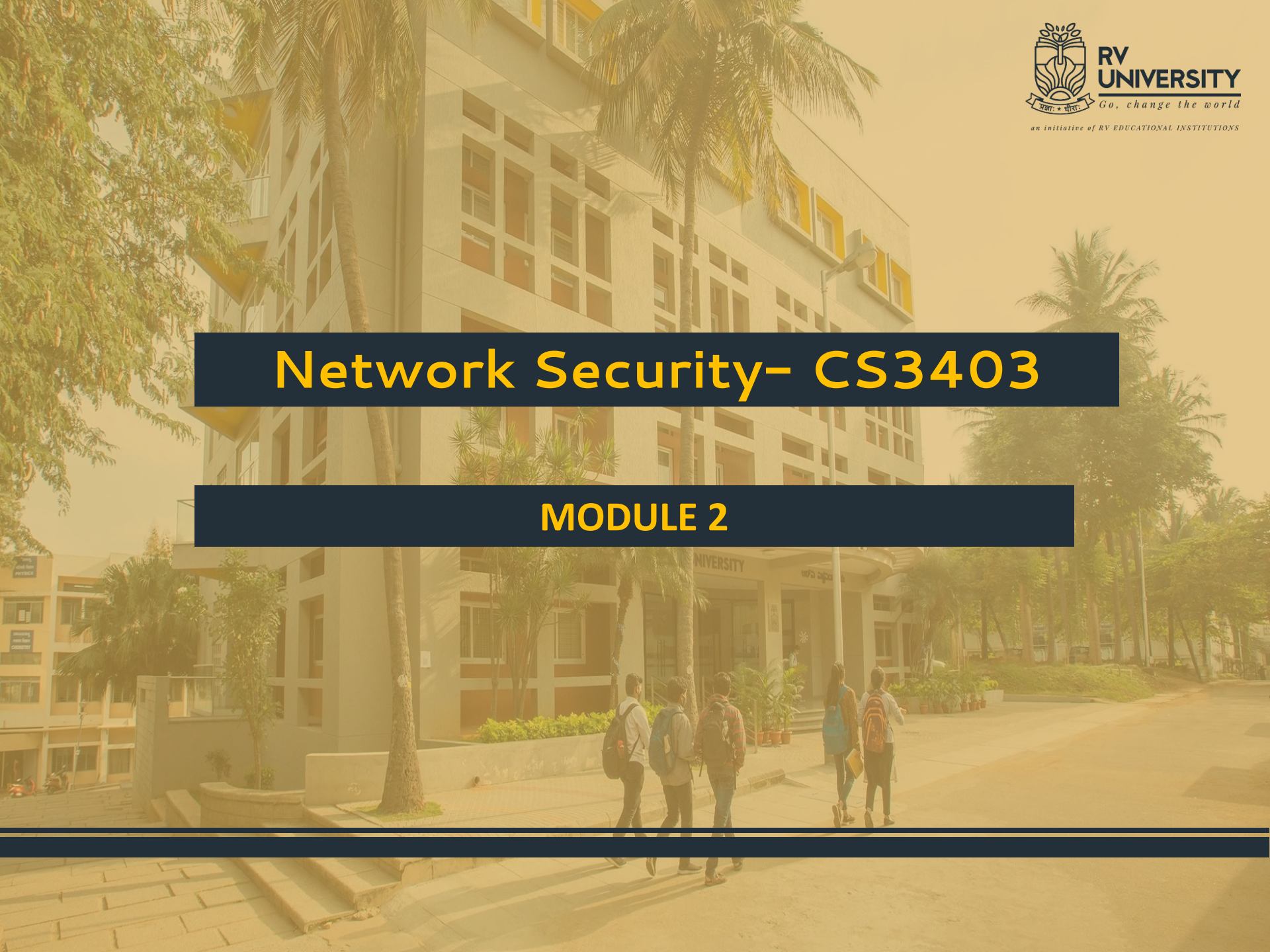


RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

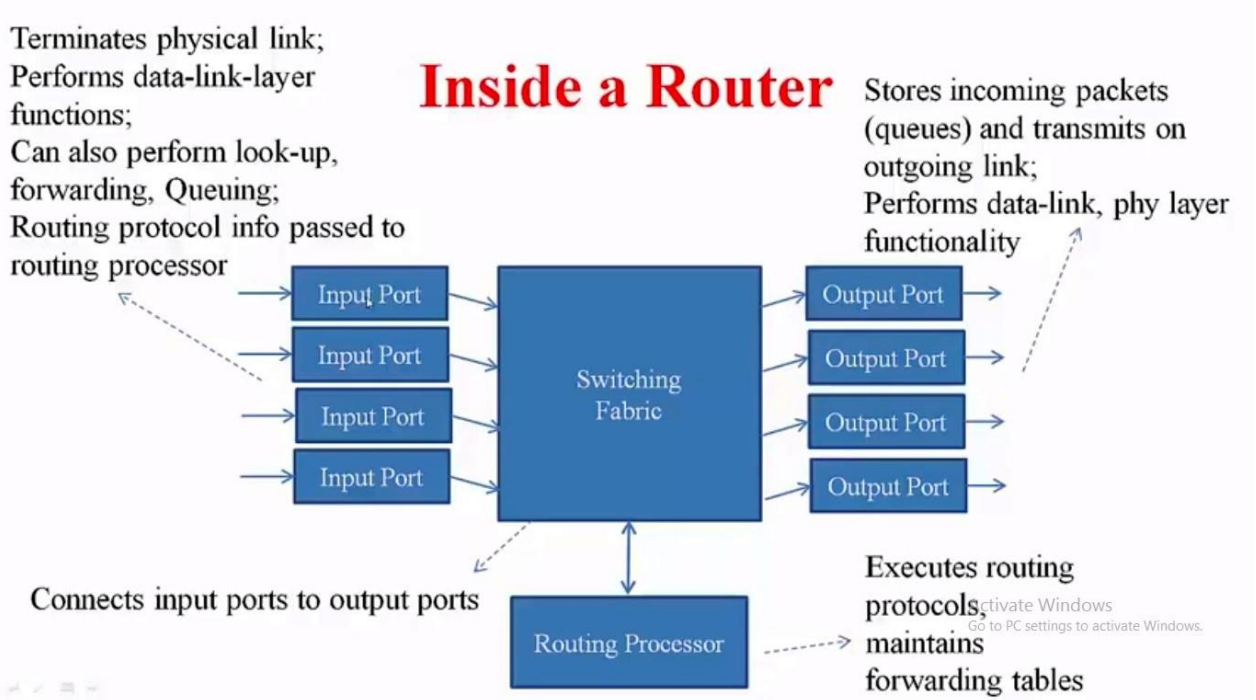
Network Security- CS3403

MODULE 2

Forwarding Table

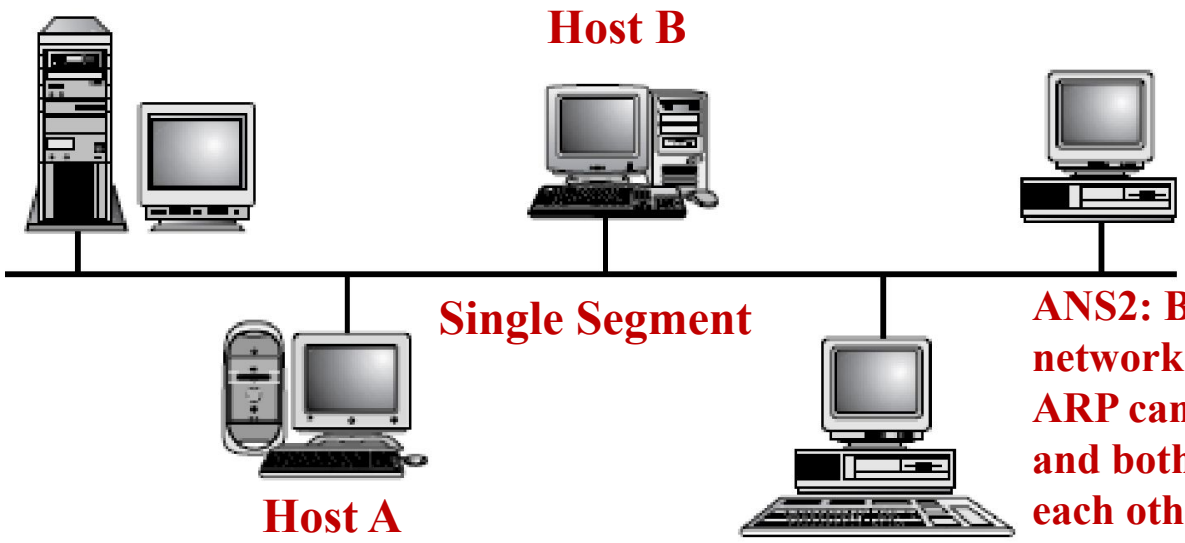
- So far, we have assumed that the switches and routers have enough knowledge of the network topology so they can choose the right port onto which each packet should be output.
- In datagram networks, including IP networks, routing is an issue for every packet.
- In either case, a switch or router needs to be able to look at a destination address and then to determine which of the output ports is the best choice to get a packet to that address.
- The switch makes this decision by consulting a forwarding table.
- The fundamental problem of routing is how switches and routers acquire the information in their forwarding tables.

Inside a Router



- Every frame entering the router through one of the input ports, gets routed through an output port.
- The router looks at the destination IP address in L3 layer and chooses one of the output ports based on the information stored in routing table
- Changes the MAC addresses in L2 layer and forwards it on to output port

Basic Routing Concepts



Q1. For Host A and Host B to communicate with each other, is there a need for a Router? **ANS1: No**

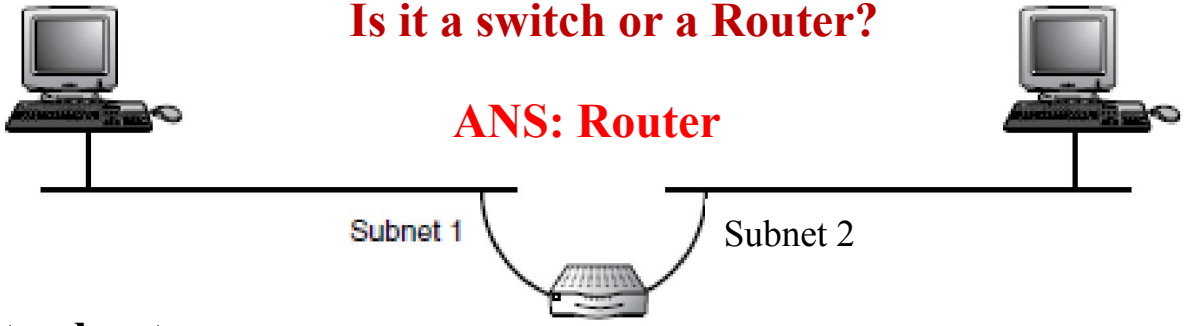
Q2. Why is it not needed?

ANS2: Because both the hosts are on the same network. If the IP addresses of them are known ARP can be used to get their MAC addresses and both the hosts can communicate with each other without any routers involved.

There is no routing involved here, only forwarding.

- **Routing** involves the **delivery of datagrams** between **end systems** located on **different networks.**
- **Without routers** and **routing protocols**, end **host communication** would be **limited** to **only** those **systems** on the **same physical segment**.

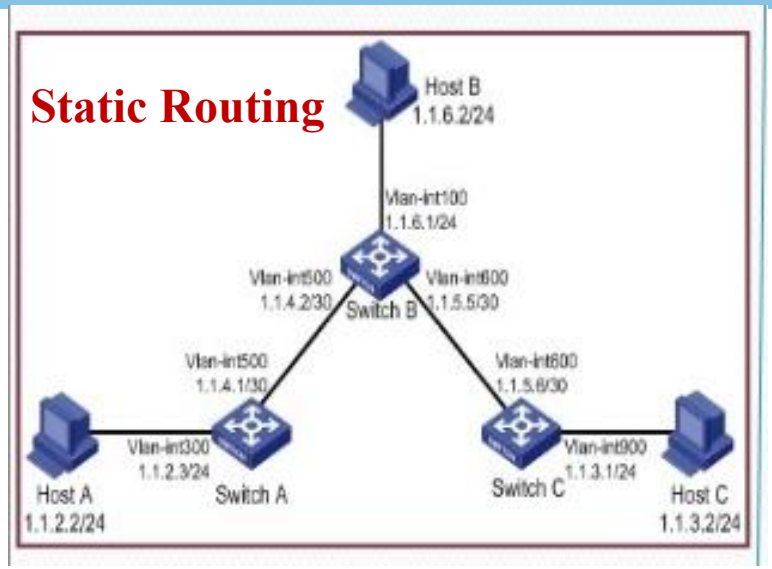
Communication over Multiple Segments

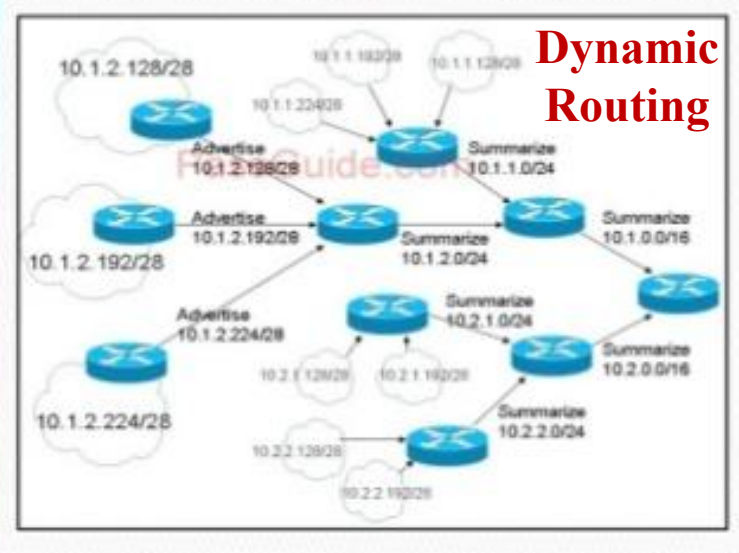


Note: **Different subnets** are **different networks**.

- **Routers** provide the **physical connection between networks**.
- **Routers** must be **configured** with some type of routing mechanism to **enable communication** between hosts **beyond their local segments**.
- Routers **connect multiple subnets** together allowing **remote hosts** to **communicate**.
- Above, the router forwards traffic between hosts on subnets 1 and 2.

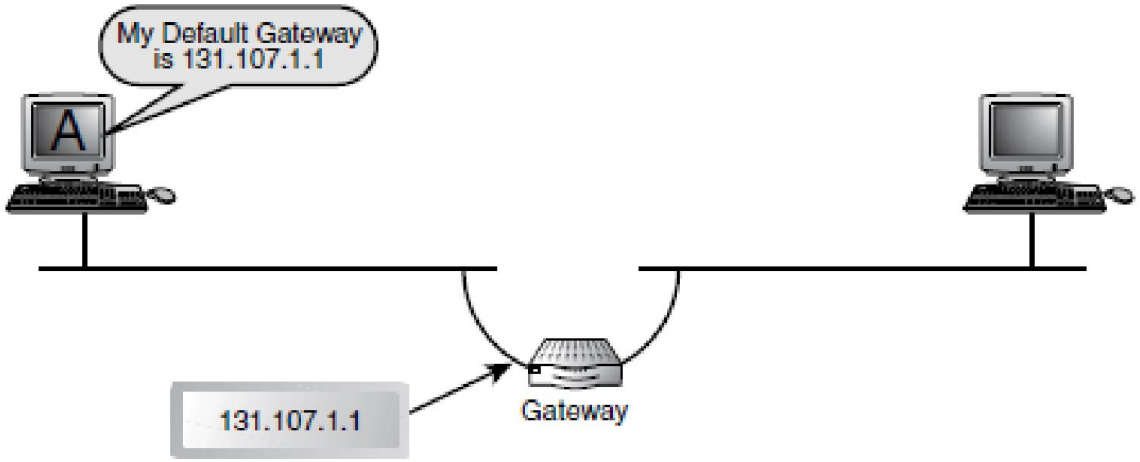
Static Vs Dynamic Routing





- **Routing mechanisms** are either **static** or **dynamic** in nature.
- Static means manual configuration of routing table entries.
  - Possible on small networks
- Dynamic mechanisms involve routing protocols that facilitate the exchange of information, allowing routers to learn and adapt to changes in a network's topology and update their routing tables
  - Using routing protocols (RIP, OSPF, BGP etc.)

Routing: Static and Dynamic



How does it dynamically find default router's IP address?

ANS: DHCP

- Whether a router is configured statically or dynamically or a combination of both the objective is the same, to facilitate communication between remote hosts.
- For hosts to communicate with other hosts located on different networks, end systems must be configured with the IP address of at least one local router (also referred to as the **default router**).
- **Hosts** may be **statically configured** or **dynamically discover** their **local router's IP address**.

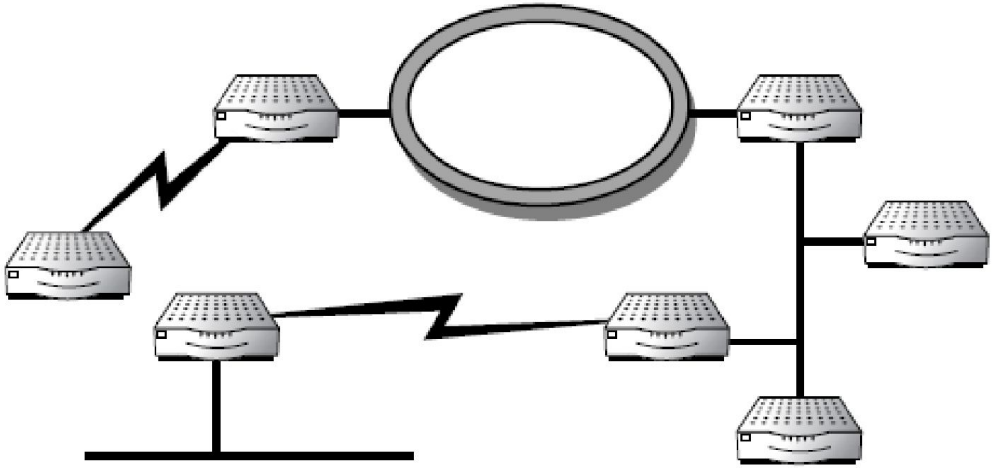
Route Table

- All routers must have a local route table.
- Routers use different routing mechanisms to build and maintain a table known as a route table (also referred to as a forwarding database). - We will study them shortly.
- Several routing mechanisms exist (directly connected, static, default and dynamic).
- These **mechanisms** serve as route table **input sources** providing a router with network and subnet information necessary **to build** and **maintain** the **route table**.
- No matter what the source, the end result is the same. The router builds a table that identifies known **networks** (**cities**) and **subnets** (**streets**).
  - If we **compare** computer **network system** with **postal system**, **networks** correspond to **cities** and **streets** in **those cities** correspond to **subnets**.
  - Remember, the **network ID** part of the **IP address** will be in the **same range** for **all the subnets** of a **network**.
  - A router which understands subnet masks, can identify the subnets too.

Route Table: Multiple Entries for the same Destination

- If **multiple paths exist** to a **destination**, **more than one route** may be **included** in the **route table**.
- Typically, when more than one path to a destination exists, one path needs to be selected (as the best path) by the routing protocol and placed in the route table.
- This would be the primary (active) path the router would use to forward the traffic to that destination.
- However, some routing protocols support **load balancing** across multiple paths.
- In those cases, more than one path could become active for the destination and entries of both the paths are placed in the route table.
- Then, all the active paths could then be used by the router while forwarding the IP datagrams, thus balancing the traffic load across those paths.

Autonomous Systems or Routing Domain

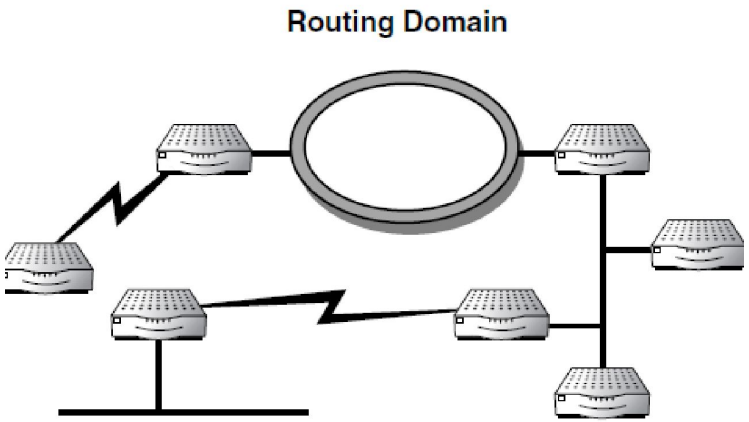


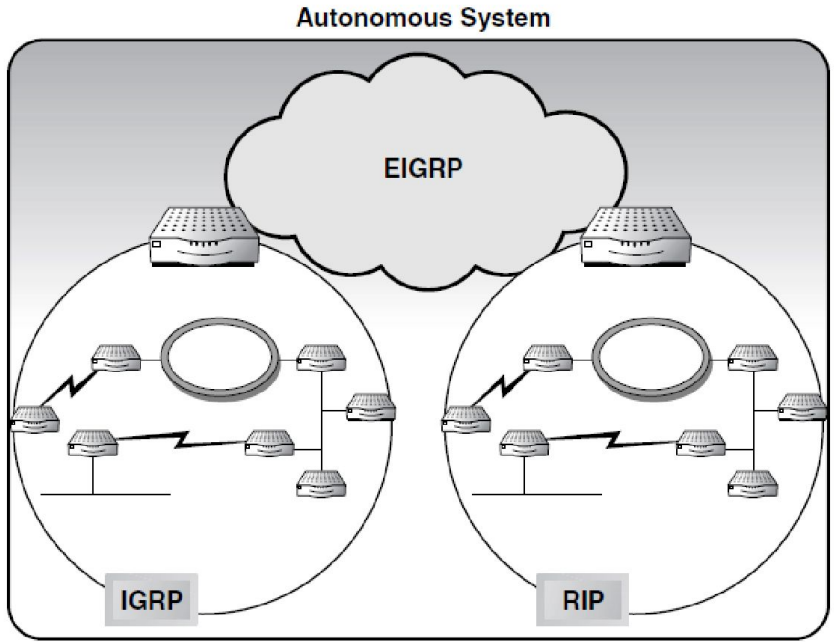
- Most routing occurs within logical boundaries referred to as Autonomous Systems (AS), or routing domains.
- Up until recently these terms were used interchangeably within the industry to describe a collection of related networks, subnets, and routers that use the same routing protocol and share information within the common area controlled by a single administrative entity.
- However, that is not necessarily the case these days, and most companies do not operate in this manner

Autonomous Systems and Routing Domain

- Take an organization's network that spans a large geographic area.
- It might deploy several different routing protocols (for example, RIP and OSPF) within each geographic location.
- Each location might have a separate IT (Information Technology) department (administrative body) controlling it.
- In this example, RIP and OSPF would be considered separate routing domains.
- Each routing domain consists of the routing protocol (RIP or OSPF) and the networks, subnets, and routers within this domain.
- The organization's network as a whole, regardless of the number of routing protocols operating within it, is considered a single Autonomous System

Autonomous Systems and Routing Domain



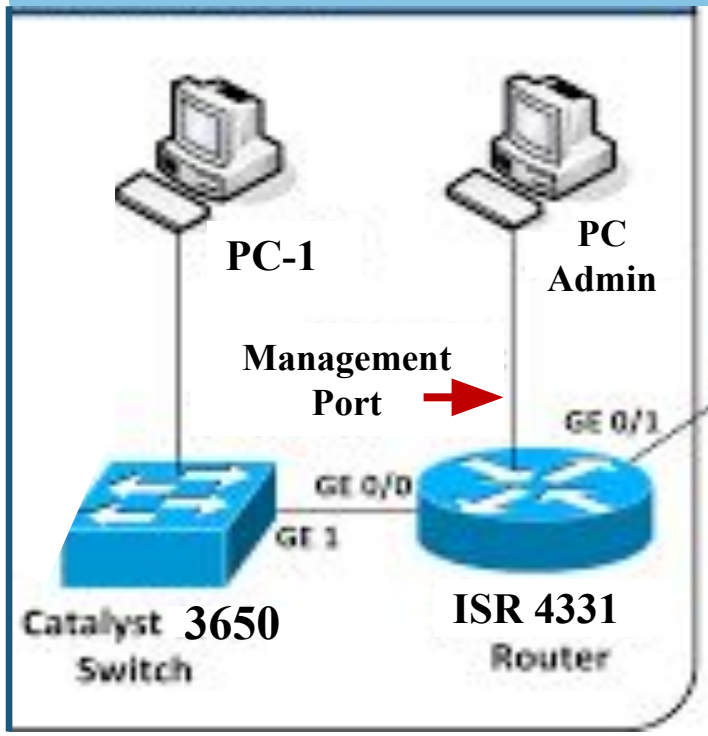


- Presently it is more common to use the term routing domain when referring to routers and networks sharing a common routing protocol.
- The term AS is now used to describe a group of routing domains.
- For example, an organization running three routing protocols, such as **RIP**, **IGRP** and **EGRP** would be considered to have **three** separate **routing domains** within a **single Autonomous System**.

Routing Mechanisms

- Routers learn about paths (routes) to destinations through several routing mechanisms.
- Typically, routers use a combination of the following **routing methods** to **build** router's **route table**:
  - Directly connected interface
  - Static
  - Default
  - Dynamic
- Although there are specific advantages and disadvantages for implementing them, they are not mutually exclusive.

Directly Connected Interfaces



ISR: Integrated Services Router
Catalyst 3650: 48 Port L3 Switch
Catalyst 2960: 24 Port L3 Switch

GE0/0 and **GE0/1**: Gigabit Ethernet Interfaces on the Router.

Note: Referred to as **GigE Interface**

GE: **GigE** port on the **Switch** side.

Note that a GigE Interface can also be connected to other lower speed (**FE** or 10 Mbps) ports. The ports share their capabilities with each other and negotiate to finally choose the lower speed, for exchanging frames between them.

- Directly connected interfaces are routes that are local to the router.
- That is, a router normally has interfaces that are directly connected to one or more networks or subnets.

How many **Interfaces** are in the **Router**? **ANS: 2**

- These networks are inherently known to the routers by configuring the interfaces attached to the networks,

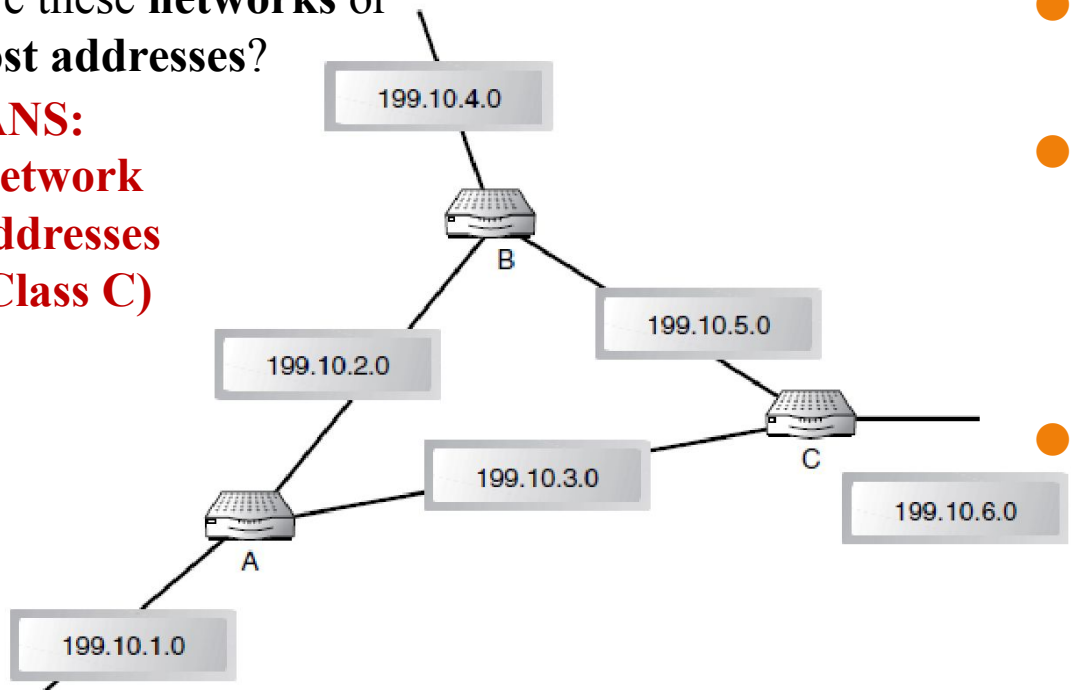
Management port: **For configuring the device and not for traffic.**

FE: Fast Ethernet, speed 100 Mbps.

Directly Connected Interfaces

Are these **networks** or **host addresses**?

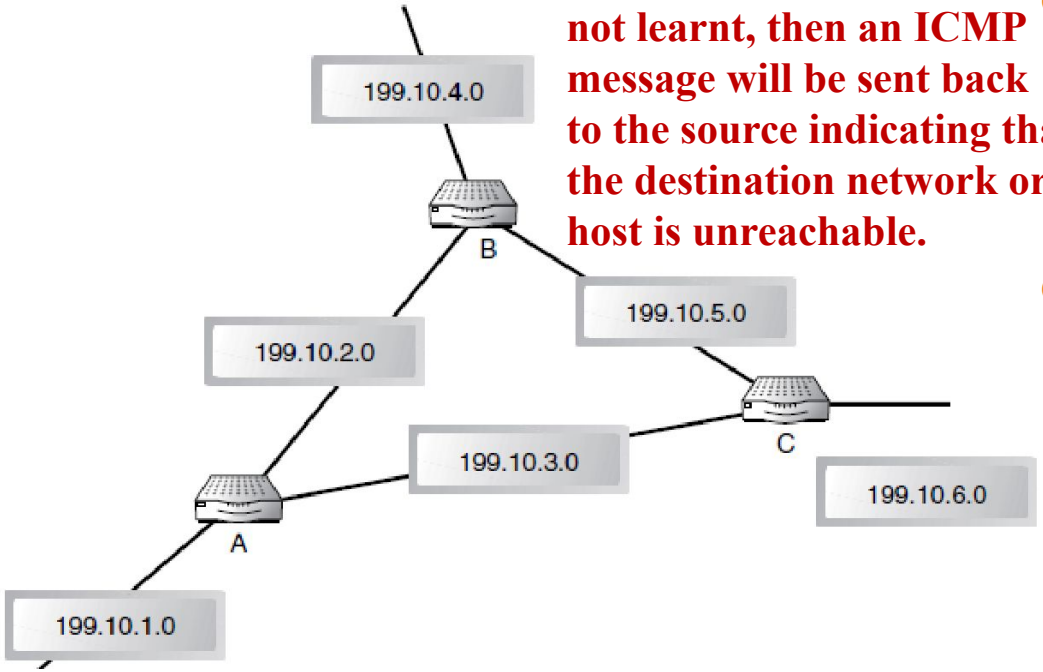
ANS: Network addresses (Class C)



- This network has three Routers (A, B, and C).
- These networks are immediately recognizable by the routers which are connected directly to them.
- Traffic directed to these networks can be forwarded by them without any help from routing protocols.

- How many directly connected networks does each router has? **ANS: 3**
- For example, Router A is directly connected to networks 192.10.1.0, 192.10.2.0, and 192.10.3.0 through local interfaces.
- All the router's interfaces are configured with the network addresses they are connected to.

Directly Connected Interfaces

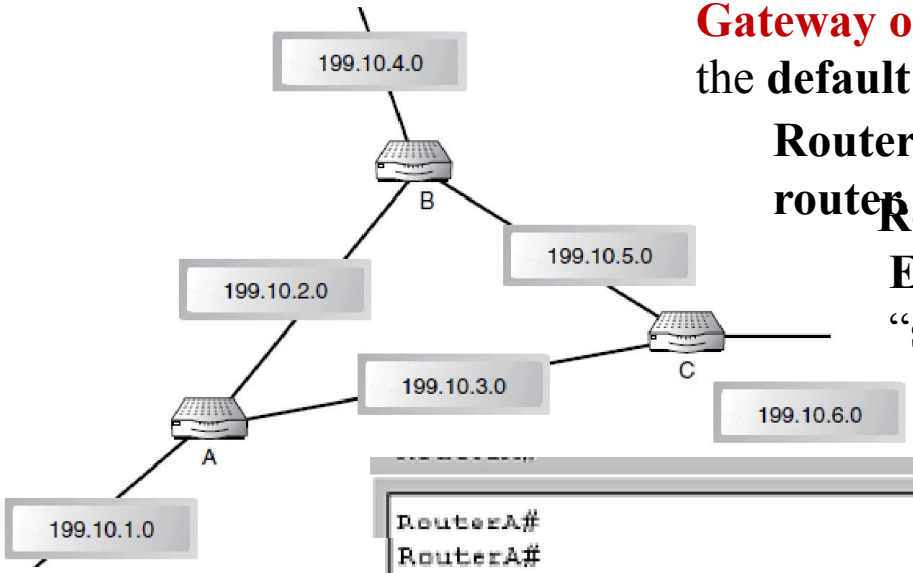


Note: **If the routers have not learnt, then an ICMP message will be sent back to the source indicating that the destination network or host is unreachable.**

- Datagrams received by Router A destined for any of these attached networks will be forwarded without assistance.
- Does the **Router A** know how to reach **199.10.5.0 without** a specific **entry** in its **routing table**? **ANS: NO**

- What are the networks that **Router A** needs entries in its Routing table to forward datagrams destined to them?
  - **ANS**: For the networks that are not directly connected to Router A, 199.10.4.0, 199.10.5.0, and 199.10.6.0.
- Router A needs to learn about these networks to forward traffic to them, since they are not directly connected to it.

Router A: Routing Table Entries



Gateway of last resort not set, indicates that the **default route** is **not set** in the **Routing Table (RT)**.

RouterA prompt shows the **name** of the **router**.

RouterA# , the hash symbol indicates **privileged EXEC mode** in which commands like "**show ip route**" can be executed.

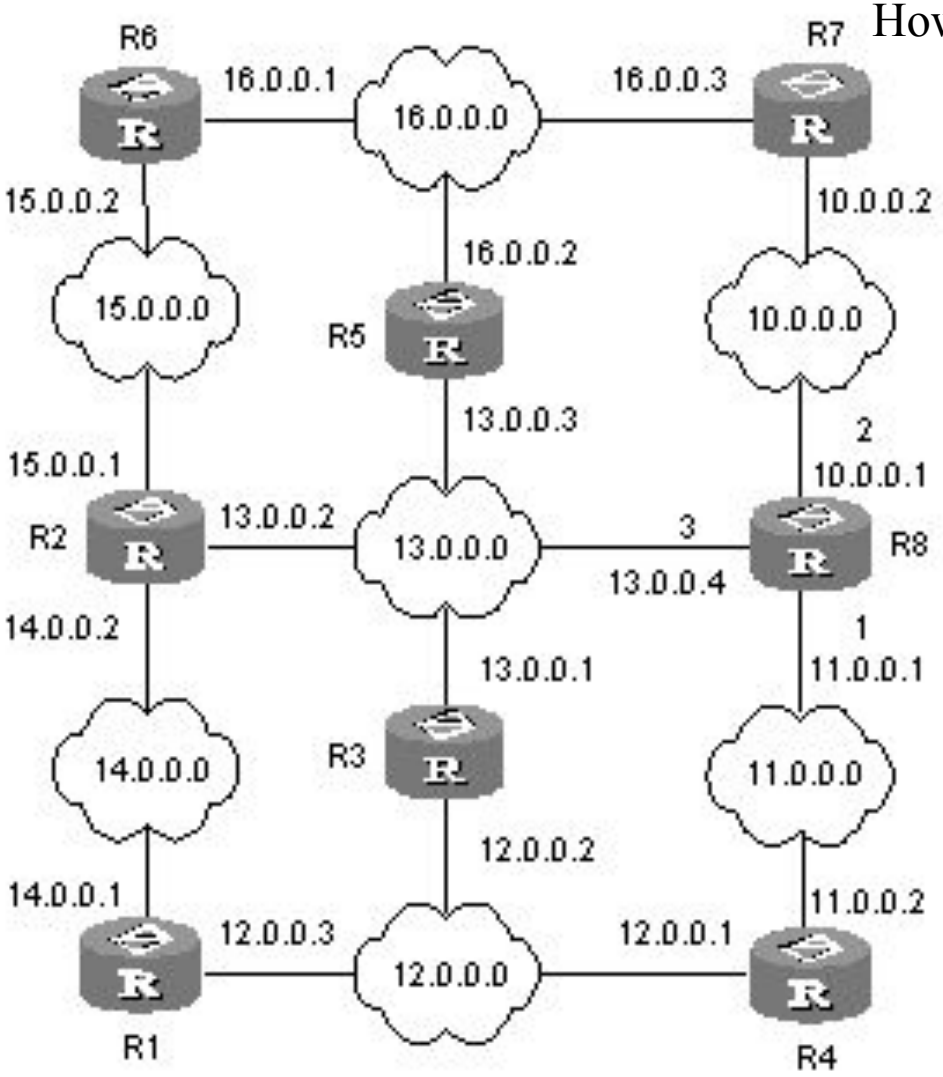
Message on a CISCO router:

```
RouterA#
RouterA#
RouterA#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

C    199.10.3.0/24 is directly connected, Serial1
C    199.10.2.0/24 is directly connected, Serial0
C    199.10.1.0/24 is directly connected, Ethernet0
RouterA#
RouterA#
RouterA#
```

Network names are given for each of the networks as well.

Static Routing

Static Routing: An Example



How many entries on RT on each Router? **ANS: 7**

Routing table of router R8

Entry Type	Destination network	Next hop	Interface
C	10.0.0.0	10.0.0.1	2
C	11.0.0.0	11.0.0.1	1
S	12.0.0.0	11.0.0.2	1
C	13.0.0.0	13.0.0.4	3
S	14.0.0.0	13.0.0.2	3
S	15.0.0.0	13.0.0.2	3
S	16.0.0.0	10.0.0.2	2

C: Directly Connected (router adds based on the interface configurations)

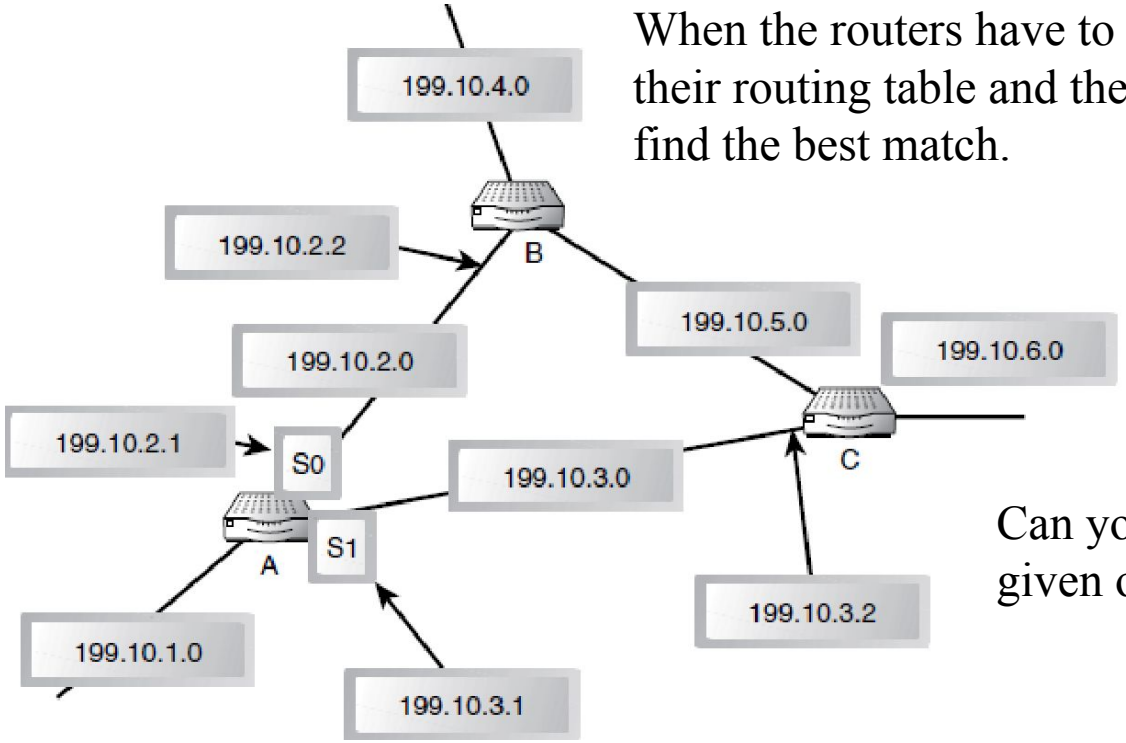
S: Static Entry, manually made

Homework: Practice making RT entries for other routers in the network.

Static Routing

- Static routes are routes to destination hosts or networks that an administrator has manually entered into the router's route table.
- Static routes define the IP address of the next hop router and local interface to use when forwarding traffic to a particular destination.
- Because this type of route has a static nature, it does not have the capability of adjusting to changes in the network.
- If the router or interface defined fails or becomes unavailable, the route to the destination fails.
- This type of routing method has the advantage of eliminating all traffic related to routing updates.
- Static routing tends to be ideal where the **link is temporary** or **bandwidth** is **an issue**, so this method is used for dial-up networks or point-to-point WAN links.
- Static routes in conjunction with other routing methods are implemented to use backup links when the primary links implementing dynamic routing protocols have failed.

Configuring Static Routes



When the routers have to forward packets, they will check their routing table and they use longest prefix matching to find the best match.

If it doesn't have a best match then the router will use a default route (if you have one in RT). Otherwise, the packet will be dropped and an ICMP msg is generated.

Can you **identify** these **commands** are given on which **router**? **ANS: RA**

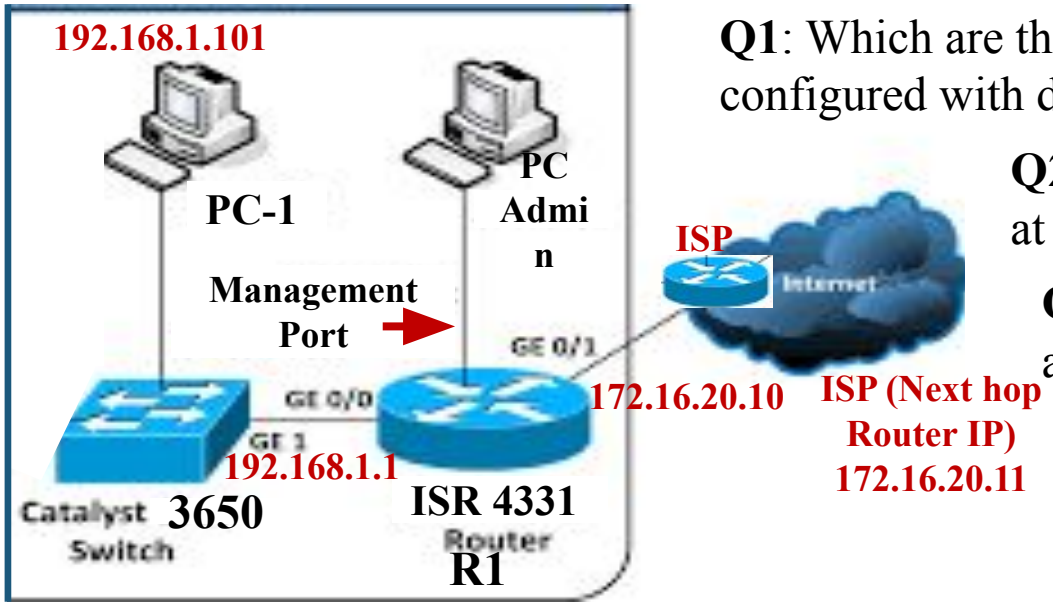
Commands to **set up Static routes** are **shown below.**

```
ip route 199.10.4.0 255.255.255.0 199.10.2.2
ip route 199.10.5.0 255.255.255.0 199.10.2.2
ip route 199.10.6.0 255.255.255.0 199.10.3.2
!
```

Issues with Static Routing

- Designing an entire network with only static routing method requires that entries are made on every router for each network they are not directly attached to, which is highly impractical.
- In addition, if a link or a router within the internetwork fails or is added, you would have to reconfigure each router, removing the failed route or adding a new route.
- Meanwhile, until the routing tables are updated manually, the routers obviously cannot forward traffic to those destinations because the original paths have become invalid.
- Static routing can have an extreme amount of overhead in the form of intense administrative hours spent getting the network up and keeping it going.
- Dynamic routing algorithms are versatile and adapts to network changes
  - Whereas, static routes conserve bandwidth because they do not cause routers to generate route update traffic.

Quiz 1: Default Route Entries



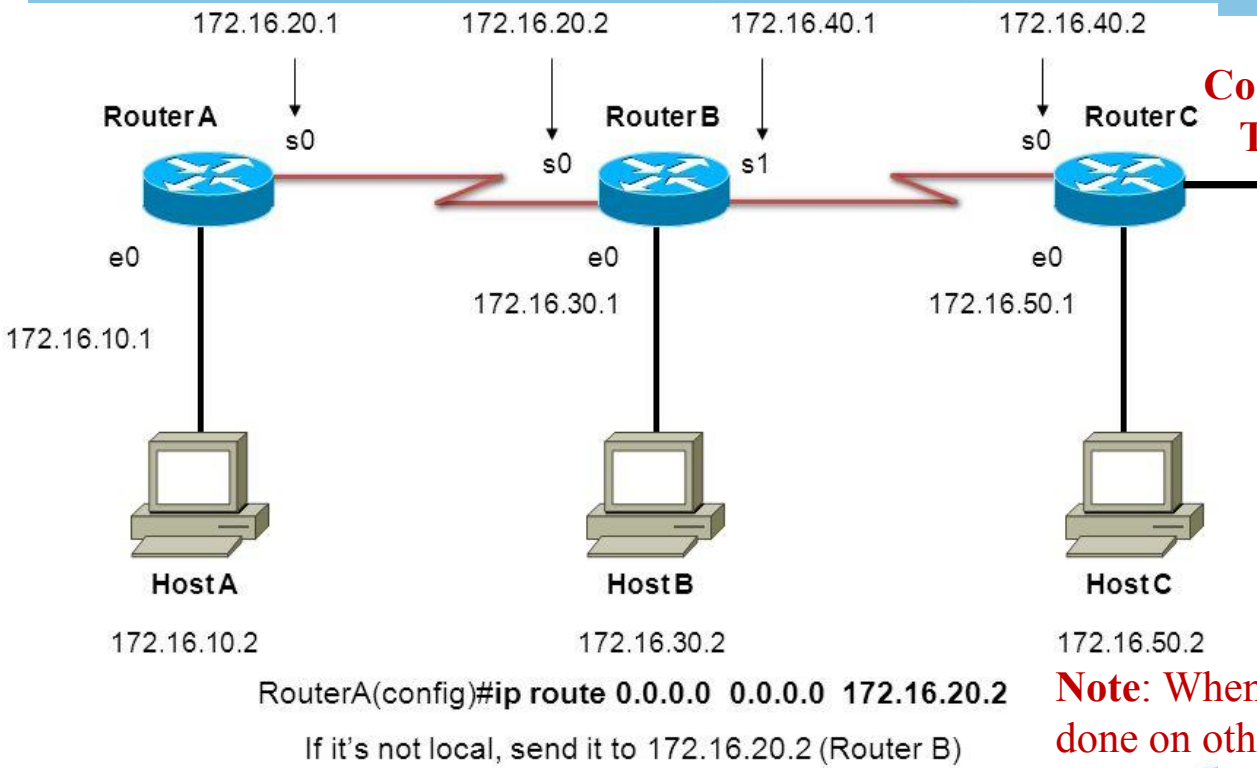
Q1: Which are the devices need to be configured with default route? **ANS1: Both the hosts and the Router**

Q2: What is the default route to be set at the host (**PC-1**)? **ANS2: 192.168.1.1**

Q3: What is the default route to be set at the router (**R1**)? **ANS3: 172.16.20.11**

- Every IP host needs to have a default route either manually configured or dynamically learned through DHCP.
  - Default routes provide end hosts a way out of their local subnet.
- When Routers are configured with a default route it is the route of last resort.
  - If no other route (specifically relating to the destination) exists in the routers route table, default route is taken.

Default Route: Another Example



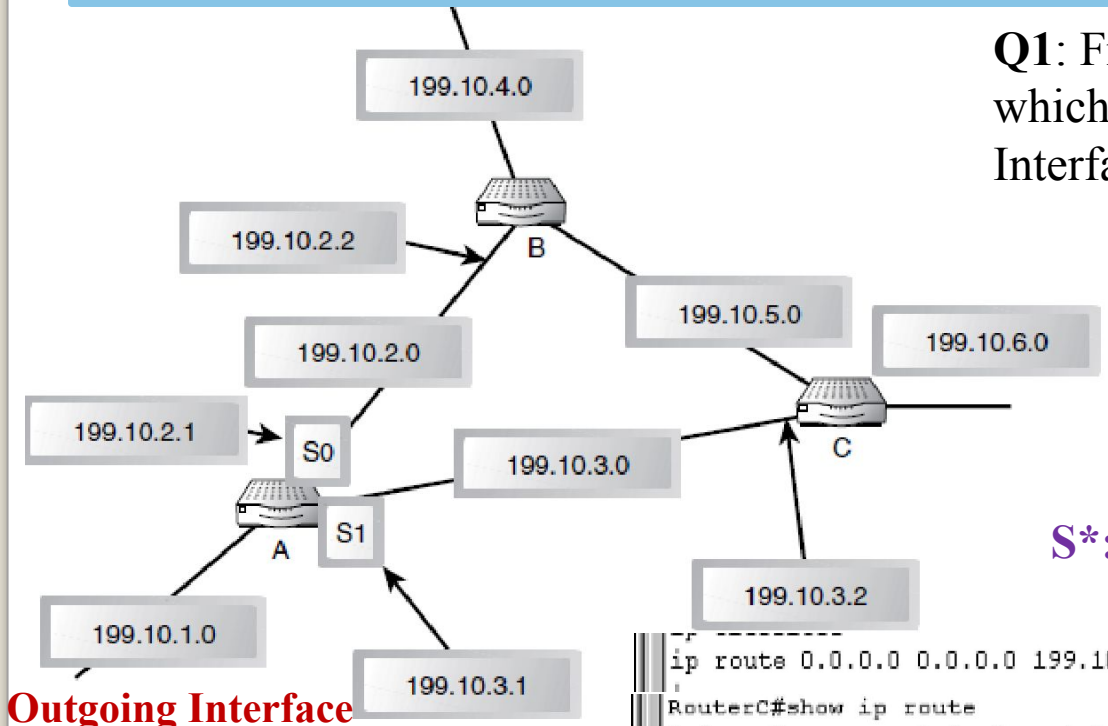
Subnet mask being all zeros (**0.0.0.0**) means that there is **no network ID bits** of the **given IP** address is to be **compared** against, to **send** a packet **out**, though the **default route**.

Note: When all the **longest prefix match** done on other entries in Routing table fails, **Default route** is taken (which is **catch-all**)

Command Syntax: **ip route** "destination IP address" "Subnet mask of destination network" "next hop router's IP address"

The **default route** in Internet Protocol Version 4 (**IPv4**) is **designated** as the zero-address **0.0.0.0/0** in **CIDR** notation, often called the **quad-zero route.**
The **subnet mask** is given as **/0**, which effectively **specifies all networks and it is the shortest match** available after **all other matches** have **failed**.

Default Route: Configuration on CISCO Router



Outgoing Interface to Internet

Q1: From the configuration shown below which link is considered to be outgoing Interface connected to Internet here? **ANS1:**

Since RC is considering 199.10.3.1 as the default route's next hop router IP the outgoing interface from the RA is the one considered to be connected to Internet through the Network 199.10.1.0

S*: The * signifies that this route is a candidate for default routing

```
ip route 0.0.0.0 0.0.0.0 199.10.3.1

RouterC#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is 199.10.3.1 to network 0.0.0.0

C    199.10.5.0/24 is directly connected, Serial0
C    199.10.6.0/24 is directly connected, Ethernet0
C    199.10.3.0/24 is directly connected, Serial1
S*   0.0.0.0/0 [1/0] via 199.10.3.1
RouterC#
RouterC#
RouterC#
```

Note: Static routing method is being used here to configure the default route on all the Routers.

Homework: Configure **default routes** on RA and RB

Routing Metrics and Costs

Metrics of Links

- **Metrics** are **cost values** used by routers to determine the **best path** to a **destination network.**
- Several factors help dynamic routing protocols decide which is the preferred or shortest path to a particular destination.
- These factors are known as **metrics** and **algorithms.**
- **Metrics** are the **network variables** used in deciding what path is preferred in terms of these metrics.
- For some routing protocols these metrics are static and may not be changed.
- For other routing protocols these values may be assigned by a network administrator.
- The most common metric values are hop, bandwidth, delay, reliability, load, and cost.

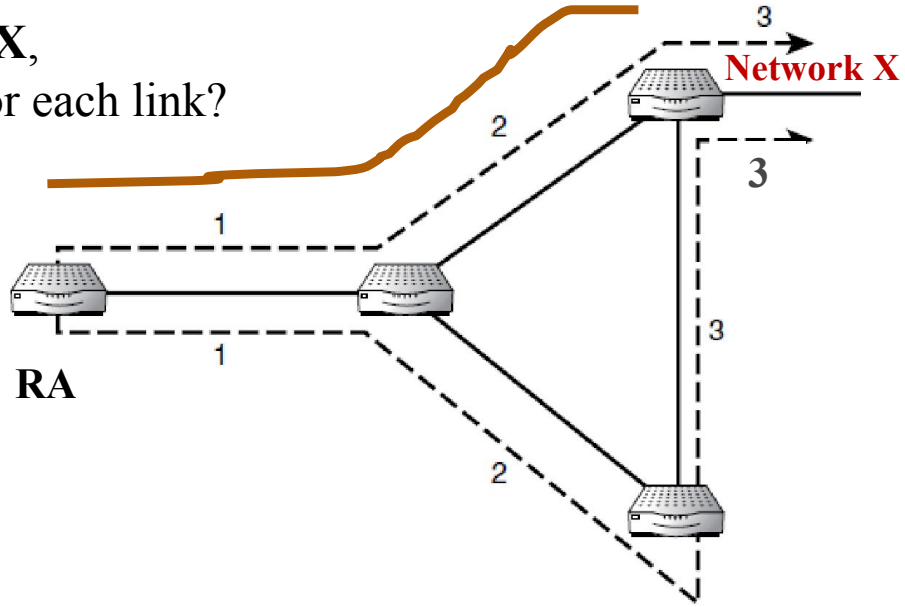
Metric: Hop

- A **hop** is a metric value used to measure **distance** based on the **number of networks** a **datagram** traverses.
- Each time a router forwards a datagram onto a segment this counts as a single hop.
- Routing protocols that observe hops as their primary metric value consider the best or preferred path (when multiple paths exist) to a destination to be the one with the least number of network hops.

Q1: Which is the **preferred path to Network X**, from RA, if the costs are given over the path for each link?

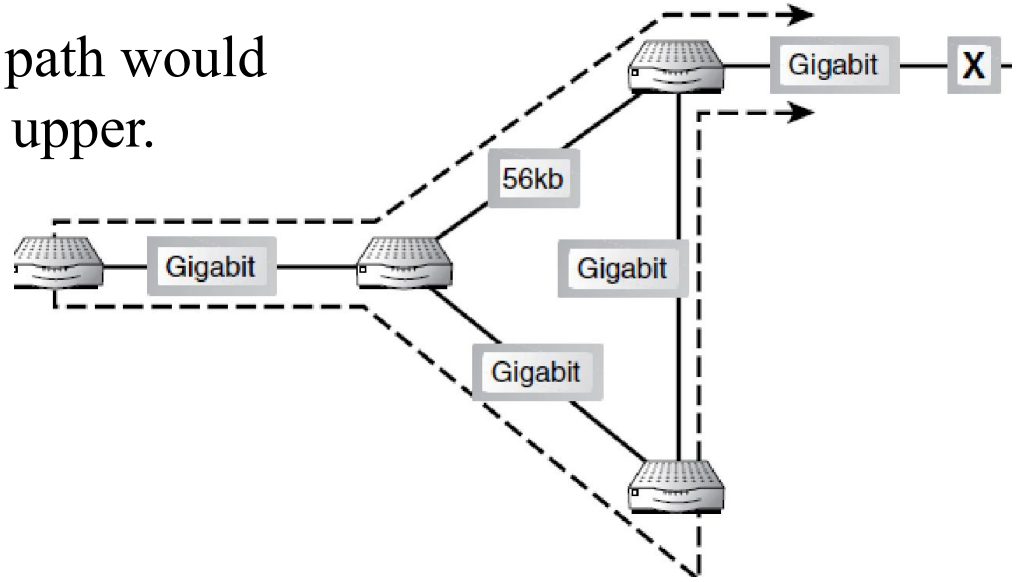
ANS1: **Preferred path is the one with least amount of total costs. The Upper path to Network is preferred.**

Total cost of the **chosen path** is: **1 + 2 = 3**



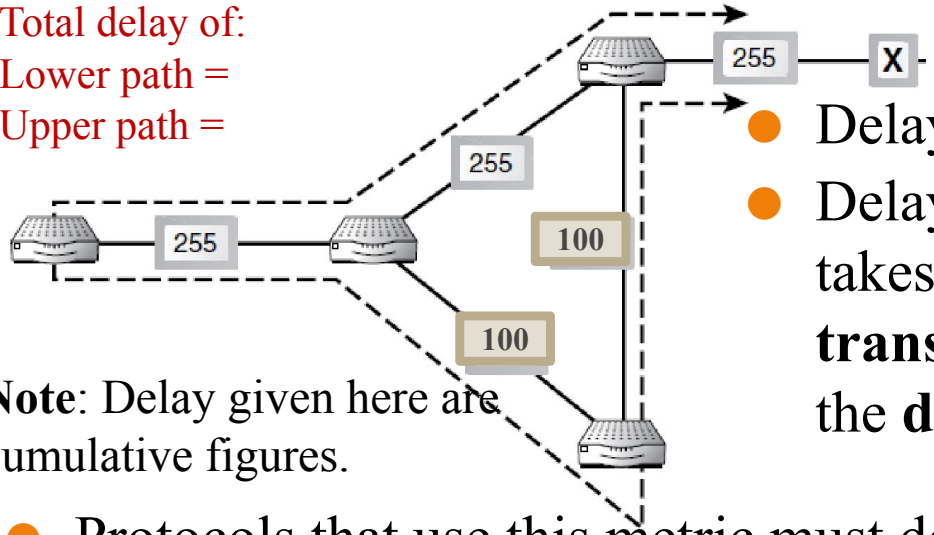
Metric: Bandwidth (Data transfer rate of the link)

- Routing protocols that only reference hops as their metric do not always select the best path through a network.
- Just because a path to a destination contains fewer network hops than another does not make it the best.
- The upper path may contain a slower link, such as 56kb dial-up link along the second hop, whereas the lower path may consist of more hops but faster links, such as gigabit Ethernet.
- If this were the case, the lower path would undoubtedly be faster than the upper.
- However, routing protocols that use hops do not consider other metric values in their routing decisions.



Metric: Delay

Total delay of:
Lower path =
Upper path =



Note: Delay given here are cumulative figures.

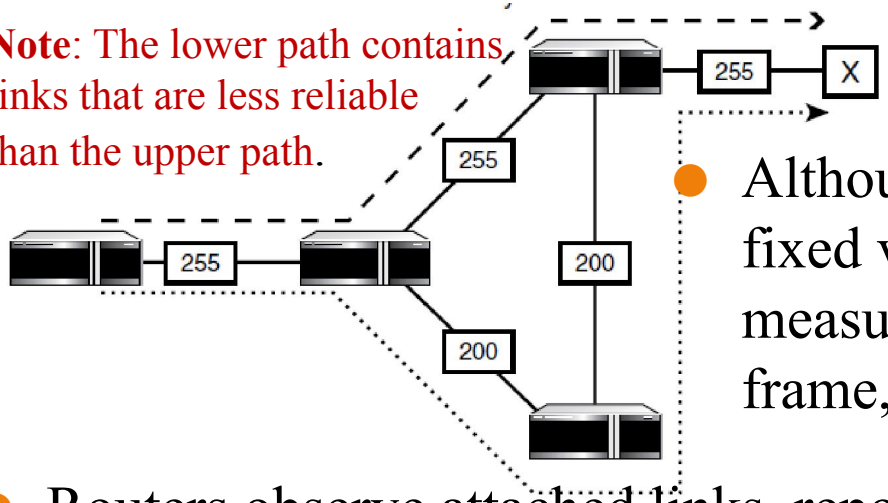
- Delay is measured in tens of μ seconds.
- Delay represents the amount of time it takes for a **router to process**, **queue**, and **transmit** a datagram **out** an **interface plus** the **delay introduced** by the **link**.

- Protocols that use this metric must determine the delay values for all links along the path end to end, considering the path with the lowest (cumulative) delay to be a better route.
- Although the lower path in the above figure is obviously longer in terms of hops, it is faster in terms of delay.
- The lower path has an overall delay time of 455 microseconds end to end, while the upper path has a delay of 510 microseconds.
- The lower the delay time the better the path.

Note: Delays introduced by the routers are not considered here.

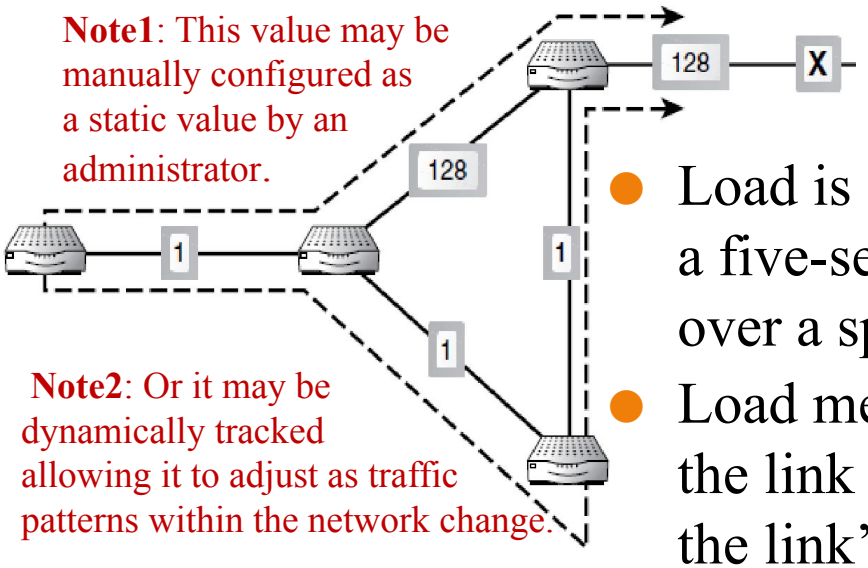
Metric: Reliability

Note: The lower path contains links that are less reliable than the upper path.



- Although this metric may be configured as a fixed value by an administrator, it is generally measured dynamically over a specific time frame, such as five seconds
- Routers observe attached links, reporting problems, such as link failures, interface errors, lost datagrams and so on.
- Links experiencing more problems would be considered less reliable than others making them less desirable paths—the higher the reliability the better the path.
- Because network conditions are constantly changing, link reliability will change.
- This value is generally measured as a percentage of 255, with 255 being the most reliable and 1 being least reliable.

Metric: Load



Note1: This value may be manually configured as a static value by an administrator.

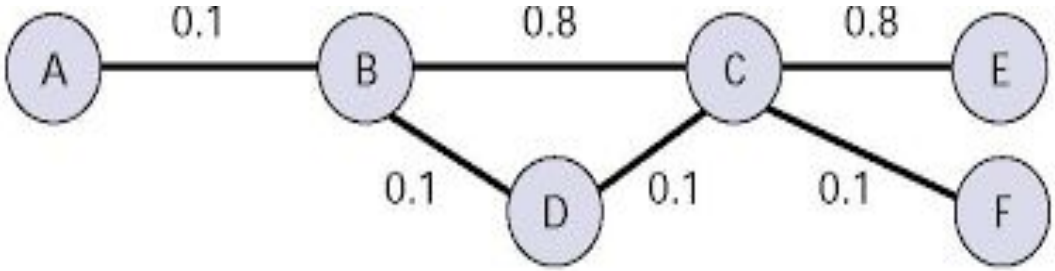
Note2: Or it may be dynamically tracked allowing it to adjust as traffic patterns within the network change.

Note3: It is important to remember that as traffic increases load across a link will increase. This Load metric value changes as traffic patterns change.

- Load is a variable value, generally measured over a five-second window indicating the traffic load over a specific link.
- Load measures the amount of traffic occupying the link over this time frame as a percentage of the link's total capacity

- The value 255 is equivalent to 100% utilization or load—the higher the value the higher the traffic load (bandwidth utilization) across this link.
- As traffic increases, this value increases.
- Values approaching 255 indicate congestion, while lower values indicate moderate traffic loads—the lower the value, the less congested the path, the more preferred.

Quiz 2: Choose the Optimal Path



The numbers given over the links are the delay experienced by the Data packets over those links.

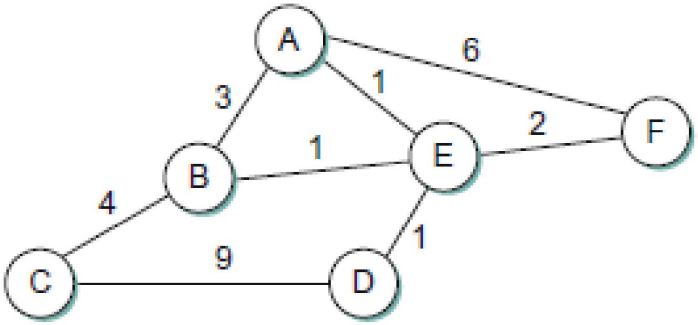
- Assume the Routing algorithm operates on the delay metric and hop count find the Optimal path for the traffic going between the nodes:

	Metric: Delay	Metric: Hop Count
◦ **A to F:**	**A□B□D□C□F**	**A□B□C□F**
◦ **B to E:**	**B□D□C□E**	**B□C□E**
◦ **F to D:**	**F□C□D**	**F□C□D**

Note: Network administrators can affect the way routers make path decisions by setting **arbitrary metric values** on links along the path end to end. These arbitrary values are typically single integers with lower values indicating better paths, which are called as **costs of the links**.

Network as a graph

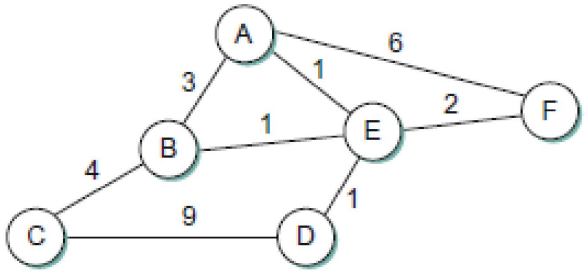
- Routing is, in essence, a problem of graph theory.
- The figure shows a graph representing a network
- The nodes of the graph, labeled A through F, may be:
  - **Hosts, switches, routers, or networks** **What** do the **nodes** represent?



- What do the nodes represent?
- Switches could also be represented by the nodes, because switches could also be IP aware or L3 switches
- As you are aware, telnet into L3 switches are also possible, which means that a L3 switch could also be similar to a host with its own (VLAN) IP.
- Routers are of course considered as nodes, interconnected with links

Cost of the links

- For **our** initial **discussion**, we will **focus** on the case where the **nodes** are **only routers**.
- The edges of the graph correspond to the network links between the routers.
- Each edge has an associated *cost*, which gives some indication of the desirability of sending traffic over that link.



Note: Undirected edges with each edge assigned a single cost are assumed here. Though edges could be directed with pair of edges between two nodes, each with its own edge costs.

- Cost of a link can be derived based on various metrics that we had discussed in the last session (bandwidth, delay, reliability, load, etc.)
- The basic problem of routing is to find the lowest-cost path between any two nodes.
- Where the cost of a path equals the sum of the costs of all the edges that make up the path.

Issues with the Static Routing (To find the Shortest Paths)

- We could imagine just calculating all the shortest paths and loading them into some nonvolatile storage on each node.
- Such a static approach has several **shortcomings**:

1. It does not deal with node or link failures.
2. It does not consider the addition of new nodes or links.
3. It implies that edge costs cannot change, even though we might reasonably wish to have link costs change over time.
   - **Example**: assigning high cost to a link that is heavily loaded.

- For these reasons, **routing** is achieved in most practical networks **by** running **routing protocols** among the nodes.
- These protocols provide a **distributed**, dynamic way to solve the problem of finding the lowest-cost path in the presence of link and node failures and changing edge costs.

Distributed Routing Protocols: Explained

- The distributed nature of routing algorithms is one of the main reasons why this has been such a rich field of research and development.
- There are a lot of **challenges** in making **distributed algorithms** work well.
- For example, distributed algorithms raise the possibility that **two routers** will **at one instant** have **different ideas** about the **shortest path** to **some destination**.
- In fact, each one may think that the other one is closer to the destination and decide to send packets to the other one.
- Clearly, such **packets** will be **stuck in a loop** until the discrepancy between the two routers is resolved, and it would be good to resolve it as soon as possible.
- This is just one example of the type of problem routing protocols must address.

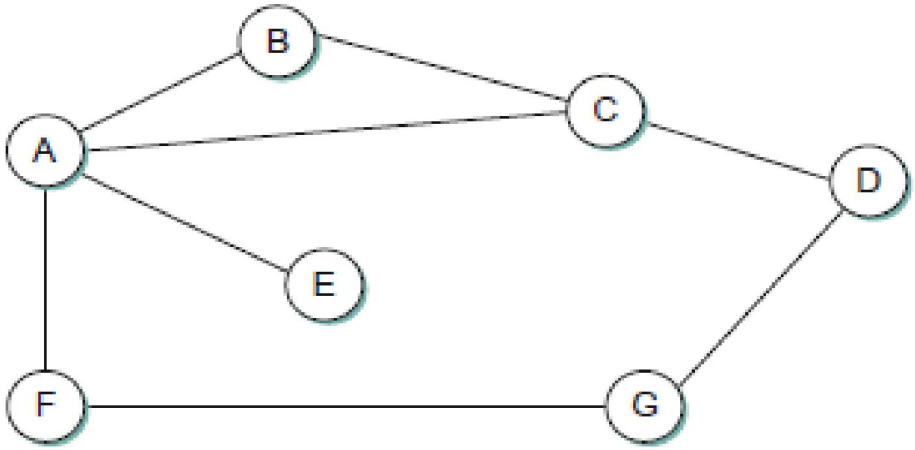
Distance Vector Routing

Distance Vector Routing

- The idea behind the distance-vector algorithm is suggested by its name.
- Each node constructs a one-dimensional array (a vector) containing the "distances" (costs) to all other nodes and distributes that vector to its immediate neighbors.
- The starting assumption for distance-vector routing is that each node knows the cost of the links to each of its directly connected neighbors.
- These costs may be provided when the router is configured by a network manager, along with the IP address of each interface of the routers connected to the network.
- A link that is down is assigned an infinite cost.

1. Distance Vector Algorithm

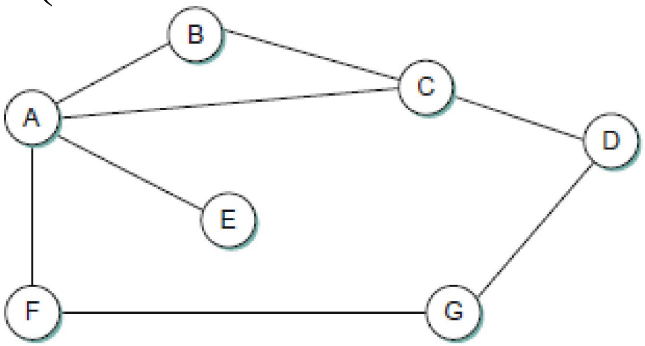
- To see how a distance-vector routing algorithm works, it is easier to consider an example like the one depicted in the figure.



- In this example, the cost of each link is set to 1, so that a **least-cost path** is simply the **one with the fewest hops**.
- Since **all edges** have the **same cost**, costs are **not shown** in the picture above.

2. Distance Vector Algorithm

- We can represent each node's knowledge about the distances to all other nodes as a table.
- Each node knows only the information in one row or column of the table (the one that bears its name).



Initial Distances Stored at Each Node (Global view)

Information Stored at Node	Distance to Reach Node						
	A	B	C	D	E	F	G
A	0	1	1	∞	1	1	∞
B	1	0	1	∞	∞	∞	∞
C	1	1	0	1	∞	∞	∞
D	∞	∞	1	0	∞	∞	1
E	1	∞	∞	∞	0	∞	∞
F	1	∞	∞	∞	∞	0	1
G	∞	∞	∞	1	∞	1	0

Note:. The **global view** that is presented here is **not available** at any **single point** in the **Network**, they are available on each router in the network.

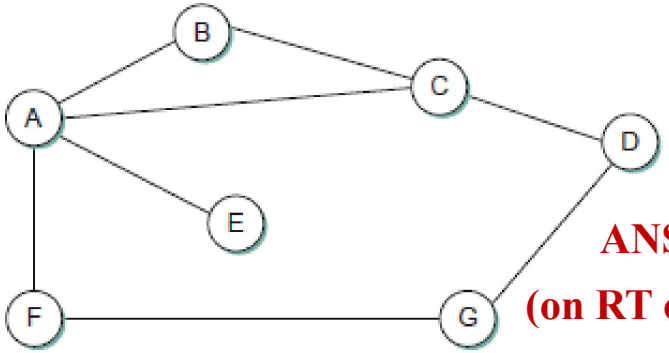
Q1. Each row or **column** is **stored** in each **router?** **ANS1:**

Entries in the row and column of each router are the same!!!

3. Distance Vector Algorithm

- Each **row** in **the routing table** is the current beliefs of that node on the distance to all other nodes.
- Thus, A initially believes that it can reach B in one hop and the Node A does not even know the existence of nodes D & G.
- The routing table stored at A reflects this set of beliefs and includes the name of the next hop that A would use to reach any reachable node.

Note: Though entries are shown, cost with ∞ are not present in RT.



ANS1:

(on RT of nodeA)

Initial Routing Table at Node A

Destination	Cost	NextHop
B	1	B
C	1	C
D	∞	—
E	1	E
F	1	F
G	∞	—

Q1: How many entries on power on?

4

1.There is **no entry** for itself (**node A**) because this is the **routing table** stored in **Node A**. Remember we are considering only routers as nodes here.

2. **Next hop** entries in **RT** are **actually** the **IP addresses** of the **Next hop Router Interfaces,** directly connected to the Router.

4. Distance Vector Algorithm (C to A)

Initial RT at node A

Destination	Cost	NextHop
B	1	B
C	1	C
D	∞	—
E	1	E
F	1	F
G	∞	—

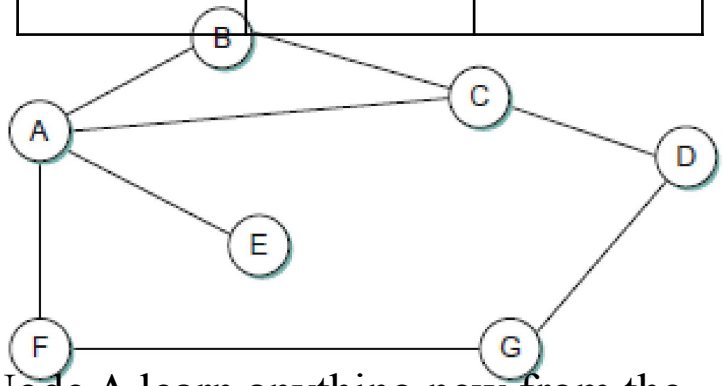
From Node C to Node A

Initial RT at node C

Destination	Cost	Next Hop
A	1	A
B	1	B
D	1	D
E	∞	--
F	∞	--
G	∞	--

RT of Node A after Receiving update from Node C

Destination	Cost	Next Hop
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	∞	--



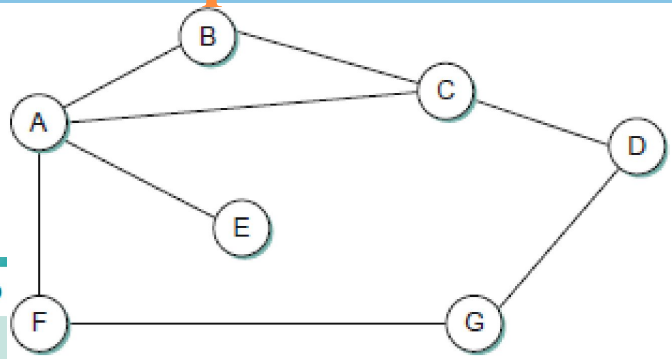
1. Does the Node A learn anything new from the update it receives from Node C? **ANS1: Yes.**

2. What changes does Node A make to its RT based on the update from Node C? **ANS2: Node A adds a new entry for Node D, which it didn't know about.**

5. Explanation on Node A learning from the Initial update from Node C

Initial RT at node A

Destination	Cost	NextHop
B	1	B
C	1	C
D	∞	—
E	1	E
F	1	F
G	∞	—



Changes to RT at Node A from is initial state based on the first update from Node C

Destination	Cost	Next Hop
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	∞	--

- For example, A learns from C that D can be reached from C at a cost of 1; it adds this to the cost of reaching C (1) and decides that D can be reached via C at a cost of 2, which was not in the RT earlier on Node A's RT.
- At the same time, A learns from C that B can be reached from C at a cost of 1, so it concludes that the cost of reaching B via C is 2.
- Since this is worse than the current cost of reaching B (1), this new information is ignored, no update is done related to its entry about Node B.

6. Distance Vector Algorithm (B to A)

RT at Node A after the update from C was received

Destination	Cost	Next Hop
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	∞	--

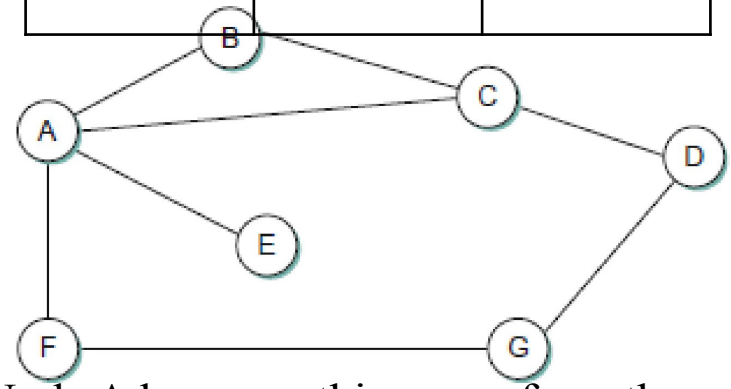
From Node B to Node A

Initial RT at node B

Destination	Cost	Next Hop
A	1	A
C	1	C
D	∞	--
E	∞	--
F	∞	--
G	∞	--

RT of node A after Receiving Update from node B

Destination	Cost	Next Hop
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	∞	--



1. Does the Node A learn anything new from the update it receives from Node B? **ANS1: No.**
2. What is the change Node A makes to its RT?
   ANS2: No changes.

7. Distance Vector Algorithm (E to A)

RT at Node A after updates from B & C were received

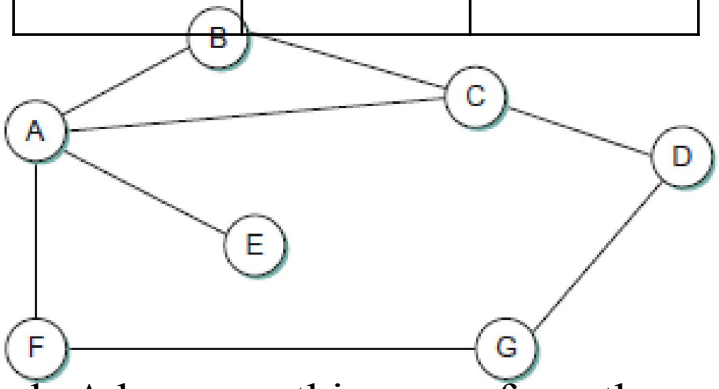
Destination	Cost	Next Hop
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	∞	--

Initial RT at node E

Destination	Cost	Next Hop
A	1	A
B	∞	--
C	∞	--
D	∞	--
F	∞	--
G	∞	--

RT of Node A after Receiving update from Node E

Destination	Cost	Next Hop
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	∞	--



1. Does the node A learn anything new from the Update it receives from node E? **ANS1: No.**

2. What is the change does node A make? **ANS2: None.**

8. Distance Vector Algorithm (F to A)

RT at node A after updates from B, C & E were received

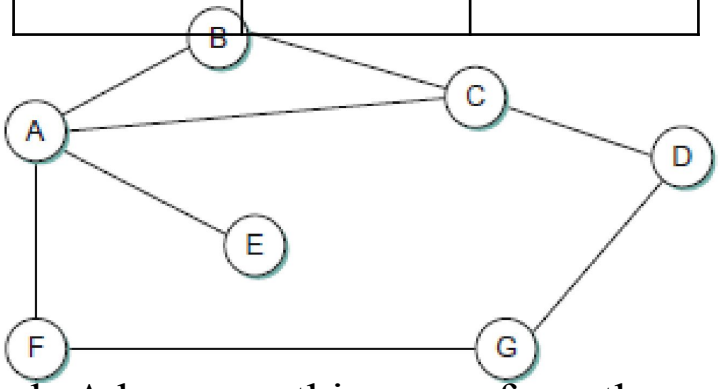
Destination	Cost	Next Hop
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	∞	--

Initial RT at node F

Destination	Cost	Next Hop
A	1	A
B	∞	--
C	∞	--
D	∞	--
F	∞	--
G	1	G

RT of node A after Receiving Update from node F

Destination	Cost	Next Hop
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	2	F



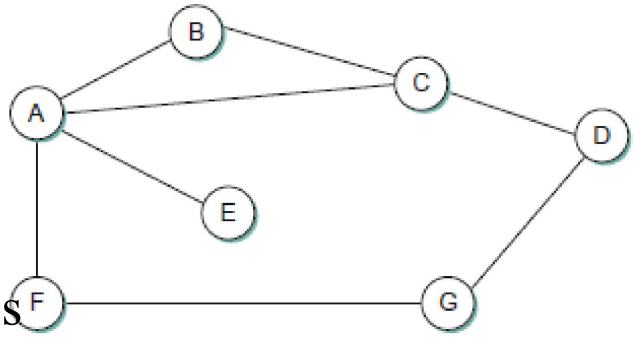
1. Does the node A learn anything new from the Update it receives from node E? **ANS1: Yes.**

2. What is the change does node A make?

ANS2: It updates the entry for G with a hop distance of 2 through node F.

9. Distance Vector Algorithm (Global view and local view at node A)

- In the absence of any topology changes, it takes only a few exchanges of information between neighbors before each node has a complete routing table.
- The process of getting consistent routing information to all the nodes is called ***convergence.***
- The final set of costs from each node to all other nodes when routing has converged is shown below.



RT at node A receiving all the updates
All the nodes in the network

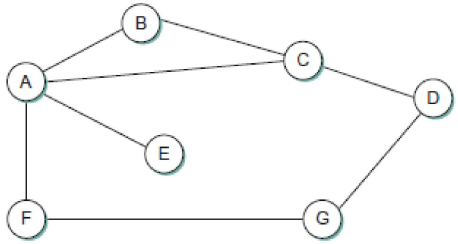
Destination	Cost	Next Hop
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	2	F

Individual view at node A is the same as the global view, after convergence

Information stored at Node	Distance to Reach Node A	B	C	D	E	F	G
A	0	1	1	2	1	1	2
B	1	0	1	2	2	2	3
C	1	1	0	1	2	2	2
D	2	2	1	0	3	2	1
E	1	2	2	3	0	2	3
F	1	2	2	2	2	0	1
G	2	3	2	1	3	1	0

10. Distance Vector Algorithm: Summary

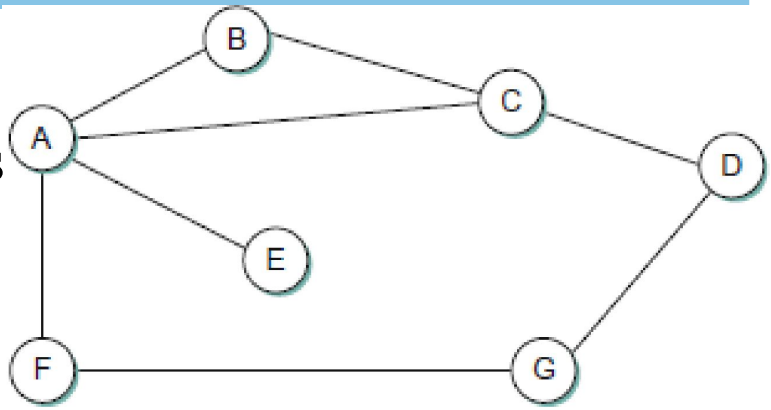
- Remember that there is no one node in the network that has the global view of all the nodes in the network, each node only knows about the contents of its own routing table.



- The beauty of a distributed algorithm like this is that it enables all nodes to achieve a consistent view of the network in the absence of any centralized authority.
- There are **two different circumstances** under which a given node decides to send a routing update to its neighbors.
- One of these circumstances is the ***periodic*** **update**.
- In this case, each node automatically sends an update message every so often, even if nothing has changed.
- This serves to let the other nodes know that this node is still running. It also makes sure that they keep getting information that they may need if their current routes become unviable.

11. Distance Vector Algorithm: Summary

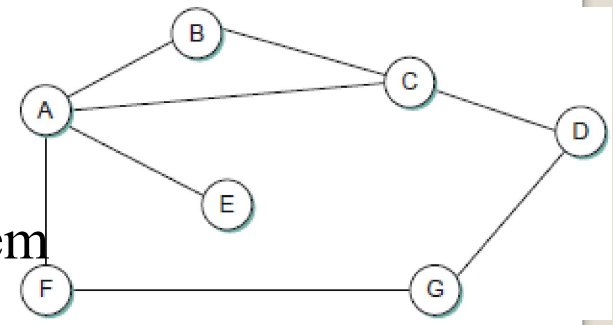
- The frequency of these periodic updates varies from protocol to protocol, but it is typically on the order of several seconds to several minutes.



- The **second mechanism**, sometimes called a ***triggered*** update, happens whenever a node notices a link failure or receives an update from one of its neighbors that causes it to change one of the routes in its routing table.
- Whenever a node's routing table changes, it sends an update to its neighbors, which may lead to a change in their tables, causing them to send an update to their neighbors.

12. Distance Vector Algorithm: (when a node or link fails)

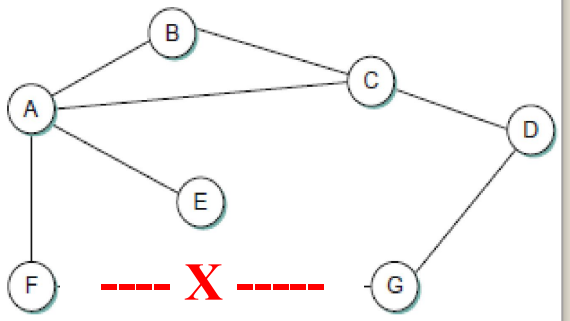
- Now consider what happens when a link or node fails.
- The nodes that notice first, send new lists of distances to their neighbors, and normally the system settles down fairly quickly to a new state



- As to the question of how a node detects a failure, there are a couple of different answers.
- In one approach, a node continually tests the link to another node by sending a control packet and seeing if it receives an acknowledgment.
- In another approach, a node determines that the link (or the node at the other end of the link) is down if it does not receive the expected periodic routing update for the last few update cycles.

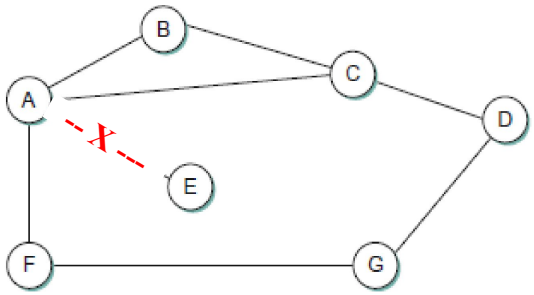
13. Distance Vector Algorithm: (Example: Link to G from F has failed)

- To understand what happens when a node detects a link failure, consider what happens when F detects that its link to G has failed.
- First, F sets its new distance to G to infinity and passes that information along to A.



- Since A knows that its 2-hop path to G is through F, A would also set its distance to G to infinity. However, with the next update from C, A would learn that C has a 2-hop path to G.
- Thus, A would know that it could reach G in 3 hops through C, which is less than infinity, and so A would update its table accordingly.
- When it advertises this to F, node F would learn that it can reach G at a cost of 4 through A, which is less than infinity, and the system would again become stable.

Distance Vector Algorithm: Count to infinity Problem (Example: Link to E from A fails)



- Unfortunately, slightly different circumstances can prevent the network from stabilizing.
- Suppose, for example, that the link from A to E goes down.
- In the next round of updates, A advertises a distance of infinity to E, but B and C advertise a distance of 2 to E, of course via the node A.
- Depending on the exact timing of events, the following might happen: Node B, upon hearing that E can be reached in 2 hops from C, concludes that it can reach E in 3 hops and advertises this to A; node A concludes that it can reach E in 4 hops and advertises this to C; node C concludes that it can reach E in 5 hops; and so on.
- This cycle stops only when the distances reach some number that is large enough to be considered infinite.
- In the meantime, none of the nodes actually knows that E is unreachable, and the routing tables for the network do not stabilize.
- This situation is known as the ***count to infinity*** **problem**.

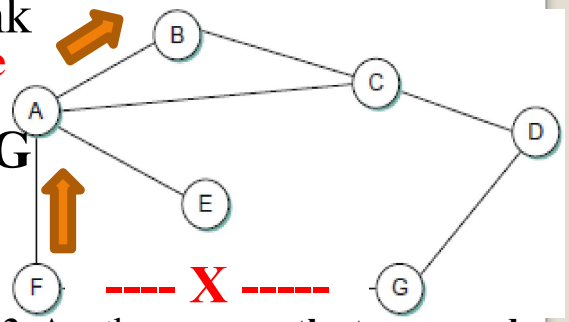
Distance Vector Routing
Link Failure

Problem 1: Link Between G and F has failed
(Reconstruct convergence using DVA)

1. Which are the routers come to know of the failure of the link shown below? **ANS1: Both F and G**

Assume, both take corrective action simultaneously.

2. What are the **stable configurations** on the **Routers F** and **G prior** to this **link Failure**?



1. Original Converged RT at node F

Destination	Cost	Next Hop
A	1	A
B	2	A
C	2	A
D	2	G
E	2	A
G	1	G

2. Original Converged RT at node G

Destination	Cost	Next Hop
A	2	F
B	3	F
C	2	D
D	1	D
E	3	F
F	1	F

Q3: Are there **any paths to any node** which are at **equal distance** from the **Routers F** and **G, prior** to **link failure**?

ANS3: Distance between G and B, is 3, both via F and D, from G.

Let us assume path chosen by both B and G is via A and F.

Q4: **F passes** the **update** to **which node**?

ANS4: To node A. Now, give the new RT at node A.

3. Assume, **F detects** the **link failure**. **New RT** at **F** will be:

3. Changed RT at node F after Link down

Destination	Cost	Next Hop
A	1	A
B	2	A
C	2	A
D	2 to ∞	G to --
E	2	A
G	1 to ∞	G to --

4. Updated RT at node A based on F

Destination	Cost	Next Hop
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	2 to ∞	F to --

5. Updated RT at node B based on A

Destination	Cost	Next Hop
A	1	A
C	1	C
D	2	C
E	2	A
F	2	A
G	3 to ∞	A to --

Problem 1: Link to G from F has failed
(Reconstruct convergence using DVA) … contd.

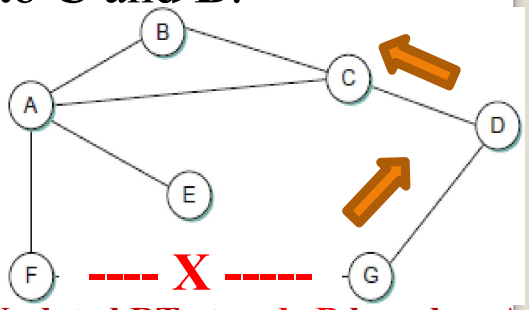
6. Let us **follow** the **updates** from the **node G** to **D** and **then** to **C** and **B**:

6. Original Converged RT at node G

Destination	Cost	Next Hop
A	2	F
B	3	F
C	2	D
D	1	D
E	3	F
F	1	F

7. Changed RT at node G after Link down

Destination	Cost	Next Hop
A	2 to ∞	F to --
B	3 to ∞	F to --
C	2	D
D	1	D
E	3 to ∞	F to --
F	1 to ∞	F to --



5. Updated RT at node B based on A

Destination	Cost	Next Hop
A	1	A
C	1	C
D	2	C
E	2	A
F	2	A
G	∞	--

8. Updated RT at node D based on G

Destination	Cost	Next Hop
A	2	C
B	2	C
C	1	C
E	3	C
F	2 to ∞	G to --
G	1	G

9. Updated RT at node C based on D (no changes)

Destination	Cost	Next Hop
A	1	A
B	1	B
D	1	D
E	2	A
F	2	A
G	2	D

10. Updated RT at node B based on C

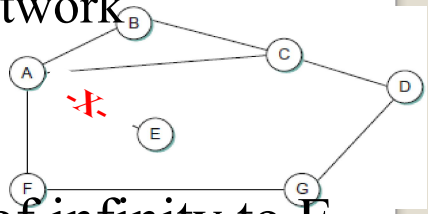
Destination	Cost	Next Hop
A	1	A
C	1	C
D	2	C
E	2	A
F	2	A
G	∞ to 3	-- to C

Note: You can see now that node B is connected to G via C, instead of via A. Similarly, A will also learn that it can reach G via C. All the nodes converge, once the updates from F and G reach all of them. All the nodes will be connected again.

Periodic update

Routing Loops

Distance Vector Algorithm: Count to infinity Problem (Example: Link between E and A fails) - Explained



- Unfortunately, slightly different circumstances can prevent the network from stabilizing.
- Suppose, for example, that the link between A and E goes down.
- In the next round of updates, **A** advertises to B, a distance of infinity to E, since B was reaching E via A, it updates its distance also to infinity, to E.
- Depending on the exact timing of events, the following might happen:
  - Before A's new update reaches C, C might give its periodic update to B a distance of 2 to E via A
  - Node B, upon hearing that E can be reached in 2 hops from C, it concludes that it can reach E in 3 hops and
  - B advertises this to A; node A concludes that it can reach E in 4 hops and advertises this latest update to C; node C concludes that it can reach E in 5 hops; and so on.
- This cycle stops only when the distances reach some number that is large enough to be considered infinite.

Note: C accepts the increased cost from A because it was originally reaching E via A only.

- In the meantime, none of the nodes actually knows that E is unreachable, and the routing tables for the network do not stabilize.
- This situation is known as the ***count to infinity*** **problem**.

Routing Loops: Description

- One of the main problems inherent with Distance Vector routing protocols is routing loops (route updates going into a loop).
- Routing loops occur in networks when old (bad) route information exists in a route table.
- The problem stems primarily from the inter-mixing of periodic route updates, sharing stale routes before receiving updates, triggered by link failures.
- Similar to what happened when the link to E failed from A.

Explained in the Previous Slide

- Because of the intervals between the periodic route updates, and the delay in the updates based on link failures, reaching all the nodes, routers may not learn about topology changes in a timely manner.
- They may be relying on outdated or incorrect route information.
- Slow convergence can result in routing loops, causing datagrams to bounce between routers endlessly if not detected, causing the routers to start a count to infinity.

Routing Loops: Remedies

Routing Loops: Remedies

- Routing protocols can take advantage of one or more loop avoidance mechanisms to minimize the impact a loop has on the network.
- These techniques, or combinations of these techniques, can minimize routing loops passing on incorrect routing information:

1. **Count to infinity**
2. **Split horizon**
3. **Poison reverse**
4. **Holddown timers**

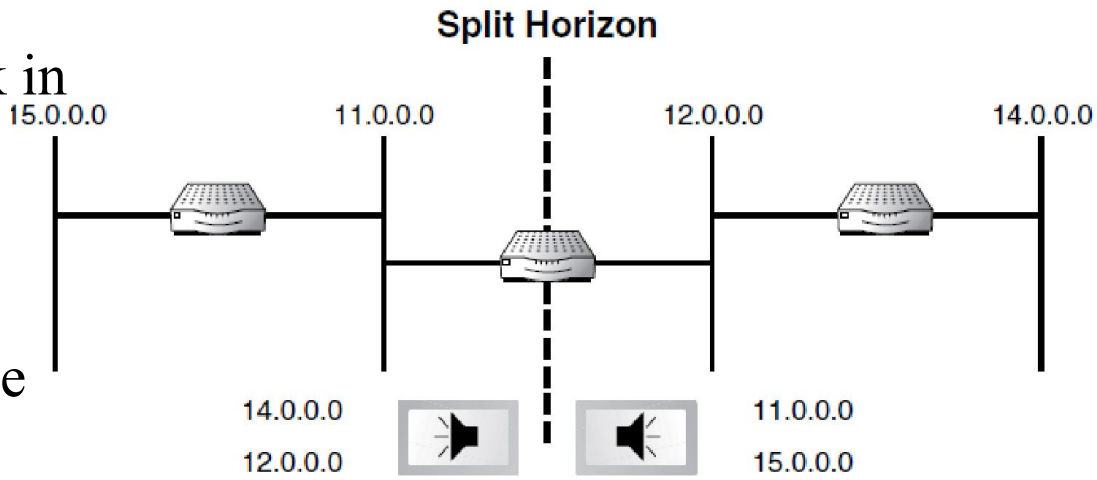
Note: Implementations vary based on vendor support and routing protocol.

1. Count to Infinity (max limit)

- Count to infinity is a loop avoidance mechanism that sets a maximum hop count value, that, when exceeded, equates to infinity hops (or destination unreachable).
- The maximum hop count for **RIP** is **15** and **IGRP** is **255**.
- Any value above this is considered **infinity** (**unreachable**).
- Distance-vector protocols limit the distance (hops) a datagram may traverse.
- If a route loop exists within the topology, the router automatically trashes the datagram when it exceeds the maximum hop set by the routing protocol running on the router.
- If for some reason datagrams continue to be forwarded after the maximum hop has been exceeded (infinity has been reached), then a fall back method, the TTL timer.
- Infinity value and TTL timer combination provides a viable remedy to most routing loops.

2. Split Horizon

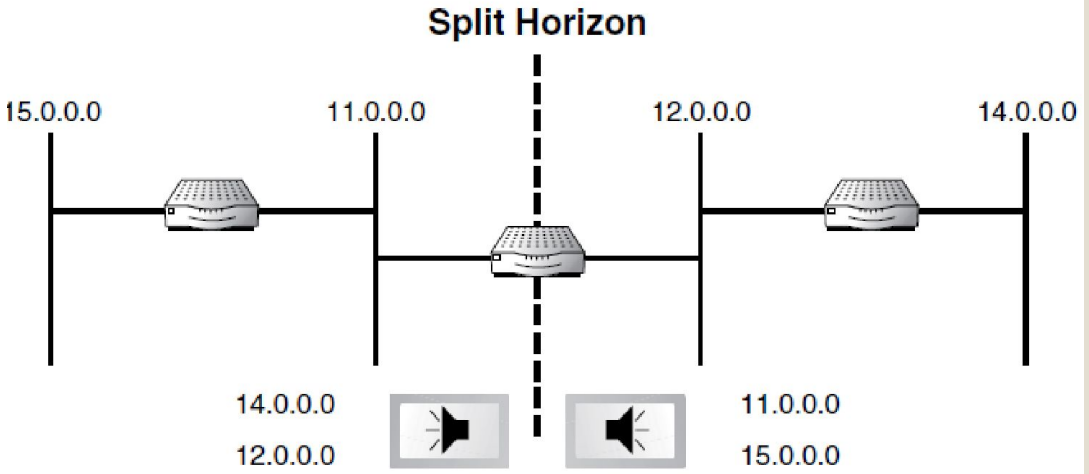
- Split horizon prevents information being sent back in the direction from which that information was received.
- When a change occurs in the network, routers only advertise that change in one direction.



- That means that they send the updates out to all other ports except the one from which it was learned.
- With split horizon, any router is the starting point.
- Split horizon sends only information learned from other ports.
- Split horizon never sends information out the same port it learned it from.

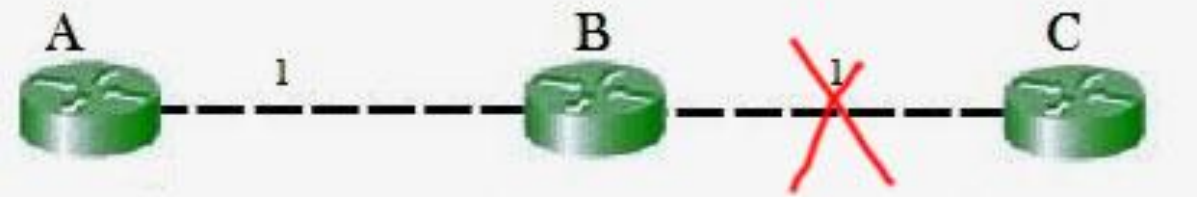
Split Horizon: Rules Explained

- The split horizon rule states that “route information learned through an interface may not be transmitted out that same interface”
- The middle router learns about networks 12.0.0.0 and 14.0.0.0 from the interface on the right.



- The router in the middle can only propagate this information out the opposite interface.
- It learns about networks 11.0.0.0 and 15.0.0.0 from the interface on the left and can only propagate this information out the opposite interface

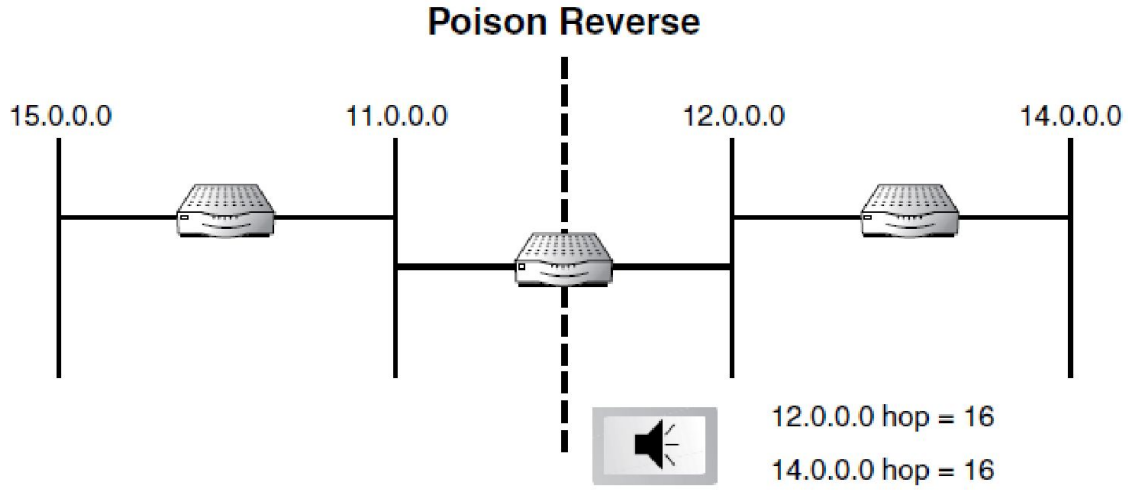
Split Horizon: Solution



- If the link between B and C goes down, and B had received a route from A , B could end up using that route via A.
- **A** would send the packet right back to B, creating a loop.
- But according to Split horizon Rule, Node A does not advertise its route for C (namely A to B to C) back to B.
- On the surface, this seems redundant since B will never route via node A because the route costs more than the direct route from B to C.
- But when the link between B and C fails, the update from A would have caused B to believe there is another longer route via A to C
- But, split horizon prevents such a confusion and C is considered unreachable quickly preventing routing loops.

3. Poison Reverse

- Poison reverse allows routers to break the split horizon rule by advertising information learned from an interface out the same interface



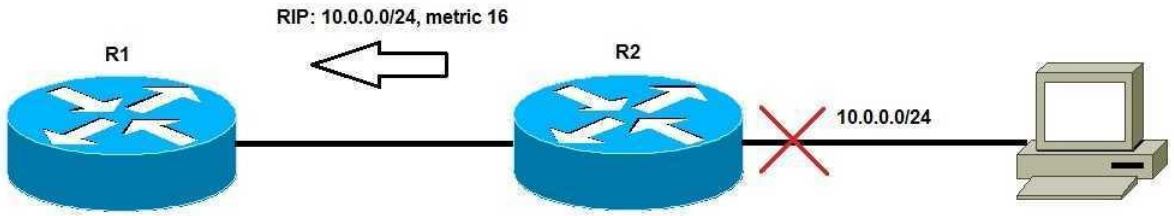
- However, it can advertise routes learned from an interface out the same interface with a 16-hop count, which indicates a destination unreachable, "**poisoning**" the route
- While the other routers slowly converge, the router maintains the poisoned route in its table and ignores updates from other routers about better routes to the poisoned network.
- Poison reverse prevents updates with inconsistencies from spreading.
- Poison reverse when implemented takes precedence over split horizon.

4. Holddown Timers

- Another technique typically used in combination with route poisoning is holddown timers.
- Holddown timers start as soon as a router receives an update from a neighbor indicating that an attached network has gone down.
- Until the timer elapses, the router ignores updates regarding this route from other routers unless it receives an update from the neighboring router that initially informed the network of the downed link.
- The timer stops if it receives a message from the neighboring router.
- At that point, the network is marked as reachable again and the route table is updated.
- Routers use holddown timers after they have learned that a route is unavailable to ensure that this route will not be mistakenly reinstated by an advertisement received from another router that has not yet learned about this route being unavailable.

4. Holddown Timers … Example

- Typically, this timer is greater than the total convergence time, providing time for accurate information to be learned, consolidated, and propagated through the network by all routers.



Note: **Default** value of **Holddown timer** is **180 seconds** in **RIP**

- We have a network of two routers. Both routers are running RIP and R2 has advertised the **10.0.0.0/24** network to R1. Consider what happens if the network fails:
  1. R2 advertises the **10.0.0.0/24** network with the infinitive metric (**16**) to R1, indicating that the network is no longer accessible.
  2. R1 receives the routing update, marks the route as unreachable, and starts the **holddown timer**.
  3. During the holddown period, R1 will not process any routing update about that route received from other routers. Only updates from R2 will be processed.

Routing Information Protocol (RIP)

Routing Information Protocol (RIP)

- The Routing Information Protocol (RIP) is one of the oldest distance-vector routing protocols used in computer networks to facilitate the exchange of routing information between routers.
- There are two versions RIP, RIP version 1 and RIP version 2.
  - RIP v1 and v2 both use **hops** as their distance value.
  - RIPv1 (1988) and RIPv2 (1993)
- Distance Vector networks such as RIP are limited in scope (**diameter**).

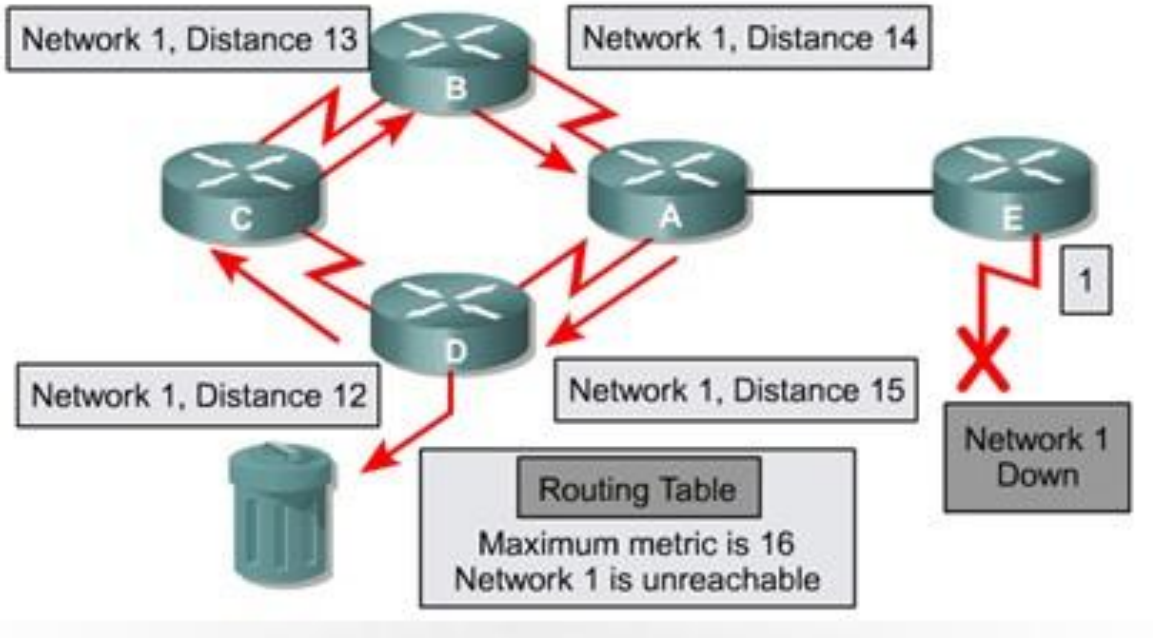
RIP: Network Diameter

- RIP is designed to work in small size networks, with the maximum hop count reaches 16, it is declared as unreachable
- RIP protocol was not designed to operate in medium to large internetworks with hundreds of links and routers connecting hundreds or thousands of hosts.
- The maximum network diameter specifies the distance a datagram may travel (for example, the maximum number of hops) before the destination is considered unreachable, causing the datagram to be discarded.
- This maximum distance is measured in hops from transmitter to receiver.
- You can think of the maximum rule as “No two devices can communicate through more than x hops.”

RIP: Network Diameter: Explained

(Destination NW, Hop Distance)

- As datagrams traverse the network, routers forward them and increment the hop count by one before passing it on to the next hop router.
- Hop count is an element in the RIP packet



- Using RIP as an example, when a datagram reaches the 15th router, that router will not forward it further, instead it must discard the datagram because a value of 16 or greater is considered too far by RIP.
- The router discarding the datagram generates an ICMP message back to the source indicating that the destination is unreachable.

Max Routing Diameter: As Infinity Distance

- Routers also use this maximum hop count value to maintain their route tables.
- When a network link fails, the router sends news of this failure in its next route update.
- It relays this news by applying a hop count one higher than the maximum (16 for RIP and 256 for IGRP) to the failed link, which indicates the distance to this network as infinity (unreachable).
- Routers receiving this news know to remove this route from their route table

Comparison: RIPv1 Vs RIPv2 Vs IGRP

Characteristic	Routing Protocol		
	RIP v1	RIP v2	IGRP
Route Updates:			
Broadcasts	X		X
Multicasts		X	
Includes entire route table	X	X	X
Periodic timer	30 seconds	30 seconds	90 seconds
Metrics:			
Hops	X	X	
Combined Metrics:			
Bandwidth and Delay			X
VLSM (updates include subnet mask also)		X	
Maximum network diameter	15 hops	15 hops	255 hops*
Authentication		X	

X means it is supported or present

Router updates include the **entire RT** and not only the modified entries.

Link State Routing Algorithm

Link State Routing Algorithm

- Link-state routing Algorithm is the second major class of intra-domain routing protocol.
- The starting assumptions for link-state routing are rather similar to those for distance-vector routing.
- Each node is assumed to be capable of finding out the state of the link to its neighbors (up or down) and the cost of each link.
- The aim of this protocol is to provide each node with enough information to enable it to find the least-cost path to any destination.
- The basic idea behind link-state protocols is very simple:
- Every node knows how to reach its directly connected neighbors, and if we make sure that the totality of this knowledge is disseminated to every node.
  - Which means that each node is given all the information that every other node has generated about its directly connected neighbours

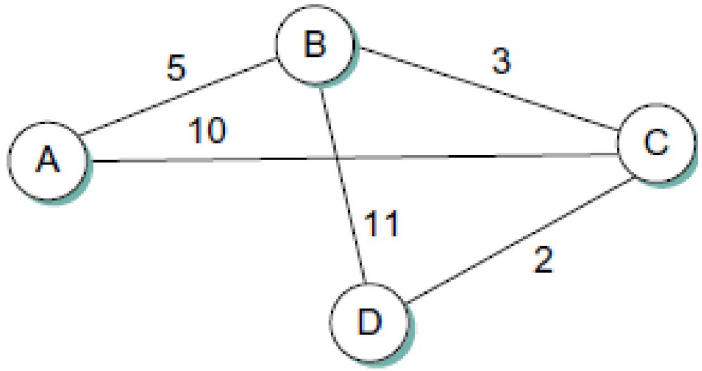
Link State Routing: Path Finding

- Then every node will have enough knowledge of the network to build a complete map of the network.
- This is clearly a sufficient condition (although not a necessary one) for finding the shortest path to any point in the network.
- Thus, link state routing protocols rely on two mechanisms:

1. Reliable dissemination of link-state information
2. The calculation of routes from the sum of all the accumulated link-state knowledge

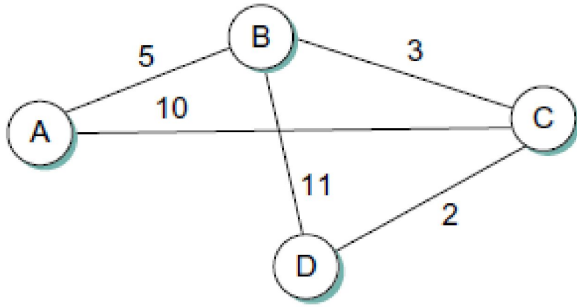
Link State Advertisement (LSA) or LSP

- LSA which is also called as **LSP** (**Link State Packet**) is the information that every router in the network shares with every other node or router in the network (routing domain) – **using reliable flooding (next slide)**



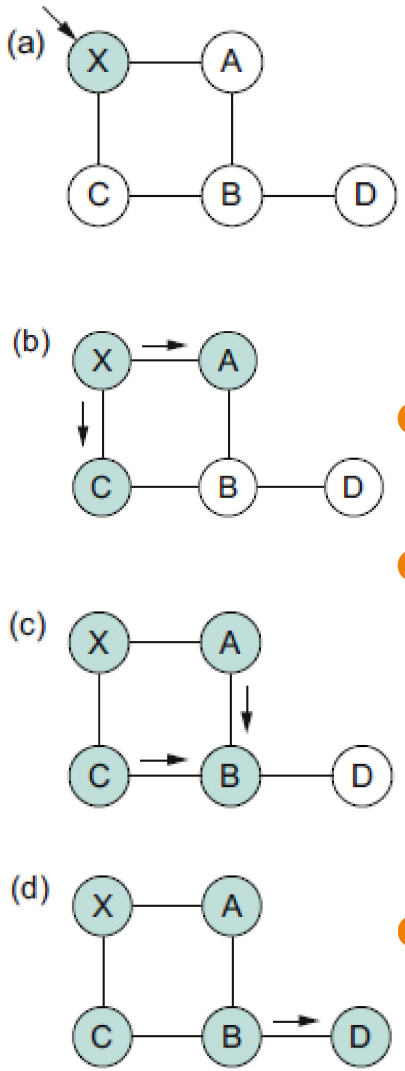
- LSA contains the following information: (For example **from D**)
- The ID of the node that created the LSP **D**
- A list of directly connected neighbors of that node, with the cost of the link to each one **(Neighbour, cost, next hop) (B, 11, B), (C, 2, C)**
- A sequence number **A 64 bits wide number which starts with zero, never expected to wrap round**
- A **LSA Age** for this packet **It is different from the TTL in IP packet**
  - It is **initialized to zero** by the originator.
  - It is **incremented** by the routers while passing it to others.
  - It is **incremented** based on the **time elapsed** while LSA is stored in the routers.
  - When it ages (1 hour), it is discarded. Originator needs to refresh it.

Reliable Flooding of LSP



- The first stage of the LSA is sharing of LSP by every node participating in the LSA with every other node in the routing domain.
- Each node **keeps** the **latest** and the **current LSP** received from every other node. (based on **Seq #**)
- Flooding works in the following way.
- First, the transmission of LSPs between adjacent routers is made reliable using **acknowledgments** for every LSP message exchanged.
- However, several more steps are necessary to reliably flood an LSP to all the nodes (routers) in the network (routing domain).
- **Sequence number** is used to make sure that the receiving node always keeps the **latest LSP** of each node and discards the older one.
- This makes sure that the correct and the latest LSP of every node is available at every other node, for each of them to independently find the shortest path to every other node in the domain. **(distributed algorithm)**

Reliable Flooding of LSP: Example



- It shows an LSP being flooded in a small network.
- Each node becomes shaded as it stores the new LSP.
- When an LSP arrives at node X, it sends it to its neighbors A and C, that is flooding the received LSP to all the other node it is connected to. (**Ref** the note below)
- A and C do not send it back to X, but they both send it on to B.
- Since B receives two identical copies of the LSP, it will accept whichever arrived first and ignore the second as a duplicate.
- B then passes the LSP onto D, which has no neighbors to flood it to, and the process is complete.
- Let us see when does a node send an LSP …

Note: Not all LSA are not accepted by a node. If a stale LSA is received it is discarded and not forwarded further (based on the Seq #).

Generation of LSP: Reasons

- Just as in RIP, each node generates LSPs under two circumstances.

1. Either the expiry of a periodic timer (**1 hour**) or
2. A change in topology causes a node to generate a new LSP.

- However, the only topology-based reason for a node to generate an LSP is, if one of its directly connected links or immediate neighbors has gone down.
- The failure of a link can be detected in some cases by the link-layer protocol.
- The demise of a neighbor or loss of connectivity to that neighbor can be detected using periodic "hello" packets.
- Each node sends "hello" to its immediate neighbors at defined intervals.
- If a sufficiently long time passes without receipt of a "hello" from a neighbor, the links to that neighbor will be declared down and a new LSP will be generated to reflect this fact, by the node which detects it.

LSA Age: (equivalent to TTL)

- LSPs also carry a time to live field element.
- This is used to ensure that old link state information is eventually removed from the network.
- A node **increments** the **LSA Age** field of a newly received LSP before flooding it to its neighbors.
- Each node also "ages" the LSP periodically while it is stored in the node.
- That is, each node increments the LSA Age of LSPs of every node it has, which is set back to received value when the node receives a new LSP
- When the LSA Age reaches **Max Age** of any node, the node re-floods that it with a value of Max Age, which is interpreted by all the nodes in the network as a signal to delete that LSP that they have with them.
- This action makes sure that any node which is dead is removed from all the nodes, by removing the LSP belonging to the dead node.

LSA: Sequence Number

- If a node goes down and then comes back up, it starts with a sequence number of 0.
- If the node was down for a long time, all the old LSPs for that node would have timed out (because of LSA Age field in LSP);
- Otherwise, this node will eventually receive a copy of its own LSP with a higher sequence number (which it had sent or flooded before going down)
- Now, it can use its own previous old sequence number by incrementing it by one and use as its own sequence number.
- By this, other nodes in the network won't even know that this node went down and came back to life ☺
- And, this also ensures that its new LSP replaces any of its old LSPs left over on other nodes before the node went down.

Dijkstra's Shortest Path algorithm (OSPF)

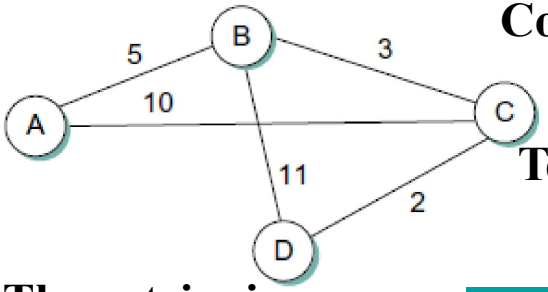
What is the next step?

- We have made sure that every node, part of the LSA routing domain has the following information with them:

1. The latest and valid LSPs of every other node which are currently alive.
2. Each node has the most reliable information about every other node which it has received from those nodes themselves, giving information about their neighbours and the cost to reach each of them from it.

- Now the next step is how every node uses this global network topology information to independently find the shortest path to all the nodes in the network routing domain.
- Each node runs **Dijkstra's Shortest Path algorithm** to find the shortest path to every other node!!!

Dijkstra's Shortest Path Algorithm: Example
(Algorithm is running on Node D)



Confirmed List: It has the shortest path from the current node to other nodes which are confirmed and finalized.

Tentative List: It has the nodes for which the shortest paths are yet to be finalized from the node which is running this algorithm.

The entries in the list are:
(Neighbor, Cost, NextHop)

The algorithm ends when all the nodes are added to the confirmed list.

Each node runs this algorithm and finds out shortest paths to all other nodes.

Table 3.14 Steps for Building Routing Table for Node D

Step	Confirmed	Tentative	Comments
1	(D,0,–)		Since D is the only new member of the confirmed list, look at its LSP.
2	(D,0,–)	(B,11,B) (C,2,C)	D's LSP says we can reach B through B at cost 11, which is better than anything else on either list, so put it on Tentative list; same for C.
3	(D,0,–) (C,2,C)	(B,11,B)	Put lowest-cost member of Tentative (C) onto Confirmed list. Next, examine LSP of newly confirmed member (C).
4	(D,0,–) (C,2,C)	(B,5,C) (A,12,C)	Cost to reach B through C is 5, so replace (B,11,B). C's LSP tells us that we can reach A at cost 12.
5	(D,0,–) (C,2,C) (B,5,C)	(A,12,C)	Move lowest-cost member of Tentative (B) to Confirmed, then look at its LSP.
6	(D,0,–) (C,2,C) (B,5,C)	(A,10,C)	Since we can reach A at cost 5 through B, replace the Tentative entry.
7	(D,0,–) (C,2,C) (B,5,C) (A,10,C)		Move lowest-cost member of Tentative (A) to Confirmed, and we are all done.

Dijkstra's Shortest Path Algorithm: Explained

1. Initialize the Confirmed list with an entry for myself; this entry has a cost of 0.
2. For the node just added to the Confirmed list in the previous step, call it node Next and select its LSP.
3. For each neighbor (Neighbor) of Next, calculate the cost (Cost) to reach this Neighbor as the sum of the cost from myself to Next and from Next to Neighbor.
   (a) If Neighbor is currently on neither the Confirmed nor the Tentative list, then add (Neighbor, Cost, NextHop) to the Tentative list, where NextHop is the direction I go to reach Next.
   (b) If Neighbor is currently on the Tentative list, and the Cost is less than the currently listed cost for Neighbor, then replace the current entry with (Neighbor, Cost, NextHop), where NextHop is the direction I go to reach Next.
4. If the Tentative list is empty, stop. Otherwise, pick the entry from the Tentative list with the lowest cost, move it to the Confirmed list, and return to step 2.

OSPF: Open Shortest Path First

- **Open Shortest Path First** (**OSPF**) is an adaptive routing protocol for Internet Protocol (IP) networks. It uses a link state routing algorithm and falls into the group of interior routing protocols, operating within a single **autonomous system** (AS).
- OSPF is perhaps the most widely-used interior gateway protocol (IGP) in large enterprise networks. **IS-IS**, another link-state dynamic routing protocol, is more common in large service provider networks.

Difference Between DVA and LSA

The difference between the distance-vector and link-state algorithms can be summarized as follows. In distance-vector, each node talks only to its directly connected neighbors, but it tells them everything it has learned (i.e., distance to all nodes). In link-state, each node talks to all other nodes, but it tells them only what it knows for sure (i.e., only the state of its directly connected links).

RIPv1 Vs RIPv2 Vs OSPF

Features	RIP		OSPF
	Version 1	Version 2	
Algorithm	Bellman-Ford		Dijkstra
Path Selection	Hop based		Shortest Path
Routing	Classful	Classless	Classless
Transmission	Broadcast	Multicast	Multicast
Administrative Distance	120		*110*
Hop Count Limitation	15		No Limitation
Authentication	No	MD5	MD5
Protocol	UDP		IP
Convergence Time	RIP>OSPF		

Administrative distance (AD) or route preference: is a number of arbitrary unit assigned to dynamic routes, static routes and directly-connected routes. The value is used by vendor-specific routers to rank routes from most preferred (low administrative distance value) to least preferred (high administrative distance value)

Internet Domains

What is a Domain?

- A domain is the web address that you type in or search for when you browse the internet.
  - **Example**: Google.com, Yahoo.com, Netflix.com, Amazon.com, Wikipedia.com, and so on.
- There are managed by **ICANN** (Internet Corporation for Assigned Names and Numbers)
- Way back in 1985 when the internet domain name system was implemented, there were 7 top level domains:
  - **com, edu, net, org, arpa, gov, and mil**
- The most popular domain extension is **.com** followed by **.net** and **.org**.
- As of the end of the **second quarter of 2024**, there were an estimated **362.4 million** domain name registrations across all top-level domains (**TLDs**).
- A domain in networking refers to an Autonomous System (**AS**) that operates under a single administrative control.

Internet Domain Name Structure

Component	Value	Description
Scheme	**http**	Protocol used to access the resource (HTTP).
Subdomain	**www**	Indicates the World Wide Web subdomain, typically used for web services.
SLD	**google**	The main domain name representing the entity (Google Inc.).
TLD	**.com**	Top-level domain indicating a commercial entity.

SLD: Second Level Domain

Top Level Domains (TLDs) - Info

Top-Level Domains (examples)

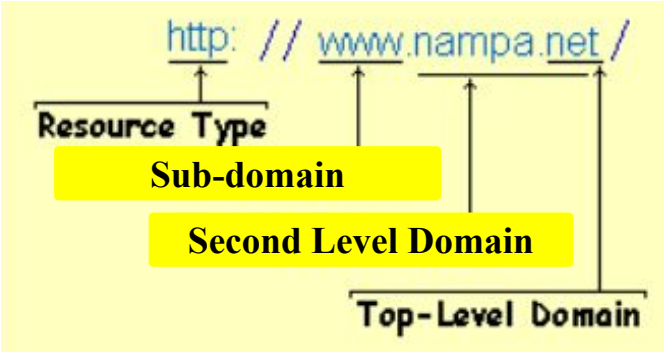
.com	Commercial
.net	Network etc
.org	Non-profit
.edu	Education
.mil	US Military
.int	International
.gov	US Government
.biz	Business
.info	Information

Note: The most popular TLD is:

.com (37%)

More about domains.

Internet Domain Structure: Explained



Top-Level Domain **Generic TLD (gTLD)**

The top level domain is a huge group of similar types of domain names. Some are restricted to only certain people, like .gov **Other Examples: .org, .net, .edu, .mil**

Second Level Domain Name

Registered domain name such as google

Sub-domain

The **www** subdomain is commonly used for the main website, but many other subdomains can be used for different purposes

Other alternative sub-domains:

Common: blog, mail, support, help, etc.
Location-based: us, uk, in, etc.
Business: admin, portal, hr, test, etc.

Resource Type

The type of page. Typical web pages are HTTP, an acronym for 'Hyper Text Transfer Protocol'. HTTPS is secure HTTP. Other resources include, FTP, GOPHER, TELNET etc.

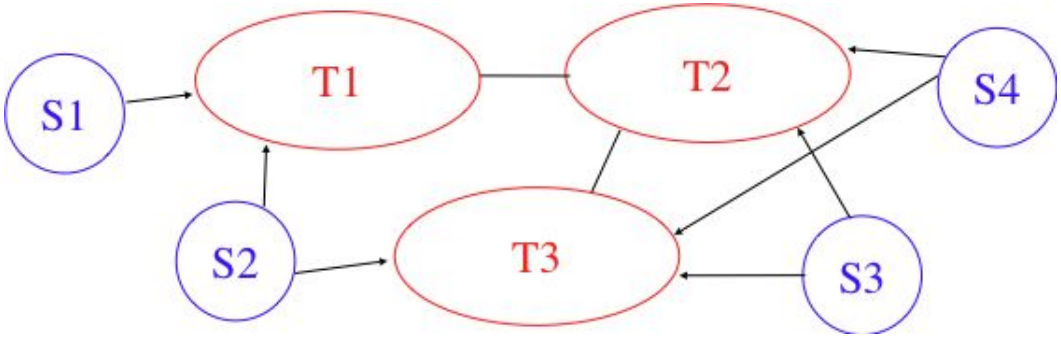
Most visited Websites (as on Nov 24) - Info

Ranking	Website	Monthly Traffic
1	google.com	132,300,000,000
2	youtube.com	72,000,000,000
3	facebook.com	12,900,000,000
4	wikipedia.org	6,700,000,000
5	instagram.com	6,500,000,000
6	reddit.com	5,700,000,000
7	bing.com	4,800,000,000
8	taboola.com	4,200,000,000
9	x.com	4,100,000,000
10	whatsapp.com	3,900,000,000

Source

Domain Types: Stub and Transit

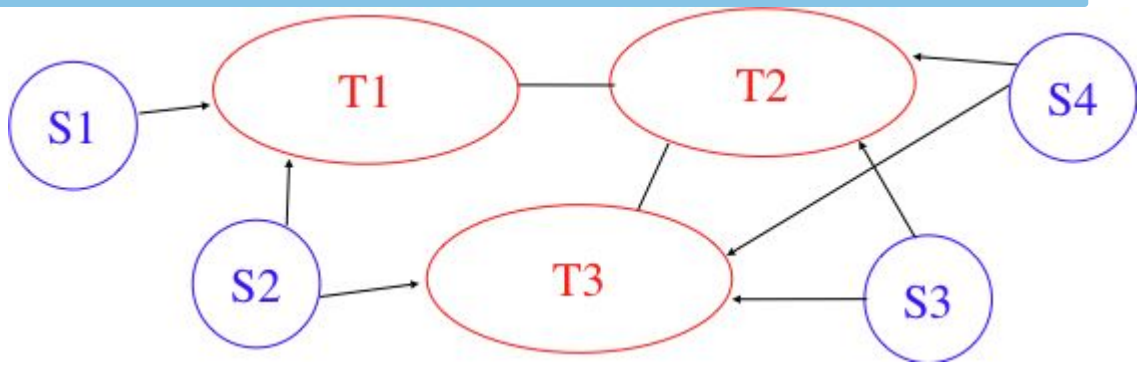
Stub: The short part of something that is left after the main part has been used



- Each domain contains a set of routers.
- Based on how **traffic flows** through them, domains can be classified into **Stub** Domains and **Transit** Domains
- A **stub** domain sends and receives packets whose source or destination are one of its own hosts.
  - Stub domain routers have default routes pointing to their only upstream provider.
- A **transit** domain is a domain that provides a transit service for other domains.
- That is, routers in the transit domain forward packets whose source and destination do not belong to the transit domain.
  - About **85%** of the **domains** in the **Internet** are **stub domains**

Domain Types: Stub and Transit

- S1 to S4 are **Stub domains**
- **T1 to T3** are **Transit domains**



- A **stub** domain that is **connected** to a **single transit domain** is called a **single-homed stub**.
- A **multi-homed stub** is a **stub** domain **connected** to **two or more transit providers**.
- List the following:
  - **Single-homed stub domains**: **S1**
  - **Multi-homed stub domains**: **S2, S3 and S4**

Ref: Domain Names

Stub Vs Transit Domains

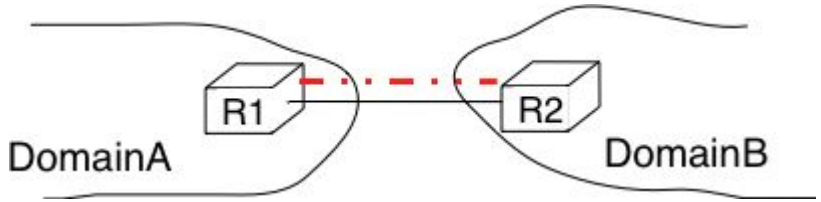
Feature	Stub Domain	Transit Domain
Connectivity	Single connection to another AS	Connected to multiple ASes
Purpose	Only sends/receives its own traffic	Forwards traffic between ASes
Routing Protocols	Uses static routes, OSPF, EIGRP	Uses BGP for external routing
Traffic Handling	Does not carry transit traffic	Routes traffic for other ASes
Examples	Small office, enterprise network	ISP, cloud provider, backbone networks

AS: Autonomous System

BGP: Border Gateway Protocol

EIGRP: Enhanced Interior Gateway Routing Protocol

Interconnection of Domains



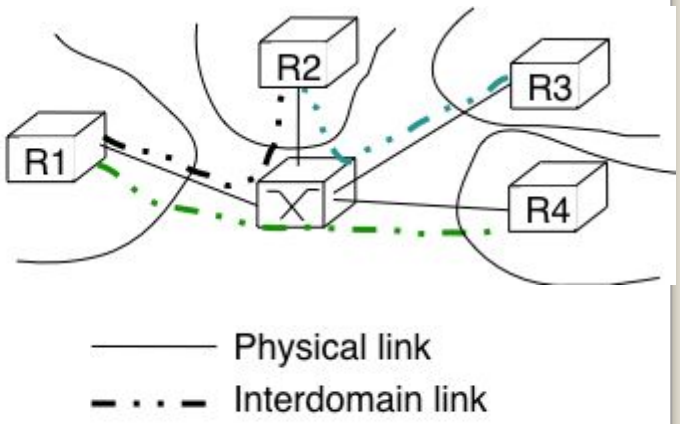
Note: Private peering links are useful when, an enterprise or university network needs to be connected to its ISP.

Physical link
Interdomain link

- Domains need to be interconnected to allow a host inside a domain to exchange IP packets with hosts located in other domains.
- From a physical perspective, domains can be interconnected in two different ways.
- The first solution is to directly connect a router belonging to the first domain with a router inside the second domain.
  - Such links between domains are called **private inter-domain** links or **private peering links**.
- In practice, for redundancy or performance reasons, **distinct physical links** are usually established **between different routers in the two domains** that are **interconnected**.

IXP: Internet Exchange Point

- However, some domains are required to be connected to hundreds of other domains.
- For some of these domains, using only private peering links would be too costly.
- A better solution to allow many domains to interconnect cheaply are the Internet eXchange Points (**IXP**)



- An IXP is usually some space in a data center that hosts routers belonging to different domains.
- A domain willing to exchange packets with other domains present at the IXP, installs one of its routers on the IXP and connects it to other routers inside its own network.
- The IXP contains a Local Area Network to which all the participating routers are connected.

IXPs in India: Mumbai, Chennai, Delhi, Kolkata, Mumbai, Bardhaman: National Internet Exchange of India (**NIXI**)

As of Dec 2023, 40 IXPs are in India

ISPs (Tier-1, 2 and 3) - Info

- **Tier-1 ISPs (Backbone Providers)**
  - Earn revenue from lower-tier ISPs, large enterprises, and CDNs.
  - **Examples**: **AT&T, Tata Communications, CenturyLink, NTT, etc**.
- **Tier-2 ISPs (Regional ISPs)**
  - Pay Tier-1 ISPs for Internet access.
  - Earn revenue from Tier-3 ISPs and businesses.
  - **Examples**: **Vodafone, Airtel, Reliance Jio, Comcast.**
- **Tier-3 ISPs (Local ISPs)**
  - Pay Tier-2 ISPs for Internet access.
  - Earn revenue from end-users (home users, businesses).
  - **Examples: BSNL, ACT Fibernet, Local cable ISPs.**

CDN: Content Delivery Network
CDN providers: Cloudflare, Akamai, Amazon CloudFront, Google Cloud CDN, Fastly

Inter-domain Routing
Peering Relationships

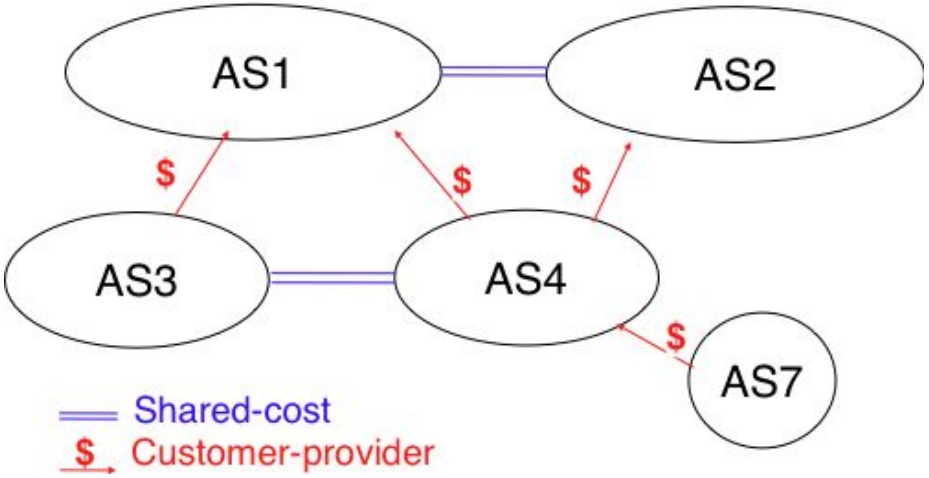
Inter-domain Vs Intra-domain Routing

- In today's highly commercial Internet, **inter-domain** routing mainly takes into account the **economical relationships** between the domains.
  - For inter-domain routing, the cost of using a route is often more important than the quality of the route measured by its delay or bandwidth
- I**ntra-domain** routing usually prefers some routes over others based on their technical merits
  - For example, minimum number of hops, minimum delay, high bandwidth routes over low bandwidth ones, etc

Inter-domain Routing: Peering Relationships

AS: **Autonomous Systems**

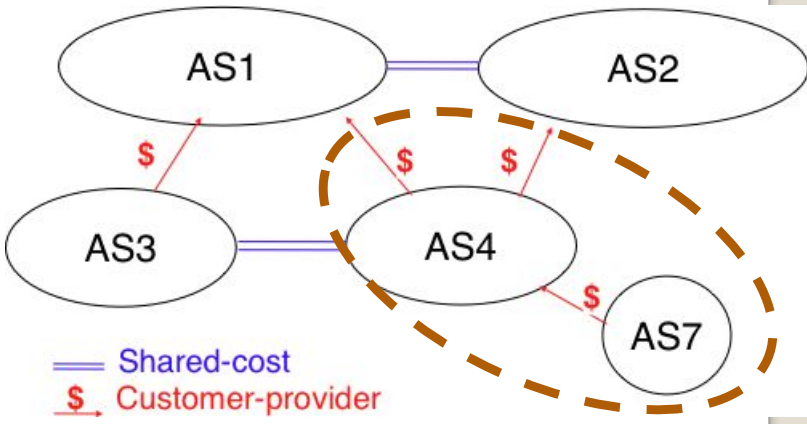
A unit of router policy, either a single network or a group of networks that is controlled by a common network administrator



- The peering relationship in Inter-domain is the **customer □ provider** relationship.
- Such a relationship is used when a customer domain pays an Internet Service Provider to be able to exchange packets with the global Internet over an inter-domain link.
- A similar relationship is used when a small ISP pays a larger ISP to exchange packets with the global Internet.

Customer Provider Relationship

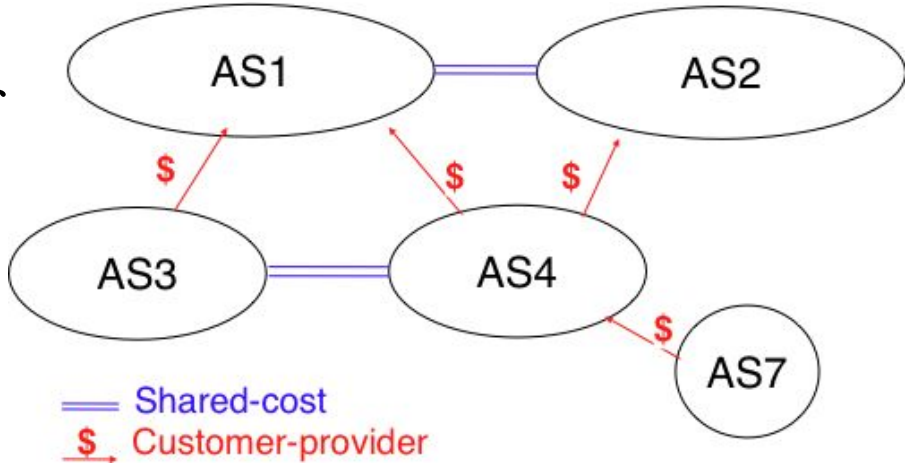
- AS7 is a stub domain that is connected to one provider, AS4.
- The contract between AS4 and AS7 allows a host inside AS7 to exchange packets with any host in the internetwork



- To enable this exchange of packets, AS7 must know a route towards any domain and all the domains of the Internet must know a route via AS4 that allows them to reach hosts inside AS7.
- From a routing perspective, the commercial contract between AS7 and AS4 leads to the following routes being exchanged:
  1. Over a **customer->provider** relationship, the customer domain advertises to its provider all its routes and all the routes that it has learned from its own customers.
  2. Over a **provider->customer** relationship, the provider advertises all the routes that it knows to its customer.

Customer Provider Relationship

- The first rule allows the routes of the customer domain to be distributed throughout the Internet.

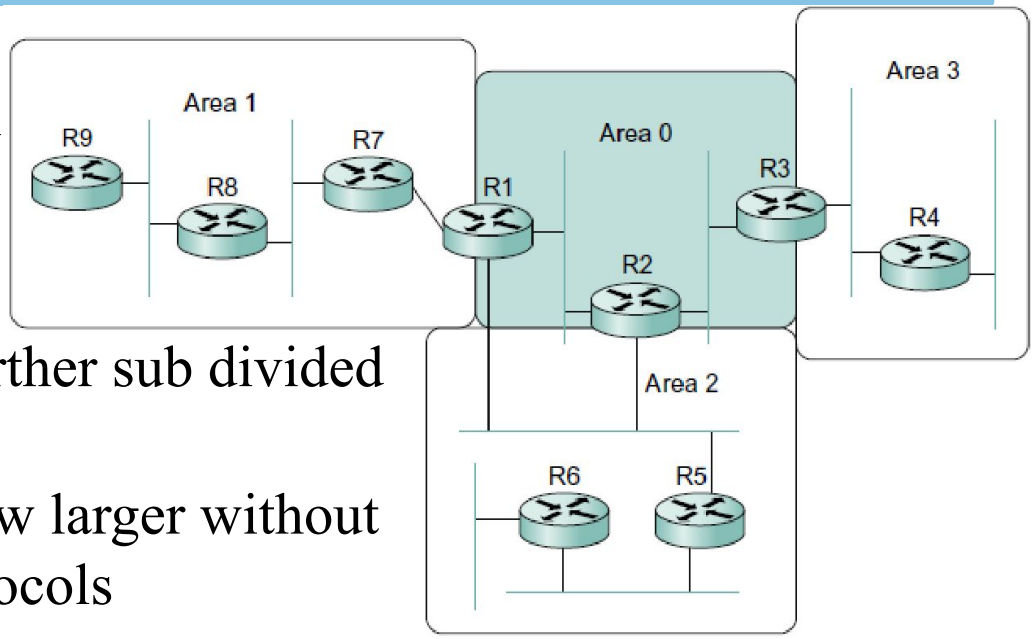


- The second rule ensures that the customer domain receives a route towards all destinations that are reachable via its provider.
- Coming back to the figure above, AS4 advertises to its two providers AS1 and AS2 its own routes and the routes learned from its customer, AS7.
- On the other hand, AS4 advertises to AS7 all the routes that it knows.
- A **shared-cost peering** relationship is usually established between domains having a similar size and geographic coverage.
  - Usually, it does not involve a payment from one domain to the other.

Routing Areas

Routing Domains and Areas

- An area is a set of routers that are administratively configured to exchange link-state information with each other.

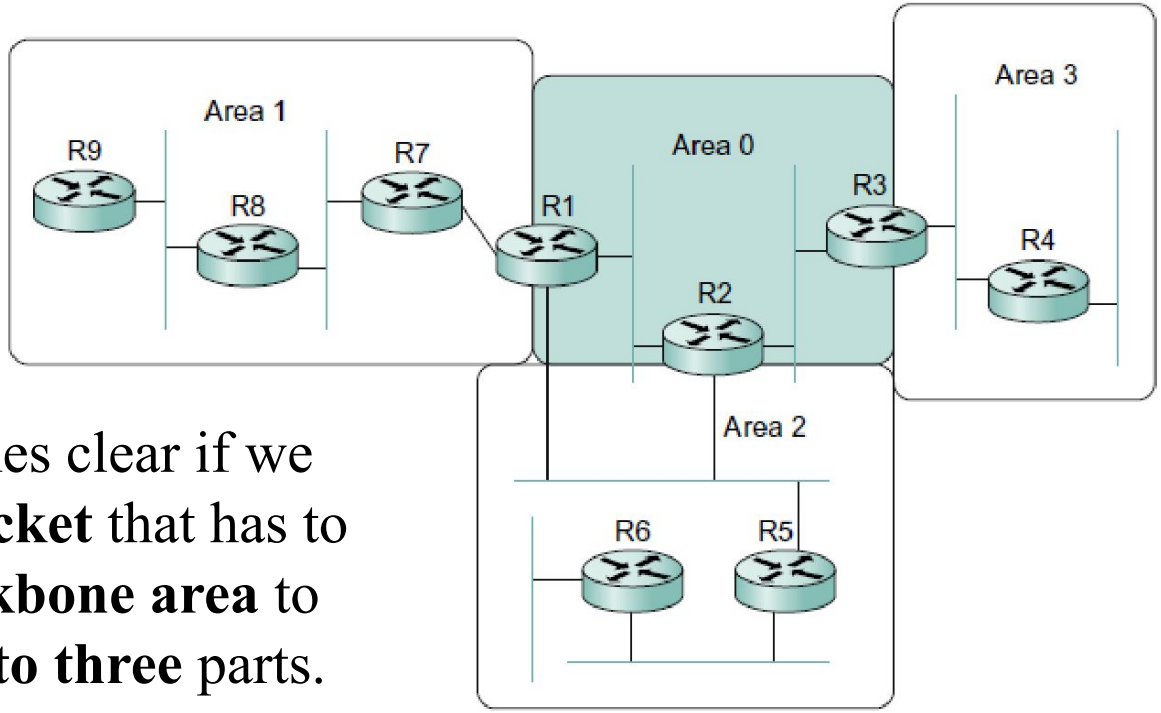


- An OSPF routing domain is further sub divided into subdomains called **areas.**
- It enables single domain to grow larger without overburdening the routing protocols

- There is one special area—the backbone area, also known as area 0.
- Routers R1, R2, and R3 are members of the backbone area.
- They are also members of at least one non-backbone area; R1 is actually a member of both area 1 and area 2.
- A router that is a member of both the backbone area and a non-backbone area is an Area Border Router (**ABR**)

Routing Between Areas

- How does a router in one area determine the right next hop for a packet destined to a network in another area?

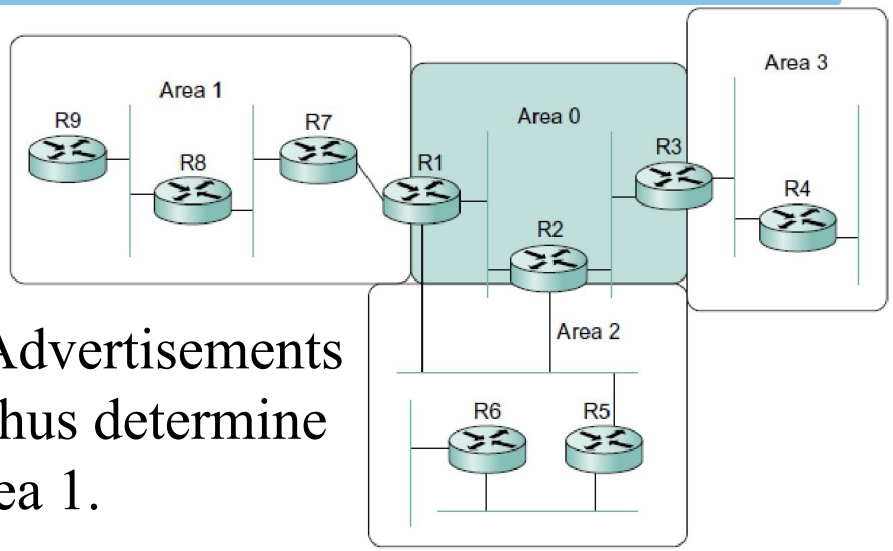


- The answer to this becomes clear if we imagine the **path of a packet** that has to travel from **one non-backbone area** to **another** as being **split into three** parts.

1. It travels from its source network to the backbone area.
2. It crosses the source network and enters into backbone area
3. Then, finally the packet travels from the backbone area to the destination network.

Routing Between Areas: Explained

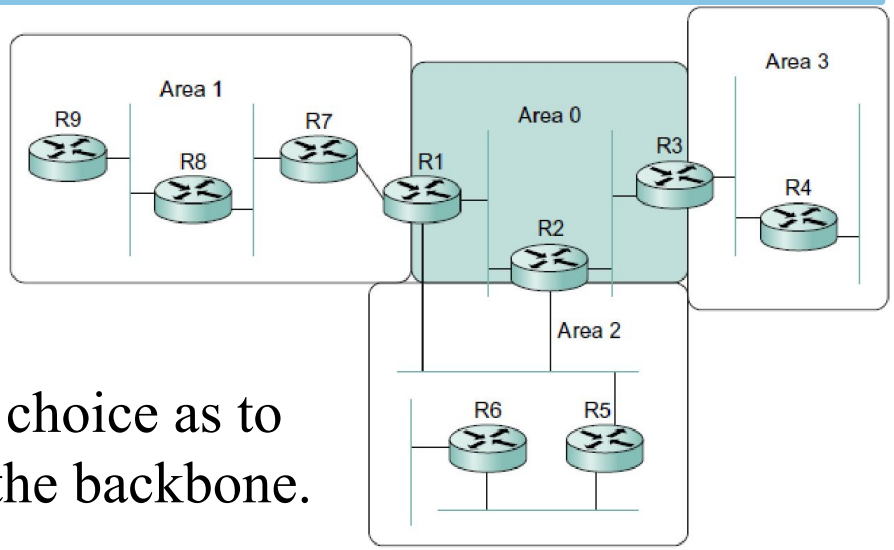
- The ABRs summarize routing information that they have learned from one area and make it available in their advertisements to other areas.



- For example, R1 receives Link-State Advertisements from all the routers in area 1 and can thus determine the cost of reaching any network in area 1.
- When R1 sends LSP into area 0, it advertises the costs of reaching the networks in area 1 much as if all those networks were directly connected to R1.
- This enables all the area 0 routers to learn the cost to reach all networks in area 1.
- The area border routers then summarize this information and advertise it into the non-backbone areas.
- Thus, all routers learn how to reach all the networks in the domain.

Areas with more than one ABRs (Area 2)

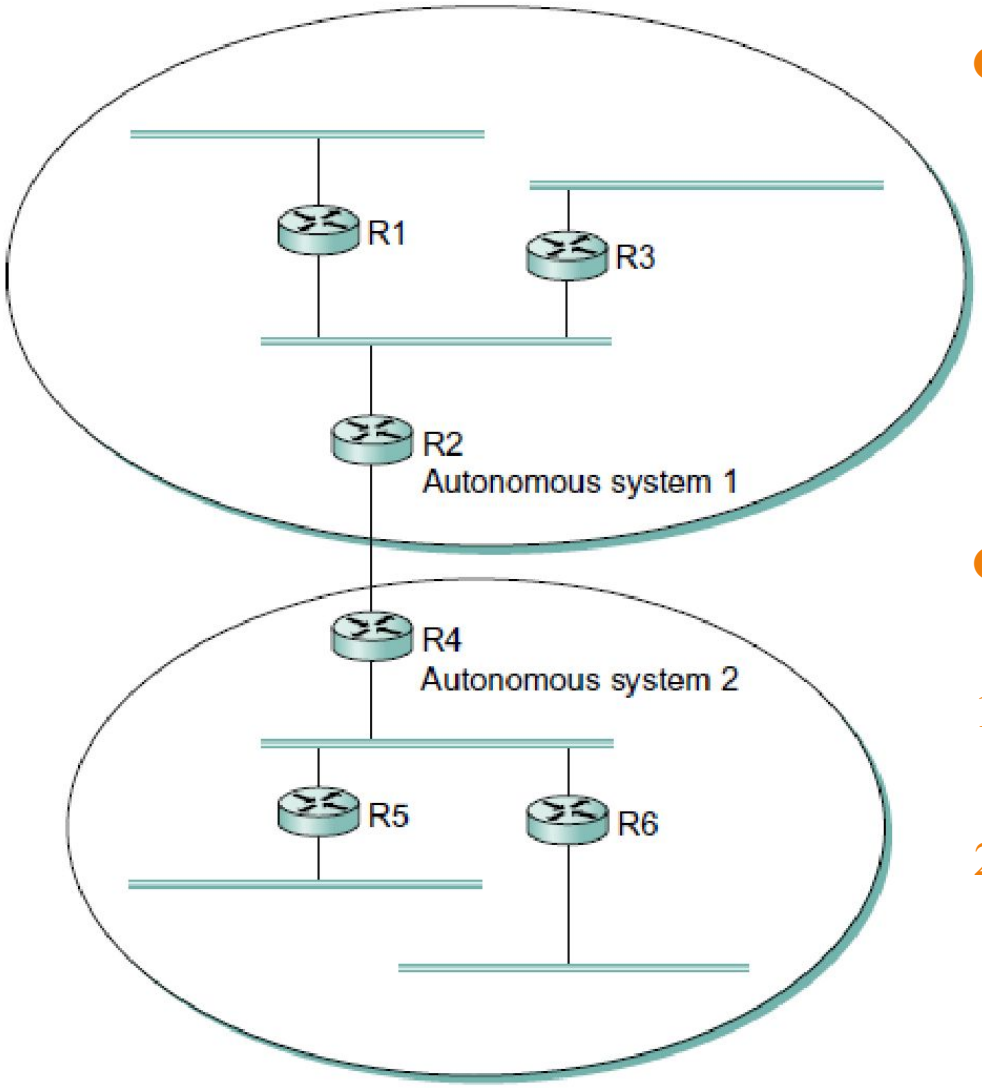
- Note that, in the case of area 2, there are two ABRs, which are R1 and R2.



- Thus area 2 routers have to make a choice as to which one they would use to reach the backbone.
- This is easy enough, since both R1 and R2 will be advertising costs to various networks, so it will become clear which is the better choice as the routers in area 2 run their shortest-path algorithm.
- For example, it is pretty clear that R1 is going to be a better choice than R2 for destinations in area 1.
- The use of areas forces all packets traveling from one area to another to go via the backbone area, even if a shorter path might have been available.

Autonomous Systems

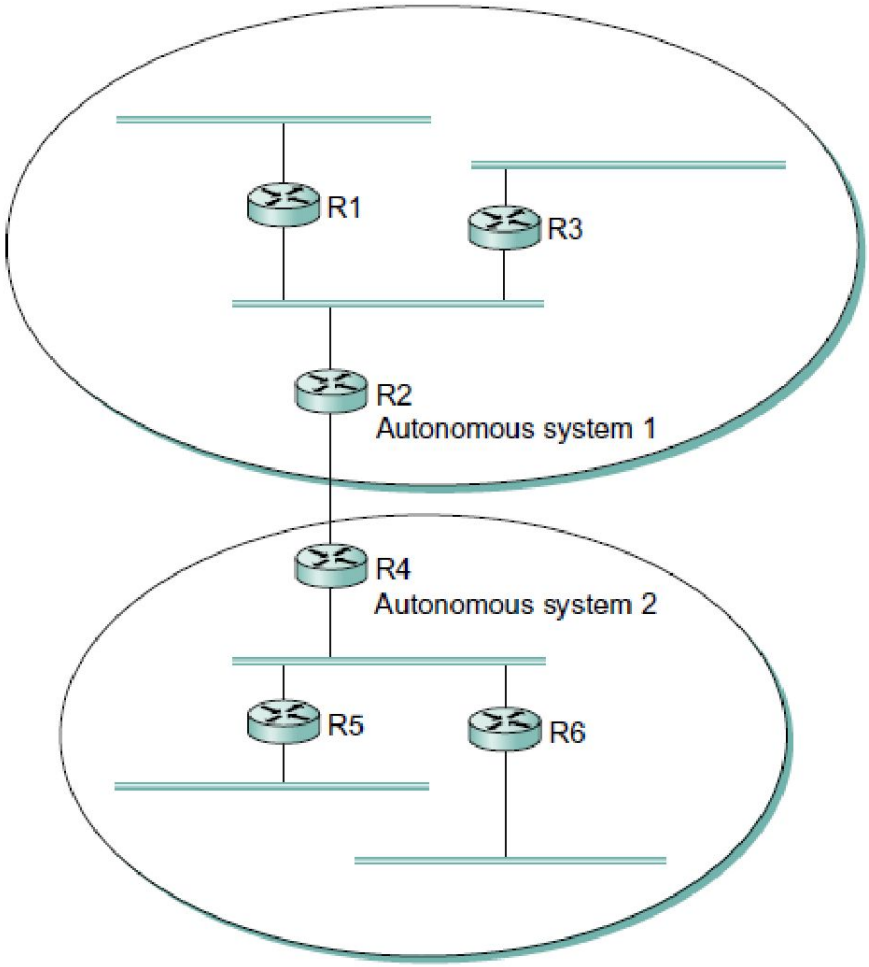
Interconnection of Autonomous Systems



- The basic idea behind autonomous systems is to provide an additional way to hierarchically aggregate routing information in a large internet, thus improving scalability.
- We now divide the routing problem into two parts:

1. Routing within a single autonomous system and
2. Routing between autonomous systems.

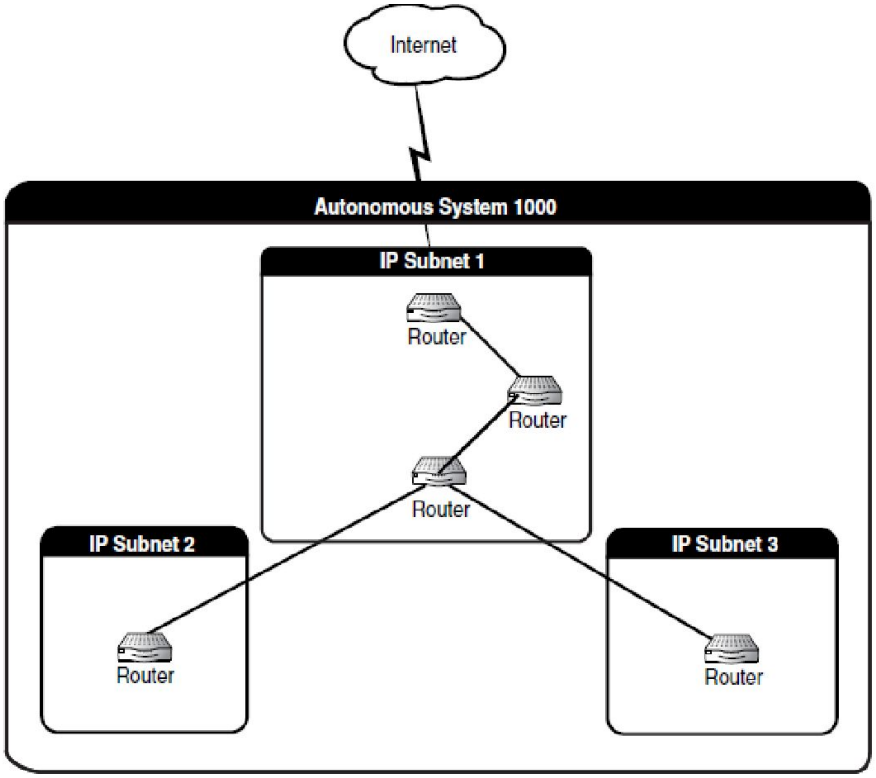
Autonomous Systems: Inter-domain Routing



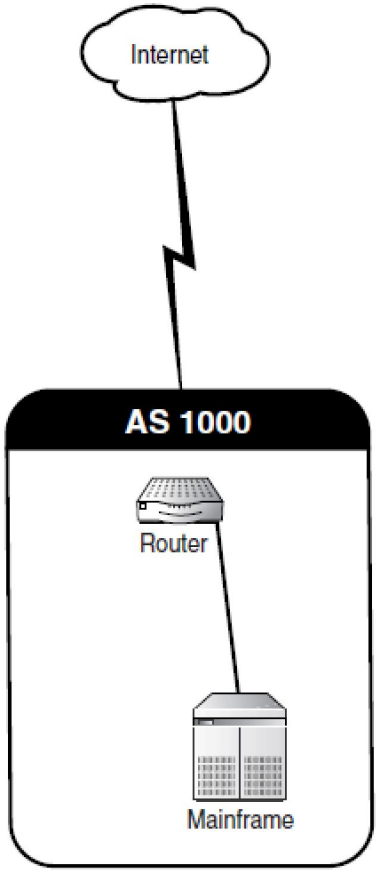
- The AS model decouples the intra-domain routing that takes place in one AS from that taking place in another.
- Thus, each AS can run whatever intra-domain routing protocols it chooses.
- It can even use static routes or multiple protocols, if desired.
- The inter-domain routing problem is then one of having different ASs share reachability information—descriptions of the set of IP addresses that can be reached via a given AS—with each other.

Autonomous System Numbers (ASN)

Autonomous Systems: Examples



An Autonomous System is a collection of subnets, routers and routing protocols



A single Autonomous System with one router.

Autonomous System Numbers (ASN)

- An Autonomous System Number, like an IP address, must be assigned by a governing body and it is a 32-bit integer.
- In most cases the ISPs assign ASN as a subset of its own.
- In the United States, the governing body in charge of registering and releasing Autonomous System numbers is **ARIN** (the American Registry for Internet Numbers).
- There are two main categories of ASNs, **public** and **private**.
- Public numbers are assigned to entities requiring their network be advertised to the Internet.
- Most often ISPs and other large, global companies are assigned public ASNs.

Public ASN Example:

Internet Service Providers (ISPs): AT&T, Verizon, Tata Communications
Cloud Providers: AWS, Google Cloud, Azure
Large Enterprises that operate their own global networks: Facebook, Netflix

Private ASNs

- Like their IP address counterparts these numbers cannot be advertised to the Internet and are not routable.
- Rather these numbers are used for iBGP (Internal BGP) routing within a larger BGP network.
- The numbers in the private ASN range can be used freely by anyone.
- To qualify for a public Autonomous System number, a network needs to supply a proof of **multi-homing**.
- Typically, multi-homed connections are used to load balance traffic and provide fault tolerance.

BGP: Border Gateway Protocol

Summary of ASN Allocation - Info

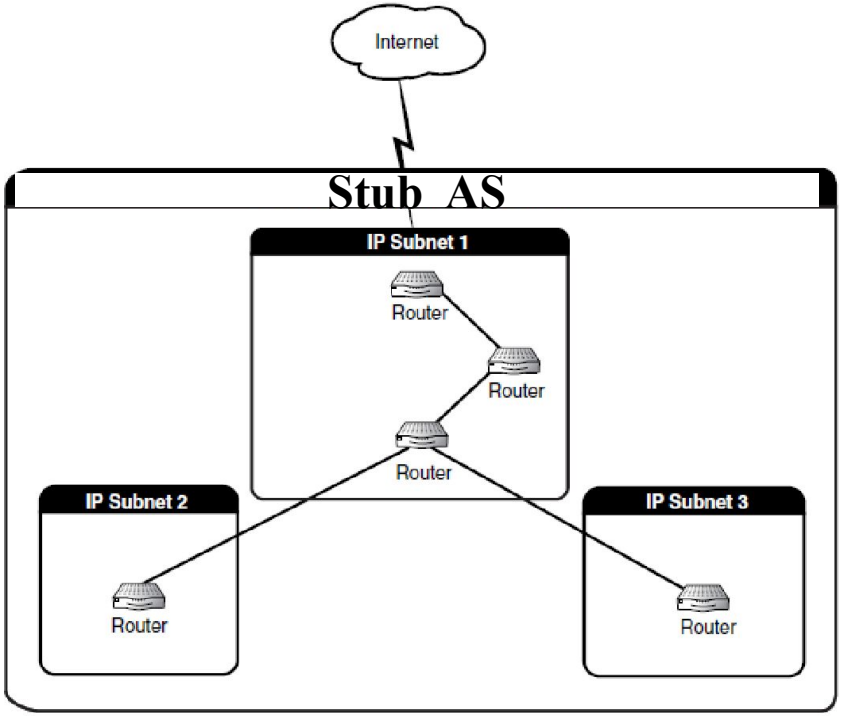
ASN Table [edit]

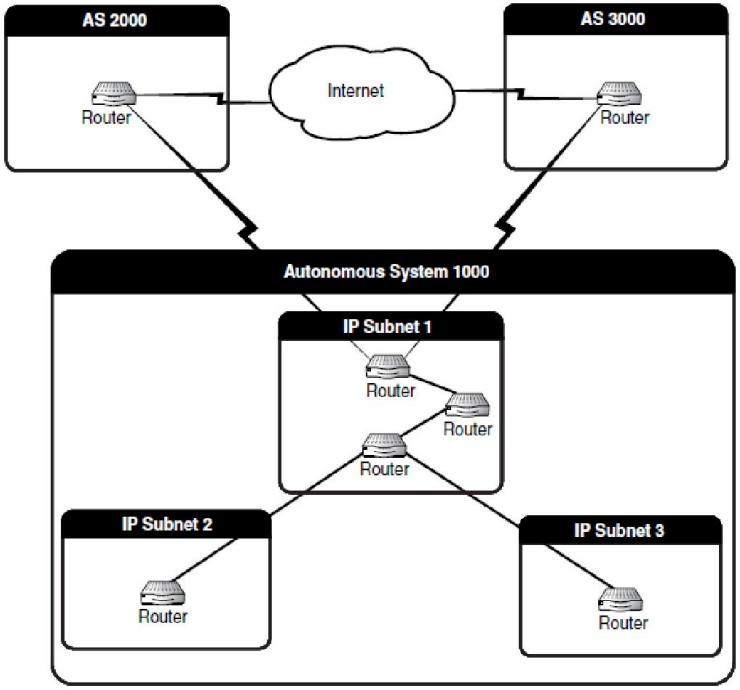
A complete table of 16-bits and 32-bits ASN available:[8]

Number	Bits	Description	Reference
0	16	Reserved	RFC1930
1 - 23455	16	Public ASN's	
23456	16	Reserved for AS Pool Transition	RFC6793
23457 - 64534	16	Public ASN's	
64000 - 64495	16	Reserved by IANA	
64496 - 64511	16	Reserved for use in documentation/sample code	RFC5398
64512 - 65534	16	Reserved for private use	
65535	16	Reserved	
65536 - 65551	32	Reserved for use in documentation and sample code	RFC4893, RFC5398
65552 - 131071	32	Reserved	
131072 - 4199999999	32	Public 32-bit ASN's	
4200000000 - 4294967294	32	Reserved for private use	RFC6996
4294967295	32	Reserved	

Note: Prior to the current 32-bit ASNs, 16-bit ASN was in use.

Three Classes of ASs





A stub AS has a single link connecting it to another AS, with only one way in and out.

The above AS cannot be a public ASN.

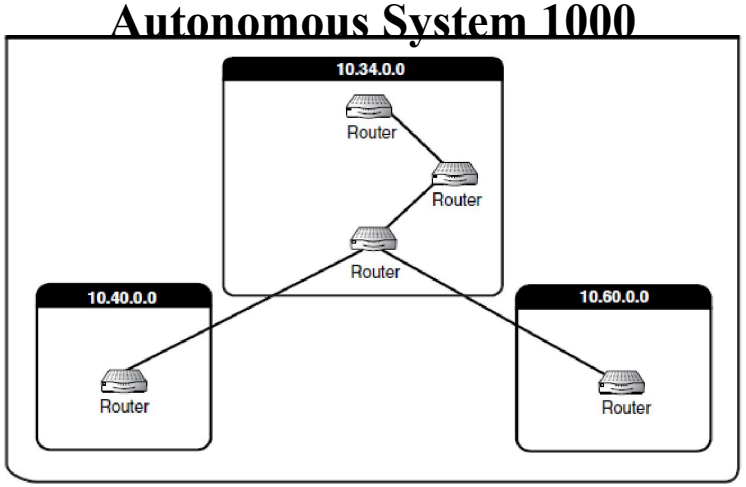
Multihomed AS 1000 has connections to two other AS', 2000 and 3000.

The above AS can be assigned a Public ASN

Third type of **AS** is **Transit AS**

ASNs and IP Addresses

- IP addresses play a big part in the operation of BGP and the formation of Autonomous Systems.
- An ASN needs to be associated with the IP address segments of the network to which it is attached.



- This ensures the traffic destined to a particular AS gets routed to it properly
- BGP uses ASNs to route packets to different ASes.
- Having a proper IP addresses configured for the ASNs ensure that any external AS, route the correct packets to the relevant **border gateways**.
- If a subnet is left out or IP addressing scheme is changed without modifying the ASN, the network will not correctly receive data bound for it.

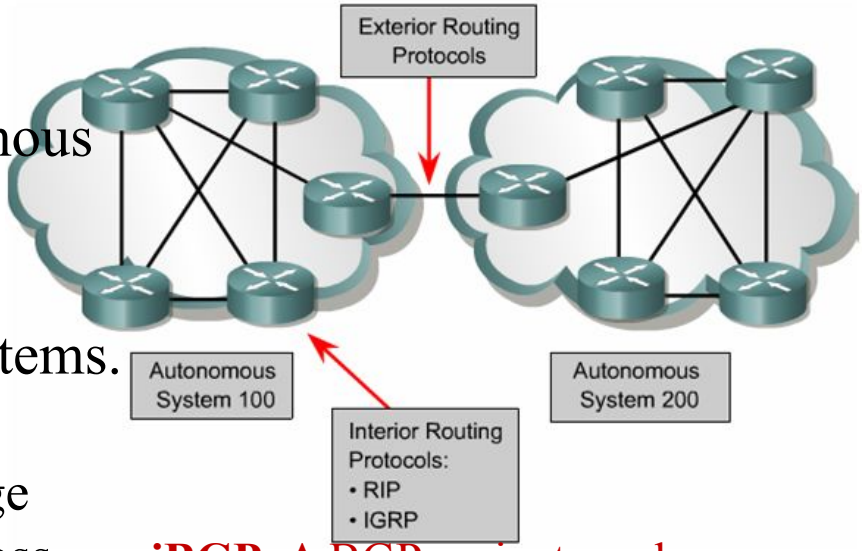
Border Gateway Protocol (BGP)

The Border Gateway Protocol (**BGP**) is the inter-domain routing protocol of the Internet. It is the protocol that connects tens of thousands of networks in the Internet to form one big interconnected network. It is the only widely used inter-domain routing protocol in the Internet and is therefore very important for the correct functioning of the Internet.

BGP: Introduction

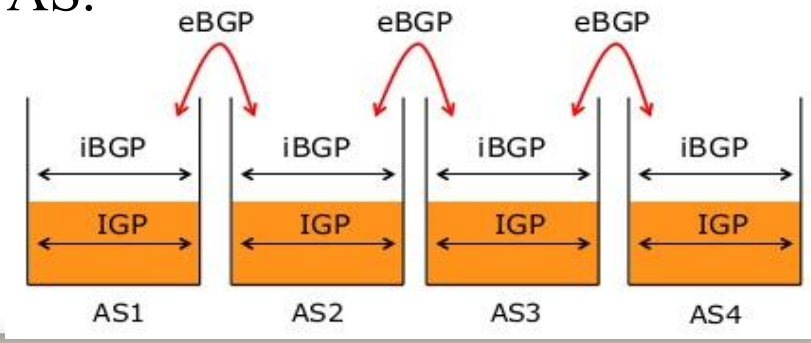
iBGP: Internal BGP
eBGP: External BGP

- BGP allows you to create loop-free inter-domain routing between autonomous systems (ASs).
- BGP exchanges network reachability information between Autonomous Systems.
  - **IGP** handles internal routing within a network, while **iBGP** and **eBGP** manage routing between different networks across autonomous systems.

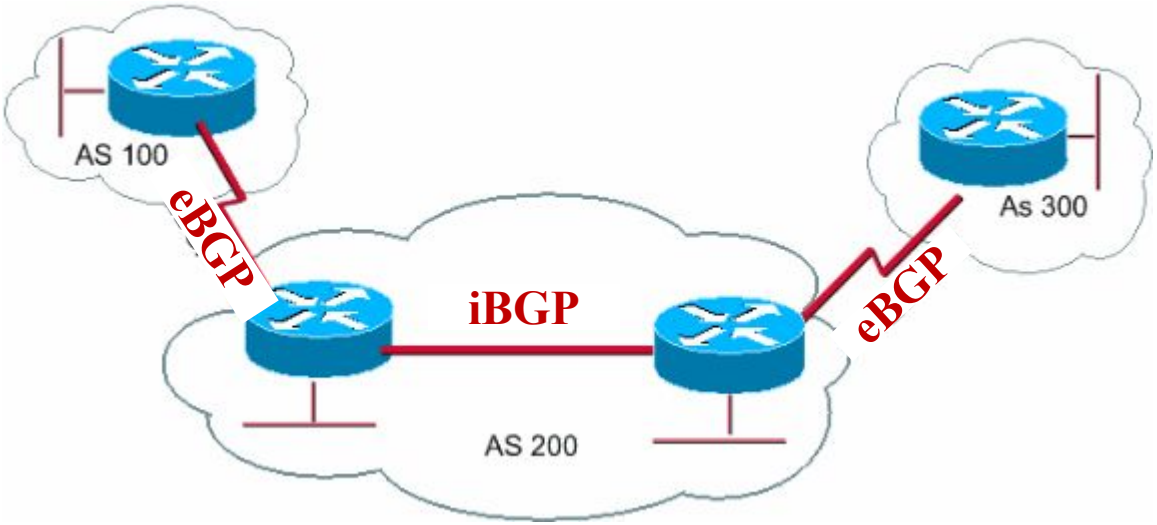


iBGP: A BGP variant used within an AS. Has full mesh of connections within an AS.

- Routers in an AS use multiple Interior Gateway Protocols (IGPs) to exchange routing information within the AS.
  - Example: RIP, OSPF, IGRP, etc.
- The routers use an exterior gateway protocol to route packets outside the AS.
  - Example: **BGP**



iBGP and eBGP

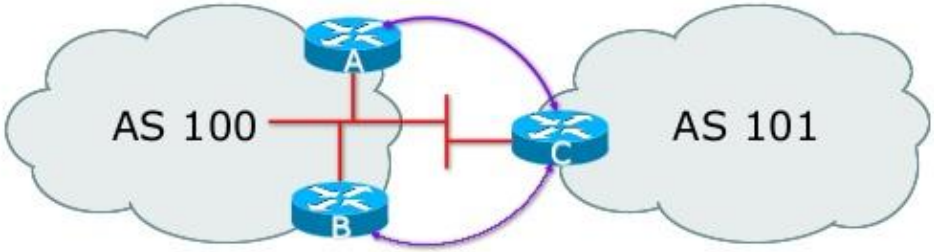


- Internal BGP (**iBGP**) is for **peering** between routers inside an AS.
- When BGP runs between routers that belong to two different ASs, it is called as exterior BGP (**eBGP**).
- eBGP takes care of exchanging the routing information across different ASs.
- Whereas, iBGP's responsibility is to disseminate the route information learnt from other ASs with the other **BGP routers** inside an AS.

BGP Routers: The routers within an AS that runs BGP and connected to other BGSP routers.

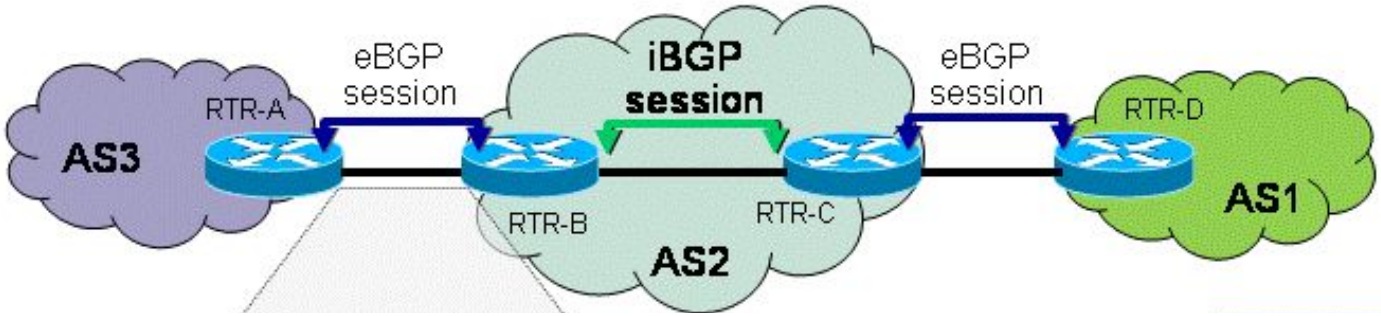
BGP Speakers

- All the routers within an AS (that are configured for BGP) are known as BGP speakers.
  - They speak BGP language which means they run BGP protocol.



- These BGP speakers need to be configured by an administrator with the ASN of the Autonomous System they belong to.
- If the AS is comprised of more than one IP subnet, all of the IP networks need to be associated with the ASN.
- If a BGP speaker in an AS has an IP address that is not associated with the ASN, then the speaker will not be able to participate in the AS.
- The BGP speakers on different ASs should be directly connected.
- But IGP is not run between the eBGP peers.

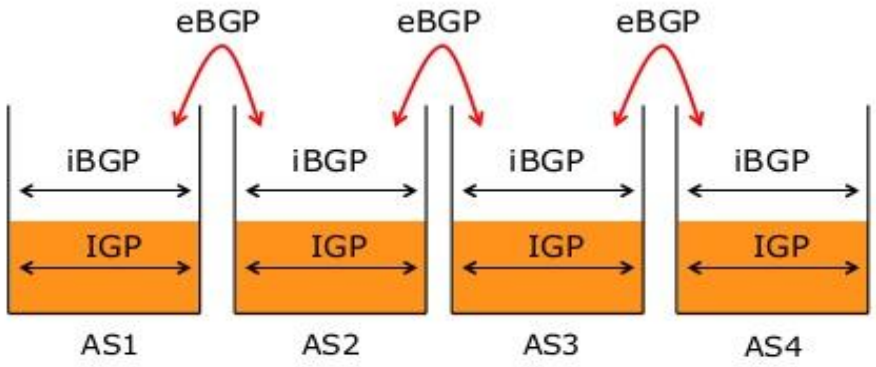
BGP: TCP Session

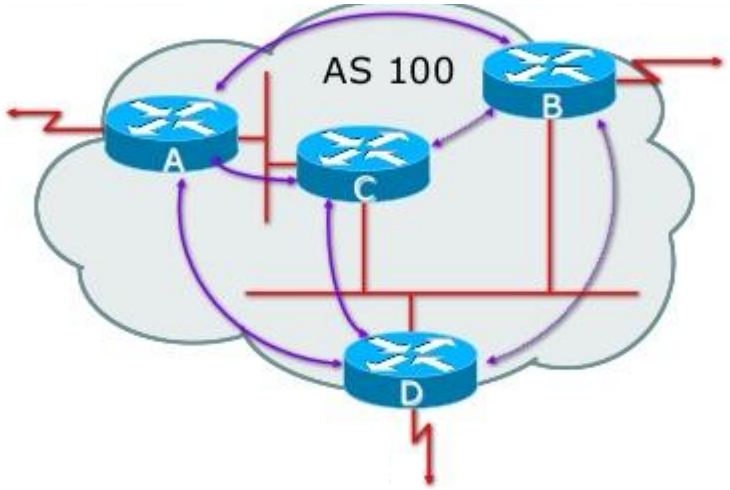


Note: Which router initiates TCP connection does not matter once a TCP session is established.

- BGP uses TCP as the transport protocol, on port 179.
- Two BGP routers form a TCP connection between one another.
- These **peer routers** or called as **BGP Speakers**.
- The peer routers exchange messages to open and confirm the TCP connection parameters, that include:
  - The **version** of BGP supported by the router
  - **Autonomous System Numbers** each **Router** belongs to
  - **Hold timer (**time it will wait for **"keep alive"** messages**) – 60 secs**
  - The router's **BGP identifier**
  - The **optional parameters** the router supports

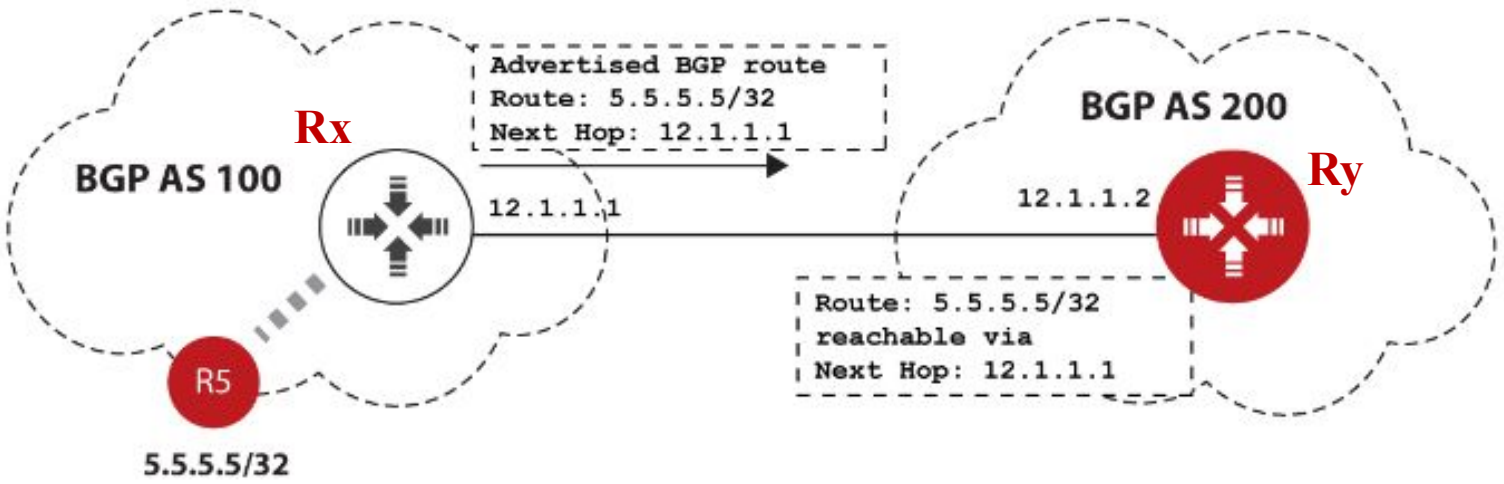
iBGP: Internal BGP





- iBGP runs between the peer routers within the same AS
- They are not required to be directly connected, but indirectly connected.
- IGP takes care of inter-BGP speaker connectivity
- iBGP speakers within an AS must be fully connected or meshed.
  - i.e., each iBGP speaker must peer with every other iBGP speaker in the AS
- They pass on IP prefixes (network IDs) learned from outside ASs
- They do not pass on prefixes learned from other iBGP speakers

Quiz 1: BGP Explained with an Example



Rx and **Ry** are called as BGP neighbours

R5 is another BGP Speaker getting added to AS100

- What is the protocol run between Rx and Ry? (iBGP or eBGP) **eBGP**
- How is Rx able to route packets to R5? **Through IGP (running within the AS)**
- How does R5 learn about the routes that Rx has learnt from Ry? **iBGP**
- What is being shared by Rx with Ry using eBGP? **An RT entry to be added into Ry for R5**

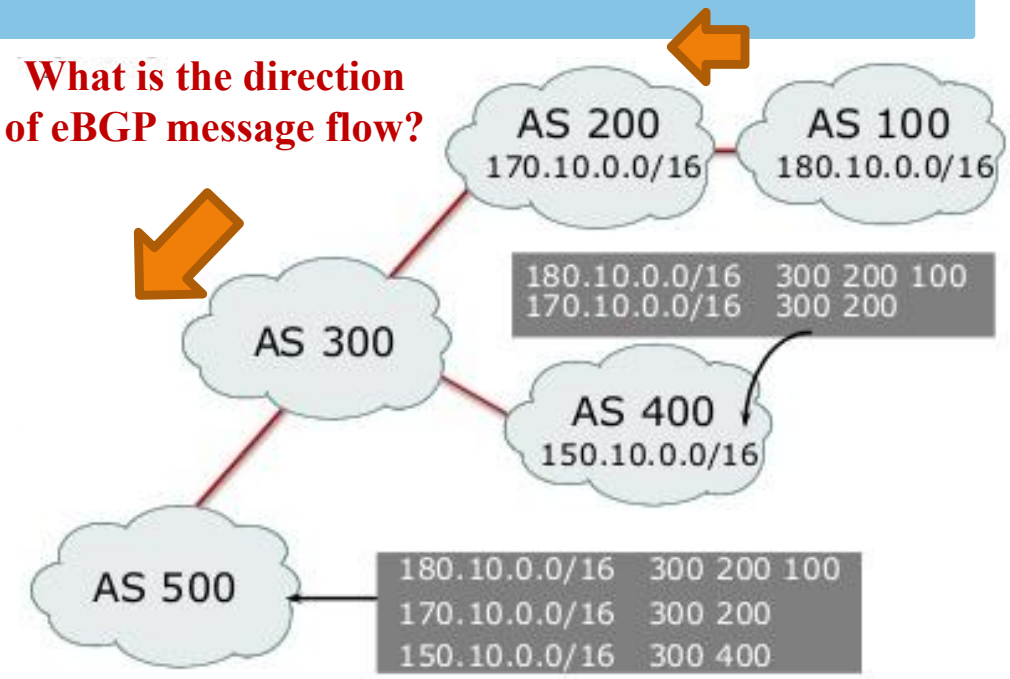
How does BGP work?

- BGP routers exchange network reachability information.
- This information is mainly an indication of the **full paths** that a route must take in order to reach the destination network.
- The **paths** are **AS numbers** given as a **list**.
- This information helps in the construction of a graph of ASs that are loop-free.
- BGP peers initially exchange the full BGP routing tables (for ASes).
- After this exchange, the peers send incremental updates as the routing table changes.
- BGP keeps a version number of BGP tables and they need to be the same on all the participating routers, to have a consistent view of the network.
- The version number changes whenever BGP updates the table with routing information changes.

Thus, **BGP** is called a **path-vector protocol**

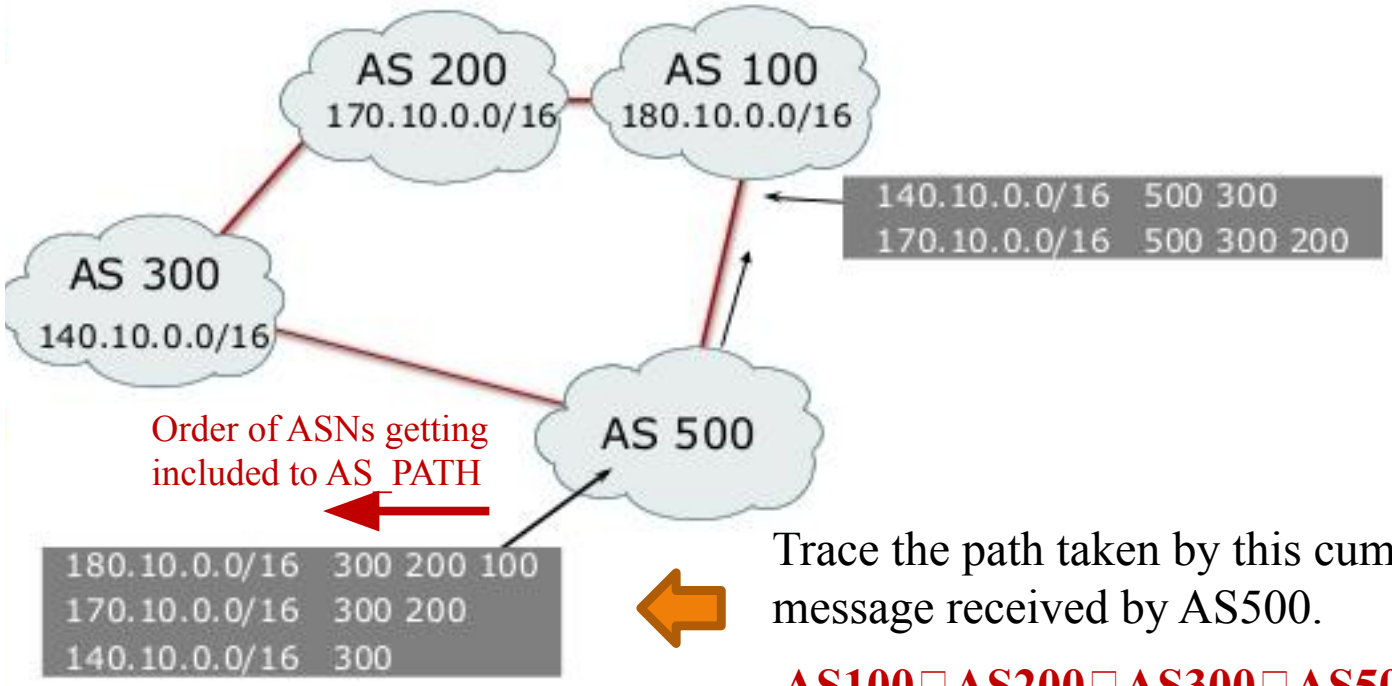
AS_PATH: An Attribute of eBGP

- It is a sequence of ASes a route has traversed.
- It has a mandatory **transitive** attribute.
- It is also useful for finding a loop and also to apply specific routing policies.



- When AS100 sends eBGP message the AS_PATH has an entry
  - 180.10.0.0/16 100 (that this network can be reached through AS100)
- When AS200 receives it appends it's own reachability info to it
  - 180.10.0.0/16 200 100 and 170.10.0.0/16 200
- When the message goes to AS300 and then when it finally reaches AS400 and AS500, the respective contents are show above.

AS_PATH Loop Detection



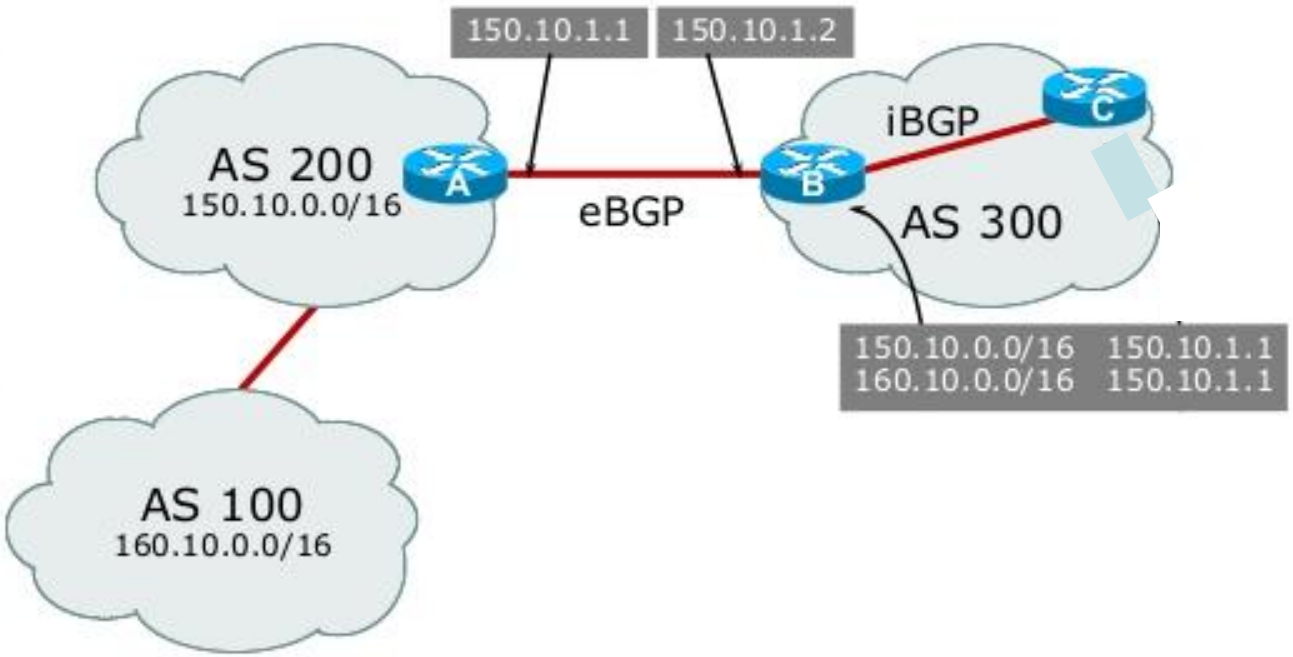
Trace the path taken by this cumulative eBGP message received by AS500.

AS100□AS200□AS300□AS500

- 180.10.0.0/16 entry coming from AS 500 is not accepted by AS100 because it will cause a loop since it has AS100 already in it.
- What is the direction a datagram to a host in AS100 from a host on AS500 would travel? **AS500□AS300□AS200□AS100**

Note: i.e., AS_PATH is always transitive.

Next Hop Entry

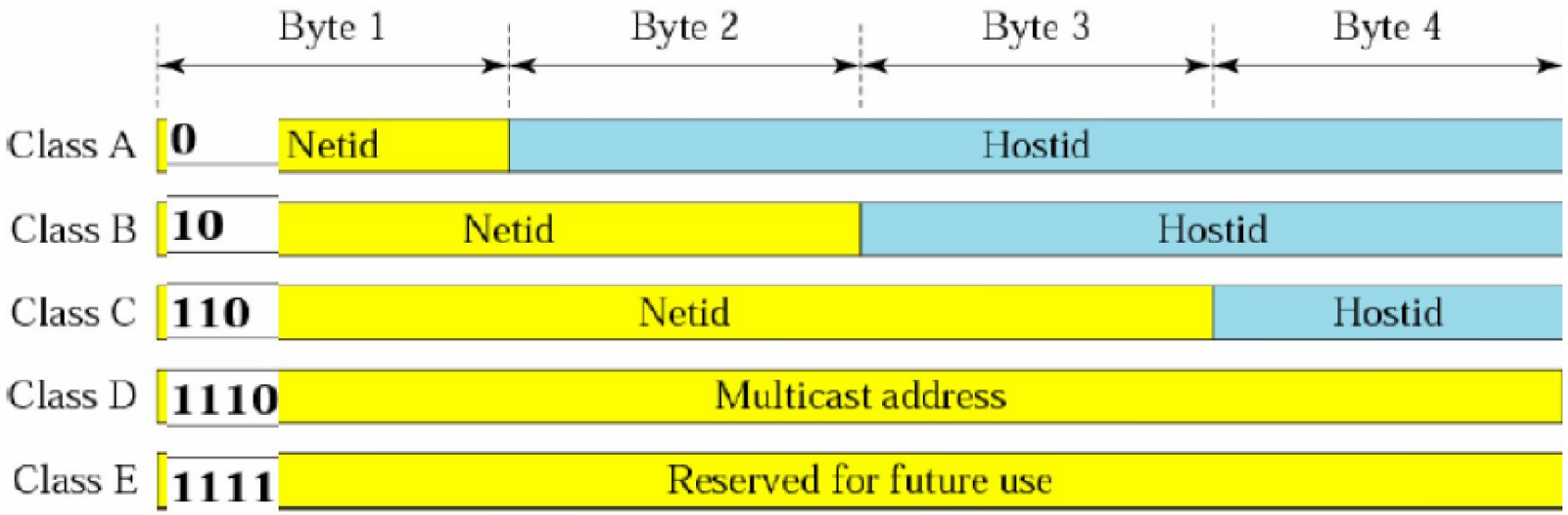


- The Router B's RT entries show that the next hop for the hosts on AS200 and AS100 are set to the directly connected interface of Router A (150.10.1.1)
- Router A takes care of forwarding the packet to AS100 by learning the route from other eBGP routers within AS200

Routing and Loop Avoidance: Explained

- When a router advertises a destination within its own AS to a neighbor in another AS, it puts its local ASN in the AS_PATH.
  - AS_PATH is one of the attributes in eBGP
- As the route is advertised to subsequent autonomous systems, each AS border router adds its own ASN to the attribute.
- The AS_PATH, then, becomes a list of ASNs that describes the path back to the destination.
- A router can choose a shortest path by choosing the route with the fewest ASNs listed in its AS_PATH.
- How can a router find if there is a loop within the AS_PATH?
- If a router sees its own ASN listed in the AS_PATH of a route advertised to it by a neighbor, it drops the route.
  - Because if the router accepts the route and adds itself to it, it's ASN will be twice in the path causing a loop.

IP Multicast Addresses



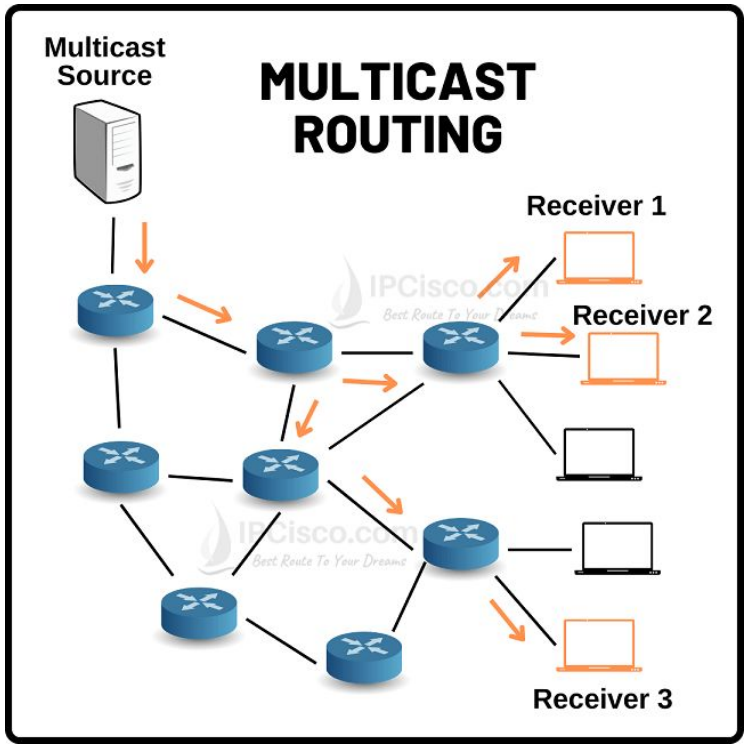
- IP multicast addresses, also known as Class D addresses, fall within the range of **224.0.0.0** to **239.255.255.255**

Note: 224 □ 0xE0, 239 □ EF

- These addresses are used for sending data to a group of recipients simultaneously, rather than individually, which is efficient for **one-to-many** communication.
- The source IP address of a multicast packet is always a unicast address, while the destination is a multicast address

Multicast Routing: Need for it

- Multicast is used wherever any media is to be shared with a large number of subscribers who have registered for it.
  - Mostly in media or live telecast, etc.
- This involves media to be streamed to many customers simultaneously
- It is highly suboptimal and waste of network bandwidth if individual customers are streamed the same media from the server.



- IP Multicast addresses are used for this purpose, network bandwidth can be preserved as much as possible based on the physical location of the customers.
- Need to have a mechanism for registration for a multicast stream and updating the routing tables in the routers accordingly. **Solution: IGMP**

Some Popular IP Multicast Addresses - Info

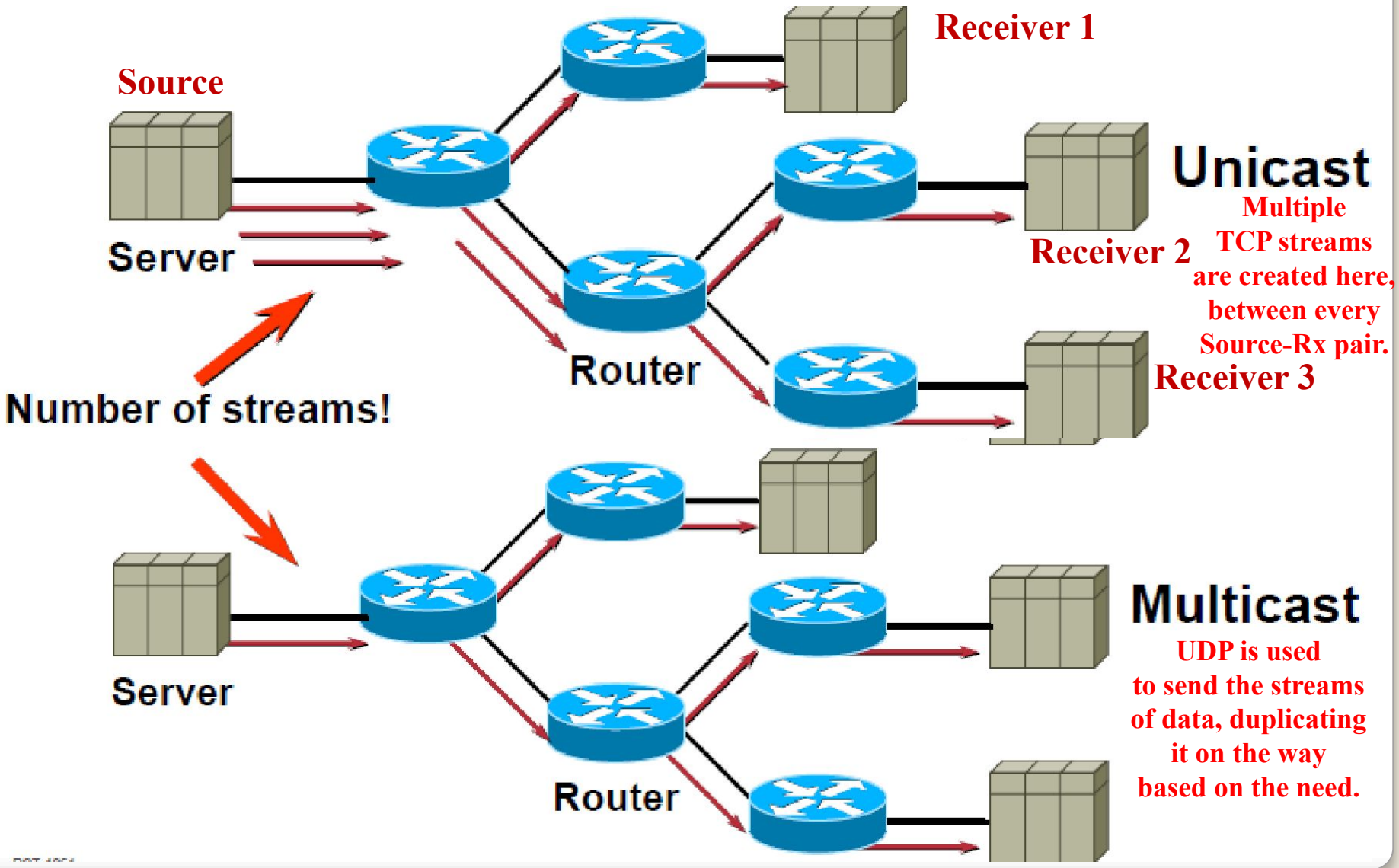
Multicast Address	Use Case
233.89.188.1 – 233.89.188.255	BBC Multicast Streaming
233.56.12.1 – 233.56.12.255	CNN, Fox News, Bloomberg TV
239.1.1.1 – 239.1.1.255	IPTV and media streaming
239.255.1.1 – 239.255.1.255	Content Delivery Networks (CDN)

- The 239.x.x.x range (Administratively Scoped Multicast) is often used for **private multicast streaming**.

Note: 233 □ 0xE9 239 □ 0xEF

How does IP Multicast work?

IP Unicast Vs Multicast

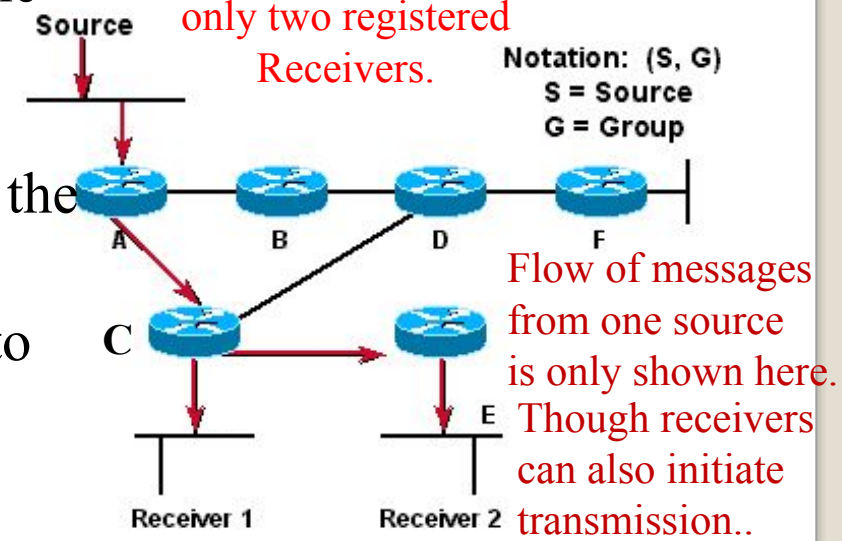


IP Multicast Message Delivery

- Normal IP communication, in which each packet must be addressed and sent to a single host, is not well suited to multicast applications.
- If an application has data to send to a group, it would have to send a separate packet with the identical data to each member of the group.
- This would consume more bandwidth than necessary.
- Furthermore, the redundant traffic (multiple copies of the same data) is not distributed evenly but rather is focused around the sending host.
- And may easily exceed the capacity of the sending host and the nearby networks and routers.
- IP provides an IP-level multicast analogous to the link-level multicast provided by multi-access networks like Ethernet.
- Using IP multicast, instead of sending identical packets to each member of the group, a host sends a single copy of the packet addressed to the group's multicast address.

IP Multicast Message Flow: An Example

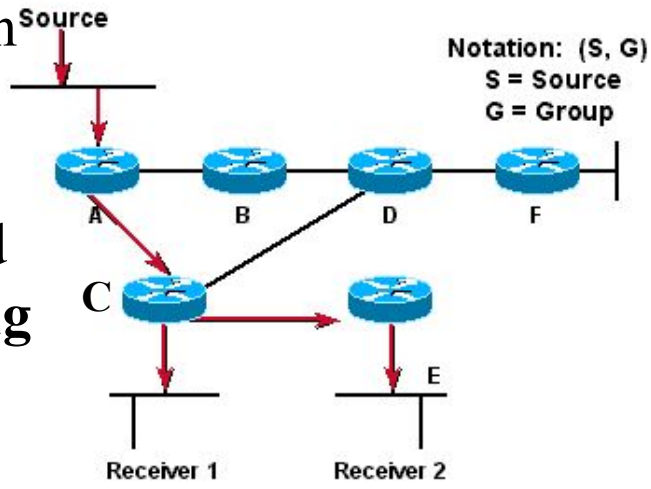
- The sending host doesn't need to know the individual unicast IP address of each member of the group because, as we will see, that knowledge is distributed among the routers in the internetwork.
- Similarly, the sending host doesn't need to send multiple copies of the packet.



- Because, the routers will make copies whenever they have to forward the packet over more than one link, based on the presence of receivers.
- Compared to using unicast IP to deliver the same packets to many receivers, IP multicast is more scalable.
- Because it eliminates the redundant traffic (packets) that would have been sent many times over the same links, especially those near to the sending host.

Many-to-many One-to-many

- IP's original many-to-many multicast has been supplemented with support for a form of **one-to-many multicast**.
- In this model of one-to-many multicast, called **Source-Specific Multicast** (**SSM**), a **receiving host** specifies both a **multicast group** and a specific **sending host**.



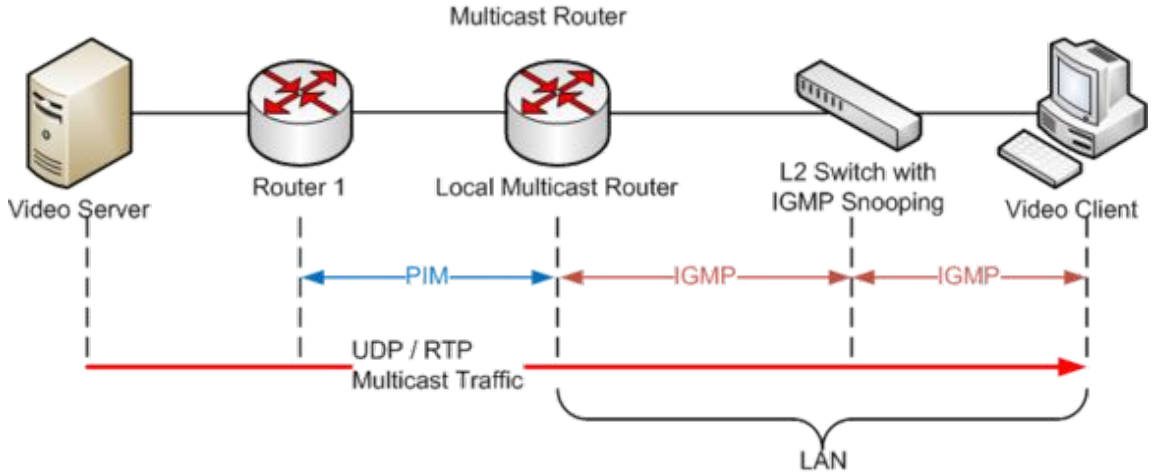
- The receiving host would then receive multicasts addressed to the specified group, but only if they are from the specified sender.
- Many Internet multicast applications (e.g., radio broadcasts) fit the SSM model.
- To contrast it with SSM, IP's original many-to-many model is sometimes referred to as **Any Source Multicast** (**ASM**).

Multicast Routing

DVMRP

Distance Vector Multicast Routing Protocol (**DVMRP**)
is a multicast routing protocol that enables
efficient multicast data distribution in an IPv4 network

IP Multicast: Various Protocols Involved



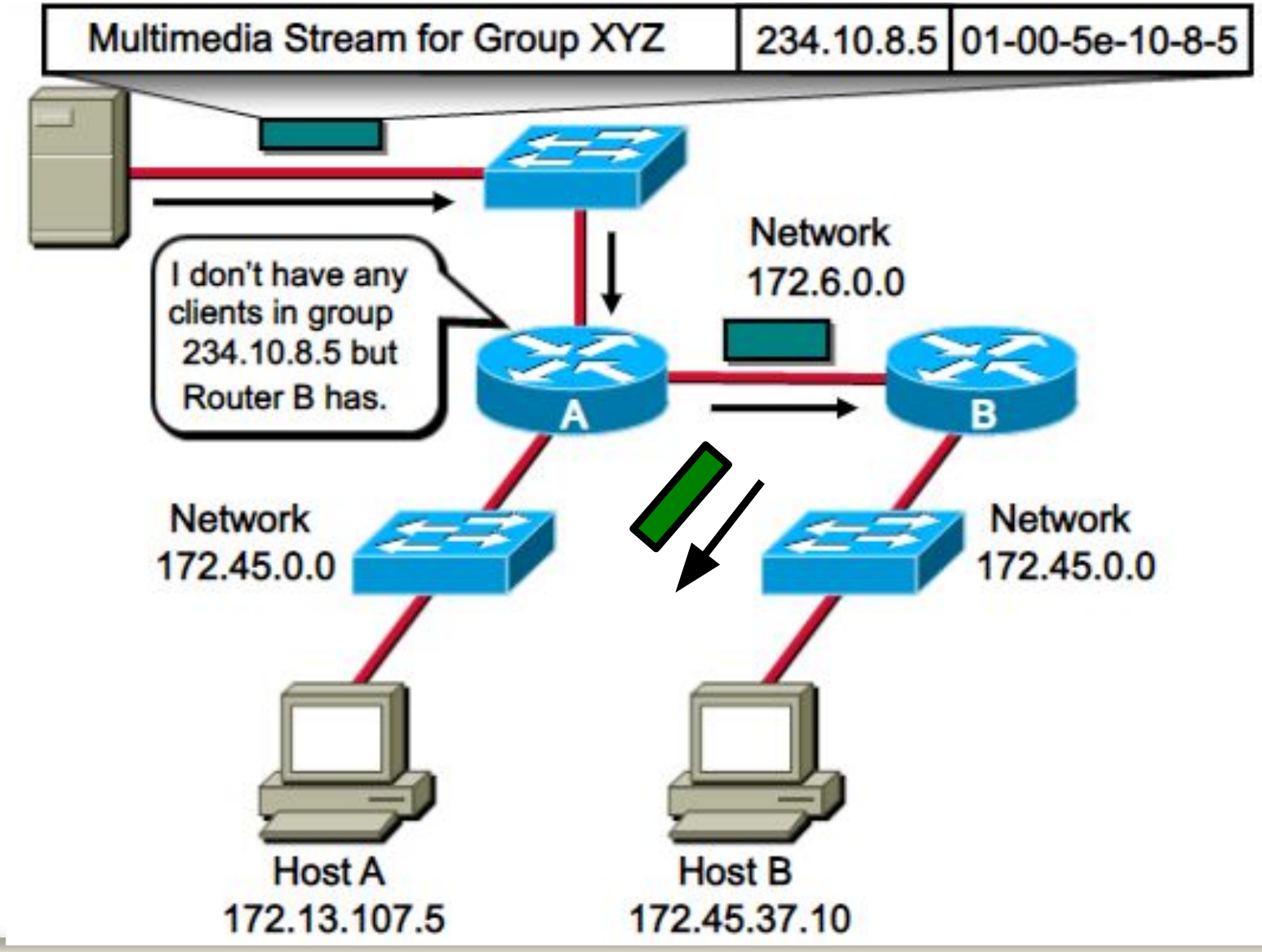
1. The actual video data are sent as **UDP** packets.

2. RTP (Real-time Protocol) is used to enable prioritized delivery of UDP packets by the routers enroute from the sender to the receivers.

- IGMP operates between a host and a local multicast router.
- IGMP operates on the network layer, just the same as other network management protocols like ICMP.
- Switches featuring IGMP snooping derive useful information by observing these IGMP transactions.
- Protocol Independent Multicast (**PIM**) routing protocol is then used between the local and remote multicast routers, to direct multicast traffic from hosts sending multicasts to hosts that have registered through IGMP to receive them.

DVMGRP: Distance vector based IP multicast routing Protocol also used apart from PIM.

IP Multicast Routing: Example



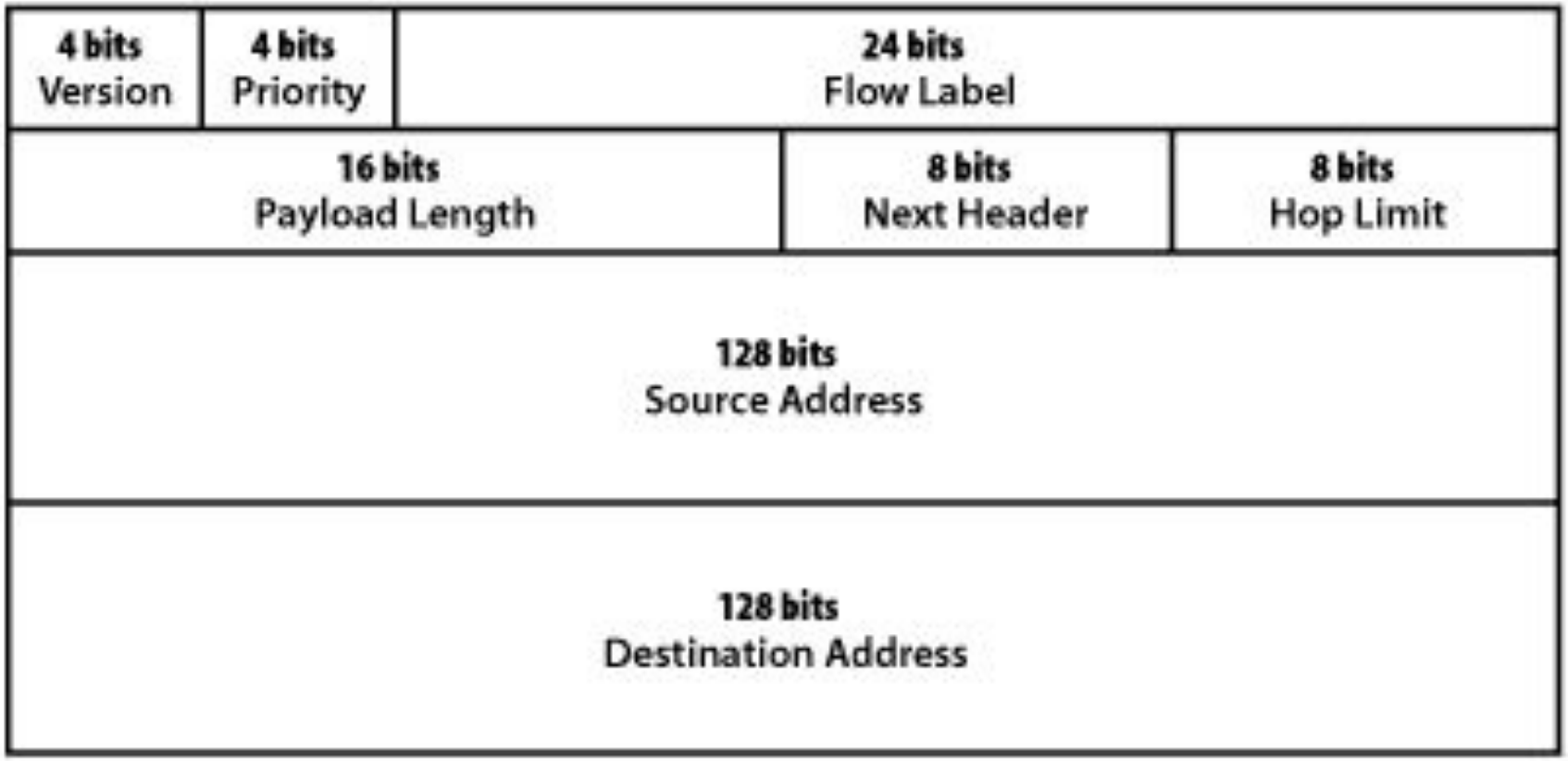
IPv6 or IPng

IPng: IP Next Generation

IPv6 Motivation: Design Goals

- **Address Space Expansion**: IPv6 provides a 128-bit address space (300 trillion trillion trillion unique addresses).
- **Simplified Header Format**: More efficient, improving routing performance and making the packet processing faster.
- **Built-in Security**: Unlike IPv4, IPSec is a mandatory feature in IPv6, and integrated into it, enabling end-to-end encryption and authentication at the network layer. – IPSec will be covered soon.
- **Improved Support for Mobility and Multicast**: IPv6 offers native support for mobile IP and more efficient multicast and **anycast** communication. No support for broadcast.
- **Auto-configuration** (Stateless Address Autoconfiguration - SLAAC): Devices can automatically generate their own IP addresses, simplifying network configuration without relying on DHCP.
  - **Router Advertisement (RA)** message from gateway gives its network prefix (e.g., 2001:db8:abcd:1234::/64) and auto-configure their address using SLAAC, device uses this prefix and appends its interface identifier (MAC) having an unique IPv6 address for itself.

IPv6 Header


4 bits
Version
4 bits
Priority
24 bits
Flow Label
16 bits
Payload Length
8 bits
Next Header
8 bits
Hop Limit
128 bits
Source Address
128 bits
Destination Address

IPv6 Header Fields

QoS: Quality of Service

Note: The **network prefix** is inferred from:
Routing tables, Interface Configurations,
Router Advertisements (RAs)
It is not part of the IPv6 header.
Though **/64** is common **NW prefix** it can be different.

- **Version**: Indicates the IP version; always set to **6** for **IPv6**.
- **Traffic Class**: Used for **prioritizing packets** and specifying **QoS**.
- **Flow Label**: Identifies **flows of packets** for special handling like real-time services.
- **Payload Length**: Specifies the **length of the data** following the header (in bytes).
- **Next Header**: Indicates the **type of the next header**, such as TCP, UDP, or extension headers.
- **Hop Limit**: Limits packet lifetime by **decrementing at each hop**; replaces IPv4's TTL.
- **Source Address**: The **IPv6 address of the sender**.
- **Destination Address**: The **IPv6 address of the receiver**.

IPv6 Addressing Conventions

- IPv6 addresses are 128 bits long, represented in **8 groups of 16-bit hexadecimal blocks**, separated by colons (:).
  - **Example: 2001:0db8:0000:0000:0000:ff00:0042:8329**
- **Zero compression**: A single contiguous sequence of all-zero blocks can be replaced with :: **only once** in the address.
  - **Example: 2001:0db8::ff00:0042:8329**

Only once, to avoid ambiguity on the number of zeros in it.

- Leading zeros in each block can be omitted.
  - **Example: 2001:db8::ff00:42:8329**
- IPv6 uses CIDR-style prefix length to denote subnetting, written as /n
  - **Example**: **2001:db8::223:45/64** the **first 64 bits are the network prefix**.
  - **Network ID: 2001:db8::/64**
  - **Host ID: ::223:45 (or 0000:0000:0223:0045 in full)**
- **Multicast ff00::/8**
- **Loopback ::1**

Note: Only one loopback address in IPv6, not a block of loopback addresses as in IPv4

IPv6 Addressing Scheme

- **Global Unicast**: Publicly routable addresses on the Internet, **e.g., 2001:db8::34:13** **Note**: Need to **start** with **binary 001** or in the **2000::/3 range**
- **Link-Local Unicast**: Used for communication within the same link (not globally routed), **e.g., fe80::45:21**
- **Unique Local Address** (ULA): Private addressing within an organization, not routed on the Internet, **e.g., fd00::abcd**
- **Multicast**: One-to-many communication, sent to all subscribed interfaces, **e.g., ff02::34** (all nodes on local link).
- **Anycast**: Same **public IP address** is assigned to multiple devices; delivered to the nearest one, **e.g., a DNS resolver with 2001:db8::53**
- **Loopback**: Refers to the local device, used for testing, **e.g., ::1**
- **Unspecified Address**: Used when an address is not yet assigned (e.g., during DHCPv6), **e.g., ::**

Note: NAT is not required in IPv6, though private IP addresses are used for local use.
Note: Anycast is used for specific services like DNS, for CDN edge servers, etc.

IPv6: Salient Features

- **Larger address space:**
  - Global reachability and flexibility
  - Aggregation
  - Multihoming
  - Autoconfiguration
  - Plug-and-play
  - End-to-end without NAT
  - Renumbering
- **Mobility and security:**
  - Mobile IP RFC-compliant
  - IPsec mandatory (or native) for IPv6
- **Simpler header:**
  - Routing efficiency
  - Performance and forwarding rate scalability
  - No broadcasts
  - No checksums
  - Extension headers
  - Flow labels
- **Transition richness:**
  - Dual stack
  - 6to4 and manual tunnels
  - Translation

Note: No **IP header checksum**, because of **mandatory support for IPSec** in **IPv6**