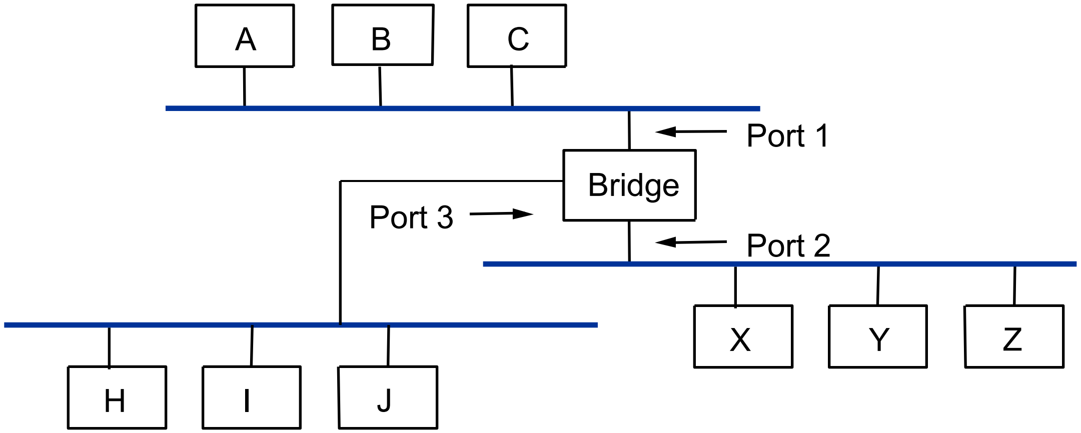
Multicast Routing Papers: DVMRP and MOSPF

Computer Networks
Dr. Jorge A. Cobb

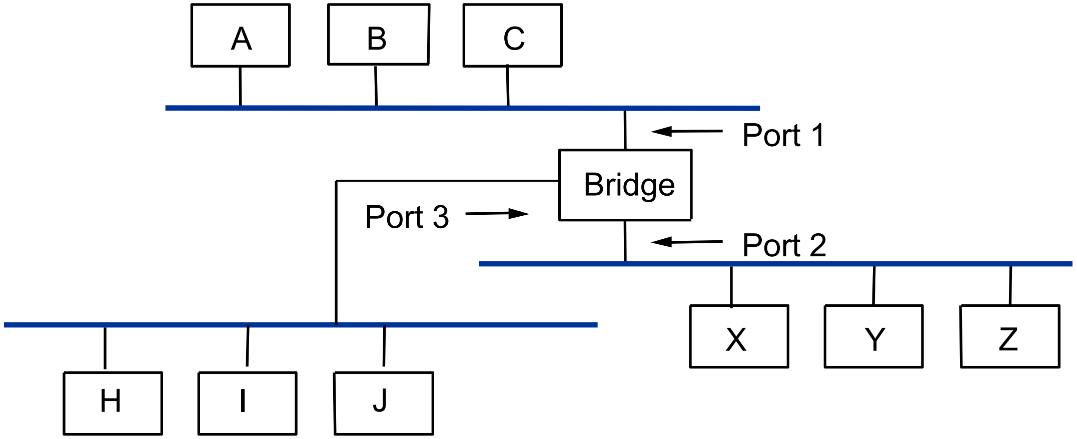
UTD

Bridges and Extended LANs

- LANs have physical limitations (e.g., 2500m)
- **Bridges** connect two or more LANs together.
- They are transparent to hosts (does not change packets in any way)
  - Bridges see all messages in its attached LANs
  - Copy message from one LAN to another if necessary



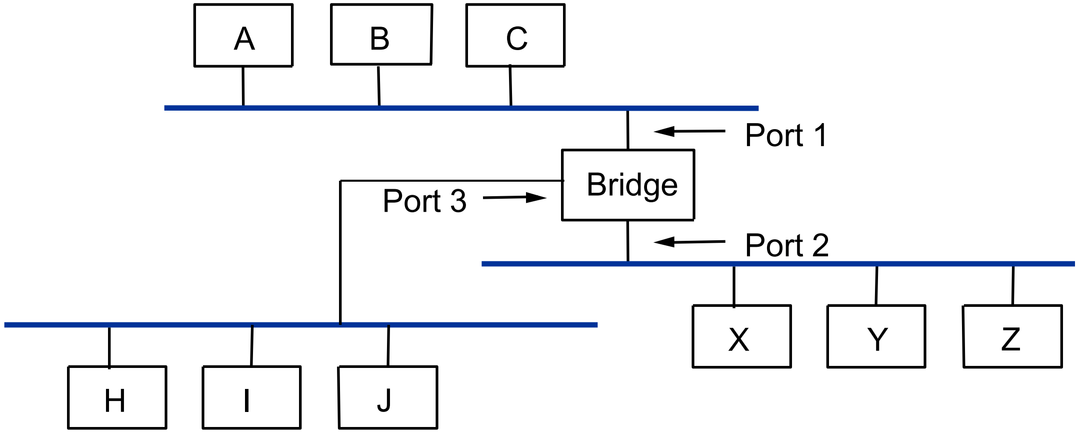
Learning Bridges



Host	Port
A	1
B	1
C	1
X	2
Y	2
H	3

- Forward messages only when necessary (without modifying the message in any way).
- Maintain a forwarding table
  - Often incomplete (initially empty!)
  - Learn table entries based on source address of messages seen.
  - If destination is not on table, forward over all other ports

Learning Bridges

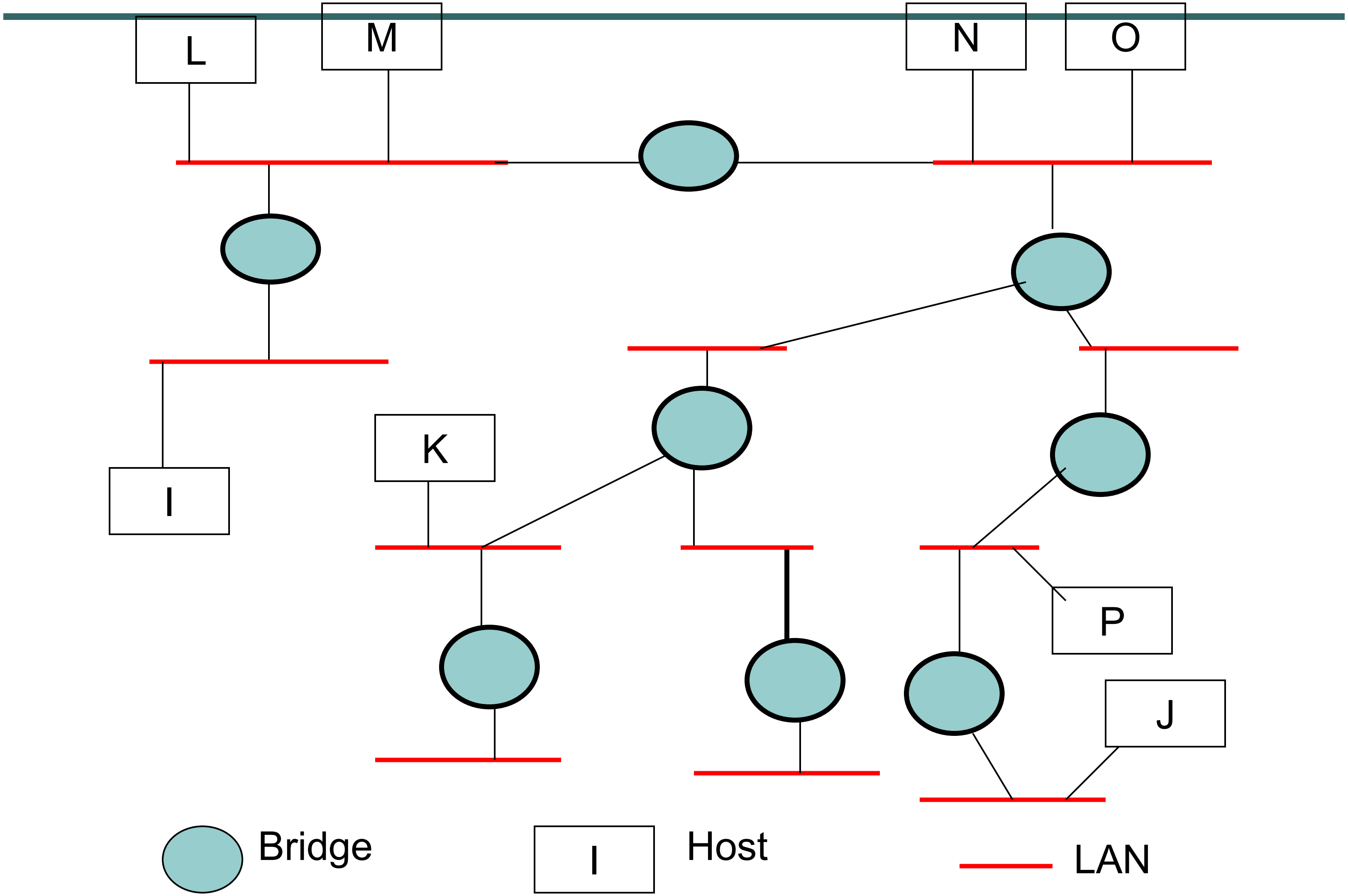


Host	Port
A	1
B	1
C	1
X	2
Y	2
H	3

- I sends a message to Z
- Bridge “learns” I comes from port 3
  - I is added to the table
  - Z is unknown
    - bridge copies the message over BOTH port 1 and port 2

- Z sends a message to I (e.g. some reply)
- Bridge “learns” Z comes from port 2
  - Z is added to the table
  - bridge copies the message ONLY over port 3 (I is already in the table)

Example: Learning Bridges

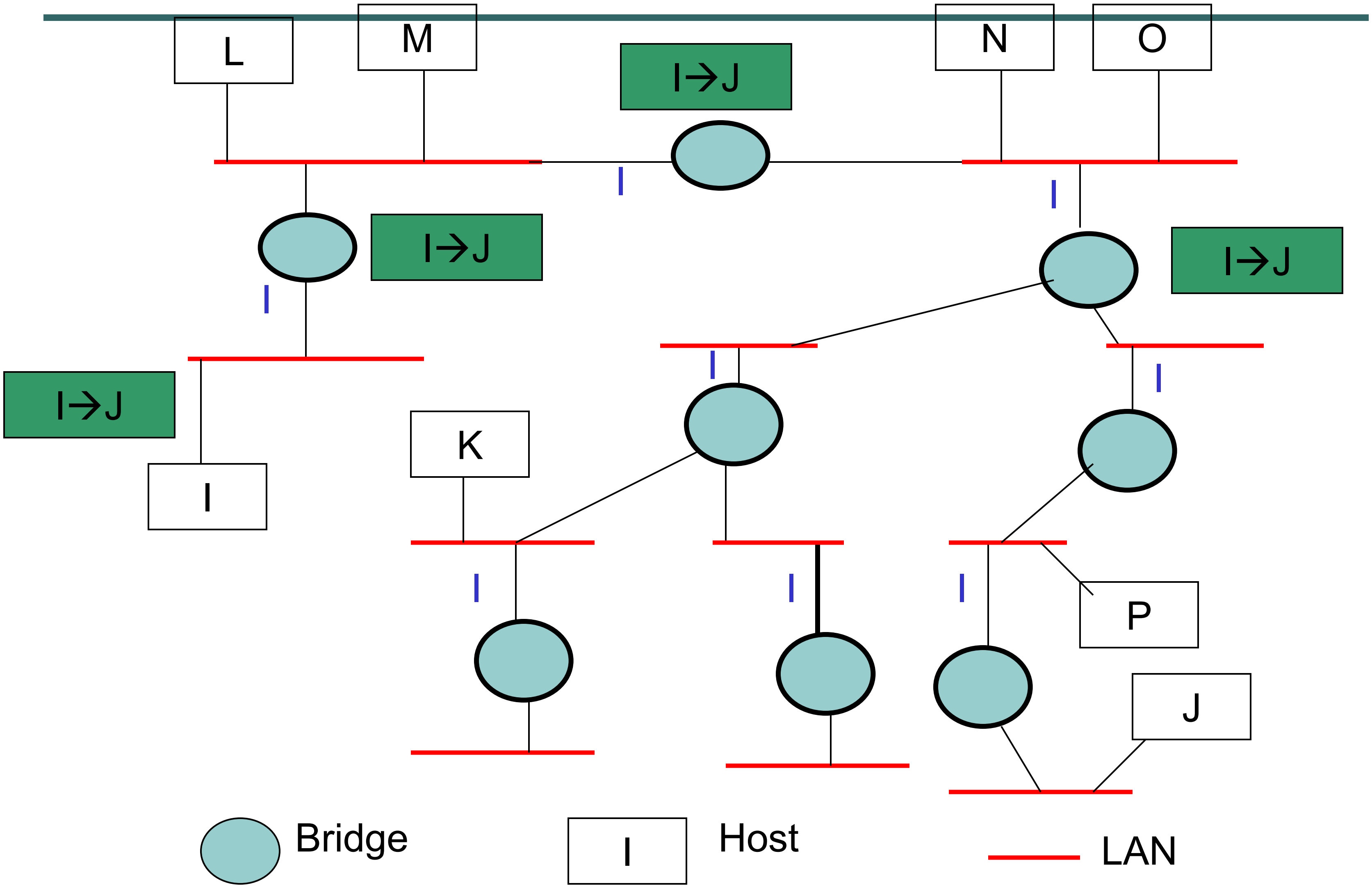


Example: Learning Bridges

Assume initially all bridges forwarding tables are empty

- Host I sends data message to Host J:
  - This data message is broadcasted to every LAN.
    Why? Because no one knows where J is!
  - Due to which, every bridge learns about host I location and creates forwarding table entry for Host I.

Example: Learning Bridges



Example: Learning Bridges

- Host J then sends data message to Host I:

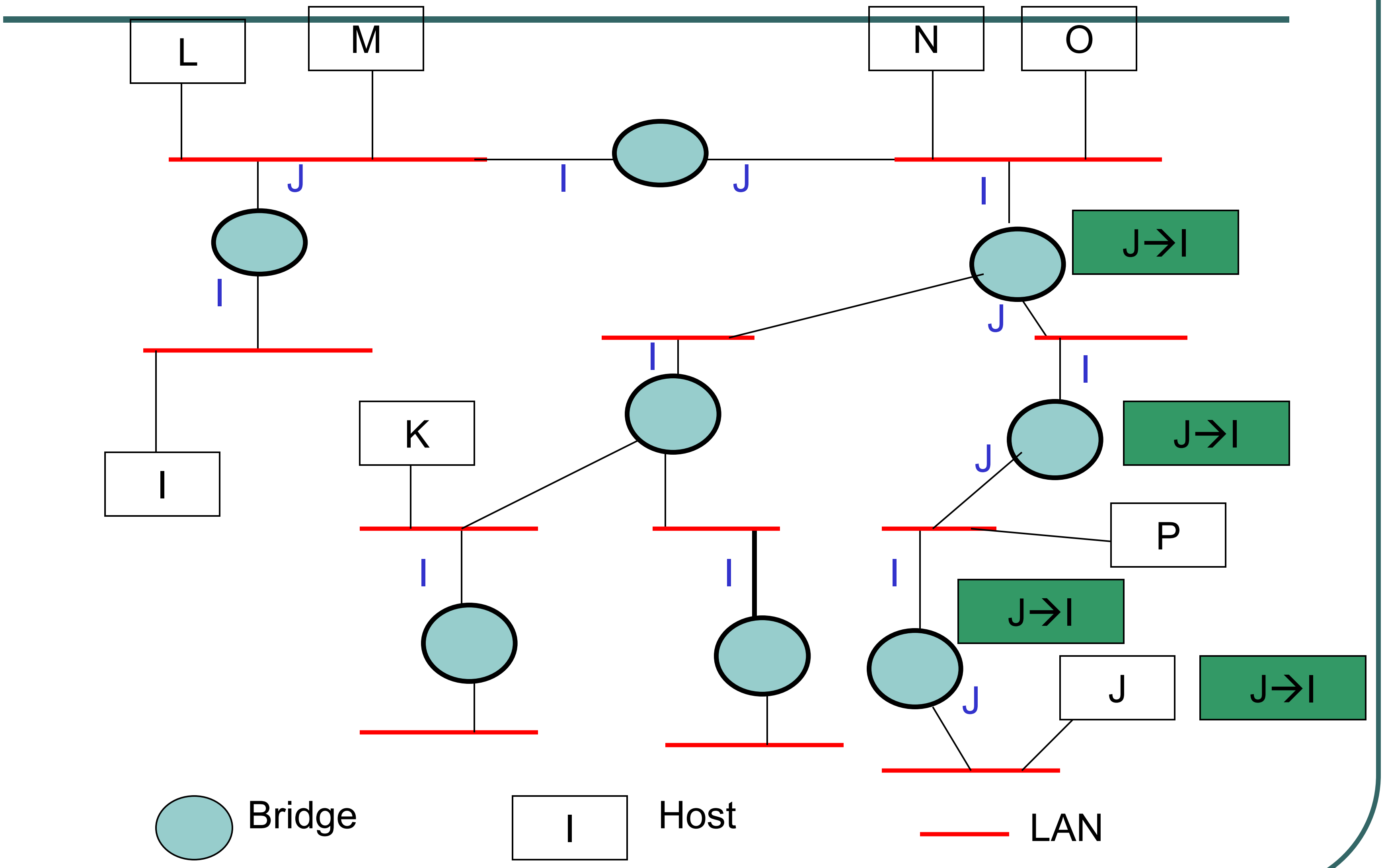
  - Since all bridges have an entry for host I, message goes only through Host J →B_7 →B_8 →B_3 →B_2 →B_1 →Host I

  - Only these bridges have now an entry for Host J.

  - All other nodes do not, e.g., B_4 does not forwarding table entry for Host J.

  - Note that from now on (until the lifetimes expire) all data messages between Host I and Host J go only through the LANs on the path from Host I to Host J.

Example: Learning Bridges

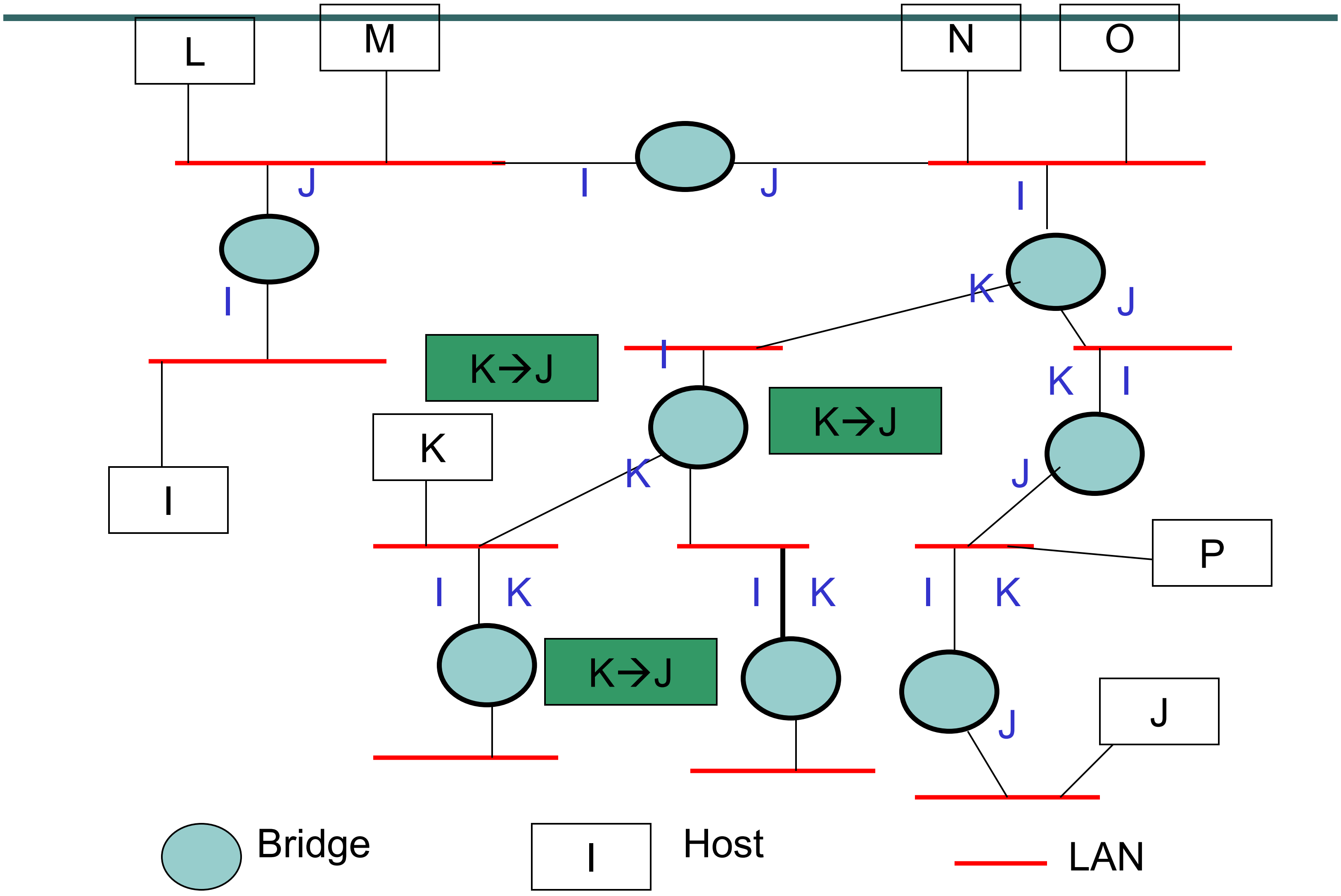


Example: Learning Bridges(cont)

Host K sends a data message to Host J:

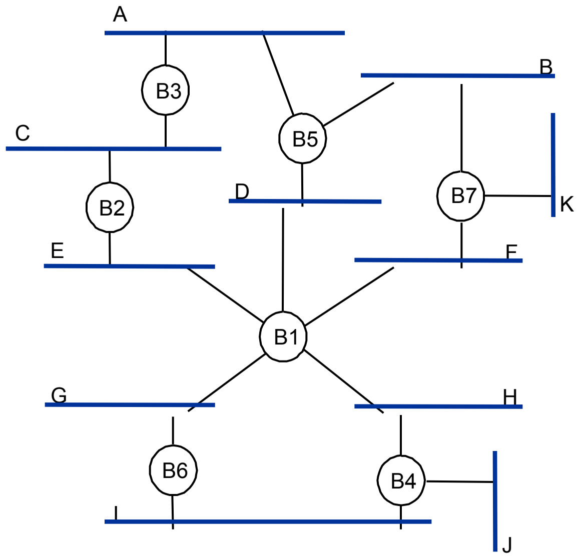
- The nodes seeing this data message are B_4 B_5 B_6 B_3 B_8 B_7
- only these nodes learn Host K location.
- Host k packet has been broadcast in a limited fashion.

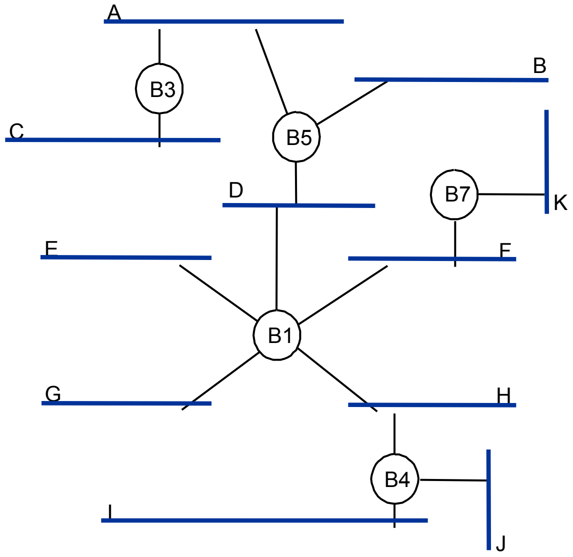
Example: Learning Bridges



Spanning Tree Algorithm

- Problem: loops





- Bridges run a distributed spanning tree algorithm
  - Logically delete certain bridges to have a spanning tree
  - We will assume we have a spanning tree

Multicast Routing in Datagram Internetworks and Extended LANs

S. Deering and D. Cheriton

ACM Transactions on Computer Systems
1990

Efficient multicast in Extended LANs

- How to do efficient multicast in Extended LANs? (I.e. in LANs with bridges) (NOT IP !!!!)
- A simple way: bridges propagate multicast (and broadcast) packets across every segment of the extended LAN
  - Way too inefficient, especially for multicast applications with sparsely located receivers

How to do multicast efficienty in Extended LANs?

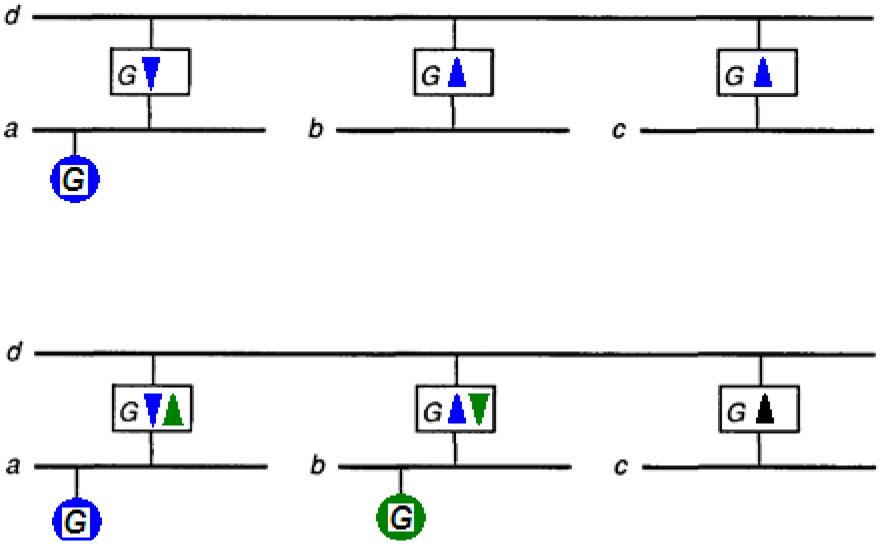
How to locate receivers

- **<u>If</u>** bridges know which interface leads to members of a given group, they will forward the packets on those interfaces only
  - Remember, we have a tree
  - There is only one path from the bridge to each receiver.
- **<u>But</u>**, in general, how do bridges learn which interface leads to individual hosts?
  - When a packet arrives from a host, bridge records the (source-host addr, interface, age) into a table
  - What about receivers?
  - Receivers don’t send data!

How to locate receivers (contd)

- We want to forward multicast packets only over LANs leading to receivers
- To learn the location of receivers, we force receivers (i.e., group members) to periodically transmit a *membership-report*
- A member (receiver) of a group G periodically sends a ***membership-report*** of the form:
    - LAN source address = G,
    - LAN destination address = “ALL-BRIDGES” multicast address.

Bridge Multicast Table



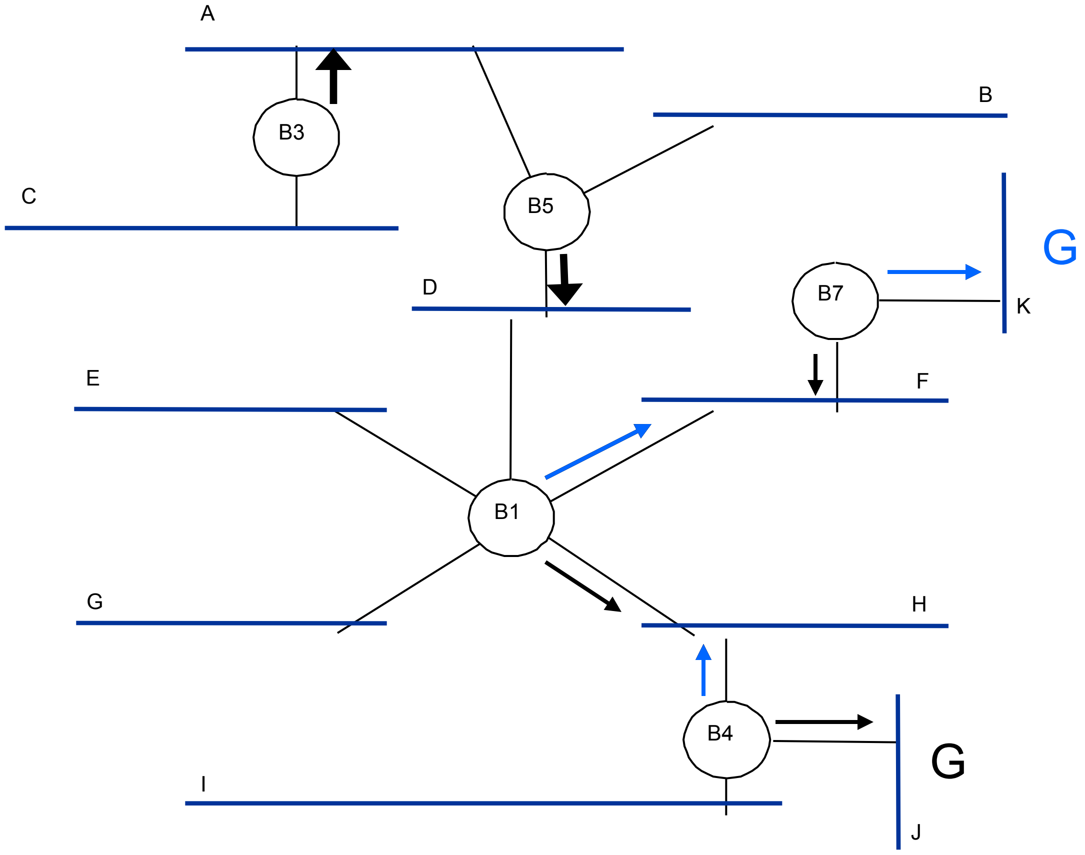
Arrow indicates where group members are located

- Multicast table at each bridge has the following row for each multicast group: **(muticast addr, (outgoing interface, age), (outgoing interface, age), …)**
- A bridge receiving a repot records the incoming interface of the report as an outgoing interface for group G
- It then forwards the report over all of its interfaces in the extended LAN (why over all interfaces? I.e., why a broadcast?)

Single Spanning-Tree Multicast Routing

- Bridge algorithm (summary)
  - If source address is a mcast group address (i.e. a report),
    - record arriving interface as an outgoing-interface with an age of zero for this mcast address
    - and forward to all other interfaces.
  - Periodically increment age of outgoing interface
    - when *age=expiry threshold,* delete this outgoing-interface info from the table’s entry
    - if no outgoing-interfaces remain, delete the entire entry
  - If a packet arrives with a multicast destination address
    - forward a copy on every outgoing-interface recorded in the table entry (if any) excluding the arriving interface

Another example



- Arrows indicate the location of the group member
- Membership reports are forwarded over all links of a bridge

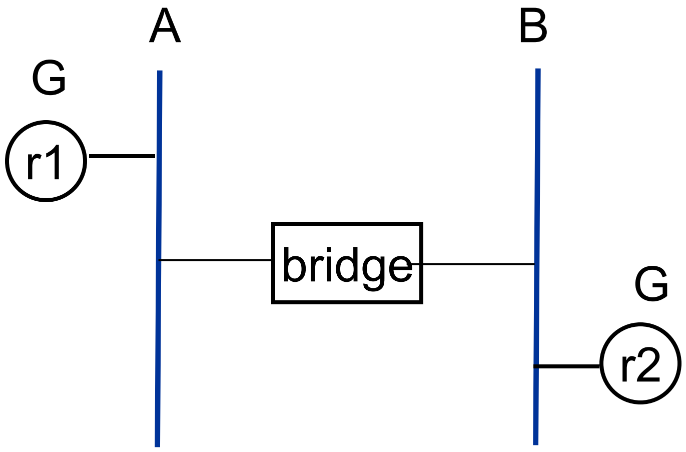
G = group G member (receiver)

What if a host at E sends a msg to group G?
What if a host at A sends a msg to group G?

Suppressing Membership Reports

- An efficiency improvement to suppress unnecessary membership reports
  - Hosts send membership-reports as (G,G)
  - This suppresses #membership reports to 1 per report interval per LAN segment
    - because all other group members (hosts) on the same LAN segment heard the first membership-report (G,G), and will not send a report
  - Bridge, on receiving a pkt with address (G,G) (bridges receive ALL packets, remember?)
    - **changes it to (G, all-bridges)** and forwards to other interfaces
    - why?

Why change the destination?



- Assume member r1 sends a membership report (G,G) over Lan A
- Assume the bridge forwards it as (G,G) over Lan B
- This will suppress the membership report of r2
- The bridge will never know there is a group member in Lan B

- This problem is avoided by the bridge changing the message to (G, “all-bridges)

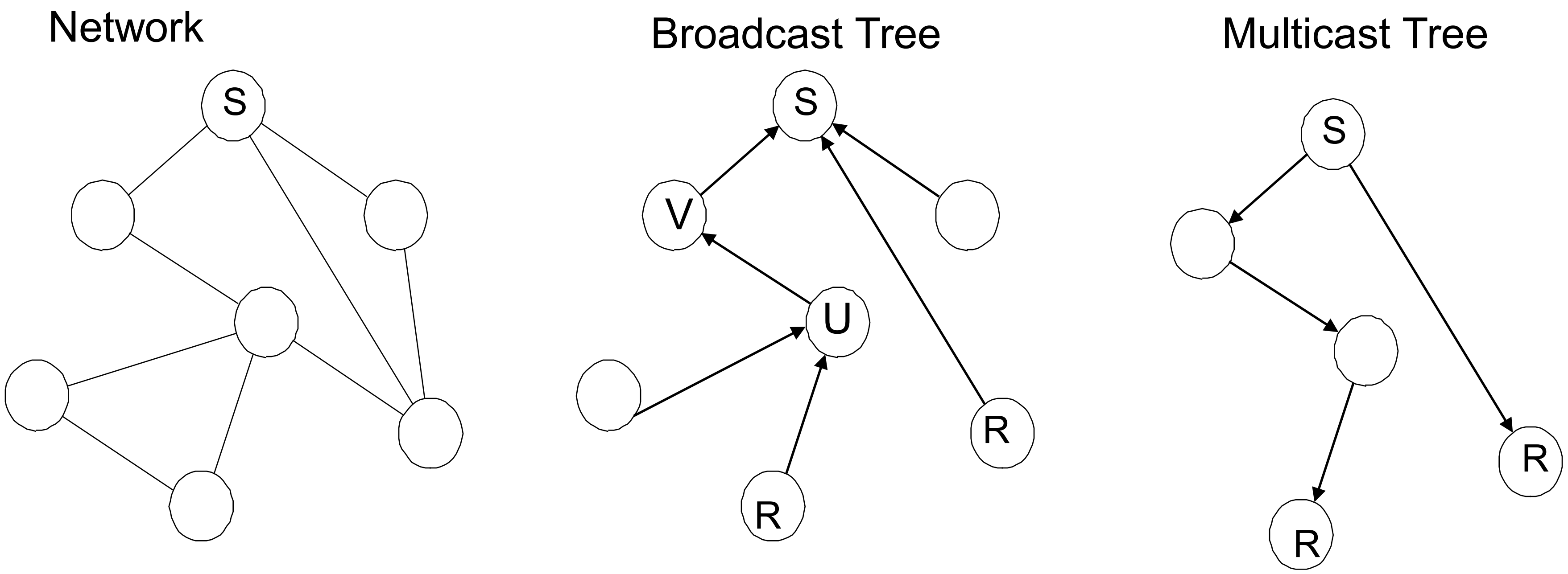
- Done with extended LANS (bridges)
- FORGET about bridges (until the exam ☺)
- Now we do IP multicast over multiple LANs (across IP routers)

Distance Vector Multicast Routing

- How to support multicast routing in a distance-vector environment? (this is general enough for ANY unicast routing protocol)
- Compute a spanning (i.e. broadcast) tree across all the links
  - prune it to become a multicast tree
- Specifically, a source-based shortest path spanning trees
  - Tree is rooted at the source site
  - It corresponds to shortest path from each receiver to the source
    - Main assumption is path symmetry *(links have the same costs in both directions)*
- Observation:

  Every shortest-path multicast tree rooted at the sender is a subtree of a single shortest-path spanning (i.e. broadcast) tree rooted at this sender

Broadcast Trees and Multicast Trees in Point-to-Point Networks



Network

Broadcast Tree

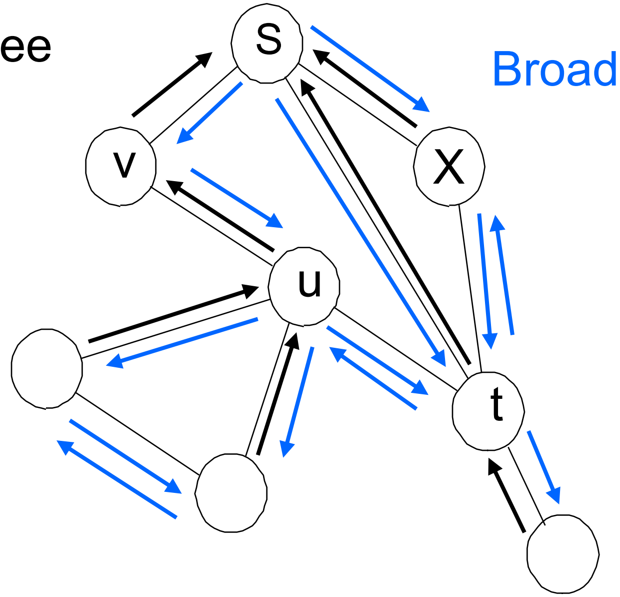
Multicast Tree

S = source (host) node, also root of tree
R = receiver
For any router U, parent(U,S) = next-hop(U,S) = V

Reverse Path **Flooding** (RPF) algorithm (**broadcast** in **point-to-point** networks)

- This is not a standard protocol, is just a general method to do broadcast.
- When a router receives a broadcast packet from source S
  - If the packet arrives via the next-hop router to S
  - Then,
    - Forward the packet to all outgoing interfaces (except the incoming one, of course)
  - Otherwise
    - Throw the packet away

Example



Broadcast tree

Broadcast messages

- Which of all three copies will be forwarded by t?
- Note: each router forwards each packet only once (why?)
- Router needs to know the shortest path back to S
    - Given by the distance vector routing algorithm

Internetworks (review)

- A multicast packet sent by a source host S will have
  - LAN source: LAN address of S
  - LAN destination: LAN multicast address for group G
  - IP source: IP address of S
  - IP destination: IP multicast address for group G
- Routers do not modify the IP src or dst addresses
- Routers send the packet over the LAN with
  - LAN source: LAN address of router
  - LAN destination: LAN multicast address for group G
- All routers "listen" (receive a copy) of all LAN packets addressed to any multicast group.
- Original RPF is designed for <u>point-to-point</u> links
- In the Internet we typically have <u>multi-access LANs</u> (e.g. Ethernet)
  - There are many routers per LAN

A problem with original RPF

- When multi-access (shared) links (Ethernet) are used between routers, two problems arise

- Problem 1: multiple copies of the packet is sent on the shared link
  - Multiple routers are attached to the same LAN (Ethernet)
  - Waste of bandwidth on the link & waste of router resources
  - We want only one copy to be sent per LAN

- Problem 2: identifying the parent on the tree
  - In RPF, routers only accept a broadcast packet from the next router on the path back to S.
  - If there are multiple routers on the next-hop-LAN, from which to accept?
    - If we solve problem 1, we automatically solve problem 2.
    - Why? A packet is accepted if it comes from the LAN of its next-hop, regardless of which router sent it.

Reverse Path Broadcasting (RPB)

- Objective: eliminates duplicate broadcasts on shared links in RPF

- Each router R must determine, for each neighboring LAN λ,
  - Is λ the parent of R on the tree?
  - Is λ a child of R on the tree?

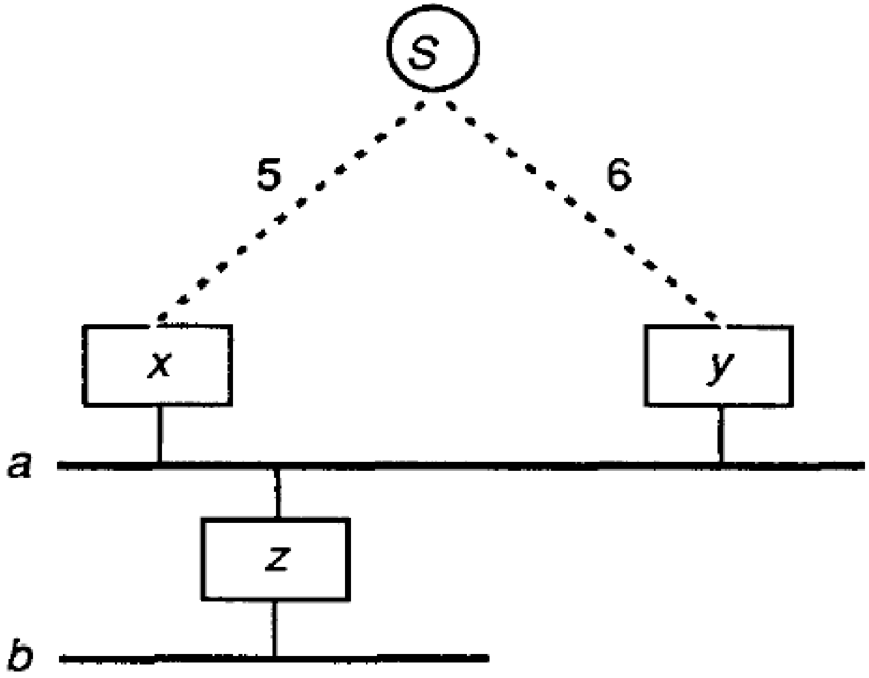
Is LAN λ the parent of R on the tree?

- R accepts multicasts only from the LAN λ where its next-hop router to S is located.

- Note, the next-hop router to S may not be the router that places the multicasts from S on λ.
  - Nonetheless, R accepts the multicast.

Is LAN λ a child of R on the tree?

- Identify a single “parent” router for each LAN w.r.t. S
  - For the LAN S is attached to, S is considered the parent
  - Otherwise, the router with min distance to S is the parent
  - In case of tie, router w/ lowest IP address is the parent

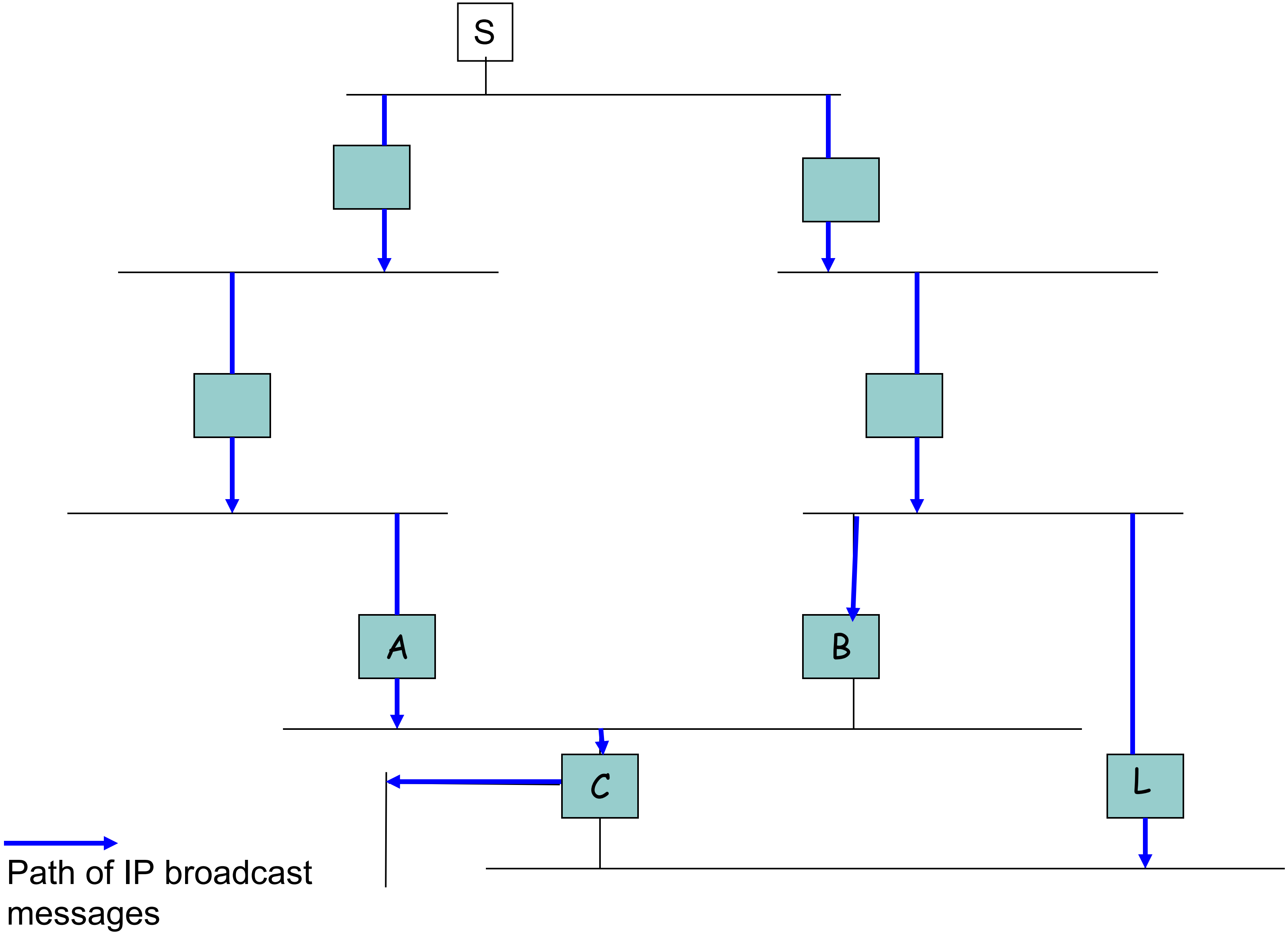


In RPB:

x, y and z learn that only x should inject packets into LAN a

z forwards any multicast from S coming from LAN a

What if:
- x and y have the same cost to S
- IP(x) < IP(Y) i.e. x injects multicast packets into LAN a
- routing table at z for S points to y and not x
- is there a problem here?



Path of IP broadcast messages

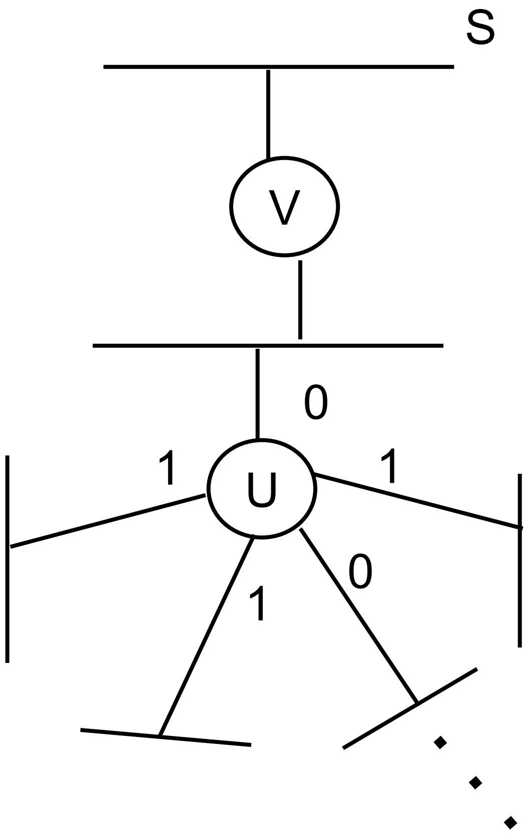
Observations

- **How to identify min distance router to S? (i.e. parent of LAN)**
  - Routers exchange distance vector records with each other
  - Therefore, each router independently (on its own) finds if it is the parent of each of its neighboring LANs
- ANY host, on ANY LAN, at ANY moment can send a multicast (or broadcast) message
  - I.e., the router must ALWAYS know its parent LAN and who are its children LANs regardless of where the source is.

Overhead

- Being always prepared requires that routers add a *children* bitmap to each routing table entry (i.e. each destination LAN could be a possible source LAN S)
  - the bit-map for “destination” S has one bit for each incident link λ
  - Bit (S,λ) is set, if λ is a child link (i.e. if I am the parent of this link) for broadcasts originating from *S*

Example



Unicast routing table at U

Dest	NxtHop	Cost	ChildMap
P	X	5	01110
Q	Y	2	01000
. . .			
S	V	2	01011

We take advantage that we already have a unicast routing table with an entry per LAN and who tells us who the next hop (parent) is.

Pruning the Tree

- Both RPF and RPB are broadcast algorithms !
- To provide shortest path multicast delivery from source S to group members, broadcast tree of S must be pruned back to reach only links with receivers

(small detour) Why not do membership reports?

- Recall multicast in extended LANs?
  - We had one tree
  - Each receiver of a group G would broadcast a membership report along the tree
  - All bridges would learn the direction of the receivers.

- Can we do the same here?
- We have N trees, where N is the number of LANs (subnets)
  - each LAN could be the root of one tree
- Each receiver of group G would have to send a membership report on each of these trees.
  - The overhead is N times!
  - Too costly

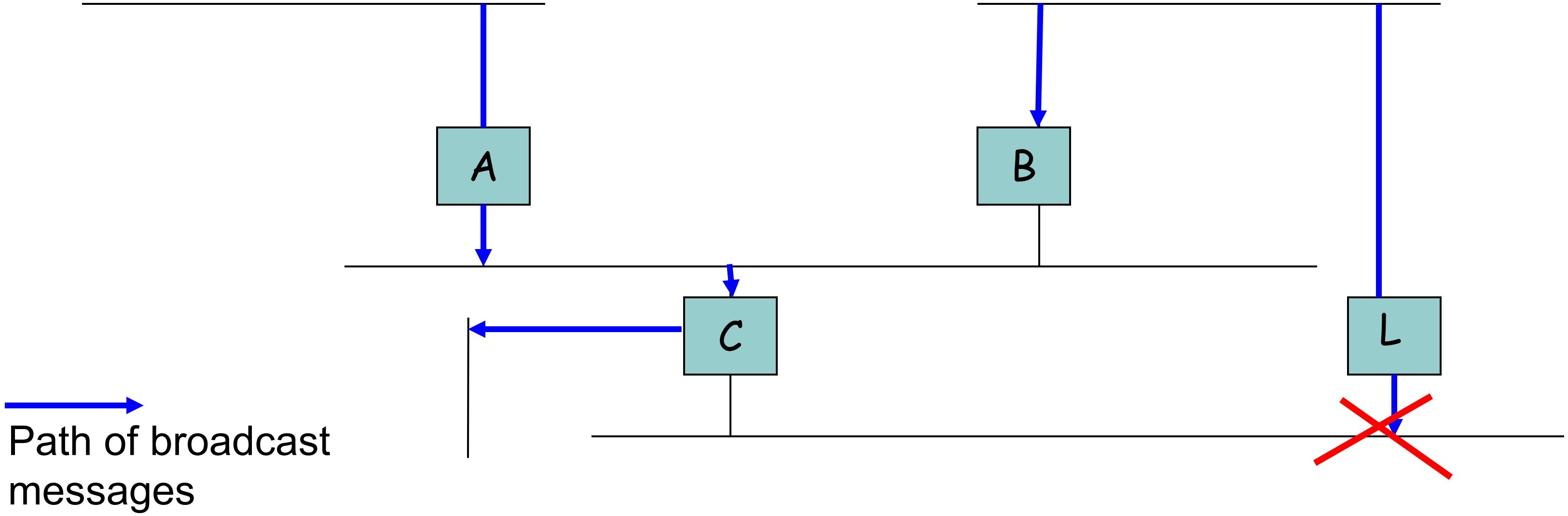
What do we do?

- We will prune the broadcast tree of RPB into a multicast tree
- We will do so in several stages.

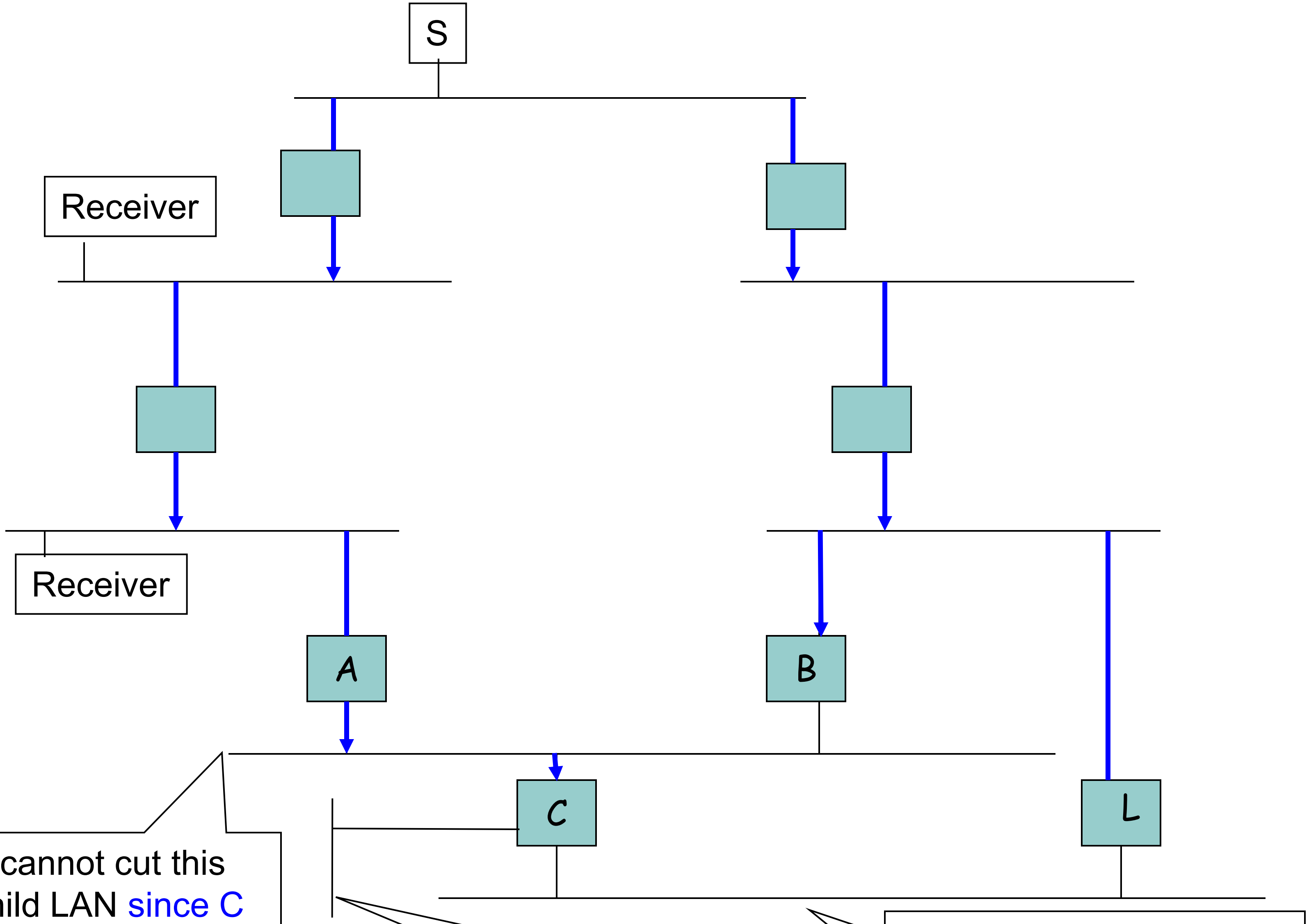
Truncated Reverse-Path Broadcast (TRPB)

- An alternative in which only non-member *leaf* LANs are deleted from each broadcast tree
  - A router truncates a child link if
    - no router uses this link to receive multicast messages from the source (i.e., it is a leaf link)
    - No host is a group member on this link (LAN)

Leaf Truncation example



- At L, L truncates the lower LAN in the figure if
  - if C does not accept multicast messages from this LAN (we are assuming C uses the top LAN as next hop)
  - no host is a group member on this link (LAN)



A cannot cut this child LAN since C uses this LAN as its next hop (not leaf link)

C truncates this LAN since no routers use it (it is a leaf link) and there are no receivers.

L truncates this LAN since C does not use L to reach S (it is a leaf link) and there are no receivers.

Algorithm Implementation.

- If a multicast packet (S, G) arrives from the *next-hop-link* for S,
  - forward a copy of the packet on *all child links for S*,
  - except leaf links (no other router receives from this link) that have no members of G
- To implement this, we need two things:
  - The router needs to learn if the child link is a leaf link
    - I.e. If no other router uses this link to receive messages from the group.
  - The router also needs to know if no host group members are on this link.
- We tackle each of these in turn.

Leaf Link or not?

- Distance vectors tell me the distance from routers on this link to S, but not if I am their next hop.

- However, if DV with split-horizon and poisoned-reverse is used then
  - If at least one router gives me a distance of infinity then it uses my LAN as the next hop
    - I.e., the link is not a leaf link
  - The link is a leaf link if no router gives me a distance of infinity.

Leaf Bitmaps

- Leaf pruning requires that routers add a *leaf* bitmap to each routing table entry (i.e. for each possible source)
  - the leaf bit-map for routing table entry S has one bit for each incident link λ
  - bit for (S,λ) is set, if λ is a leaf link of this router for multicasts originating from *S*

Receivers on a Link?

- (Paper ☺) Hosts send a membership report message over their LAN using the mcast group address as the destination (all hosts group members listen to this, and only one report is sent per interval)
  - Real Life: use IGMP protocol.

- Routers maintains a table with one entry per link (LAN)
  - The entry contains a bit-map field, *link-groups,* with one bit per (active-group,link)
  - Bit (G,λ) is set if members of group G are on link λ

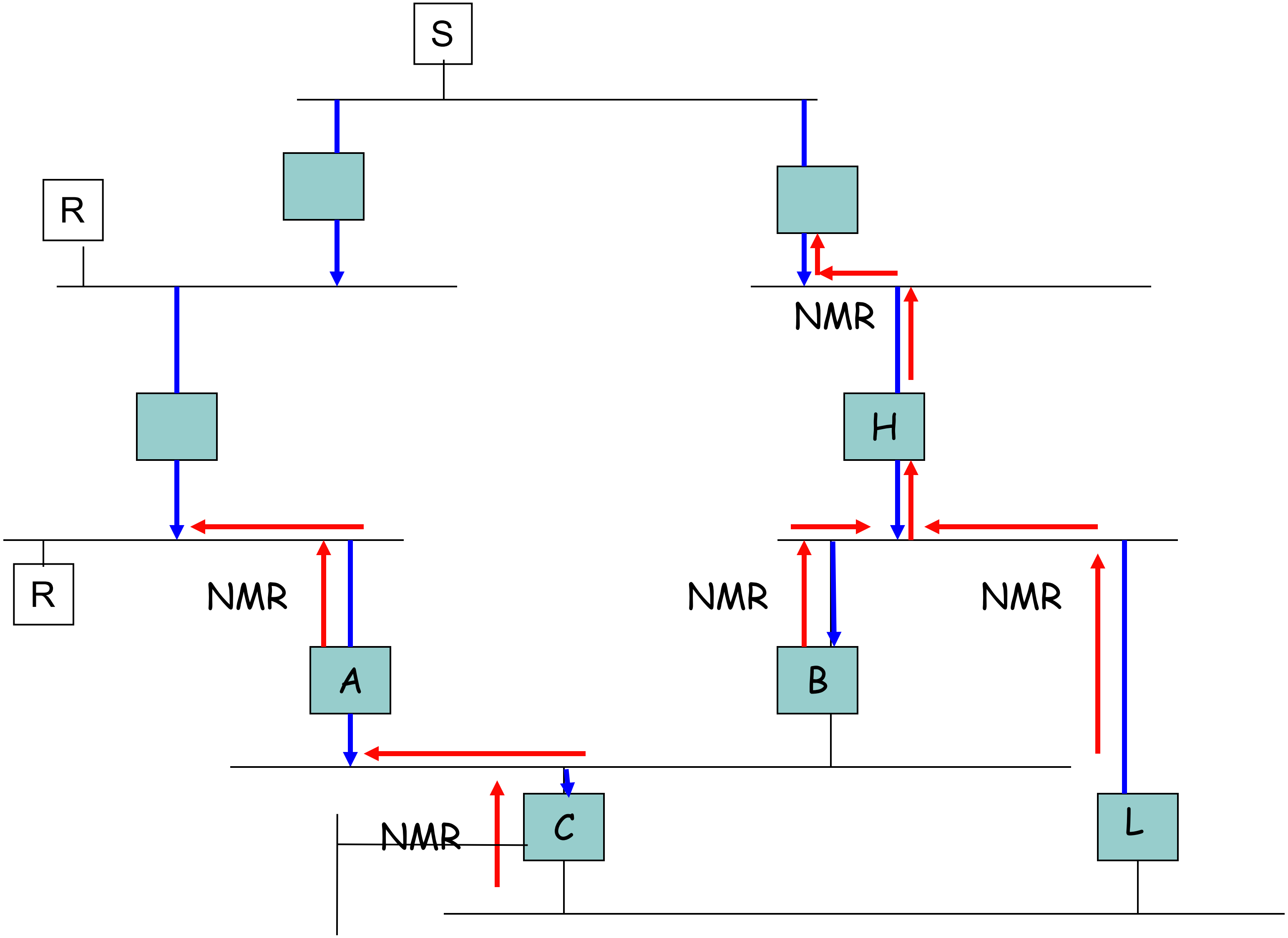
Overview of “bitmaps” required

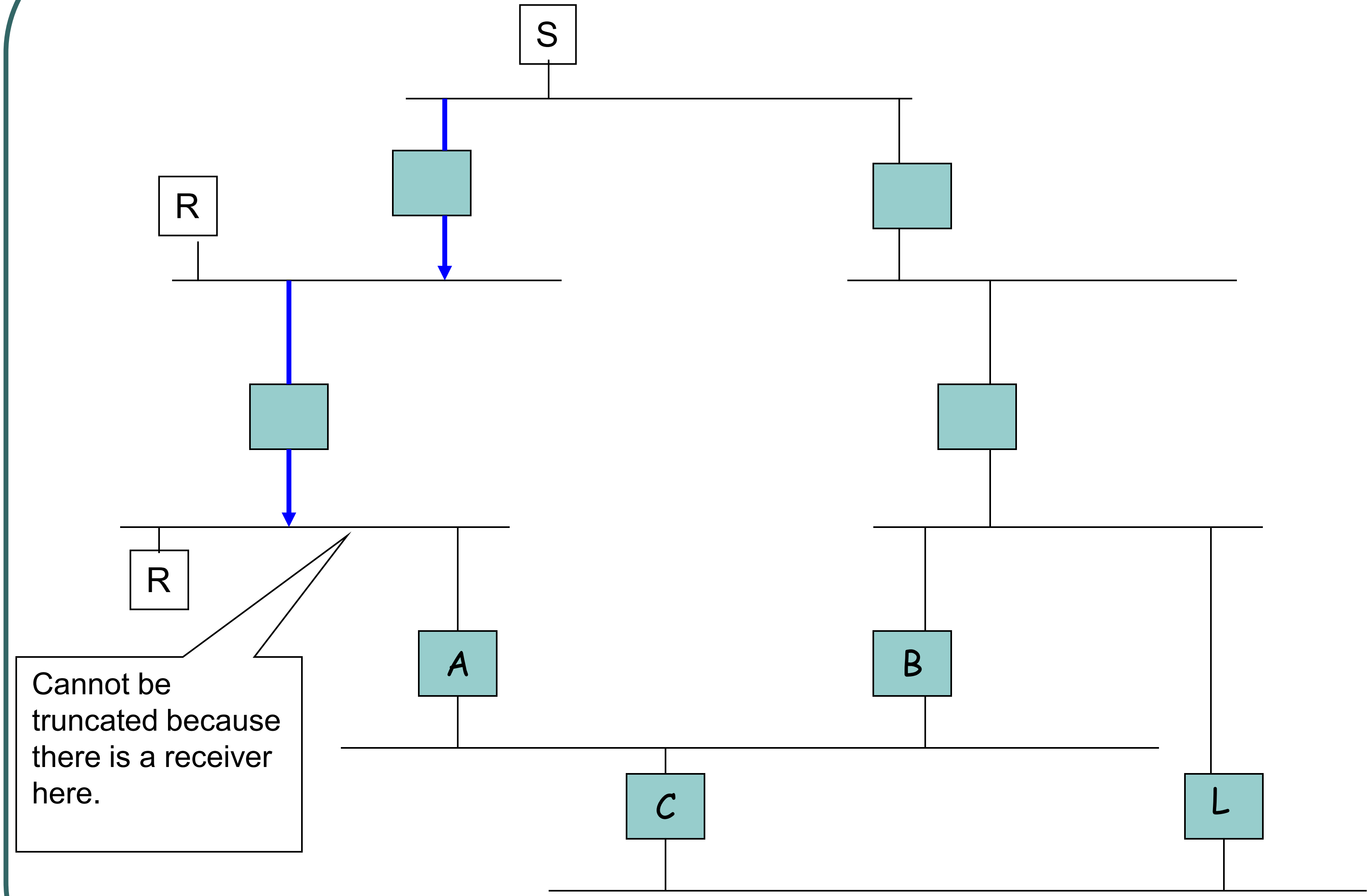
- For each possible source LAN S and each link λ,
  - Is λ a child link of the router on the broadcast tree of S?
  - Is λ a leaf link of the router on the broadcast tree of S?
  - Both of these are
    - obtained via the DV routing algorithm (**the second one via split horizon with poisoned reverse**)
    - stored along the unicast routing table

- For each link λ and active group G,
  - Does λ have any members of G?
  - Obtained via membership reports (IGMP)

Reverse Path Multicasting (RPM)

- Prune shortest path multicast tree as follows
  - First packet for (S,G) is forwarded to everyone on the truncated shortest path broadcast tree according to TRPB
  - **Leaf routers** (i.e., all child links are leafs) with no attached members send non-membership report (NMR) to the parent router on the LAN.
  - If a router receives NMR from all of its children routers
    (how do you know who these are?)
    and itself has no directly attached members,
    - then it also sends NMR to its parent router on the tree.


S
R
NMR
H
R
NMR
NMR
NMR
A
B
NMR
C
L



Cannot be truncated because there is a receiver here.

NMR expiration

- NMR reports include an *age* field,
  - when it expires data flows all the way to leaves again and gets re-pruned back
- Routers remember NMR reports that they sent
  - When a new host joins G, they send a cancellation message to undo the effect of NMR

Pros and Cons

- Reverse path multicasting, when used with distance vector routing, is known as distance-vector multicast routing protocol (DVMRP)
- Advantages: good when there are many receivers, since multicast messages are initially flooded to the entire network.
- Disadvantages
  - Bad if there are few receivers
    - Again, the first multicast messages are sent throughout the network unnecessarily.
    - Routers need to remember the “prune” state, i.e. they need to maintain state even when there are no receivers below them on the tree
  - The path from source to receiver may not be optimal if the cost of links is not bi-directional.

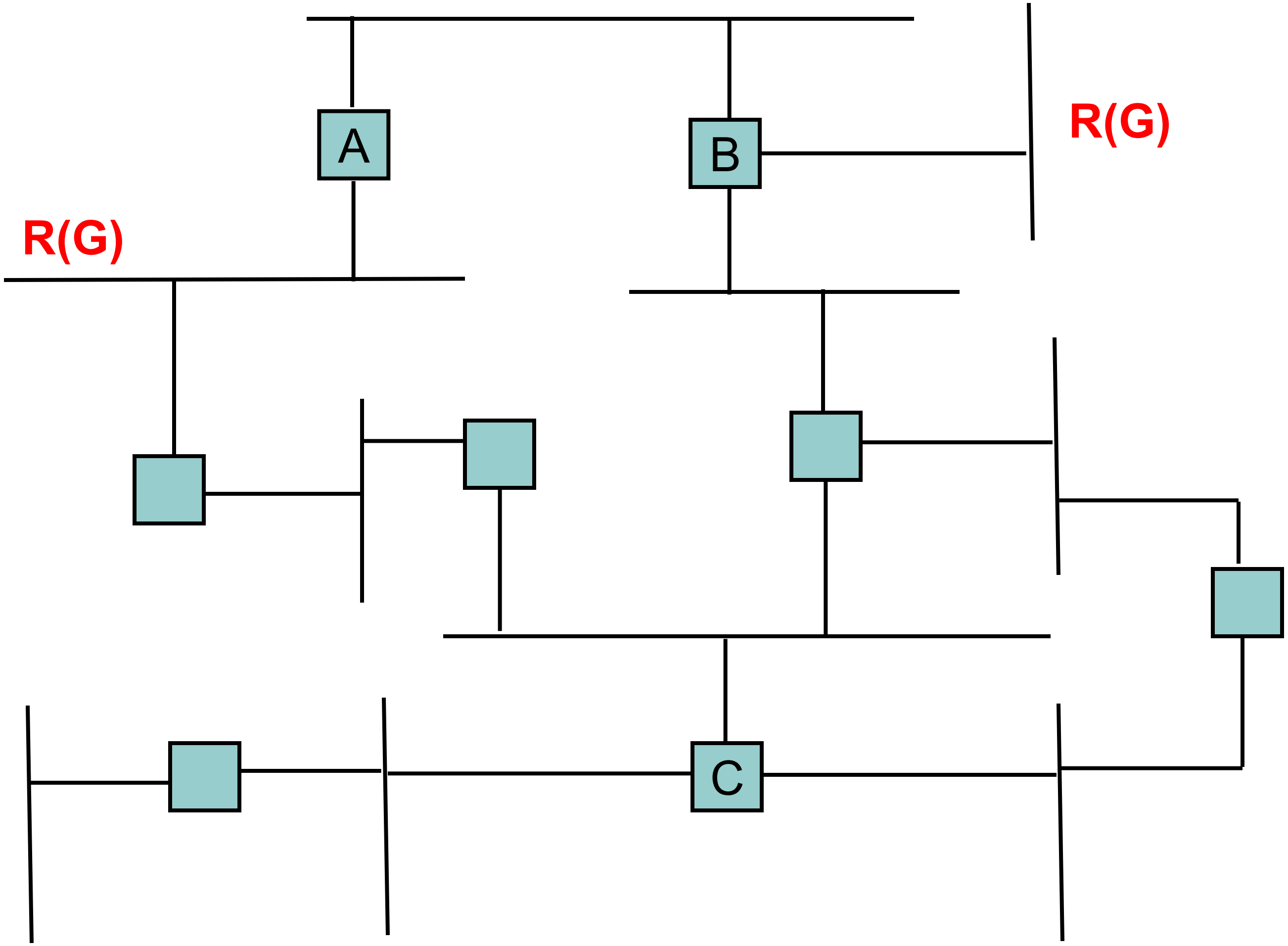
Link-State Multicast Routing

- This is covered in the book.

- Extend OSPF – MOSPF
  - Send group membership information in OSPF link state advertisement (LSA) messages

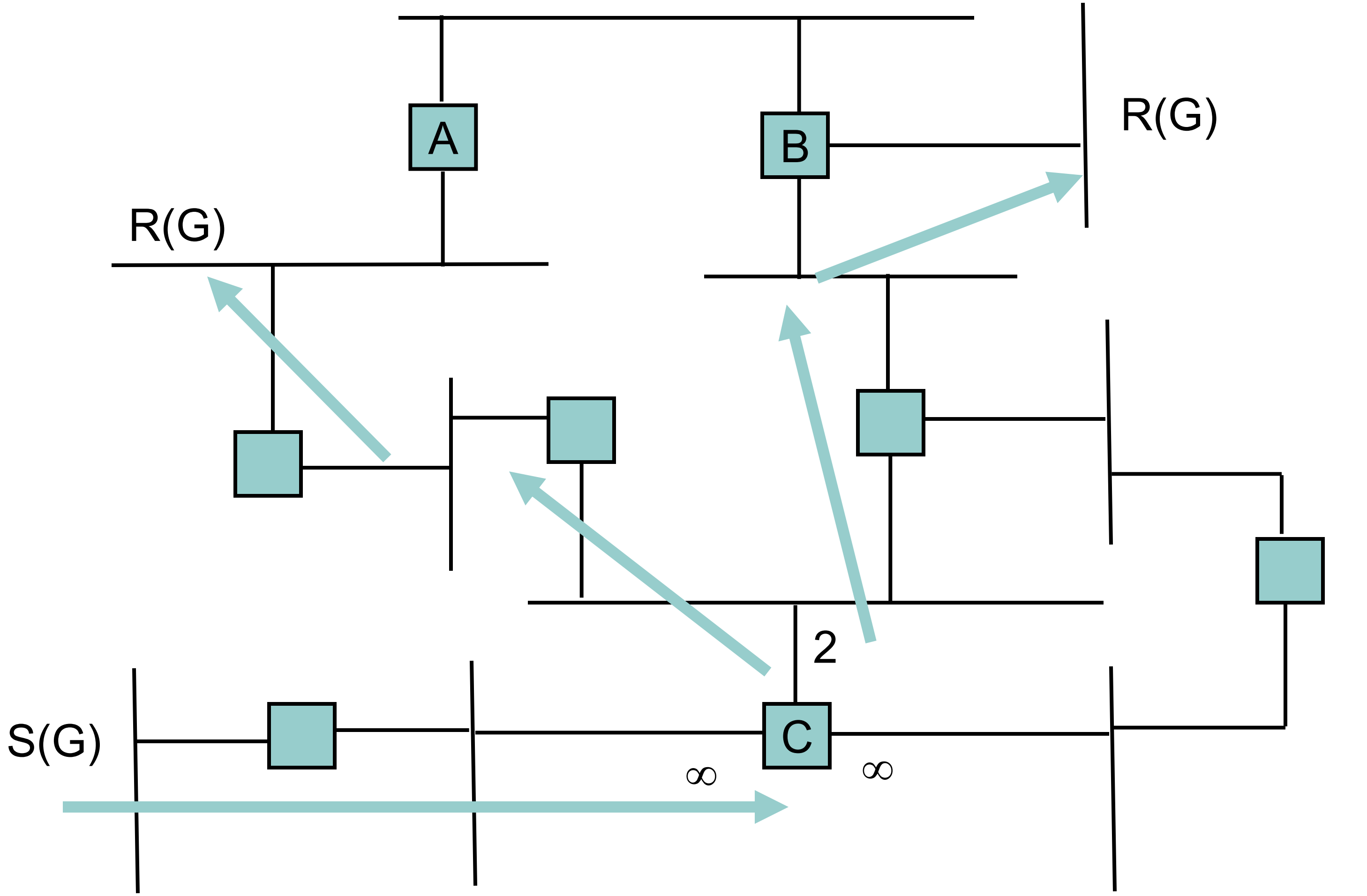
  - I.e., each router learns the entire set of receivers, and their location.

  - Each router can (when necessary) compute the minimum cost path from every source to the current set of receivers of the multicast group.



Routers A and B mention in their LSA that they have receivers in their adjacent LANs.

Hence, C can recreate the above picture in detail.



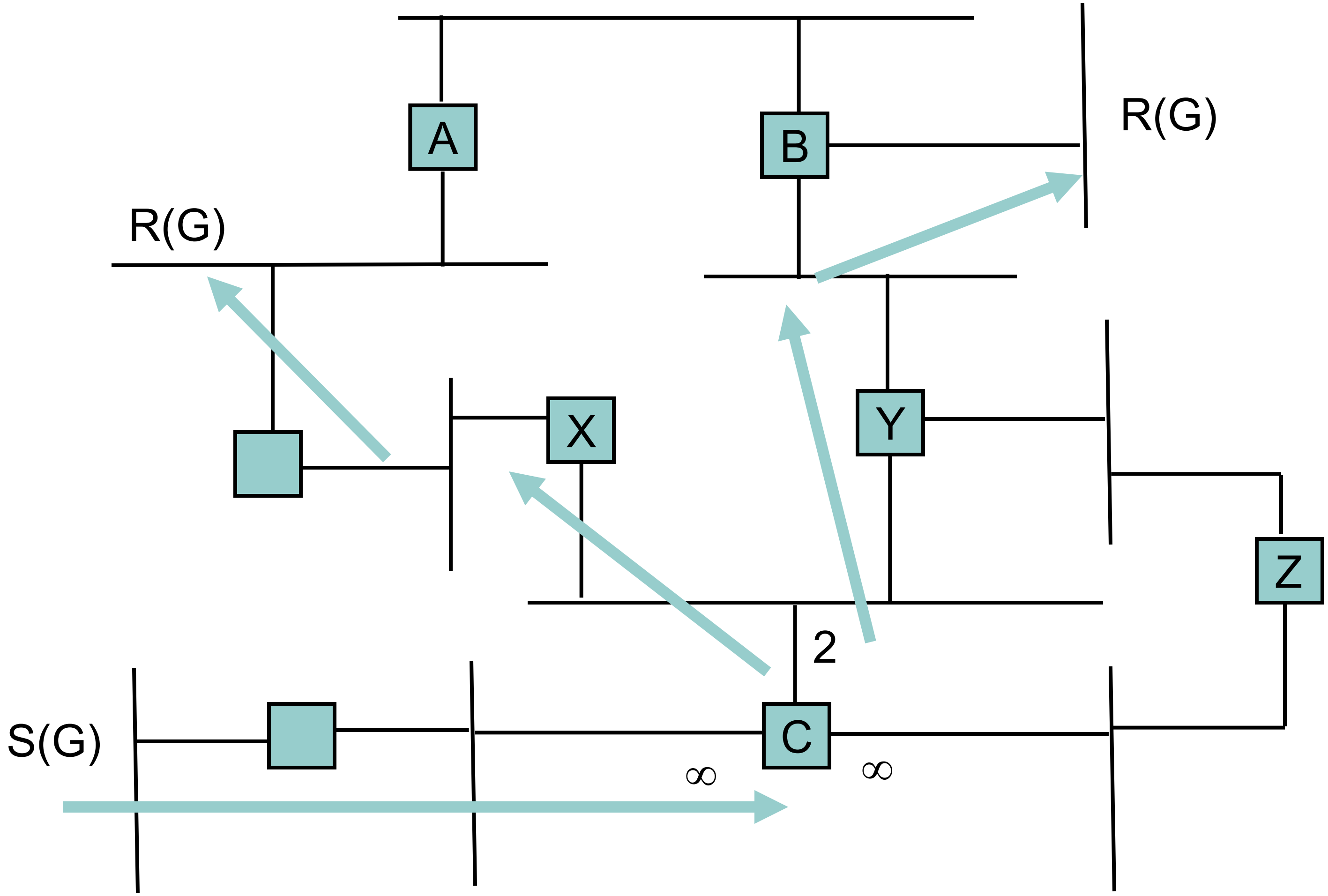
C computes the shortest path tree from S(G) to the receivers
Each link is tagged with its distance to the closest R(G)

How do you know who are the sources?

- You could precompute, for every group G, a tree for every source (LAN) S
- This is way too expensive
- Instead, use caching

MOSPF Cache

- The cache has entries of the following form:
  - (S, G, **iif**, MHV)
  - MHV is a VECTOR with an entry per output link (i.e. a list of pairs (λ,min-hops))
    - For each link, this vector contains the minimum number of hops needed to reach a group member via the link
    - If a link does not reach a group member (not on tree) use infinity for # hops.
  - If a packet is received from S to G from the **iif**,
    - The packet is sent over all links such that the time-to-live of the packet is at least the link's entry in min-hops
- If no cache-entry of (S,G) compute the tree on the fly (incurring delay)
- Cache entries are not timed-out, they are just flushed out if new ones are needed (or when the network graph changes)



Which of X, Y, Z, compute the tree?
Same question, but the receiver at B does not exist.