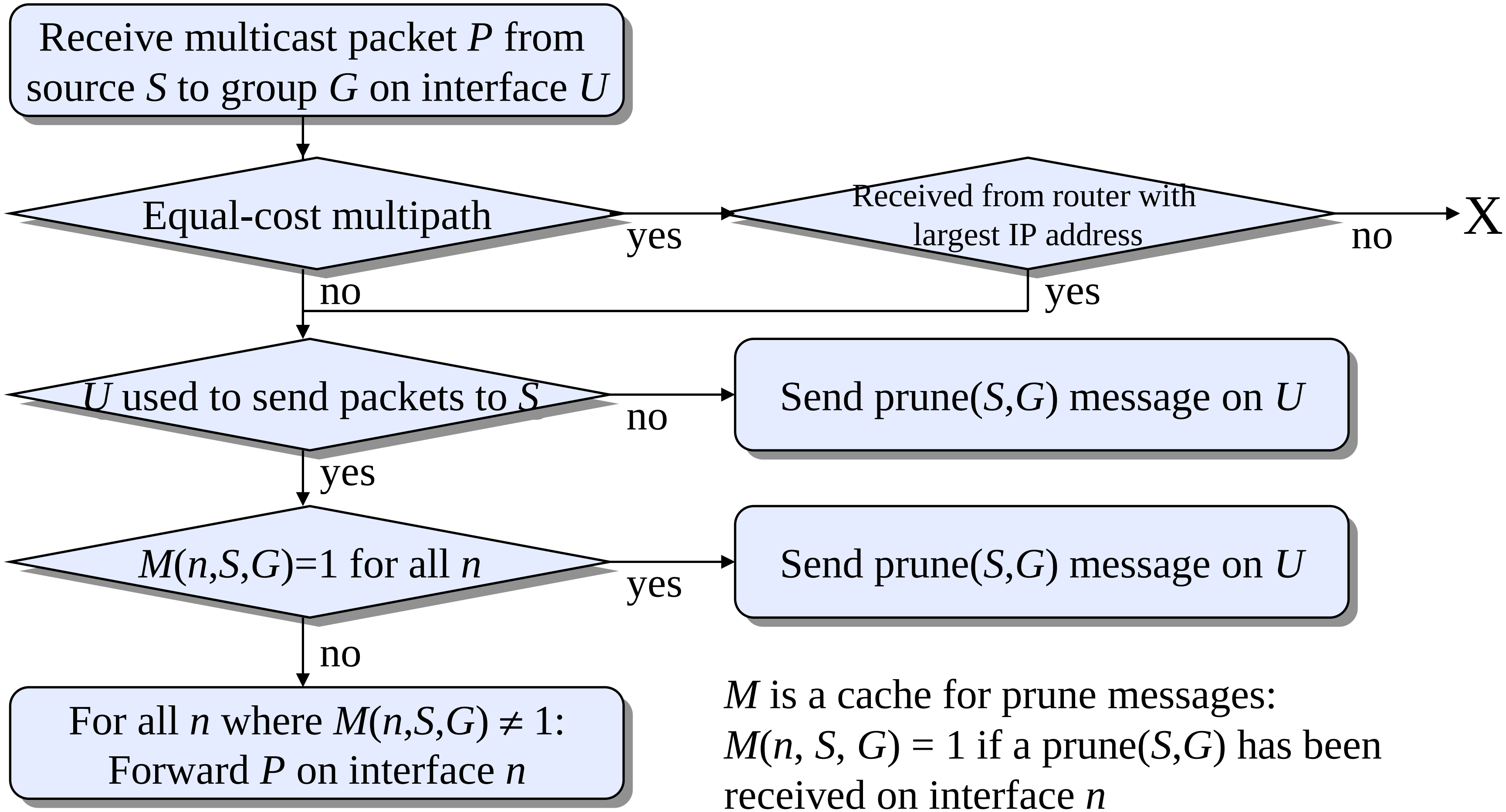
  ⇒ Assumes the links are symmetric

  ⇒ No tunnels
- Messages sent in IGMP packets

PIM Dense Mode

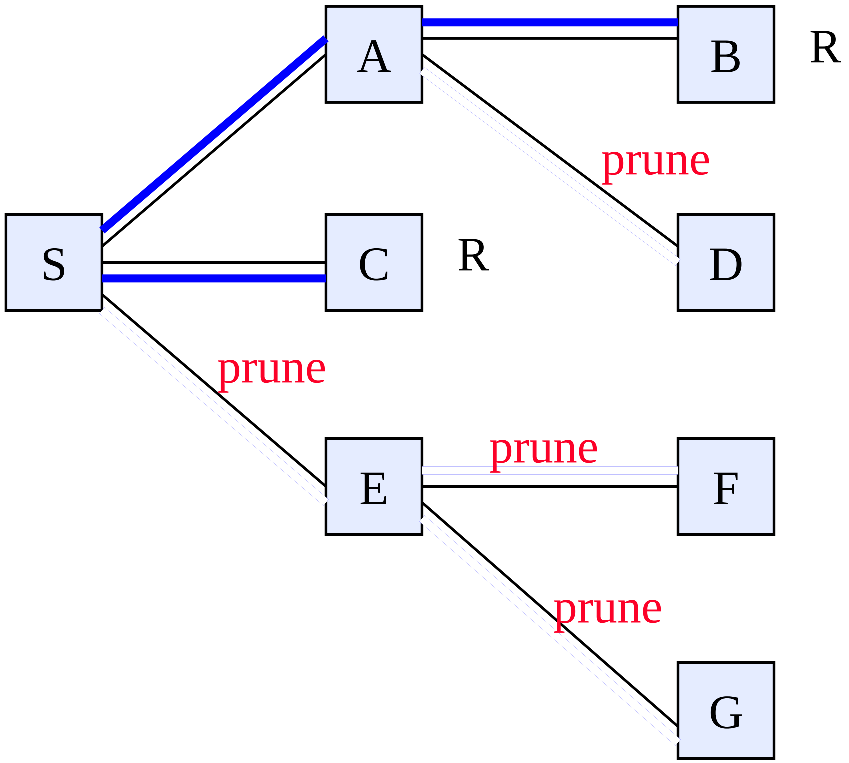
- For dense multicast groups
  - Dense: The probability is high that a small randomly picked area contains at least a group member, e.g. LAN
- Based on RPF / ”flood-and-prune”
- Principle similar to DVMRP
  - Simpler
  - Less efficient

PIM-DM implementation of RPF



M is a cache for prune messages:
$M(n, S, G) = 1$ if a prune(*S*,*G*) has been received on interface *n*
Least recently used entries may be dropped

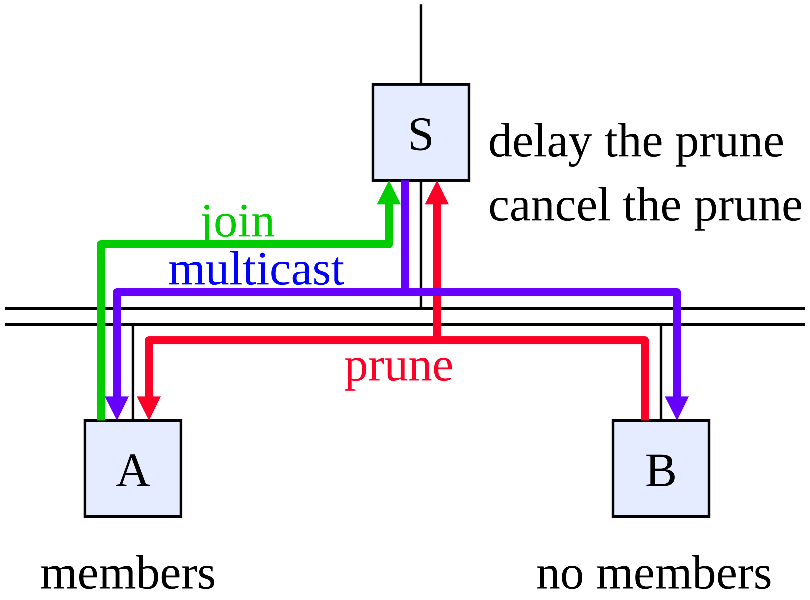
PIM-DM – Pruning



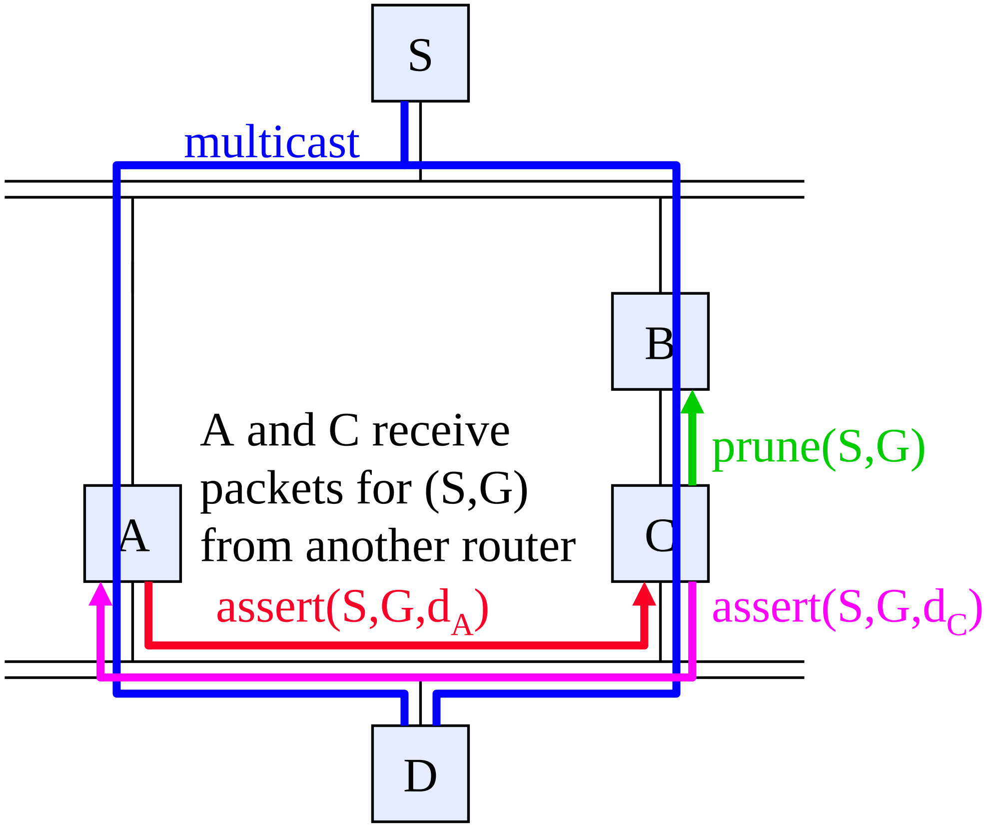
R = receiver

PIM-DM – Pruning on broadcast networks

- Prune messages sent to ”all-routers” (224.0.0.2)



PIM-DM – Resolving multicasts received on multiple path



if $d_{\text{other router}} < d_{\text{my distance}}$:
prune interface

PIM Sparse Mode

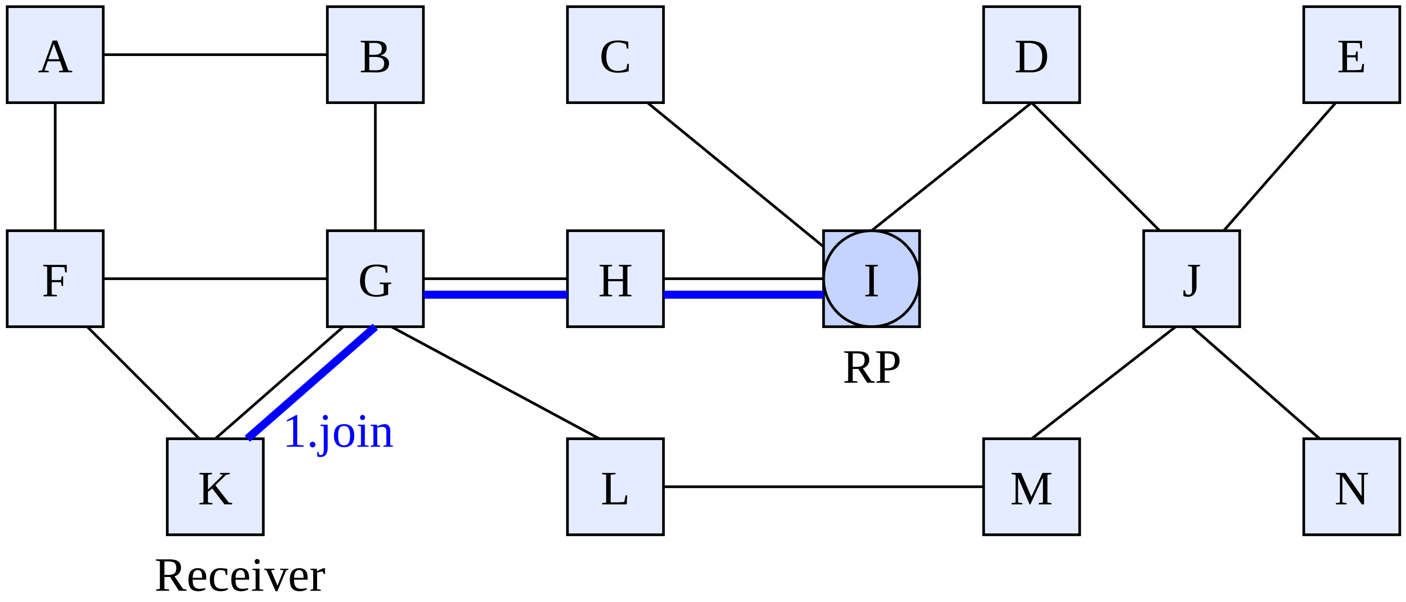
- RFC 2362
- Uses the center-based tree algorithm
- Evolved from the Core-Based Tree (CBT) protocol
- Rendezvous point (=center) connects the receivers with the senders
- Receivers must explicitly join

PIM-SM route entries

- Route entry includes
  - source address
  - group address
  - incoming interface
  - list of outgoing interfaces
  - timers, flags
- Packets match on the most specific entry
  - (S,G) – a specific source in a specific group
  - (*,G) – all sources in a specific group
  - (*, *, RP) – all groups that hash to a specific RP

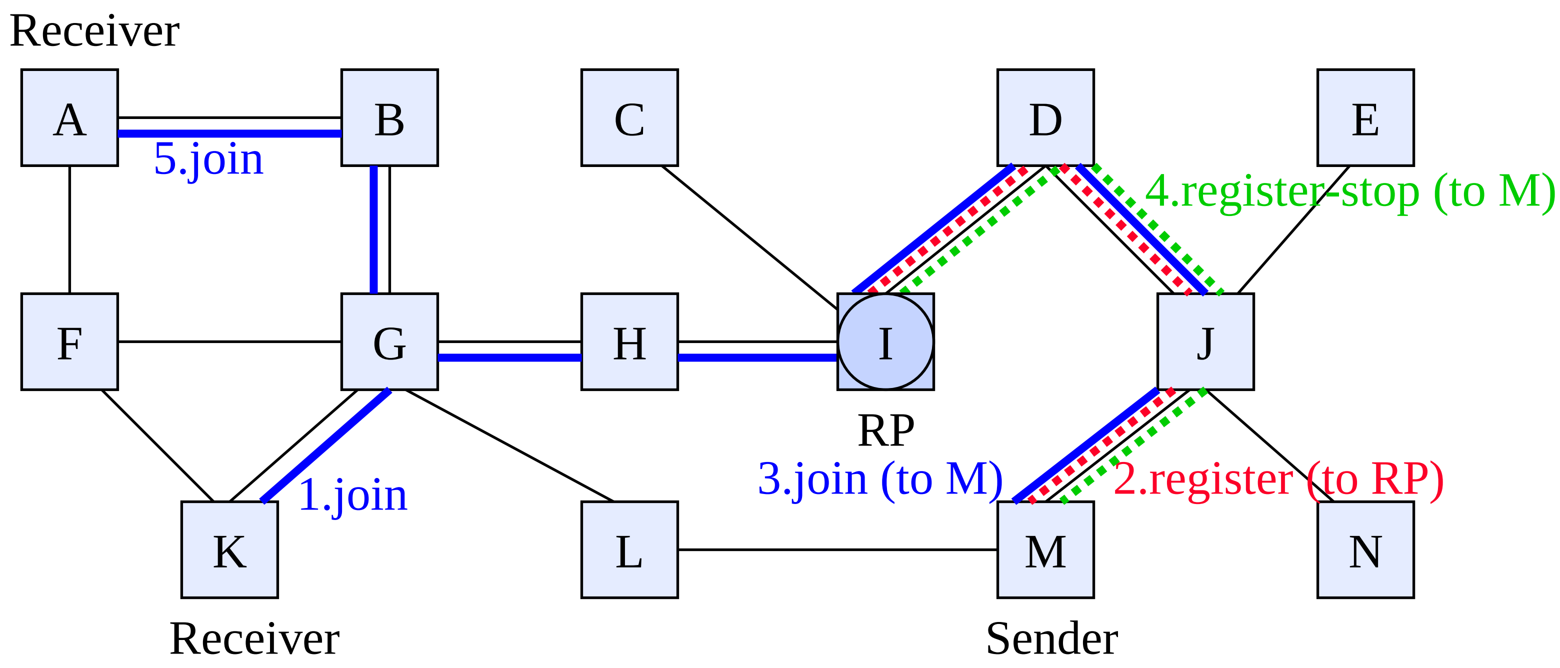
PIM-SM example (1)

- Join packets are sent toward the RP
  - Address=G, Join=RP, wildcard (WC) bit, RP-tree (RPT) bit, Prune=(empty)
- Intermediate routers set up (*, G) state and forward the join



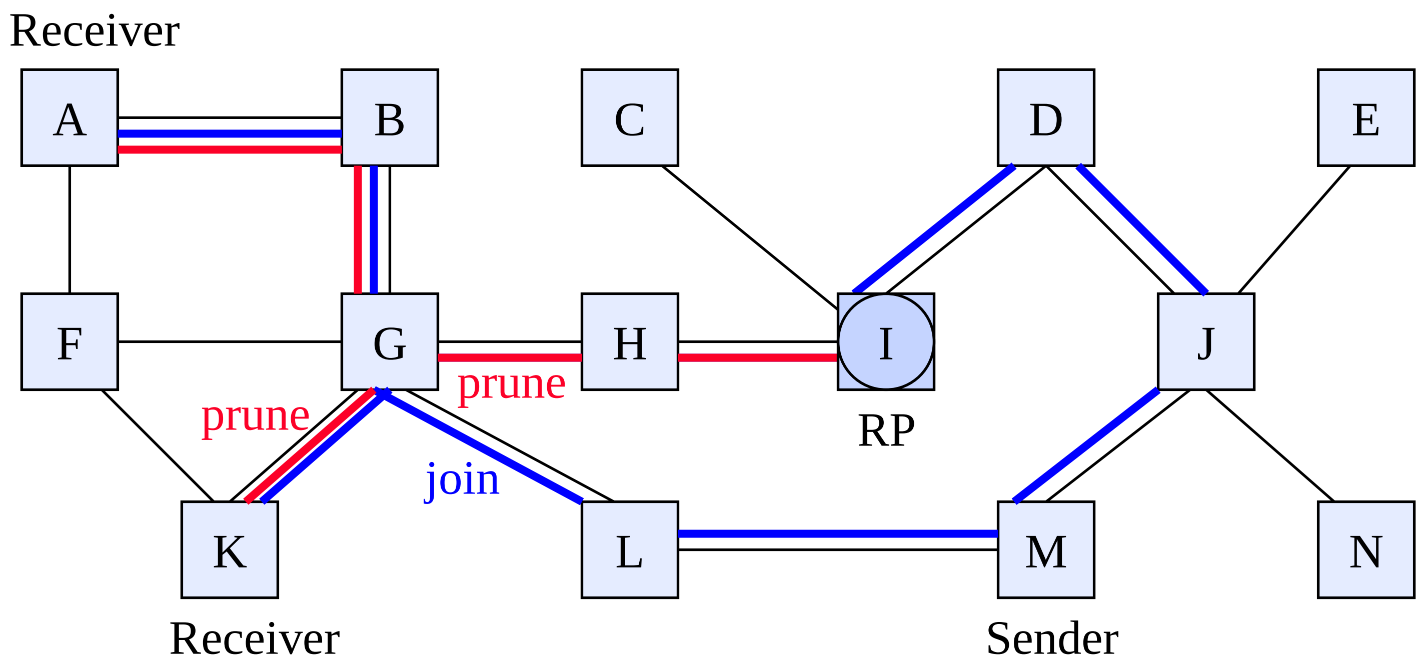
PIM-SM example (2)

- Senders send packets to RP encapsulated in register messages
- RP resends packets on the tree
- RP may contruct a (S,G) entry, and send periodic joins to the sender



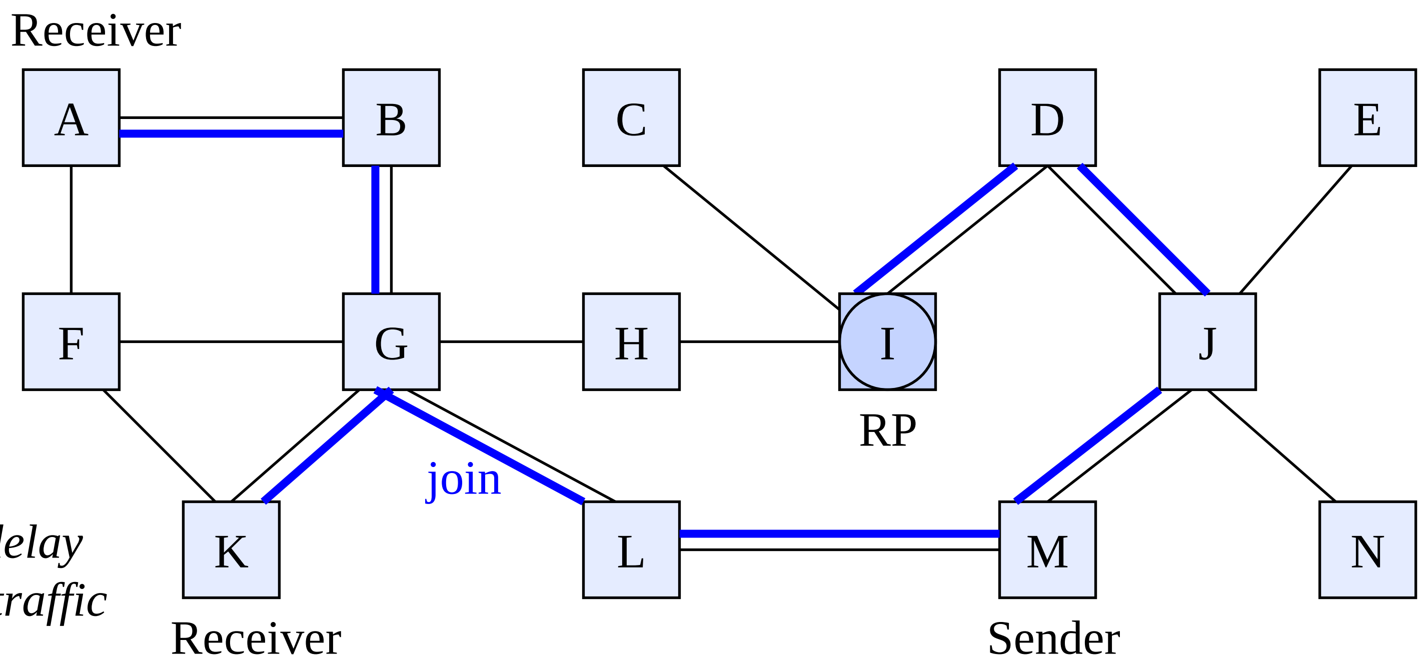
PIM-SM example (3)

- If the last-hop router (K and A) sees many packet from the source, it can switch from a shared tree to a shortest path tree for (S,G)
- It sends a join directly to the source, and prunes the previous path



PIM-SM example (4)

- Copies of the packets are still sent to RP
- Join/prune messages are sent periodically for each route entry



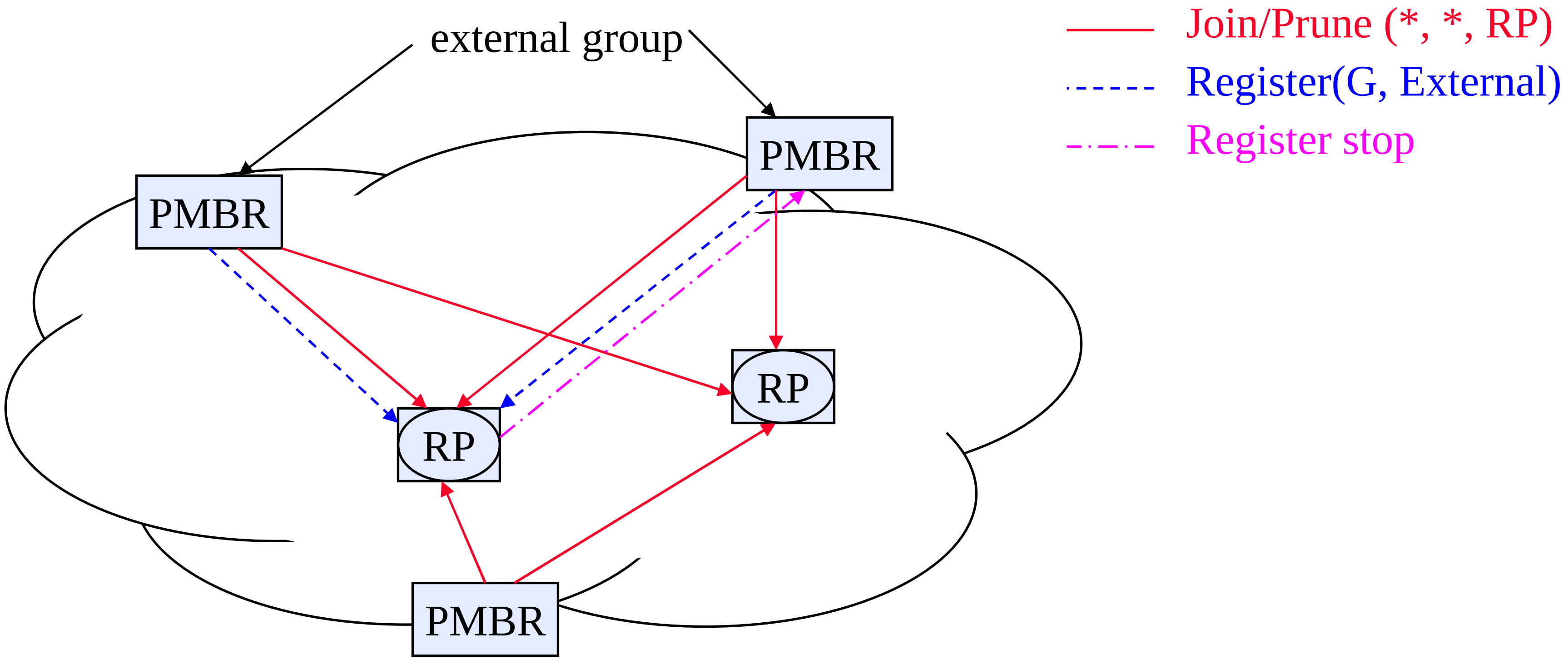
+ *minimize delay*
+ *distribute traffic*

Selection of Rendezvous Point

- A small group of routers configured as bootstrap routers candidates
- One of them selected as bootstrap router (BSR) for the domain
- BSR periodically sends Bootstrap messages through the domain
- A set of routers are configured as candidate RPs
  - typically same as candidate BSRs
- Candidate RPs periodically unicast Candidate-RP-Advertisements to the BSR, which includes them in the Bootstrap message
  - Candidate RP's own address
  - Optional group address and mask length
- The RP is selected by a hash function from the valid candidate RPs

PIM-SM can interoperate with DVMRP and other multicast protocols

- PIM Multicast Border Routers (PMBR) connects PIM-SM with other multicast protocols



Considerations

- PIM can switch from sparse mode to dense mode
  – Controlled by a parameter, which defines when the group is dense enough
- The RP may be a single point of failure
- The RP may be a bottle-neck

Summary of Multicast Protocols for the Internet

Tree type	Shared tree	Source based trees	
Algorithm	Center based tree	Flood and prune	Domain-wide reports
Protocols	PIM Sparse* Core Based tree*	DVMRP PIM Dense*	MOSPF

* Rely on unicast routing protocol to locate MC-sources.
  - Those that don’t, can route MC on routes separate from unicast routes.
* For shared tree protocols an additional step of finding the Core or Rendezvous Point must be performed.
* Directories are useful on service management level.

MOSPF

Multicast extensions to OSPF

MOSPF – Multicast extensions to OSPF (1)

- Idea: if the location of receivers is known to all routers, multicast should be possible to exactly the receivers only!
- MOSPF is an extension of OSPF, allowing multicast to be introduced into an existing OSPF unicast routing domain.
- Unlike DVMRP, MOSPF is not susceptible to the normal convergence problems of distance vector algorithms.
- MOSPF limits the extent of multicast traffic to group members, something e.g. DVMRP cannot always do.
  - Restricting the extent of multicast datagrams is desirable for high-bandwidth multicast applications or limited-bandwidth network links (or both).

MOSPF – Multicast extensions to OSPF (2)

- Defined in RFC 1584
- Unlike OSPF, MOSPF does not support multiple equal-cost paths
- MOSPF calculates the source-based trees on demand
- MOSPF can be, and is in isolated places, deployed in the MBONE. A MOSPF domain can be attached to the edge of the MBONE, or can be used as a transit routing domain within the MBONE's DVMRP routing system.

MOSPF can be deployed gracefully

- Introduces multicast routing by adding a new type of LSA to the OSPF link-state database and by adding calculations for the paths of multicast datagrams.
- The introduction of MOSPF to an OSPF routing can be gradual
  - Multicast capability marked with a M-bit in the option flag
  - MOSPF will automatically route IP multicast datagrams around those routers incapable of multicast routing, whereas unicast routing continues to function normally.
  - No tunnels $\Rightarrow$ there may be a unicast path, but no multicast path

Group-membership-LSA is created and flooded when an IP user joins an MC-group using IGMP

- LS Type 6 = Group Membership LSA

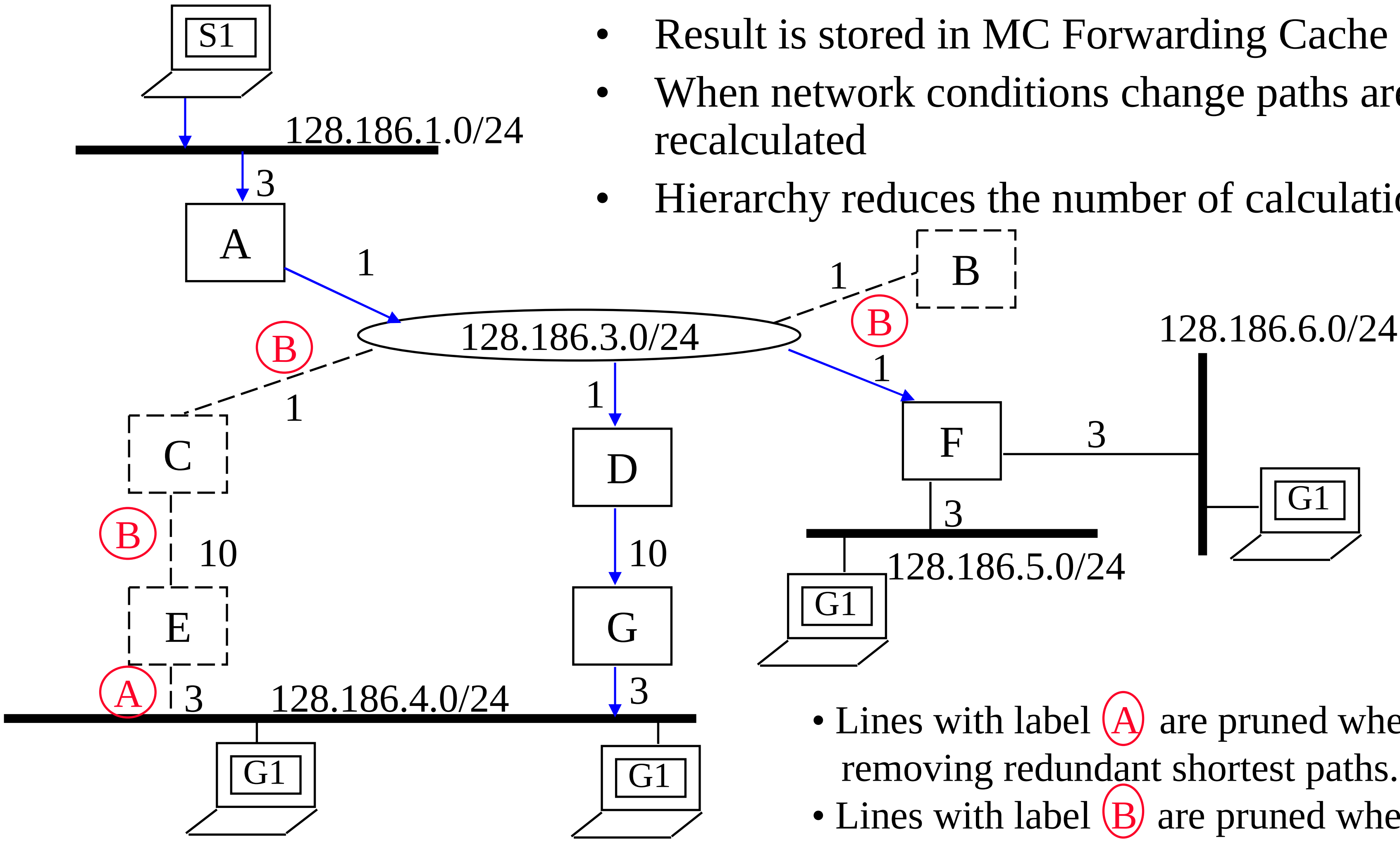
32 bits

LS age	**Options**=E	**LS type**=6
Link state ID = 226.1.7.6 (group G1)		
Advertising router = 128.186.4.1 (router E)		
LS sequence number = 0x80000001		
LS checksum	**length**	
Referenced LS type = 2 (network)		
Referenced Link State ID = 128.186.4.1		

Common LS header

MOSPF calculates shortest-path trees on demand

- Result is stored in MC Forwarding Cache Entry
- When network conditions change paths are recalculated
- Hierarchy reduces the number of calculations



- Lines with label A are pruned when removing redundant shortest paths.
- Lines with label B are pruned when removing links that do not lead to G1

Forwarding cache entry stores multicast path routing info

- For each source network and group ⇒

Router or network for multicast reception	
List of interfaces, multicasts must be sent	Metrics to nearest group member

- A cache entry may be deleted at any time ⇒ Will be recalculated on demand.
- Cache entries must be deleted, when changed LSAs are received
  - Router-LSA, Network-LSA (on router or link failure or cost change) ⇒ Delete all entries since it is not possible to tell which are affected.
  - Group-m-LSA ⇒ Delete entries of that group.
  - Hierarchy ⇒ The farther away the change is the fewer cache entries are deleted.

On demand route calculations use Dijkstra's shortest path first algorithm

- Calculation is rooted on the source
  - not in the current router as for unicast
- For a new multicast, every router performs the same calculation
- Stub networks do not appear in MOSPF calculation
  - e.g router F
- For equal cost routes, the previous hop router with the highest address is chosen
  - e.g. G over E