Unicast Routing Protocol

RIP, OSPF, BGP

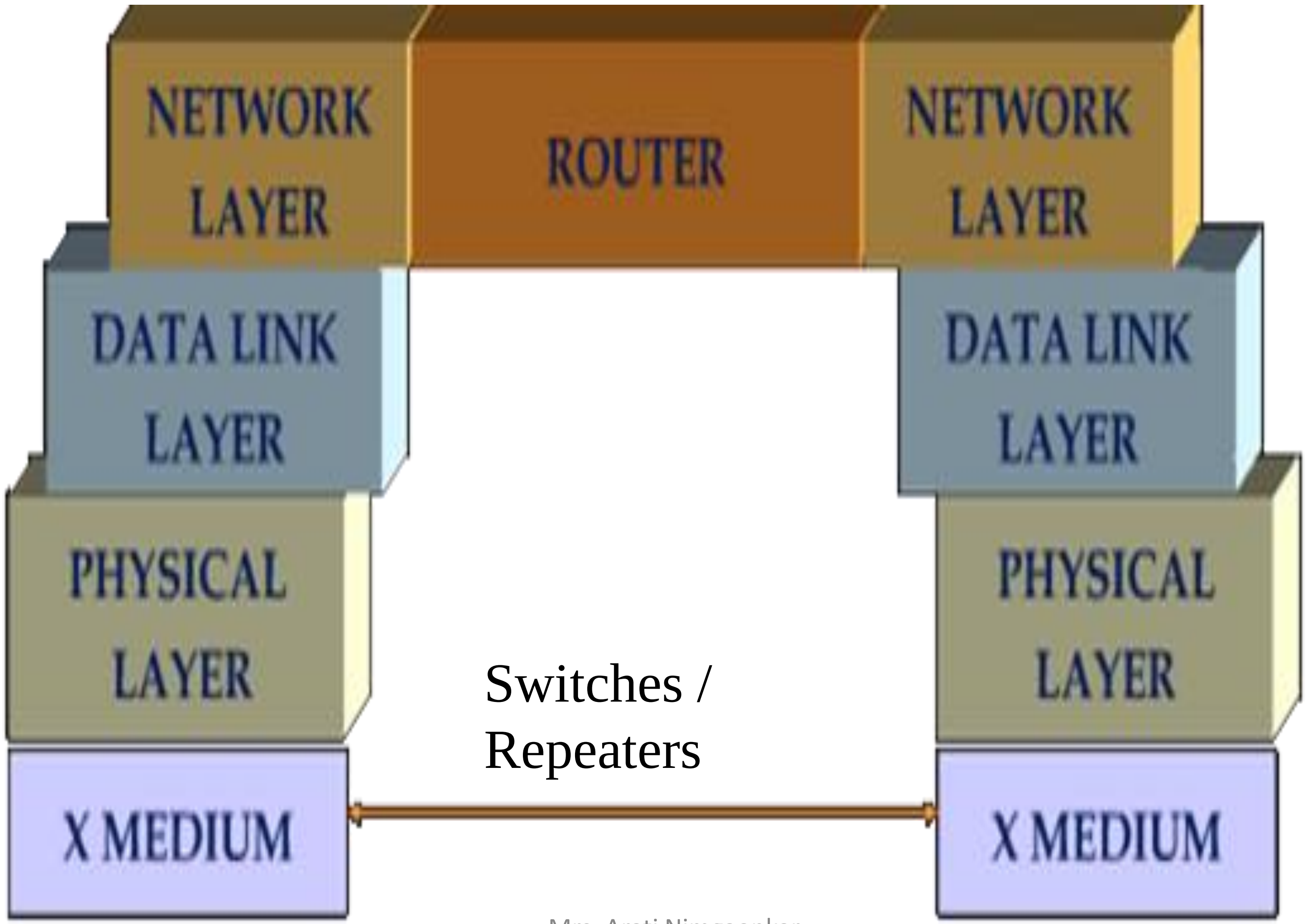
Introduction……….

- Unicast communication means communication between one sender and one receiver, a one-to-one communication.
- Here, will discuss how routers create their routing tables to support unicast communication.
- Internet is divided into administrative areas known as autonomous systems to efficiently handle the exchange of routing information.
- Two dominant routing protocols are used inside an autonomous system and one routing protocol is used for exchange of routing information between autonomous systems.

Objectives………..

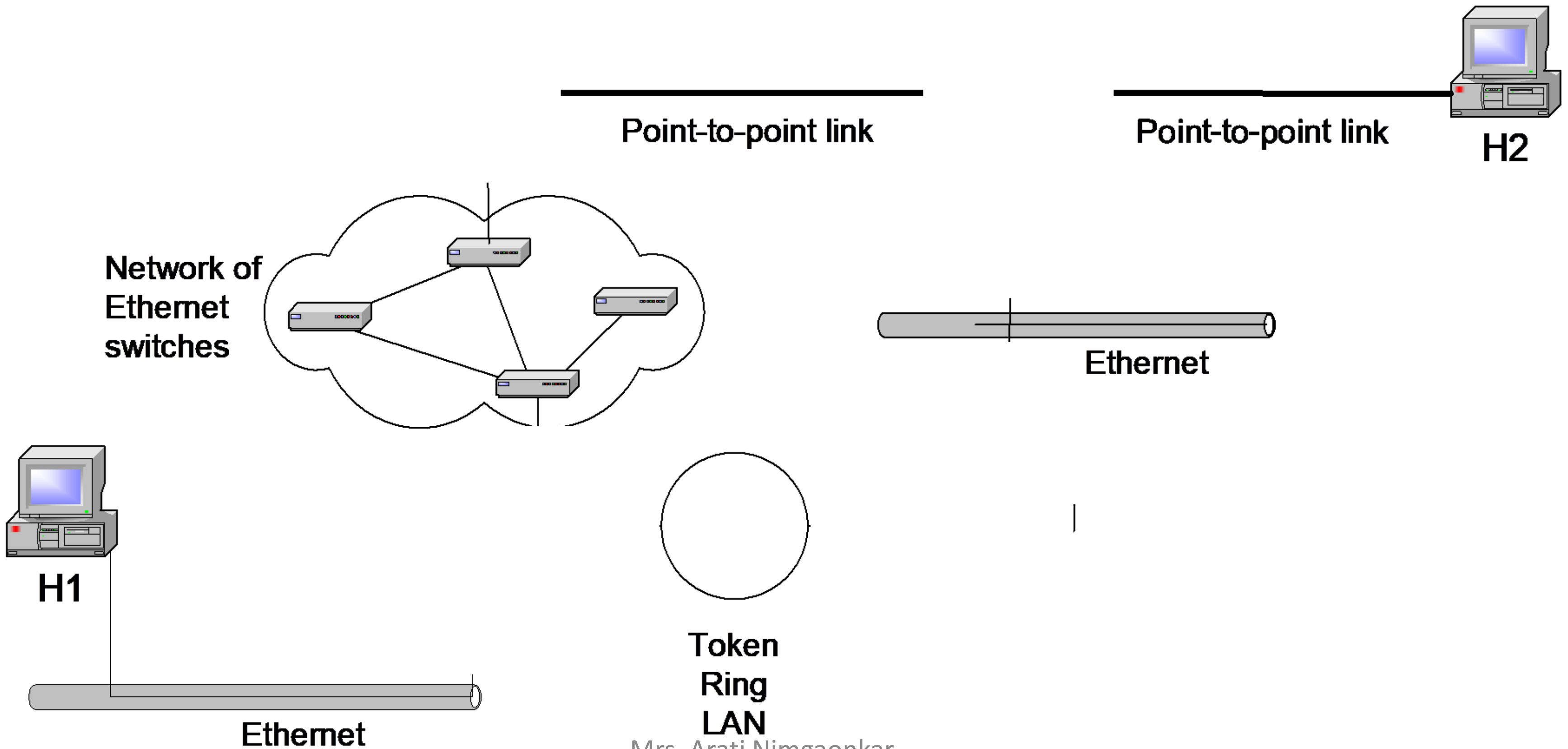
- To discuss structure of router and routing table
- To discuss the concept of intra and inter domain routing.
- To introduce the idea of autonomous systems (As)
- To discuss different routing protocols,
  - **Routing Information Protocol (RIP)** which implements the idea of distance vector routing in the Internet.
  - **Open Shortest Path First (OSPF)** which implements the idea of link state routing in the Internet.
  - **Border Gateway Protocol (BGP)** which implements the idea of path vector routing in the Internet.

Prerequisite ……….



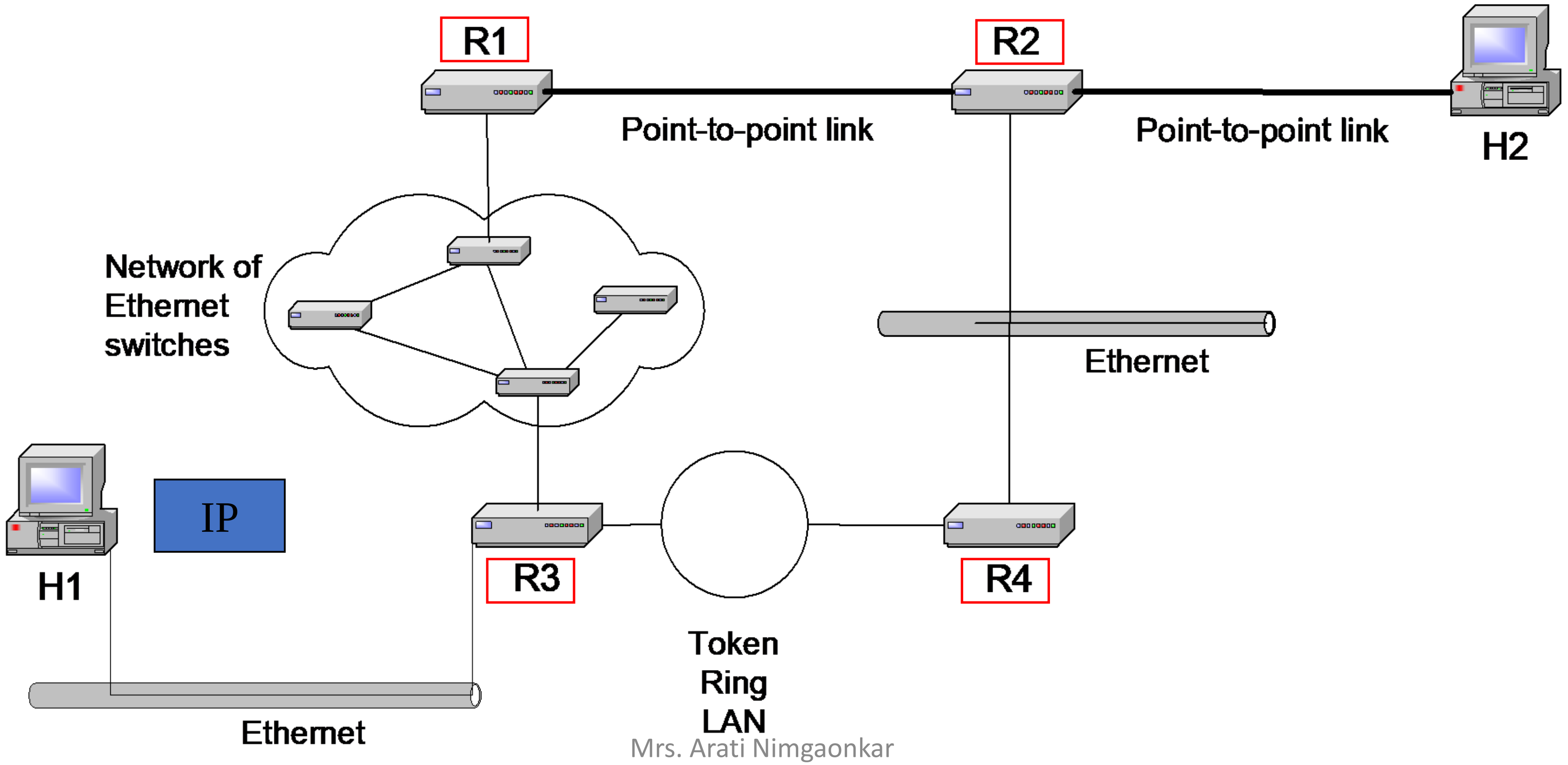
View at the DLL- Connection-oriented

- Internetwork is a collection of LANs, point-to-point links , switched networks that are connected by routers



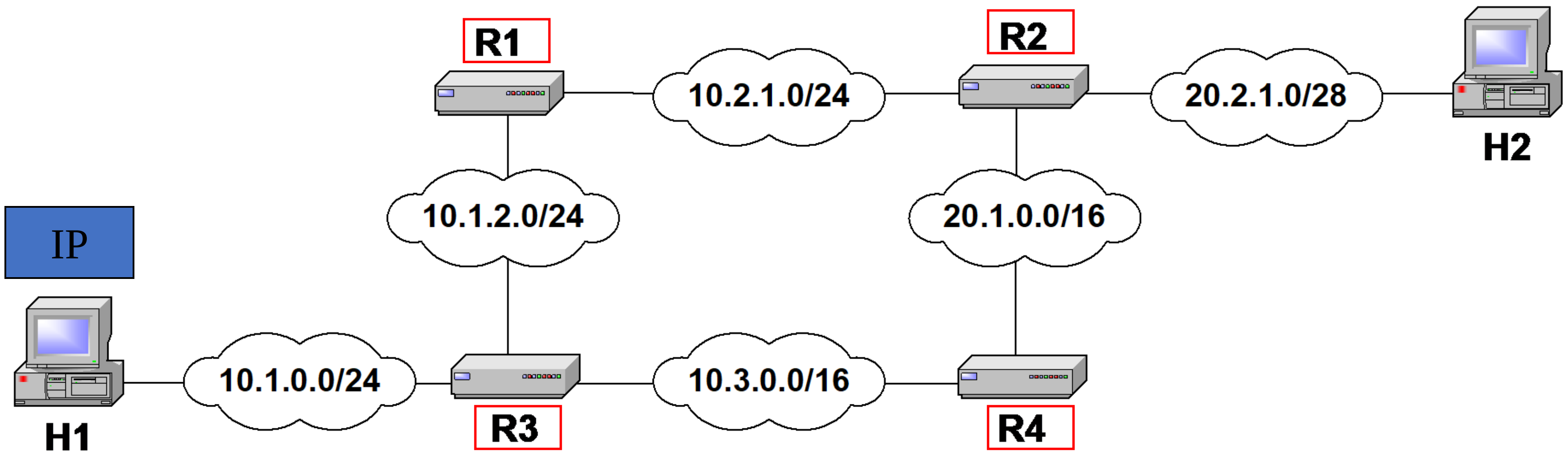
View at the DLL- Connection-oriented

- Internetwork is a collection of LANs, point-to-point links , switched networks that are connected by routers

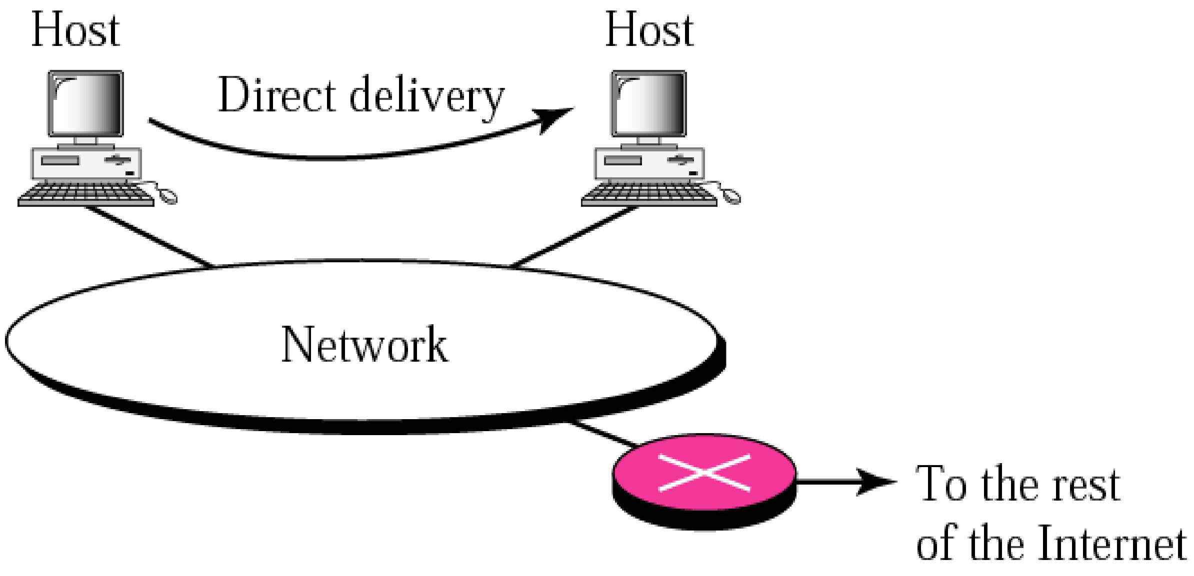


View at the IP layer- Connectionless

- An IP network is a logical entity with a network number
- We represent an IP network as a “cloud” .
- The IP delivery service takes the view of clouds, and ignores the data link layer view



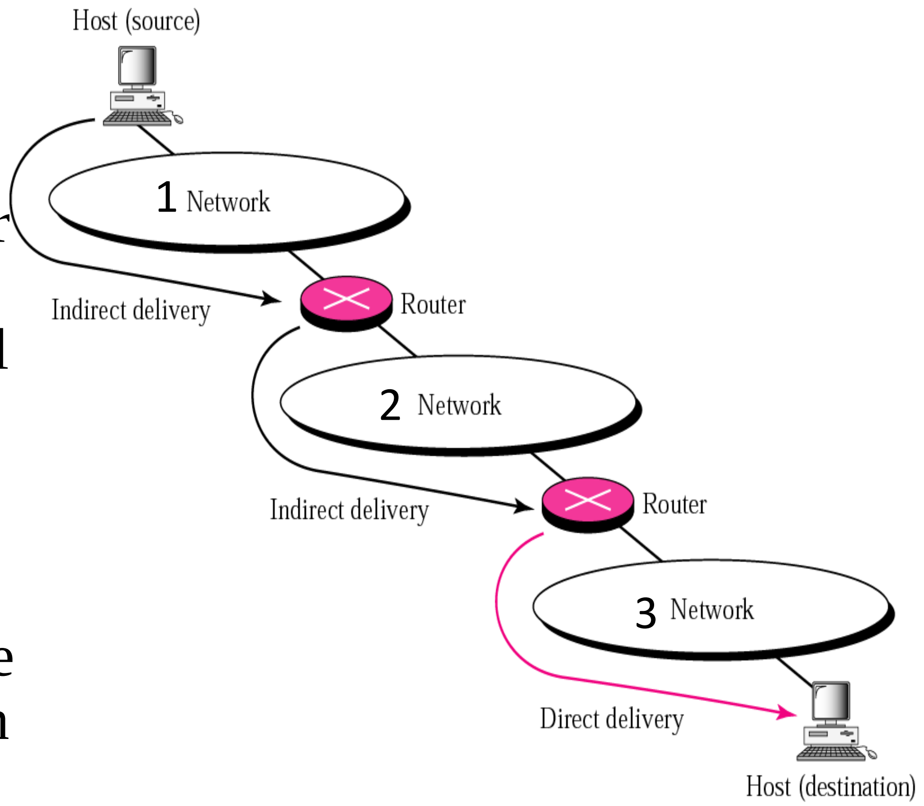
Delivery of IP datagram – Direct Vs Indirect



- Direct delivery occurs when source and destination of the packet are located on the same physical network.

Delivery of IP datagram – Direct Vs Indirect

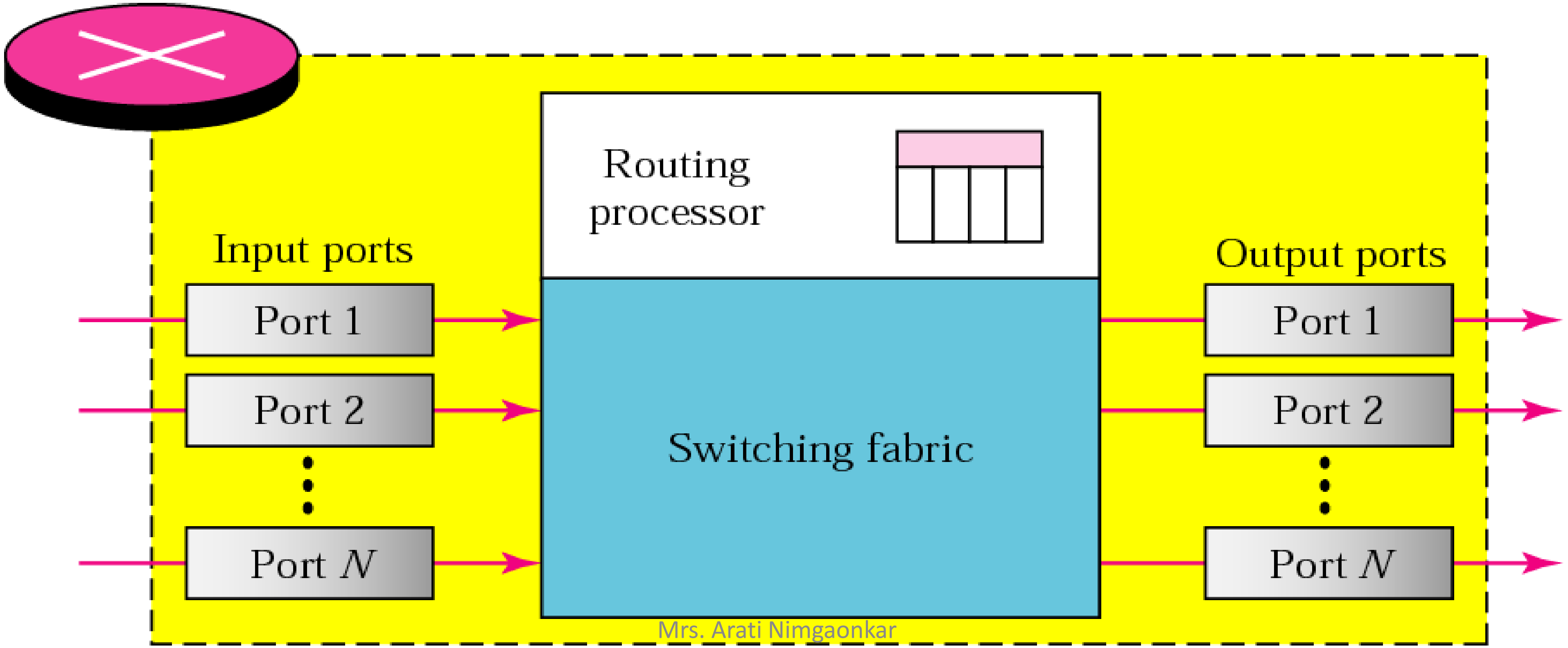
- If the destination host is not on the network as the deliverer, packet is delivered indirectly.
- The packet goes from router to router until it reaches the one connected to the same physical network as its final destination.
- The sender uses the destination IP address and a routing table to find the IP address of the next router to which packet should be delivered.



Routing – Structure of a router

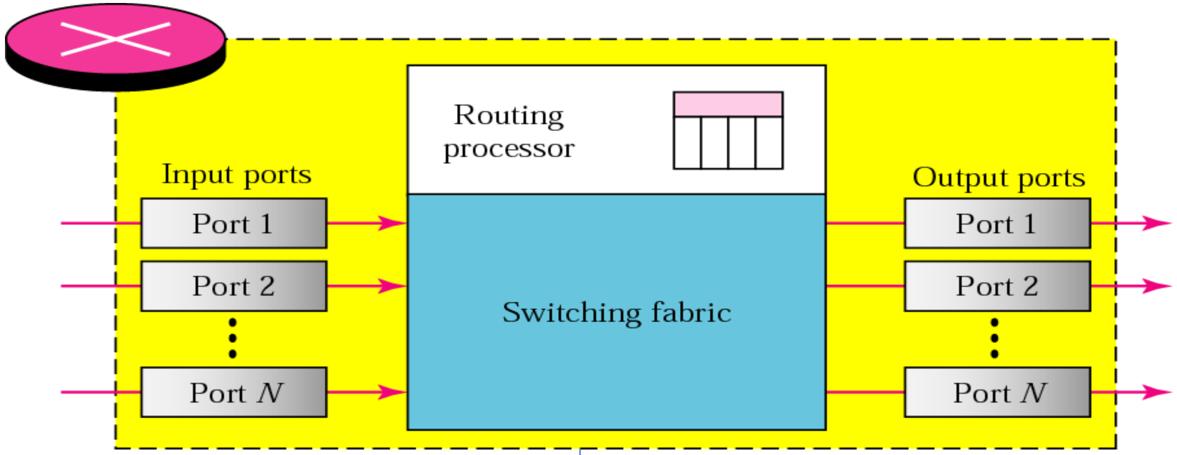
Four components –

- Input ports
- Output ports
- Routing Processor
- Switching fabric



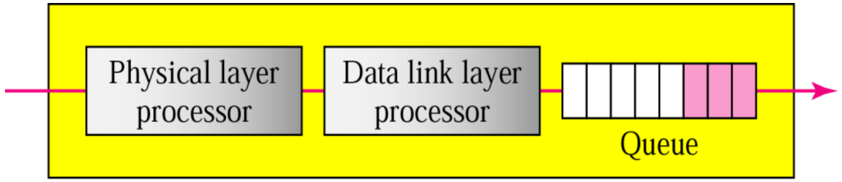
Routing – Structure of a router

Input–Output port



Input Port

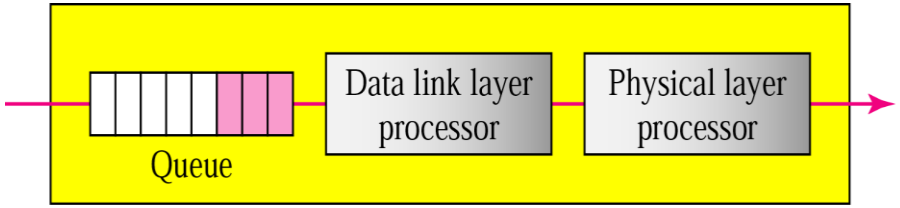
- Constructs bits from the received signal.
- Decapsulates packet from frame.
- Holds packets in a queue before directed to the switching fabric.



Input port

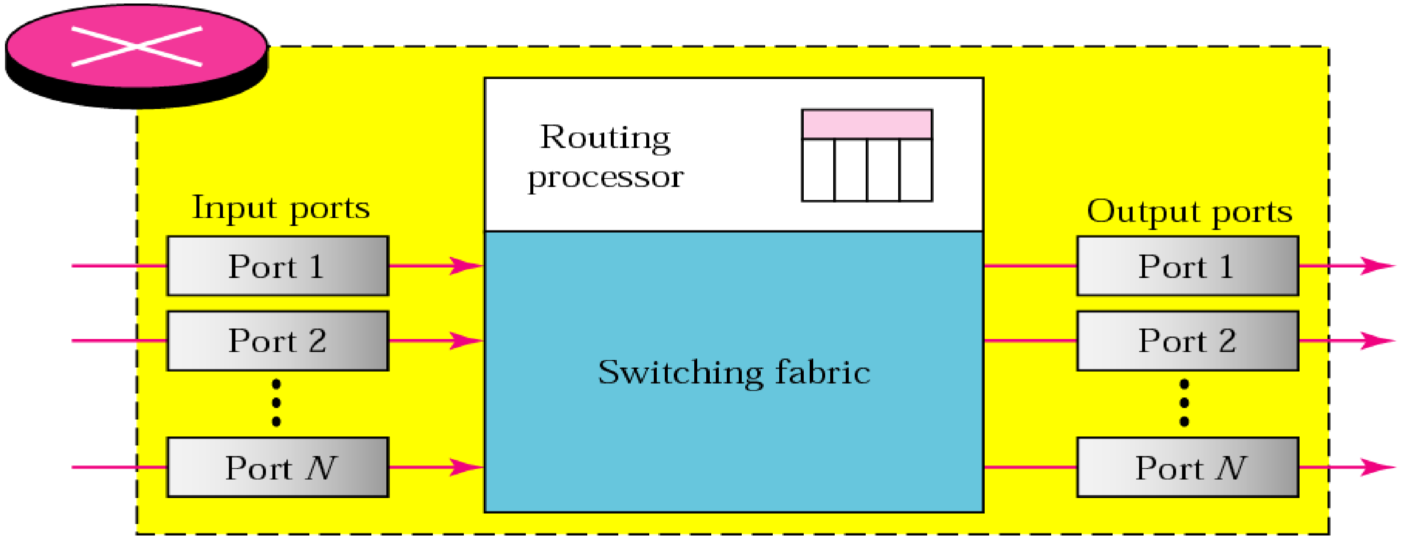
Output Port

- Outgoing packets are queued.
- Packet is encapsulated in a frame.
- Performs physical & DLL functions - Constructs signal from bits.



Output port

Routing – Structure of a router – Routing Processor & Switching fabric

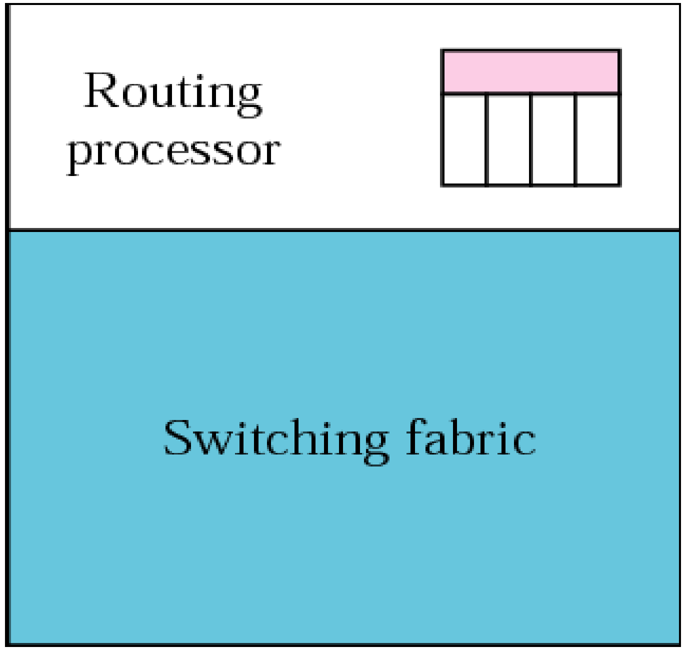


Routing Processor

- Performs functions of the network layer.
- Does table look up function.

Switching fabric

- Combination of hardware and software that moves data coming in to a network node out by the correct port to the next node in the network.



Delivery of IP datagram

There are two distinct processes for delivering IP datagram:

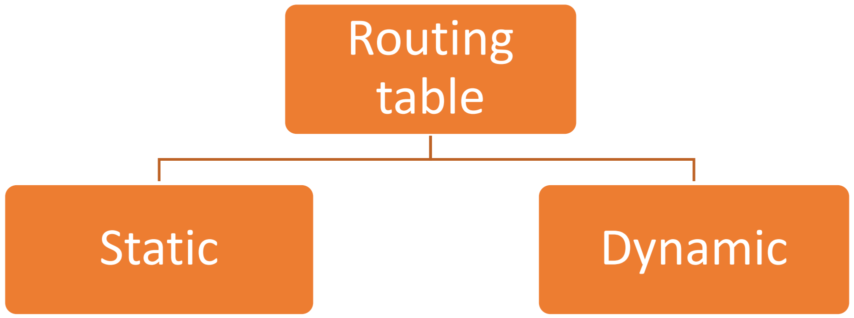
1. **Routing:**

   How to find and setup the **routing tables**?

2. **Forwarding:**

   How to pass a packet from an input interface to the output interface?

Routing



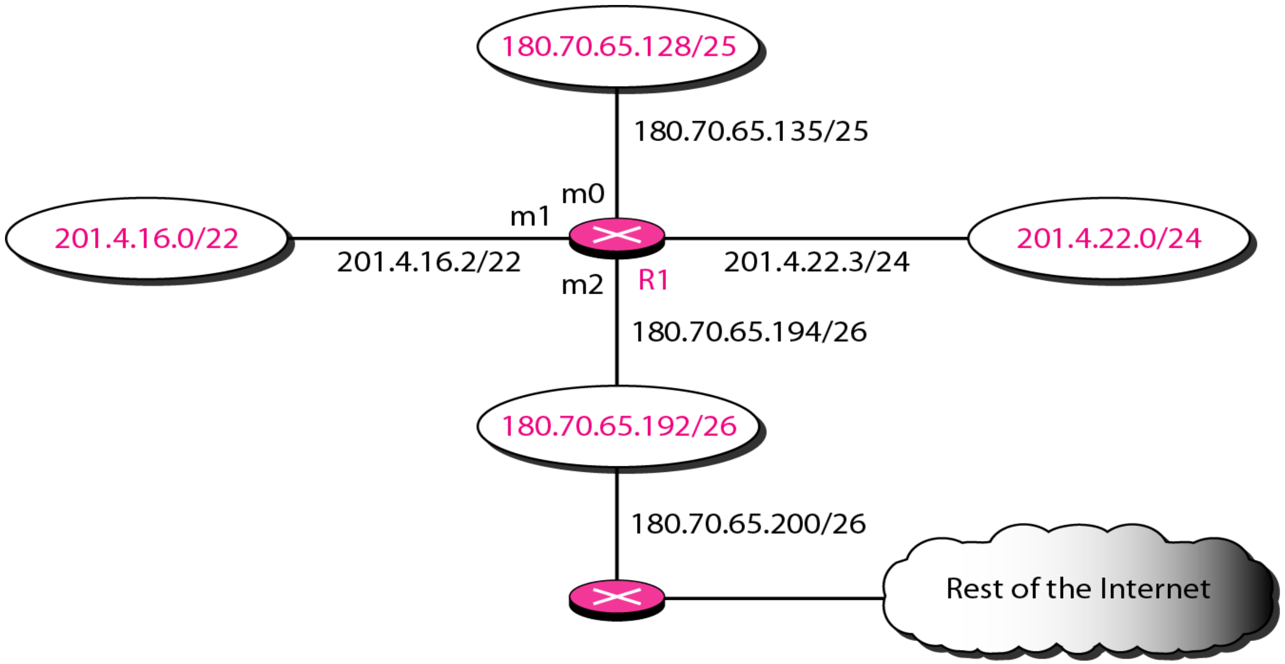
- Routing deals with the issues of creating and maintaining routing tables.
- Router maintains a **<u>Routing table</u>** with an entry for each destination or a combination of destinations, to route IP datagram.
- A static table is one with **manual entries**.
- A dynamic table is one that is **updated automatically** when there is a change somewhere in the Internet.
- A routing protocol is a combination of rules and procedures that lets routers in the Internet inform each other of changes.

Routing - Routing table

- ***Network Mask:*** Subnet mask that is used to match a destination IP address to the network ID.
- ***Network address:*** Destination network ID to the route.
- ***Next Hop:*** The IP address of the next hop (gateway address).
- ***Interface:*** The outgoing network interface the device should use when forwarding the packet to the next hop or final destination.
- ***Metric:*** A number used to indicate the cost of the route so the best route among possible multiple routes to the same destination can be selected. A common use of the metric is to indicate the **number of hops** (routers crossed) to the network ID.
- **Flag :**
  - U (Up) : Router is up and running.
  - G (gateway) : Destination is in another n/w.
  - H : Host specific address.
  - D : entry is added by redirection message.
  - M : Modified by redirection.

Routing - Routing table……example

Make a routing table for router R1…………….



Mask	*Network Address*	*Next Hop*	*Interface*
/26	180.70.65.192	—	m2
/25	180.70.65.128	—	m0
/24	201.4.22.0	—	m3
/22	201.4.16.0	....	m1
Any	Any	180.70.65.200	m2

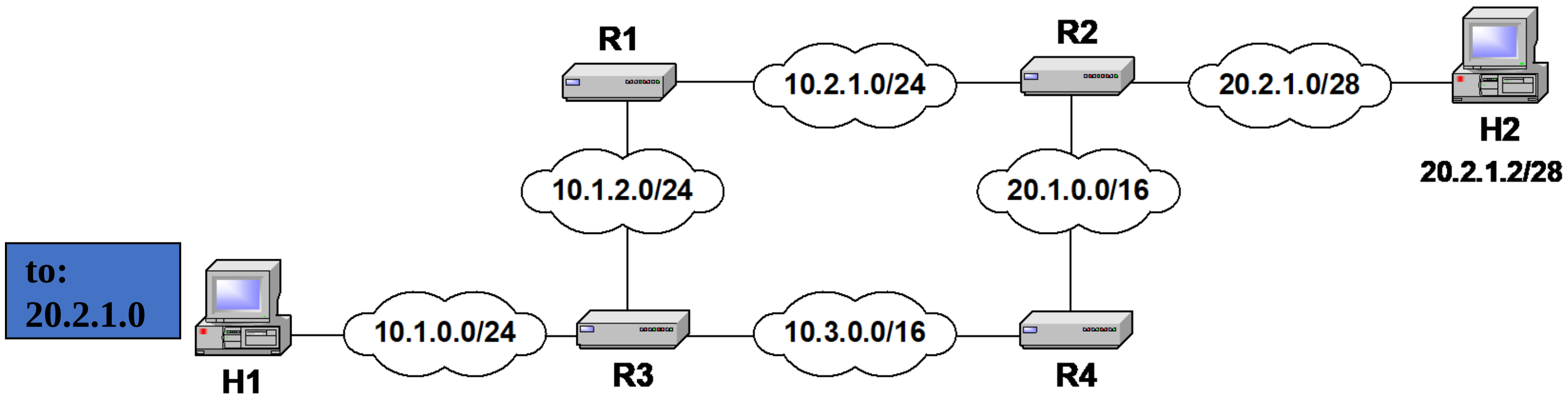
Forwarding –

- Forwarding means to place the packet in its route to its destination. Forwarding requires a host or a router to have a routing table.
- When a host has a packet to send or when a router has received a packet to be forwarded, it looks at this table to find the route to the final destination.

Destination	Next Hop
10.1.0.0/24	R3
10.1.2.0/24	direct
10.2.1.0/24	direct
10.3.1.0/24	R3
20.2.0.0/16	R2
30.1.1.0/28	R2

Destination	Next Hop
10.1.0.0/24	R1
10.1.2.0/24	R1
10.2.1.0/24	direct
10.3.1.0/24	R4
20.1.0.0/16	direct
20.2.1.0/28	direct

Destination	Next Hop
10.1.0.0/24	R2
10.1.2.0/24	R2
10.2.1.0/24	R2
10.3.1.0/24	R2
20.1.0.0/16	R2
20.2.1.0/28	direct



Destination	Next Hop
10.1.0.0/24	direct
10.1.2.0/24	R3
10.2.1.0/24	R3
10.3.1.0/24	R3
20.1.0.0/16	R3
20.2.1.0/28	R3

Destination	Next Hop
10.1.0.0/24	direct
10.1.2.0/24	direct
10.2.1.0/24	R4
10.3.1.0/24	direct
20.1.0.0/16	R4
20.2.1.0/28	R4

Destination	Next Hop
10.1.0.0/24	R3
10.1.2.0/24	R3
10.2.1.0/24	R2
10.3.1.0/24	direct
20.1.0.0/16	direct
20.2.1.0/28	R2

Forwarding –

How to pass a packet from an input interface to the output interface?

- Route specific
- Next hop method
- Network specific
- Host specific
- Default

Forwarding – **Route Specific**

Route Specific: In route specific routing, Routing table holds the information (IP address) about the complete route.

Routing table for host A

Destination	Route
Host B	R1, R2, Host B

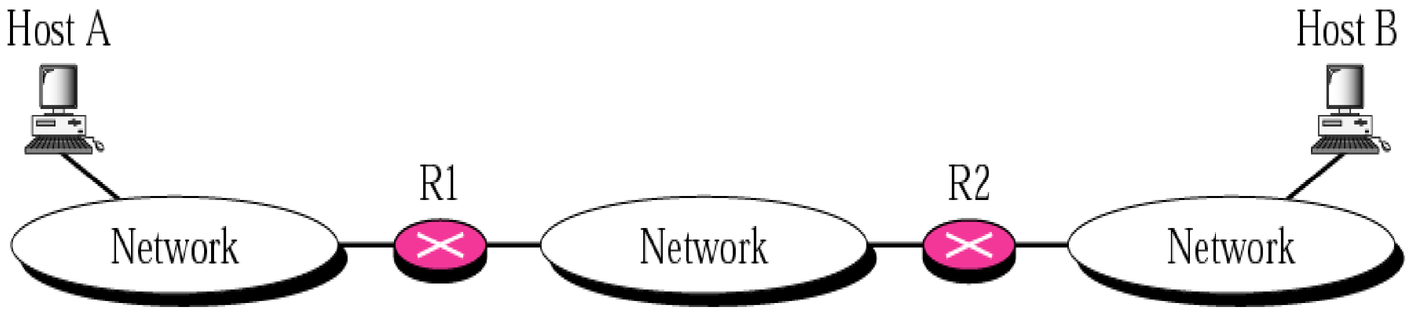
Routing table for R1

Destination	Route
Host B	R2, Host B

Routing table for R2

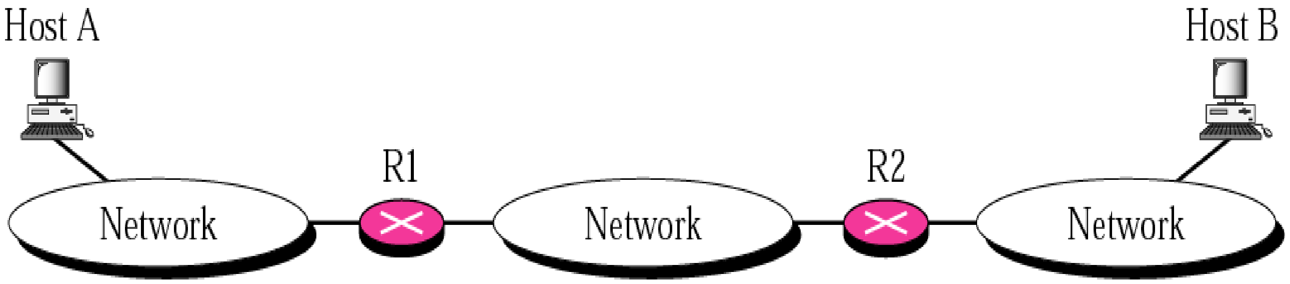
Destination	Route
Host B	Host B

a. Routing tables based on route



Forwarding – **Next hop method**

Next hop method: In next-hop routing, Routing table holds the information (IP address) that leads to the next hop (router) instead of holding information about the complete path.



Routing table for host A

Destination	Next Hop
Host B	R1

Routing table for R1

Destination	Next Hop
Host B	R2

Routing table for R2

Destination	Next Hop
Host B	Ñ

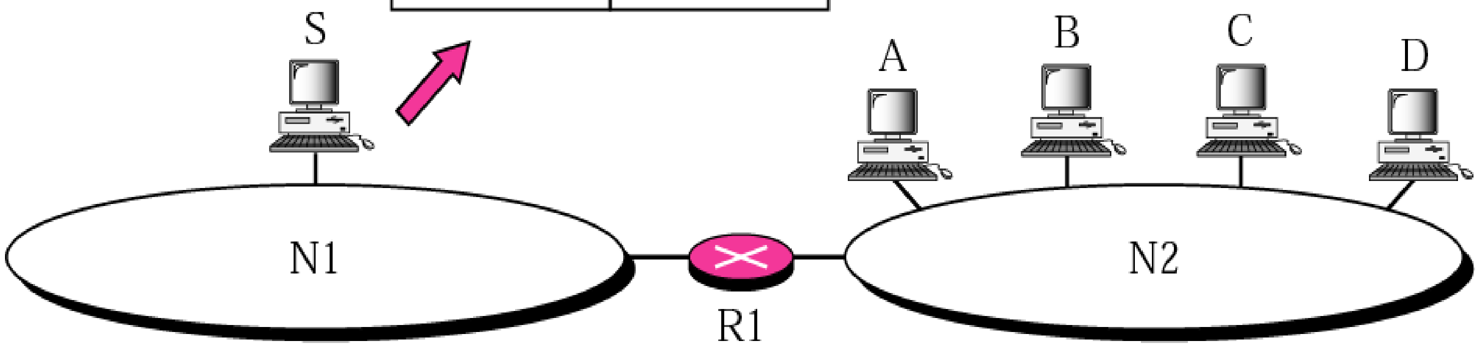
b. Routing tables based on next hop

Forwarding – **Network specific method**

Network Specific method: Instead of having entry for each host connected to the same network, the table contains only a single entry for the address of the destination network itself.

Routing table for host S based on network-specific method

Destination	Next Hop
N2	R1



Example **Network specific method**

	Mask	Destination address	Next-hop address	Interface
	/8	14.0.0.0	118.45.23.8	m1
Host Specific →	/32	192.16.7.1	202.45.9.3	m0
Network Specific →	/24	193.14.5.0	84.78.4.12	m2
Default →	/0	/0	145.11.10.6	m0

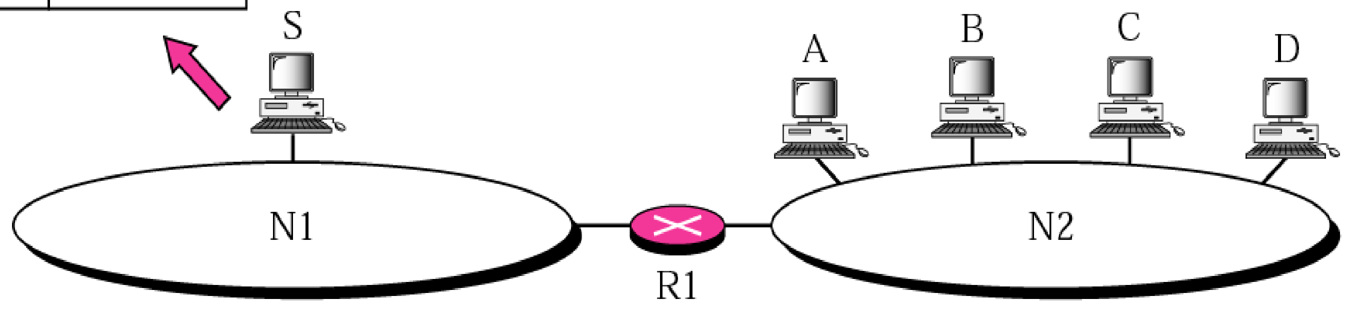
- Using the table above, the router receives a packet for destination **193.14.5.22**.
- For each row, the mask is applied to the destination address until a match with the next-hop address is found.
- In this example, the router sends the packet through **interface m2 (network specific).**

Forwarding – **Host Specific**

Host Specific Method: The destination host address is given in the routing table. It is used for checking the route or providing security measures.

Routing table for host S based on host-specific method

Destination	Next Hop
A	R1
B	R1
C	R1
D	R1



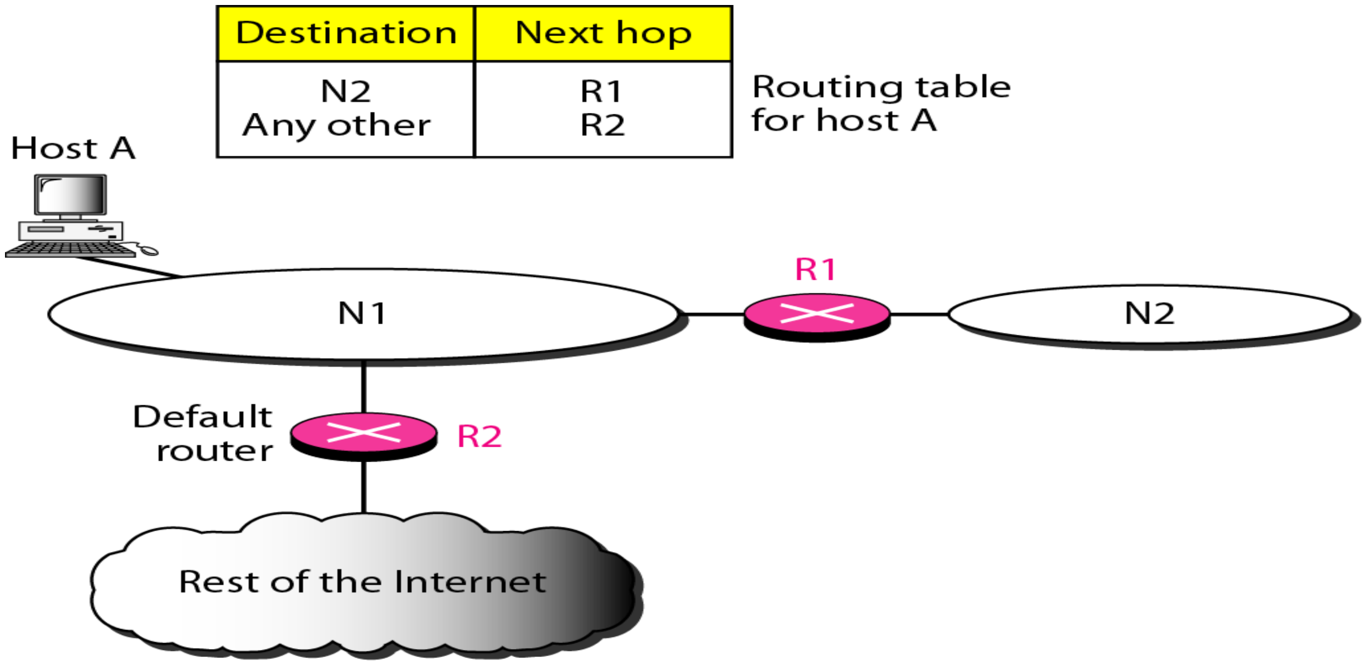
Example **Host Specific**

	Mask	Destination address	Next-hop address	Interface
	/8	14.0.0.0	118.45.23.8	m1
Host Specific →	/32	192.16.7.1	202.45.9.3	m0
Network Specific →	/24	193.14.5.0	84.78.4.12	m2
Default →	/0	/0	145.11.10.6	m0

- Using the table above, the router receives a packet for destination **192.16.7.1**.
- For each row, the mask is applied to the destination address until a match with the destination address is found.
- In this example, the router sends the packet through **interface m0 (host specific).**

Forwarding – Default Method

Default Method: Default router is used if the destination network address is not found in the routing table.



Example 3 - Default Method

	Mask	Destination address	Next-hop address	Interface
	/8	14.0.0.0	118.45.23.8	m1
Host Specific →	/32	192.16.7.1	202.45.9.3	m0
Network Specific →	/24	193.14.5.0	84.78.4.12	m2
Default →	/0	/0	145.11.10.6	m0

- Using the table above, the router receives a packet for destination **200.34.12.34**.
- For each row, the mask is applied to the destination address, but no match is found.
- In this example, the router sends the packet through the **default interface m0.**

General Tips for forwarding

	Mask	Destination address	Next-hop address	Interface
	/8	14.0.0.0	118.45.23.8	m1
Host Specific	/32	192.16.7.1	202.45.9.3	m0
Network Specific	/24	193.14.5.0	84.78.4.12	m2
Default	/0	/0	145.11.10.6	m0

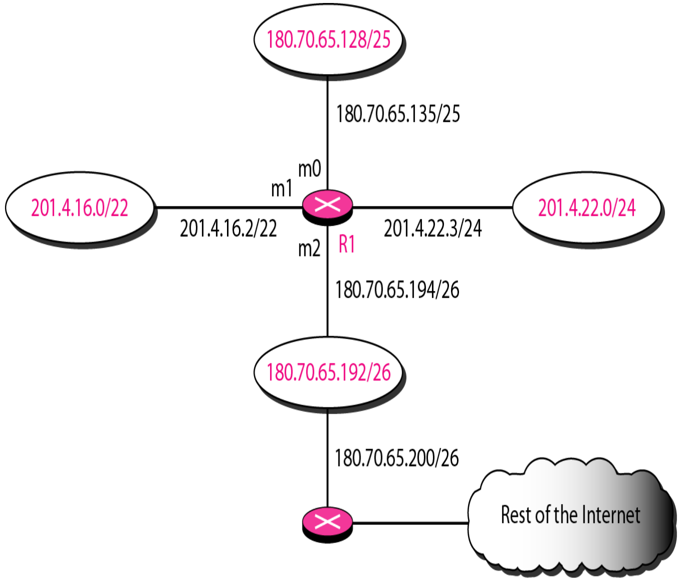
When a packet arrives:

- Apply all the available masks to the IP destination address
- If a match is found in the destination address column, the packet has to be forwarded to the next hop IP address through the corresponding interface
- If no match is found, send the packet through the default interface Otherwise, a message “host unreachable error” is sent back to the sender.

Example - 4

Show the forwarding process if a packet arrives at R1 with the destination address 180.70.65.140.

Mask	*Network Address*	*Next Hop*	*Interface*
/26	180.70.65.192	—	m2
/25	180.70.65.128	—	m0
/24	201.4.22.0	—	m3
/22	201.4.16.0	....	m1
Any	Any	180.70.65.200	m2



Solution

The router performs the following steps:

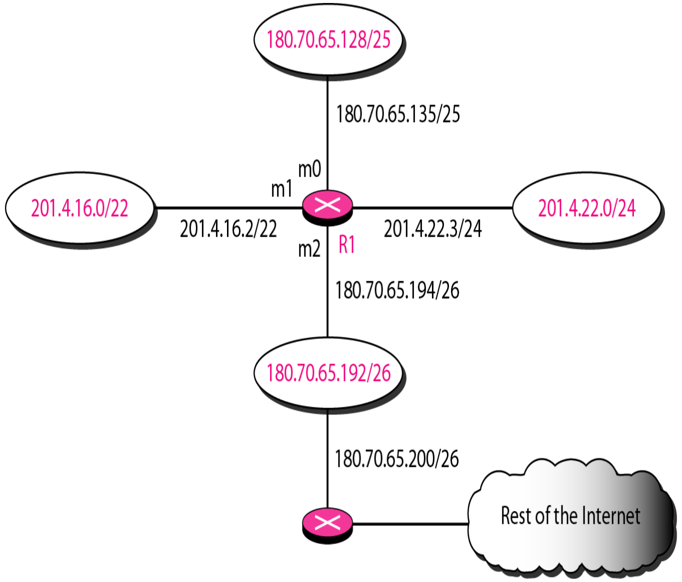
1. *The first mask (/26) is applied to the destination address. The result is 180.70.65.192, which does not match the corresponding network address.*

2. *The second mask (/25) is applied to the destination address. The result is 180.70.65.128, which matches the corresponding network address.*

Example - 5

Show the forwarding process if a packet arrives at R1 with the destination address 201.4.22.35.

Mask	*Network Address*	*Next Hop*	*Interface*
/26	180.70.65.192	—	m2
/25	180.70.65.128	—	m0
/24	201.4.22.0	—	m3
/22	201.4.16.0	....	m1
Any	Any	180.70.65.200	m2



Solution

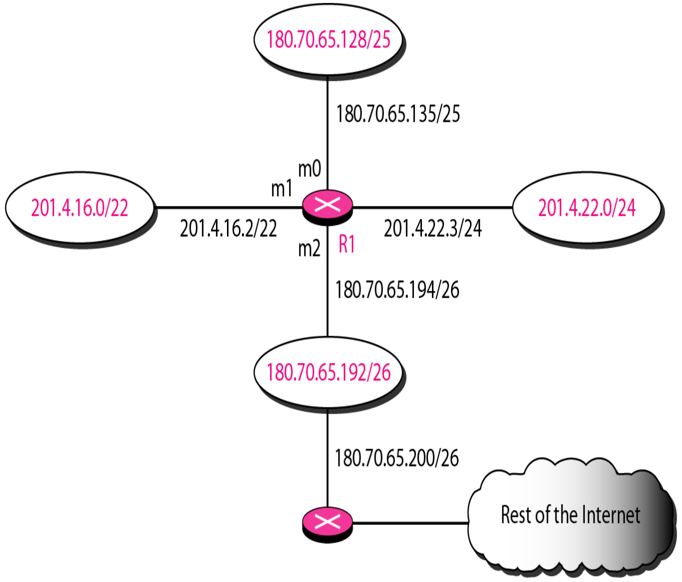
The router performs the following steps:

1. *The first mask (/26) is applied to the destination address. The result is 201.4.22.0, which does not match the corresponding network address.*
2. *The second mask (/25) is applied to the destination address. The result is 201.4.22.0, which does not match the corresponding network address (row 2).*
3. *The third mask (/24) is applied to the destination address. The result is 201.4.22.0, which matches the corresponding network address but since subnet mask is not matched packet wont be forwarded..*

Example - 6

Show the forwarding process if a packet arrives at R1with the destination address 180.24.32.78.

Mask	*Network Address*	*Next Hop*	*Interface*
/26	180.70.65.192	—	m2
/25	180.70.65.128	—	m0
/24	201.4.22.0	—	m3
/22	201.4.16.0	....	m1
Any	Any	180.70.65.200	m2

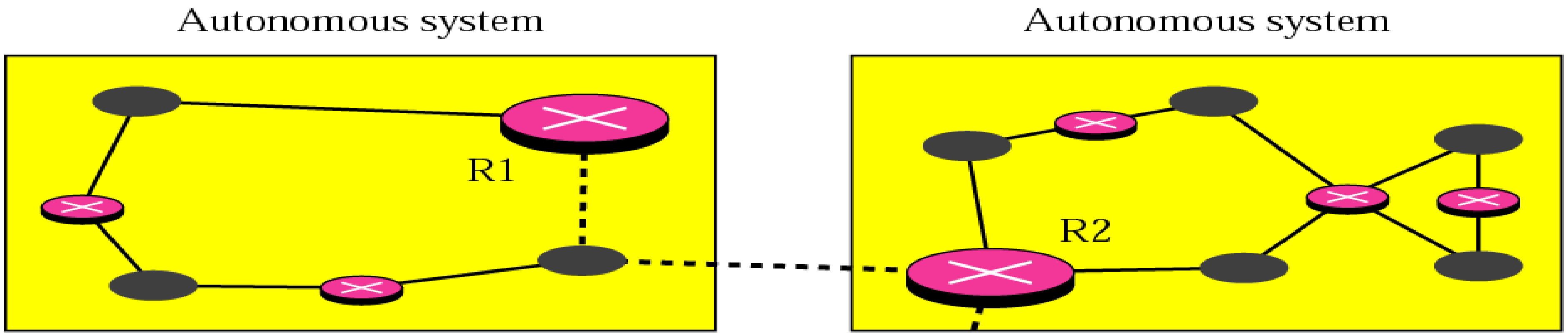


Solution

This time all masks are applied, one by one, to the destination address, but no matching network address is found. When it reaches the end of the table, the module gives the next-hop address 180.70.65.200 and interface number m2 to ARP. This is probably an outgoing package that needs to be sent, via the default router, to someplace else in the Internet.

Intra and Inter Domain Routing

- Internet is divided into **Autonomous Systems (AS)**
- An **AS** is a group of networks and routers under a single administration.



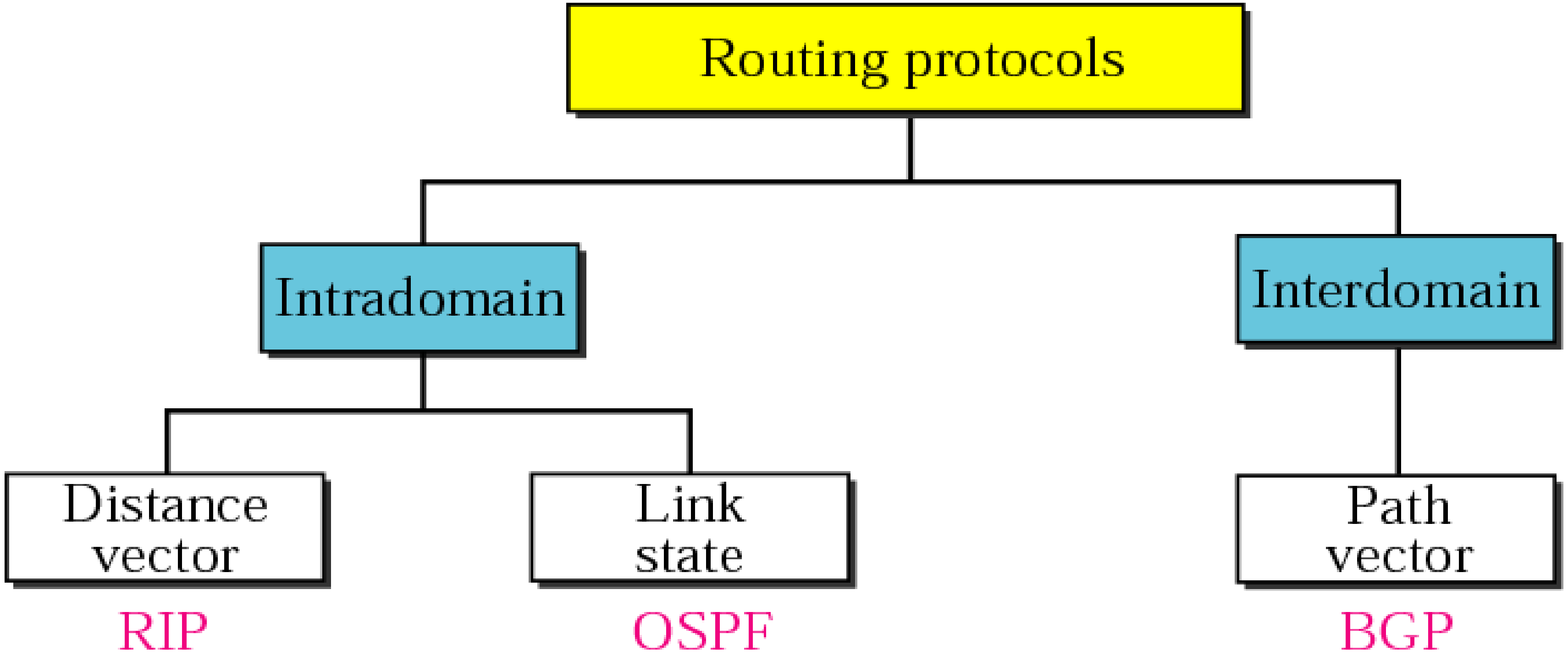
- Routing inside an autonomous system is referred to as **intra-domain** routing.
- Routing between autonomous systems is **inter-domain routing**.

Autonomous System

- On the Internet, Autonomous system (AS) is either a **single network** or a **group of networks** that is controlled by a common network administrator.
- **Autonomous System Number (ASN)** - An autonomous system is assigned a globally unique number
- **Intra-AS (interior) routing protocol** - Routers in **same** AS run **same** routing protocol.
- **Inter-AS** or **exterior routing** - Routing between autonomous systems.

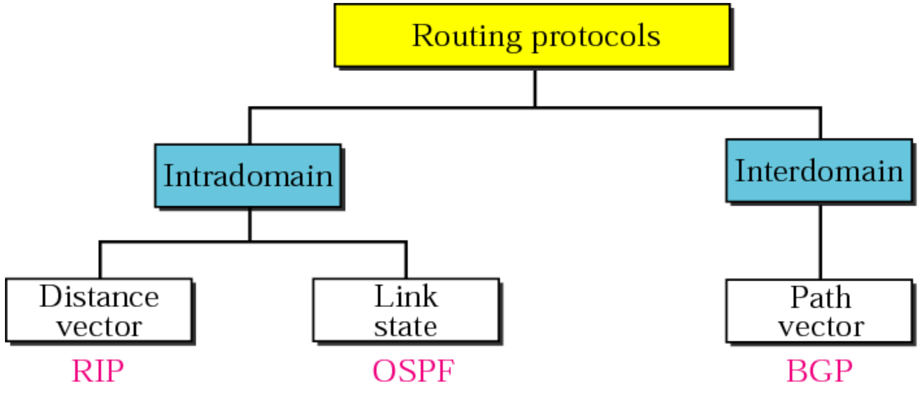
- **Border routers or gateways routers** - AS systems are connected by special routers called **boarder router or gateway router.**
- Gateways routers (border routers) are special routers in AS that run **intra-AS routing** protocol and responsible for routing to destinations outside AS by running **inter-AS (exterior) routing** protocol with other gateway (boarder) routers

Autonomous System



RIP – Routing Information Protocol

Distance Vector Routing



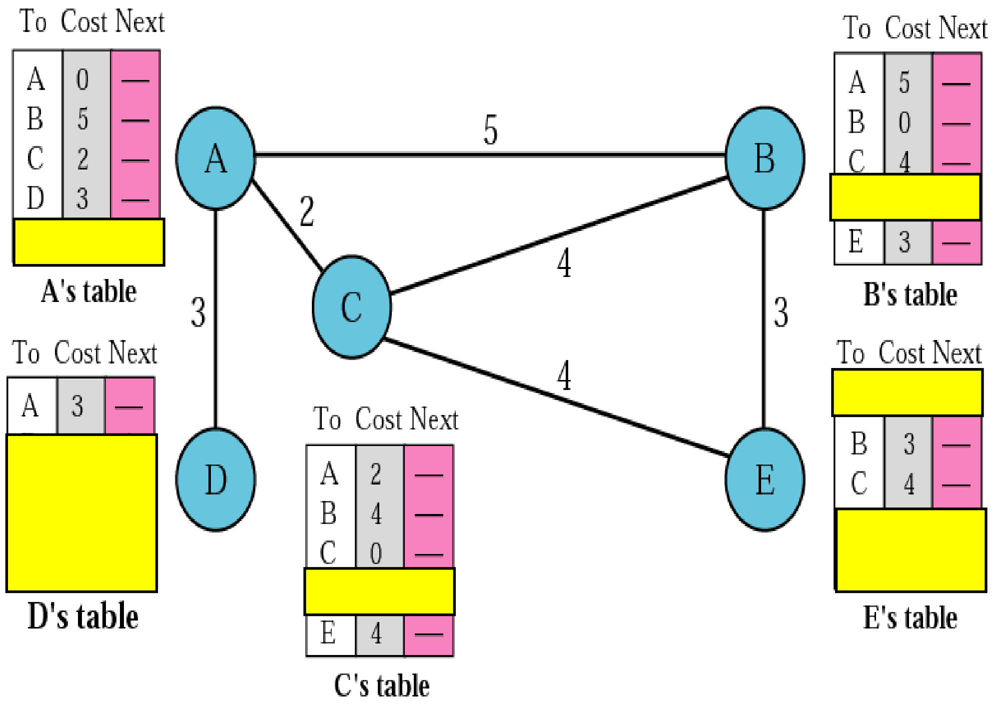
- Intra domain routing protocol used inside an autonomous system.
- Based on **Distance Vector Routing** which finds the least cost route.

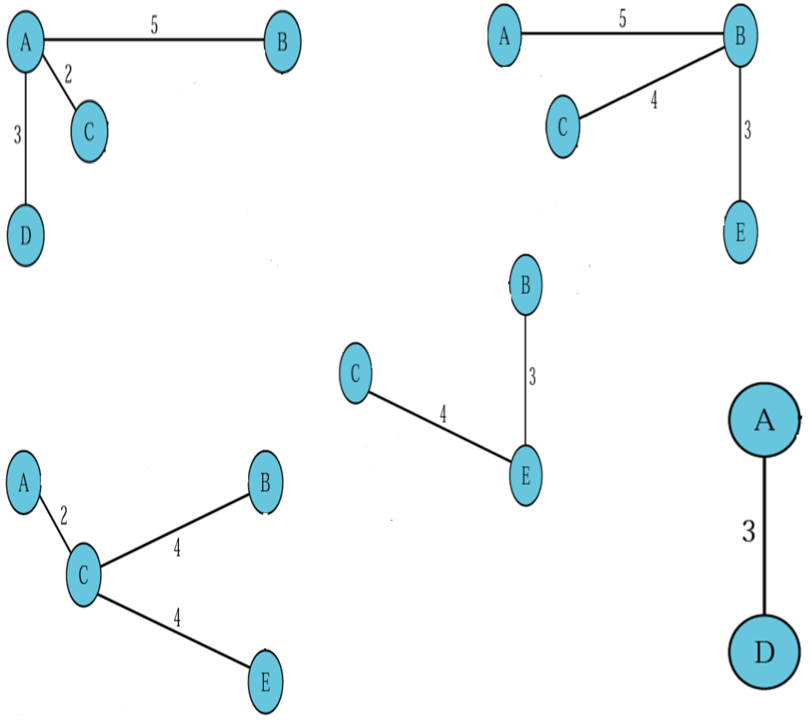
- Implements distance vector as ….
  - Destination in routing table is network.
  - Distance is no. of hops – no. of n/w have to use to reach destination.
  - Next node defines the address of the router to which the packet is to be sent.

Distance vector routing

- Assume each router knows its own address and cost to reach each of its directly connected neighbors

- Uses **Bellman-Ford algorithm** -Distributed route computation is done using only neighbor's info



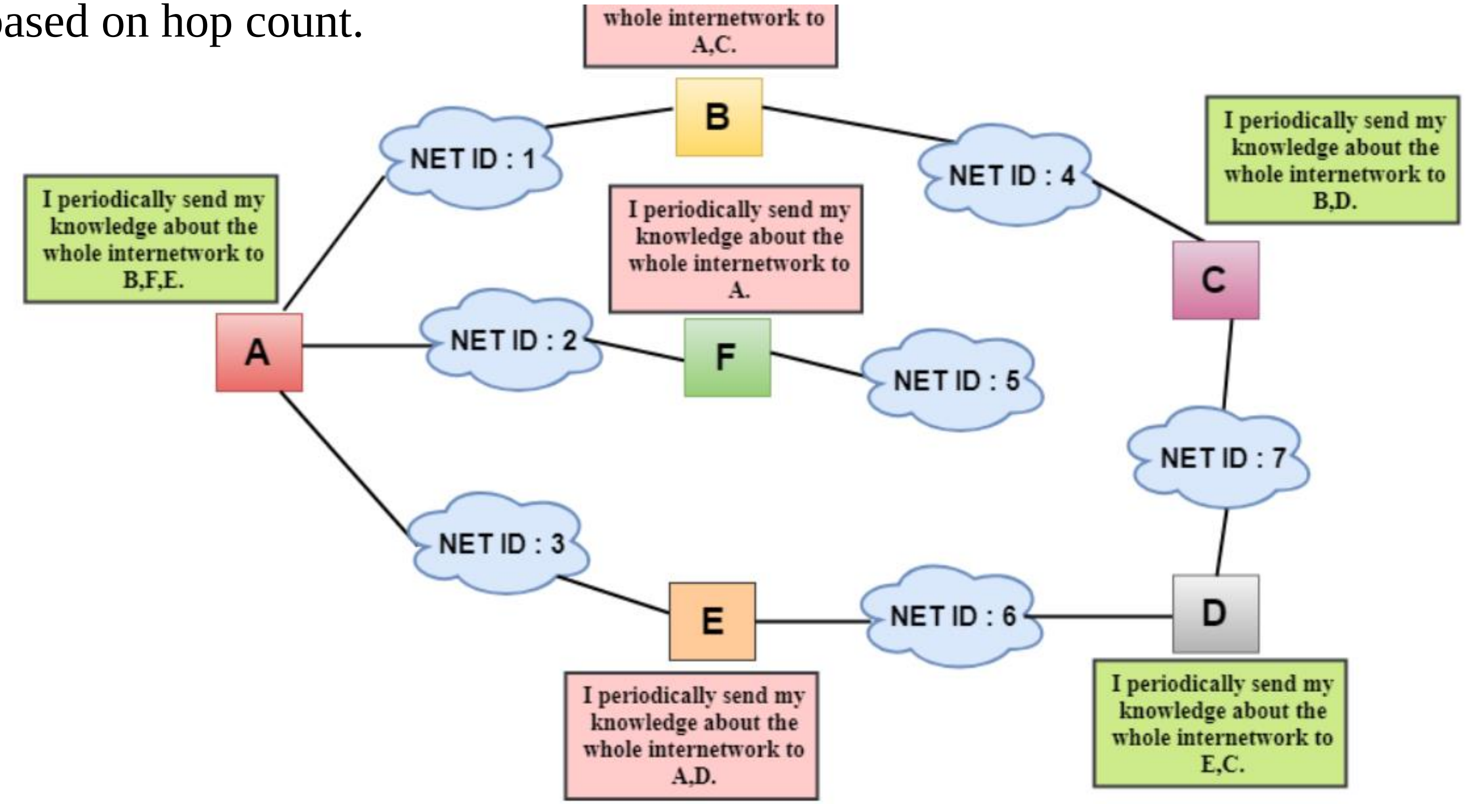


Distance vector routing

- Method considers AS with all **routers & network** as a ***graph***. Routers are considered as a node & network as a line.

- The cost of every link is one unit. Therefore, the efficiency of transmission can be measured by the number of links to reach the destination.

- In Distance vector routing, the cost is based on hop count.



Distance vector routing

- Each router (node) maintains table(vector) of minimum distances to every router (node).
- Table also guides the packet to the desired node by showing next hop.
- Each table updates it's table **asynchronously**
- Table Entry contains :
  - Destination
  - Hop count (cost)
  - Next router to deliver the packet to.

Distance vector routing - Steps

keys to understand how this algorithm works:

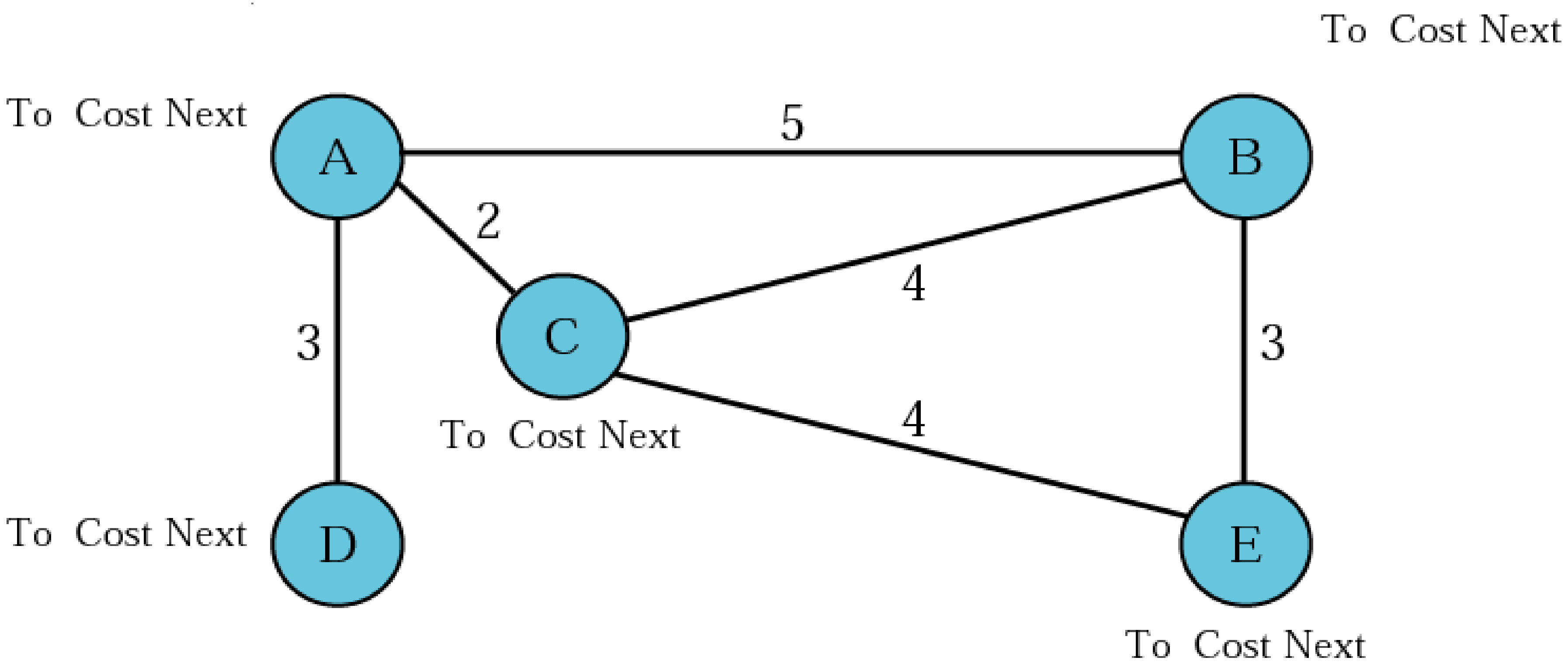
- Sharing knowledge about the entire AS.
- Each router shares its knowledge about the entire AS with neighbours. It sends whatever it has.
- Sharing only with immediate neighbours. Each router sends whatever knowledge it has through all its interface.
- Sharing at regular intervals. sends at fixed intervals, e.g. every 30 sec.

Steps……………..

1. Initialization
2. Sharing
3. Update

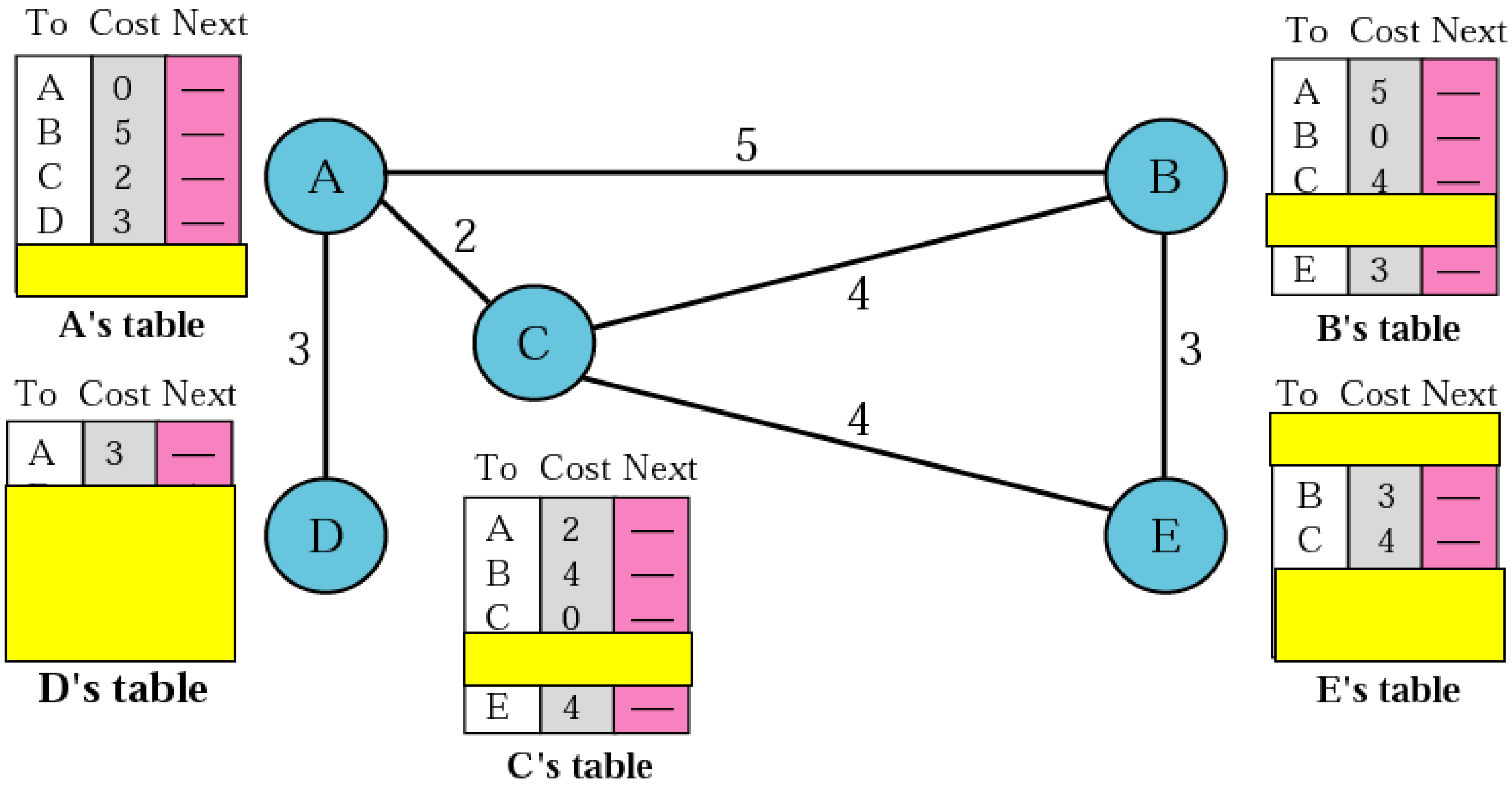
Distance vector routing Initialization

- At the beginning, each node knows only the distance between itself & it's immediate neighbors those are directly connected.



Distance vector routing - Initialization

- At the beginning, each node knows only the distance between itself & it’s immediate neighbors those are directly connected.



Distance vector routing – Sharing & Updating

- Each node shares its partial routing table with its immediate neighbors periodically (30 sec.) and when there is a change (triggered update).

- The neighbors add this knowledge to their own knowledge and sends the updated table to their own neighbors. In this way, routers get its own information plus the new information about the neighbors.

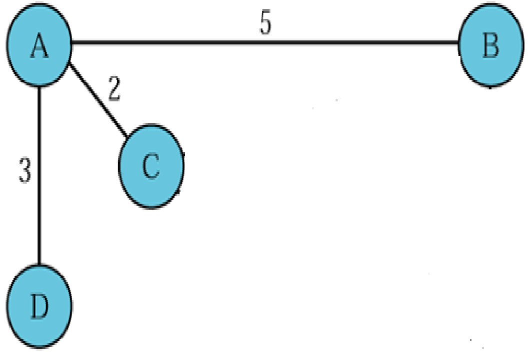
- The receiving node follows the following steps to update it's table -
    1. Adds the cost between itself and sending node to each value of the table.
    2. Adds the name of the sending node to each row as third column.
    3. Compares each row of it's old table with the corresponding row of the modified version.
        - ✓ If next node entry is different, smaller cost row is selected.
        - ✓ If there is a tie , old one is kept.
        - ✓ If To node entry is not in the old table, the receiving node chooses the new row for that

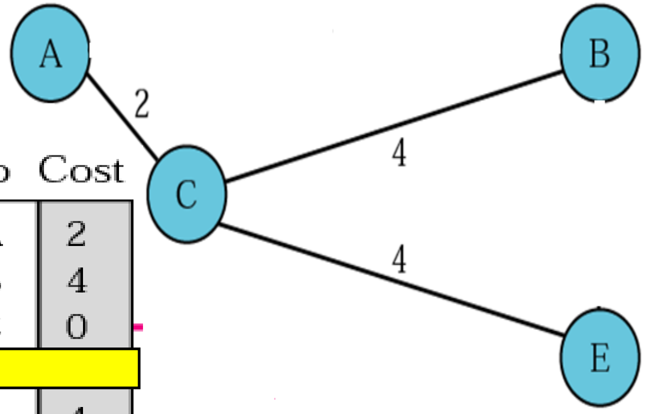
Distance vector routing - Sharing & Updating

C is sharing its table to A

To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—

A's table





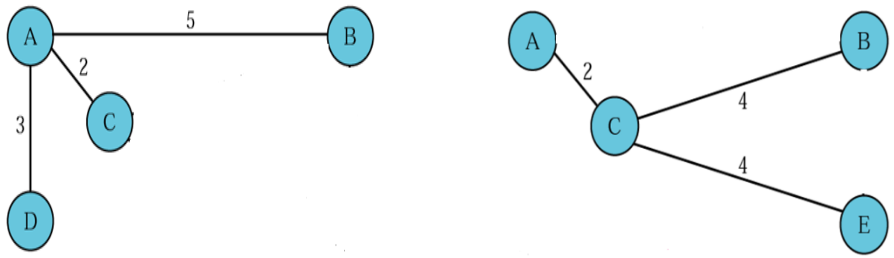
To	Cost
A	2
B	4
C	0
E	4

Distance vector routing – Sharing & Updating

- The receiving node follows the following steps to update it's table -
  1. Adds the cost between itself and sending node to each value of the table.
  2. Adds the name of the sending node to each row as third column.
  3. Compares each row of it's old table with the corresponding row of the modified version.
     - ✓ If next node entry is different, smaller cost row is selected.
     - ✓ If there is a tie , old one is kept.
     - ✓ If To node entry is not in the old table, the receiving node chooses the new row for that

Distance vector routing - Sharing & Updating

1. Adds the cost between itself and it’s sending node to each value of its own table.



To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—

A's table

To	Cost
A	2
B	4
C	0
E	4

Received from C

To	Cost
A	4
B	6
C	2
E	6

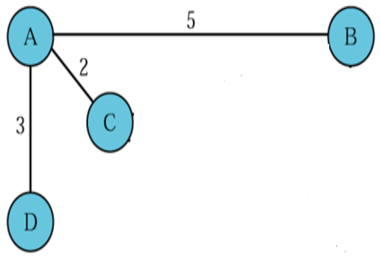
Here table is shared by C, so add cost of C to from A’s table to the received table

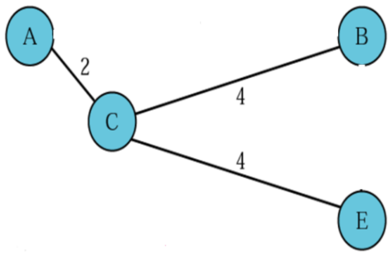
Distance vector routing – Sharing & Updating

- The receiving node follows the following steps to update it’s table -
  1. Adds the cost between itself and sending node to each value of the table.
  2. Adds the name of the sending node to each row as third column.
  3. Compares each row of it’s old table with the corresponding row of the modified version.
     - ✓ If next node entry is different, smaller cost row is selected.
     - ✓ If there is a tie , old one is kept.
     - ✓ If To node entry is not in the old table, the receiving node chooses the new row for that

Distance vector routing - Sharing & Updating

2. Adds the name of the sending node to each row as third column.





To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—

A's table

To	Cost
A	2
B	4
C	0
E	4

Received from C

To	Cost	Next
A	4	C
B	6	C
C	2	C
E	6	C

A's Modified Table

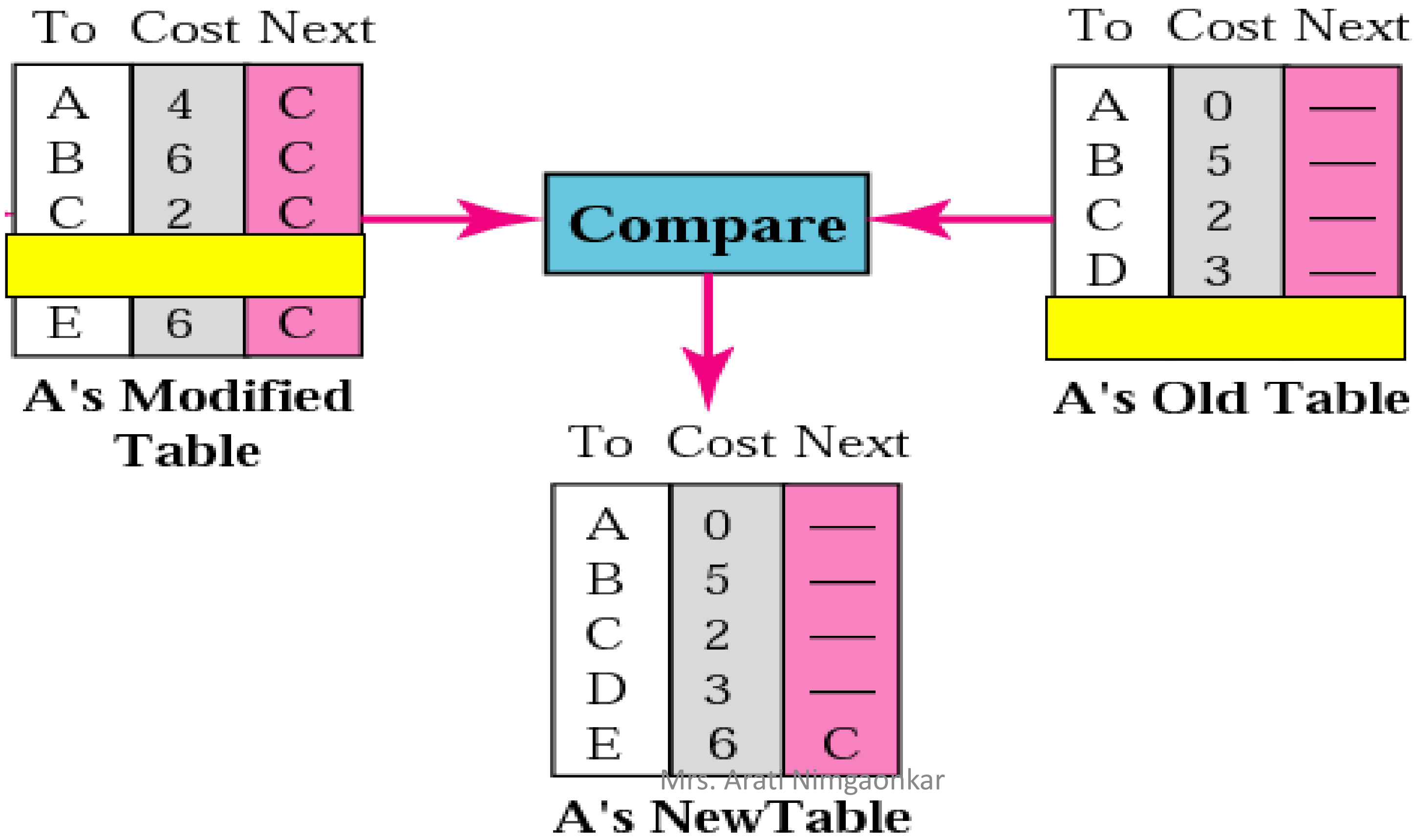
Distance vector routing – Sharing & Updating

- The receiving node follows the following steps to update it's table -
  1. Adds the cost between itself and sending node to each value of the table.
  2. Adds the name of the sending node to each row as third column.
  3. Compares each row of it's old table with the corresponding row of the modified version.
     - ✓ If next node entry is different, smaller cost row is selected.
     - ✓ If there is a tie , old one is kept.
     - ✓ If To node entry is not in the old table, the receiving node chooses the new row for that

Distance vector routing - Sharing & Updating

3. Compares each row of it’s old table with the corresponding row of the modified version.

- ✓ If next node entry is different, smaller cost row is selected.
- ✓ If there is a tie , old one is kept.
- ✓ If To node entry is not in the old table, the receiving node chooses the new row for that.



Distance vector routing - Sharing & Updating

Complete process ………

To	Cost
A	2
B	4
C	0
D	∞
E	4

Received from C

To	Cost	Next
A	4	C
B	6	C
C	2	C
D	∞	C
E	6	C

A's Modified Table

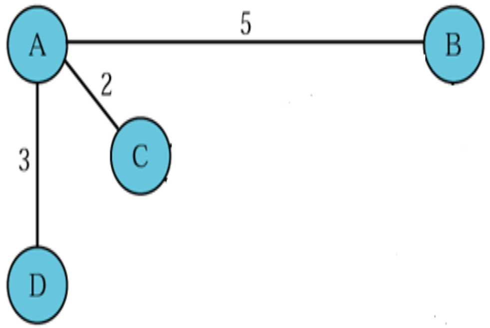
Compare

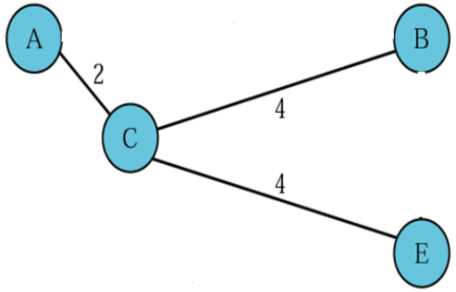
To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	∞	

A's Old Table

To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

A's NewTable





Distance vector routing - Sharing & Updating

Complete process ………

To	Cost
A	2
B	4
C	0
D	∞
E	4

Received from C

To	Cost	Next
A	4	C
B	6	C
C	2	C
D	∞	C
E	6	C

A's Modified Table

To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	∞	

A's Old Table

Compare

To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

A's NewTable

Distance vector routing - Sharing & Updating

- **Periodic Update:**

  A node sends its table, normally every **30sec**, in a periodic update.

- **Triggered Update:**

  A node sends its routing table to its neighbors anytime there is a change in its routing table.

  This change can result from the following:

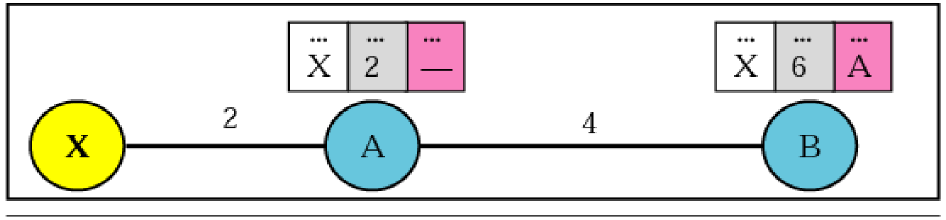
  ✓ A node receives a table from a neighbor, resulting in changes in its own table after updating.

  ✓ A node detects some failure in the neighboring links which results in a distance change to infinity.

Distance vector routing – Two node loop instability

Two-node instability – what can happen with distance vector routing

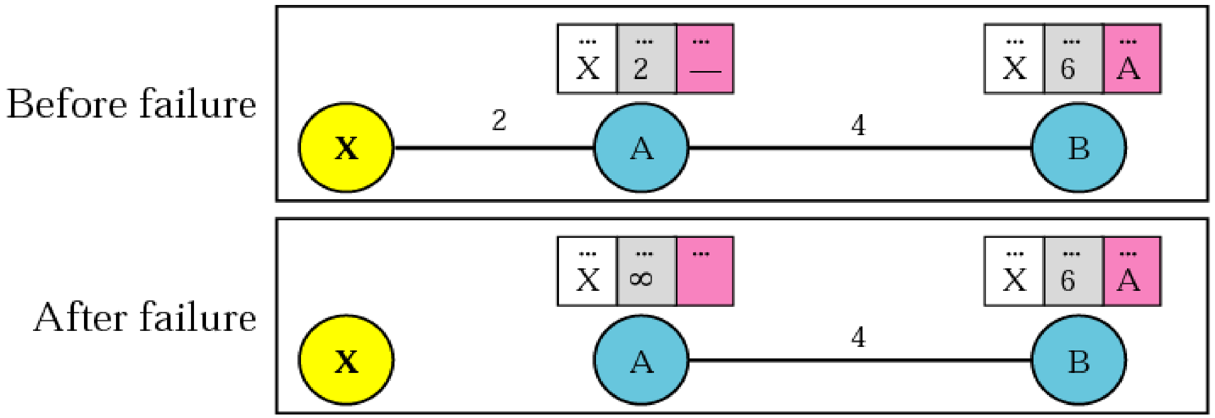
Before failure



Both A and B know where X is.

Distance vector routing – Two node loop instability

Two-node instability – what can happen with distance vector routing

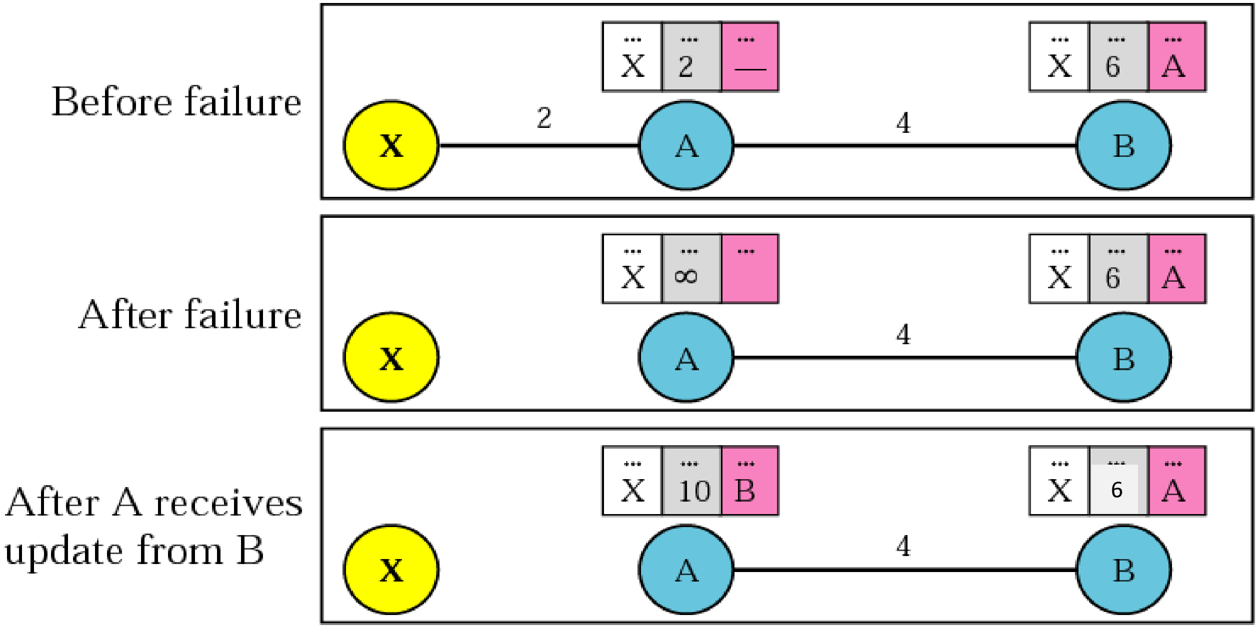


Both A and B know where X is.

Link between **A and X fails**. **A** updates its table immediately.

Distance vector routing – Two node loop instability

Two-node instability – what can happen with distance vector routing



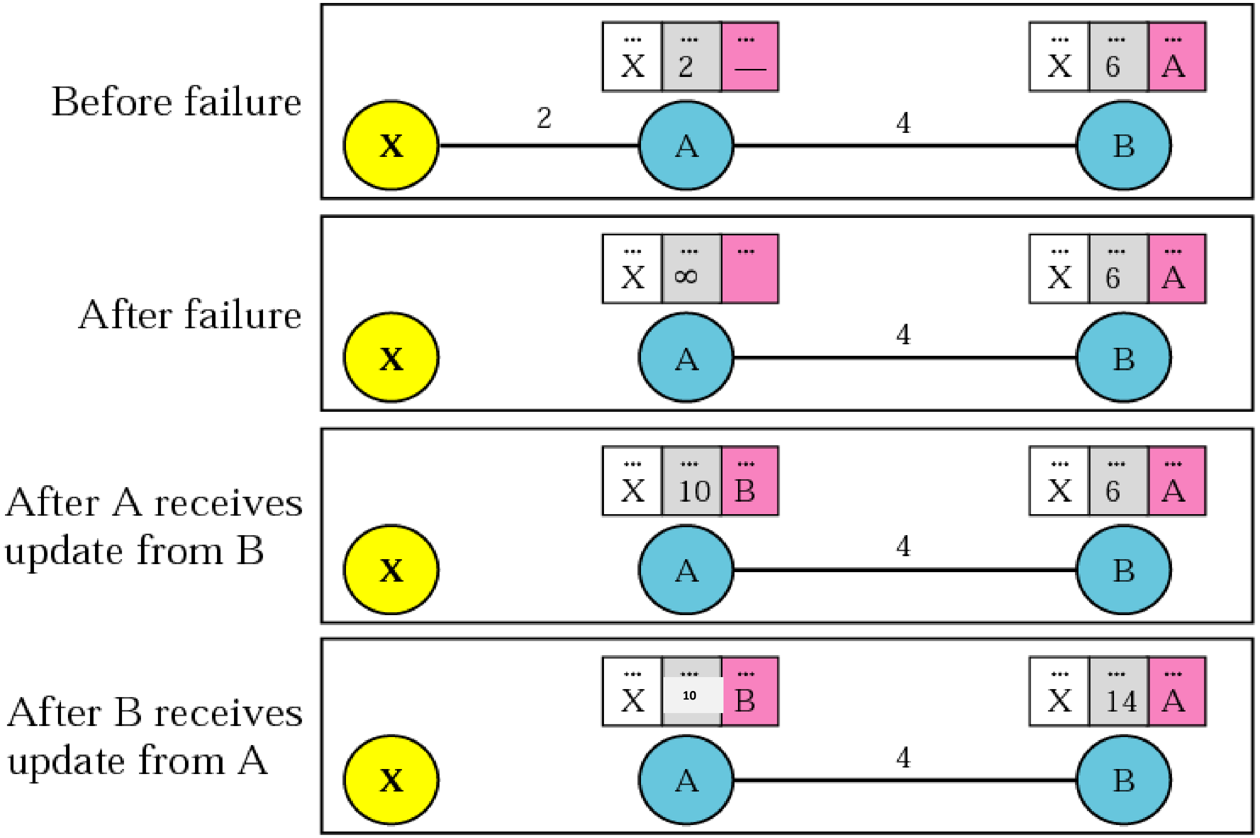
Both A and B know where X is.

Link between **A and X fails**. **A** updates its table immediately.

But before A can tell B, B sends its info to A!

Distance vector routing – Two node loop instability

Two-node instability – what can happen with distance vector routing



Both A and B know where X is.

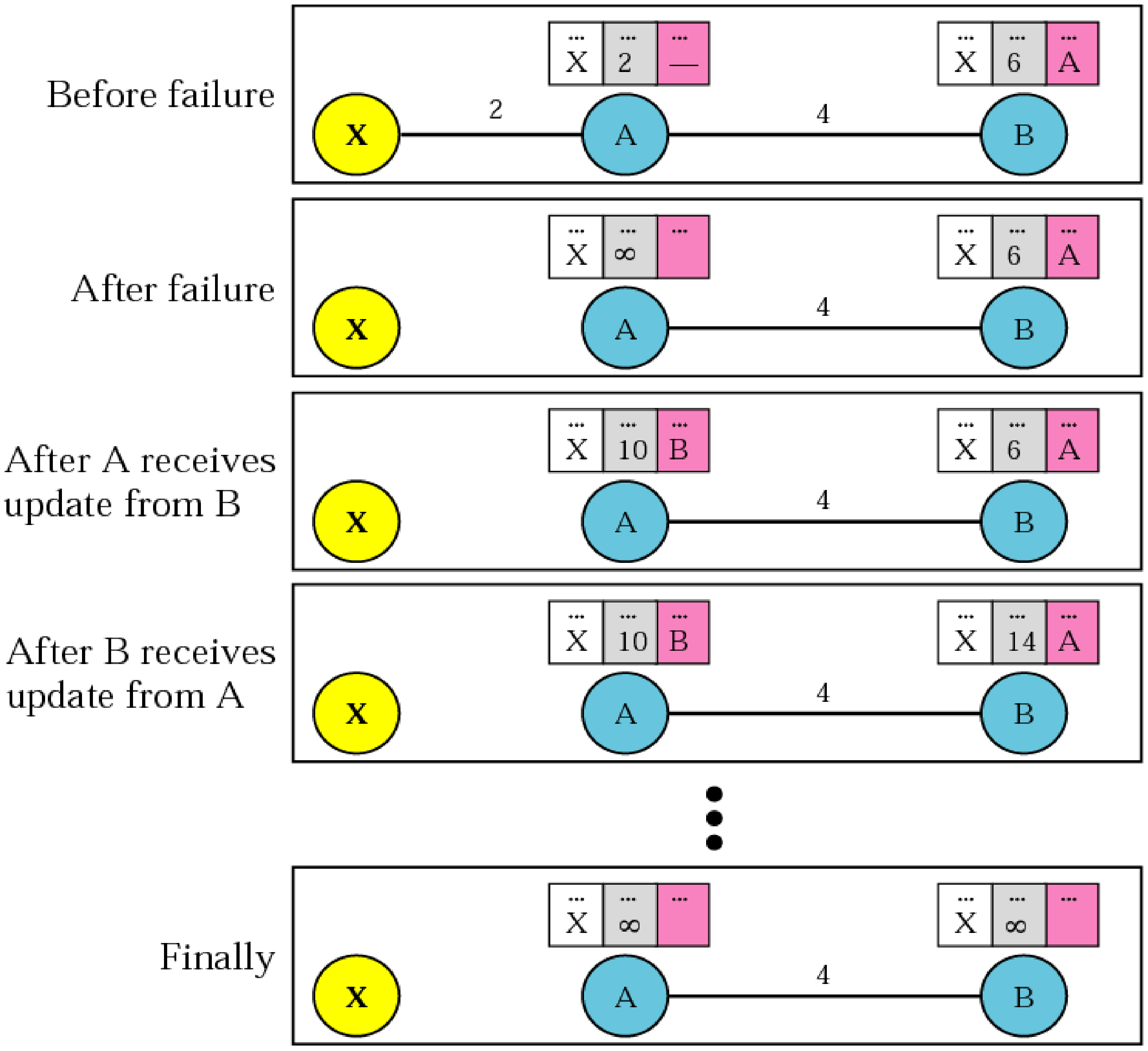
Link between **A and X fails**. **A** updates its table immediately.

But before A can tell B, B sends its info to A!

A, using B's info, updates its table **(error!)**. Then A send its table to B and B updates its table **(more error).**

Distance vector routing – Two node loop instability

Two-node instability – what can happen with distance vector routing



Both A and B know where X is.

Link between **A and X fails**. **A** updates its table immediately.

But before A can tell B, B sends its info to A!

A, using B's info, updates its table **(error!)**. Then A send its table to B and B updates its table **(more error).**

Both routers keep updating tables, eventually hitting infinity.!

Distance vector routing – Two node loop instability-Solution

- Defining Infinity
- Split Horizon

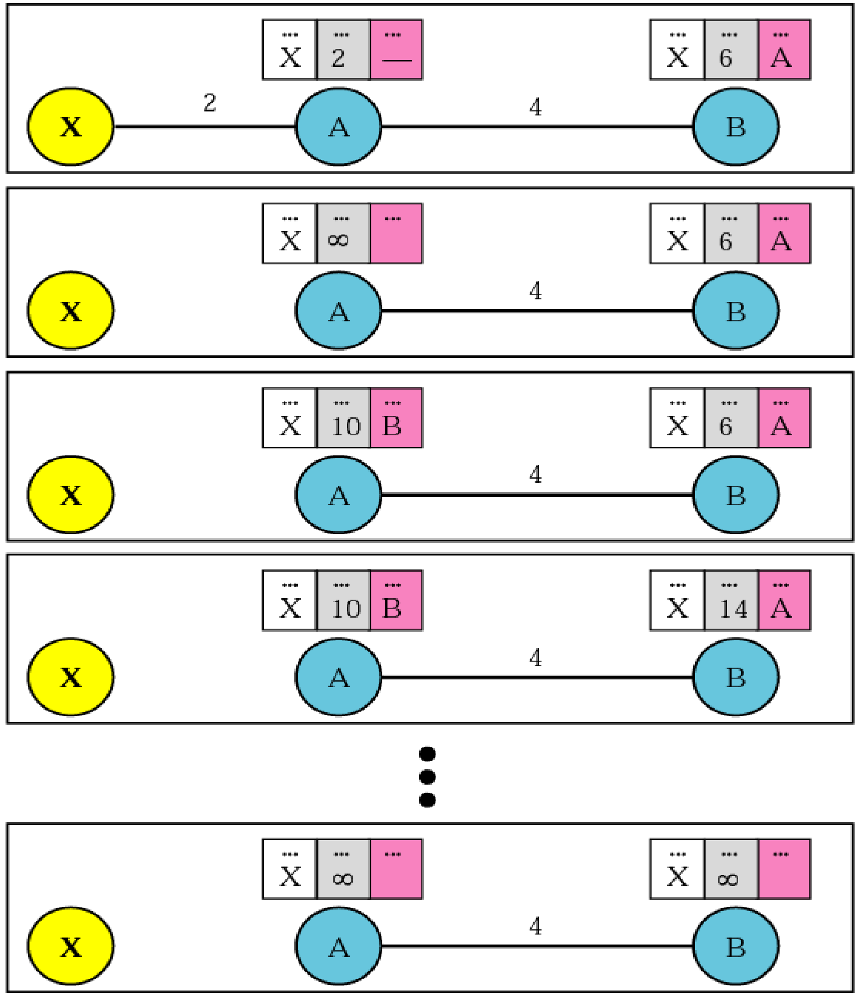
Before failure

After failure

After A receives update from B

After B receives update from A

Finally



Distance vector routing – Solution to Two node loop instability

1. **Define infinity**

- To be a much smaller value, such as 16.
- Then it doesn't take too long to become stable.
- **But now you can't use distance vector routing in large networks.**

2. **Split Horizon** –

- Instead of flooding entire table to each node, **only part of its table** is sent.
- More precisely, if node B thinks that the optimum route to reach X is via A, then B does not need to advertise this piece of info to A – **the info has already come from A.**

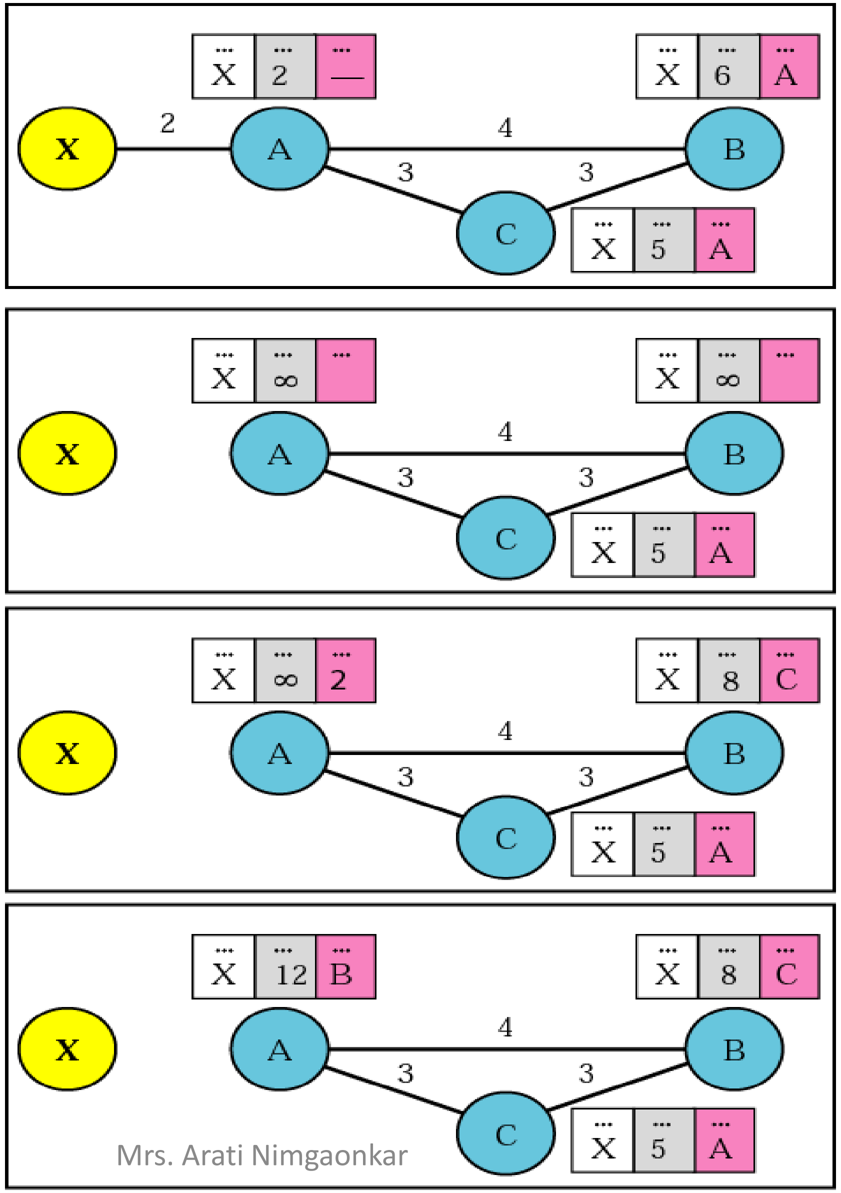
Distance vector routing – Three node loop instability

Before failure

After A sends the route to B and C, but the packet to C is lost

After C sends the route to B

After B sends the route to A



Distance vector routing – Three node loop instability Solution

Holddown Timer:

- Hold-down timers can be used to avoid the formation of loops.
- Hold-down timer immediately starts when the router is informed that attached link is down.
- Till this time, router ignores all updates of down route unless it receives an update from the router of that downed link.
- During the timer, If the down link is reachable again, routing table can be updated.

RIP – Routing Information Protocol

- Intra domain routing protocol used inside an autonomous system.
- Based on **distance vector routing**.
- Implements distance vector as ….

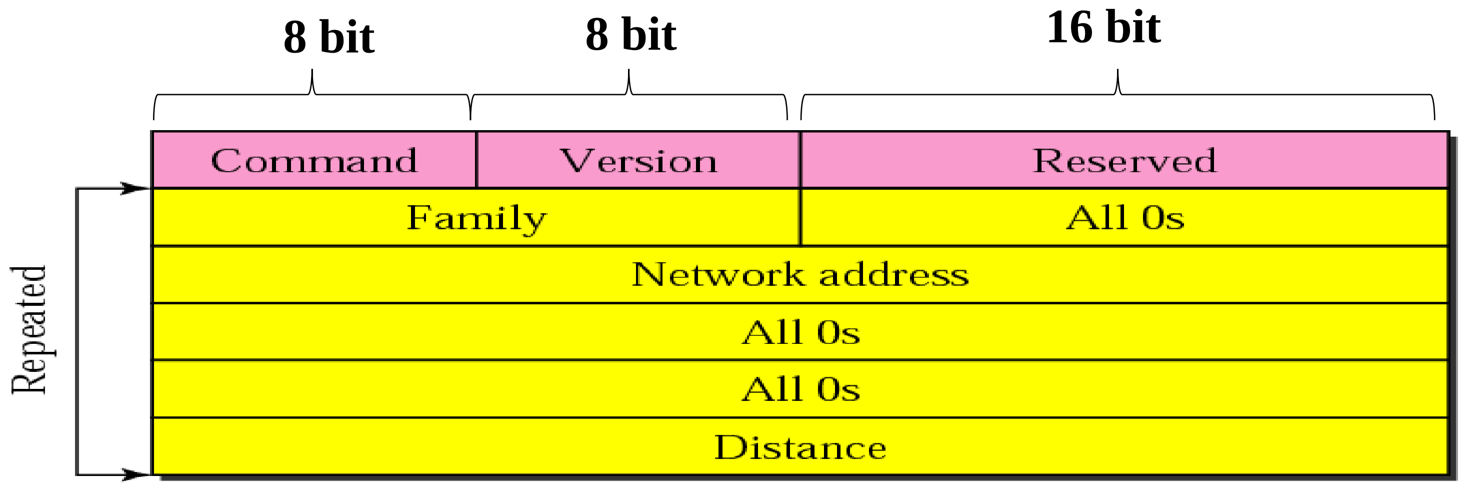
    - Destination in routing table is network.
    - Distance is no. of hops or n/w have to use to reach destination.
    - Infinity is 16.
    - Next node defines the address of the router to which the packet is to be sent.

- RIP uses the services of UDP on the port 520.

- RIP message is encapsulated in UDP datagram

RIP – Message format version 1

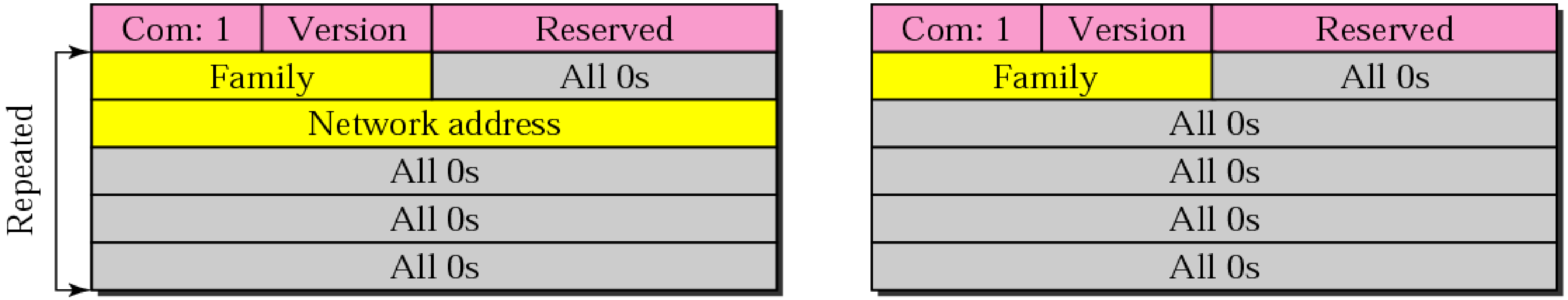
- Command - Type of message.
  - request (1) : sent by router
  - response (2) : Can be solicited or unsolicited
- Version. This 8-bit field defines the version.
- Family – Defines family of protocol. . For TCP/IP the value is 2.
- Network address - 4 bytes field defines address of the destination network.
- Distance – 32 bit field define hop count.



Note that part of the message is repeated for each destination network. We refer to this as an *entry*

RIP – Request and Response messages

- **Request Message**
  - A request message is sent by a router that has just come up or by a router that has some time-out entries.
  - A request can ask about specific entries or all entries.

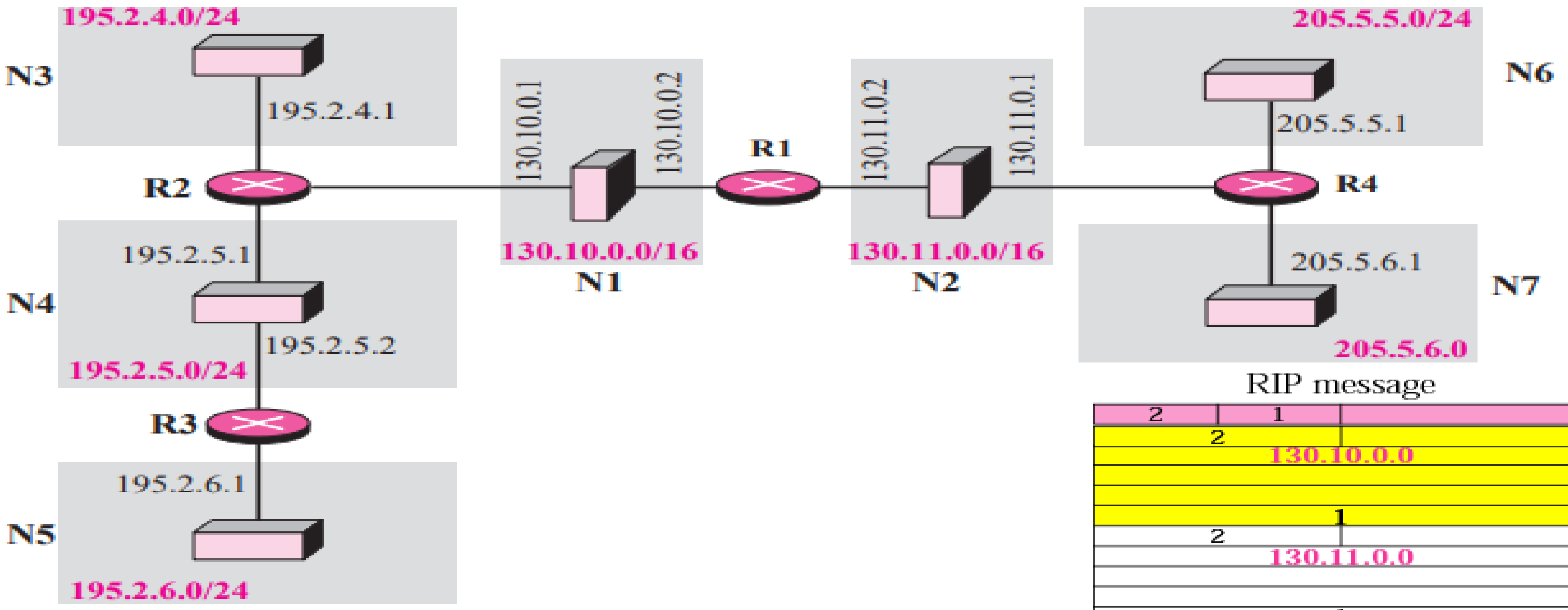


a. Request for some

b. Request for all

- **Response (Called as Update Packet)**–
  - Solicited Response – Answer to request.
  - Unsolicited Response - Periodically , when there is a change in the routing table.

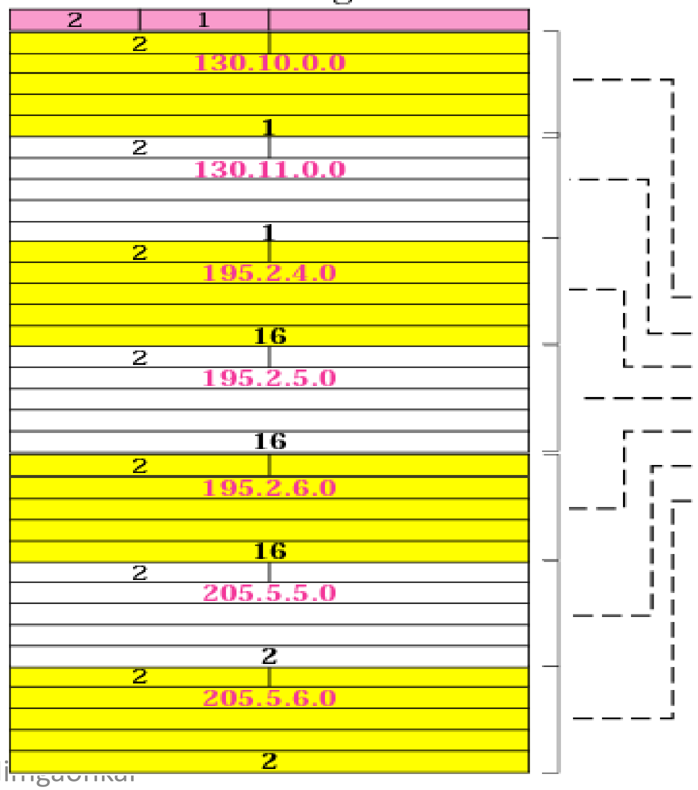
RIP – *update message sent from router R1 to router R2*



Dest.	Cost	Next
130.10.0.0/16	2	195.2.5.1
130.11.0.0/16	3	195.2.5.1
195.2.4.0/24	2	195.2.5.1
195.2.5.0/24	1	——
195.2.6.0/24	1	——
205.5.5.0/24	4	195.2.5.1
205.5.6.0/24	4	195.2.5.1

R3 Table

RIP message

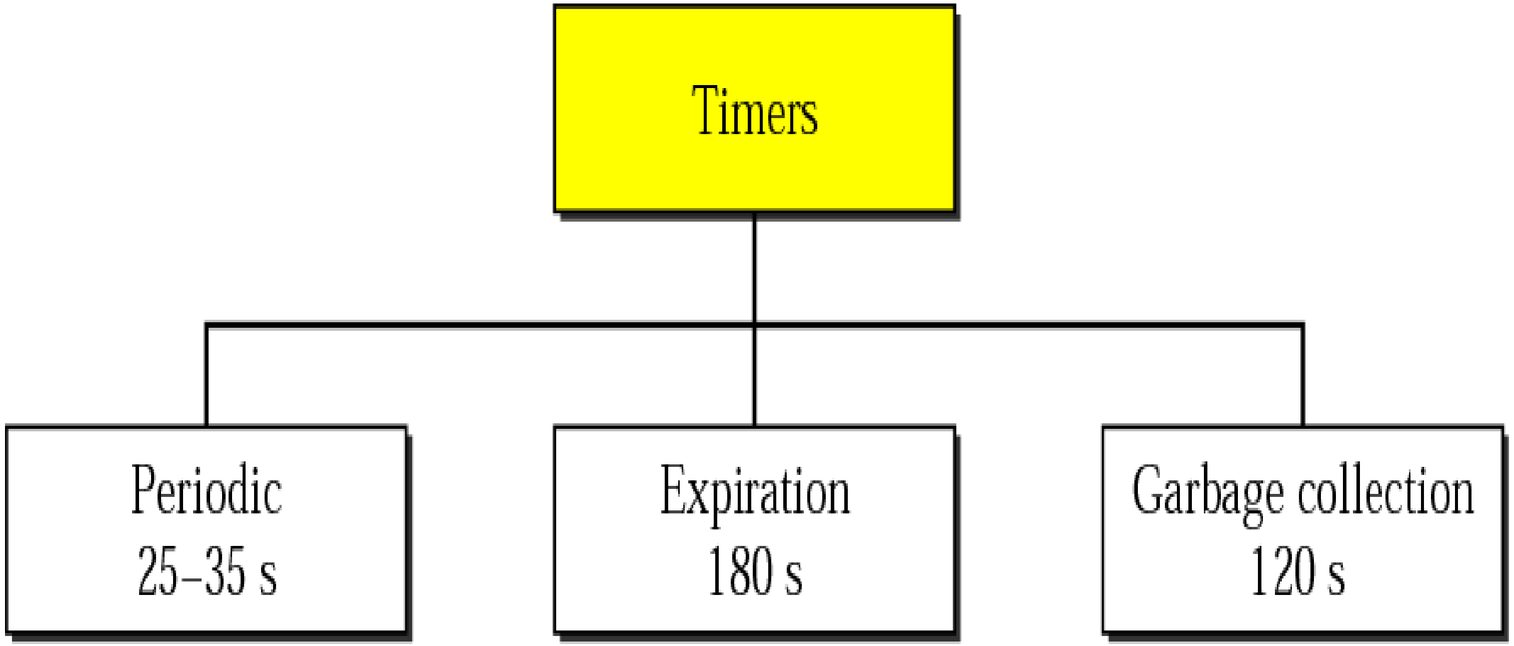


Dest.	Hop
130.10.0.0	1
130.11.0.0	1
195.2.4.0	16
195.2.5.0	16
195.2.6.0	16
205.5.5.0	2
205.5.6.0	2

Table extracted from message before incrementing

RIP – Timers

- **Periodic :** Control advertising of regular update messages (25-35 sec)
- **Expiration :** Governs validity of the route.
- **Garbage Collection :** Advertise failure of the route.



RIP – Timers

- **Periodic : <u>Control advertising of regular update messages (25-35 sec)</u>**
  - ✓ Each router has a periodic timer that is randomly set to a number between 25 and 35s.
  - ✓ It is used to prevent synchronization.
  - ✓ When zero is reached, the update message is sent, and timer is set again.

- **Expiration : <u>Governs validity of the route. (180Sec)</u>**
  - ✓ Every time an update (on a 30 sec average) is received the timer is reset.
  - ✓ If no update received within 180 sec from this timer the metric (hop count) is set to 16, which means destination is unreachable.

- **Garbage Collection : <u>Advertise failure of the route. (120)</u>**
  - ✓ A route can be advertised with a 16 metric for 120 sec before it is removed.
  - ✓ Allow neighbors to have knowledge of the invalidity of a route.

Example -

A routing table has 20 entries. It does not receive information about 5 routes for 200 s. How many timers are running at this time?

Solution:

The **21 timers** are listed below:

- Periodic timer: 1
- Expiration timer: $20 - 5 = 15$
- Garbage collection timer: 5

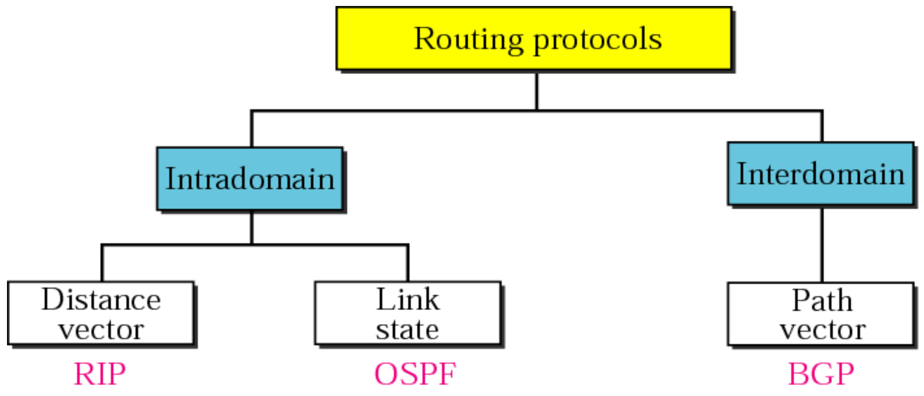
Example -

A routing table has 32 entries. It does not receive information about 6 routes for 150 s. How many timers are running at this time?

Solution:

The **33 timers** are listed below:

- Periodic timer: 1
- Expiration timer: 32
- Garbage collection timer: 0



Open Shortest Path First (OSPF)

Linked state Routing

Linked state Routing

- Link state routing is a technique in which each router shares the knowledge of its neighborhood with every other router in the internetwork.

- A router broadcast its identities and cost of the directly attached links to other routers.

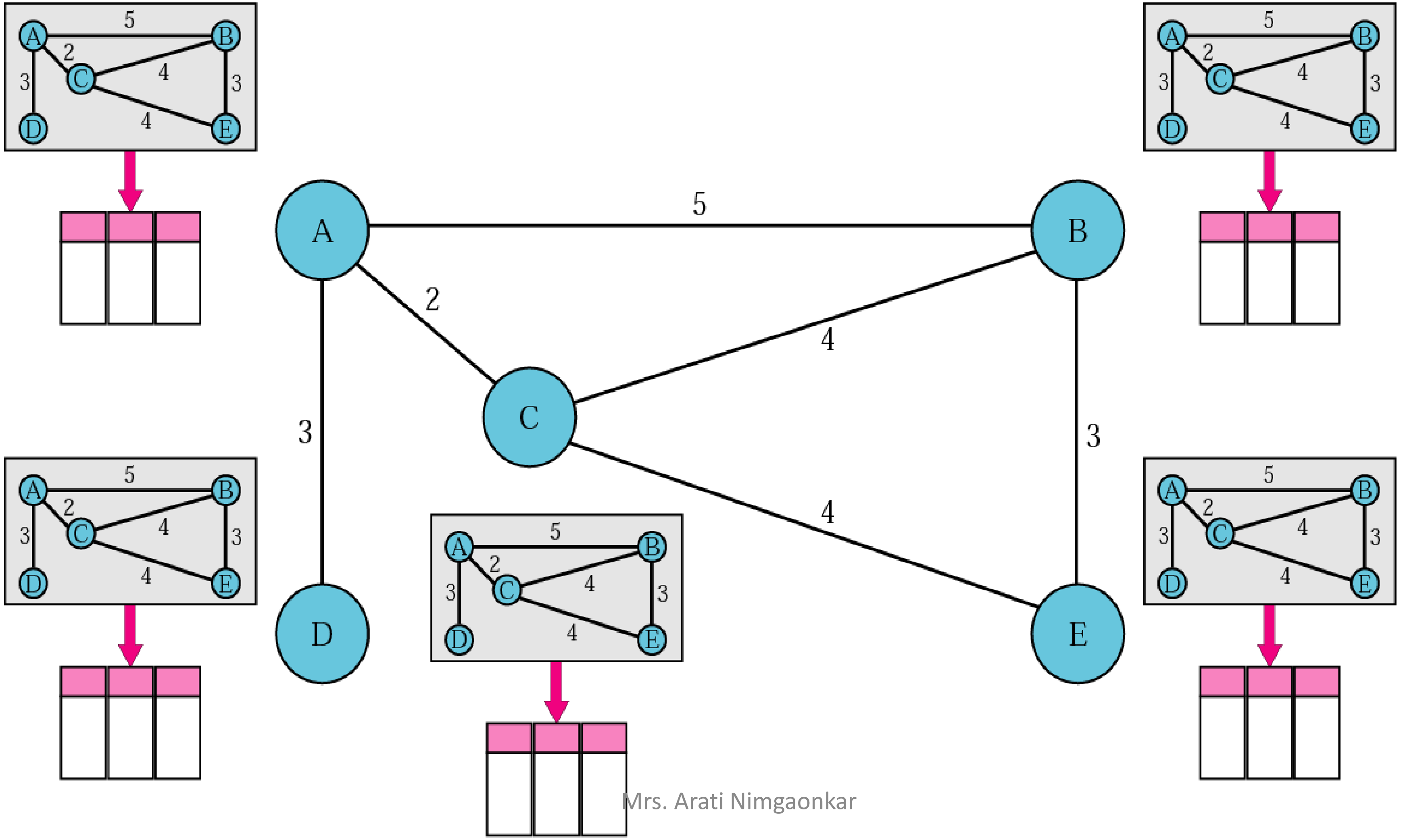
- Each node in network domain has the entire topology of the network domain with
  - List of nodes & links
  - How they are connected including the type , Cost
  - Condition of the links (up or down)

- Two persons in two different cities may have the same **map**, but each needs to take a different route to reach his destination.

- Uses Dijkstra's algorithm.

Linked state Routing

Each node uses the same topology to create a routing table, but the routing table for each node is unique



Linked state Routing

1. <u>Creation of Link state packet (LSP)</u> – creation of small packet that contains routing information.)
2. <u>Flooding LSP</u>…. (Sending LSP)
3. <u>Formation of shortest path tree</u> (Dijkstra's algorithm)
4. <u>Calculation</u> of a routing table based on shortest path tree

Linked state Routing –
Creation of Link state packet (LSP)

- Link state packet is created by each and every node in the network.

- LSP contains information such as,
  - Node identity
  - List of link

  } Used to make topology
  - Sequence number – facilitates flooding & distinguishes new & old LSP.
  - Age (in terms of time) – prevents old LSPs from remaining in the domain for long time.

- LSP are generated in 2 cases
  - When change in topology
  - On periodic basis

Linked state Routing –
Flooding LSP

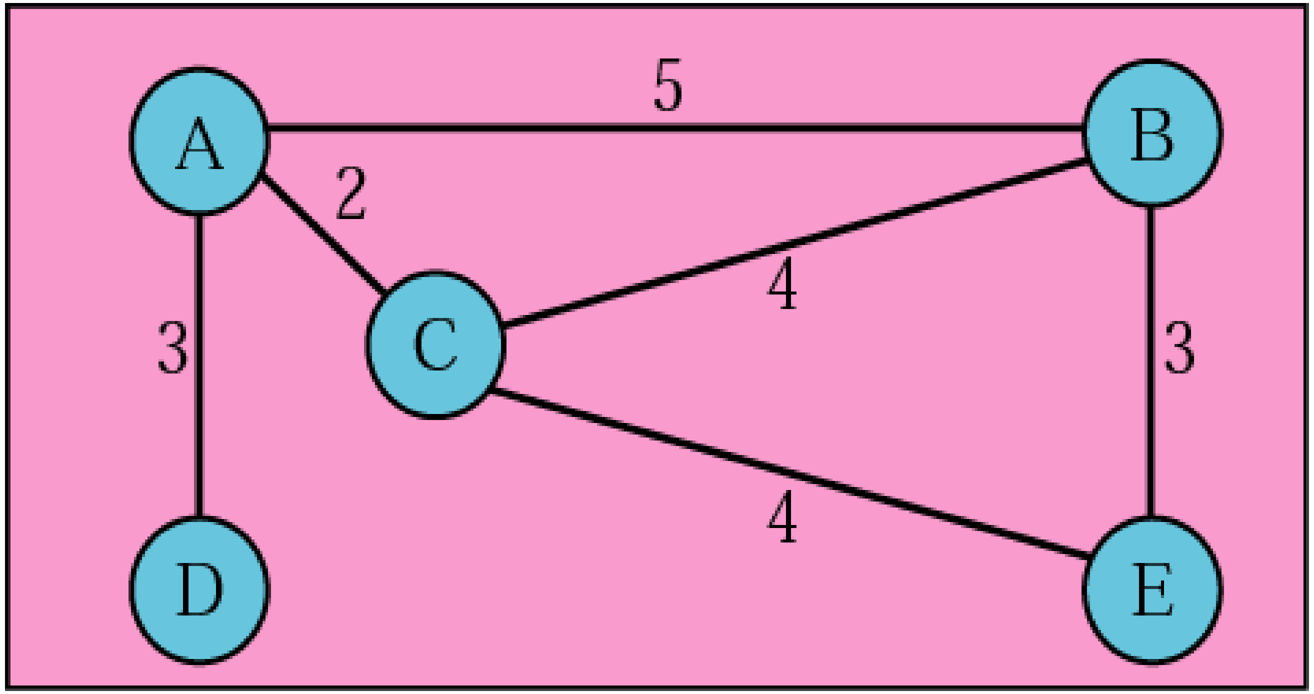
- Creation of LSP
- **Flooding LSP**
- Dijkstra's algorithm
- Calculate routing table

- After node has prepared LSP, it must be sent to all other nodes.
- Each router sends(broadcasts) LSP to every other router on the internetwork except its neighbors. This process is known as Flooding.
- Node that receives the LSP compares it with the copy it may already have.
- If received LSP is older than the one, discard it. (Sequence number & age).
- If received LSP is newer,
    - Discard the old, keep new.
    - Sends copy of it to each interface **except from which it is received.**

Linked state Routing –
Formation of shortest path

- Creation of LSP
- Flooding LSP
- **Dijkstra's algorithm**
- Calculate routing table

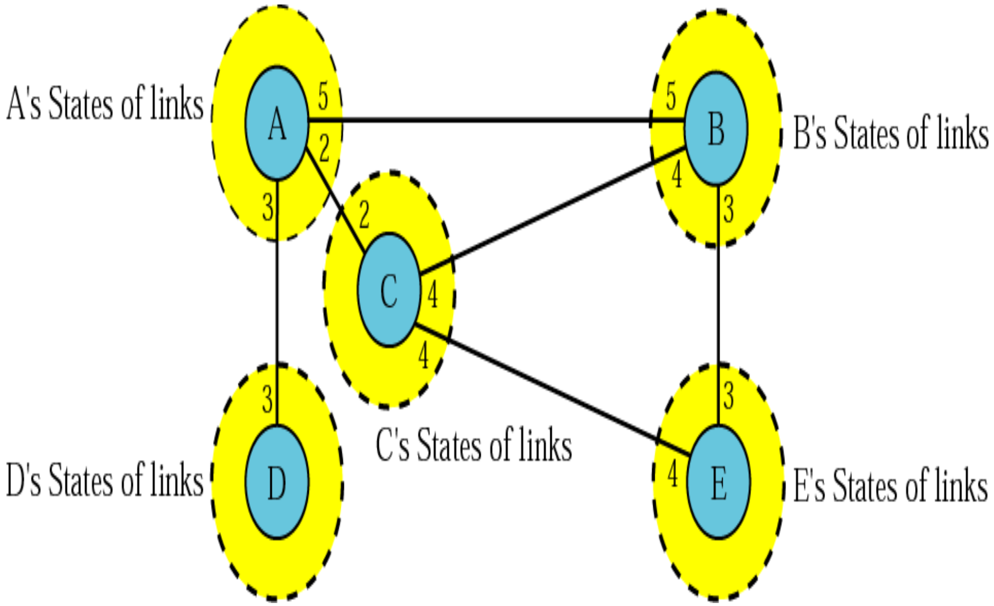
- Uses Dijkstra's algorithm.

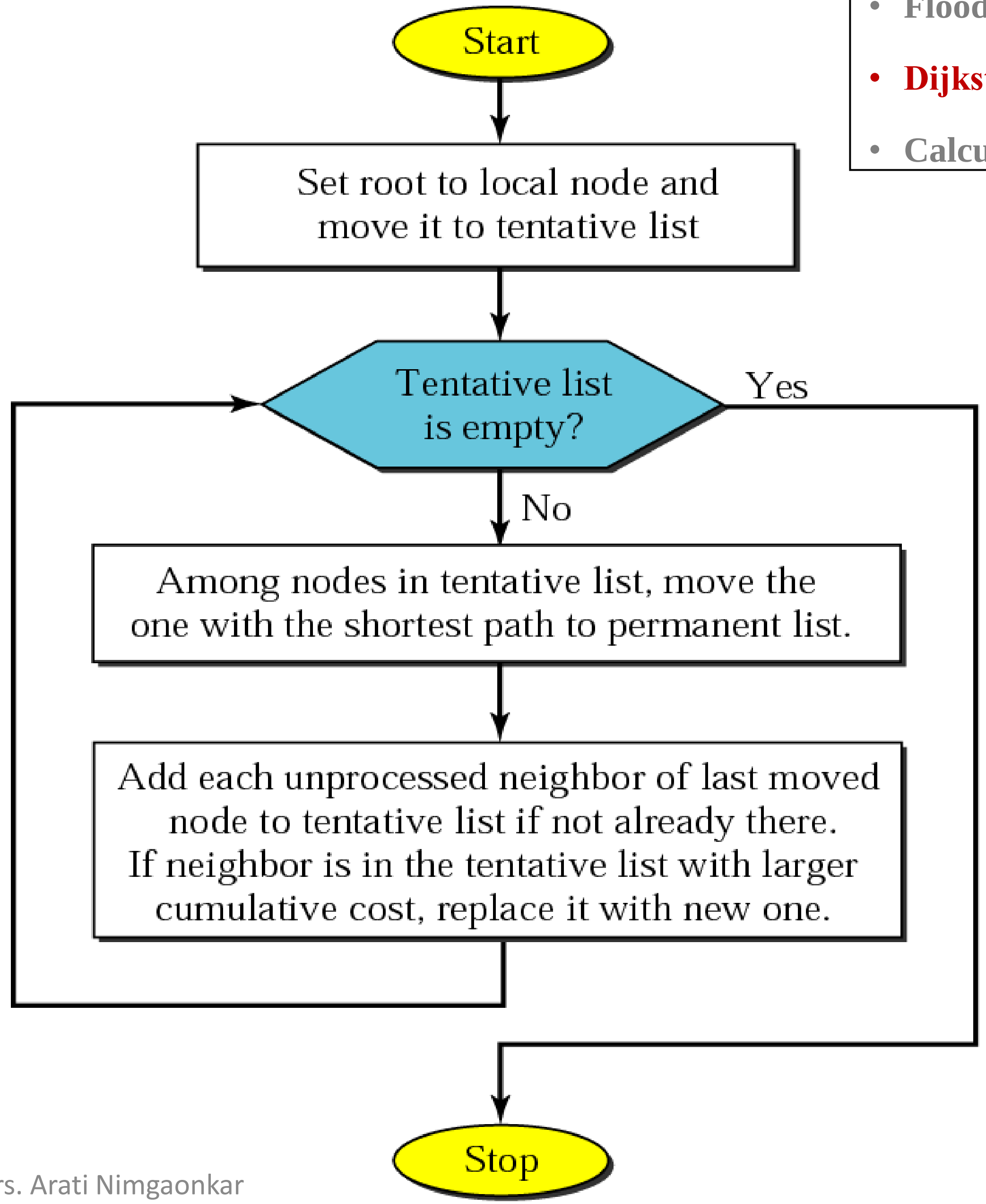


Topology

Linked state Routing –
Formation of shortest path

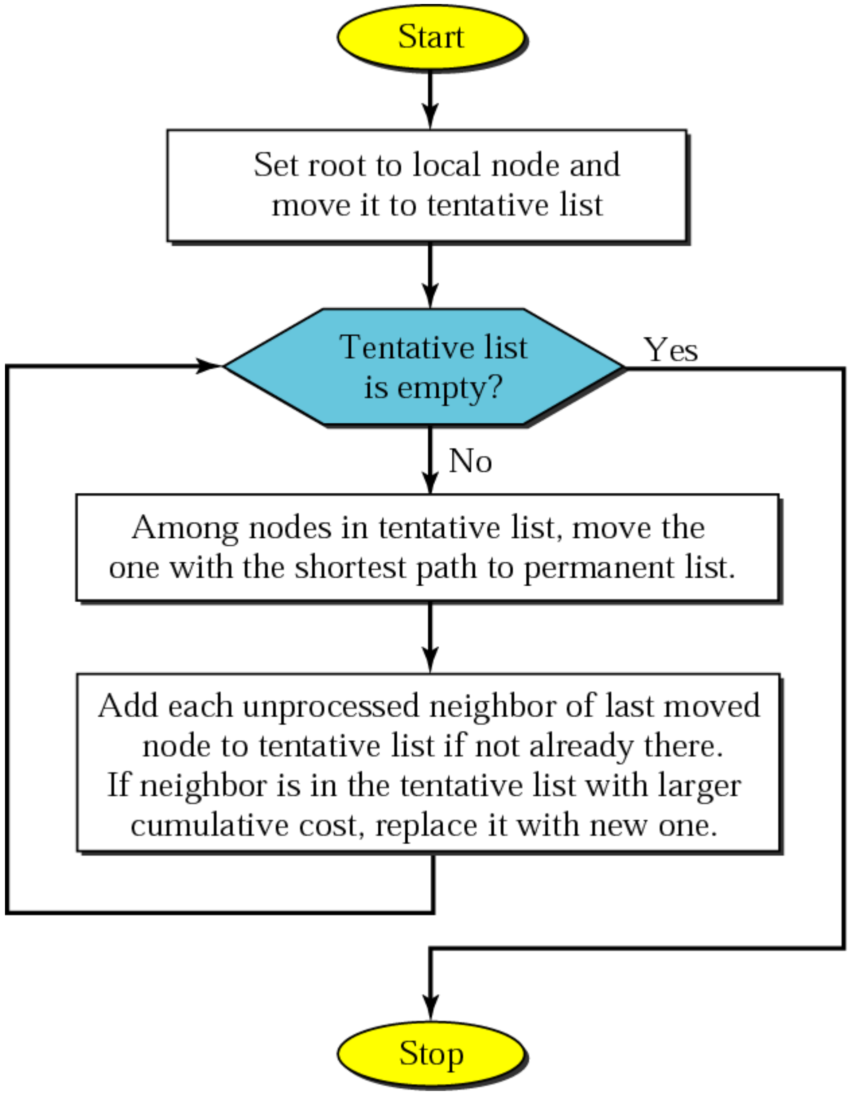
- Creation of LSP
- Flooding LSP
- **Dijkstra's algorithm**
- Calculate routing table

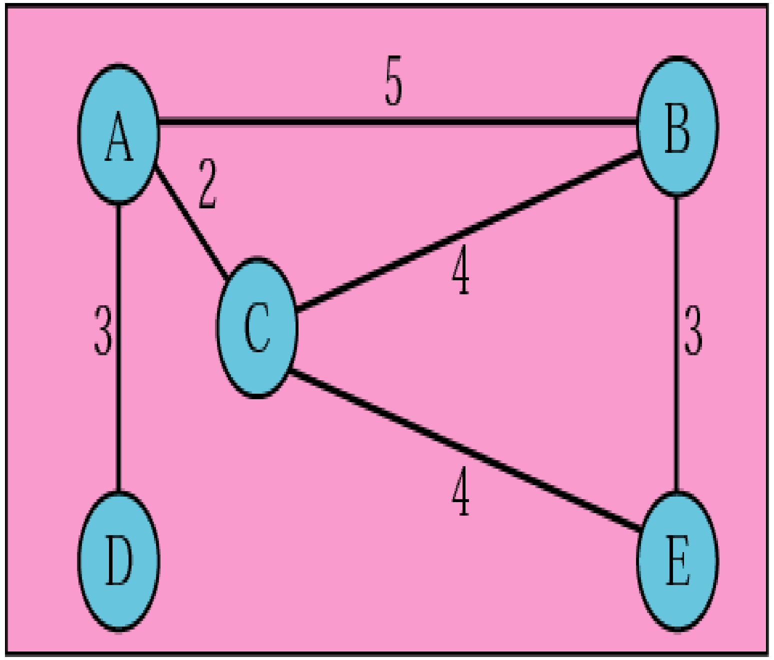




Linked state Routing –

Formation of shortest path – Example

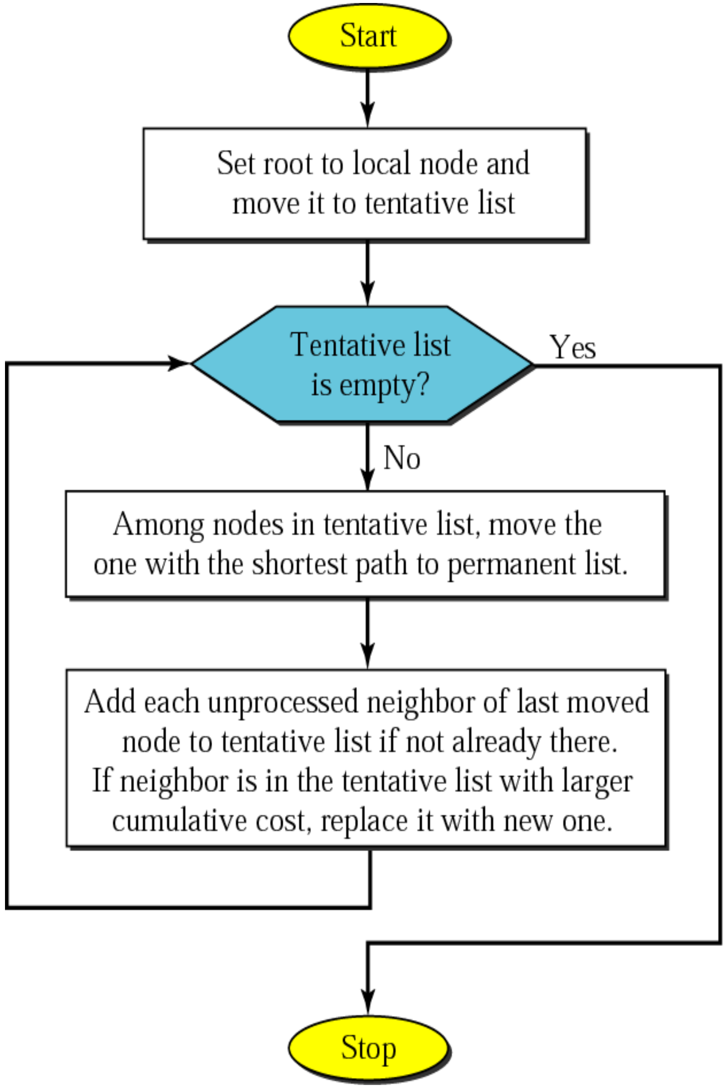


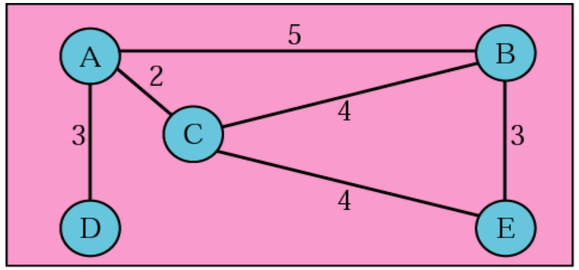


Topology

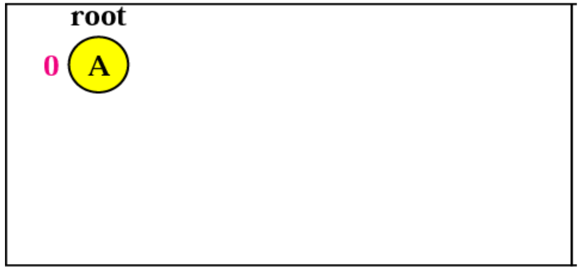
Linked state Routing –

Formation of shortest path – Example





Topology

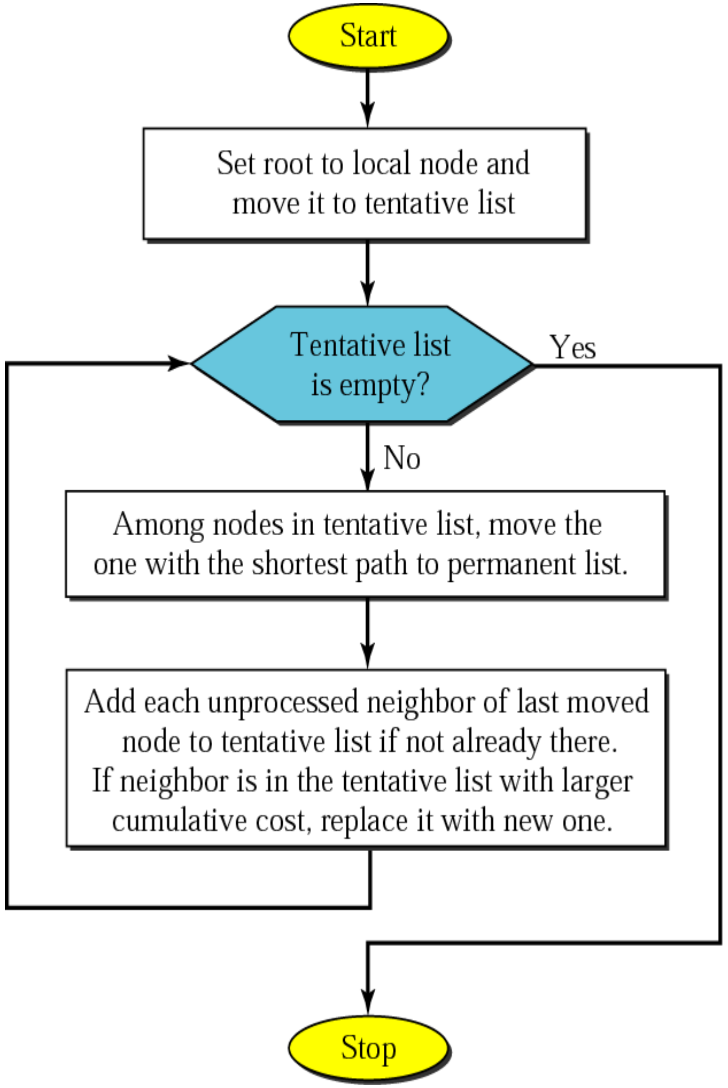


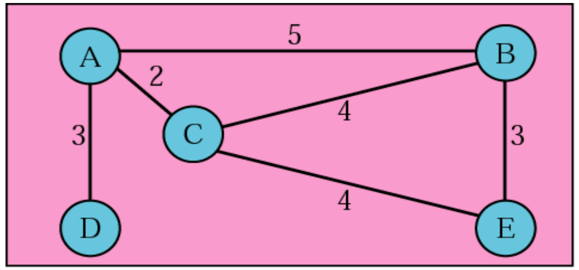
1. Set root to A and move A to tentative list

Permanent List

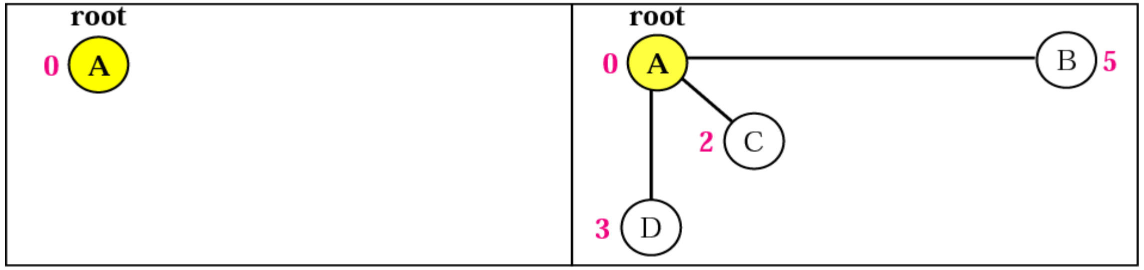
Linked state Routing –

Formation of shortest path – Example





Topology



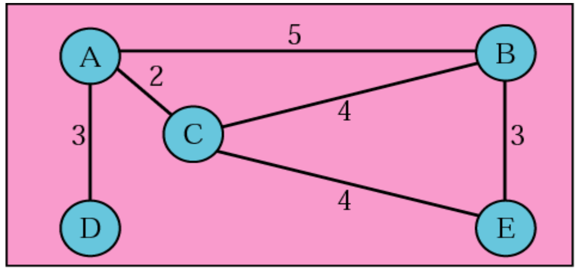
1. Set root to A and move A to tentative list

2. Move A to permanent list and add B, C, and D to tentative list

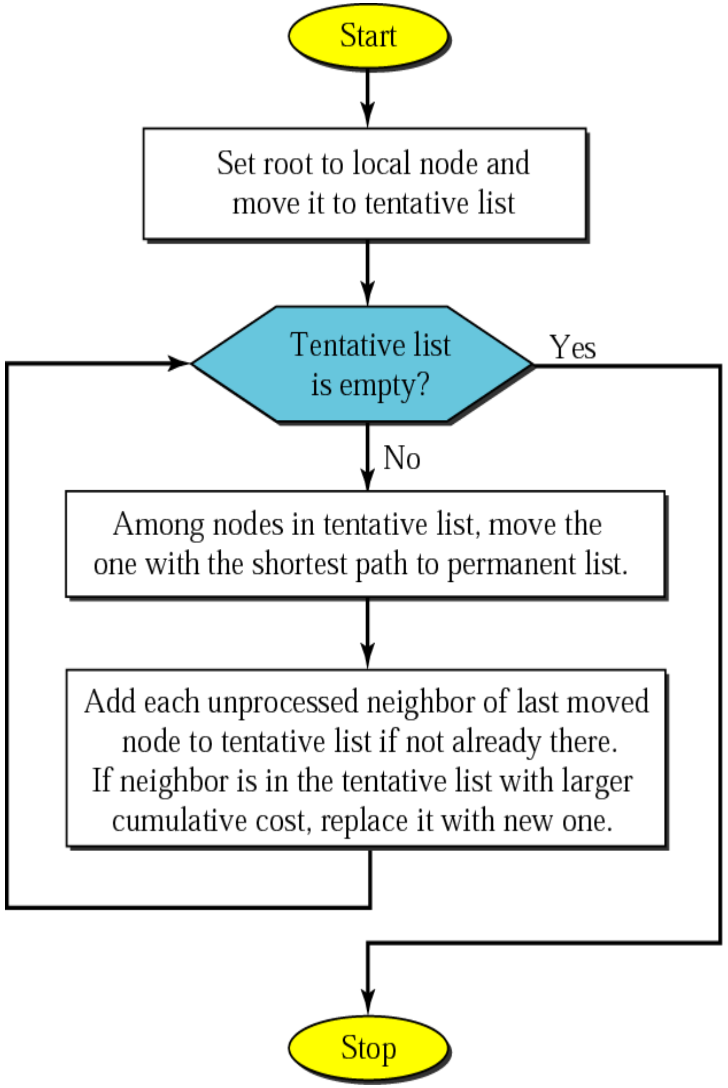
Permanent List

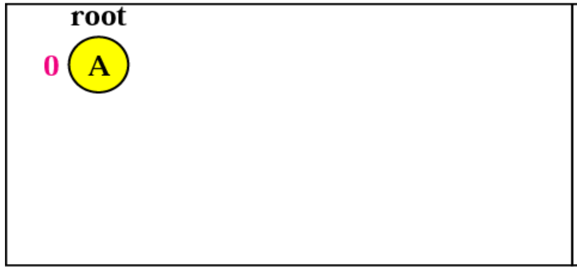
A(0),

Linked state Routing –
Formation of shortest path – Example

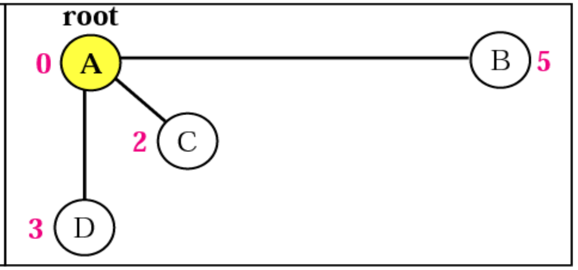


Topology

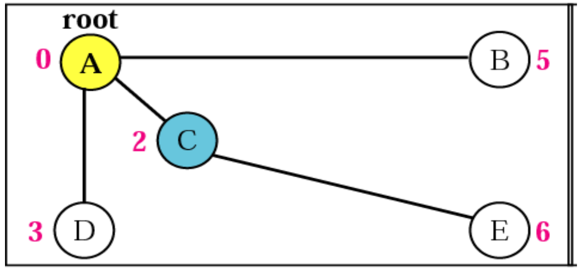




1. Set root to A and move A to tentative list



2. Move A to permanent list and add B, C, and D to tentative list



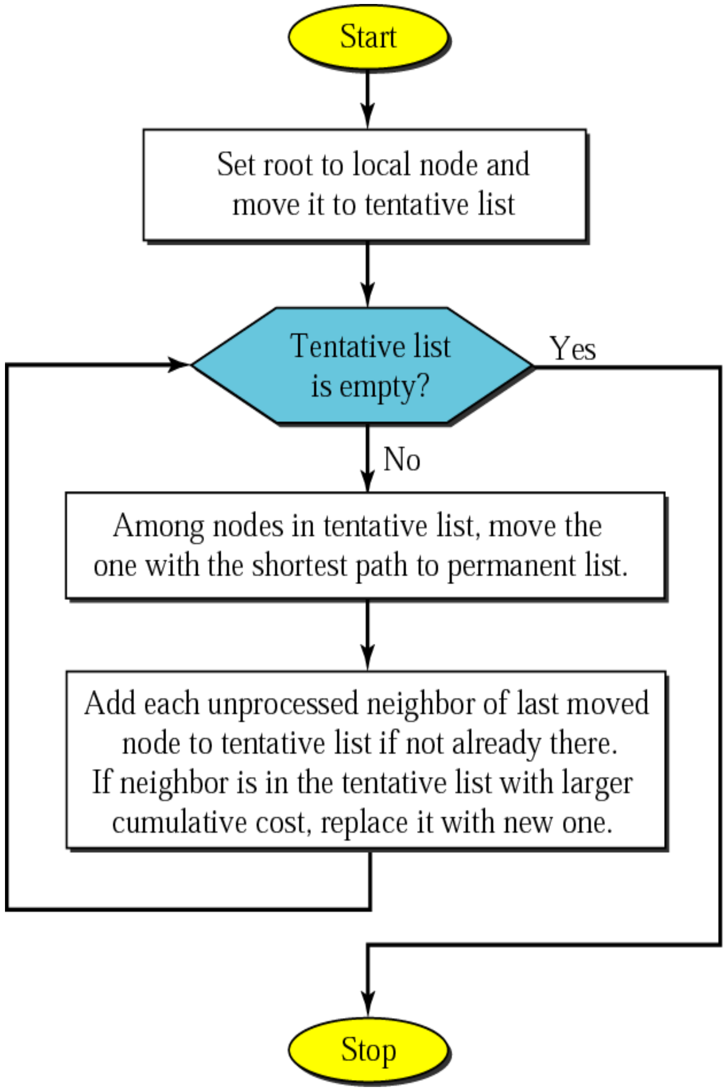
3. Move C to permanent and add E to tentative list

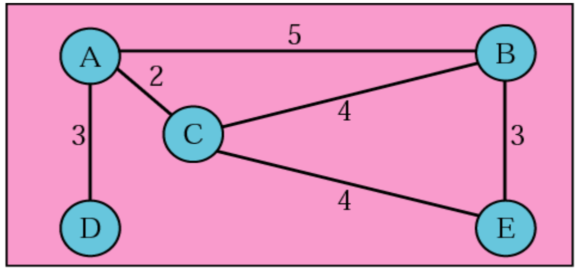
Permanent List

A(0), C(2),

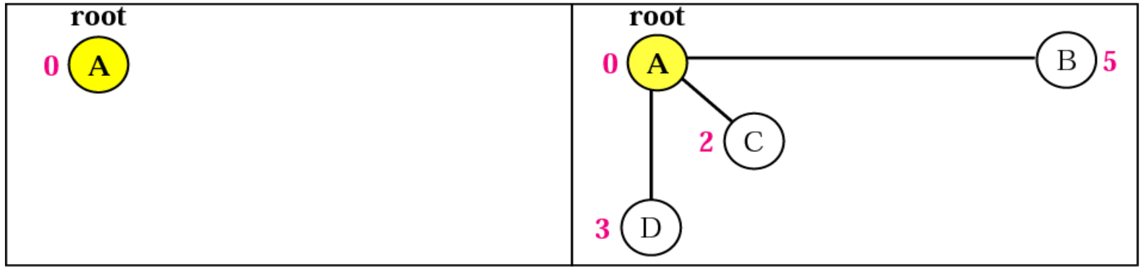
Linked state Routing –

Formation of shortest path – Example



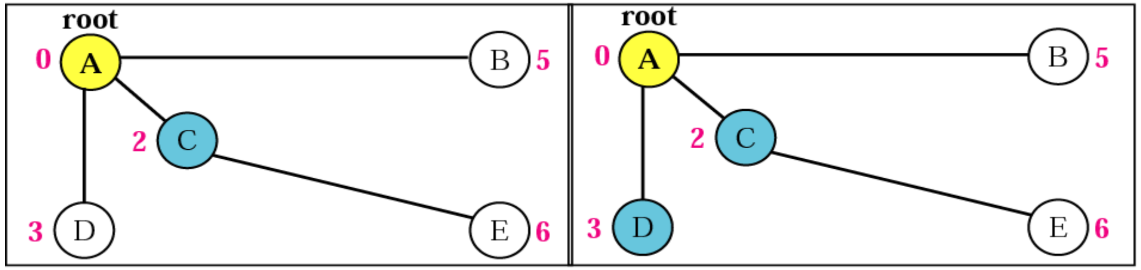


Topology



1. Set root to A and move A to tentative list

2. Move A to permanent list and add B, C, and D to tentative list



3. Move C to permanent and add E to tentative list

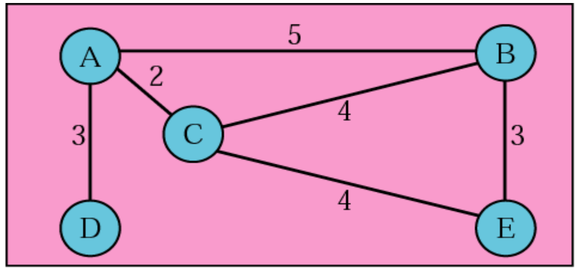
4. Move D to permanent list.

Permanent List

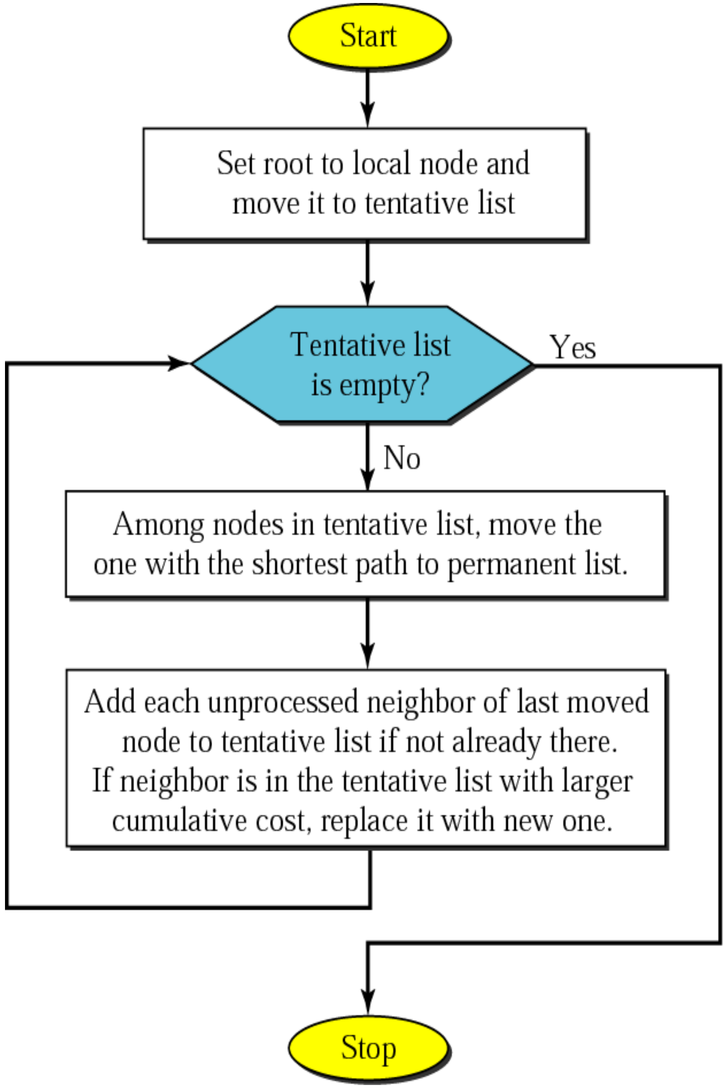
A(0), C(2), D(3),

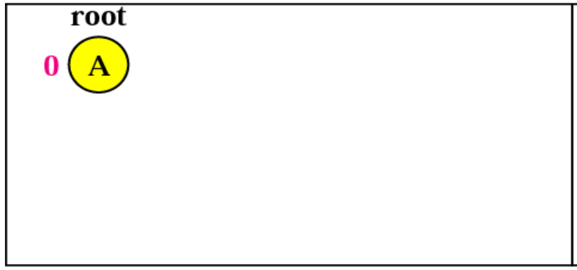
Linked state Routing –

Formation of shortest path – Example

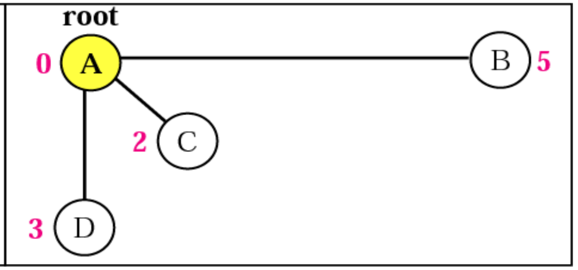


Topology

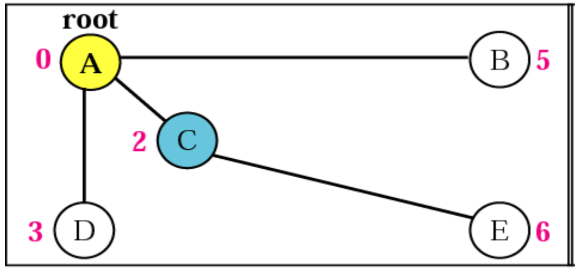




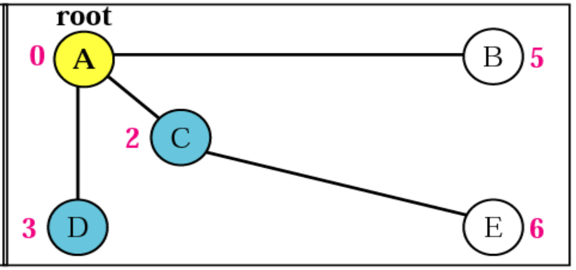
1. Set root to A and move A to tentative list



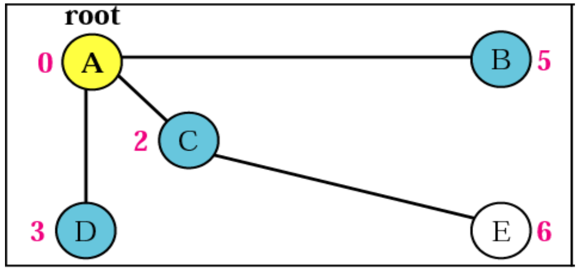
2. Move A to permanent list and add B, C, and D to tentative list



3. Move C to permanent and add E to tentative list



4. Move D to permanent list.



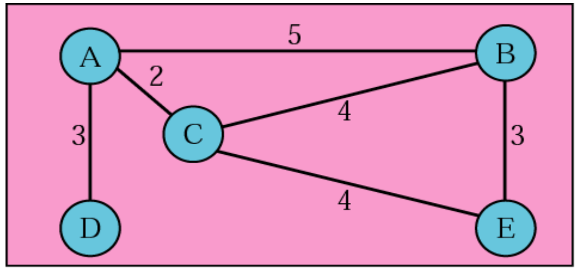
5. Move B to permanent list

Permanent List

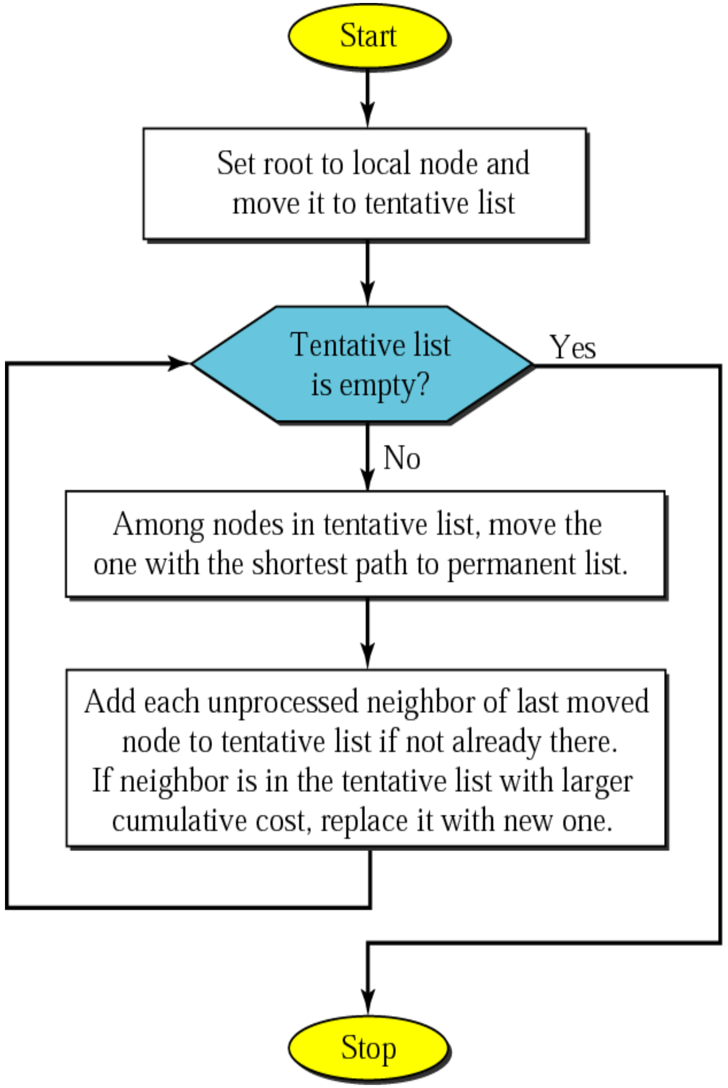
A(0), C(2), D(3), B(5)

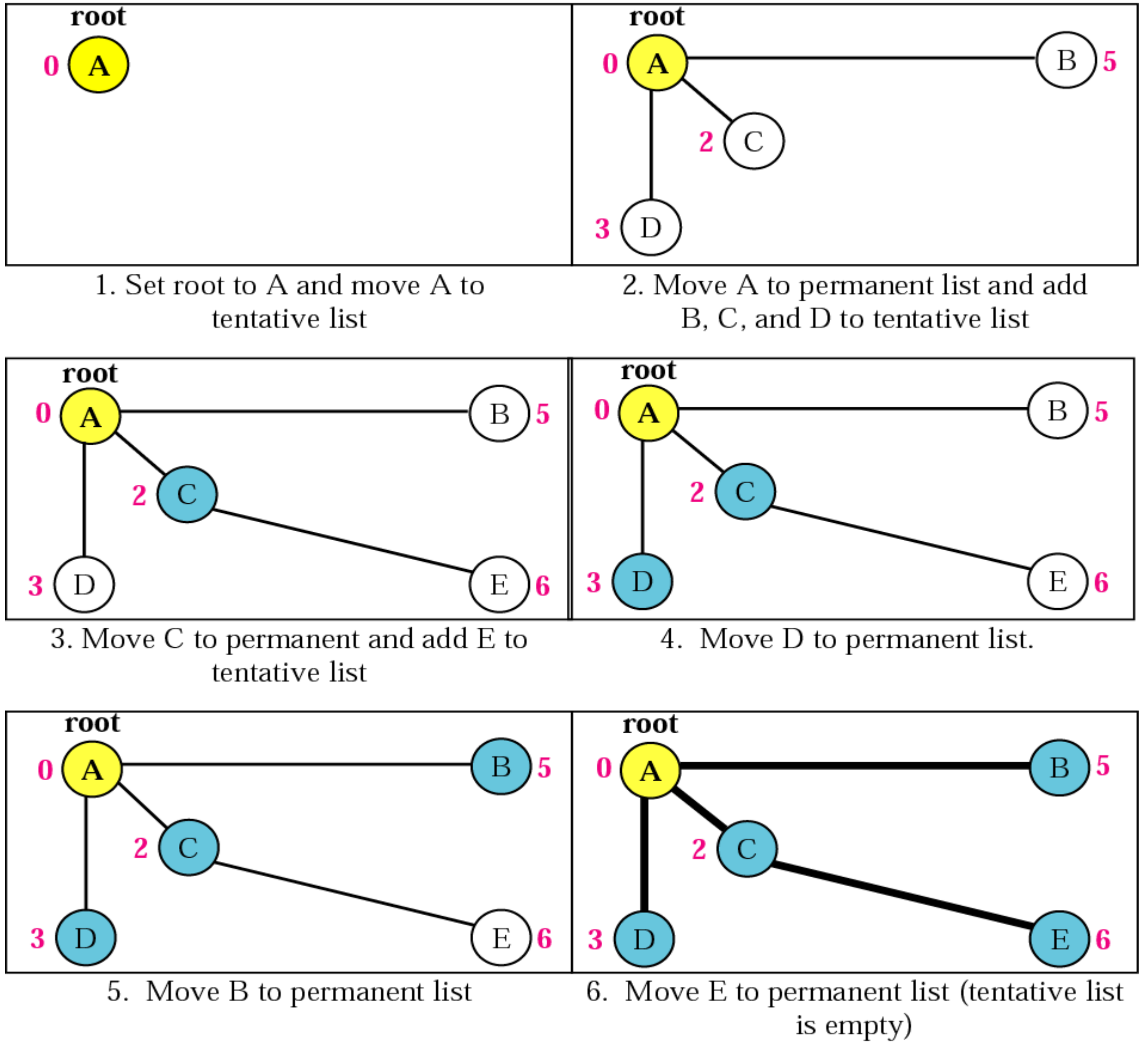
Linked state Routing –

Formation of shortest path – Example



Topology





Permanent List

A(0), C(2) ,D(3), B(5), E(6)

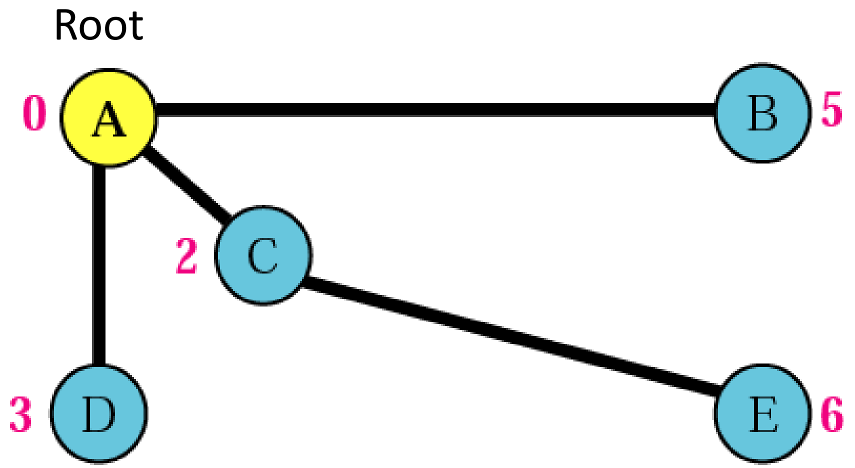
Linked state Routing –
Calculating Routing Table –

- Creation of LSP
- Flooding LSP
- **Dijkstra's algorithm**
- Calculate routing table

- On the basis of this tree, routing table is populated..

Routing table for node A

Node	*Cost*	*Next Router*
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

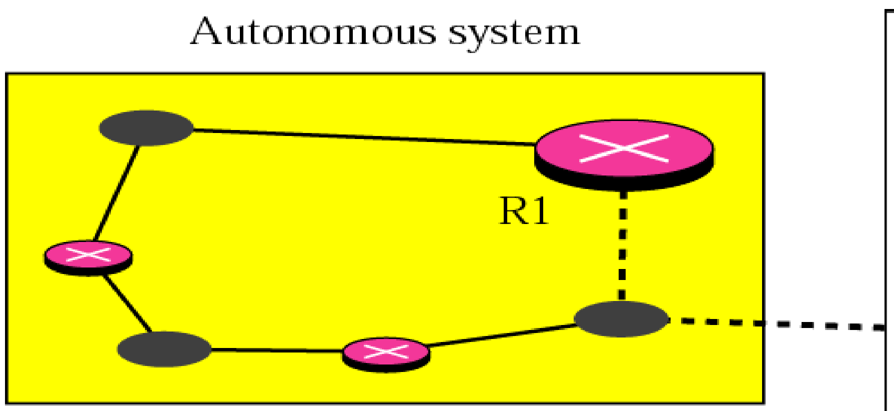


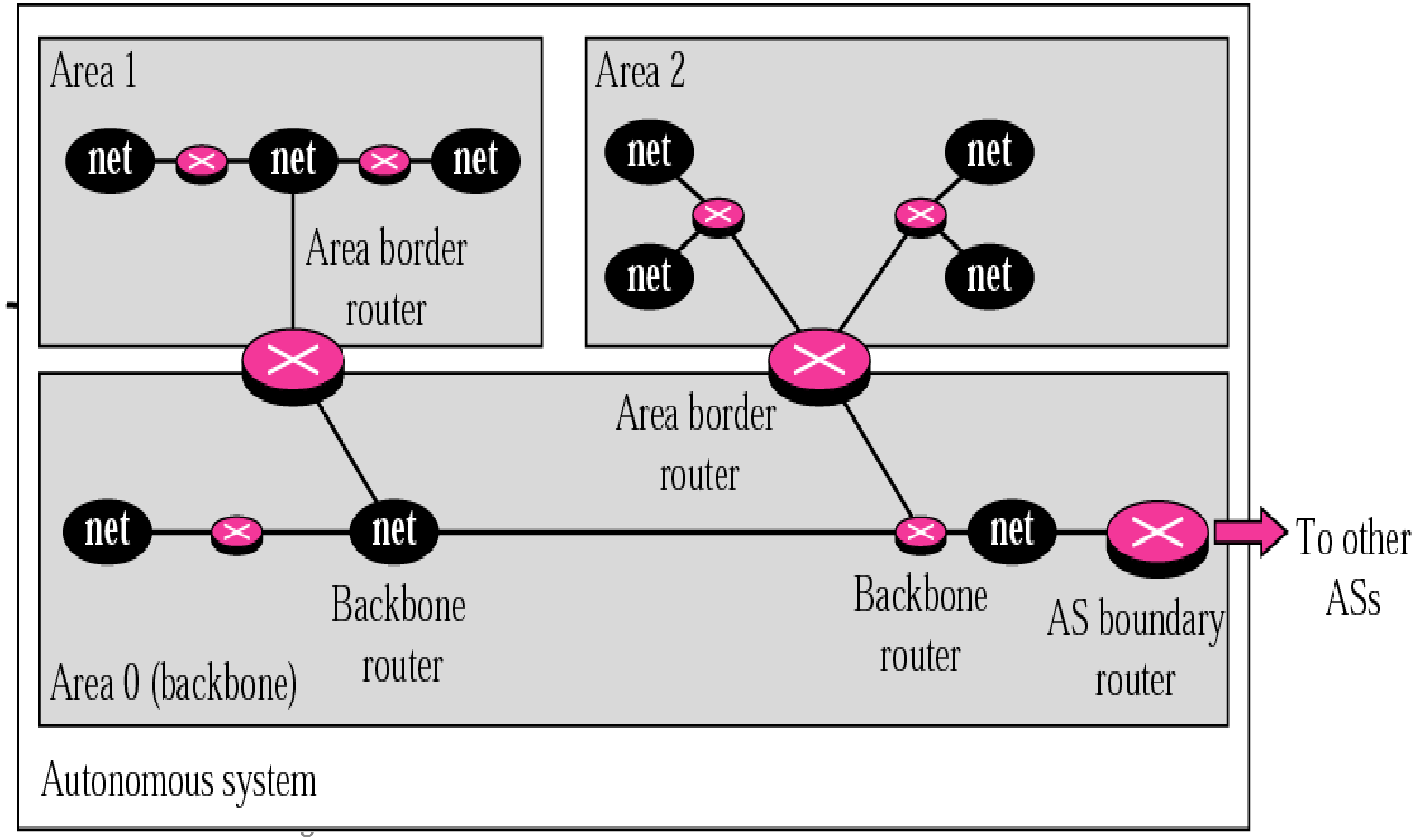
OSPF – Open Shortest Path First

- Intra domain routing protocol used inside an autonomous system
- It is based on Linked State routing.

OSPF – Autonomous System - Area

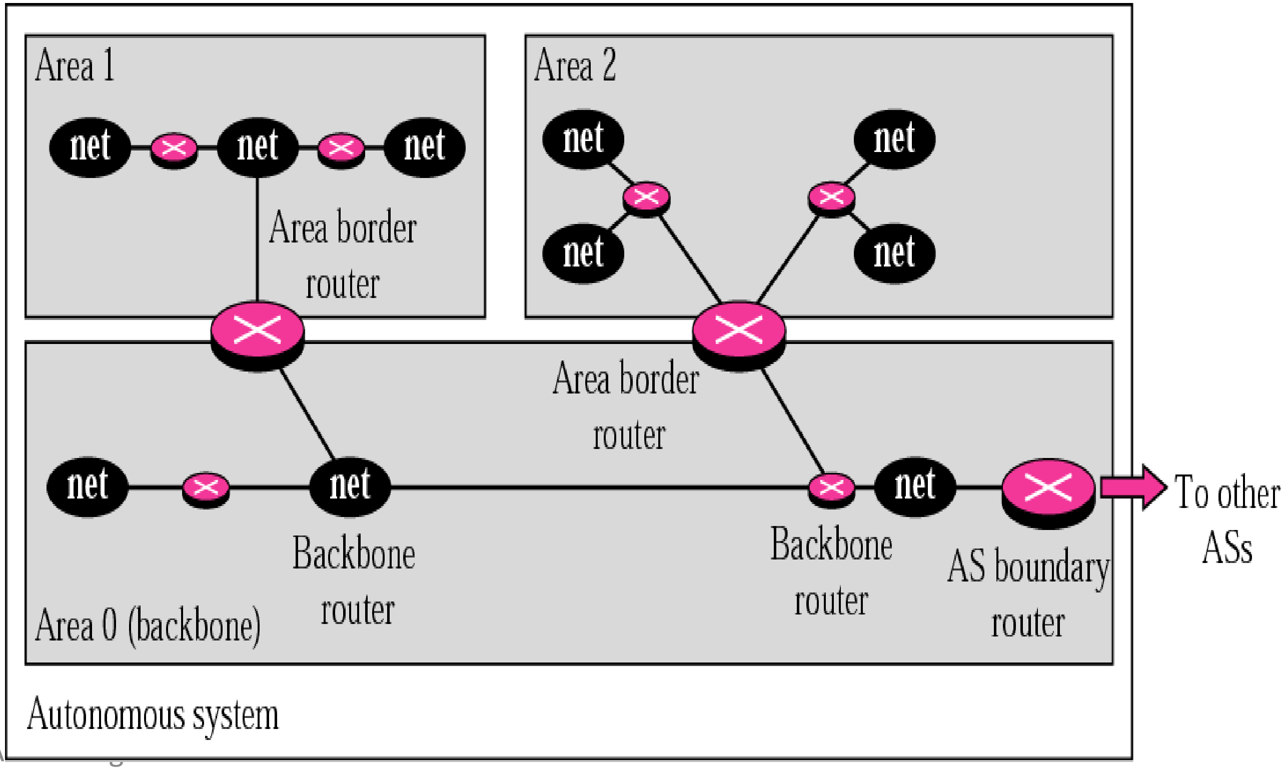
- OSPF divides autonomous system into areas.
- Area is a collection of networks, hosts and routers.





OSPF – Autonomous System - Area

- Routers inside an Area populate the area with routing information
- **Backbone area** serves as a primary area.
- Routers inside backbone area are called **backbone routers**
- **Area bounder router** summarizes all the information and send it to other AS.

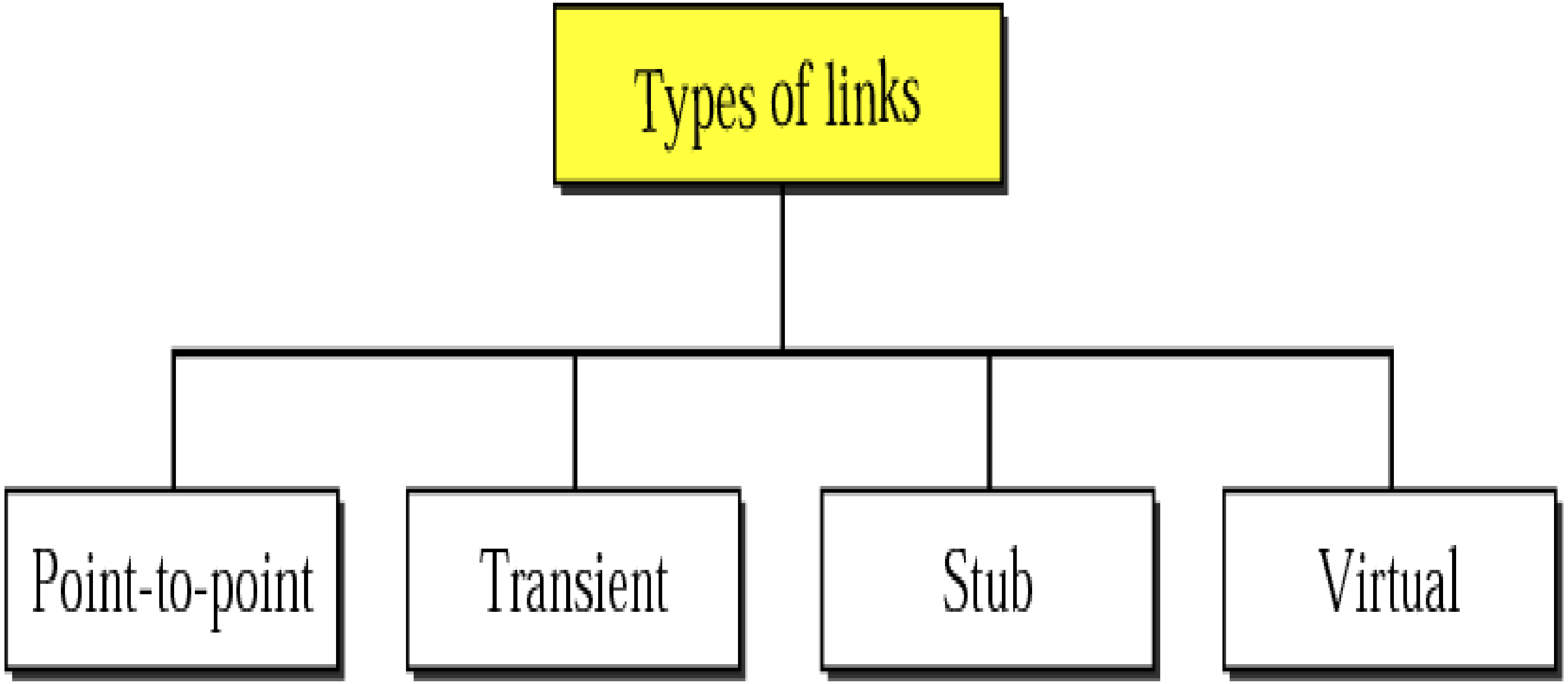


OSPF – Metrics

- Allows administrator (vendor) to assign a cost, called metric to each route.
- Metric (cost) is assigned to every route. (using formula Cost = Reference bandwidth / Interface bandwidth in bps.)
- This can be based on services opted by the packet. **(minimum delay, maximum throughput)**
- Router can have multiple routing table based on the type of the service

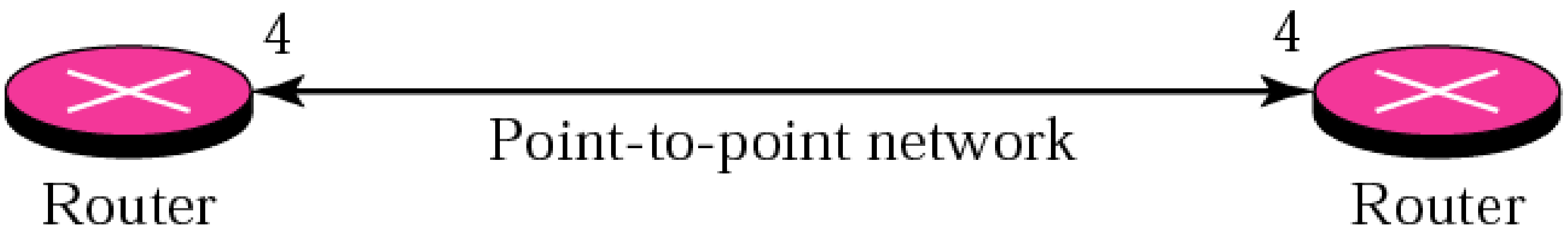
OSPF – Types of link

- Connection is called as link.



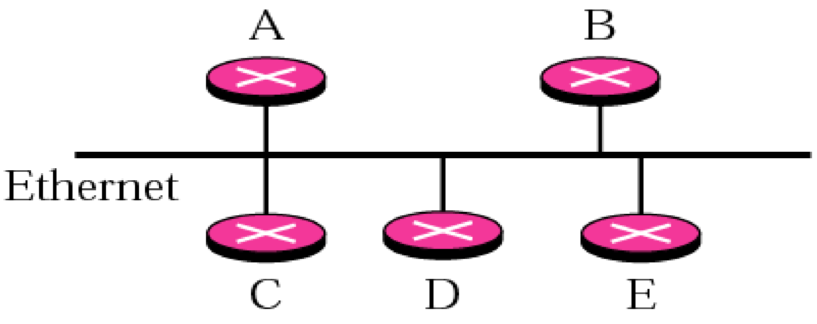
OSPF – Types of link – Point to point

- Connects two routers without any host or router in between.
- Routers are represented by nodes and link is represented by a bidirectional edge.
- For e.g. two routers connected by a telephone line.

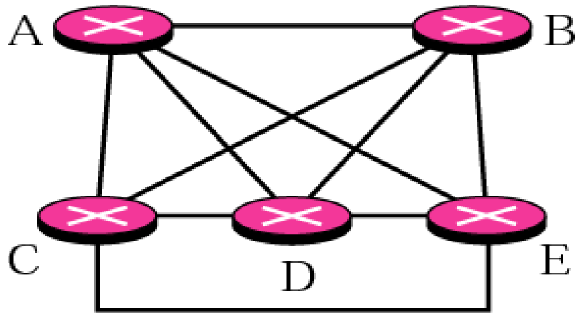


OSPF – Types of link – Transient link

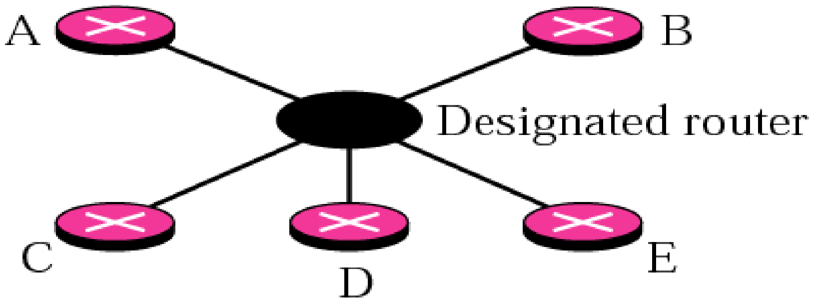
- A network with several routers attached to it
- Data can enter through any router and leave through any router.
- One router is selected as designated router
- Eg: LAN’s and WAN’s



a. Transient network



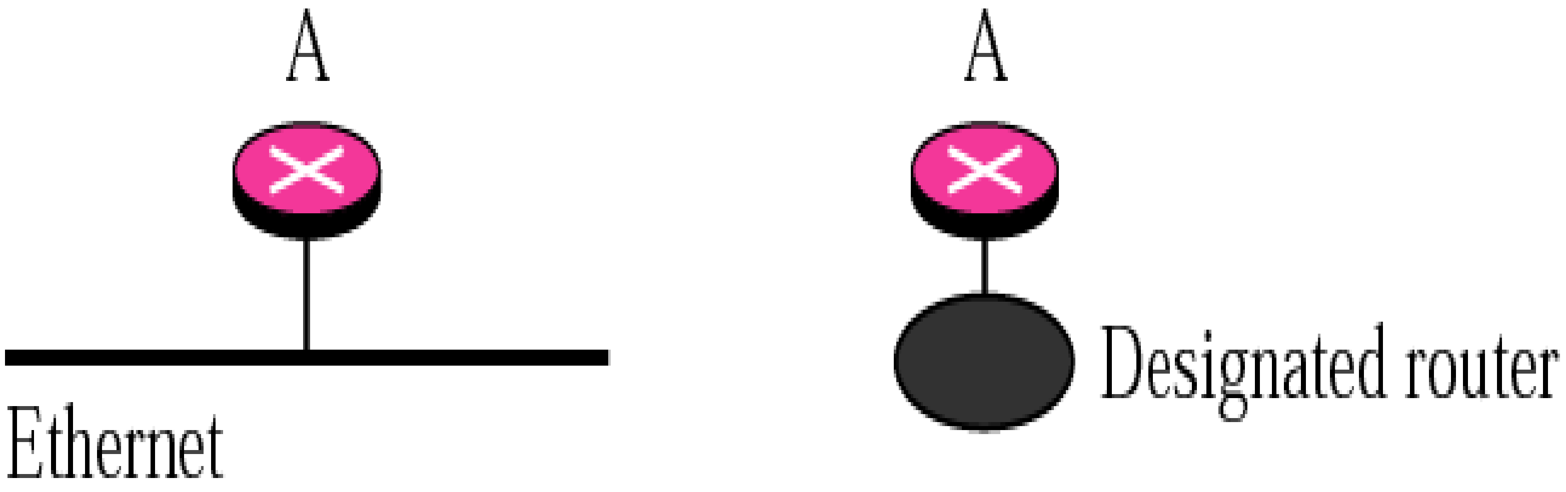
b. Unrealistic representation



c. Realistic representation

OSPF – Types of link – Stub link

- A network that connects to only one router
- Unidirectional link from the router to the network

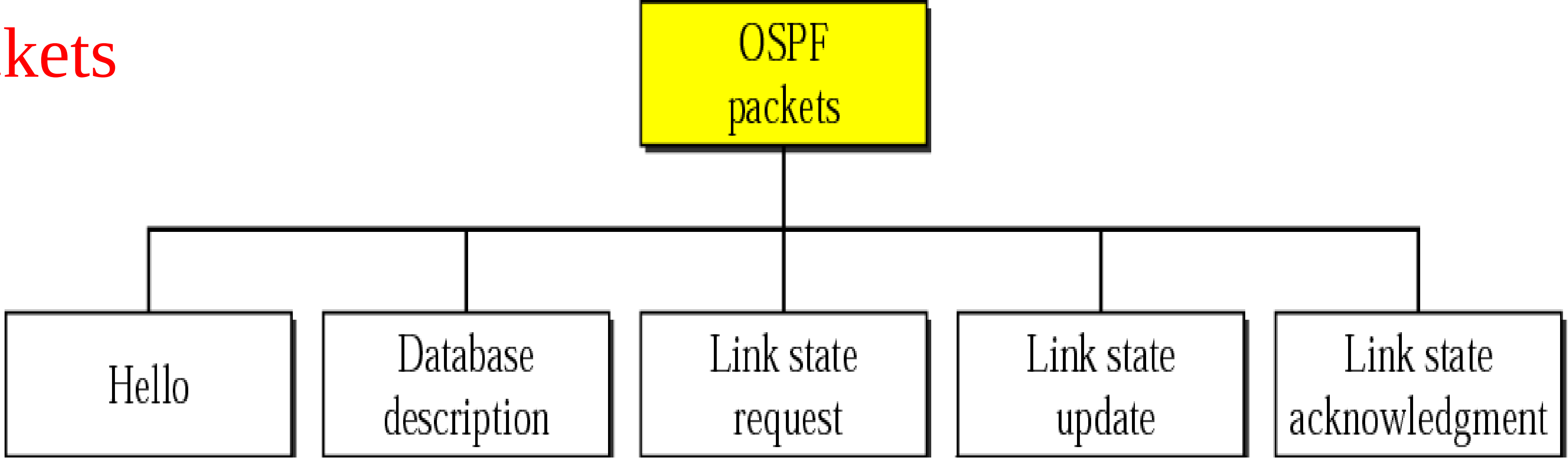


a. Stub network b. Representation

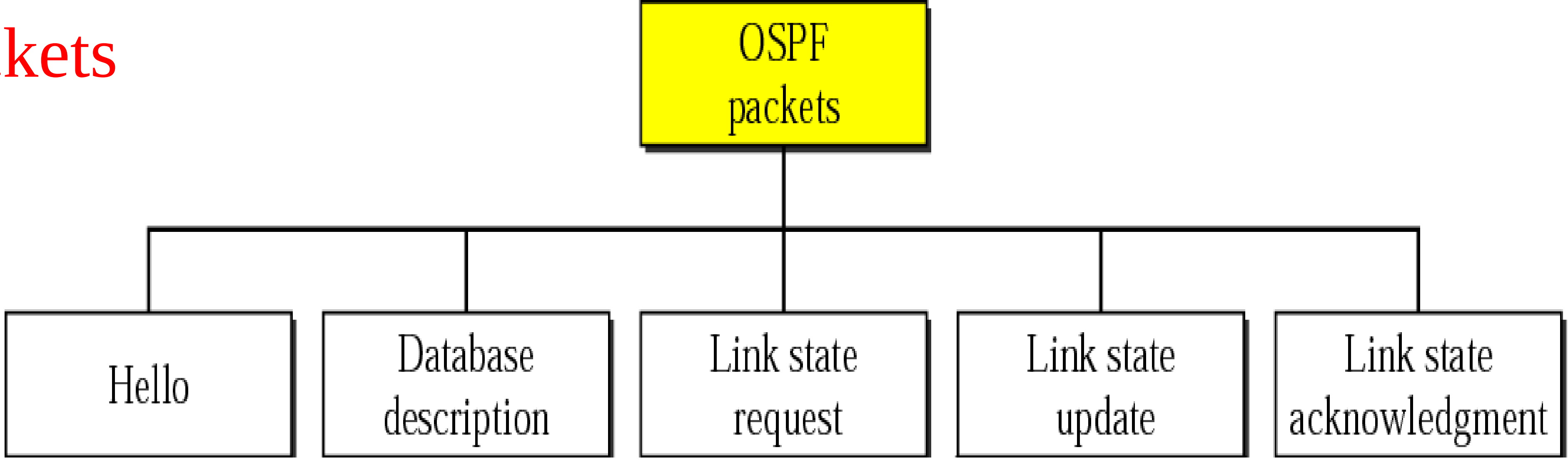
OSPF – Types of link – Virtual link

- When the link between two routers is broken, the administration, may create a virtual link between them.
- Alternative path , may pass through several routers.

Types of OSPF packets



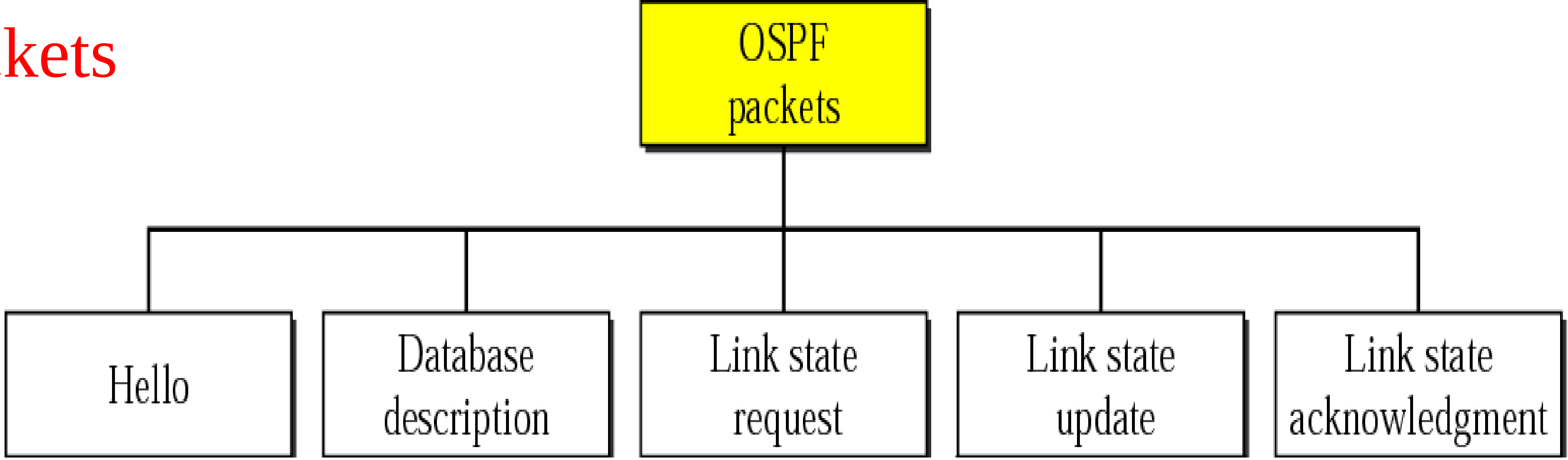
Types of OSPF packets



The Hello Packet :

- The hello packets are sent over a period of time on all interfaces for the purpose of establishing and maintaining neighbor relationships.
- Hello packets are multicast on the networks having multicast capability, which enables discovery of neighboring routers dynamically.

Types of OSPF packets

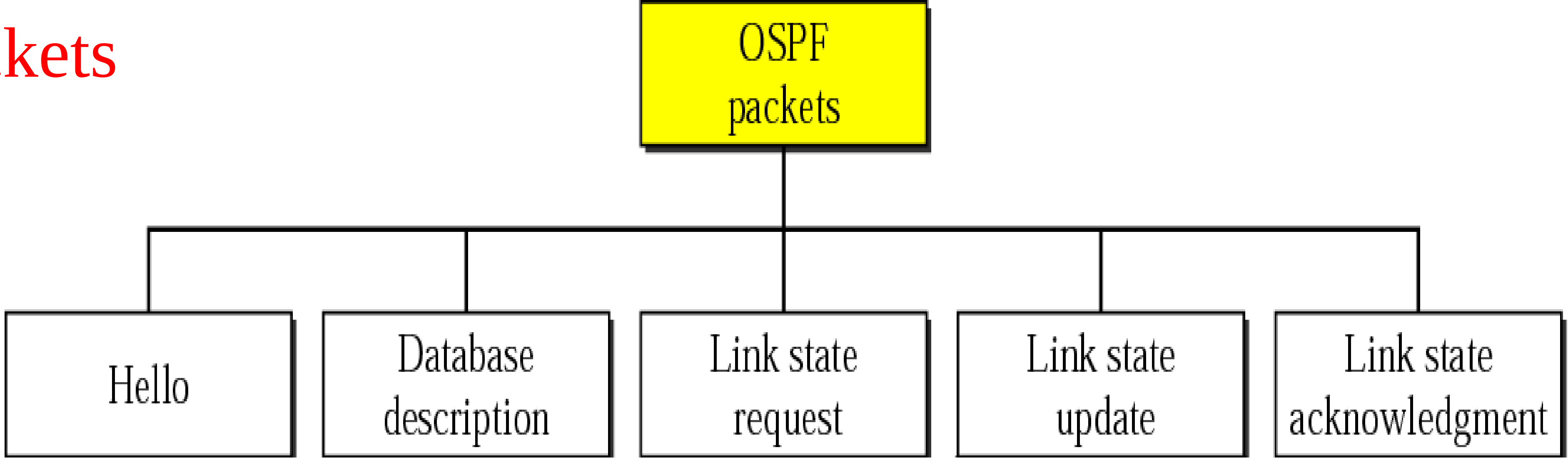


The Database Description Packet :

- These packets describe topological database contents.

- The database may be described by using multiple packets.

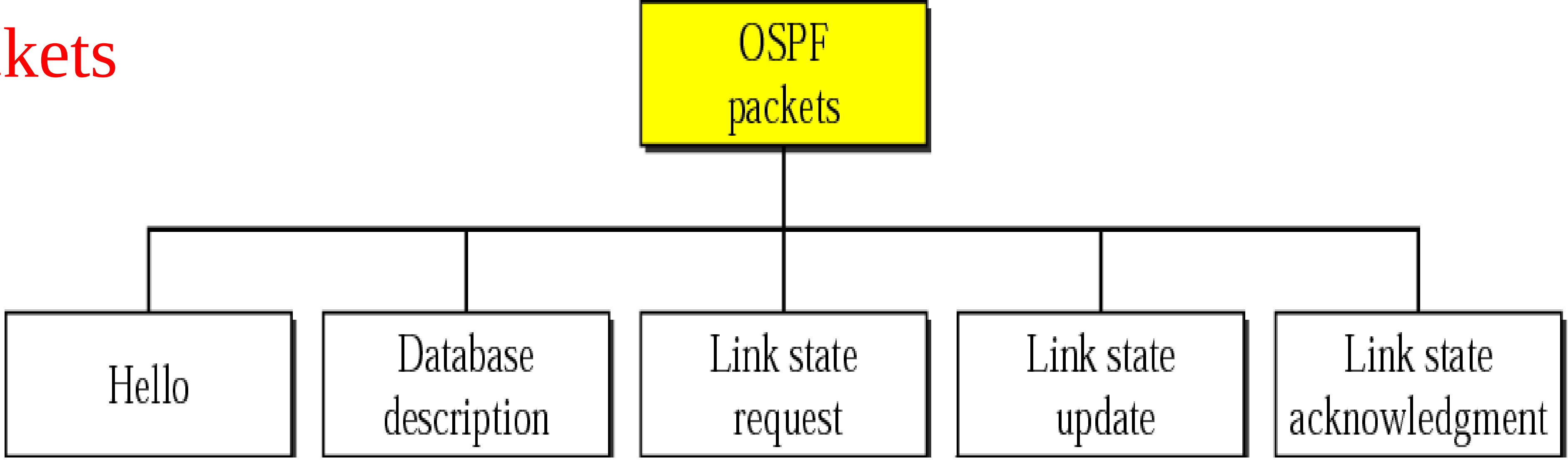
Types of OSPF packets



The Link State Request packet :

- A router may find the parts of its topological database are out of date, after database description package exchange with a neighboring router.
- The Link State Request packet is utilized for requesting the pieces of the neighbor's database which are more up to date.

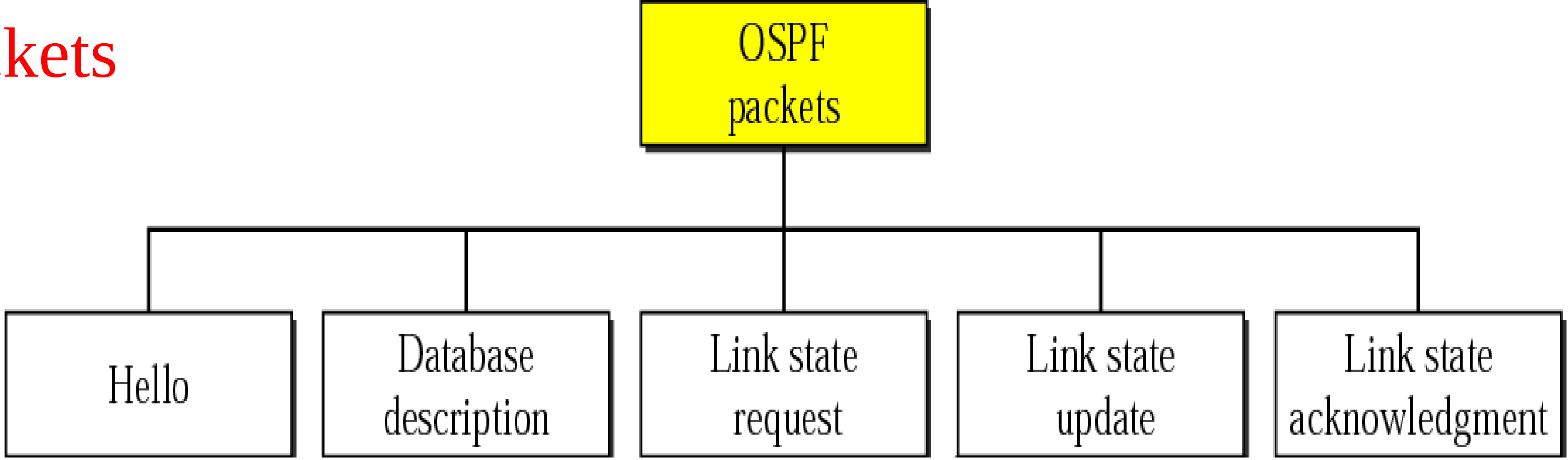
Types of OSPF packets



The Link State Update packets:

- The flooding of link state advertisements is implemented by these packets.
- A collection of link state advertisements are carried by each Link Statement Update packet, one hop further from its origin.

Types of OSPF packets

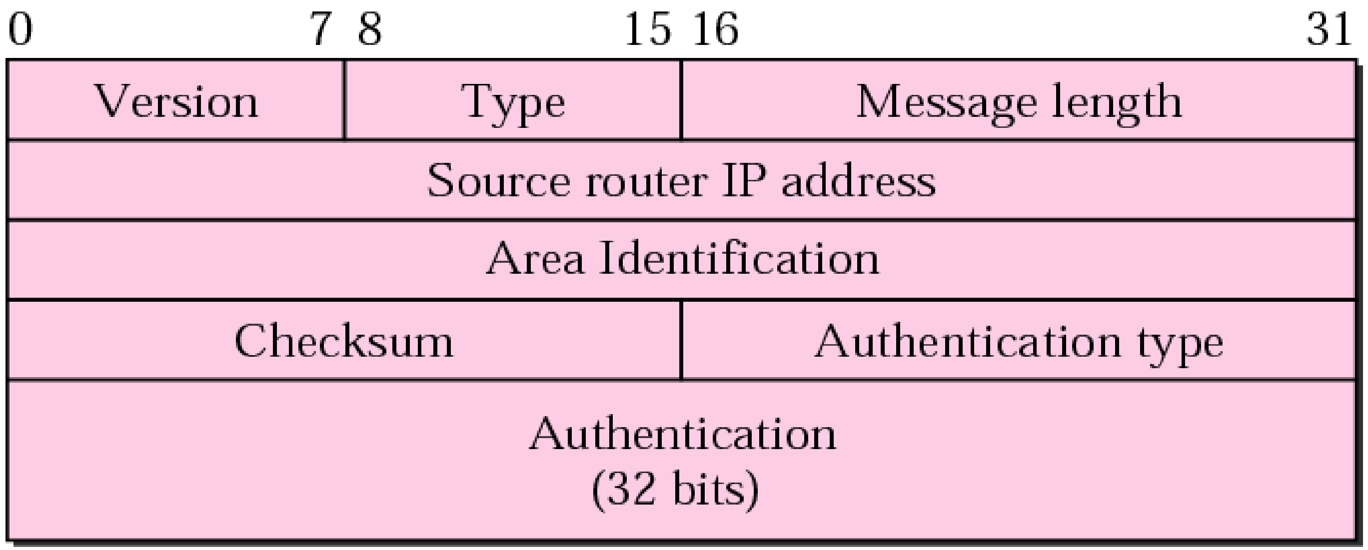


The Link State Acknowledge packets:

- The reliability of flooding link state advertisement is made by explicitly acknowledging flooded advertisements.
- The accomplishment of this acknowledgement is done through the sending and receiving of Link Sate Acknowledgement packets.
- A single Link State Acknowledgement packet is used to acknowledge the multiple link state advertisements.

OSPF common header

Total length is 160 bits



- **Version :** 8-bit The OSPF version number. version 2 of the protocol.
- **Type:** 8- bit defines the type of packet
- **Message length:** 16- bit defines length of total message including header
- **Source router IP address:** 32-bit defines IP address of the sending router
- **Area Identification:** 32-bit defines area within which routing takes place
- **Checksum:** Used for error detection on packet excluding authentication type & data
- **Authentication type:** 16-bit defines authentication protocol used in area
- **Authentication Data:** 64-bit used for authentication of the message, as needed.

Authentication Type: Indicates the type of authentication used for this message:

***Authentication Type* Value**	**OSPF Authentication Type**
0	No Authentication
1	Simple Password Authentication
2	Cryptographic Authentication

Unicast Routing Protocols

BGP

Border Gateway Protocol

(Path Vector Routing)

BGP - Border Gateway Protocol

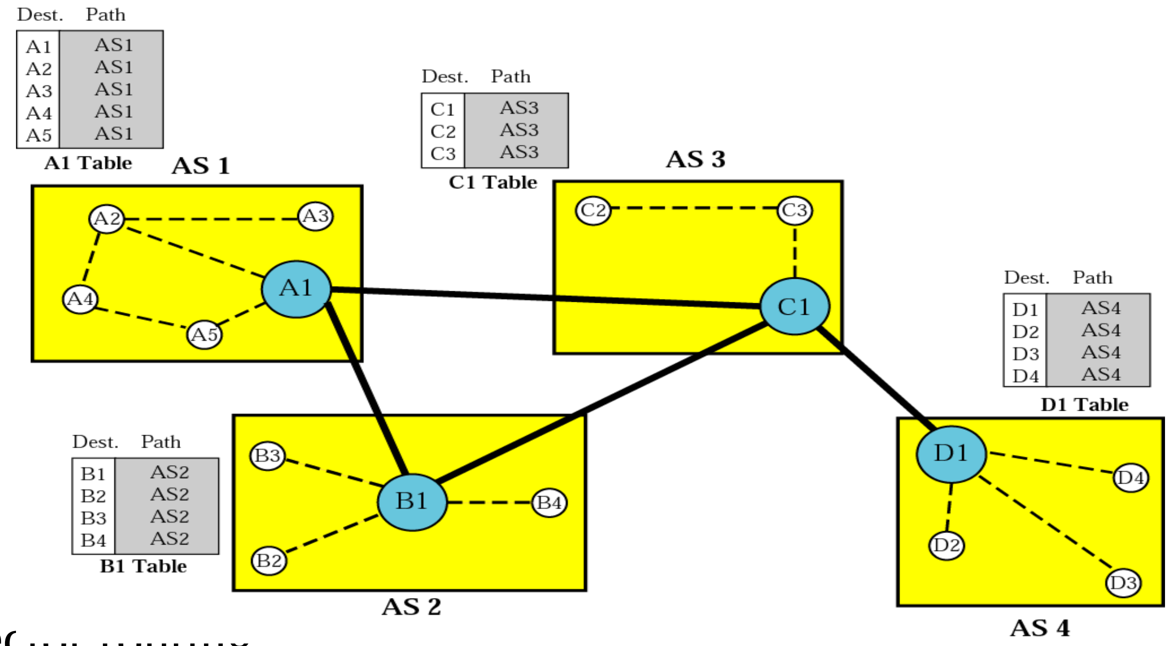
- Inter domain routing protocol.
- Uses path vector routing.

- Categories of autonomous systems:
  - Stub AS : Only one connection to other AS.

  - Multihomed AS : More than one connection

  - Transient AS : Multihomed AS that also allows transient traffic

PATH VECTOR ROUTING



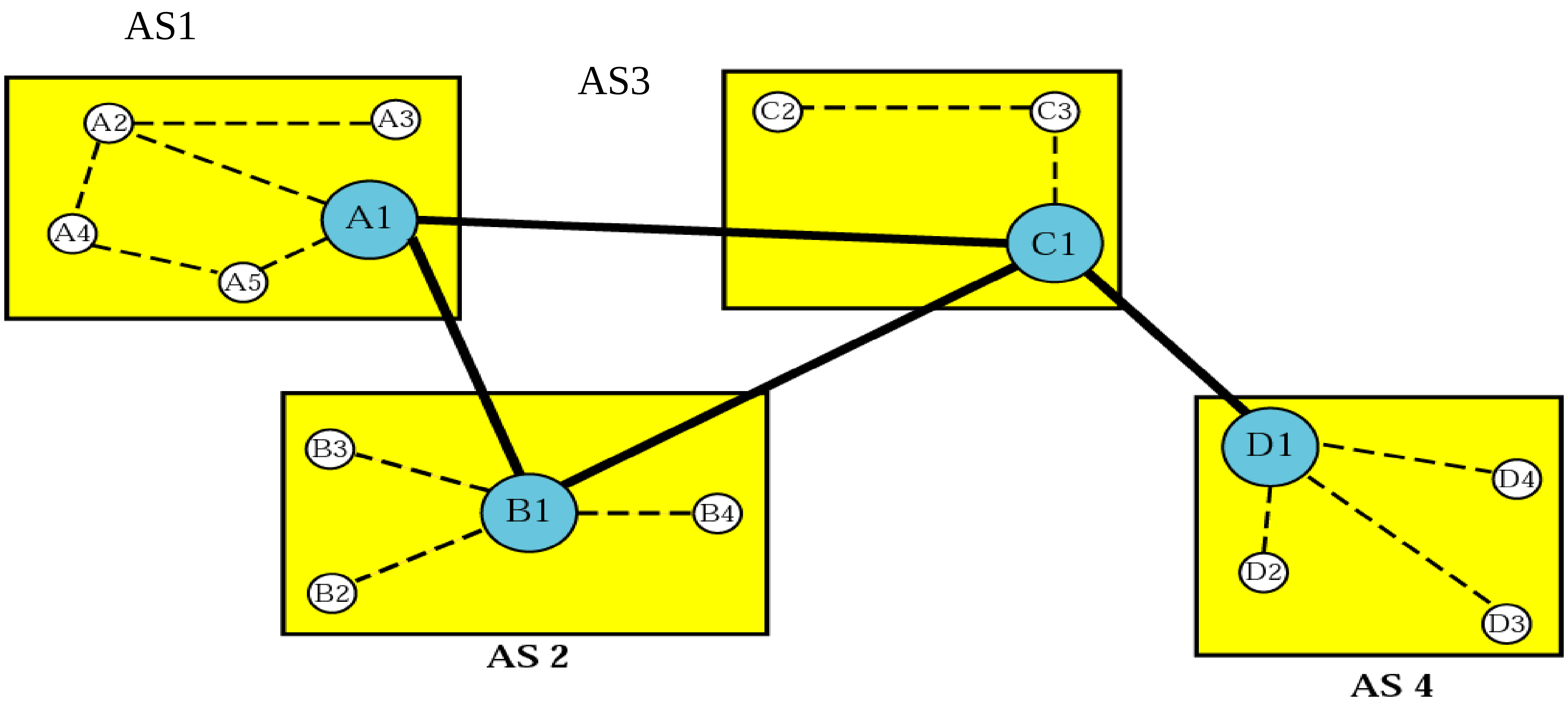
- Principal of path vector routing is similar to distance vector routing.
- Inter-domain routing protocol
- There is at least one node, called the speaker node, in each AS.
- It creates a routing table and advertises it to speaker nodes in the neighboring ASs.
- Only Speaker nodes can communicate with each other
- Speaker node advertises the path.

Path Vector Routing

- Initialization
- Sharing
- Updating

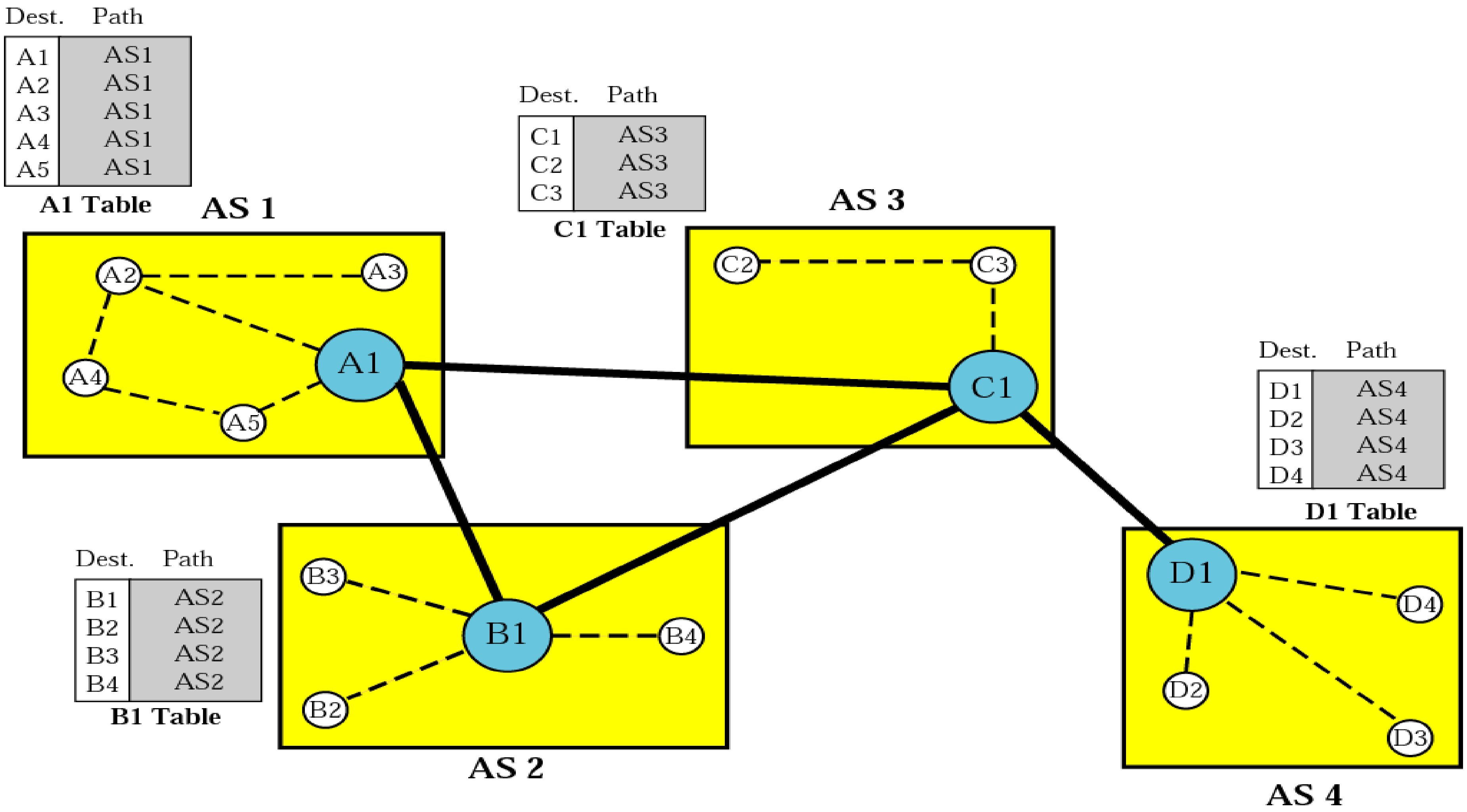
Initialization

- Each speaker node initially is reachable only to its neighbors within an autonomous system



Sharing

- Speaker node shares its table with immediate neighbors.



Update

A1 Table

Dest.	Path
A1	AS1
...	
A5	AS1
B1	AS1-AS2
...	...
B4	AS1-AS2
C1	AS1-AS3
...	...
C3	AS1-AS3
D1	AS1-AS3-AS4
...	
D4	AS1-AS3-AS4

B1 Table

Dest.	Path
A1	AS2-AS1
...	
A5	AS2-AS1
B1	AS2
...	...
B4	AS2
C1	AS2-AS3
...	...
C3	AS2-AS3
D1	AS2-AS3-AS4
...	...
D4	AS2-AS3-AS4

C1 Table

Dest.	Path
A1	AS3-AS1
...	
A5	AS3-AS1
B1	AS3-AS2
...	...
B4	AS3-AS2
C1	AS3
...	...
C3	AS3
D1	AS3-AS4
...	...
D4	AS3-AS4

D1 Table

Dest.	Path
A1	AS4-AS3-AS1
...	
A5	AS4-AS3-AS1
B1	AS4-AS3-AS2
...	...
B4	AS4-AS3-AS2
C1	AS4-AS3
...	...
C3	AS4-AS3
D1	AS4
...	...
D4	AS4

Updating

- Loop Prevention
  - When router receives a message, if its autonomous system is in the path list to the destination, looping is involved and message is ignored.

- Policy Routing
  - If one of the autonomous systems listed in the path is against its policy, it does not update its routing table with this path, and it does not forward this message to its neighbors.

BGP - Border Gateway Protocol

- Inter domain routing protocol.
- Uses path vector routing.
- Current version is 4

- Categories of autonomous systems:
    - Stub AS : Only one connection to other AS.

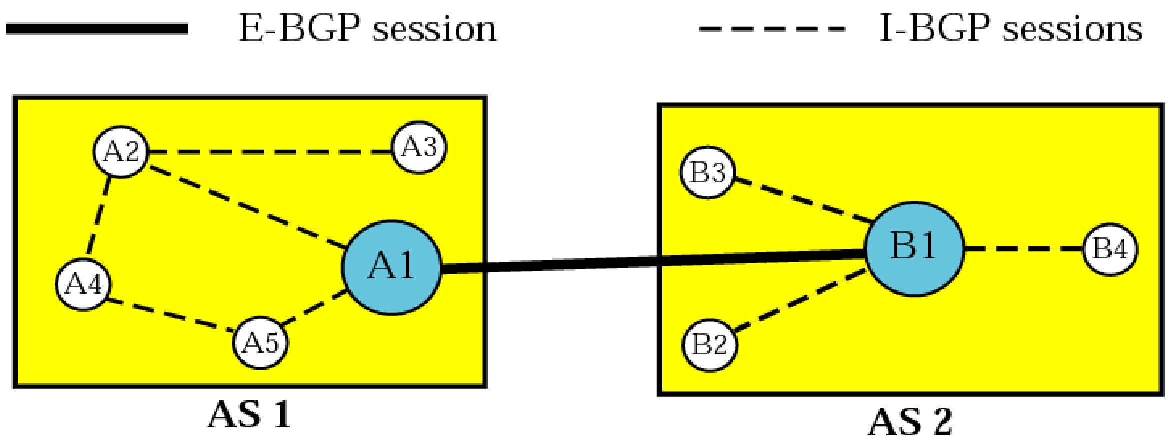
    - Multihomed AS : More than one connection

BGP Sessions

- Session is an exchange of routing information between two routers.

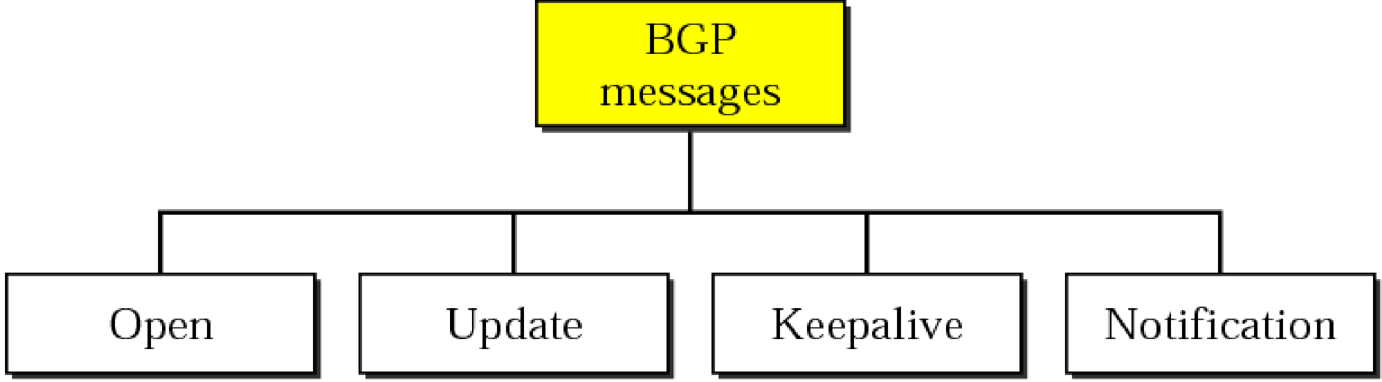
- Connection established using TCP.

- Two types of sessions
  - External BGP
  - Internal BGP



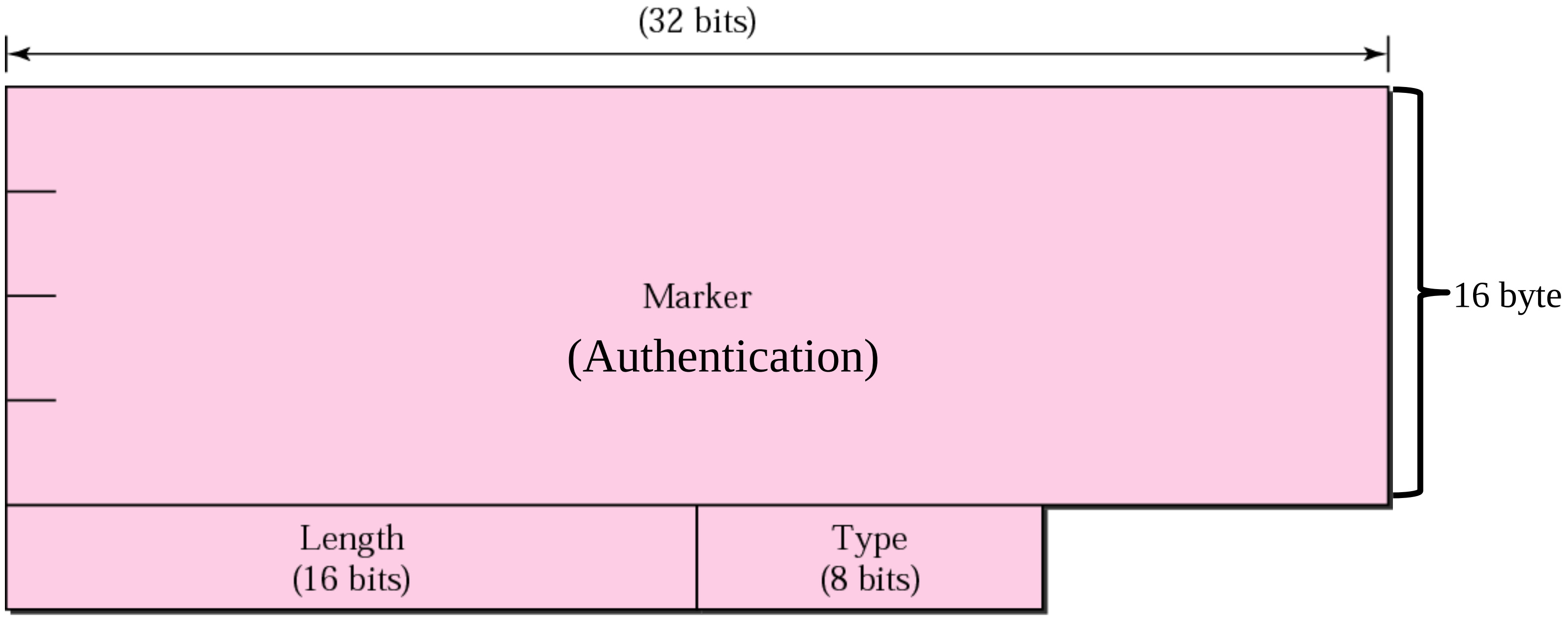
- E-BGP : exchange information between **two speaker nodes**
- I-BGP : between **two routers inside** an autonomous system.

Types of BGP messages



Message Number	Message Type	Message Description
1	OPEN message	Used to open BGP sessions
2	UPDATE message	Carries route updates for established BGP sessions
3	NOTIFICATION message	Notifies a peer router of an error condition
4	KEEPALIVE message	Sent between BGP peering routers to verify BGP session

BGP packet format



•***Marker:*** This large field at the start of each BGP message is used for synchronization and authentication.
•***Length:*** The total length of the message in bytes, including the fields of the header. The minimum value of this field is 19 for a *Keepalive* message; it may be as high as 4,096.

Type: Indicates the BGP message type:

Type Value (Decimal)	Message Type
1	*Open*
2	*Update*
3	*Notification*
4	*Keepalive*

Summary……….

- Direct & indirect delivery.
- Discussed structure of router and routing table
- Delivery of IP Datagram – two processes
  - Routing - Contents of routing table
  - Forwarding –
    - ➢ Route specific
    - ➢ Next hop method
    - ➢ Network specific
    - ➢ Host specific
    - ➢ Default

Summary………..

- Introduced the idea of autonomous systems (As)
- Discussed the concept of intra and inter domain routing.
- To discuss Unicast routing protocols,
  - **<u>Routing Information Protocol (RIP)</u>** which implements the idea of distance vector routing in the Internet.
  - **<u>Open Shortest Path First (OSPF)</u>** which implements the idea of link state routing in the Internet.
  - **<u>Border Gateway Protocol (BGP)</u>** which implements the idea of path vector routing in the Internet.