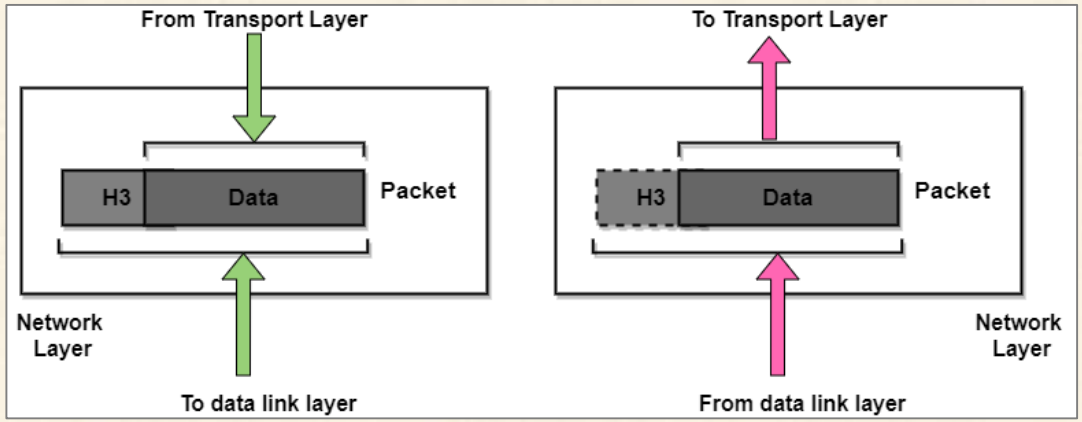
CN: UNIT-3 Network Layer

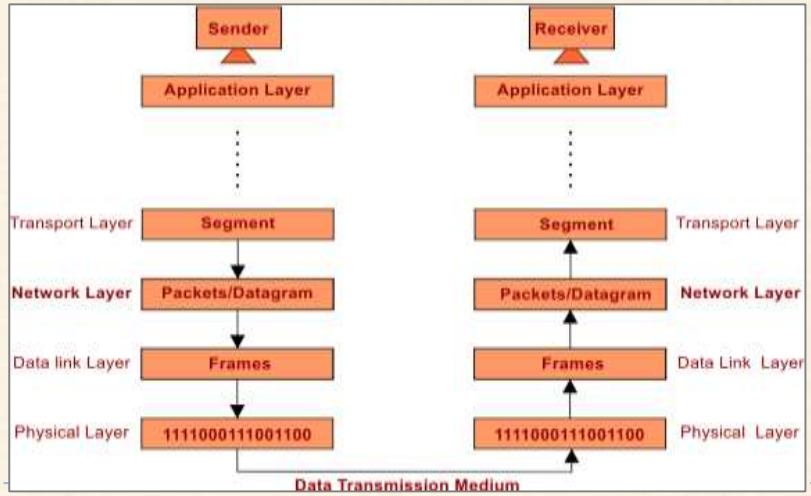
Network Layer

- Network Layer is layer 3 of the OSI reference model. The network layer controls the operation of the subnet. The main aim of this layer is to **deliver packets** from source to destination across multiple links (networks). It **routes the signal through different channels** to the other end and acts as a network controller.
- As the data link layer oversees the delivery of the packets between two systems **on the same network**; the network layer mainly ensures that each packet gets from its **point of origin to the final destination**.
- It also divides the outgoing messages into **packets and to assemble incoming packets into messages** for higher levels.
- If two computers (system) are connected on the **same link, then there is no need** for a network layer. But in case if two systems ate attached to **different networks(links)** with connecting devices between the networks(links), then there is a **need for the network layer** in order to accomplish the source-to-destination delivery.



Network Layer

- The network layer is responsible for converting **logical addresses into physical addresses.** It decides the path from the source to the destination and manages **issues such as switching, routing, and data packet congestion.**
- The network layer's primary function is to transport packets from the sending host to the receiving host.
- **At Sender Side,** Network layer receives **segments** from transport layer and convert these segments into packets/datagrams and transmit these packets/datagrams to data link layer.
- **At Receiver Side,** Network layer receives frames from data link layer and convert these frames to packets/datagrams and then transmit to transport layer.



Design Issues

Network Layer Design Issues

- A key design issue is **determining how packets are routed from source to destination**. Routes can be based on static tables that are wired into the network and rarely changed. They can also be highly dynamic, being determined a new for each packet, to reflect the current network load.
- If **too many packets** are present in the subnet at the same time, they will get into one another's way, forming **bottlenecks**. The **control of such congestion** also belongs to the network layer.
- Moreover, the **quality of service** provided(delay, transmit time, jitter, etc) is also a network layer issue.
- When a packet has to **travel from one network to another to get to its destination**, many problems can arise such as:
  - The addressing used by the second network may be different from the first one.
  - The second one may not accept the packet at all because it is too large.
  - The protocols may differ, and so on.
- It is up to the network layer to overcome all these problems to allow heterogeneous networks to be interconnected.

Design Issues

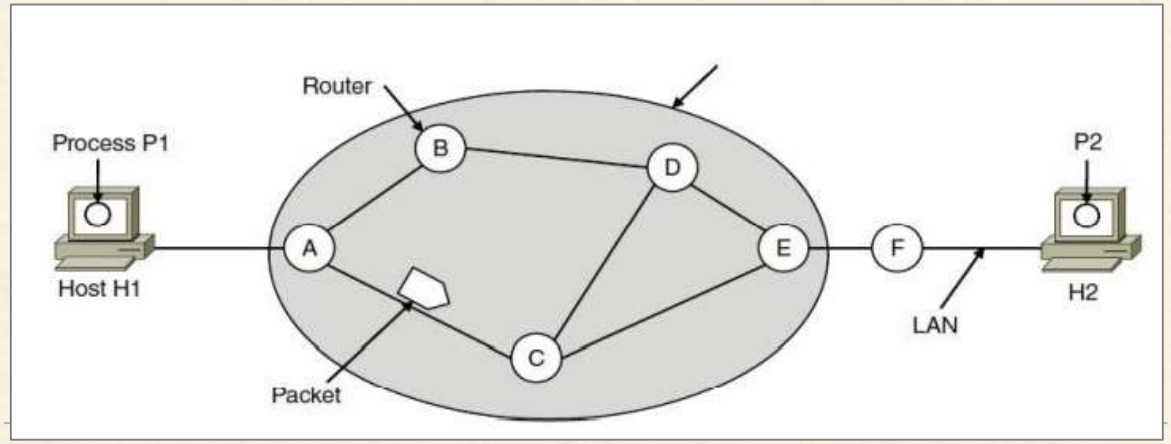
Network Layer Design Issues

1. Store-and-forward packet switching
2. Services provided to transport layer
3. Providing of connectionless service
4. Providing of connection-oriented service
5. Comparison of virtual-circuit and datagram networks

Design Issues

1. Store-and-forward packet switching

- A host with a packet to send **transmits it to the nearest router,** either on its own LAN or over a point-to-point link to the ISP.
- The packet is **stored there until it has fully arrived** and the link has finished its processing by verifying the checksum.
- Then it is **forwarded to the next router** along the path until it reaches the destination host, where it is delivered.
- This mechanism is **store-and-forward packet switching.**



Design Issues

2. Services provided to transport layer

- The network layer provides service its immediate upper layer, namely transport layer, through the network – transport layer interface.

The two types of services provided are –

i. Connection – Oriented Service – In this service, a **path is setup** between the source and the destination, and all the data packets belonging to a message are routed along this path.

ii. Connectionless Service – In this service, each packet of the message is considered as an **independent entity** and is **individually routed** from the source to the destination.

The objectives of the network layer while providing these services are –

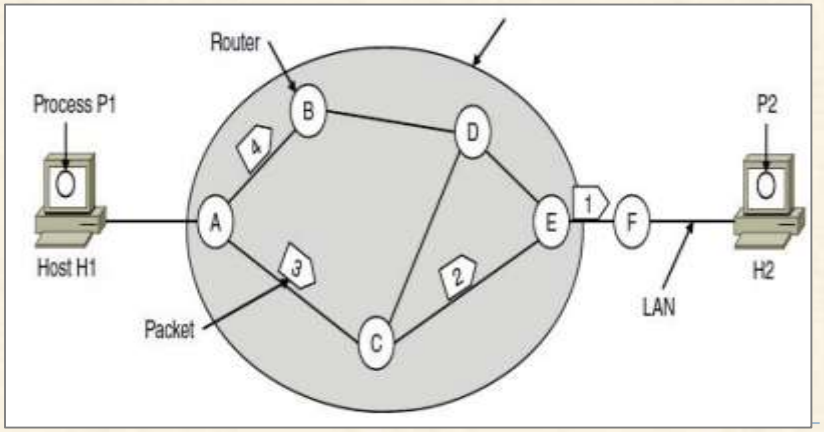
1. The services should **not be dependent** upon the router technology.
2. The router **configuration details should not** be of a concern to the transport layer.
3. A **uniform addressing** plan should be made available to the transport layer, whether the network is a **LAN, MAN or WAN.**

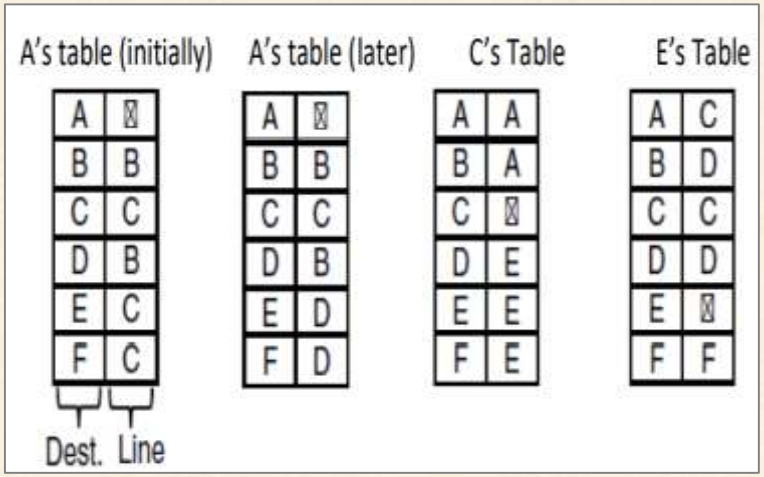
Design Issues

3. Providing of connectionless service

- If connectionless service is offered, packets are **injected** into the network **individually and routed independently** of each other.
- No advance setup is needed. In this context, the packets are frequently called **datagrams** (in analogy with telegrams) and the network is called a **datagram networks or datagram subnets.**
- An example of connectionless service is **Internet Protocol or IP.**

Let us assume for this example that the message is four times longer than the maximum packet size, so the network layer has to break it into four packets, 1, 2, 3, and 4, and send each of them in turn to router A.





Design Issues

- Every router has an internal table telling it where to send packets for each of the possible destinations. Each table entry is a pair(destination and the outgoing line). Only directly connected lines can be used.

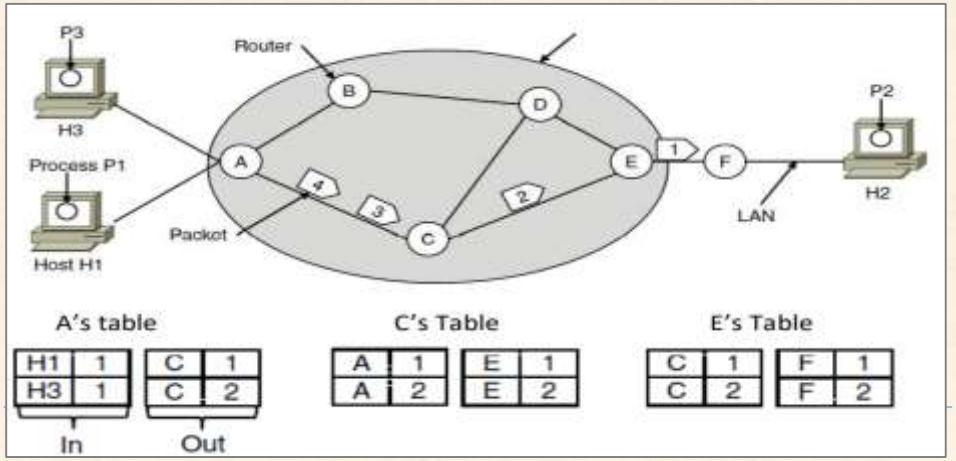
A's initial routing table is shown in the figure under the label "initially."

- At A, packets 1, 2, and 3 are stored briefly, having arrived on the incoming link. Then each packet is forwarded according to A's table, onto the outgoing link to C within a new frame. Packet 1 is then forwarded to E and then to F.
- However, something different happens to packet 4. When it gets to A it is sent to router B, even though it is also destined for F. For some reason (traffic jam along ACE path), A decided to send packet 4 via a different route than that of the first three packets. Router A updated its routing table, as shown under the label "later."
- The algorithm that manages the tables and makes the routing decisions is called the **routing algorithm.**

Design Issues

4. Providing of connection-oriented service

- If connection-oriented service is used, a **path** from the source router all the way to the destination router **must be established** before any data packets can be sent. This connection is called a **VC (virtual circuit)**, and the network is called a **virtual-circuit network.**
- When a connection is established, a route from the source machine to the destination machine is chosen as part of the **connection setup and stored in tables** inside the routers.
- That route is used for **all traffic flowing over the connection**, exactly the same way that the telephone system works. When the connection is released, the virtual circuit is also terminated.
- With connection-oriented service, each packet carries an identifier telling which virtual circuit it belongs to.



Design Issues

- Here, host H1 has established connection 1 with host H2. This connection is remembered as the first entry in each of the routing tables. The first line of A's table says that if a packet bearing connection identifier 1 comes in from H1, it is to be sent to router C and given connection identifier 1. Similarly, the first entry at C routes the packet to E, also with connection identifier 1.
- Now let us consider what happens if H3 also wants to establish a connection to H2. It chooses connection identifier 1 (because it is initiating the connection and this is its only connection) and tells the network to establish the virtual circuit. This leads to the second row in the tables. Note that we have a conflict here because although A can easily distinguish connection 1 packets from H1 from connection 1 packets from H3, C cannot do this. For this reason, A assigns a different connection identifier to the outgoing traffic for the second connection. Avoiding conflicts of this kind is why routers need the ability to replace connection identifiers in outgoing packets.
- In some contexts, this process is called **label switching.** An example of a connection-oriented network service is **MPLS (Multi Protocol Label Switching).**

Design Issues

5. Comparison of virtual-circuit and datagram networks

Issue	Datagram network	Virtual-circuit network
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

Functionality

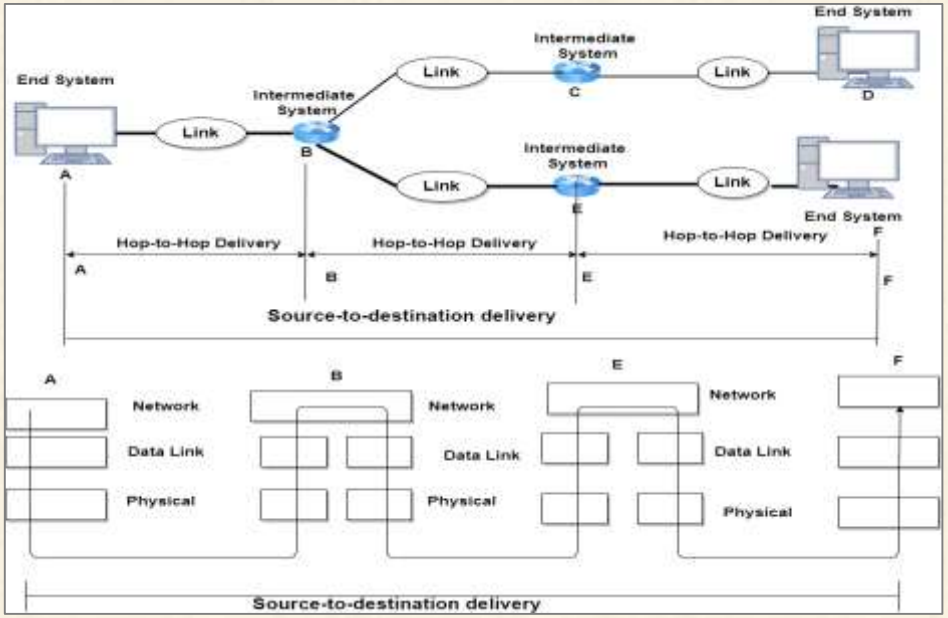
Functions of Network Layer:

1. **Source to Destination Delivery**
2. **Addressing**
3. **Routing**
4. **Internetworking**
5. **Fragmentation**
6. **Congestion Control**
7. **Flooding**

Functionality

1. Source to Destination Delivery

- Network layer provides Source to destination Delivery which also called HOST to HOST delivery. Following figure shows host to host delivery.



- In the above figure, the network layer at the A node sends the packet to the network layer at the B node. When the packet arrives at router B then the router makes the decision of the path based on the final destination that is the F node of the packet transmitted. Router B makes use of its routing table for finding the next hop that is router E. The Network layer at node B sends the packet to the network layer at E which then sends the packet to the network layer at F.

Functionality

2. Logical Addressing:

- The **physical addressing** is implemented by the data link layer, while the **logical addressing** is implemented by the network layer. The network layer appends a header to the packet that contains the logical addresses of the sender and recipient.
- Network layer is used when source to destination delivery is required in different networks or over the **internet**. Network layer use the **Logical (IP) Address to communicate** over the internet. IP address contains the **network ID and Host ID** of destination machine. Network layer use **IPV4 or IPV6 for addressing purpose.**

3. Routing: Networking layer uses the **Router device** to determine the best optimal path out of the multiple paths from source to the destination. Router uses **routing protocols (i.e. RIP, OSPF etc.)**

4. Internetworking: The network layer's primary function is to establish **logical connections between different** types of networks.

5.Fragmentation: Sometimes when a sender sends a packet to router then router may not have enough space to accommodate entire packet. So, it is required to **break these packets into fragmentations** (parts). So, fragmentation of packets/datagram is also responsibility of network layer.

Functionality

6. Congestion Control:

- The case, when maximum nodes send the data at a time to same router even with fragmentations, then buffer of router may be out of capacity. In this case, traffic must be control. Controlling of traffic is called congestion control. So, in some cases congestion control is required which is also the responsibility of Network layer.

6. Flooding:

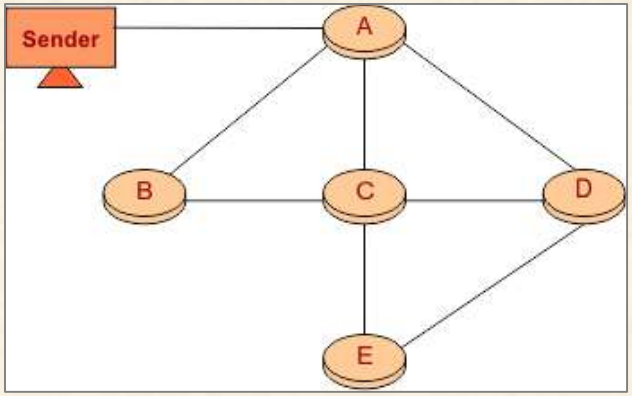
When a data packet arrives at a router, then router sent this data to all the outgoing links from router except the link from which the data arrived.

Example

- Suppose there are 5 routers (A, B, C, D and E) which are connected though transmission lines as given below.

By using flooding technique

1. An incoming data packet from sender will send to Router A.
2. Router A will forward the data packet to other routers B, C and D.
3. B will send the packet to C.
4. C will send the packet to B,D and E.
5. D will send the packet to C and E.
6. E will send the packet to D.



Note: Main advantage of flooding is that, the shortest path is always chosen by flooding because in flooding **each router holds the information's of their neighbours.**

Services

Forwarding and Routing:

- A router is used on the network layer to forward packets. A forwarding table is included on every router. A router passes a packet by **inspecting the header field and then indexing it into the forwarding table** using the header field value. The forwarding table value matching to the header field value specifies the router's outgoing interface connection to which the packet is to be forwarded.

Network Layer Services:

1. **Guaranteed delivery:** This layer offers a service that **ensures the packet arrives** at its destination.
2. **Guaranteed delivery with bounded delay:** It is another service provided by the network layer and it guarantees that the packet will surely be delivered within a specified **host-to-host delay bound.**
3. **In-Order packets:** This service assures that packets reach their destination in the **order they were delivered.**
4. **Guaranteed maximum jitter:** This service assures that the **time between two consecutive transmissions** at the sender **equals the time between** their receipt at the destination.
5. **Security services:** These are provided at the network layer through the use of a **session key between** the source and destination hosts. The payloads of datagrams transmitted to the destination host are **encrypted** by the network layer of the source host. The payload would subsequently be **decrypted** by the network layer at the target host. In this manner, the network layer ensures data integrity and source authentication services.

Pros & Cons

Advantages of Network Layer Services

Given below are some benefits of services provided by the network layer:

1. By forwarding service of the network layer, the data packets are transferred from one place to another in the network.
2. In order to reduce the traffic, the routers in the network layer **reduce collisions and broadcast the domains.**
3. Failure in the data communication system gets eliminated by **packetization.**

Disadvantages of Network layer Services

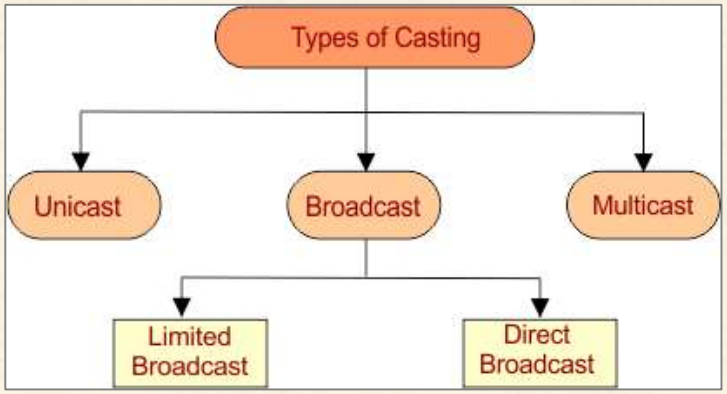
1. In the design of the network layer, there is a **lack of flow control.**
2. In the network layer, there is a **lack of proper error control** mechanisms; due to the **presence of fragmented** data packets the implementation of error control mechanism becomes difficult.
3. Due to the presence of **too many datagrams** there happens **occurrence of congestion.**

Casting And Its Types

Casting is a method of transferring a packet to various hosts simultaneously by using an IP address.

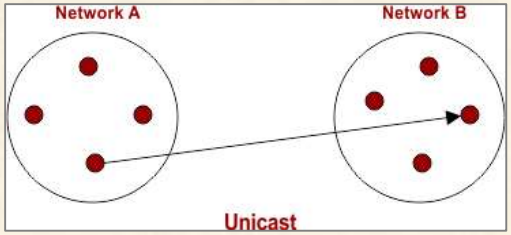
Types of Casting

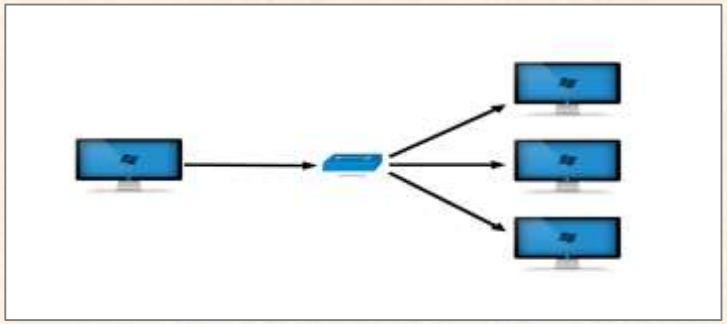
- There are three types of casting



1. Unicast

- Transmitting a packet from **one source node to one destination node** is called as unicast.
- We can say, it is a **one to one** transmission.





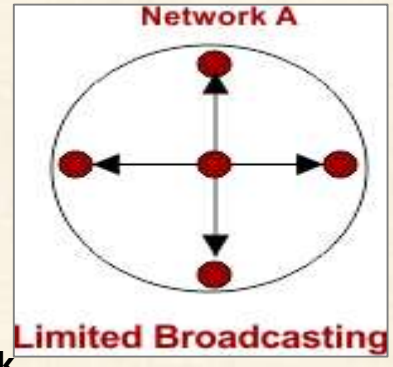
Example

- Consider a Node/Host "A" having IP Address 11.2.2.31 in one network is sending data to Node/Host "B" having IP Address 21.11.41.21 in another network.
- Then, Source IP Address of Host A = **11.2.2.31** Destination IP Address of Host B = **21.11.41.21.**

Casting And Its Types

2. Broadcast

- In Broadcasting, Packet is send to all residing host in the same or different network, depending on its types. It is a one to all transmission.



Broadcasting is of two types

1. Limited Broadcast
2. Direct Broadcast

1. Limited Broadcast

- According to Limited Broadcasting, Packet is send to **all residing host in the same network.**
- If a Host need to send a broadcast message with in the same network then All 32 bits of **IP address are set to 1.As**
- Limited Broadcast Address for any network = **11111111.11111111.11111111.11111111 = 255.255.255.255**
- This IP address cannot pass through router to go for another network.

Example

- Consider a Node/Host "A" having IP Address 11.2.2.31 is sending data to all other hosts residing in the same network.

Then,

- Source IP Address = **IP Address of host A = 11.2.2.31**
- Destination IP Address = **255.255.255.255**

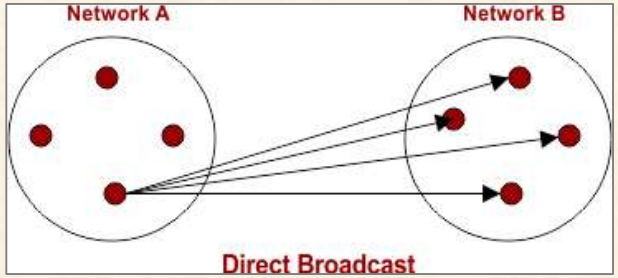
Casting And Its Types

2. Direct Broadcast

- According to direct Broadcasting, Packet is send to **all residing host in the other network**
- If a Host in a network wants to send a broadcast message to other network then **all hosts bits of IP address are set to 1.**
- This IP address can pass through router to go for another network.

Example

- Host "A" in one Network having IP Address 11.11.121.13 sending data to all other hosts residing in the network having IP Address 21.0.0.0

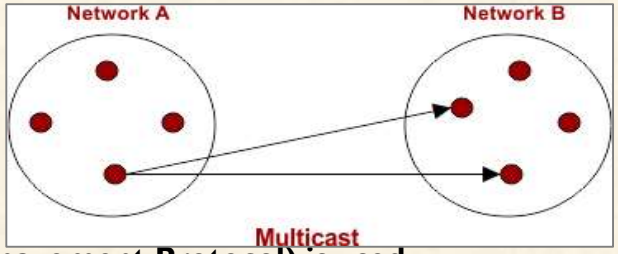


Then,

- Source IP Address of host A = **11.11.121.13**
- Destination Address = **21.255.255.255**

3. Multicast

- Transmitting data packet from one source Node to a **particular group of Nodes** is known as Multicast.
- It is also an example of **one to many** transmission.



Examples

- Sending a message to group of people on whatsapp.
- Video conference to a particular group of people

Note: To identify the group in multicast, **IGMP (Internet Group Management Protocol) is used.**

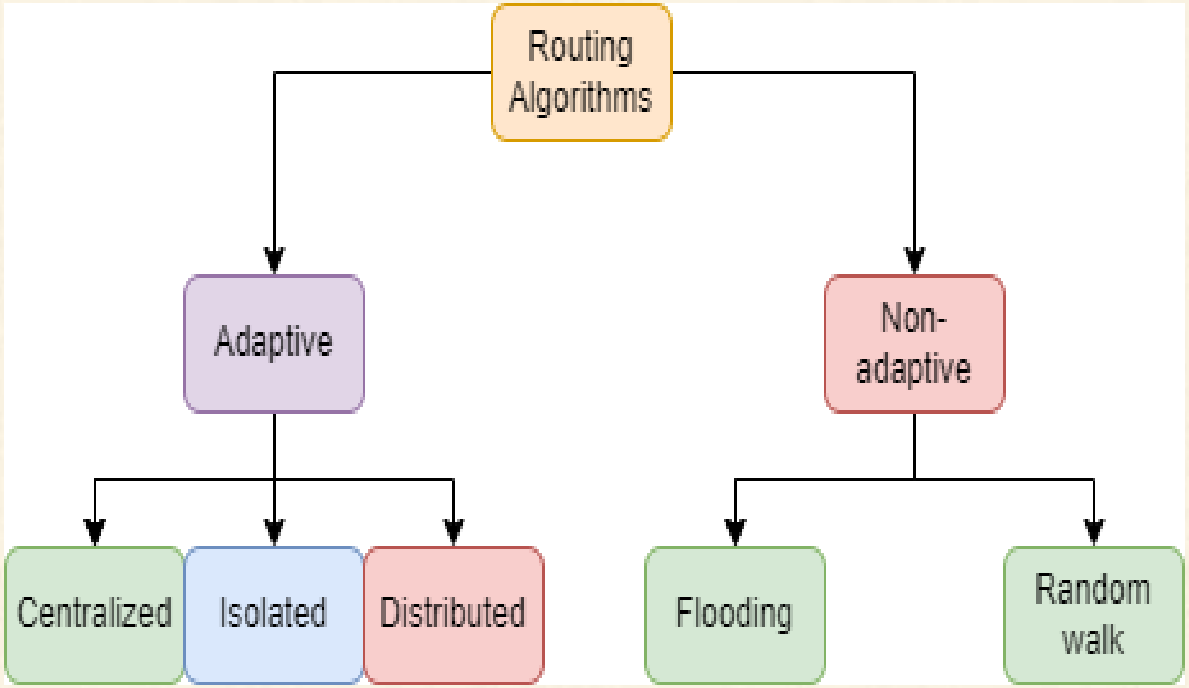
Routing Algorithms

- In order to transfer the packets from source to the destination, the network layer must determine the best route through which packets can be transmitted.
- Whether the network layer provides datagram service or virtual circuit service, the main job of the network layer is to provide the best route. The routing protocol provides this job.
- The routing protocol is a routing algorithm that provides the best path from the source to the destination. The best path is the path that has the "least-cost path" from source to the destination.
- Routing is the process of forwarding the packets from source to the destination but the best route to send the packets is determined by the routing algorithm.

Routing Algorithms

The Routing algorithm is divided into two categories:

- Adaptive Routing algorithm
- Non-adaptive Routing algorithm



Routing Algorithms

Adaptive Routing Algorithm

- It is also known as **dynamic routing.** The traffic and topology determine the decisions and select routes. It computes and optimizes:

→ Hop count.

→ Path distance to destination.

→ Transit time.

- Dynamic data such as

a.) Network Topology,

b.) Load, and

c.)Delays

are required to choose the best route.

Routing Algorithms

An adaptive routing algorithm can be classified into three parts:

- **Centralized algorithm:** It is also known as global routing algorithm as it computes the least-cost path between source and destination by using complete and global knowledge about the network. This algorithm takes the connectivity between the nodes and link cost as input, and this information is obtained before actually performing any calculation. **Link state algorithm** is referred to as a centralized algorithm since it is aware of the cost of each link in the network.
- **Isolation algorithm:** It is an algorithm that obtains the routing information by using **local information** rather than gathering information from other nodes.
- **Distributed algorithm:** It is also known as decentralized algorithm as it computes the least-cost path between source and destination in an iterative and distributed manner. In the decentralized algorithm, no node has the knowledge about the cost of all the network links. In the beginning, a node contains the information only about its own directly attached links and through an iterative process of calculation computes the least-cost path to the destination. A **Distance vector algorithm** is a decentralized algorithm as it never knows the complete path from source to the destination, instead it knows the direction through which the packet is to be forwarded along with the least cost path.

Routing Algorithms

Non-Adaptive Routing Algorithm

- It is also known as **static routing**. This algorithm's routing of data packets is free of network topology and traffic. When the network boots, the routers save the routing info.
- The best route will be selected in advance and will not change. These are two types:

Flooding Routing

- In this case, all incoming data packets are received by all outgoing links except the one from which the packet started. The loop links may generate duplicate data. We can fix this issue by using:
  a.) Sequential numbers.
  b.) Spanning trees.
  c.) Hop count.

Random walk Routing

- The data packets are transferred from host to host to one of its neighbors. The packet arrives at the destination randomly. It is a robust method. It is implemented by sending packets onto the least queued link.

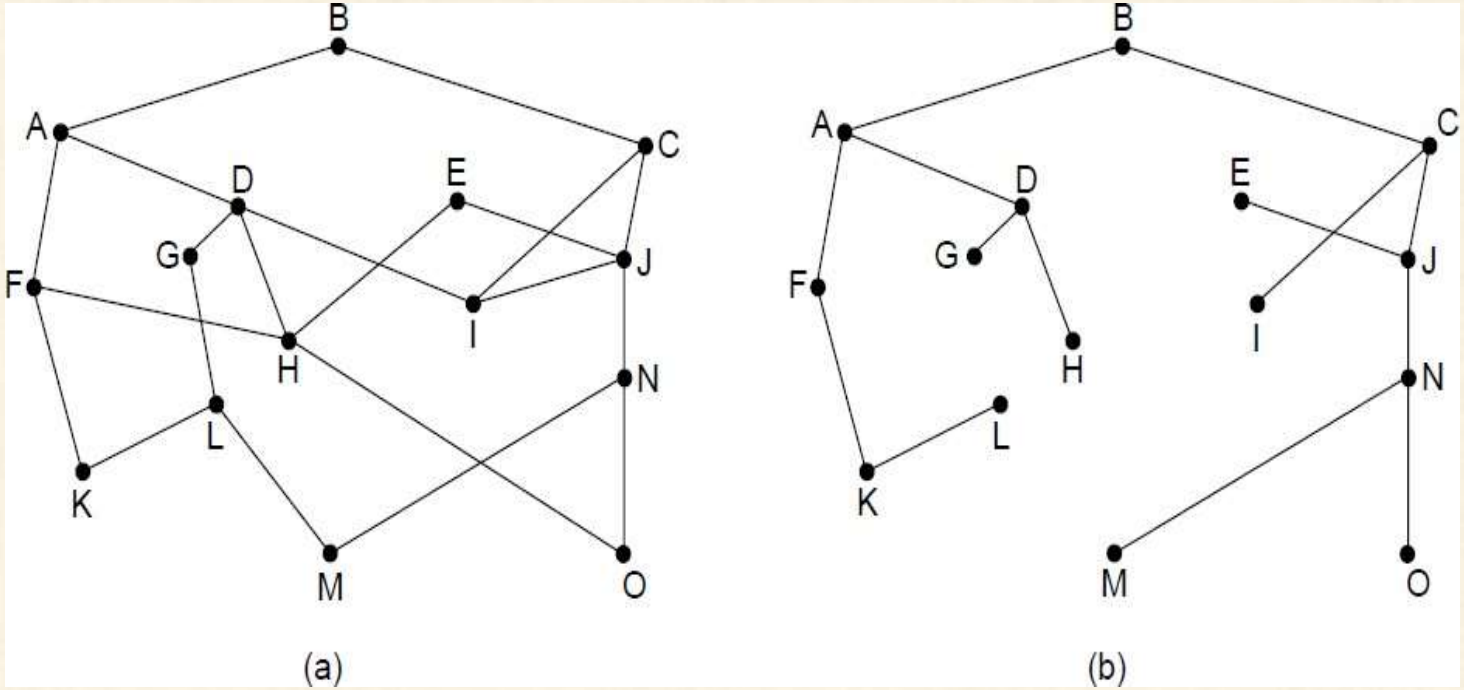
Optimality principle

- One can make a general statement about optimal routes without regard to network topology or traffic. This statement is known as the optimality principle.

- It states that if router J is on the optimal path from router I to router K, then the optimal path from J to K also falls along the same

- As a direct consequence of the optimality principle, we can see that the set of optimal routes from all sources to a given destination form a tree rooted at the destination. Such a tree is called a **sink tree**. The goal of all routing algorithms is to discover and use the sink trees for all routers

Optimality principle



(a) A network.

(b) A sink tree for router *B.*

Shortest Path Routing (a nonadaptive routing algorithm)

- Given a network topology and a set of weights describing the cost to send data across each link in the network
- Find the shortest path from a specified source to all other destinations in the network.
- Shortest path algorithm first developed by E. W. Dijkstra
- Mark the source node as permanent
- Designate the source node as the working node.
- Set the tentative distance to all other nodes to infinity.
- While some nodes are not marked permanent

  Compute the tentative distance from the source to all nodes adjacent to the working node. If this is shorter than the current tentative distance replace the tentative distance of the destination and record the label of the working node there.

- Examine ALL tentatively labeled nodes in the graph. Select the node with the smallest value and make it the new working node. Designate the node permanent.

Example of Shortest Path Routing

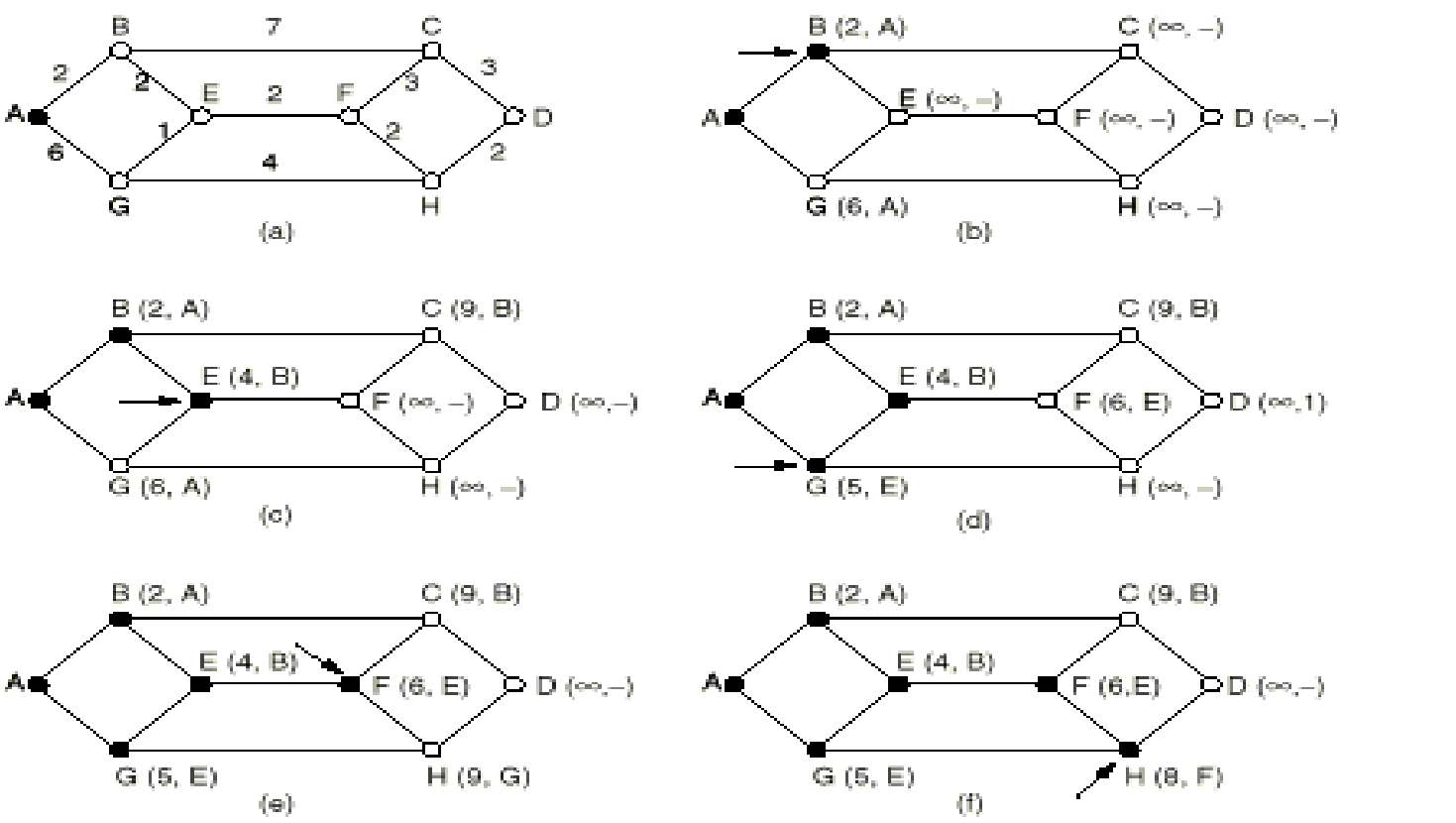


Fig. 5-6. The first five steps used in computing the shortest path from *A* to *D*. The arrows indicate the working node.

Flooding (a nonadaptive routing algorithm)

- Brute force routing
  - Every incoming packet is sent on every outgoing line
  - Always finds the shortest path quickly
  - Also finds many long paths
  - Time to live is set to size of subnet
- Selective Flooding
  - Flood only in the direction of the destination
- Practical in a few settings
  - Military Applications
  - Distributed Databases
  - Metric for comparison

Flooding
(a nonadaptive routing algorithm)

- Flooding obviously generates vast numbers of duplicate packets, in fact, an infinite number unless some measures are taken to damp the process.

- One such measure is to have a hop counter contained in the header of each packet, which is decremented at each hop, with the packet being discarded when the counter reaches zero. Ideally, the hop counter should be initialized to the length of the path from source to destination.

- A variation of flooding that is slightly more practical is **selective flooding**. In this algorithm the routers do not send every incoming packet out on every line, only on those lines that are going approximately in the right direction.

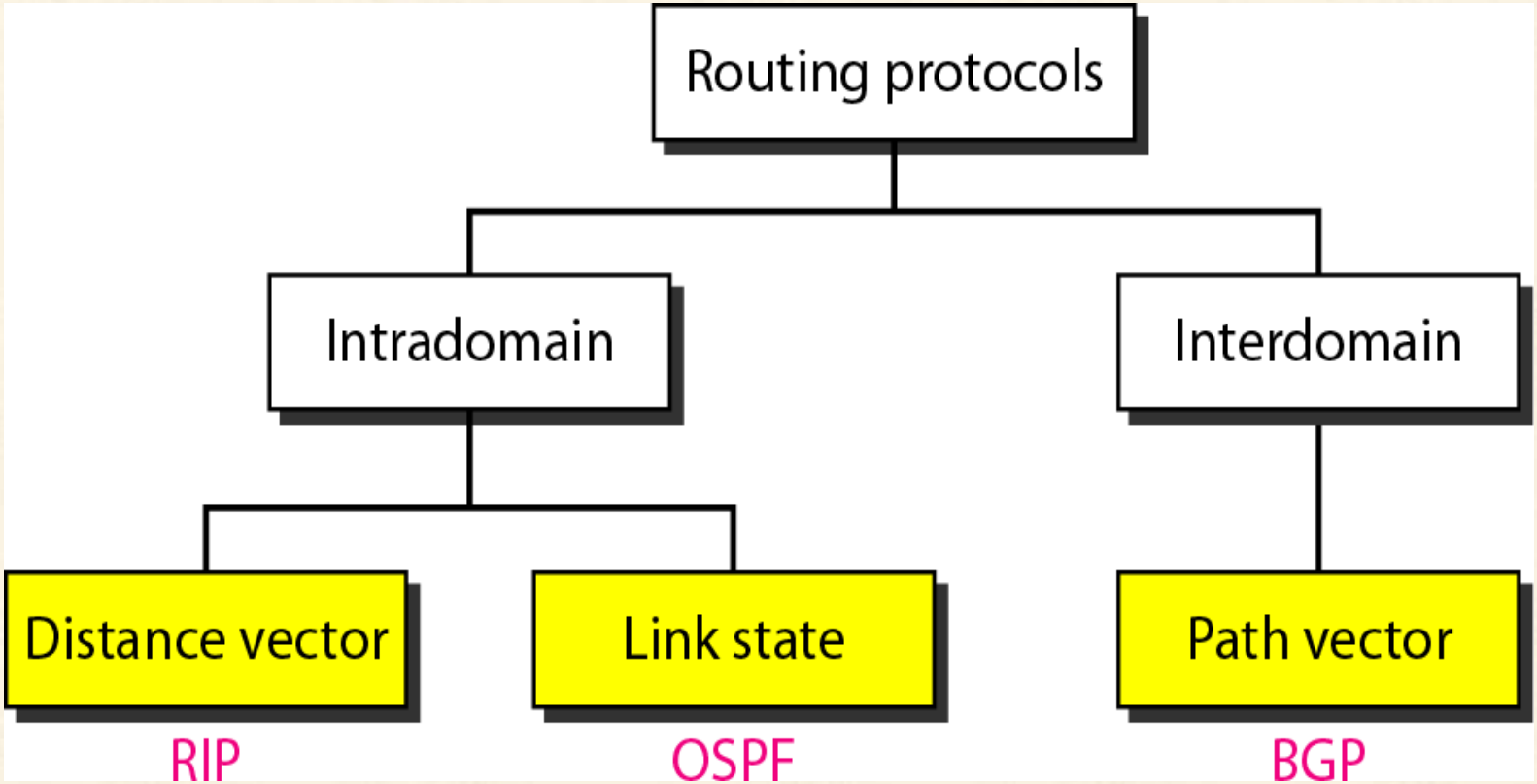
Intra and Inter domain Routing

- An autonomous system (AS) is a group of networks and routers under the authority of a single administration.

- Routing inside an autonomous system is referred to as intra domain routing. (DISTANCE VECTOR, LINK STATE)

- Routing between autonomous systems is referred to as inter domain routing. (PATH VECTOR) Each autonomous system can choose one or more intra domain routing protocols to handle routing inside the autonomous system. However, only one inter domain routing protocol handles routing between autonomous systems.

Intra and Inter domain Routing



Distance Vector Routing (an adaptive routing algorithm)

- In distance vector routing, the least-cost route between any two nodes is the route with minimum distance.

- In this protocol, as the name implies, each node maintains a vector (table) of minimum distances to every node.

- Mainly 3 things in this

1.Initialization

2.Sharing

3.Updating

Distance Vector Routing (an adaptive routing algorithm)

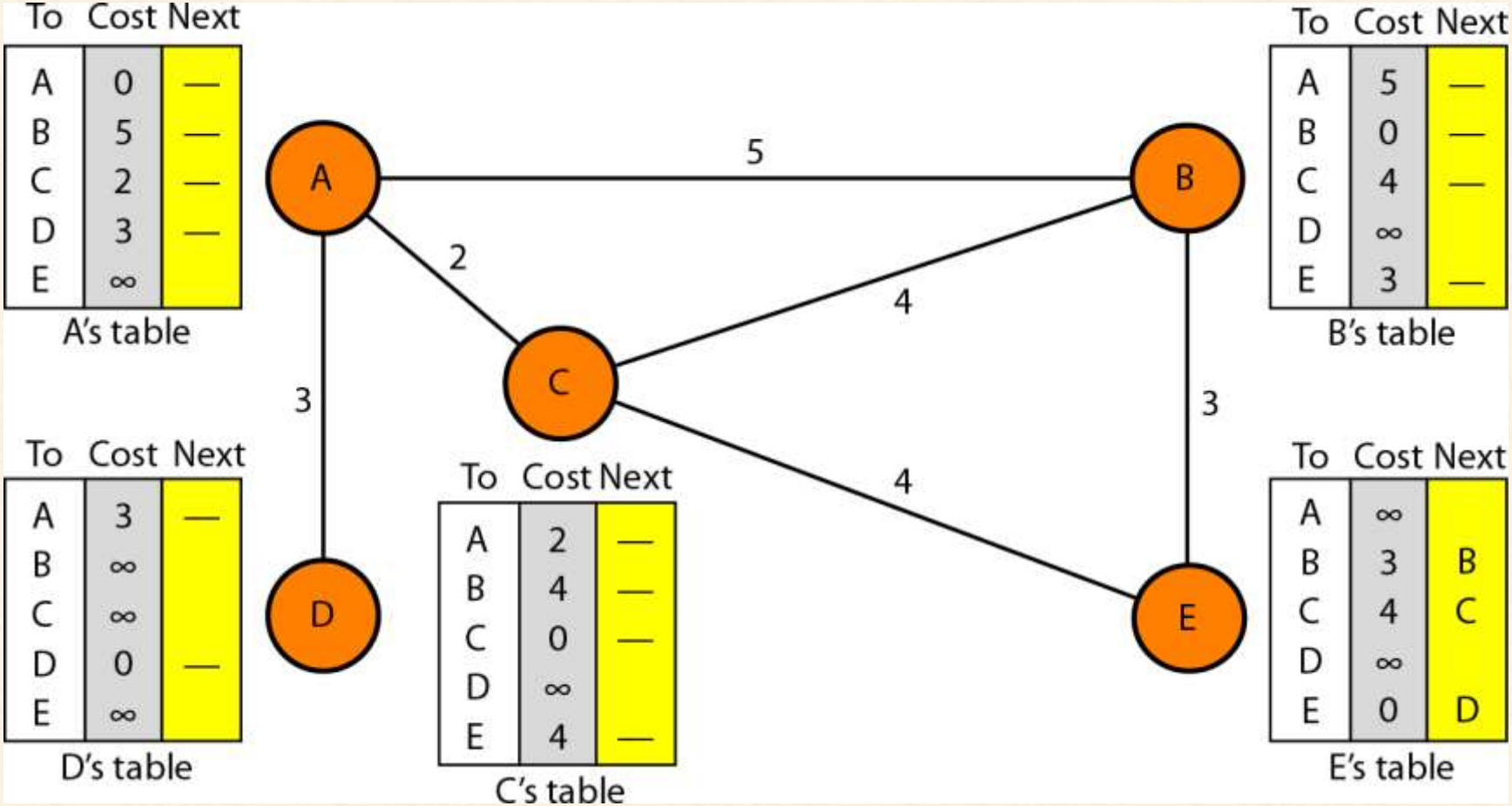
Initialization

- Each node can know only the distance between itself and its immediate neighbors, those directly connected to it.

- So for the moment, we assume that each node can send a message to the immediate neighbors and find the distance between itself and these neighbors.

- Below fig shows the initial tables for each node. The distance for any entry that is not a neighbor is marked as infinite (unreachable).

Distance Vector Routing (an adaptive routing algorithm)



Distance Vector Routing (an adaptive routing algorithm)

Sharing

- The whole idea of distance vector routing is the sharing of information between neighbors. Although node A does not know about node E, node C does. So if node C shares its routing table with A, node A can also know how to reach node E. On the other hand, node C does not know how to reach node D, but node A does. If node A shares its routing table with node C, node C also knows how to reach node D. In other words, nodes A and C, as immediate neighbors, can improve their routing tables if they help each other.
- NOTE: In distance vector routing, each node shares its routing table with its immediate neighbors periodically and when there is a change.

Distance Vector Routing (an adaptive routing algorithm)

Updating

- When a node receives a two-column table from a neighbor, it needs to update its routing table. Updating takes three steps:

1.The receiving node needs to add the cost between itself and the sending node to each value in the second column. (x+y)

2.If the receiving node uses information from any row. The sending node is the next node in the route.

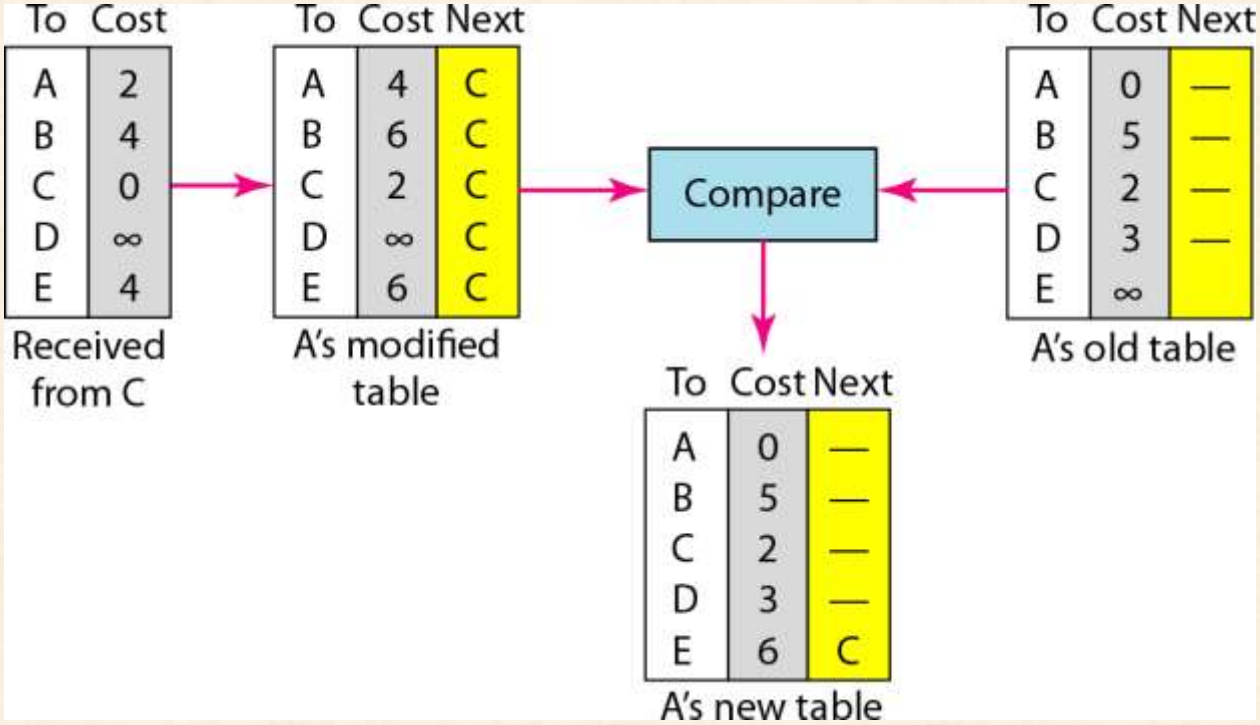
3.The receiving node needs to compare each row of its old table with the corresponding row of the modified version of the received table.

a. If the next-node entry is different, the receiving node chooses the row with the smaller cost. If there is a tie, the old one is kept.

b. If the next-node entry is the same, the receiving node chooses the new row.

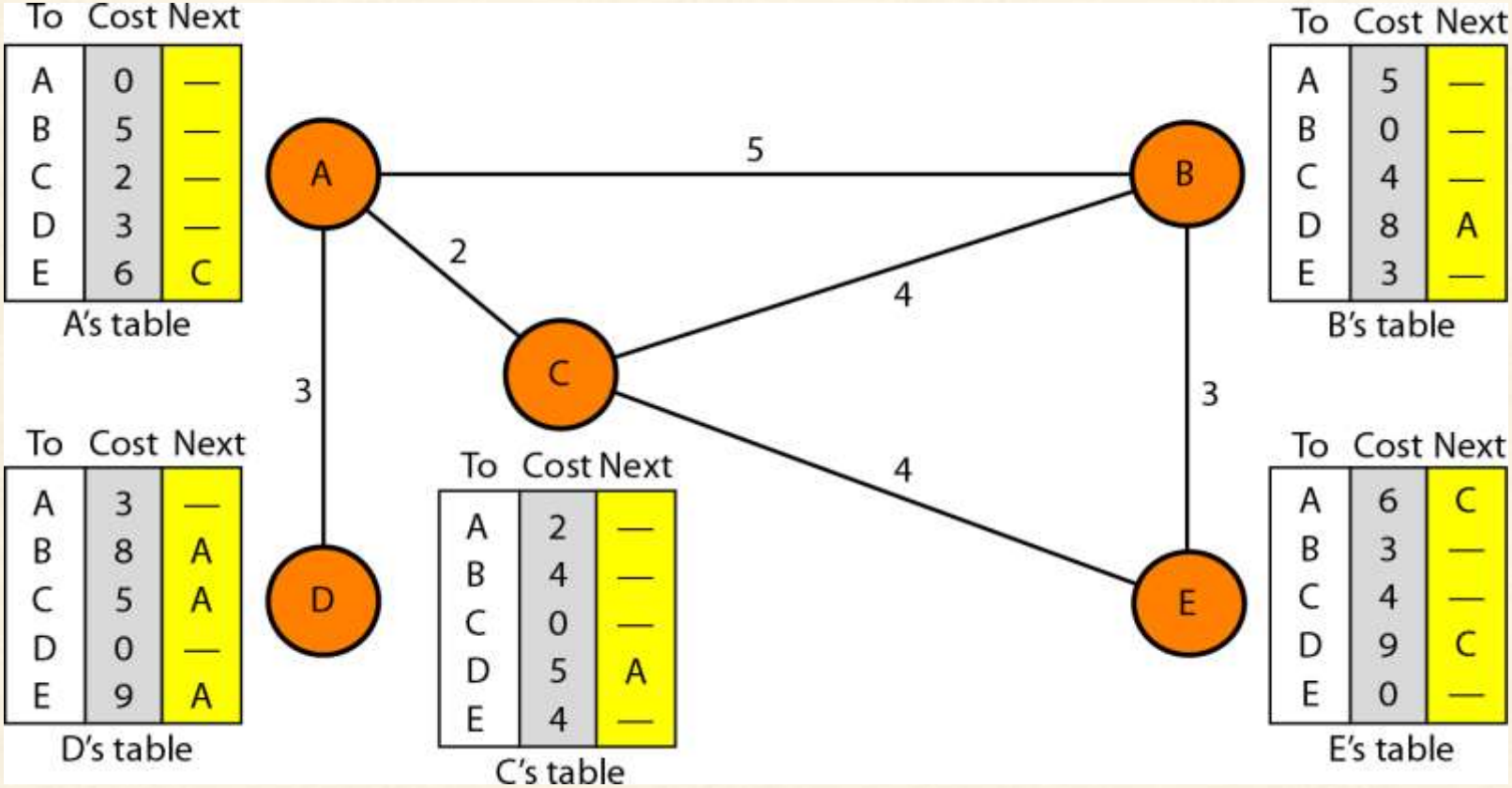
Distance Vector Routing (an adaptive routing algorithm)

- For example, suppose node C has previously advertised a route to node X with distance 3. Suppose that now there is no path between C and X; node C now advertises this route with a distance of infinity. Node A must not ignore this value even though its old entry is smaller. The old route does not exist anymore. The new route has a distance of infinity.



Distance Vector Routing (an adaptive routing algorithm)

Final Diagram

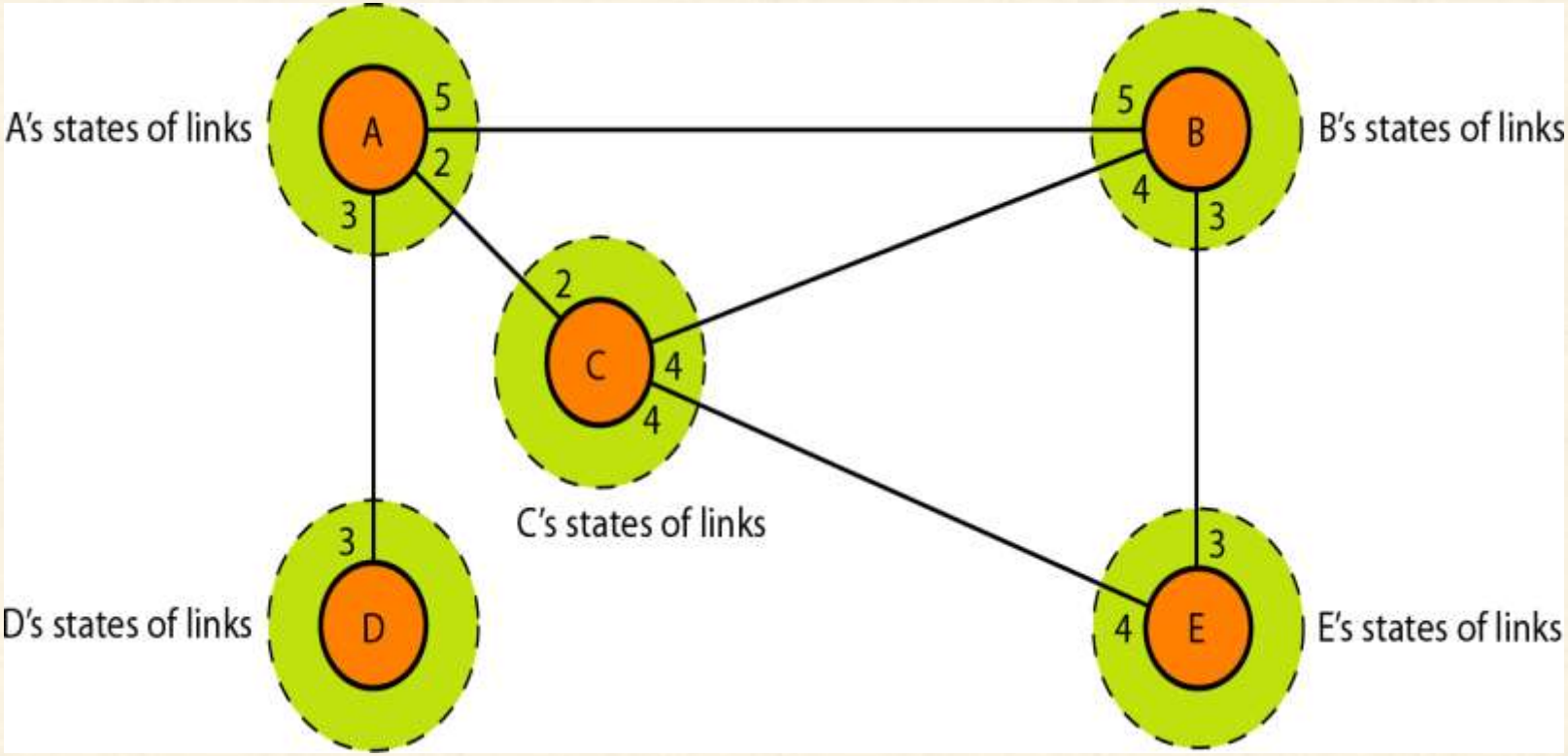


Link State Routing (an adaptive routing algorithm)

- In link state routing, if each node in the domain has the entire topology of the domain-the list of nodes and links, how they are connected including the type, cost (metric), and the condition of the links (up or down)-the node can use the Dijkstra's algorithm, to build a routing table.

- Link state routing is based on the assumption that, although the global knowledge about the topology is not clear, each node has partial knowledge: it knows the state (type, condition, and cost) of its links. **In other words, the whole topology can be compiled from the** partial knowledge of each node.

Link State Routing
(an adaptive routing algorithm)



Link State Routing
(an adaptive routing algorithm)

Building Routing Tables

1. Creation of the states of the links by each node, called the link state packet (LSP).
2. Dissemination of LSPs to every other router, called **flooding, in an efficient and** reliable way.
3. Formation of a shortest path tree for each node.
4. Calculation of a routing table based on the shortest path tree

Link State Routing
(an adaptive routing algorithm)

1.Creation of Link State Packet (LSP) :

- A link state packet can carry a large amount of information. For the moment, we assume that it carries a minimum amount of data: the node identity, the list of links, a sequence number, and age.

- The first two, node identity and the list of links, are needed to make the topology.

- The third, sequence number, facilitates flooding and distinguishes new LSPs from old ones. The fourth, age, prevents old LSPs from remaining in the domain for a long time.

Link State Routing
(an adaptive routing algorithm)

2.Flooding of LSPs: After a node has prepared an LSP, it must be disseminated to all other nodes, not only to its neighbors. The process is called flooding and based on the following

1.The creating node sends a copy of the LSP out of each interface

2.A node that receives an LSP compares it with the copy it may already have.

If the newly arrived LSP is older than the one it has (found by checking the sequence number), it discards the LSP. If it is newer, the node does the following:

a.It discards the old LSP and keeps the new one.

b.It sends a copy of it out of each interface except the one from which the packet arrived. This guarantees that flooding stops somewhere in the domain (where a node has only one interface).

Link State Routing
(an adaptive routing algorithm)

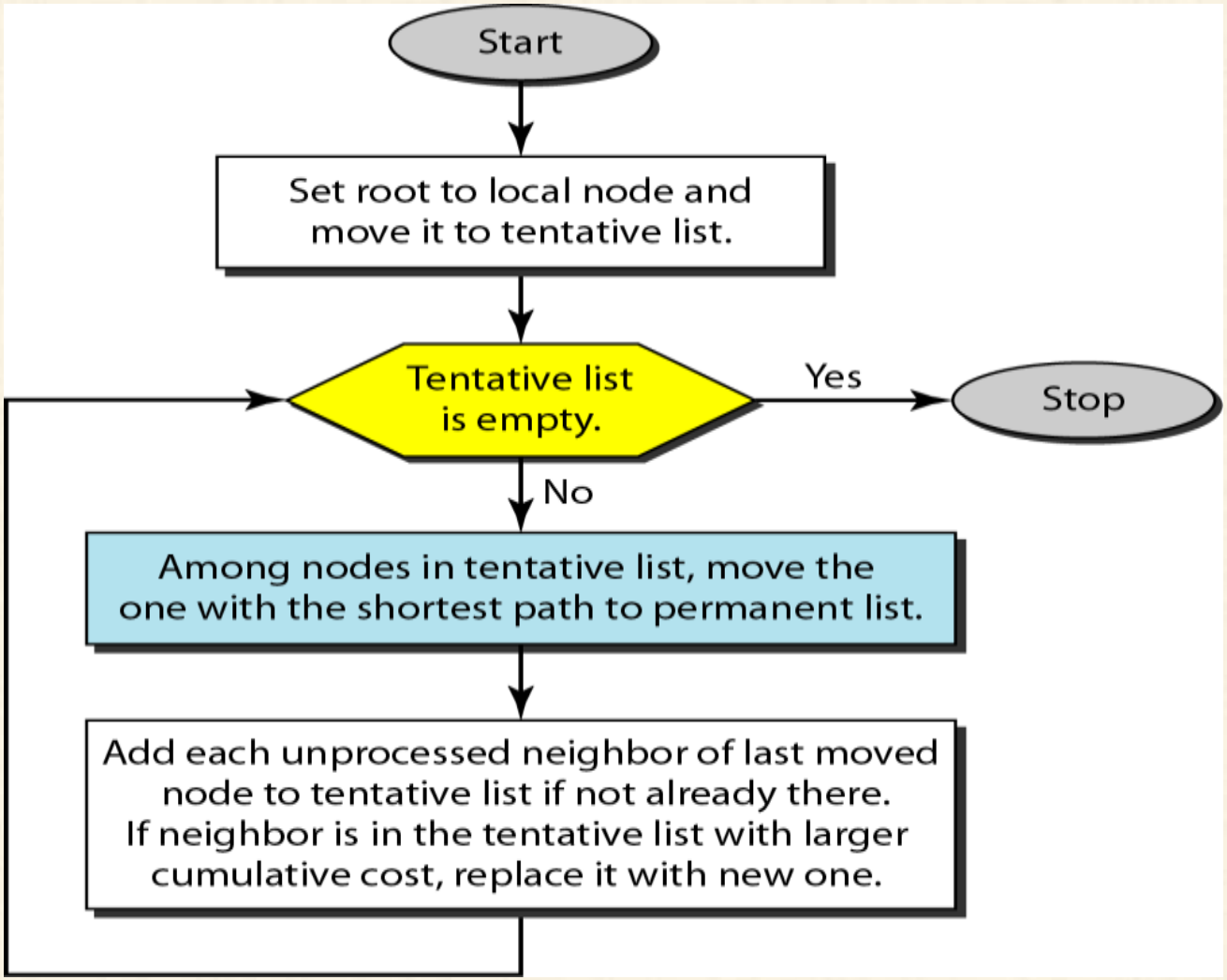
3.Formation of Shortest Path Tree: Dijkstra Algorithm

- A shortest path tree is a tree in which the path between the root and every other node is the shortest.

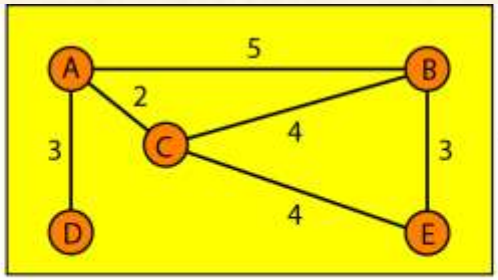
- The Dijkstra algorithm creates a shortest path tree from a graph. The algorithm divides the nodes into two sets: **tentative and permanent.**

- It finds the neighbors of a current node, makes them tentative, examines them, and if they pass the criteria, makes them permanent.

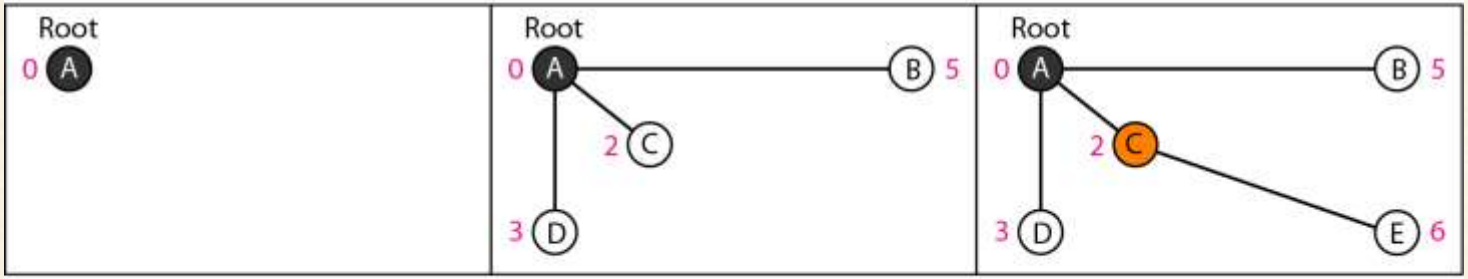
Link State Routing
(an adaptive routing algorithm)



Link State Routing
(an adaptive routing algorithm)



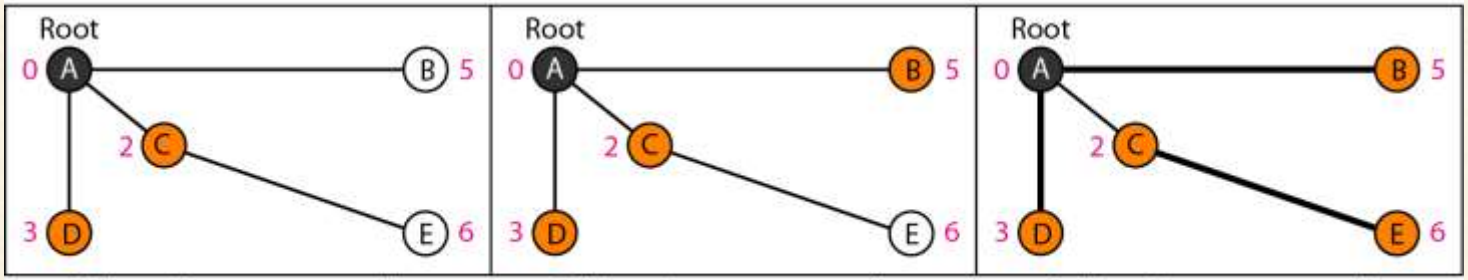
Topology



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.

6. Move E to permanent list (tentative list is empty).

Link State Routing
(an adaptive routing algorithm)

4.Calculation of a routing table:

routing table for node A

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

Path Vector Routing

- Distance vector and link state routing are both **intra domain routing protocols.** They can be used inside an autonomous system, but not between autonomous systems.
- These two protocols are not suitable for inter domain routing mostly because of scalability. Both of these routing protocols become intractable when the domain of operation becomes large.
- **Distance vector routing is subject to instability** in the domain of operation.
- **Link state routing needs a huge amount of resources** to calculate routing tables. It also creates heavy traffic because of flooding. There is a need for a third routing protocol which we call **path vector routing.**

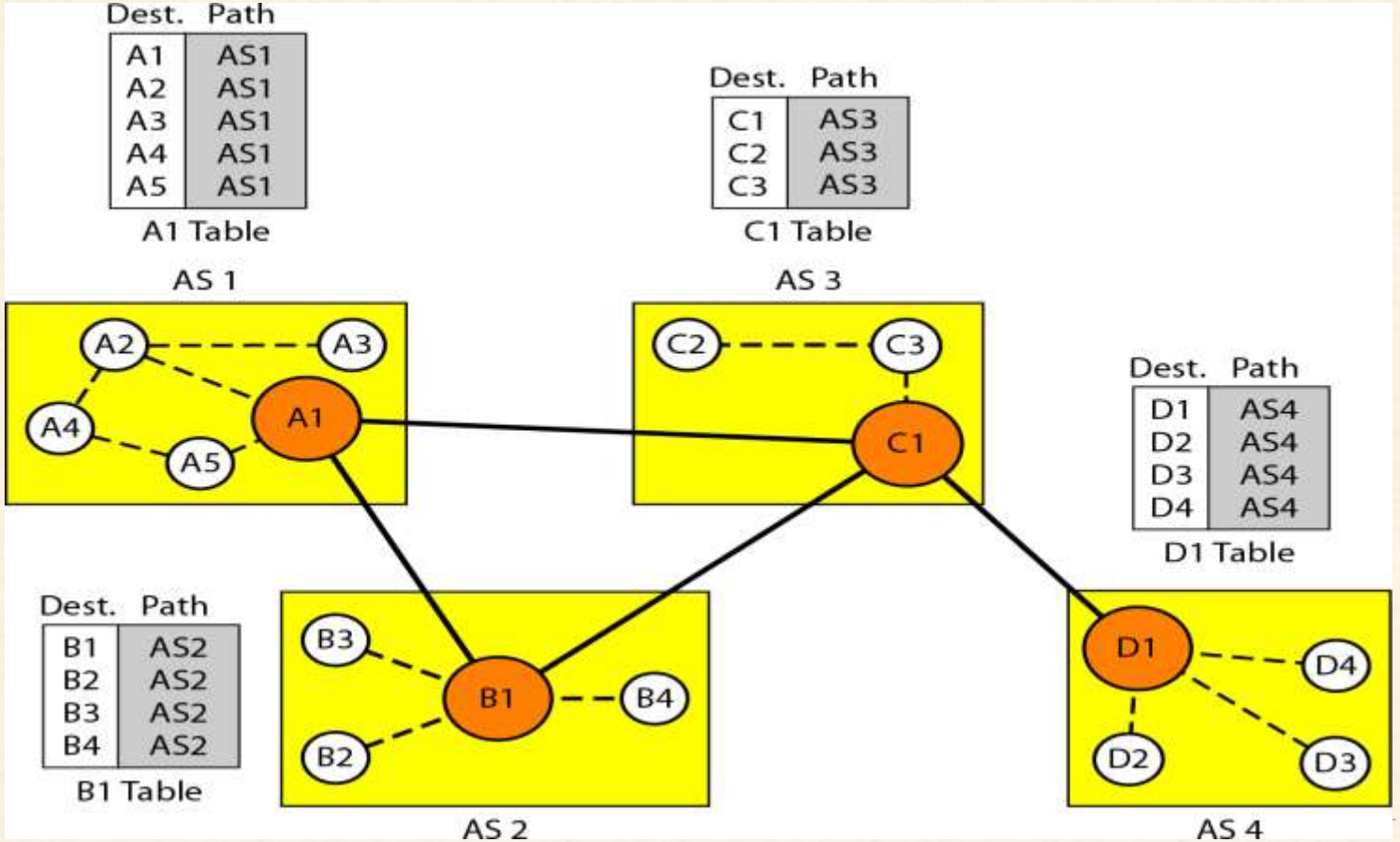
Path Vector Routing

- Path vector routing proved to be useful for **inter domain routing.** The principle of path vector routing is similar to that of distance vector routing.
- **In path vector routing, we assume that there is one node** (there can be more, but one is enough for our conceptual discussion) **in each AS** that acts on behalf of the entire AS. Let us call it the **speaker node**.
- The speaker node in an AS creates a routing table and advertises it to speaker nodes in the neighboring ASs. The idea is the same as for distance vector routing except that only speaker nodes in each AS can communicate with each other. However, what is advertised is different.
- A speaker node advertises the path, not the metric of the nodes, in its autonomous system or other autonomous systems

Path Vector Routing

Initialization

Initial routing tables in path vector routing



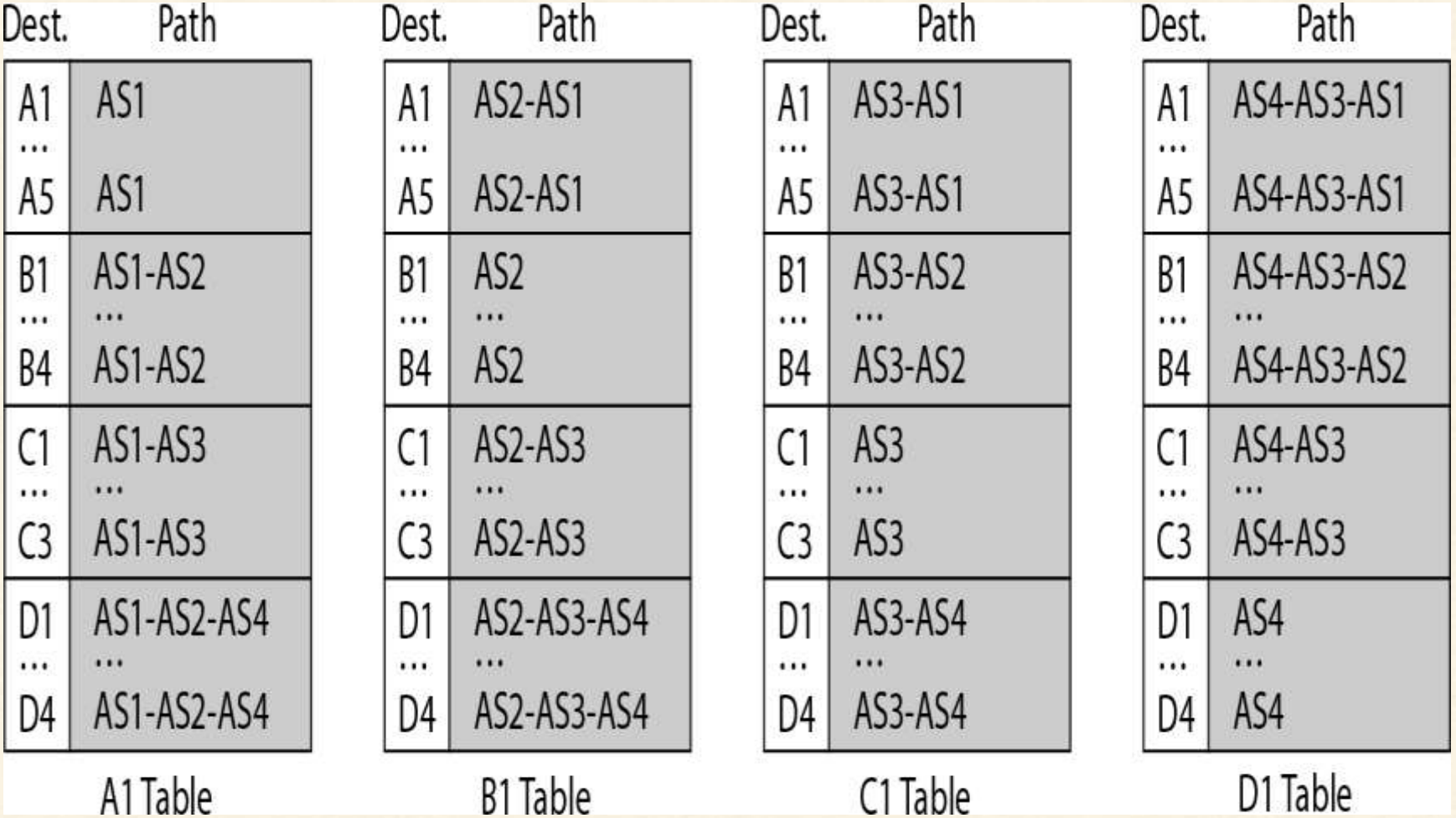
Path Vector Routing

Sharing

- Just as in distance vector routing, in path vector routing, a speaker in an autonomous system shares its table with immediate neighbors.

- In Figure, node A1 shares its table with nodes B1 and C1. Node C1 shares its table with nodes D1, B1, and A1. Node B1 shares its table with C1 and A1. Node D1 shares its table with C1.

Path Vector Routing



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

A1 Table

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

B1 Table

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

C1 Table

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

D1 Table

Path Vector Routing

Updating

- When a speaker node receives a two-column table from a neighbor, it updates its own table by adding the nodes that are not in its routing table and adding its own autonomous system and the autonomous system that sent the table.
- After a while each speaker has a table and knows how to reach each node in other Ass.

1. **Loop prevention**.
2. **Policy routing**.
3. **Optimum path.**

Hierarchical Routing

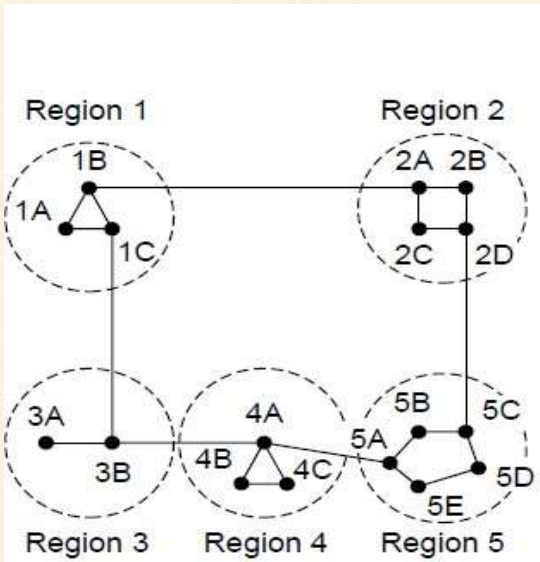
- As networks grow in size, the router routing tables grow proportionally. Not only is router memory consumed by ever-increasing tables, but more CPU time is needed to scan them and more bandwidth is needed to send status reports about them.

- At a certain point, the network may grow to the point where it is no longer feasible for every router to have an entry for every other router, so the routing will have to be done hierarchically, as it is in the telephone network.

- When hierarchical routing is used, the routers are divided into what we will call regions. Each router knows all the details about how to route packets to destinations within its own region but knows nothing about the internal structure of other regions.

Hierarchical Routing

- For huge networks, a two-level hierarchy may be insufficient; it may be necessary to group the regions into clusters, the clusters into zones, the zones into groups, and so on, until we run out of names for aggregations



(a)

Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)

IPv4 ADDRESSES

- An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a device (for example, a computer or a router) to the Internet.

- On the other hand, if a device operating at the network layer has **m** connections to the Internet, it needs to have **m** addresses. A router is such a device which needs as many IP addresses as the number of ports are there in it.

- Two devices on the Internet can never have the same address at the same time. But by using some strategies, an address may be assigned to a device for a time period and then taken away and assigned to another device.

IPv4 ADDRESSES

Address Space:

A protocol such as IPv4 that defines addresses has an address space.

- ***An address space is the total number of addresses used by the protocol***. If a protocol uses ***N* bits** to define an address, the **address space is 2^N** because each bit can have two different values (0 or 1) and *N* bits can have 2^N values.

- **IPv4 uses 32-bit addresses**, which means that the **address space is 2 POWER 32 or**

4,294,967,296 (more than 4 billion).

- This means that, theoretically, if there were no restrictions, more than 4 billion devices could be connected to the Internet. But the actual number is much less because of the restrictions imposed on the addresses.

IPv4 ADDRESSES

IPv4 Address Notations:

There are two prevalent notations to show an IPv4 address:

a. Binary notation and

b. Dotted decimal notation.

a. Binary Notation

In binary notation, the IPv4 address is displayed as 32 bits. Each octet is often referred to as a byte. So it is common to hear an IPv4 address referred to as a 32-bit address or a 4-byte address. The following is an example of an IPv4 address in binary notation:

01110101 10010101 00011101 00000010

IPv4 ADDRESSES

a. Dotted-Decimal Notation

To make the IPv4 address more compact and easier to read, Internet addresses are usually written in decimal form with a decimal point (dot) separating the bytes. The following is the dotted decimal notation of the above address:

117.149.29.2

Figure 19.1 shows an IPv4 address in both binary and dotted-decimal notation. Note that because each byte (octet) is 8 bits, each number in dotted-decimal notation is a value ranging from 0 to 255.

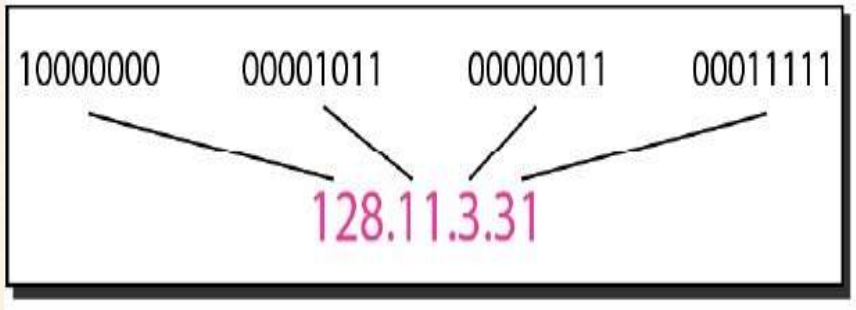


Figure 19.1 Dotted-decimal notation and binary notation for an IPv4 address

IPv4 ADDRESSES

Example:

Change the following IPv4 addresses from binary notation to dotted-decimal notation.

a. 10000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

Solution

We replace each group of 8 bits with its equivalent decimal number and add dots for separation.

a. 129.11.11.239

b. 193.131.27.255

IPv4 ADDRESSES

Example:

Change the following IPv4 addresses from dotted-decimal notation to binary notation.

a. 111.56.45.78

b. 221.34.7.82

Solution

We replace each decimal number with its binary equivalent.

a. 01101111 00111000 00101101 01001110

b. 11011101 00100010 00000111 01010010

Types of IPv4 Addressing Schemes

There are two types of IPv4 addressing schemes:

1.Classful Addressing

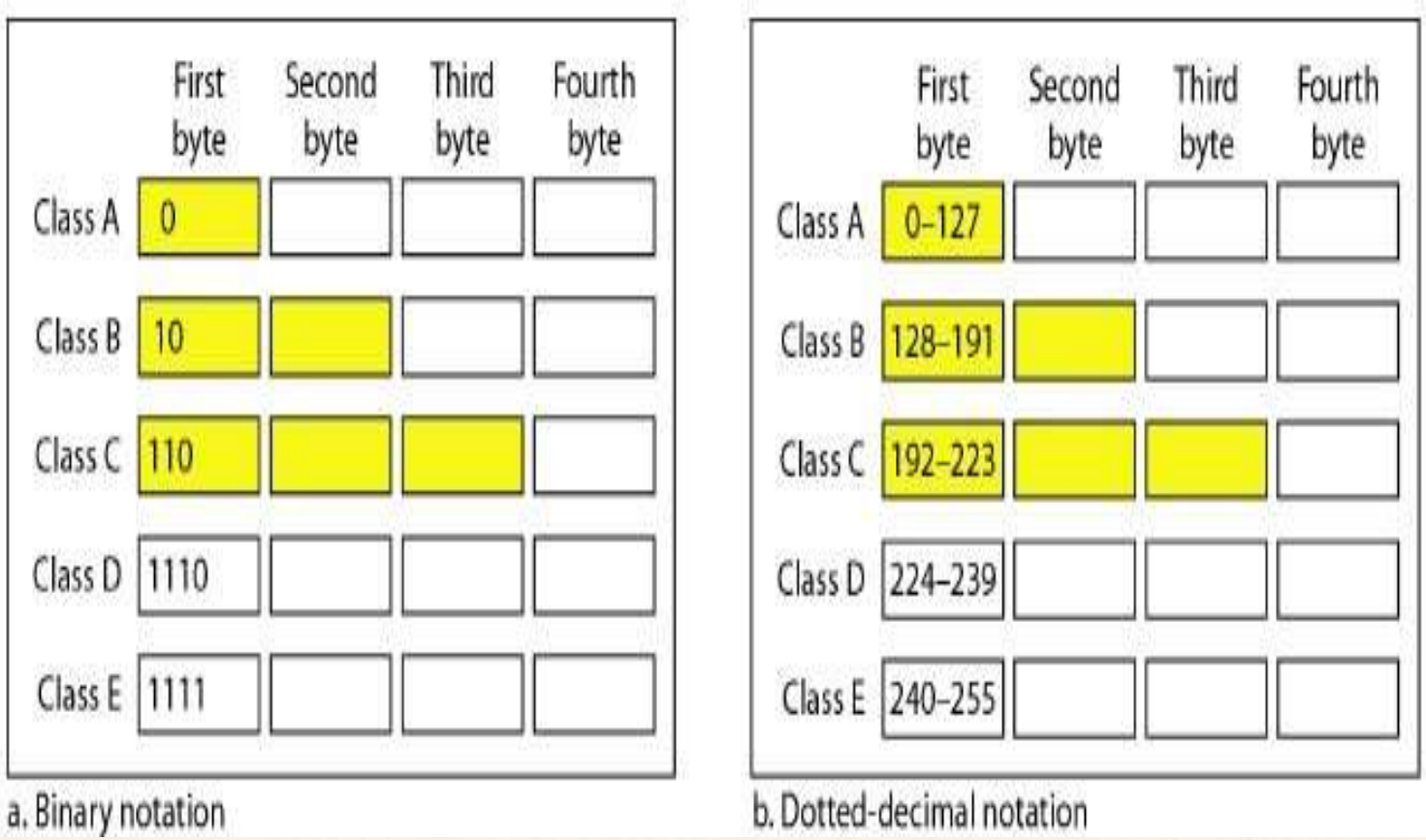
2.Classless Addressing

Classful Addressing

- IPv4 addressing, at its inception, used the concept of **classes.** This architecture is called **classful addressing.**
- In classful addressing, the address space is divided into five classes: **A, B, C, D, and E.**
- Each class occupies some part of the address space.
- We can find the class of an address when given the address in binary notation or dotted-decimal notation.
- If the address is given in binary notation, **the first few bits** can immediately tell us the class of the address.
- If the address is given in decimal-dotted notation**, the first byte** defines the class.

Classful Addressing

- Both methods are shown in Figure below.



Classful Addressing

Example:

Find the class of each address.

a. 00000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

c. 14.23.120.8

d. 252.5.15.111

Solution:

a. The first bit is 0. This is a class A address.
b. The first 2 bits are 1; the third bit is 0. This is a class C address.
c. The first byte is 14 (between 0 and 127); the class is A.
d. The first byte is 252 (between 240 and 255); the class is E.

Classful Addressing

Netid and Hostid

- In classful addressing, an IP address in class A, B, or C is divided into **netid** and **hostid.**
- These parts are of varying lengths, depending on the class of the address. Figure below shows some netid and hostid bytes.
- The netid is in color, the hostid is in white. Note that the concept does not apply to classes D and E.
- **In class A, one byte** defines the **netid** and **three bytes** define the **hostid.**
- **In class B, two bytes** define the **netid** and **two bytes** define the **hostid.**
- **In class C**, **three bytes** define the **netid** and **one byte** define the **hostid.**

Classful Addressing

Mask

A mask (also called the default mask) is a 32-bit number made of contiguous 1s followed by contiguous 0s. The masks for classes A, B, and C are shown in Table below. The concept does not apply to **classes D and E.**

- The mask can help us to find the netid and the hostid. For example, the mask for a class A address has eight 1s, which means the first 8 bits of any address in class A define the netid; the next 24 bits define the hostid.
- The last column of Table shows the mask in the form **/n** where n can be 8, 16, or 24 in classful addressing.
- This notation is also called slash notation or **Classless Interdomain Routing (CIDR) notation.**

Classful Addressing

Table : Default masks for classful addressing

Class	Binary	Dotted-Decimal	CIDR
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/16
C	11111111 11111111 11111111 00000000	255.255.255.0	/24

Classful Addressing

Classes and Blocks

- One problem with classful addressing is that each class is divided into **a fixed number of blocks** with each block having a **fixed size** as shown in Table below.

Class	Number of Blocks(How many Networks?)	Block Size(How many Hosts?)	Application
A	2^7=128	2^{24}=16,777,216	Unicast
B	2^{14}=16,384	2^{16}=65,536	Unicast
C	2^{21}=2,097,152	2^8=256	Unicast
D	Not Defined	Not Defined	Multicast
E	Not Defined	Not Defined	Reserved

Classful Addressing

- **Class A** addresses were designed for **large organizations** with a large number of attached hosts or routers.
- **Class B** addresses were designed for **midsize organizations** with tens of thousands of attached hosts or routers.
- **Class C** addresses were designed for **small organizations** with a small number of attached hosts or routers.

Classful Addressing

Class	Leading Bit	Bits For Network ID	No. of Networks	Bits For HOST ID	No. of HOSTS per Network	Total Addresses in the Class	First IP Address	First HOST Address	Last IP Address	Last HOST Address	Default Subnet Mask
Class A	0	7	2^7	24	$2^{24} - 2$	2^{31}	0.0.0.0	0.0.0.1	127.255.255.255	127.255.255.254	255.0.0.0
Class B	10	14	2^{14}	16	$2^{16} - 2$	2^{30}	128.0.0.0	128.0.0.1	191.255.255.255	191.255.255.254	255.255.0.0
Class C	110	21	2^{21}	8	$2^8 - 2$	2^{29}	192.0.0.0	192.0.0.1	223.255.255.255	223.255.255.254	255.255.255.0
Class D	1110	Not Defined	Not Defined	Not Defined	Not Defined	2^{28}	224.0.0.0	224.0.0.1	239.255.255.255	223.255.255.254	Not Defined
Class E	1111	Not Defined	Not Defined	Not Defined	Not Defined	2^{28}	240.0.0.0	240.0.0.1	255.255.255.255	223.255.255.254	Not Defined

Classful Addressing

Limitations of Classful Addressing:

- A block in **class A** address is **too large** for almost any organization. This means most of the addresses in class A were wasted and were not used.
- A block in **class B** is also **very large**, probably too large for many of the organizations that received a class B block.
- A block **in class C** is probably **too small** for many organizations.
- **Class D** addresses were designed for **multicasting**. Each address in this class is used to define one group of hosts on the Internet. The Internet authorities wrongly predicted a need for 268,435,456 groups. This never happened and many addresses were wasted here too.
- And lastly, the **class E** addresses were **reserved for future use**; only a few were used, resulting in another waste of addresses.

Classful Addressing

Address Depletion Problem

- The fast growth of the Internet led to the near **depletion of the available addresses in classful addressing scheme.** Yet the number of devices on the Internet is much less than the 2 power 32 address space.

- We have run out of class A and B addresses, and a class C block is too small for most midsize organizations.

•One solution that has alleviated the problem is the idea of **classless addressing.**

•Classful addressing, which is almost **obsolete**, is replaced with classless addressing.

Classless IP Addressing

- Classless IP addressing makes the **allocation of IP Addresses more flexible** which is also known as **Classless Inter Domain Routing (CIDR).**
- In Classless IP addressing, **CIDR block contains the required number of IP** Addresses as demanded by the user.

Purpose of Classless Addressing

There were two major problems with Classful IP addressing

1. **Wastage of IP Addresses:** As there are almost **1-crore host IP** addresses in class A. But these are **too much hosts IP** addresses for a single organization. So, it is **wastage of IP** addresses.
2. **No Flexibility:** If any user required almost 1000 IP address then there is **no class of networking** which will provide exact **1000 IP address even after subnetting.** If Class A provides 1000 IP address after subneting to that organization then still a lot of **chances of wastage of remaining IP** addresses. So, it does not provide the flexibility.

Solution of Classful Problem

- **Subneting in Classless IP** addressing is used for network flexibility. It provides the **exact number of IP addresses** as one organization or network required.

Classless IP Addressing

CIDR Block

- When a user asks for specific number of IP Addresses,
- CIDR **dynamically assigns** a block of IP Addresses based on certain rules.
- This block contains the required number of **IP Addresses as demanded** by the user.
- This block of IP Addresses is called as a **CIDR block**.

Notation of CIDR

CIDR IP Addresses look like the following - **a.b.c.d / n**

- They end with a slash followed by a number called as **IP network prefix.**
- IP network prefix tells the **number of bits** used for the **identification of network.**
- Remaining bits are used for the **identification of hosts** in the network.

Example of Classless IP addressing

- An example of CIDR IP Address is given : **90.10.12.20 / 26**

Where,

- 26 bits represents the network.
- Remaining bits (**6 out of 32** in IPv4) are used for the identification of hosts in the network.

Classless IP Addressing

Rules For Creating CIDR Block

- **Rule-01:** All the IP Addresses in the classless Addressing **(CIDR Block)** must be **contiguous.**
- **Rule-02:** Number of IP addresses (for hosts) in a CIDR block must be in the **power of 2 (i.e. $2^1, 2^2, 2^3, 2^4$ and so on).**
- **Rule-03:** First IP Address of the block in CIDR, must be **divisible by the size of the block.**

Rule 3 Explanation:

Any binary pattern of IP is divisible by 2^n, if and only if its least "n" significant bits are 0.

Examples: Consider a binary pattern of an IP address **100.2.3.64**

01100100.00000010.00000011.01000000

- Above IP is divisible by either **$2^1, 2^2$ $2^3, 2^4$,2^5or 2^6** Because its least 6 significant bits are zero.
- It is divisible by 2^6 since its least significant 6 bits are zero.
- Above IP is not divisible by 2^7 because its least 7 significant bits are not zero.

So, if the size of CIDR Block is **$2^1, 2^2$ $2^3, 2^4$,2^5 and 2^6** then Rule 3 is valid for above IP otherwise if size of CIRD Block is greater than 2^7 then Rule 3 in not valid for above IP.

Classless IP Addressing

Example: Consider a classless IP address is **21.11.40.35 / 28.** Find out the range of IP Addresses in the CIDR block or classless block.

Solution

- 28 bits are used for **network identification**. Remaining 4 bits are used for **host's identification** in the classless
- Given CIDR IP Address may be represented as

  00010101.00001011.00101000.00100011 / 28

So,

- First IP Address = **00010101.00001011.00101000.00100000= 21.11.40.32**
- Last IP Address = **00010101.00001011.00101000.00101111= 21.11.40.47**
- Put all Host bits to zero for first IP, and put all host bits to "1" for Last IP of Network.

Note: Direct broadcast IP address of a network is always the last IP of that Network.

Thus, Range of IP Addresses = **[21.11.40.32 , 21.11.40.47]**

Now Check all Three rules of CIDR block

- **Rule 01:** As all IP's are contiguous **(21.11.40.32 to 21.11.40.47).** So Rule 1 is satisfied.
- **Rule 02:** Range (21.11.40.32 to 21.11.40.47) **contains 16 IP's which is the Power of 2.** So Rule 2 is satisfied.
- **Rule 03:** According to third rule, First IP must be **divisible by block size.** As the Network Size is 28 and last 4 bits of first IP are Zero. So, it is divisible. Hence, Rule 3 is also satisfied.
- As all 3 rules are valid, so, **Classless block with 16 IP's is Valid.**

Classless IP Addressing

EXAMPLE:

Consider a block of IP Addresses ranging from **150.10.20.64 to 150.10.20.127.**

- Is it a CIDR block?
- If yes, give the CIDR representation.

Solution-

For any given block to be a CIDR block, 3 rules must be satisfied-

Rule-01:

- According to Rule-01, all the IP Addresses must be contiguous.
- Clearly, all the given IP Addresses are contiguous. **So, Rule-01 is satisfied.**

Rule-02:

- According to Rule-02, size of the block must be presentable as 2^n.
- Number of IP Addresses in given block = 64.
- Size of the block = 64 which can be represented as 2^6. **So, Rule-02 is satisfied.**

Classless IP Addressing

Rule-03:

- According to Rule-03, first IP Address must be divisible by size of the block.
- So, 150.10.20.64 must be divisible by 2^6.
- 150.10.20.64 = **150.10.20.01000000 is divisible by 2^6** since its 6 least significant bits are zero.
- So, Rule-03 is satisfied.

Since all the rules are satisfied, therefore given block is a CIDR block.

CIDR Representation-

We have-

- Size of the block = Total number of IP Addresses = 2^6.
- To have 2^6 total number of IP Addresses, 6 bits are required in the Host ID part.
- So, Number of bits in the Network ID part = 32 – 6 = 26.

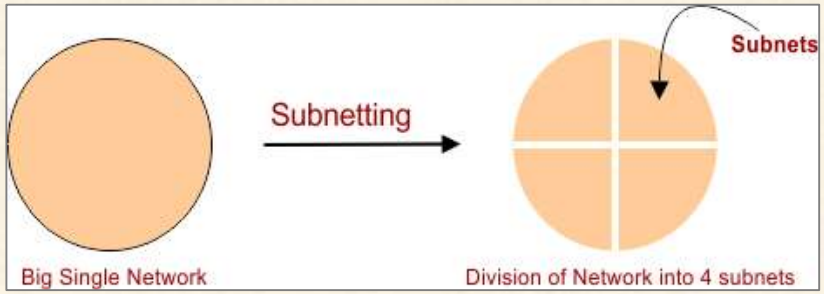
Thus, **CIDR Representation = 150.10.20.64 / 26**

Subnetting

- Subnetting is a process in which a single network is dividing into multiple sub-networks, also called as **subnets**.

Example

- Following figure shows the sub networks of a large single network into 4 smaller sub networks.

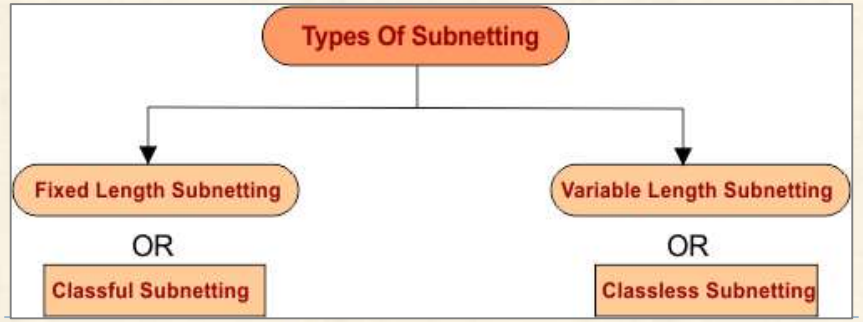


Subnet ID

- Each sub network has its unique network ID known as its **Subnet ID**.
- The subnet ID is created by borrowing some bits from the part of Host ID.
- The number of bits borrowed from hosts depends on the number of subnets created.

Types of Subnetting

- Subnetting of a network can be achieved through the following methods



Subnetting

1. Fixed Length Subnetting

- Fixed length subnetting also called as **classful subnetting.** Fixed length subnetting hold the following properties.
- Sizes of all sub networks and Subnets of all **sub networks are same.**
- All the sub networks have **equal number of hosts.**

2. Variable Length Subnetting

- Variable length subnetting also called as **classless subnetting**. Variable length subnetting hold the following properties.
- Sizes of all sub networks are not same and Subnets of all **sub networks are not same.**
- All the sub networks **do not have the equal number of hosts.**

Advantages of Subnetting

- Subnetting improves the security because the **administration and maintenance** of sub networks is easy.
- In simple words, management and maintenance of entire university is tough as compare to its different departments.

Subnetting

Disadvantages of Subnetting

Point-01:

- Subnetting leads towards the **loss of IP Addresses.**
- Two IP Addresses are always **wasted for every sub-network** (subnet).In subnetting, One IP Address is wasted for its network address and other for its direct broadcasting address.

Point-02:

- Subnetting leads toward more complicated communication process as compare to without subnetting communication.

After subnetting, the communication is done through the following 4 steps

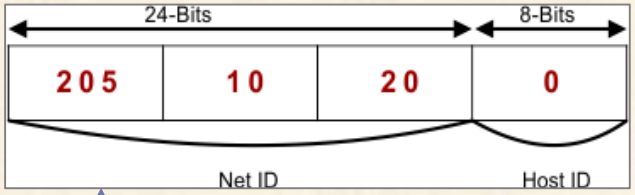
1. First Identifying the network
2. Second Identifying the sub network
3. Third Identifying the host
4. And in the last, Identifying the process

Subnetting in Classful

As we already know that, The classful subnetting is also called **Fixed length subnetting**. The Fixed length subnetting contain the following properties.

- Sizes of all sub networks and Subnets of all sub networks are same.
- All the sub networks have equal number of hosts.

Let understand with example,



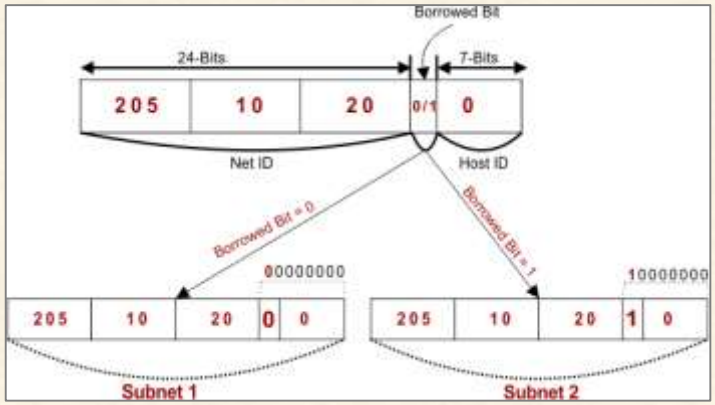
Example-01

Suppose,

- We have a big single big network having IP Address **205.10.20.0.**
- We want subnetting. So, divide this network into 2 subnets.
- As given IP belongs to Class C So, **24 bits are used for net ID and 8 bits are used for Host ID.**
- For creating two subnets (sub networks) and to represent their subnet IDs, we require **1 borrowed-bit from Host part.**

So,

- We **borrowed one bit** from the Host part.
- If borrowed **bit = 0**, then it will represent the first subnet.
- If borrowed **bit = 1**, then it will represent the second subnet.

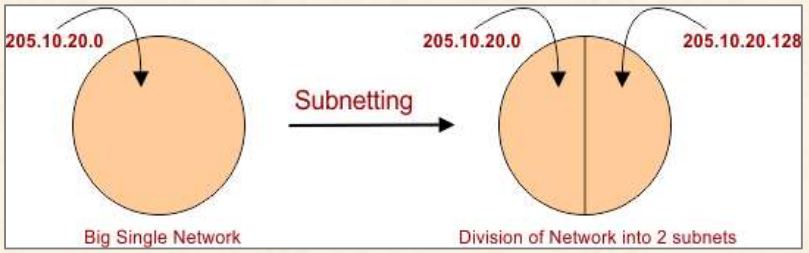


Note: After borrowing one bit, Host ID part remains with only 7 bits.

Subnetting in Classful

IP Address of the two subnets are

- 10.20.00000000 = **205.10.20.0**
- 10.20.10000000 = **205.10.20.128**



For First Subnet

- IP Address of the subnet = **205.10.20.0**
- Total number of IP Addresses = **2^7= 128**
- Total number of hosts = **128 – 2 = 126**
- Range of IP Addresses = **[205.10.20.00000000 to 205.10.20.01111111] = [205.10.20.0 to 205.10.20.127]**
- Direct Broadcast Address = **205.10.20.01111111 = 205.10.20.127**
- Limited Broadcast Address = **255.255.255.255**

For Second Subnet

- IP Address of the subnet = **205.10.20.128**
- Total number of IP Addresses = **2^7= 128**
- Total number of hosts = **128 – 2 = 126**
- Range of IP Addresses = **[205.10.20.10000000 to 205.10.20.11111111] = [205.10.20.128 to 205.10.20.255]**
- Direct Broadcast Address = **205.10.20.11111111 = 205.10.20.255**
- Limited Broadcast Address = **255.255.255.255**

Subnetting in Classful

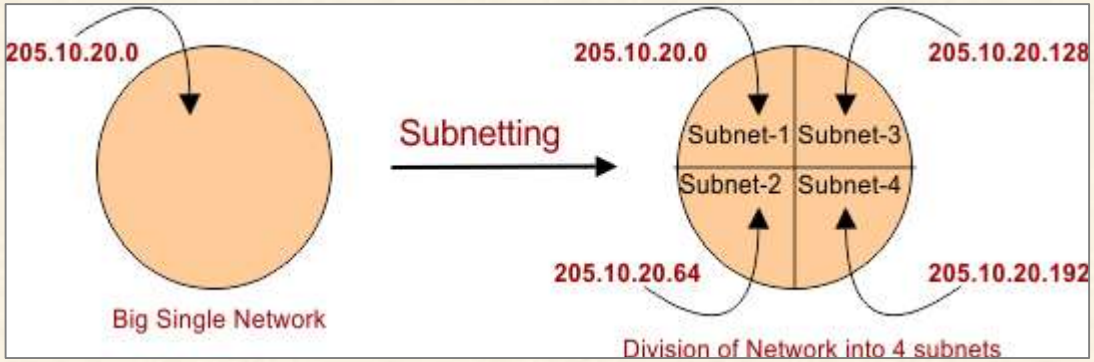
- **Important Note:** Number of subnets depends upon the number of borrowed bits. We can define this relation through the following formula.

Number of subnets = 2^n.

where "n" is number of borrowed bits. So, if number of borrowed bits are 2 then subnets will be $2^2 = 4$.

If given IP= 205.10.20.0. then first IP Address of each of four subnets are

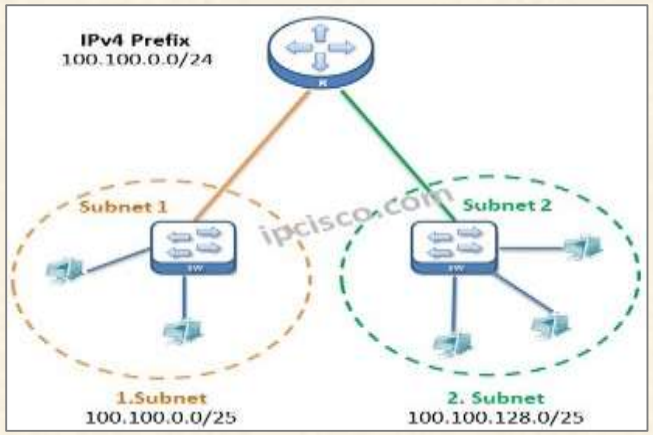
- 10.20.**00**000000 = **205.10.20.0**
- 10.20.**01**000000 = **205.10.20.64**
- 10.20.**10**000000 = **205.10.20.128**
- 10.20.**11**000000 = **20.10.20.192**



Subnetting in Classless

We have an IPv4 Prefix **100.100.0.0/24**. How can we divide this IPv4 prefix into two different subnet?

- As you can see below, we will use this **two subnets for different networks** that is connected to our router.



Firstly, lets write our IPv4 address in binary format.

- **Decimal :** 100.100.0.0/24
- **Binary :** 01100100.01100100.00000000.00000000 /24

- According to our prefix, our subnet **mask is /24.** This means that, our first **24 bits are network** and the remaining parts are host bits in this subnetting mask. Here, hosts bits are **32-24=8 bits.**
- So, to divide this network, we should **borrow some bits** from the host part. So, how many bits we will borrow? To determine this, we will check our subnet need. How many subnet do we need? For this question, we **need 2 subnets. So, we will borrow 1 bit** from host part. Why 1 bit?
- 2^0 = 1
- **2^1 = 2**

As you can see above, with **1 bit**, we can have **2 subnets** (**2^1=1**).

Subnetting in Classless

- After borrowing this address, our **network part will be 24+1=25 bits.**
- So, for these two new subnets, subnet mask will be **/25.** And **host parts will be 8-1= 7 bits.**
- To build these two subnets, we will change the borrowed bit only. We will use **0 and 1** for this borrowed bit.

As you can see below, it is the first bit of the last octet.

Original Prefix : 01100100.01100100.00000000.00000000

Original Prefix : 01100100.01100100.00000000.**0**0000000 **(Borrowed Bit)**

- **Subnet 0: 01100100.01100100.00000000.0**0000000
- **Subnet 1: 01100100.01100100.00000000.1**0000000

So, in decimal, our subnets will be like below:

- **Subnet 0 Decimal : 100.100.0.0/25**
- **Subnet 1 Decimal : 100.100.0.128/25**

Congestion Control Algorithm

•Too many packets present in (a part of) the network causes packet delay and loss that degrades performance. This situation is called **congestion**. The network and transport layers share the responsibility for handling congestion

•**Congestion control** is a mechanism to either prevent a congestion before it happens or remove the congestion after it happens.

•There are two congestion control algorithms:

1.Leaky Bucket

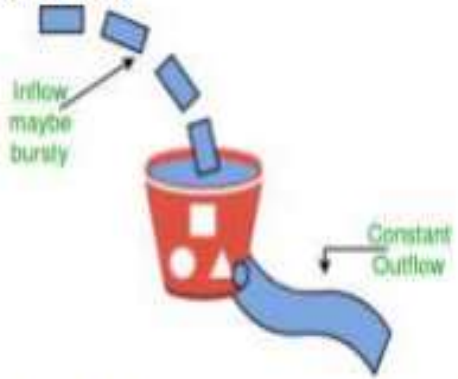
2.Token Bucket Algorithm

Congestion control algorithms

- **Leaky Bucket Algorithm**

Let us consider an example to understand

Imagine a bucket with a small hole in the bottom. No matter at what rate water enters the bucket, the outflow is at constant rate. When the bucket is full with water additional water entering spills over the sides and is lost.



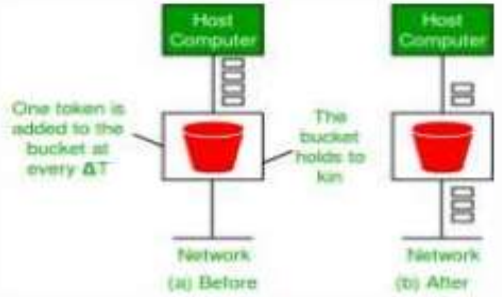
Similarly, each network interface contains a leaky bucket and the following **steps** are involved in leaky bucket algorithm:

1. When host wants to send packet, packet is thrown into the bucket.
2. The bucket leaks at a constant rate, meaning the network interface transmits packets at a constant rate.
3. Bursty traffic is converted to a uniform traffic by the leaky bucket.
4. In practice the bucket is a finite queue that outputs at a finite rate.

- **Token bucket Algorithm**
  - Need of token bucket Algorithm:-
  - The leaky bucket algorithm enforces output pattern at the average rate, no matter how bursty the traffic is.
  - So in order to deal with the bursty traffic we need a flexible algorithm so that the data is not lost.
  - One such algorithm is token bucket algorithm.

Steps of this algorithm can be described as follows:

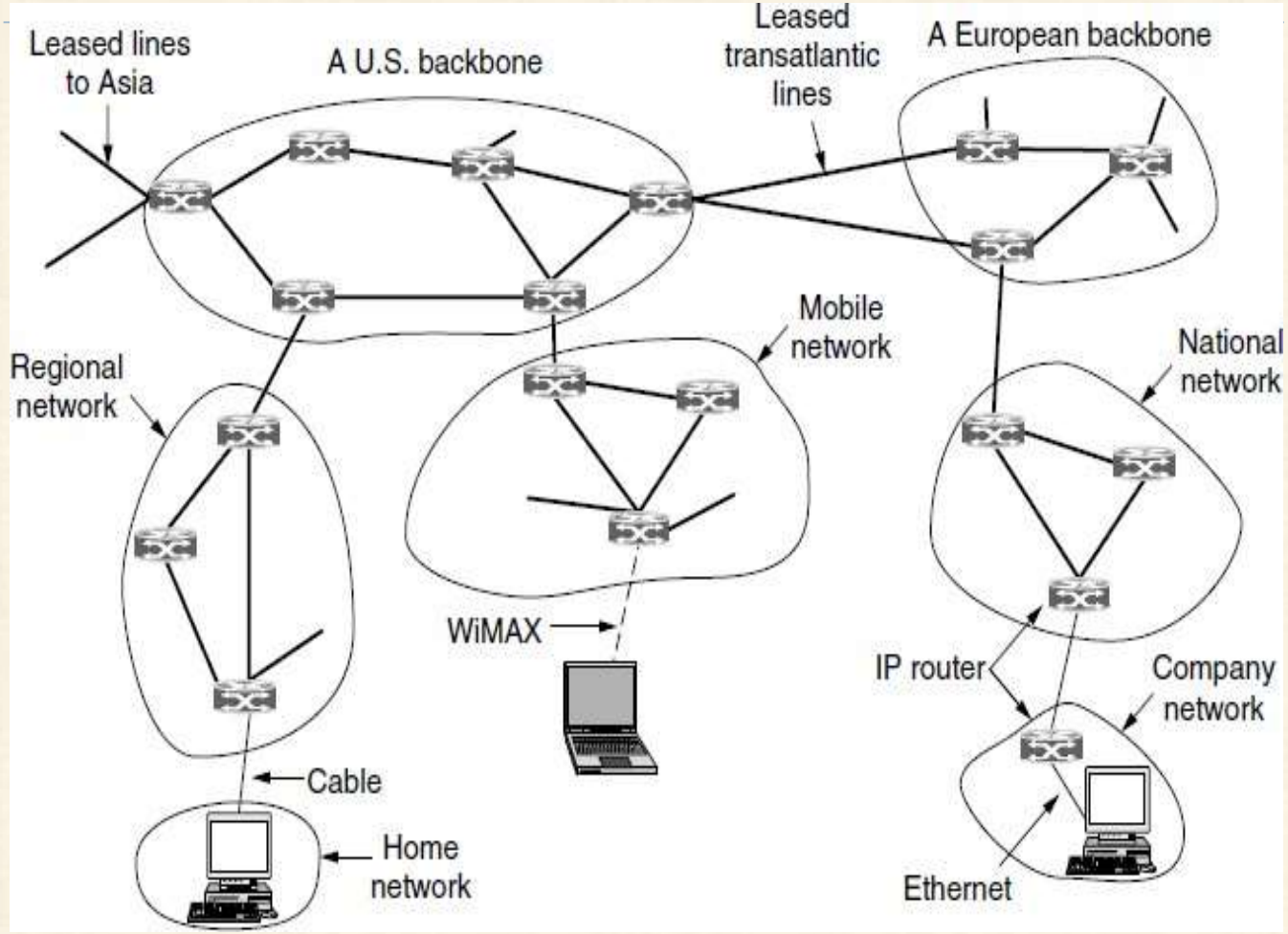
1. In regular intervals tokens are thrown into the bucket. *f*
2. The bucket has a maximum capacity. *f*
3. If there is a ready packet, a token is removed from the bucket, and the packet is sent.
4. If there is no token in the bucket, the packet cannot be sent.



The Network Layer In The Internet

There are some principles that drove its design in the past and made it the success that it is today. They are:

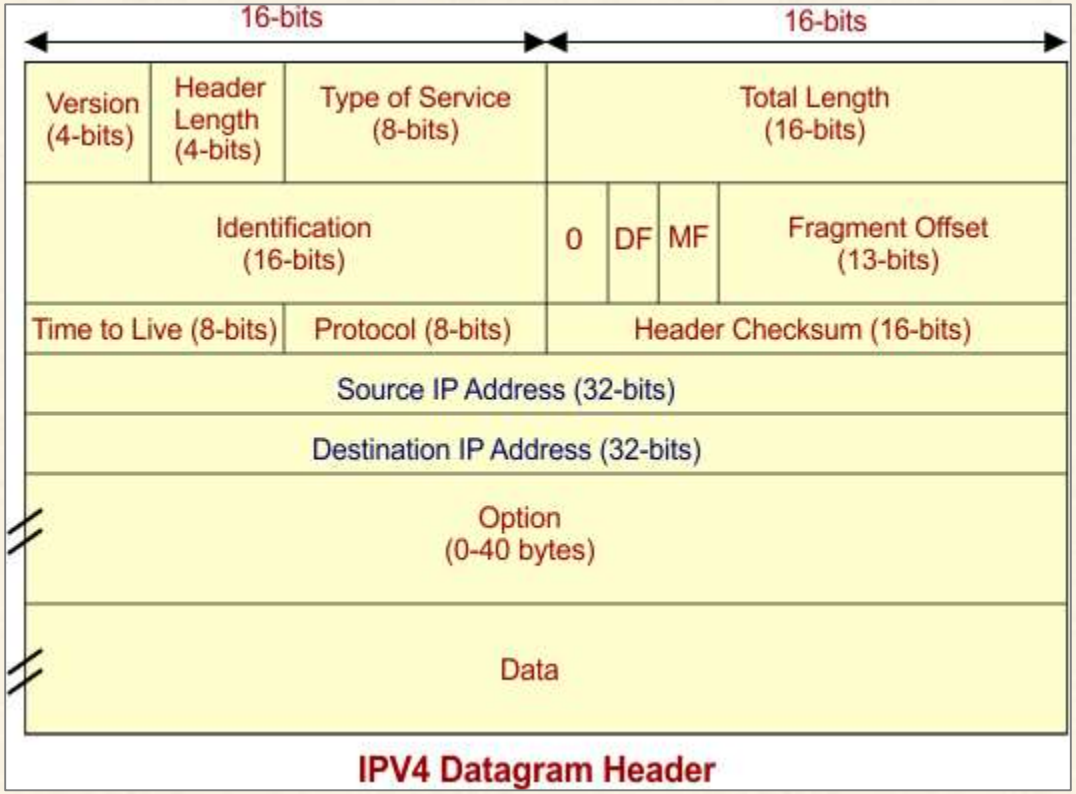
1. **Make sure it works.** Do not finalize the design or standard until multiple prototypes have successfully communicated with each other.
2. **Keep it simple.** When in doubt, use the simplest solution.
3. **Make clear choices.** If there are several ways of doing the same thing, choose only one.
4. **Exploit modularity.** This principle leads directly to the idea of having protocol stacks, each of whose layers is independent of all the other ones.
5. **Expect heterogeneity.** Different types of hardware, transmission facilities, and applications will occur on any large network. To handle them, the network design must be simple, general, and flexible.
6. **Avoid static options and parameters.**
7. **Look for a good design; it need not be perfect.**
8. **Be strict when sending and tolerant when receiving.**
9. **Think about scalability.**
10. **Consider performance and cost.**



The Internet is an interconnected collection of many networks.

IPv4 Datagram Header

IPv4 Datagram Header diagram is given below



IPv4 Datagram Header

IPV4 Datagram Header Attributes

1. VERSION: Version of the IP protocol is of 4 bits. These 4 bits are always fixed as **0100 to represent 4 in decimal for IPv4.**

2. HLEN: IV4P header length is of 4 bits. The minimum value for this field is **5 and the maximum is 15 bytes.**

- header length can be calculated by the following formula

  Header length = Header length field value x 4 bits

Examples

- If header length field contains decimal value 11 (represented in binary 1011) then Header length = 11 x 4 = 44 bytes

IPv4 Datagram Header

3. Type Of Service: it is 8 bit field that is used for Quality of Service. The division of 8-bits are explained under

Precedence (3 bits)	Delay (1 bit)	Throughput (1 bit)	Reliability (1 bit)	Cost (1 bit)	Reserved (1 bit)

- **Precedence (3 bits):** First 3 bits define the precedence. Precedence means priority i.e. immediate, routine etc. If a router is congested and needs to discard a packets, it will discard packets having lowest priority first. Bits values will be 0 or 1.
- **Delay (1 bit):** if we want a minimum delay in data packets then this field will be 1 otherwise 0. It Mostly in video calling where needs no delay.
- **Throughput (1 bit):** if need highly output then its field bit will be 1 otherwise 0.
- **Reliability (1 bit):** if need highly reliability then its field bit will be 1 otherwise 0. It is used where no data loss is tolerated.
- **Cost (1 bit):** if need low cost then its field bit will be 1 otherwise 0. It requires when to select the shortest path to its destination.
- **Last bit:** is reserved for future purpose which mostly controls the congestion Notification. Congestion mean it inform to sender to minimize the speed of sending data.

IPv4 Datagram Header

4. Total Length: It is Total length of the datagram. it is a 16 bit field which can represent $2^{16} = 65536$ value . It has minimum size of 20 bytes and max value of 65535 bytes.

Total length = Header length + Payload length

5. Identification:

- It is a 16-bit field. It is helpful for the **identification of the fragments** of an original datagram.
- When an IP datagram is fragmented, each fragmented datagram is assigned the **same identification header number.**
- This header-number is useful during the **re-assembly of fragmented datagrams.**

6. Flag Bits: It use 3 flag bits

- First flag bit is **Reserved**
- Second Flag bit **(DF Bit).** DF bit stands for **Do Not Fragment bit.** DF value may be **0 or 1.**
  - When **DF value is 0** then It gives the permission to the **intermediate devices (i.e. routers) to fragment the datagram** if required.
  - When **DF value is 1** then It indicates the **intermediate devices (i.e. router) not to fragment the datagram** at any cost.
- Third flag bit is **MF.** MF bit stands for **More Fragments bit.** MF value may be 0 or 1.
  - When **MF bit value is 0** then It tells to the receiver that the **current datagram-fragment is the last fragment** and no more segment will appear of same datagram.
  - When **MF bit value is 1** then it tells more fragments are **still to come after this fragment.** MF bit is set to 1 for all the fragments except the last one.

IPv4 Datagram Header

7. Fragment Offset: Fragment Offset is a 13 bit field. It tells the **position of a fragmented datagram** in the original un-fragmented IP datagram.

Fragment offset for a given fragmented IP datagram = Number of data bytes ahead of it in the original un-fragmented IP datagram4. Hence, The 1st fragmented datagram has a fragment offset of zero.

8. Time to live: It is 8-bit field which **prevents the datagram to go to loop.** If a datagram go to loop then congestion can happens which cause the problem. So, Time to live **(TTL) avoids in such stations.**

- According to TTL, 8-bit can represent 256 nodes. Therefore, datagram is self-loop can goes to 256 nodes, when datagram goes to a node, it is decremented by 1 in values. As the value reaches 0, the datagram is terminated.

9. Protocol: it is **an 8-bit number that defines what protocol is used inside the IP packet**. TCP, UDP, ICMP or IGMP protocols can be filtered on, although they are most common**. Protocol number of ICMP is 1 (in binary 00000001), IGMP is 2 (in binary 00000010), TCP is 6 and UDP is 17.**

10. Header Checksum: it is a 16 bits field. It is used for **checking errors in the datagram header**. At receiving end, it is used to **know that receiving data is corrupted or not** because data can lose or corrupt while passing through the network.

11. Source IP address: 32 bits IP address of the sender

12. Destination IP address: 32 bits IP address of the receiver

13. Option: Due to options field, datagram-header-size can be of variable length (20 bytes to 60 bytes). Optional information include as source route etc.

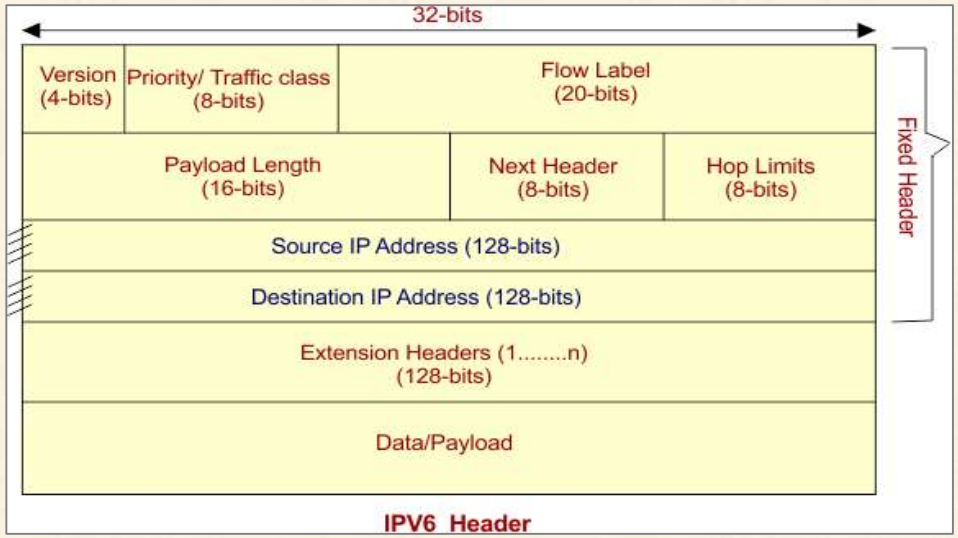
IPv6 Datagram Header

IPV6 is also a datagram and connectionless service like IPV4. But its **size and functionality** of working is a bit difference than IPV4.

IPv6 headers have **one Fixed Header and zero or more Optional** Headers.

- **Fixed header** is also called **base header**. It is the compulsory part of IPV6 header. All the **necessary information** which are compulsory for a router is kept in the Fixed Header. IPv6 fixed header is 40 bytes (320 bits) long.
- **Optional Headers** are also called **Extension Headers.** The optional Header holds some extra (optional) information that helps routers to understand how to **handle a packet or flow control**. With 40 bytes of fixed header, some extension headers can also be used which may increase the size of packet.

IPV6 Header contains the following information.



IPv6 Datagram Header

1. Version: **it is a** 4-bit field. It represents version IPV6 in binary **0110 (6).**

2. Traffic Class: **it is** 8-bits filed which also known as **priority**. These 8 bits are divided into two parts.

- The most significant **6 bits are used for Type of Service**. it replaces IPv4's 'type of service' field. It's basic purpose is to provide quality of service (QoS).
- The least significant **2 bits handles the packets in Congestion** (i.e. loop). Instead of dropping packets, last 2-bits use **Explicit Congestion Notification** (ECN) to handle the packets.

3. Flow Label: **it is** 20-bits field. It uses the **virtual circuit** for data transmission. In this way sequential flow of the packets is maintained belonging to a datagram. This field avoid to re-ordering of data packets because all data travel in a **single path**. It is designed for streaming/real-time media.

4. Payload Length: **It is** 16-bits field. It tells the routers about the size of payload which belongs to a particular packet. With 16 bits, up to 65535 bytes can be indicated of data.

IPv6 Datagram Header

5. Next Header: it is 8-bits field. It tells the type of **Extension Header** which are additionally used with base header to send more data or information's. Some extension headers are given below

Extension Header	Next Header Vaule	Description
Hop-by-Hop Options header	0	Read by all devices in transit network
Routing Header	43	Contains method to support making routing decision
Fragment Header	44	contains parameters of datagram fragmentations
Destination Options Header	60	Read by destination Device
Authentication Header	51	Information Regarding Security
Encapsulating security payload Header	50	encryption informations

- If extension header is used along with payload, then corresponding bits value is presented in this filed. It mean if **Routing header is used as extension header then 43(in bits)** is represented in Next header (8bit) field.

Note: Extension headers are optional, and are used if needed.

6. Hop Limit: it is 8-bits field. This field is used **to stop packet to loop** in the network infinitely. This is same as TTL in IPv4. The Hop-Limit field value is **decremented by 1 as it passes a link** (i.e. router). When the value of Hop-limit field reaches 0 the packet is discarded.

IPv6 Datagram Header

7. Source Address: (128-bits): This field indicates the address of originator of the packet.

8. Destination Address: (128-bits): This field provides the address of intended recipient of the packet.

9. This is the payload portion of the IPv6 packet.

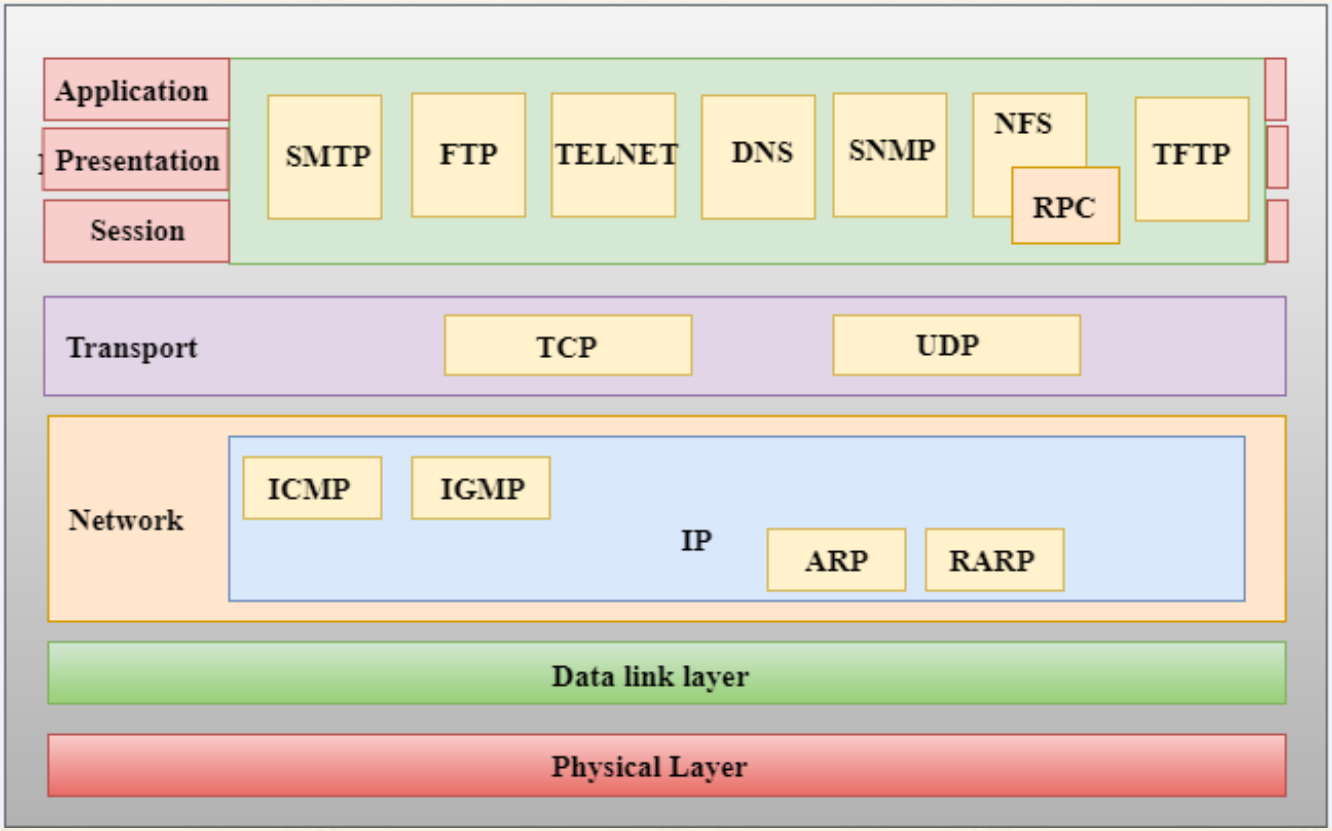
Note: We can say that an IPv6 address (128 bits) is 4 times larger than IPv4 (32bit) address but base header (40bytes) of an IPv6 is only 2 times larger than that of IPv4 header (20bytes).

IPv6 Packet Contains

- base header
- may contains zero, one or more extension headers
- Data, needs to transfer

Network Layer/IP Protocols

TCP/IP supports the following protocols



Network Layer/IP Protocols

ARP

ARP stands for Address Resolution Protocol.

- It is used to associate an **IP address with the MAC address.**
- Each device on the network is recognized by the MAC address imprinted on the NIC. Therefore, we can say that devices need the **MAC address for communication on a local area network**. MAC address can be changed easily.
- For example, if the NIC on a particular machine fails, the MAC address changes but IP address does not change. ARP is used to **find the MAC address of the node when an internet address is known**.

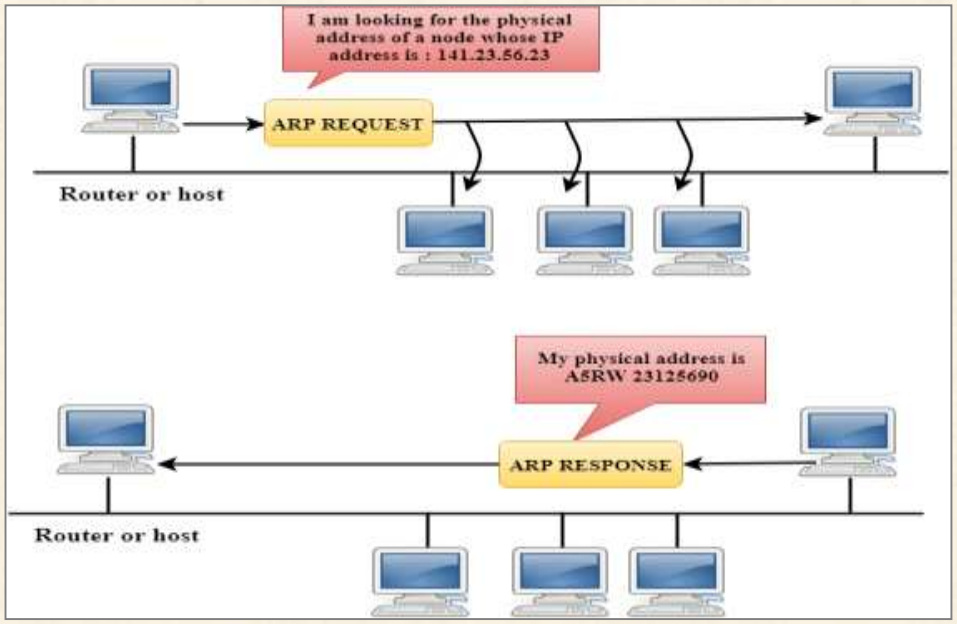
How ARP works

- If the host wants to know the physical address of another host on its network, then it sends an **ARP query packet** that includes the **IP address and broadcast it** over the network.
- Every host on the network receives and processes the ARP packet, but **only the intended recipient recognizes the IP address** and sends back the physical address.
- The host holding the datagram **adds the physical address to the cache memory and to the datagram header,** then sends back to the sender.

Network Layer/IP Protocols

How ARP works

- If the host wants to know the physical address of another host on its network, then it sends an **ARP query packet** that includes the **IP address and broadcast it** over the network.
- Every host on the network receives and processes the ARP packet, but **only the intended recipient recognizes the IP address** and sends back the physical address.
- The host holding the datagram **adds the physical address to the cache memory and to the datagram header,** then sends back to the sender.

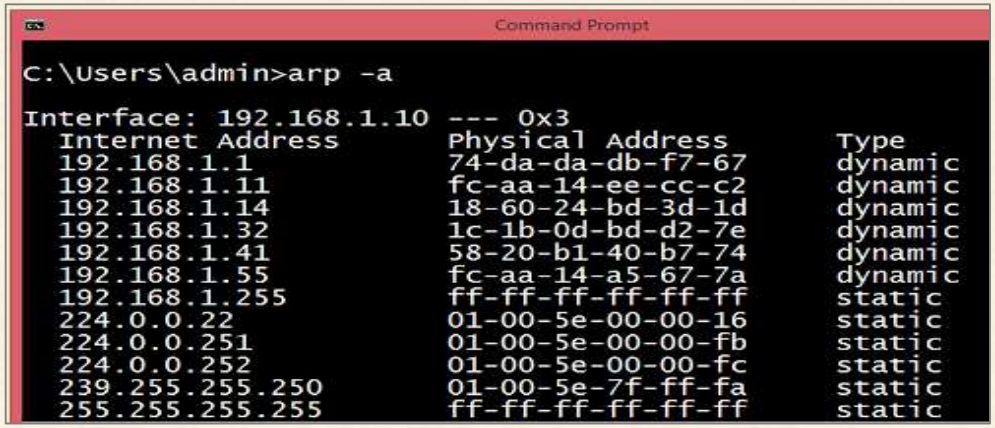


Network Layer/IP Protocols

Steps taken by ARP protocol

If a device wants to communicate with another device, the following steps are taken by the device:

- The device will first look at its **internet list, called the ARP cache** to check whether an IP address contains a matching MAC address or not. It will check the ARP cache in command prompt by using a command **arp -a**.
- If ARP cache is empty, then device **broadcast the message to the entire network** asking each device for a matching MAC address.
- The device that has the matching IP address **will then respond back** to the sender with its MAC address
- Once the MAC address is received by the device, then the **communication can take place** between two devices.
- If the device receives the MAC address, then the **MAC address gets stored in the ARP cache**. We can check the ARP cache in command prompt by using a command arp -a.



Network Layer/IP Protocols

There are two types of ARP entries:

- **Dynamic entry:** It is an entry which is **created automatically when the sender broadcast its message** to the entire network. Dynamic entries are not permanent, and they are removed periodically.

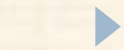
- **Static entry:** It is an entry where someone **manually enters the IP to MAC address association** by using the ARP command utility.

Internetworking

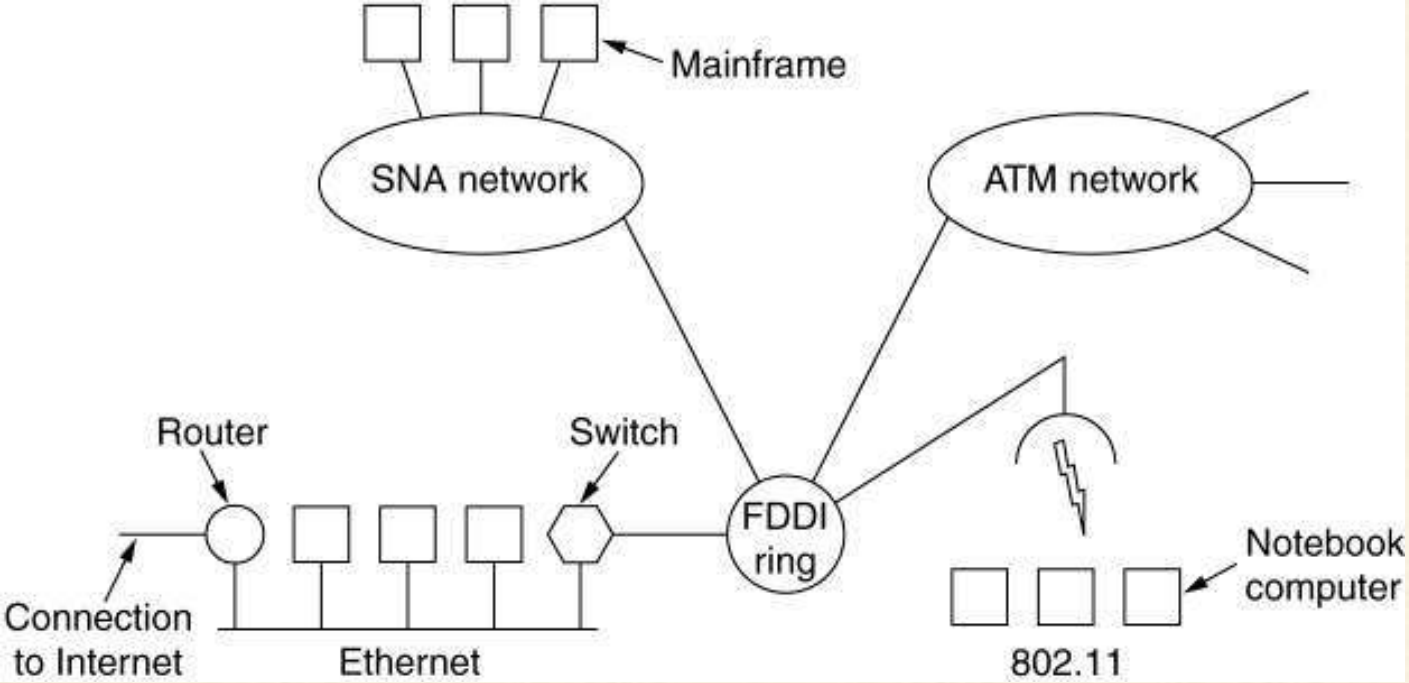
- Networks differ in various ways, so when multiple networks are interconnected problems can occur.
- Sometimes the problems can be finessed by tunneling a packet through a hostile network, but if the source and destination networks are different, this approach fails.
- When different networks have different maximum packet sizes, fragmentation may be called for.

Internetworking

- How Networks Differ
- How Networks Can Be Connected
- Concatenated Virtual Circuits
- Connectionless Internetworking
- Tunneling
- Internetwork Routing
- Fragmentation



Internetworking
Connecting Networks



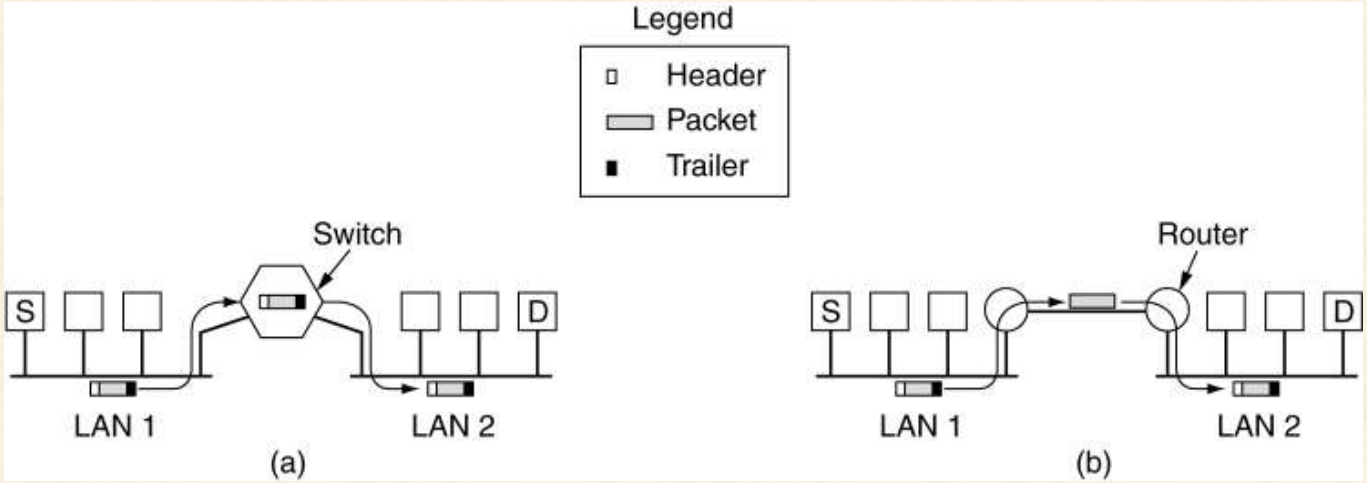
A collection of interconnected networks.

How Networks Differ

Item	Some Possibilities
Service offered	Connection oriented versus connectionless
Protocols	IP, IPX, SNA, ATM, MPLS, AppleTalk, etc.
Addressing	Flat (802) versus hierarchical (IP)
Multicasting	Present or absent (also broadcasting)
Packet size	Every network has its own maximum
Quality of service	Present or absent; many different kinds
Error handling	Reliable, ordered, and unordered delivery
Flow control	Sliding window, rate control, other, or none
Congestion control	Leaky bucket, token bucket, RED, choke packets, etc.
Security	Privacy rules, encryption, etc.
Parameters	Different timeouts, flow specifications, etc.
Accounting	By connect time, by packet, by byte, or not at all

Some of the many ways networks can differ.

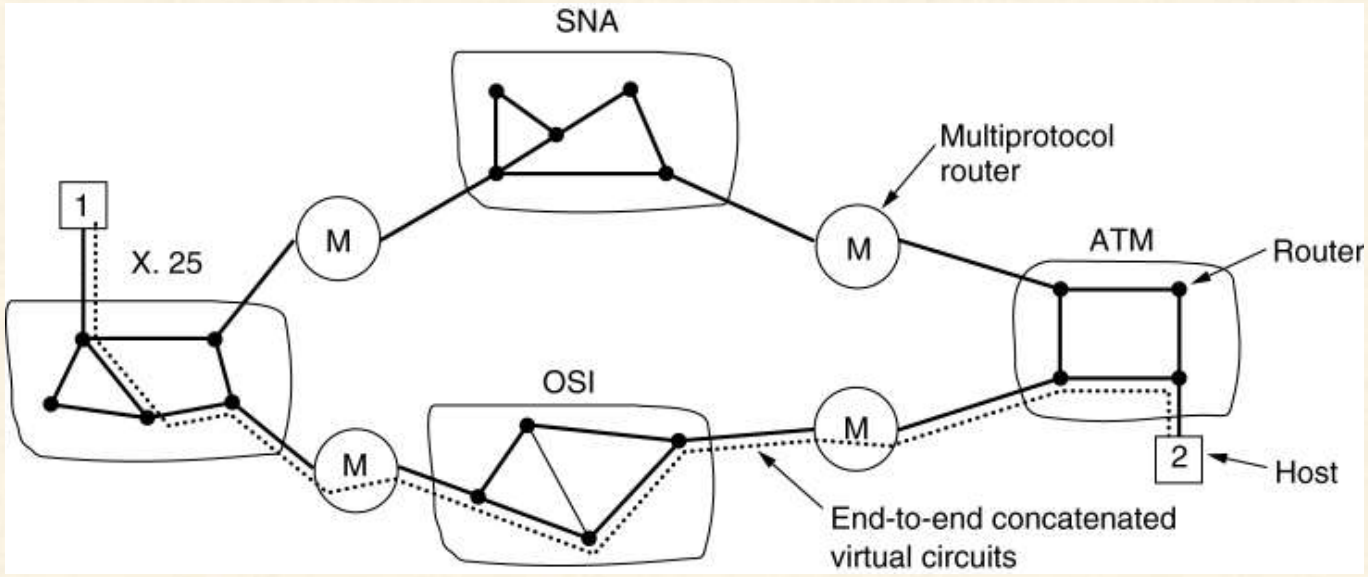
How Networks Can Be Connected



(a) Two Ethernets connected by a switch.

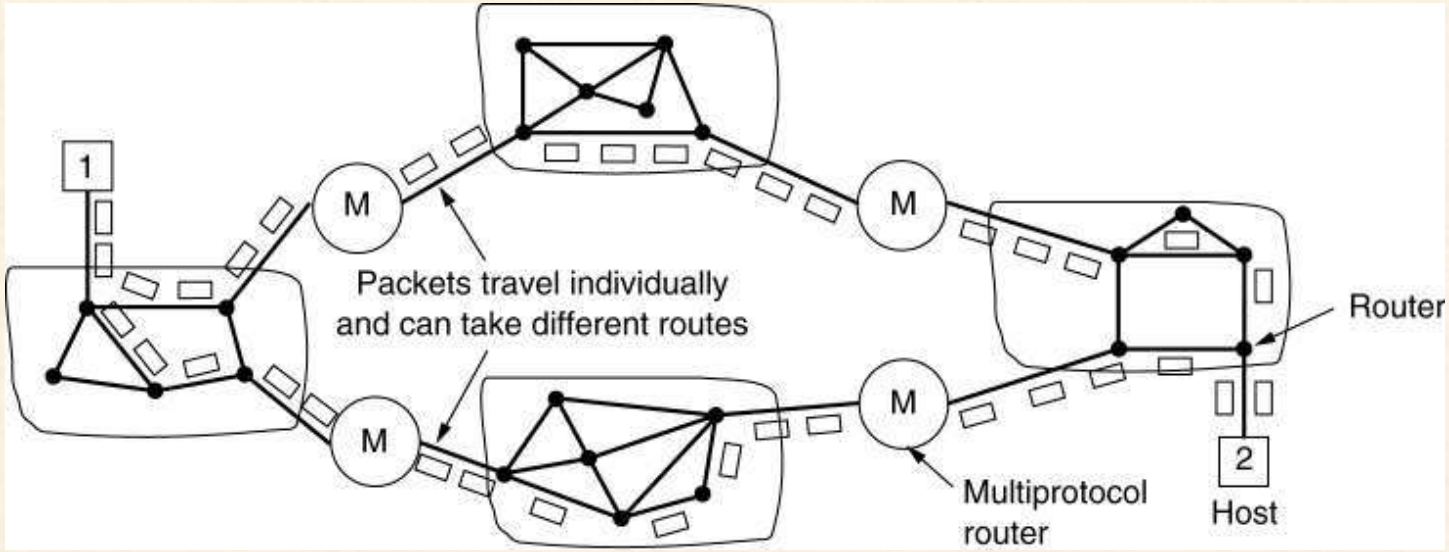
(b) Two Ethernets connected by routers.

Concatenated Virtual Circuits



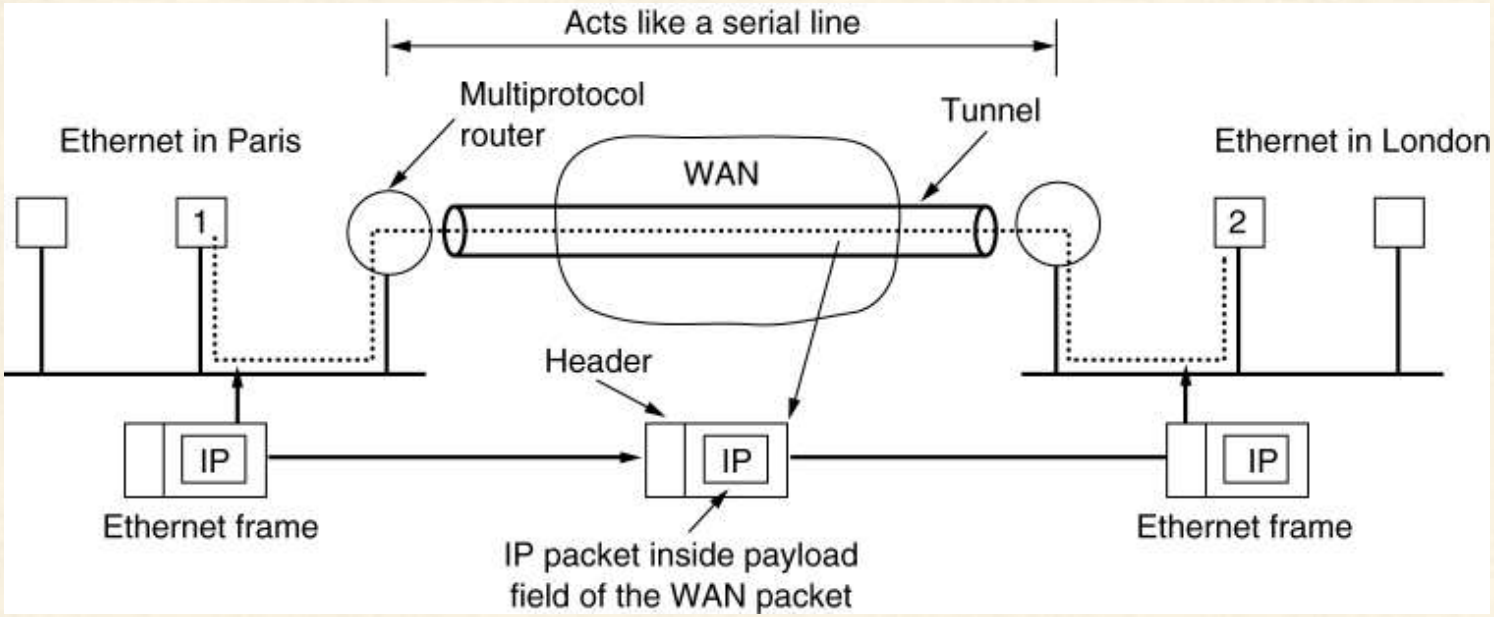
Internetworking using concatenated virtual circuits.

Connectionless Internetworking



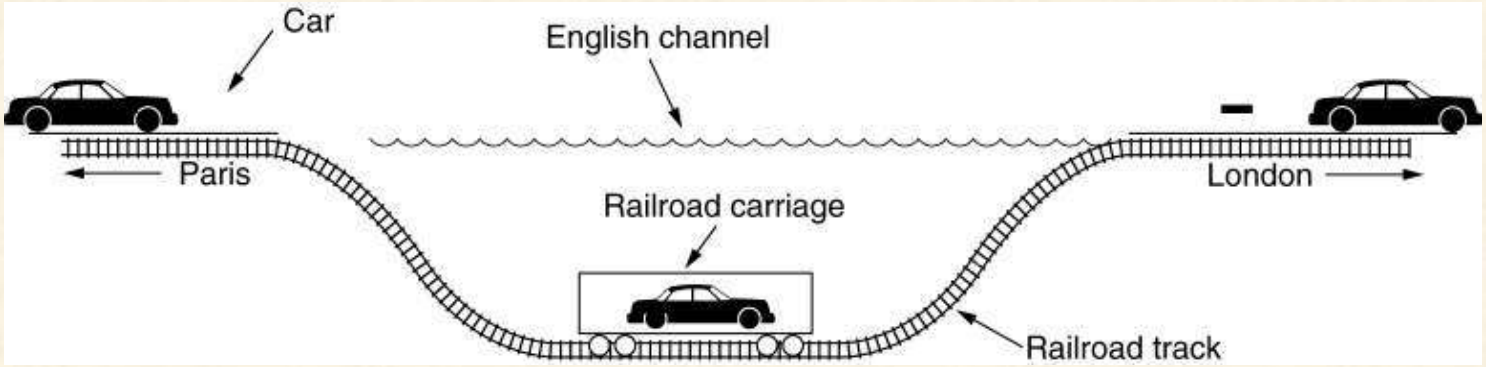
A connectionless internet.

Tunneling



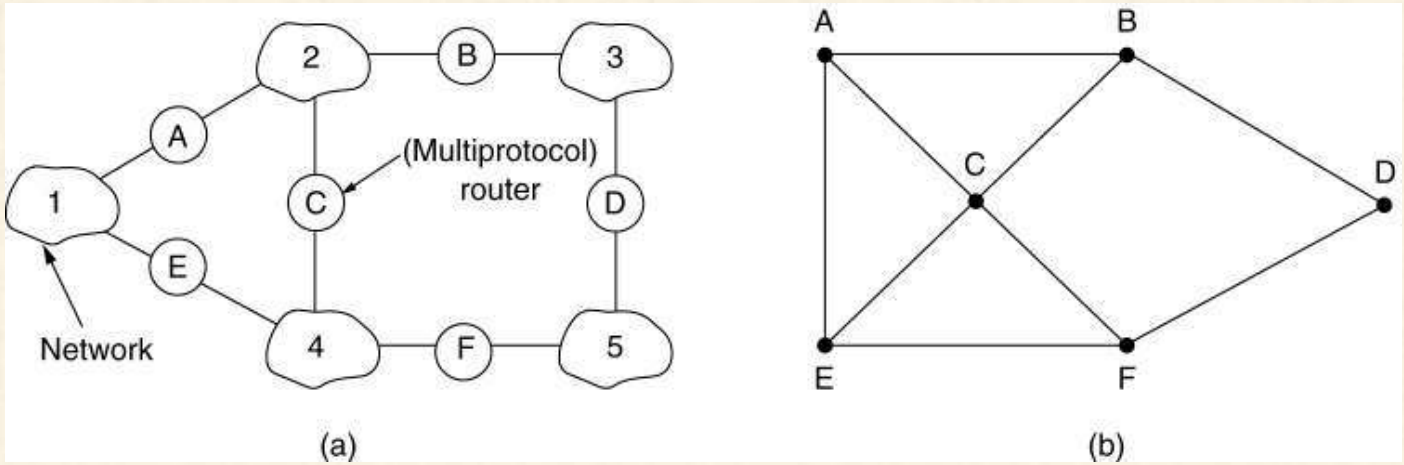
Tunneling a packet from Paris to London.

Tunneling (2)



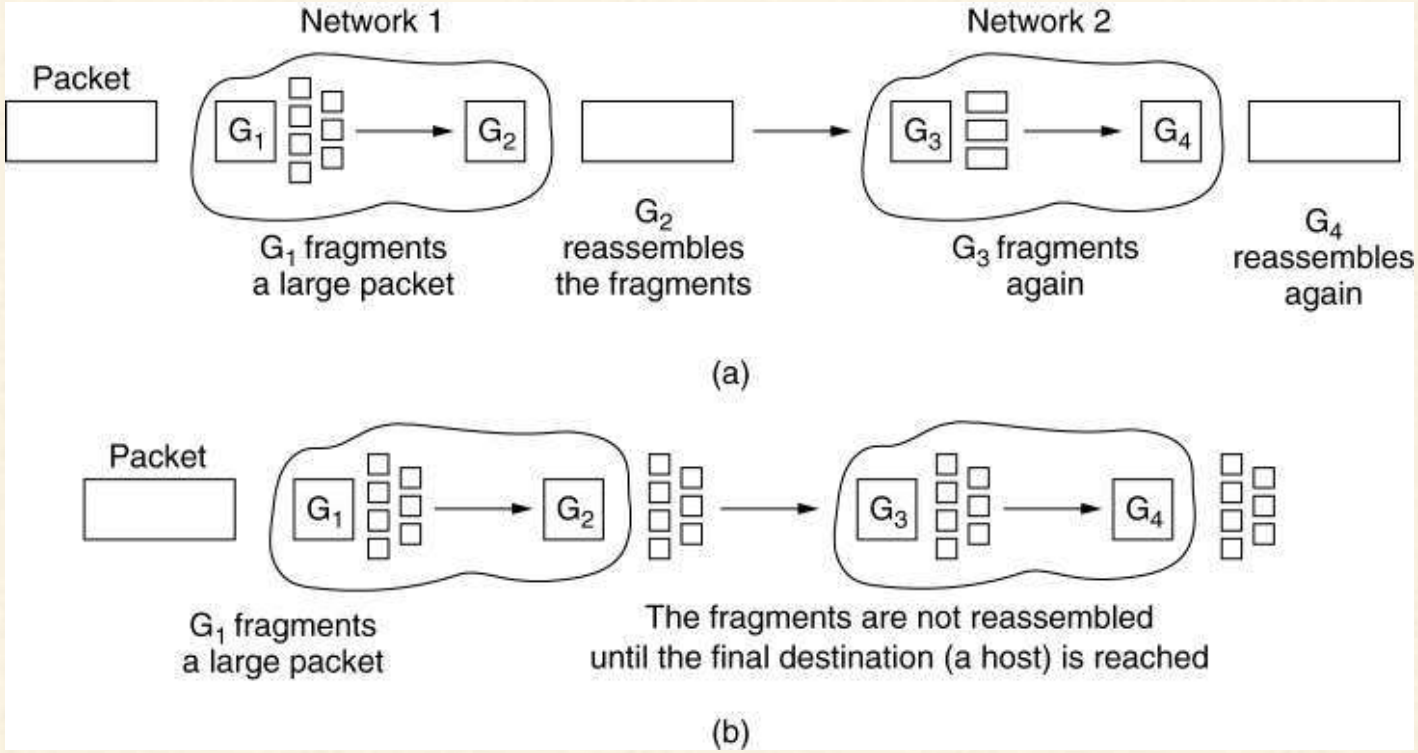
Tunneling a car from France to England.

Internetwork Routing



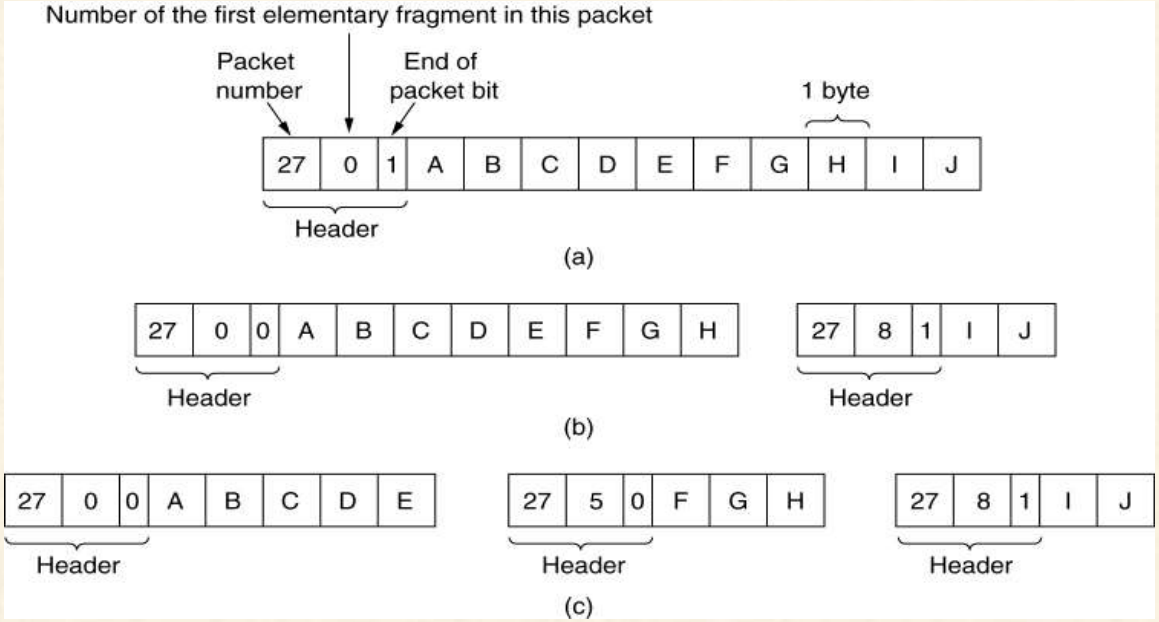
(a) An internetwork. (b) A graph of the internetwork.

Fragmentation



(a) Transparent fragmentation. (b) Nontransparent fragmentation.

Fragmentation (2)



Fragmentation when the elementary data size is 1 byte.

(a)Original packet, containing 10 data bytes.

(b)Fragments after passing through a network with maximum packet size of 8 payload bytes plus header.

(c)Fragments after passing through a size 5 gateway.