Module 5
Routing Protocols

Dr. Natarajan Meghanathan
Associate Professor of Computer Science
Jackson State University, Jackson, MS 39232
E-mail: natarajan.meghanathan@jsums.edu

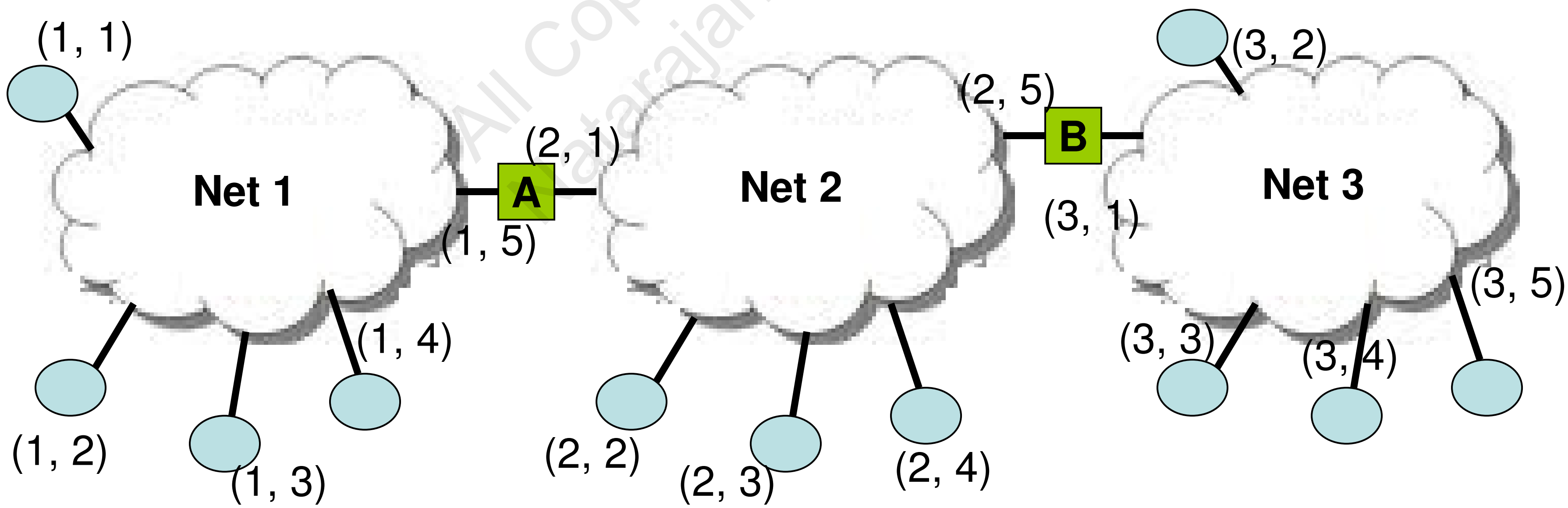
Module Topics

- 5.1 Principles of Routing in the Internet
- 5.2 Distance Vector Routing Protocol
- 5.3 Link State Routing Protocol
- 5.4 Routing across Autonomous Systems
- 5.5 Multicast Routing Protocols

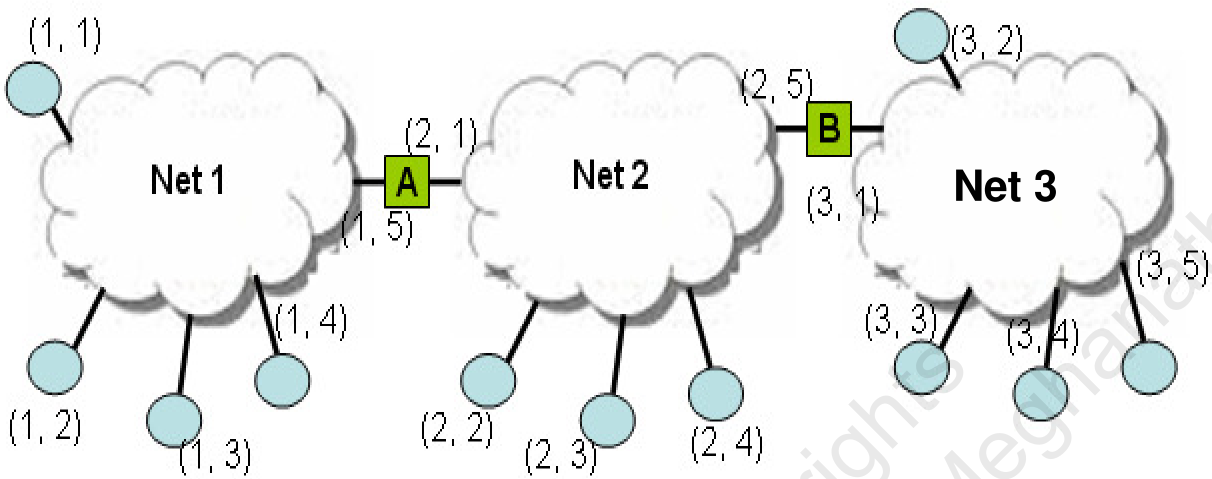
5.1 Principles of Routing in the Internet

A Generic WAN Addressing Model

- For ease of discussion, we will look at IP addressing as a generic WAN addressing model to illustrate the concepts of logical hierarchical addressing and forwarding.
- Each network is identified by a unique network address.
- Each host connected to a WAN is assigned a unique address that has two parts: the first part identifies the network to which the host is connected to and the second part identifies the host attached to that network.
- A fixed number of bits are allocated to each part of the address.



Simple Example for a Routing Table



Routing Table at A (Inefficient Version)

Destination	Next hop
(1, 1)	Local interface (1, 5)
(1, 2)	Local interface (1, 5)
(1, 3)	Local interface (1, 5)
(1, 4)	Local interface (1, 5)
(2, 2)	Local interface (2, 1)
(2, 3)	Local interface (2, 1)
(2, 4)	Local interface (2, 1)
(3, 2)	Remote interface (2, 5)
(3, 3)	Remote interface (2, 5)
(3, 4)	Remote interface (2, 5)
(3, 5)	Remote interface (2, 5)

Routing Table at B (Efficient Version)

Destination	Next hop
(1, *)	Remote interface (2, 1)
(2, *)	Local interface (2, 5)
(3, *)	Local interface (3, 1)

Routing Table at A (Efficient Version)

Destination	Next hop
(1, *)	Local interface (1, 5)
(2, *)	Local interface (2, 1)
(3, *)	Remote interface (2, 5)

Next hop forwarding

- If the destination computer address listed in the packet is not that of any computer attached to one of its interfaces, a router forwards the packet to another router that it believes to be on the path towards the destination computer.
- A router does not store the complete path to each computer in the WAN. All the router needs to know is the next hop router to which a packet addressed to a destination computer must be sent.
- The table used to store the next hop information is called the routing table and the process of forwarding a packet to its next hop is called routing.

Store and Forward Technique

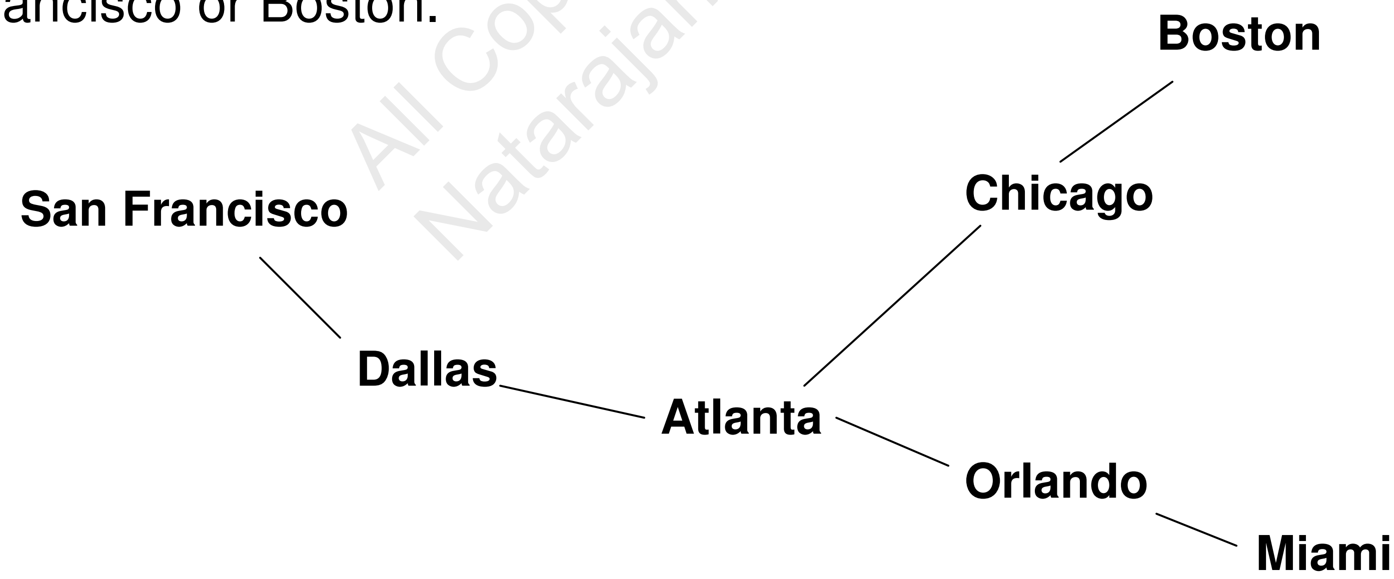
- A WAN permits multiple computers to send packets simultaneously.

- The packet switches employ a "store and forward" technique according to which a packet arriving from a computer / router is stored in a memory buffer for sometime until the processor decides the appropriate output interface through which the packet is to be forwarded so the packet is on its way to its destination router/ computer.

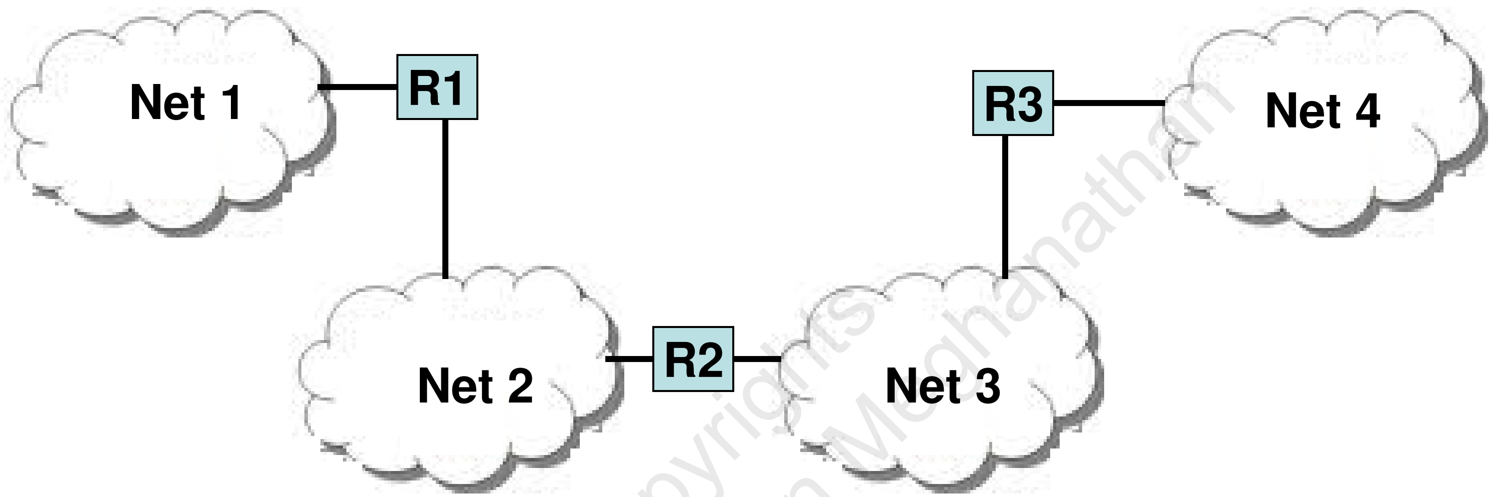
- Once the output interface for a packet is determined, the packet is placed in the queue maintained for that interface. If the queue is empty, the packet is forwarded immediately; otherwise the packet waits in the queue until its turn comes.

Source Independence

- The decision on the next hop to forward a packet at a router does not depend on the path taken to reach the router.
- The next hop of a packet addressed to a particular destination remains the same irrespective of whether the packet came from a computer attached to the network or the packet came across 100 networks before reaching the router.
- Example: The next hop to reach Miami at Atlanta is Orlando, irrespective of whether the packet came to Atlanta from San Francisco or Boston.



Simplifying Routing Tables with Default Routes



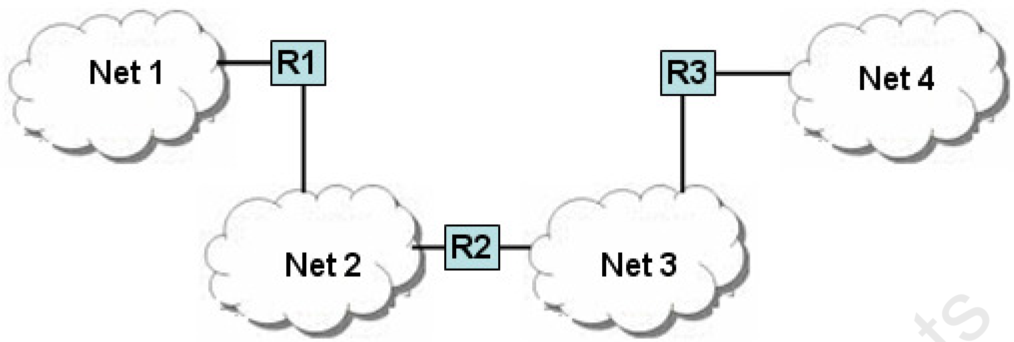
Routing Table at R1

Destination	Next hop
Net 1	Local interface Net 1
Net 2	Local interface Net 2
Net 3	R2
Net 4	R2

Simplified Routing Table at R1 using Default Entries

Destination	Next hop
Net 1	Local interface Net 1
Net 2	Local interface Net 2
*	R2

Simplifying Routing Tables with Default Routes



Note: The default entry (*) has to be the last entry in a routing table as it means that if there is no match in any of the preceding entries, the packet will be forwarded to the default next hop router. If the default entry is any where else in a routing table, then it is possible for a packet to be wrongly forwarded (in case if the appropriate next hop entry for the packet is listed somewhere below the default entry)

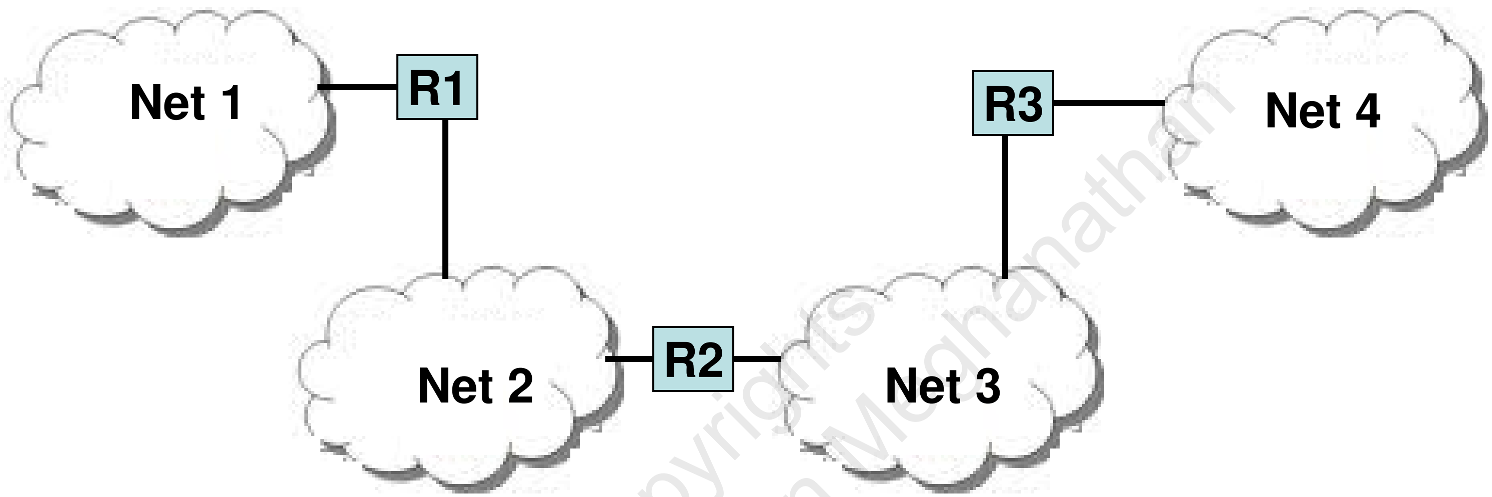
Routing Table at R3

Destination	Next hop
Net 1	R2
Net 2	R2
Net 3	Local interface Net 3
Net 4	Local interface Net 4

Simplified Routing Table at R3 using Default Entries

Destination	Next hop
Net 3	Local interface Net 3
Net 4	Local interface Net 4
*	R2

Simplifying Routing Tables with Default Routes



The Routing Table at R2 cannot be Simplified

Destination	Next hop
Net 1	R1
Net 2	Local interface Net 2
Net 3	Local interface Net 3
Net 4	R3

5.2 Distance Vector Routing

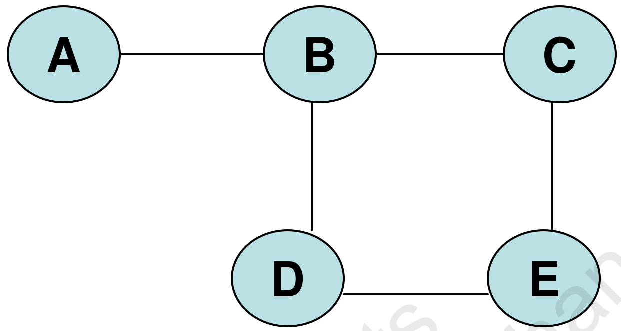
Distance Vector Routing

- Each entry in the routing table of a router has the following tuple, one per destination router: {destination network id, distance to the destination network, the next hop network to reach the destination}.
- A router periodically sends to each of its neighboring routers a routing update message that has the entries {destination network id, distance to destination network} taken from the routing table.
- The neighbors then update their routing table if a better route to any destination network is determined.
- Ties are broken in favor of the network with the lower ID or the lower alphabet (as shown in the examples).

- If hop count is the link weight, the number of routing update messages to be sent by a router before the routing table entries at all routers are filled with the optimal routes is equal to the Diameter of the network

- Diameter of a network (graph) is the maximum value of the shortest hop path between any two nodes

Distance Vector Routing Example 1



Assume time is slotted

Routing Tables at Time Slot 0

A

A	0	-
B	∞	-
C	∞	-
D	∞	-
E	∞	-

B

A	∞	-
B	0	-
C	∞	-
D	∞	-
E	∞	-

C

A	∞	-
B	∞	-
C	0	-
D	∞	-
E	∞	-

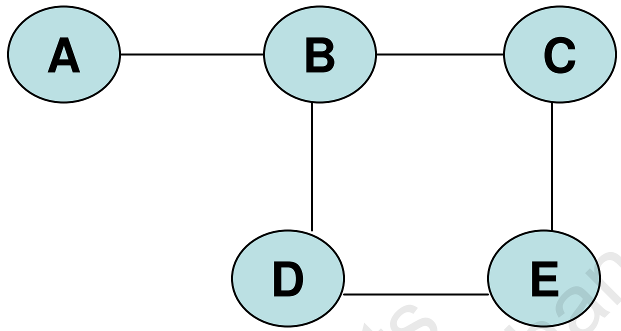
D

A	∞	-
B	∞	-
C	∞	-
D	0	-
E	∞	-

E

A	∞	-
B	∞	-
C	∞	-
D	∞	-
E	0	-

Distance Vector Routing Example 1



Assume time is slotted

Routing Tables at Time Slot 1

A

A	0	-
B	1	B
C	∞	-
D	∞	-
E	∞	-

B

A	1	A
B	0	-
C	1	C
D	1	D
E	∞	-

C

A	∞	-
B	1	B
C	0	-
D	∞	-
E	1	E

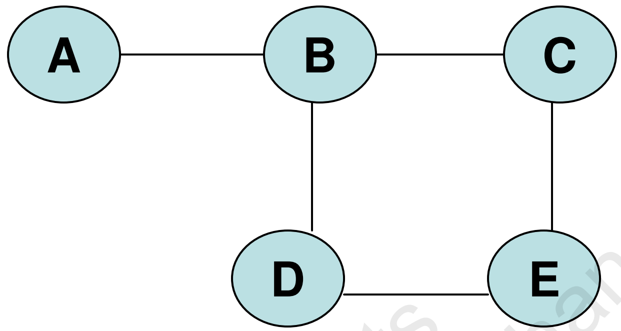
D

A	∞	-
B	1	B
C	∞	-
D	0	-
E	1	E

E

A	∞	-
B	∞	-
C	1	C
D	1	D
E	0	-

Distance Vector Routing Example 1



Assume time is slotted

Routing Tables at Time Slot 2

A

A	0	-
B	1	B
C	2	B
D	2	B
E	∞	-

B

A	1	A
B	0	-
C	1	C
D	1	D
E	2	C

C

A	2	B
B	1	B
C	0	-
D	2	B
E	1	E

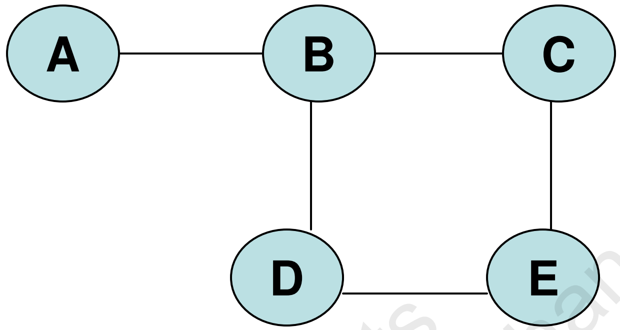
D

A	2	B
B	1	B
C	2	B
D	0	-
E	1	E

E

A	∞	-
B	2	C
C	1	C
D	1	D
E	0	-

Distance Vector Routing Example 1



Diameter = 3
The maximum of the Shortest paths hop count

Assume time is slotted

Routing Tables at Time Slot 3

A

A	0	-
B	1	B
C	2	B
D	2	B
E	3	B

B

A	1	A
B	0	-
C	1	C
D	1	D
E	2	C

C

A	2	B
B	1	B
C	0	-
D	2	B
E	1	E

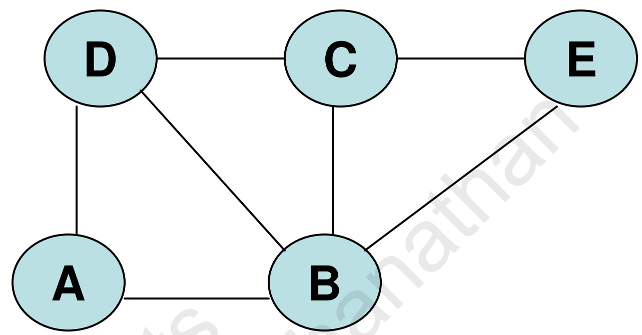
D

A	2	B
B	1	B
C	2	B
D	0	-
E	1	E

E

A	3	C
B	2	C
C	1	C
D	1	D
E	0	-

Distance Vector Routing Example 2



Assume time is slotted

Routing Tables at Time Slot 0

A

A	0	-
B	∞	-
C	∞	-
D	∞	-
E	∞	-

B

A	∞	-
B	0	-
C	∞	-
D	∞	-
E	∞	-

C

A	∞	-
B	∞	-
C	0	-
D	∞	-
E	∞	-

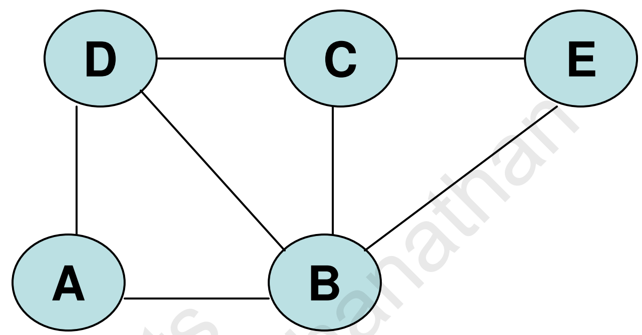
D

A	∞	-
B	∞	-
C	∞	-
D	0	-
E	∞	-

E

A	∞	-
B	∞	-
C	∞	-
D	∞	-
E	0	-

Distance Vector Routing Example 2



Assume time is slotted

Routing Tables at Time Slot 1

A

A	0	-
B	1	B
C	∞	-
D	1	D
E	∞	-

B

A	1	A
B	0	-
C	1	C
D	1	D
E	1	E

C

A	∞	-
B	1	B
C	0	-
D	1	D
E	1	E

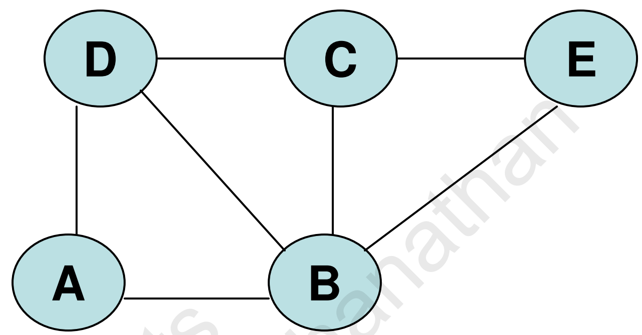
D

A	1	A
B	1	B
C	1	C
D	0	-
E	∞	-

E

A	∞	-
B	1	B
C	1	C
D	∞	-
E	0	-

Distance Vector Routing Example 2



Assume time is slotted

Routing Tables at Time Slot 2

A

A	0	-
B	1	B
C	2	B
D	1	D
E	2	B

B

A	1	A
B	0	-
C	1	C
D	1	D
E	1	E

C

A	2	B
B	1	B
C	0	-
D	1	D
E	1	E

D

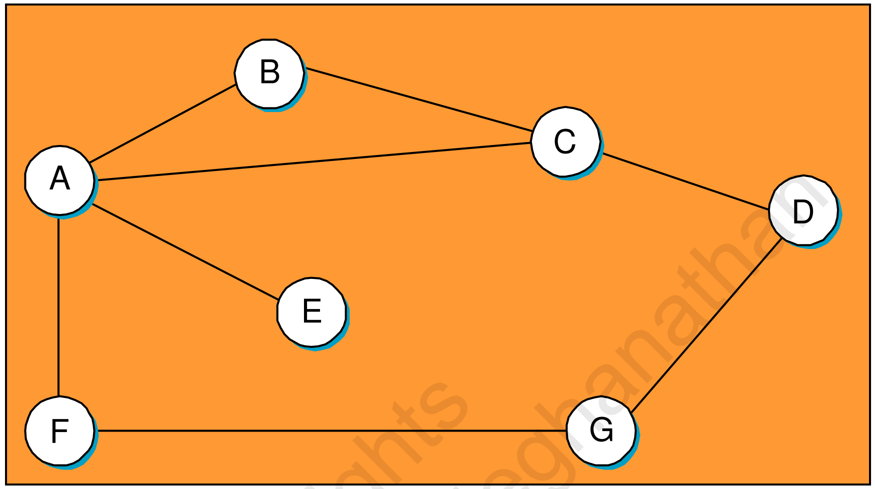
A	1	A
B	1	B
C	1	C
D	0	-
E	2	B

E

A	2	B
B	1	B
C	1	C
D	2	B
E	0	-

Diameter = 2

Distance Vector Routing Example 3



Dest. Node	A		B		C		D		E		F		G	
	Dist.	Next Hop	Dist.	Next Hop	Dist.	Next Hop	Dist.	Next Hop	Dist.	Next Hop	Dist.	Next Hop	Dist.	Next Hop
A	0	-	1	A	1	A	2	C	1	A	1	A	2	F
B	1	B	0	-	1	B	2	C	2	A	2	A	3	F
C	1	C	1	C	0	-	1	C	2	A	2	A	2	D
D	2	C	2	C	1	D	0	-	3	A	3	A	1	D
E	1	E	2	A	2	A	3	C	0	0	2	A	3	A
F	1	F	2	A	2	A	2	G	2	A	0	-	1	F
G	2	F	3	A	2	D	1	G	3	A	1	G	0	-

Global View of Routing Tables at All Nodes

Routing Updates

- Periodic update: Each node periodically sends an update message (containing the distance to reach each node in the network) to its neighbors. This lets the other nodes to know that the node is still running and also update any potential route change.

- Triggered update: Whenever a node's routing table changes, it sends an update to its neighbors, which may lead to a change in their tables and so on.

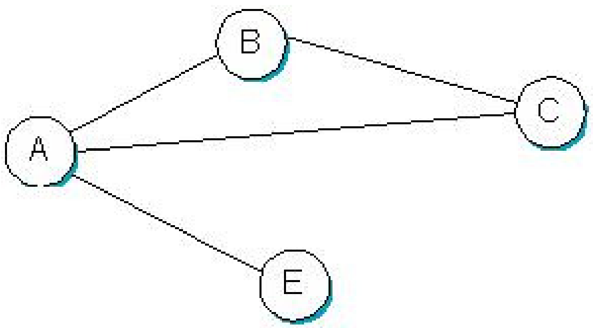
- Detecting failures of neighbor nodes:
- A node continually tests another node by sending a control packet and seeing if it receives an acknowledgement.
- A node determines that the link is down, if it does not receive the expected periodic routing update from the neighbor node for a few update cycles.

Handling Link Failures

- Suppose the link F-G fails. F sends the info (G,∞) as part of its next periodic update to A.
- A on receiving the update from F, updates the distance to reach G (in its routing table) to ∞.
- At the same time, C sends an update to A that it can reach G at at cost of 2. So, A now updates that it can reach G at a cost of 3 through C (which is better than ∞).
- A advertises (G, 3) to F. F now updates the cost to reach G as 4 and the next hop node is A.
- <u>Updated Routing Table (Global View)</u>

Dest. Node	A		B		C		D		E		F		G	
	Dist.	Next Hop	Dist.	Next Hop	Dist.	Next Hop	Dist.	Next Hop	Dist.	Next Hop	Dist.	Next Hop	Dist.	Next Hop
A	0	-	1	A	1	A	2	C	1	A	1	A	2	F
B	1	B	0	-	1	B	2	C	2	A	2	A	3	F
C	1	C	1	C	0	-	1	C	2	A	2	A	2	D
D	2	C	2	C	1	D	0	-	3	A	3	A	1	D
E	1	E	2	A	2	A	3	C	0	0	2	A	3	A
F	1	F	2	A	2	A	2	G	2	A	0	-	1	F
G	3	C	3	C	2	D	1	G	4	A	4	A	0	-

Count-to-Infinity Problem



Dest. Node	A		B		C		E	
	Distance	Next Hop	Distance	Next Hop	Distance	Next Hop	Distance	Next Hop
A	0	-	1	A	1	A	1	A
B	1	B	0	0	1	B	2	A
C	1	C	1	C	0	-	2	A
E	1	E	2	A	2	A	0	-

- Suppose the link A-E fails.
- A sends an update of (E, ∞) to B and C. At the same time, C sends (E, 2) to A and B.
- A on receiving the update from C, updates the cost to E to be 3 and the next hop to C. A advertises (E, 3) to C.
- As A is the next hop node for C to reach E, C has to update the cost to reach E to be 4 and the next hop stays to be A. C sends (E, 5) to A.
- As C is the next hop node for A to reach E, A has to update the cost to reach E to be 6 and the next hop node stays to be C. This will continue forever.

(Partial) Solutions to the Count-to-Infinity Problem

- Upper bound on the number of hops: If the maximum number of hops to reach a certain node is more than 16, then the node is no longer reachable.
  - Used by the Routing Information Protocol (RIP): An implementation of the Distance Vector Routing protocol for the Internet.

- Split Horizon: When a node sends a routing update to its neighbors, it does not send those routes it learned from each neighbor back to that neighbor.

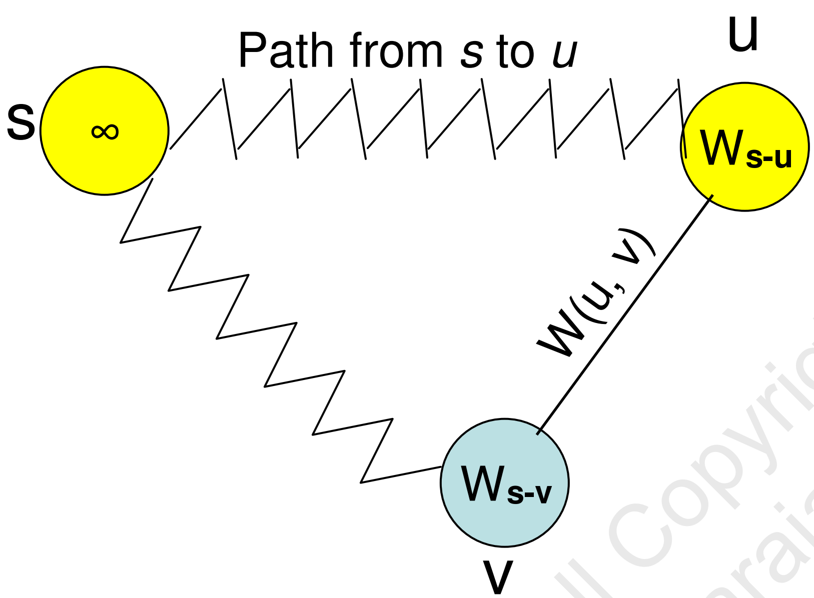
- Split Horizon with Poison Reverse: A node sends the routing update including the routes learned from its neighbor back to that neighbor, but the cost of the route is advertised to be ∞.

5.3 Link State Routing

Link-state Routing

- Each packet switch/router monitors the status of a link with its neighboring switches and broadcasts it as a message (called the link status message) to all switches.
- Each entry in the link status message broadcasted by a switch consists of the following: {switch address, weight of the link to the switch}
- Each switch collects the incoming link status messages and uses them to build a graph of the network. In other words, after a sequence of link status message exchanges, each switch has the global view of the network stored in its local memory.
- Each switch then applies the Dijkstra shortest path algorithm (or its variations like the largest bottleneck path algorithm) on the graph and computes a shortest path tree rooted at itself.
- When a packet arrives at a switch X that needs to be forwarded to a destination switch Y, the switch X checks its shortest path tree and finds the next hop switch Z that lies on the shortest path from X to Y. The packet is then forwarded to Z.

Relaxation Principle for Largest Bottleneck Path Algorithm



Principle in a nutshell

During the beginning of each iteration we will pick a vertex *u* that has the largest bottleneck weight path from *s*. We will then explore the neighbors of *u* for which we have not yet found the largest bottleneck weight path. We will try to see if by going through *u*, we can Increase the weight of the largest bottleneck path from *s* to *v*, where *v* is a neighbor of *u*.

Relaxation Condition

If $W_{s-v} < \text{Min}(W_{s-u}, W(u, v))$ then
  $W_{s-v} = \text{Min}(W_{s-u}, W(u, v))$
  Predecessor (v) = u
else
  Retain the current path from *s* to *v*

Modified Dijkstra Algorithm

```
Begin Algorithm Modified-Dijkstra (G, s)
    For each vertex v Є V – {s}
        d [v] ← -∞ // an estimate of the largest bottleneck path from s to v
    End For
    d [s] ← ∞
    S ← Φ  // set of nodes for which we know the largest bottleneck path from s
    Q ←  V // set of nodes for which we know estimate of largest bottleneck path
      from s
    While Q ≠ Φ
        u ← EXTRACT-MAX(Q)
        S ← S U {u}
        For each vertex v such that (u, v) Є E
            If d [v] < Min (d [u], w (u, v) ) then
                d [v] ← Min (d [u], w (u, v) )
                Predecessor (v) = u
            End If
        End For
    End While
End Modified-Dijkstra
```

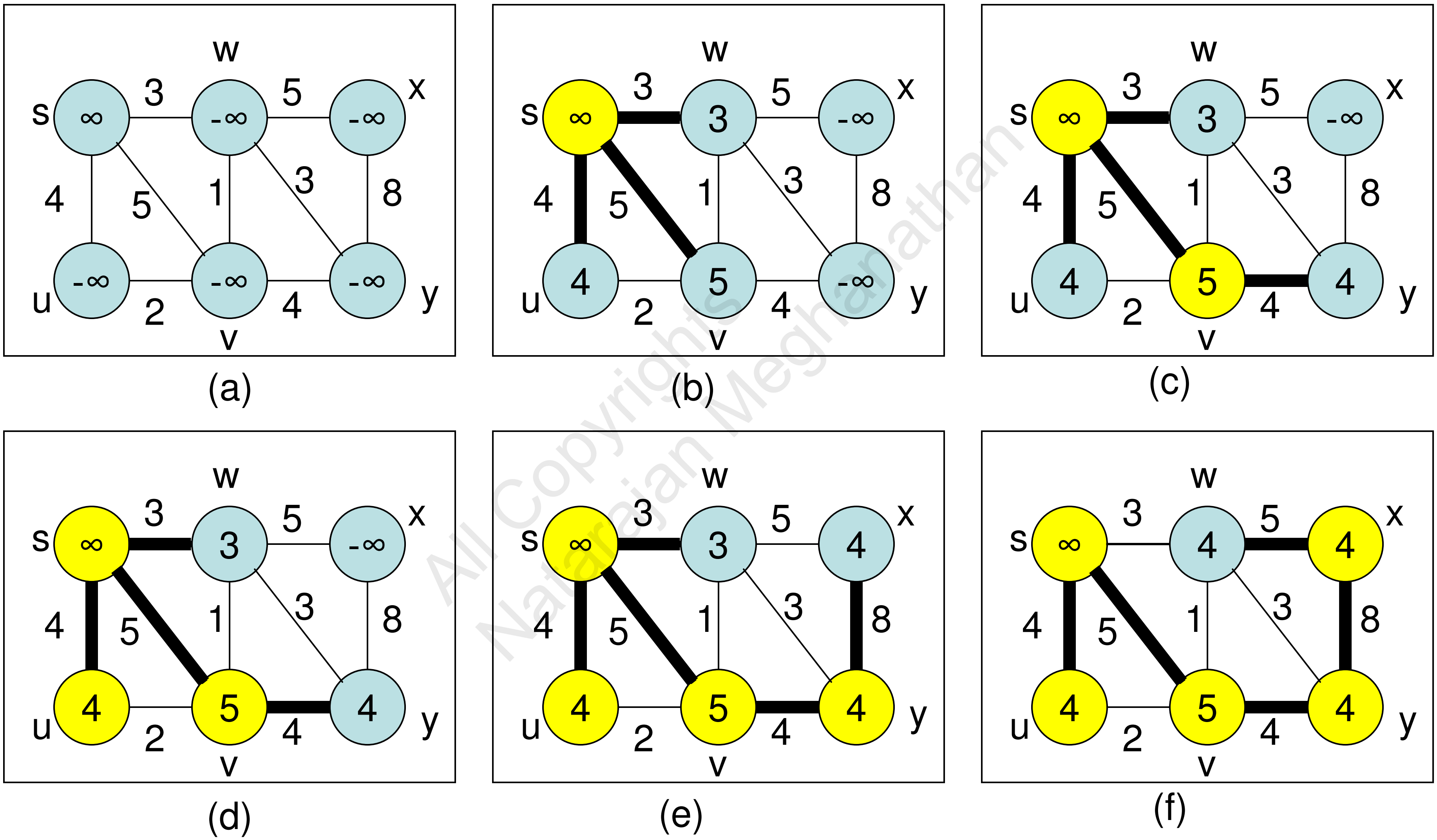

Largest Bottleneck Path Problem

The bottleneck weight of a path is the minimum weight of the constituent edges on the path. The Largest Bottleneck Path Problem is the problem of finding the path with the largest bottleneck weight

$$\underset{\forall p \in P}{Max}\left[\underset{\forall e \in p}{Min}(weight(e))\right]$$

Example: Find the path with the largest bandwidth. The bandwidth of a path is the minimum of the bandwidth of the constituent links

Largest Bottleneck Path Problem



5.4 Routing across Autonomous Systems

Autonomous System

- Term Autonomous System (AS) to specify groups of routers
- One can think of an AS as a contiguous set of networks and routers all under control of one administrative authority
- There is no exact meaning for administrative authority
- The term is sufficiently flexible to accommodate many possibilities
  - For example, an AS can correspond to an ISP, an entire corporation, or a university
  - Alternatively, a large organization with multiple sites may choose to define one AS for each site
  - In particular, each ISP is usually a single AS, but it is possible for a large ISP to divide itself into multiple ASs
- The choice of AS size can be made for
  - economic, technical, or administrative reasons

Types of Internet Routing Protocols

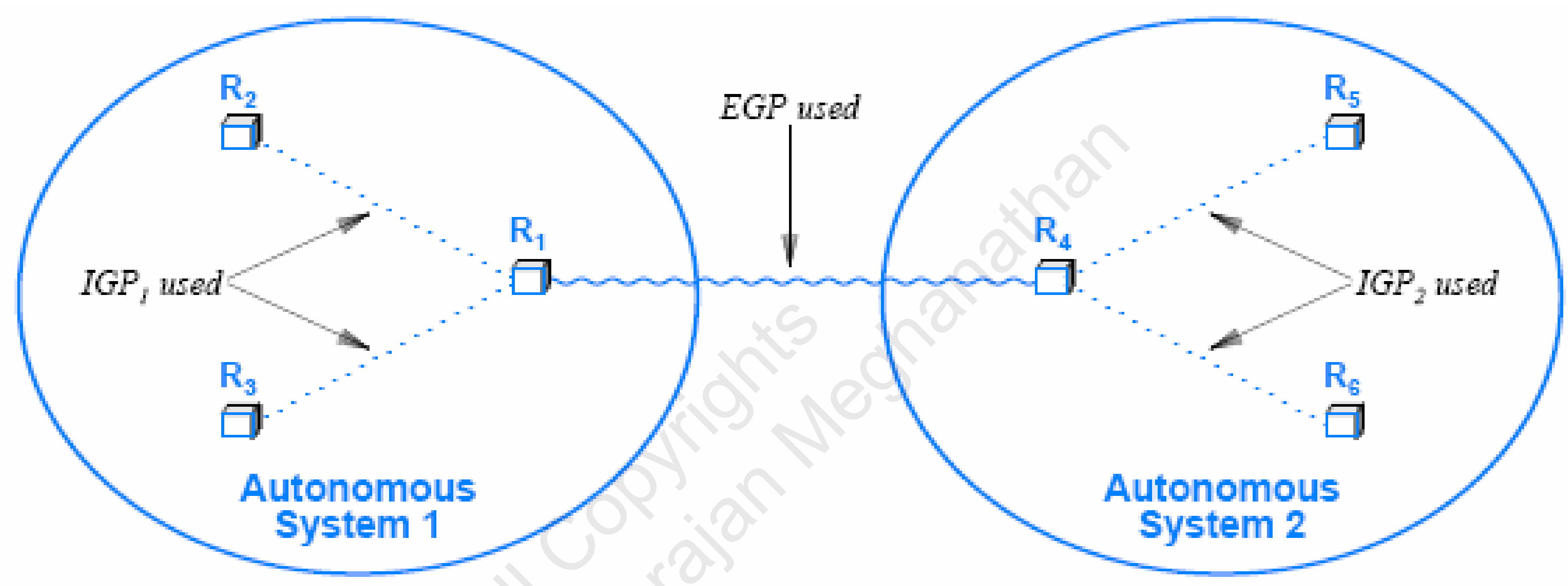
- All Internet routing protocols are divided into two major categories:
  - Interior Gateway Protocols (IGPs)
  - Exterior Gateway Protocols (EGPs)

- <u>Interior Gateway Protocols (IGPs)</u>
- Routers within an AS use an IGP exchange routing information
- Several IGPs are available
  - each AS is free to choose its own IGP
    - Examples: Distance Vector Routing, Link State Routing protocols
- Usually, an IGP is easy to install and operate
- IGP may limit the size or routing complexity of an AS

Types of Internet Routing Protocols

- Exterior Gateway Protocols (EGPs)
- A router in one AS uses an EGP to exchange routing information with a router in another AS
- EGPs are more complex to install and operate than IGPs
  - but EGPs offer more flexibility and lower overhead (i.e., less traffic)
- To save traffic
  - an EGP summarizes routing information from an AS before passing it to another AS
- An EGP implements policy constraints
  - that allow a system manager to determine exactly what information is released outside the organization
- Example: Border Gateway Protocol (BGP)

Types of Internet Routing Protocols



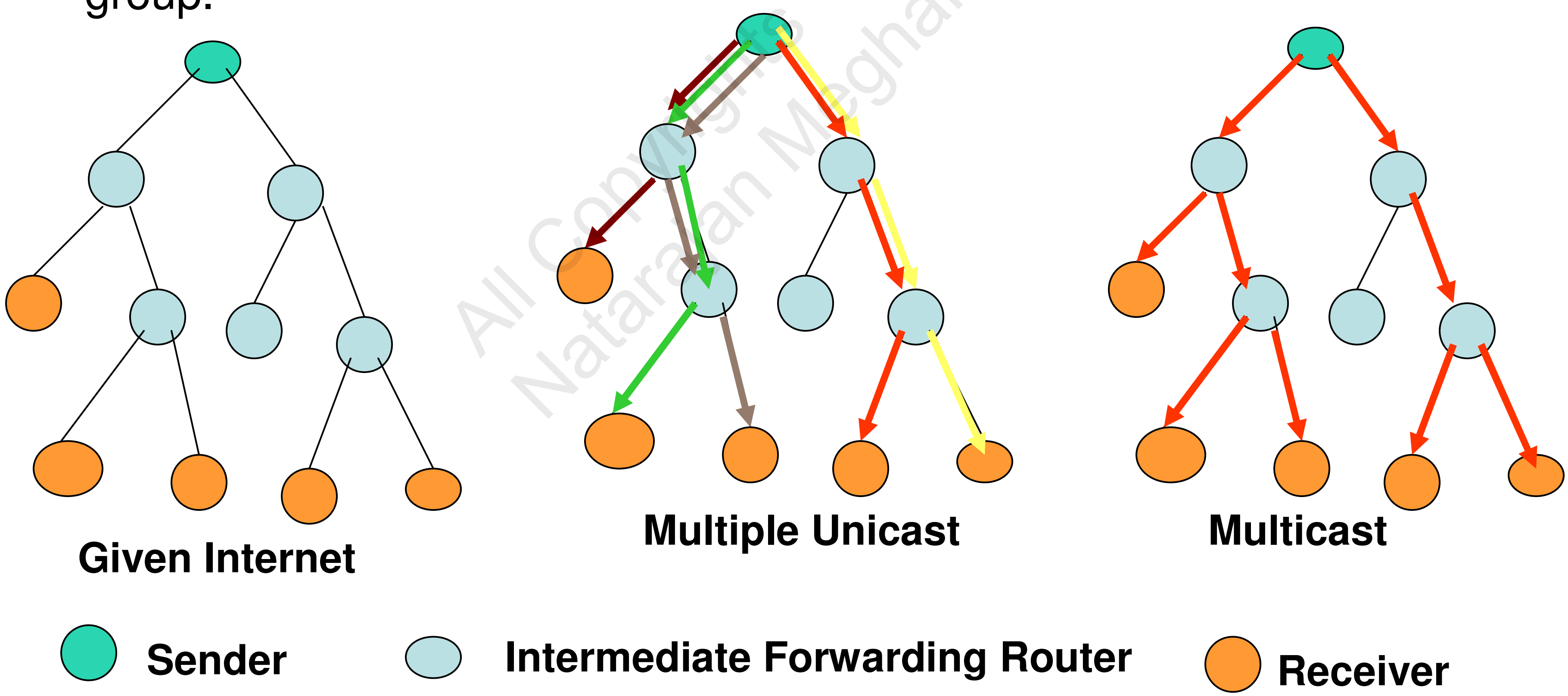
Types of Internet Routing Protocols

- IGPs and EGPs differ in an important way with respect to routing metrics:
  - IGPs use routing metrics, but EGPs do not
  - Each AS chooses a routing metric and arranges internal routing software to send the metric with each route so receiver can use the metric to choose optimal paths
- Outside an AS, an EGP does not attempt to choose an optimal path
  - Instead, the EGP merely finds a path
- Each AS is free to choose a routing metric
- An EGP cannot make meaningful comparisons
  - Suppose one AS reports the number of hops along a path to destination D and another AS reports the throughput along a different path to D
  - An EGP that receives the two reports cannot choose which of the two paths has least cost because there is no way to convert from hops to throughput
- Thus, an EGP can only report the existence of a path and not its cost

5.5 Multicast Routing Protocols

Multicasting Routing Protocols

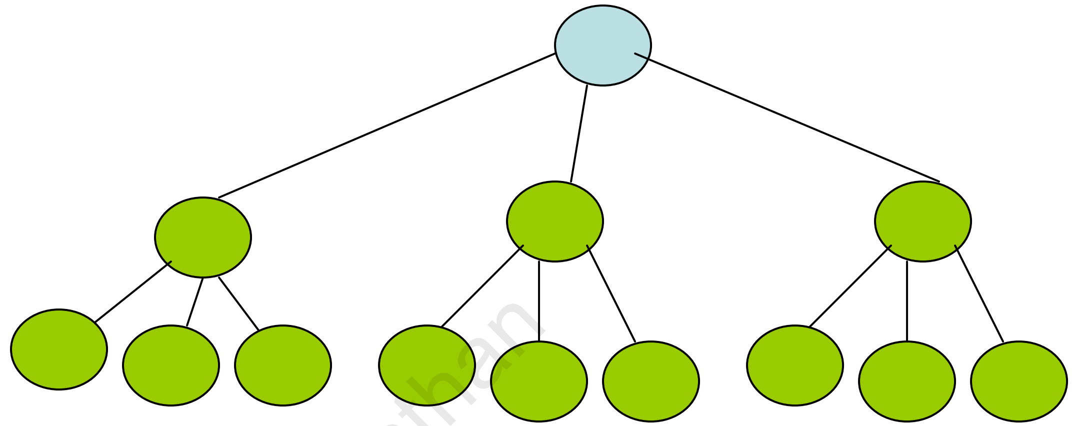
- **Basic Principle:**
- Instead of sending the same copy of the multicast packet individually addressed to each member of the multicast group, it would be efficient to use a multicast address as the target address of the multicast packet and have the packet forwarded by the routers to all the members of the group.



Numerical Problem: Multicasting

- Assume a full *k*-ary rooted multicast tree of height h where each intermediate node has exactly *k*-children and the root node is at height 0.
- If all the leaf nodes of this tree are part of the multicast group and the root node is the sender, compute the number of link transmissions involved in multicasting vs. multiple unicasting.

#Links used for multicast transmission: $k^1 + k^2 + \dots + k^h$



$$= [1 + k^1 + k^2 + \dots + k^h] - 1$$

$$= \frac{k^{(h+1)} - 1}{k - 1} - 1$$

\# Leaf nodes = k^h

\# Links used per unicast transmission = h

\# Links used for multiple unicast transmissions = hk^h.

h = 10
k = 3
\# Link transmissions in multiple unicasting = 590490
\# Link transmissions in multicasting = 88572

Internet Multicast Routing

Dynamic Group Membership – Internet Group Multicast Protocol (IGMP)

- An application can join a multicast group at any time and remain in the group for an arbitrary time.
- When an application in a computer decides to join a multicast group, the computer takes up the responsibility of becoming a member of the multicast group and informs a local router about the multicast group to which the it wants to join.
- The router manages to receive packets addressed to the multicast group and then forwards the packets to the computer.
- If multiple applications running in the computer participate in the multicast group, the computer still receives only one copy from the local router. The computer makes a copy of the received message to each of the applications that have joined the multicast group.
- An application leave the multicast group at any time. The computer periodically updates the local router and keeps its membership to the multicast group active. If all the applications running in a computer quit the multicast group, then the computer informs the local router not to forward any more messages belonging to the particular multicast group.

Internet Multicast Routing

Anonymous Multicast Group

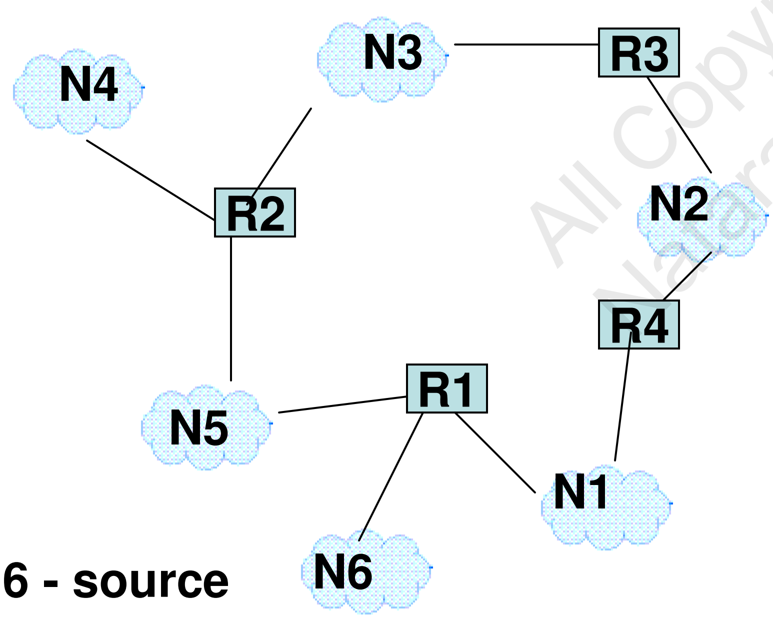
- The identity or the number of group members will be known neither to the sender nor the receiver.
- Any arbitrary application can send a datagram to a multicast group at any time. An application need not be a member of the group to send messages to the group.

Forward and Discovery Techniques

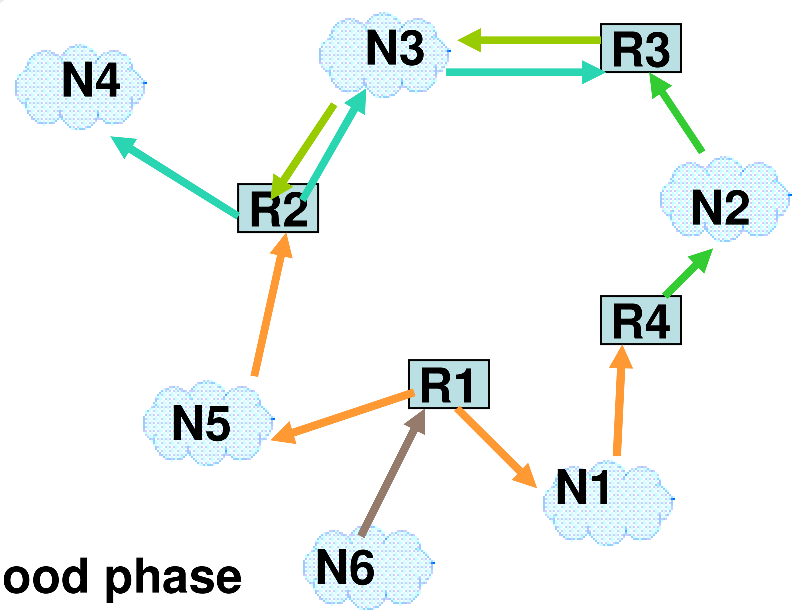
- It is the responsibility of the routers to propagate multicast routing information and arrange for the multicast group messages to reach them so that they can forward the messages to the local hosts that are the members of the group.
- Finding an optimal forwarding structure is one of the hardest task of a multicast routing protocol as it has to consider both the dynamically joining group members and the anonymous senders.

Flood and Prune

- The Flood-and-prune approach of forwarding and discovery is appropriate when the multicast group is small and all the members are attached to local area networks that are contiguous.
- **Flood phase:** Each router forwards the datagram it receives on all directly attached LANs (via hardware multicast), except the LAN from which the datagram was received. If a router receives a multicast packet it has already seen, it drops the packet (assume duplicate multicast packets can be identified using sequence numbers)



N6 - source



Flood phase

Flood and Prune

- Consider the source of the multicast group is in network N6. It sends the multicast packet to router R1.
- Assume there is a multicast receiver in networks N3 and N4.

- **Explanation of Flood Phase:**
- R1 sends the packet across networks N1 and N5 to routers R4 and R2 respectively.
- R2 sends the packet to N4 and to R3 via N3. At the same time, R4 sends the packet to R3 via N2.
- Assume R3 processes the packet from R4 prior to processing the packet from R2. Hence, R3 forwards the packet to network N3, which has already received the multicast packet through R2. Also, R3 forwards the packet to R2 as part of the multicast across N3. Thus R3 is also disturbed to process a duplicate packet.

- How to avoid multiple routers attached to a LAN flooding the packets to the LAN?
- We want only router attached to a LAN to broadcast the packet if at all any host in the LAN is a member of the multicast group.

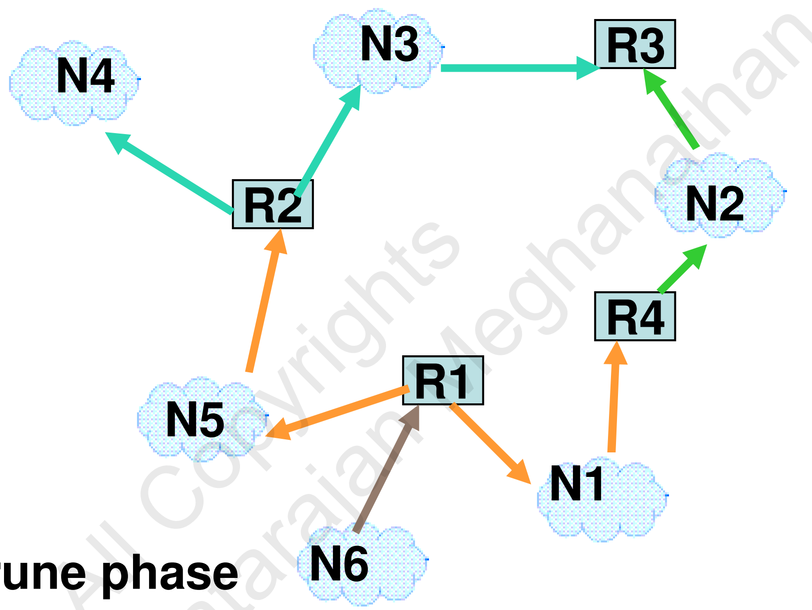
Prune Phase: Reverse Path Broadcast

- Assume, the routers have run a unicast routing protocol (RIP or OSPF), and know the next hop router on the shortest path to the source.
- For each source network and LAN, there can be only router that can broadcast the multicast packet from the source onto the LAN.
- A router forwards a multicast packet over an attached LAN, only if the router is on the shortest path from the source to the LAN. Such a router is called the “parent router” for the <source network, attached network> pair.
- If there are multiple routers that lie on the shortest path from the source to the LAN, the tie is broken using the router id.

- In the example, R3 can figure out that it is not on the shortest path from the source network N6 to the network N3. Hence, it does not forward any multicast packet coming from N6 to N3.
- R2 determines that it is the parent router for <N6, N3>. Hence, it forwards the multicast packet from N6 to N3.

- Reverse Path Broadcast (RPB) essentially implements broadcasting across the shortest path tree rooted at the source.

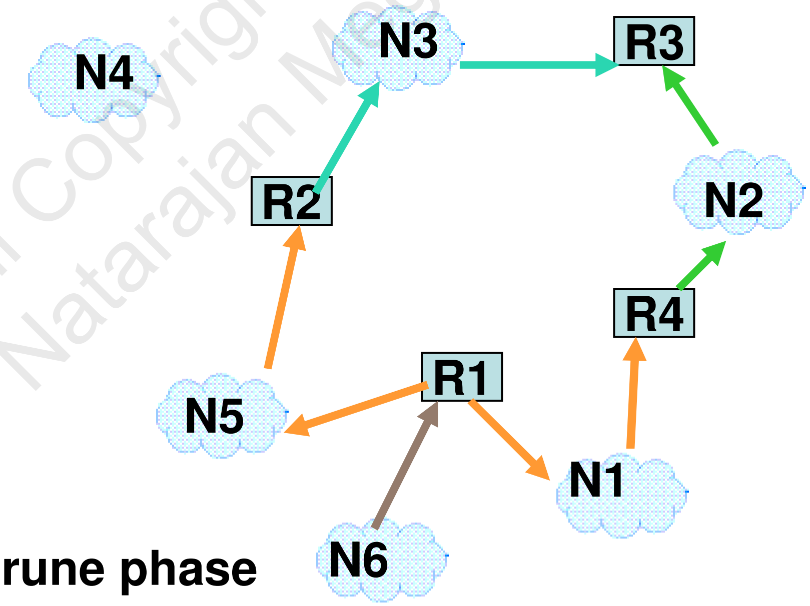
Prune Phase: Reverse Path Broadcast



**Prune phase
RPB**

Prune Phase: Reverse Path Multicast

- A further optimization can be achieved if a router learns that no hosts on an attached network are members of the group, then the router stops forwarding the multicast packet to that network. This prone technique is called Reverse Path Multicast (RPM).
- For example, consider the situation of the hosts in network N4 withdraw from the membership of the multicast group. R2 no longer forwards the multicast packet N4.



Prune phase
RPM

Multicast Extension to OSPF (MOSPF)

- Include the list of groups on each link in the link status message broadcast by each router periodically.
- Each router constructs the global topology locally, applies the Dijkstra algorithm such that the tree is rooted at the source of the multicast group rather than router itself.
- Once a shortest path tree rooted at the source is determined, the router prunes the tree using the multicast group information in the link status messages.
- For a given source and multicast group, if the router is still a forwarding node in the tree, it forwards the multicast packet along the outgoing edges in the tree.

Protocol Independent Multicast Sparse Mode (PIM-SM)

- PIM-SM is a multicast routing protocol used for efficient multicast across geographically dispersed multicast groups.
- MOSPF heavily depends on the underlying OSPF protocol; PIM-SM uses the routing information used by the underlying unicast routing protocol, but it does not depend on a specific unicast routing protocol.

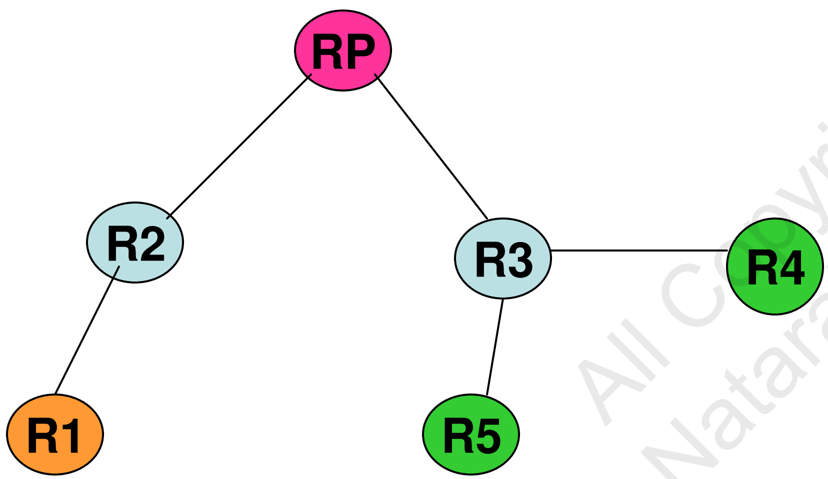
- In PIM-SM, routers need to explicitly send Join and Prune messages to join a multicast group. These messages are sent to a Core unicast address and the corresponding router is called the Rendezvous point (RP).
- For a given domain, all the routers of the domain agree on the RP.

Protocol Independent Multicast Sparse Mode (PIM-SM)

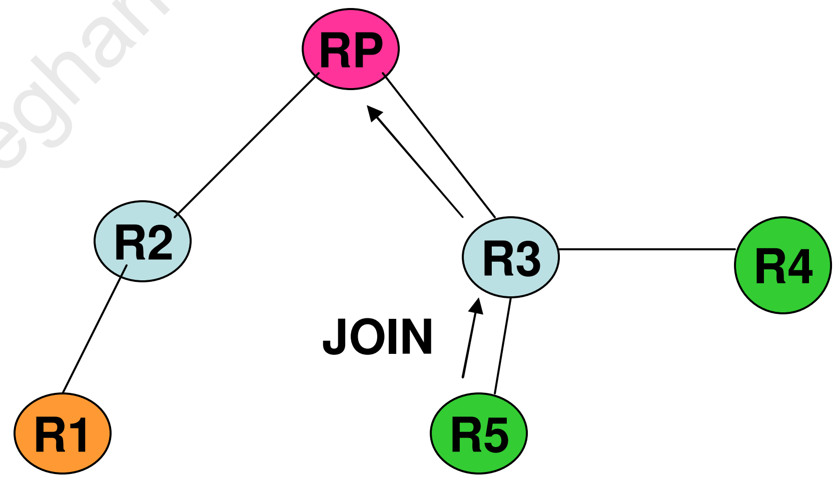
- To join a multicast group G, a router sends a Join message (unicast IP) to the RP.
- A Join message may pass through one or more routers before reaching the RP.
- Each router receiving the Join message, creates a forwarding table entry for the mutlicast tree, called a (*, G) entry. The interface on which the Join arrived is included as the interface across which a multicast packet addressed for G should be forwarded.
- Using the routing table information constructed as part of the underlying unicast routing protocol, the router then decides the next hop router to forward the Join message so that it reaches the RP on the shortest path.
- Eventually, the Join message arrives at the RP, completing the construction of the tree branch.
- When another router sends a Join message towards the RP for group G, the Join message need not travel the entire path towards the RP. When the Join message reaches an intermediate router that has already created a forwarding table entry for group G, will merely include the interface on which the Join arrived to the list of interfaces to which a multicast packet needs to be forwarded.

Protocol Independent Multicast Sparse Mode (PIM-SM)

- We basically get a shared multicast tree rooted at the RP.
- A source node for the multicast group G, sends the multicast packet using IP-in-IP encapsulation as a unicast packet to the RP and the packet is then forwarded by the RP across the shared multicast tree.

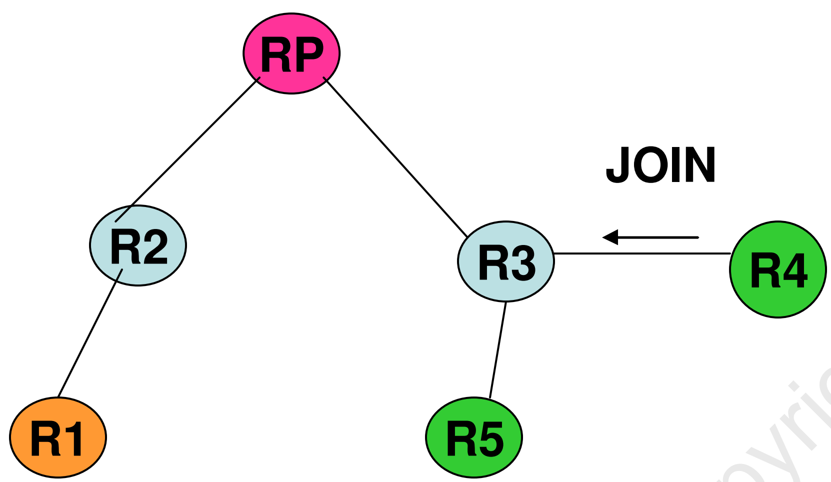


Given Internet
R1 is the source for multicast group G;
R4 and R5 are receivers;
RP is the Rendezvous point for G



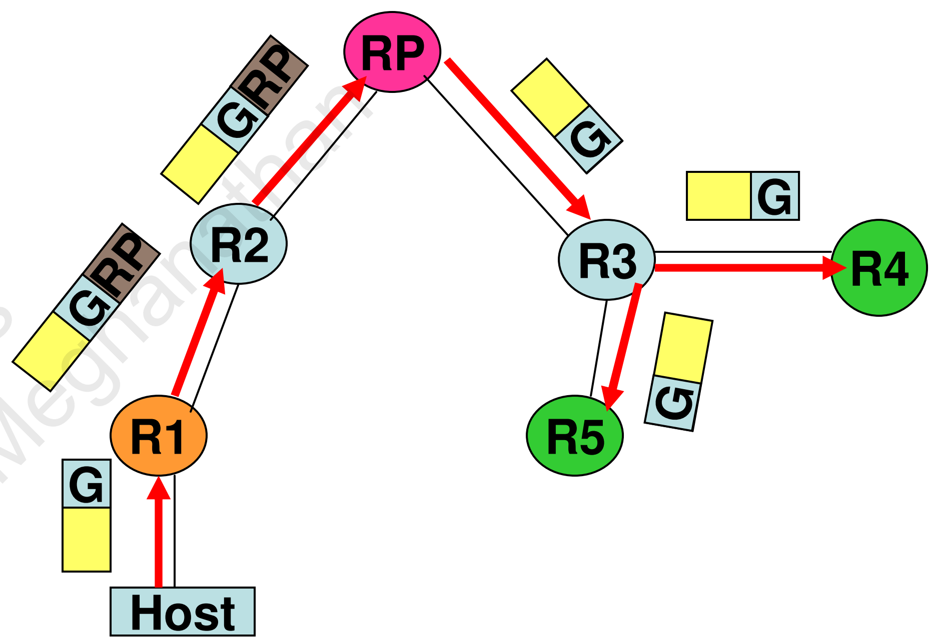
R5 sends a JOIN to RP for group G
Multicast forwarding entry is created in R3 for G.

Protocol Independent Multicast Sparse Mode (PIM-SM)



R4 sends a JOIN to RP for group G

R3 adds the R3-R4 link to the list of Interfaces to forward the multicast packet for G but does not forward the JOIN further.



R1 sends the multicast packet for G to RP using IP-in-IP tunneling.
RP decapsulates the IP datagram and forwards the multicast packet across the shared tree to routers R4 and R5.