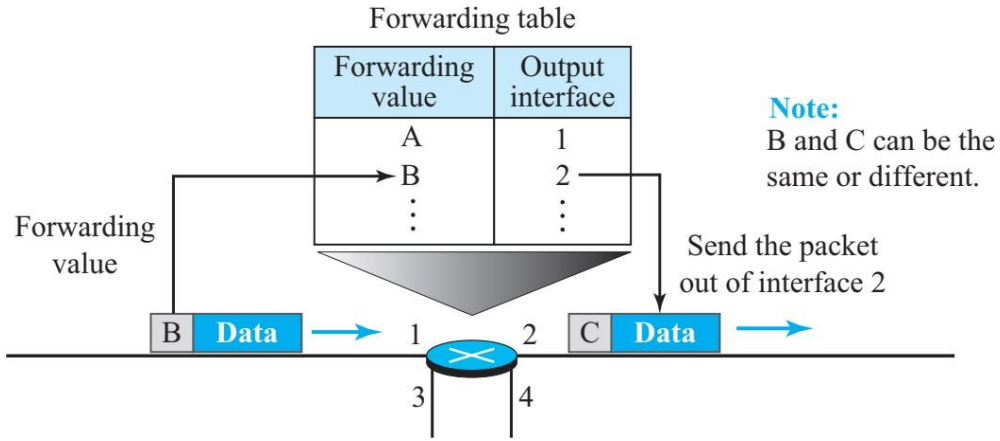
Computer Networks

Unit-4: Network Layer

S.No	Course Outcome	Intended Learning Outcomes (ILO)	Knowledge Level of ILO	No. of Hours
1		Describe The Network Layer Design Issues Store-and-Forward Switching, Services Provided to Transport Layer	K1	2
2		Discuss Implementation of Connectionless Service-Implementation of Connection Oriented Service.	K2	2
3		Construct Shortest Path algorithm	K3	1
4	CO 4	Describe Flooding	K2	1
5		Illustrate Distance Vector Routing and Hierarchical Routing	K3	2
6		Differentiate Broadcast and Multicast Routing	K2	1
7		Discuss Congestion Control Algorithms	K2	1
8		Demonstrate IP Addressing and Subnet Masking	K3	1

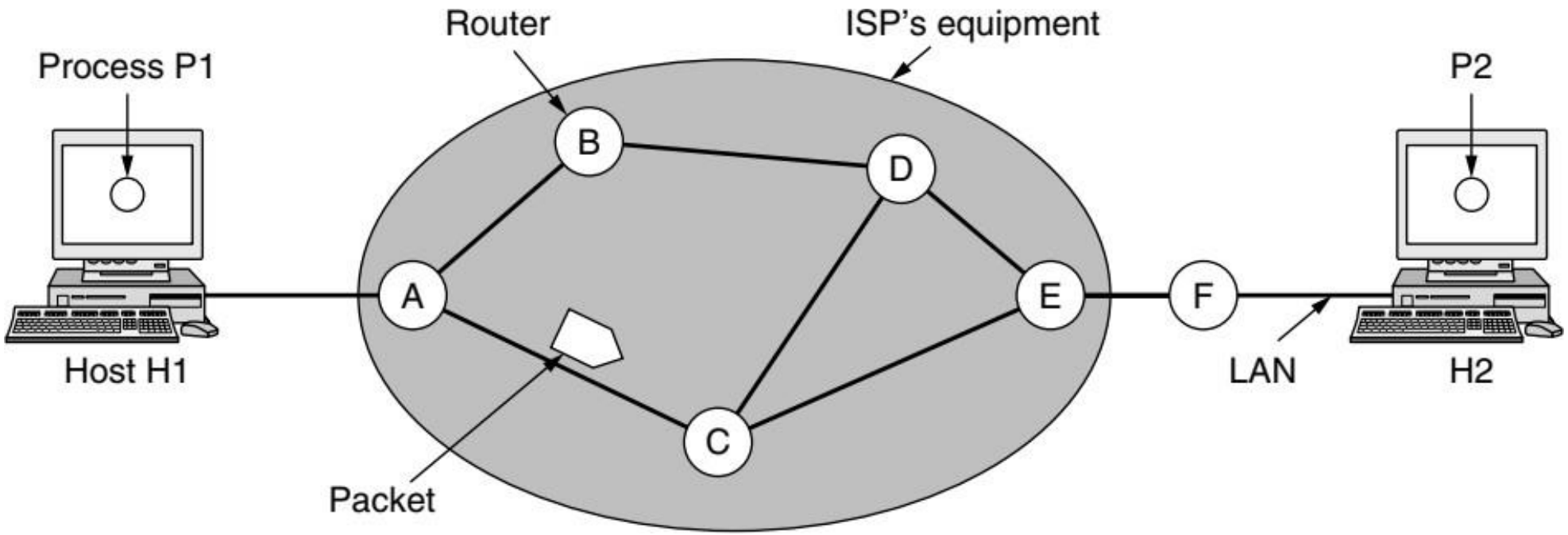
Network Layer Functionalities

- Controls the operation of a subnet.
- Packetizing: Encapsulating the payload (data received from upper layer) in a network-layer IP packet at the source and decapsulating the payload from the network-layer IP packet at the destination.
- Takes care of “Routing” from source to destinations that are in different networks (host- to-host delivery).
- Uses logical (IP) address for identifying and delivering to a node in the network.
- Other Services:
  - No Error Control
  - Directly Not Provides Flow Control
  - Congestion control can be implemented but not done in Internet (TCP/IP)
  - Optional Security in IPv6



Network Layer Design Issues

- The issues that the designers of the network layer must grapple with are:
  - Store-and-Forward Packet Switching
  - Services Provided to Transport Layer
  - Implementation of Connectionless Service
  - Implementation of Connection Oriented Service.



Store-and-Forward Packet Switching

Services Provided to Transport Layer

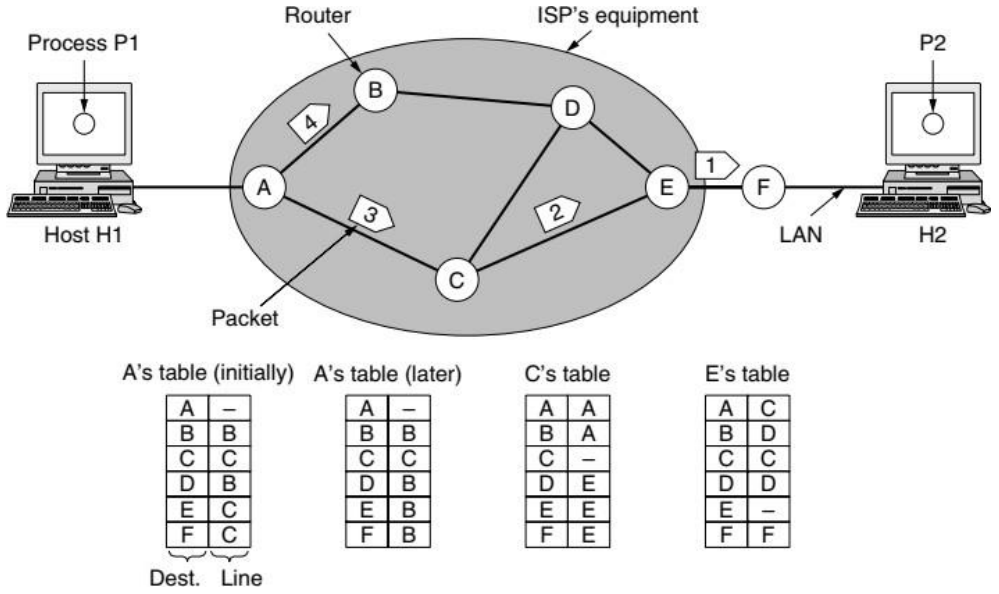
- The network layer provides services to the transport layer at the network layer/transport layer interface.
- So, what kind of services the network layer provides to the transport layer?
- The services need to be carefully designed with the following goals in mind
  - The services should be independent of the router technology.
  - The transport layer should be shielded from the number, type, and topology of the routers present.
  - The network addresses made available to the transport layer should use a uniform numbering plan, even across LANs and WANs.

Implementation of Connectionless & Connection-oriented Services

- Network layer can provide connection less service and also connection-oriented 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 network**.
- 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)**, in analogy with the physical circuits set up by the telephone system, and the network is called a **virtual-circuit network**.

Implementation of Connectionless Service

- Let us now see how a datagram network works. Suppose that the process *P1* in figure has a long message for *P2*. It hands the message to the transport layer, with instructions to deliver it to process *P2* on host *H2*. The transport layer code runs on *H1*, typically within the operating system. It prepends a transport header to the front of the message and hands the result to the network layer, probably just another procedure within the operating system.



Implementation of Connectionless Service

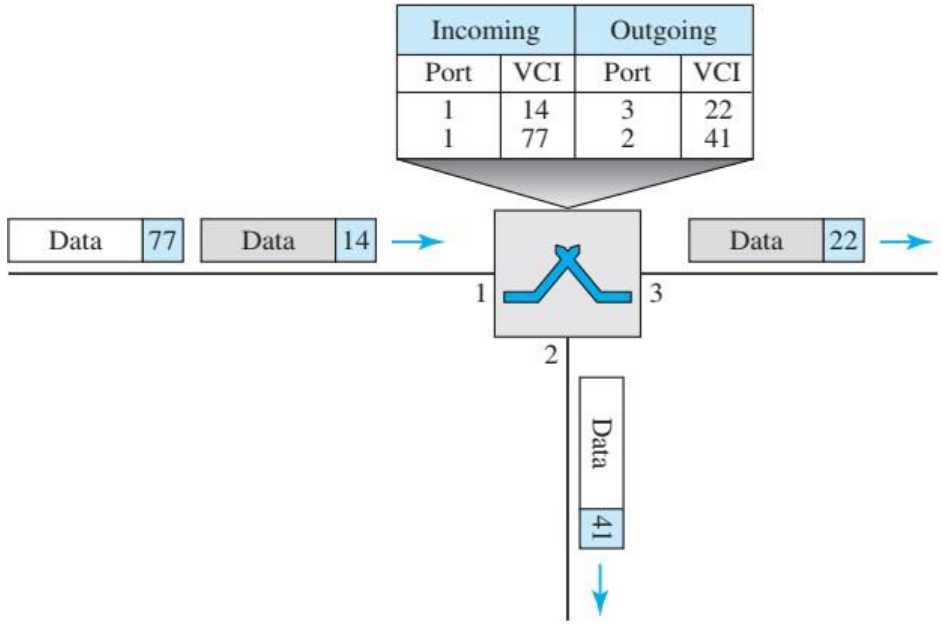
- Let us assume for this example that the message is four times longer than the maximum packet size, so the network layer has to break it into four packets, 1, 2, 3, and 4, and send each of them in turn to router *A*.
- Every router has an internal table telling it where to send packets for each of the possible destinations.
- At A, packets 1, 2, and 3 are stored briefly, having arrived on the incoming link and had their checksums verified. 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. When it gets to F, it is sent within a frame over the LAN to H2. Packets 2 and 3 follow the same route.
- 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, A decided to send packet 4 via a different route than that of the first three packets. Perhaps it has learned of a traffic jam somewhere along the ACE path and 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.

Implementation of Connection-Oriented Service

- For connection-oriented service, we need a virtual-circuit network.
- A Virtual Circuit consist of Three Phases:
  - Setup Phase
  - Data Transfer Phase
  - Teardown Phase

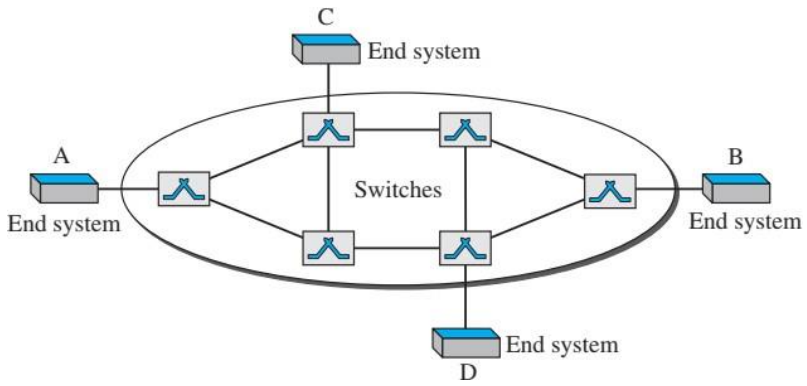
Data-Transfer Phase

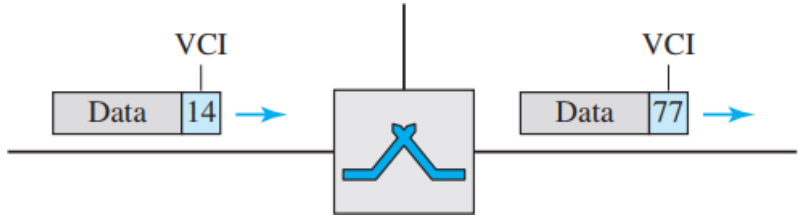
- To transfer a frame from a source to its destination, all switches need to have a table entry for this virtual circuit. The table, in its simplest form, has four columns. This means that the switch holds four pieces of information for each virtual circuit that is already set up.
- The data-transfer phase is active until the source sends all its frames to the destination. The procedure at the switch is the same for each frame of a message. The process creates a virtual circuit, not a real circuit, between the source and destination.



Virtual Circuit Networks

- Addressing
  - Global Addressing: a global address in virtual-circuit networks is used only to create a local address, i.e., a virtual-circuit identifier.
  - Virtual-Circuit Identifier (VCI): used for data transfer, unlike a global address, is a small number that has only switch scope.



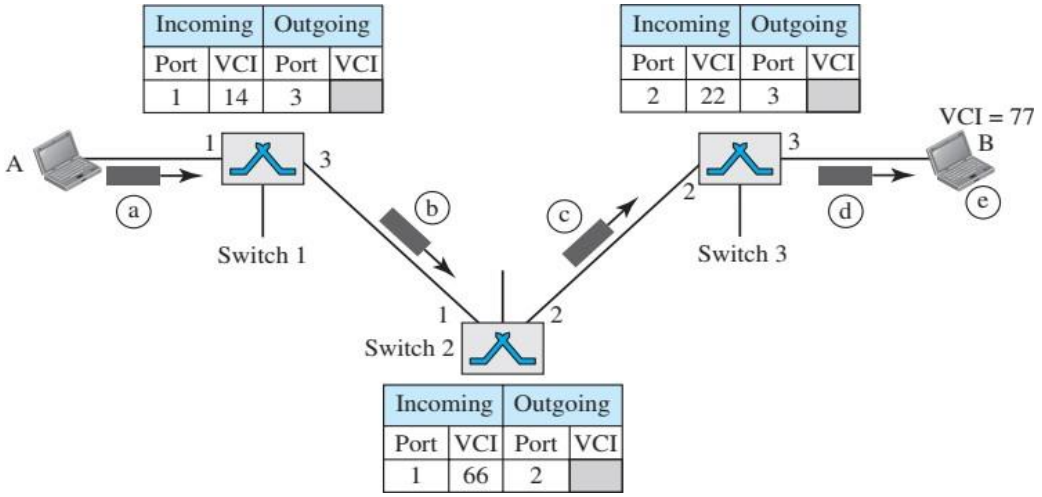


VC Networks – Setup Phase

- In the setup phase, a switch creates an entry for a virtual circuit.
- Two steps are required: the setup request and the acknowledgment.
- A setup request frame is sent from the source to the destination.
- A special frame, called the acknowledgment frame will be sent from destination to source in response.
- To transfer a frame from a source to its destination, all switches need to have a table entry for this virtual circuit. The table, in its simplest form, has four columns. This means that the switch holds four pieces of information for each virtual circuit that is already set up.

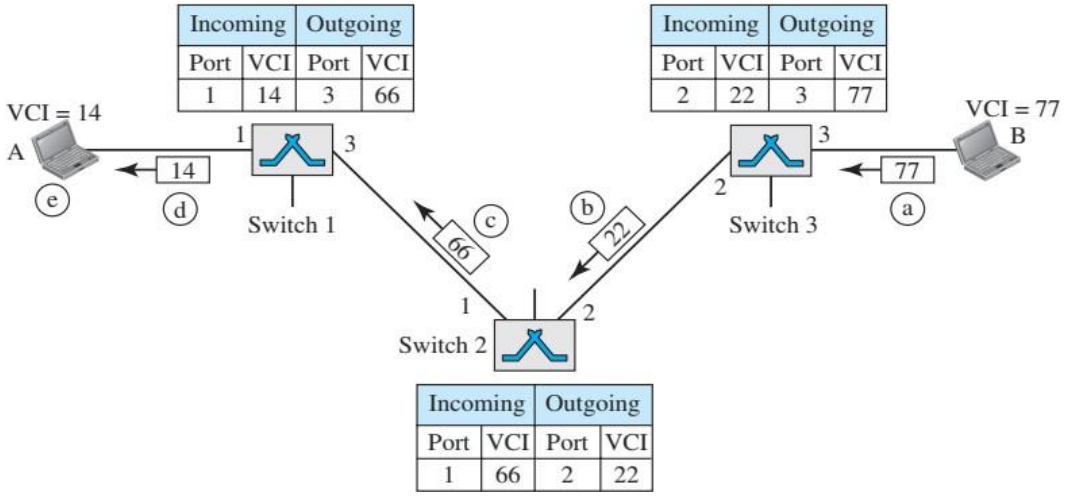
Setup Request

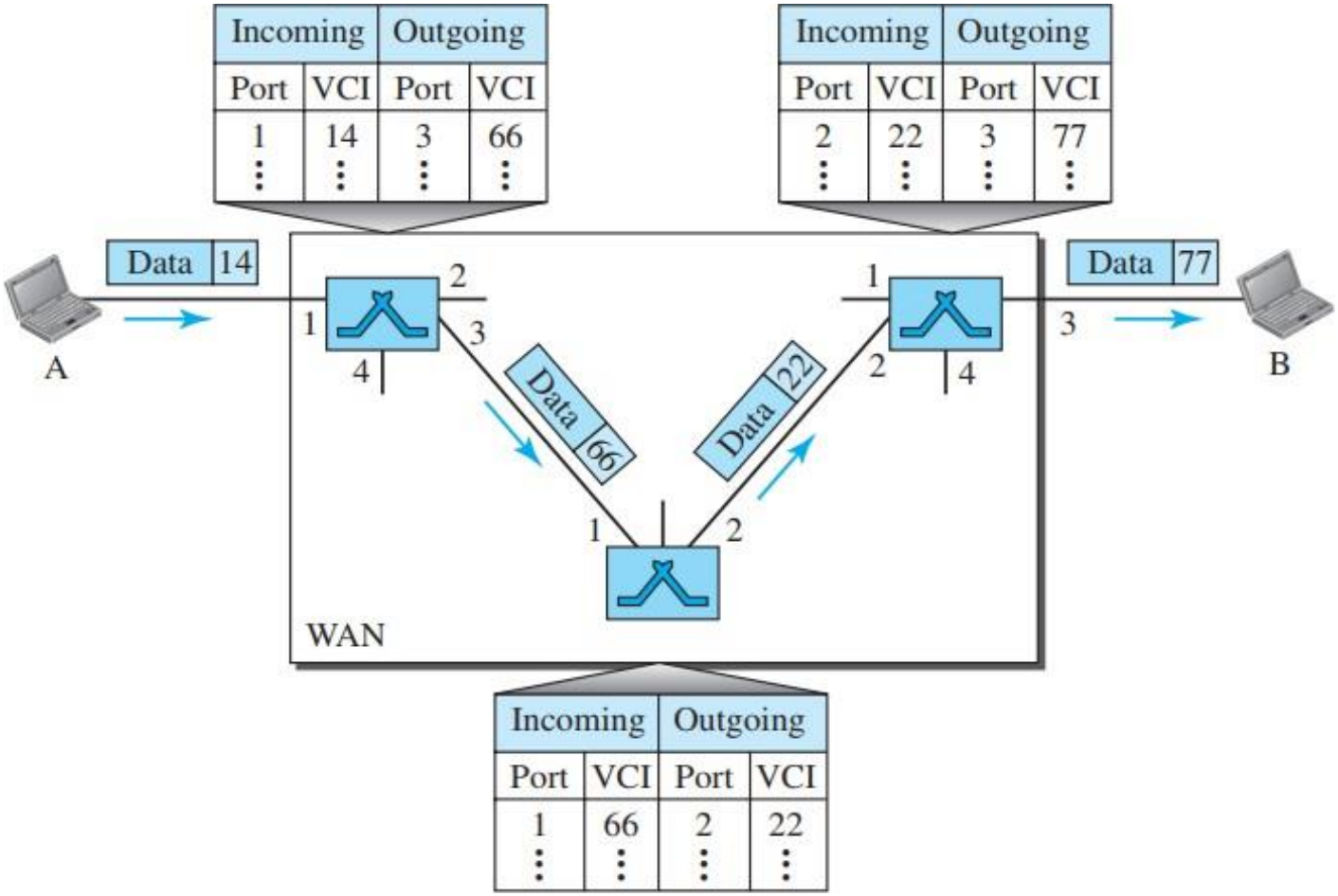
- A setup request frame is sent from the source to the destination.
- Switch 1 receives the setup request frame. It knows that a frame going from A to B goes out through port 3.
- The switch creates an entry in its table for this virtual circuit, but it is only able to fill three of the four columns. The switch assigns the incoming port (1)and chooses an available incoming VCI (14) and the outgoing port (3).
- Switch 2 receives the setup request frame. The same events happen here as at switch 1; three columns of the table are completed: in this case, incoming port (1), incoming VCI (66), and outgoing port (2).
- Switch 3 receives the setup request frame. Again, three columns are completed: incoming port (2), incoming VCI (22), and outgoing port (3).
- Destination B receives the setup frame, and if it is ready to receive frames from A, it assigns a VCI to the incoming frames that come from A, in this case 77. This VCI lets the destination know that the frames come from A, and not other sources.



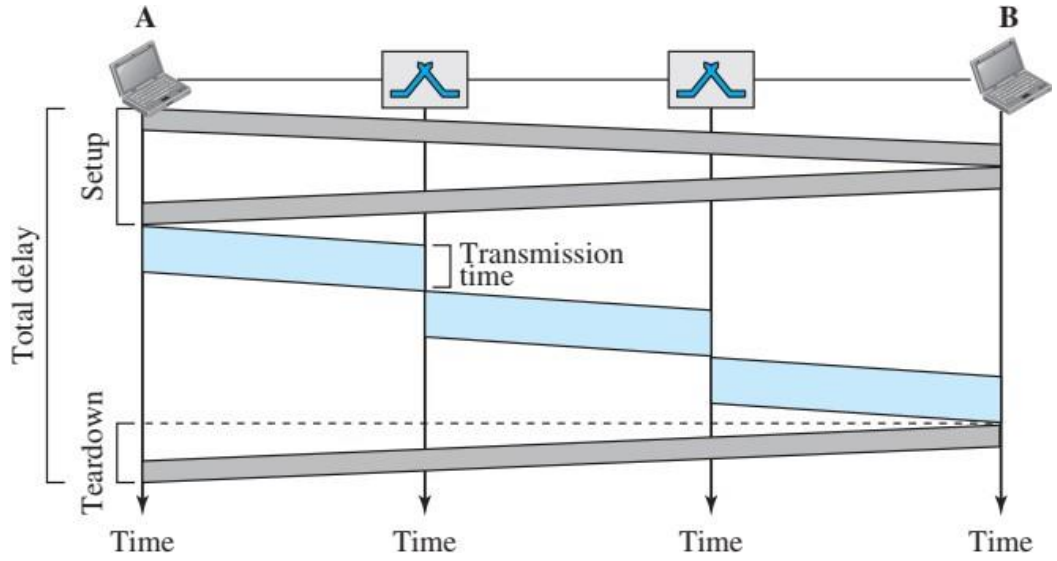
Acknowledgment

- The destination sends an acknowledgment to switch 3. The acknowledgment carries the global source and destination addresses so the switch knows which entry in the table is to be completed. The frame also carries VCI 77, chosen by the destination as the incoming VCI for frames from A. Switch 3 uses this VCI to complete the outgoing VCI column for this entry. Note that 77 is the incoming VCI for destination B, but the outgoing VCI for switch 3.
- Switch 3 sends an acknowledgment to switch 2 that contains its incoming VCI in the table, chosen in the previous step. Switch 2 uses this as the outgoing VCI in the table.
- Switch 2 sends an acknowledgment to switch 1 that contains its incoming VCI in the table, chosen in the previous step. Switch 1 uses this as the outgoing VCI in the table.
- Finally switch 1 sends an acknowledgment to source A that contains its incoming VCI in the table, chosen in the previous step.
- The source uses this as the outgoing VCI for the data frames to be sent to destination B.




Incoming
Outgoing
Port
VCI
Port
VCI
1
14
3
66
Incoming
Outgoing
Port
VCI
Port
VCI
2
22
3
77
Data
14
Data
77
A
B
1
2
3
4
1
2
3
4
Data
66
Data
22
WAN
Incoming
Outgoing
Port
VCI
Port
VCI
1
66
2
22

Teardown Phase



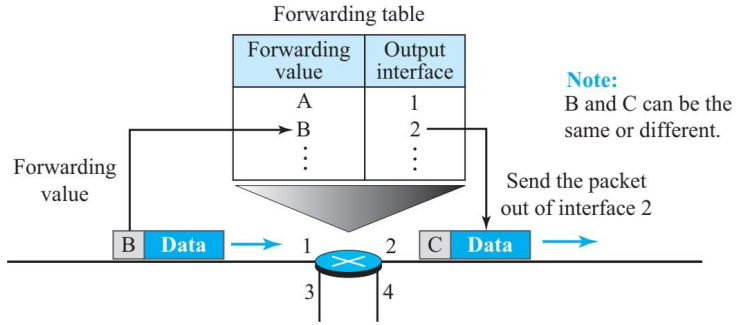
- In this phase, source A, after sending all frames to B, sends a special frame called a teardown request.
- Destination B responds with a teardown confirmation frame.
- All switches delete the corresponding entry from their tables.
- Efficiency: As we said before, resource reservation in a virtual-circuit network can be made during the setup or can be on demand during the data-transfer phase. In the first case, the delay for each packet is the same; in the second case, each packet may encounter different delays.
- Delay: In a virtual-circuit network, there is a one-time delay for setup and a one-time delay for teardown. If resources are allocated during the setup phase, there is no wait time for individual packets.

Datagram vs Virtual Circuit

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

Routing Algorithm

- The routing algorithm is that part of the network layer software responsible for deciding which output line an incoming packet should be transmitted on.
- Routing algorithms can be grouped into two major classes:
  - Nonadaptive
    - Do not base their routing decisions on any measurements or estimates of the current topology and traffic.
    - Instead, the choice of the route to use is computed in advance, offline, and downloaded to the routers when the network is booted.
    - Also called as **"Static Routing"**
  - Adaptive
    - Change their routing decisions to reflect changes in the topology, and sometimes changes in the traffic as well.
    - Also called as **"Dynamic Routing"**

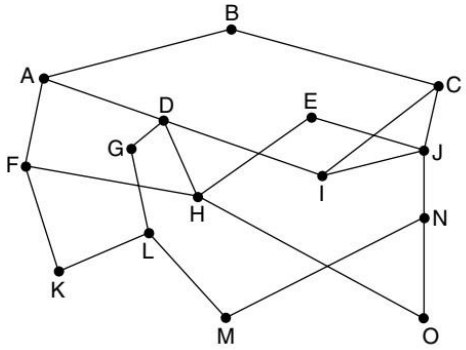


Routing Algorithms

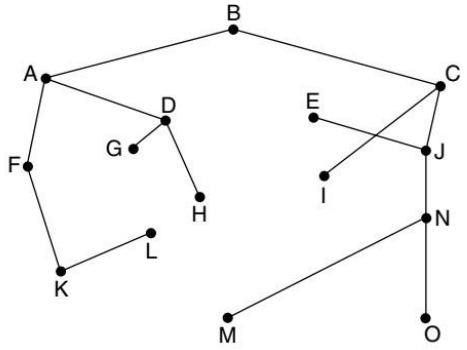
- Shortest Path Algorithm
- Flooding
- Distance Vector Routing
- Link State Routing
- Hierarchical Routing
- Broadcast Routing
- Multicast Routing

Optimality Principle

- Optimality principle 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 route.
- 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**".



A network



A sink tree for router *B*.

Routing Algorithm

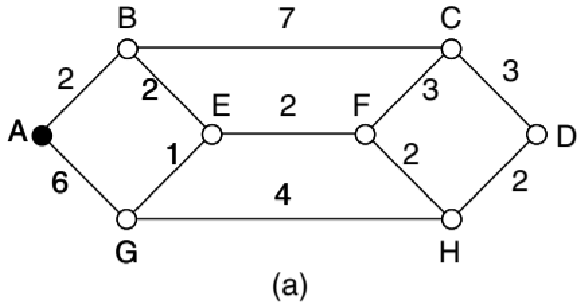
- To find the shortest path between a given pair of routers in the network.
- How to define the term “Shortest Path”?
- Ans. Distance metrics:
  - Hop Count : Number of individual links along the path.
  - Geographic Distance
  - Mean delay : Time
  - a function of the distance, bandwidth, average traffic, communication cost, measured delay, and other factors.

Shortest Path Algorithm

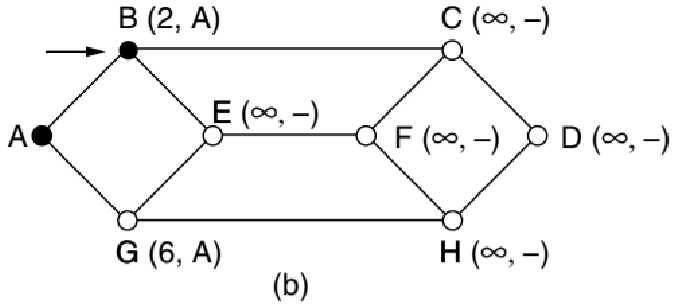
- Proposed by Dijkstra (1959) to find the shortest paths between a source and all destinations in the network.
- Each node is labeled with its distance from the source node along the best known path.
- Initially, no paths are known, so all nodes are labeled with infinity.
- As the algorithm proceeds and paths are found, the labels may change, reflecting better paths.
- Lets see how exactly the algorithm works with an example network.

Shortest Path Algorithm

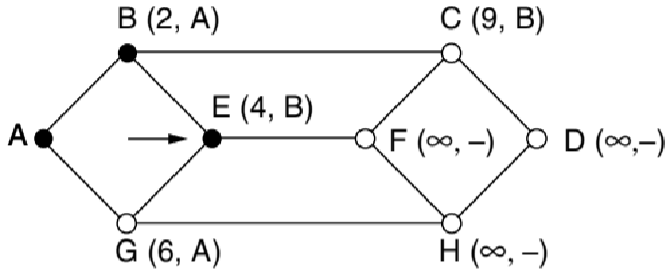
- We want to find the shortest path from *A*.
  1. We start out by marking node A as permanent, indicated by a filled-in circle. This node will be current working node.
  2. Then we examine, in turn, each of the nodes adjacent to the working node except any permanent nodes (If any), relabeling each one with the distance to A.
  3. Once all the adjacent nodes of the working node are examined, choose the next nearest node that is not made permanent and label it as permanent.
  4. Now this node becomes the working node for the next round.
  5. Repeat step 2 to 4 till shortest paths from A to all nodes are found, i.e., all nodes are marked as permanent.



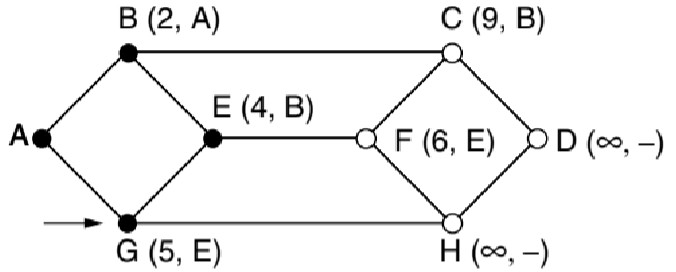
(a)



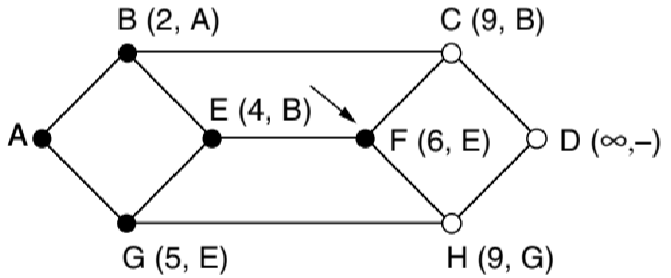
(b)



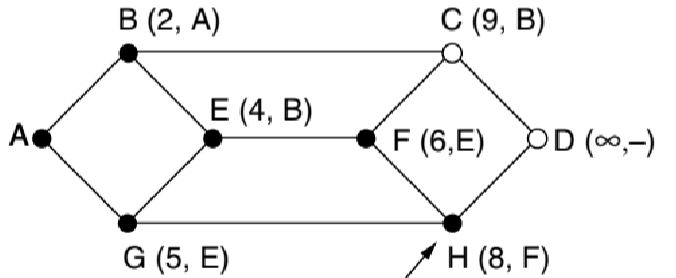
(c)



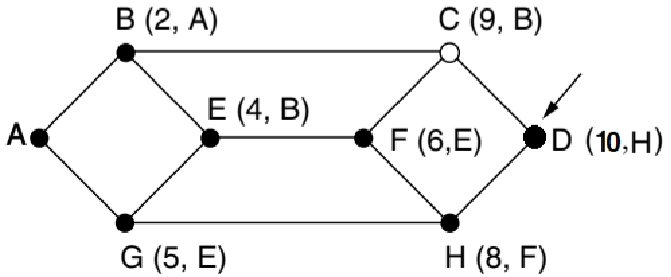
(d)



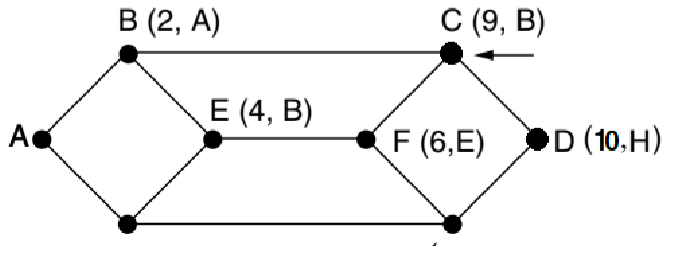
(e)



(f)



(g)



(h)

Flooding

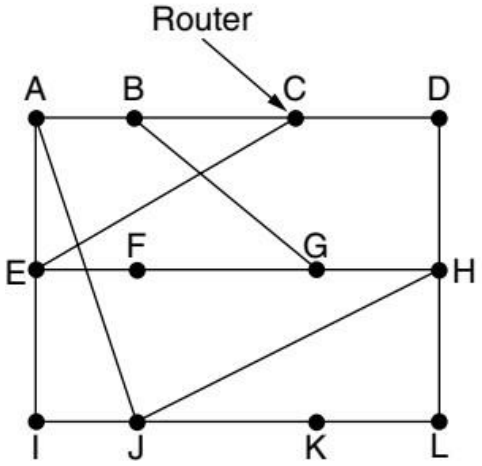
- In **flooding**, every incoming packet is sent out on every outgoing line except the one it arrived on.
- Advantages:
  - Ensures that a packet is delivered to every node in the network.
  - Tremendously robust: Even if large numbers of routers fail, flooding will find its path.
  - Requires little in the way of setup.
- Issues:
  - Duplicate packets
  - Exponentially increased traffic.
- Sol:
  - Hop Counter: a hop counter contained in the header of each packet that is decremented at each hop, with the packet being discarded when the counter reaches zero.
  - A better technique for damming the flood is to have routers keep track of which packets have been flooded, to avoid sending them out a second time. Like each source placing a sequence number in each packet it is generated.

Distance Vector Routing Algorithm

- An adaptive routing algorithm.
- A distance vector routing algorithm operates by having each router maintain a table (i.e., a vector) giving the best-known distance to each destination and which link to use to get there.
- These routing tables are indexed by, and containing one entry for each router in the network.
- This entry has two parts: the preferred outgoing line to use for that destination and an estimate of the distance to that destination.
- These tables are created and updated by exchanging the distance vectors with the neighbors.
- Eventually, every router knows the best link to reach each destination.

Distance Vector Routing Algorithm

- The router is assumed to know the ‘‘distance’’ to each of its neighbors.
- An example for the updating process done by router J of the given network was shown in the figure.
- First J will receive 4 distance vectors from its 4 neighbors, i.e., A, I, H, &K as shown.
- J already knows the distance to its neighbors as to A it is 8, to I, H & K it is 10, 12, & 6 respectively.



A network

New estimated delay from J

To	A	I	H	K	Line
A	0	24	20	21	8 A
B	12	36	31	28	20 A
C	25	18	19	36	28 I
D	40	27	8	24	20 H
E	14	7	30	22	17 I
F	23	20	19	40	30 I
G	18	31	6	31	18 H
H	17	20	0	19	12 H
I	21	0	14	22	10 I
J	9	11	7	10	0 –
K	24	22	22	0	6 K
L	29	33	9	9	15 K

JA delay is 8 | JI delay is 10 | JH delay is 12 | JK delay is 6

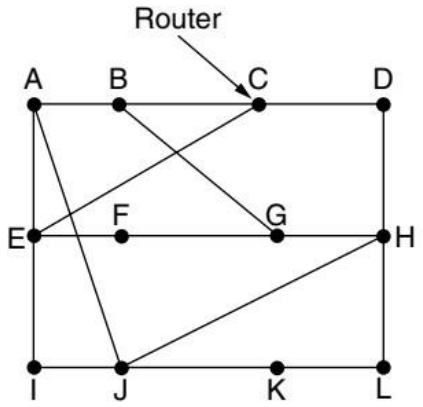
Vectors received from J's four neighbors

New routing table for J

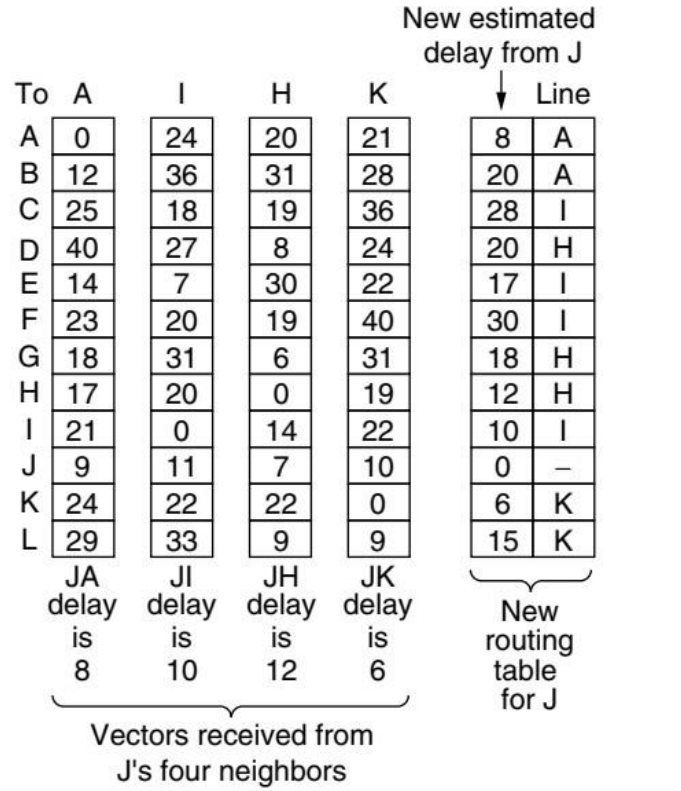
Input from *A*, *I*, *H*, *K*, and the new routing table for *J*

Distance Vector Routing Algorithm

- Now J computes new routes to all the routers through these 4 neighbours and updates its routing table.
- For example, J computes new route to G through A as 8(JA)+18(AG)=26.
- In the same way it computes new routes to G through I, H, & K as 41 (31 + 10), 18 (6 + 12), and 37 (31 + 6), respectively.
- Since the best value, i.e., the least value is 18, J will update its routing table as the shortest path to G is through H with a distance of 18.



A network



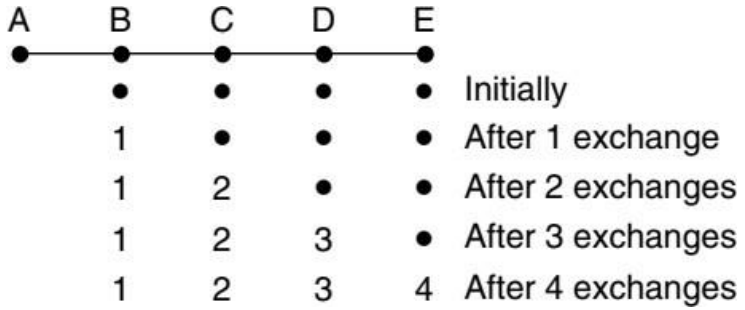
Input from *A*, *I*, *H*, *K*, and the new routing table for *J*.

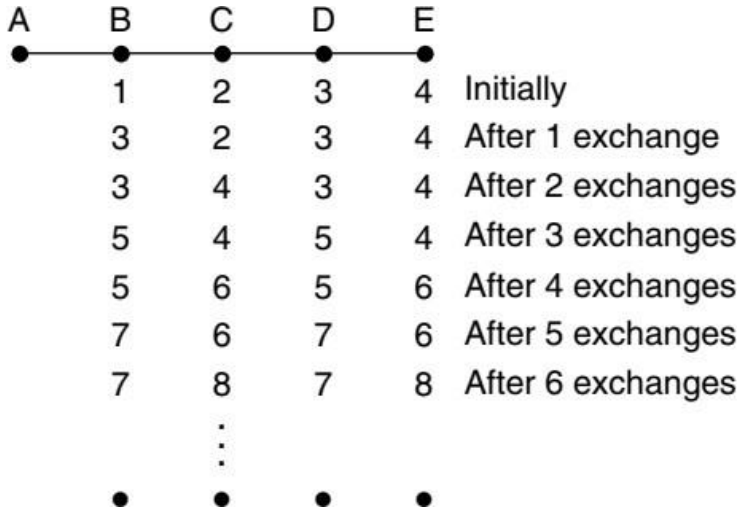
DVR: Count to Infinity Problem

- Distance vector routing converges to the correct answer, it may do so slowly. In particular, it reacts rapidly to good news, but leisurely to bad news.
- For example,
  - Consider the five-node (linear) network, where the delay metric is the number of hops. Suppose if A goes down, then all the other routers must know this. In other words, they have all recorded the delay to A as infinity.
  - Initially when A comes up, the other routers learn about it via the vector exchanges.
  - At the time of the first exchange, B learns that its left-hand neighbor has zero delay to A. B now makes an entry in its routing table indicating that A is one hop away to the left.
  - On the next exchange, C learns that B has a path of length 1 to A, so it updates its routing table to indicate a path of length 2.
  - This process goes on till all nodes know about this goo news.
  - In a network whose longest path is of length N hops, within N exchanges everyone will know about newly revived links and routers.

DVR: Count to Infinity Problem

- Now let us consider the situation, where suddenly, either A goes down or the link between A and B is cut.
- At the first packet exchange, B does not hear anything from A. Fortunately, C says ‘‘Do not worry; I have a path to A of length 2.’’
- For all B knows, C might have ten links all with separate paths to A of length 2.
- As a result, B thinks it can reach A via C, with a path length of 3.
- D and E do not update their entries for A on the first exchange.
- On the second exchange, C notices that each of its neighbors claims to have a path to A of length 3. It picks one of them at random and makes its new distance to A 4.
- It will take infinity number of exchanges to reach infinity distance to A.





Link State Routing

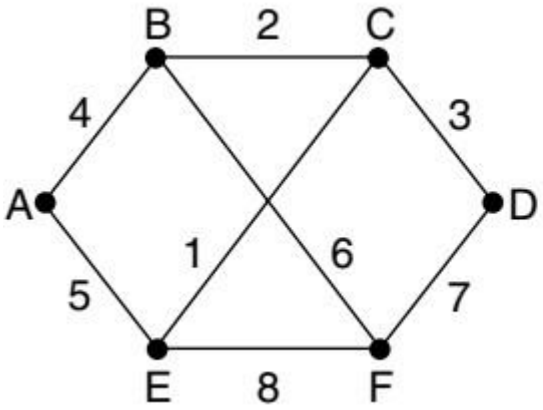
- The idea behind link state routing is fairly simple and can be stated as five parts. Each router must do the following things to make it work:
  1. Discover its neighbors and learn their network addresses.
  2. Set the distance or cost metric to each of its neighbors.
  3. Construct a packet telling all it has just learned.
  4. Send this packet to and receive packets from all other routers.
  5. Compute the shortest path to every other routers

Link State Routing

- Learning about the Neighbours
  - A router upon booting will send a special HELLO packet on each point-to-point line and gets reply with the names of neighbors.
- Setting Link Costs
  - The cost to reach neighbors can be set automatically or configured by the network operator.
  - A common choice is to make the cost inversely proportional to the bandwidth of the link.
  - The delay of the links may be factored into the cost so that paths over shorter links are better choices.

Link State Routing

- Building Link State Packets
  - Once the information needed for the exchange has been collected, the next step is for each router to build a packet containing all the data.
  - The packet starts with the identity of the sender, followed by a sequence number and age and a list of neighbors and the cost to them.
  - The link state packets are either built periodically or when there is a change in the network.



Link State Packets

A	
Seq.	
Age	
B	4
E	5

B	
Seq.	
Age	
A	4
C	2
F	6

C	
Seq.	
Age	
B	2
D	3
E	1

D	
Seq.	
Age	
C	3
F	7

E	
Seq.	
Age	
A	5
C	1
F	8

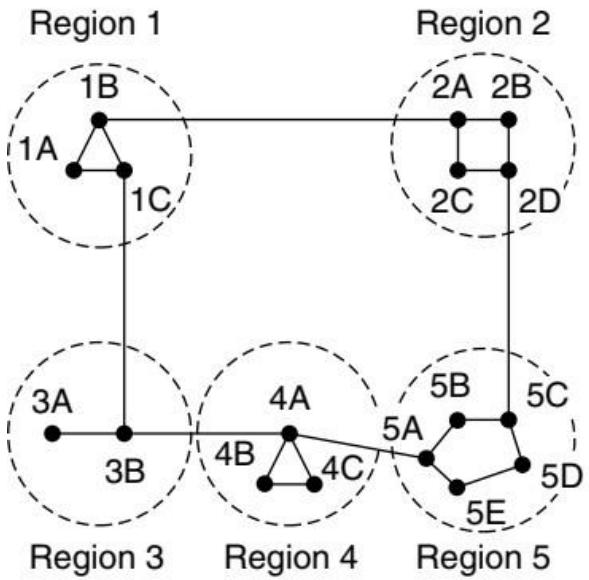
F	
Seq.	
Age	
B	6
D	7
E	8

Link State Routing

- Distributing the Link State Packets
  - The fundamental idea is to use flooding to distribute the link state packets to all routers.
  - To keep the flood in check, each packet contains a sequence number that is incremented for each new packet sent.
  - Also, each packet includes the age of the packet after the sequence number and decrement it once per second. This will help, when the router crashes and restarts the seq. no. from 0.
- Computing the New Routes
  - Once a router has accumulated a full set of link state packets, it can construct the entire network graph because every link is represented.
  - Now Dijkstra's algorithm can be run locally to construct the shortest paths to all possible destinations.

Hierarchical Routing

- As networks grow, 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.
- 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.
- 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.



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

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

Hierarchical Routing

Broadcast Routing

- Sending a packet to all destinations simultaneously is called broadcasting.
- Various methods have been proposed for doing it.
  - Flooding
  - Multi-destination Routing: each packet contains either a list of destinations or a bit map.
  - **Reverse path forwarding**

Reverse path forwarding

- When a broadcast packet arrives at a router, the router checks to see if the packet arrived on the link that is normally used for sending packets toward the source of the broadcast.
- If so, there is an excellent chance that the broadcast packet itself followed the best route from the router and is therefore the first copy to arrive at the router.
- Then, the router forwards copies of it onto all links except the one it arrived on.
- If, the broadcast packet arrived on a link other than the preferred one for reaching the source, the packet is discarded as a likely duplicate.
- The principal advantage of reverse path forwarding is that it is efficient while being easy to implement.
- The last approach is using **“Spanning Tree”.** If each router knows which of its lines belong to the spanning tree, it can copy an incoming broadcast packet onto all the spanning tree lines except the one it arrived on.

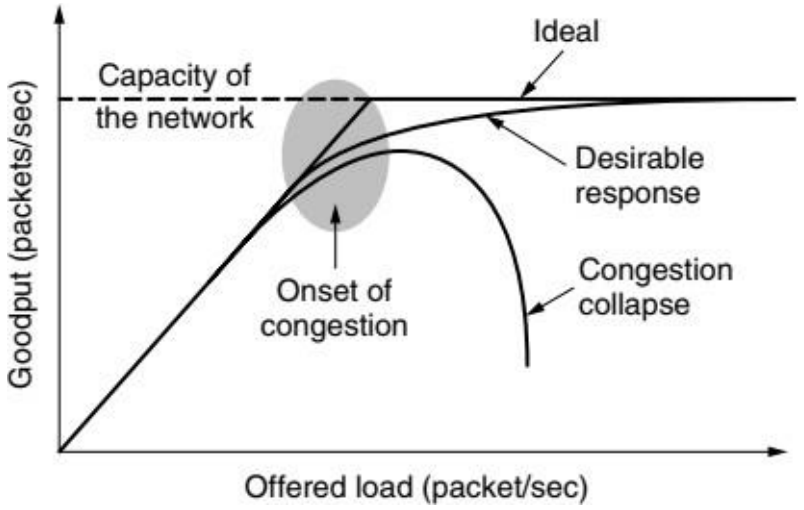
Multicast Routing

- Sending a message to such a group is called **multicasting**, and the routing algorithm used is called **multicast routing**.
- One way to do this is using the multicast spanning tree, i.e., a pruned spanning tree for forwarding the packets.
- Various ways of pruning the spanning tree are possible.
  - Multicast Open Shortest Path Forwarding (MOSPF) uses link state routing algorithm. Since a router in LSP has complete idea about the network, it can built the pruned spanning tree easily.
  - DVMRP (Distance Vector Multicast Routing Protocol) prunes spanning tree recursively.
    - Here, whenever a router with no hosts interested in a particular group and no connections to other routers receives a multicast message for that group, it responds with a PRUNE message, telling the neighbor that sent the message not to send it any more multicasts from the sender for that group.
    - When a router with no group members among its own hosts has received such messages on all the lines to which it sends the multicast, it, too, can respond with a PRUNE message.

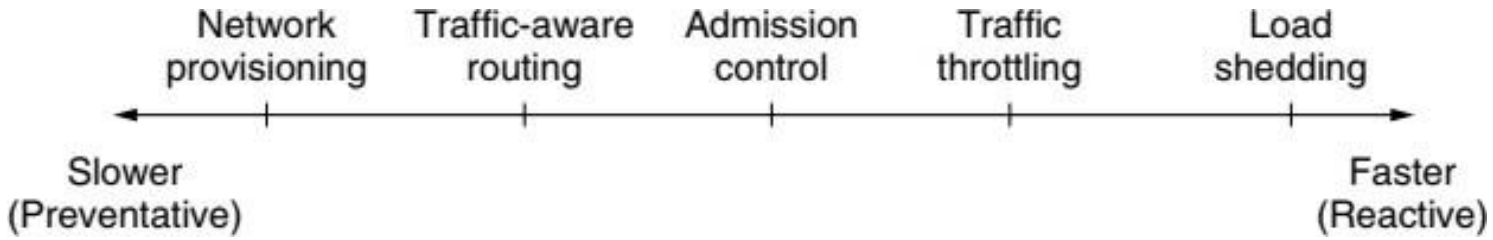
Congestion Control

- 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.
- Since congestion occurs within the network, it is the network layer that directly experiences it and must determine what to do with the excess packets.
- When the number of packets hosts send into the network is well within its carrying capacity, the number delivered is proportional to the number sent.
- However, as the offered load approaches the carrying capacity, bursts of traffic occasionally fill up the buffers inside routers and some packets are lost.
- These lost packets consume some of the capacity, so the number of delivered packets falls below the ideal curve.

Congestion Control



- Unless the network is well designed, it may experience a congestion collapse, in which performance plummets as the offered load increases beyond the capacity.
- The presence of congestion means that the load is (temporarily) greater than the resources (in a part of the network) can handle. Two solutions come to mind: increase the resources or decrease the load.
- These solutions are usually applied on different time scales to either prevent congestion or react to it once it has occurred.

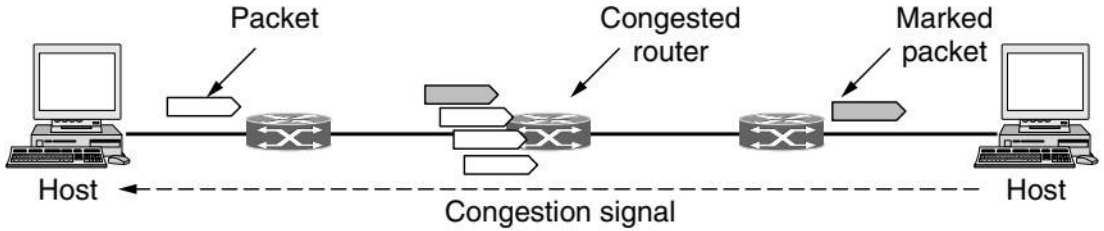


Congestion Control

- Network Provisioning
  - Build a network that is well matched to the traffic that it carries.
- Traffic-aware Routing
  - Routes may be changed to shift traffic away from heavily used paths by changing the shortest path weights. Hence, shifting the away from hotspots.
- Admission Control
  - Stopping new packets released into the network.
  - One technique that is widely used in virtual-circuit networks to keep congestion at bay is admission control.
  - Do not set up a new virtual circuit unless the network can carry the added traffic without becoming congested.

Congestion Control

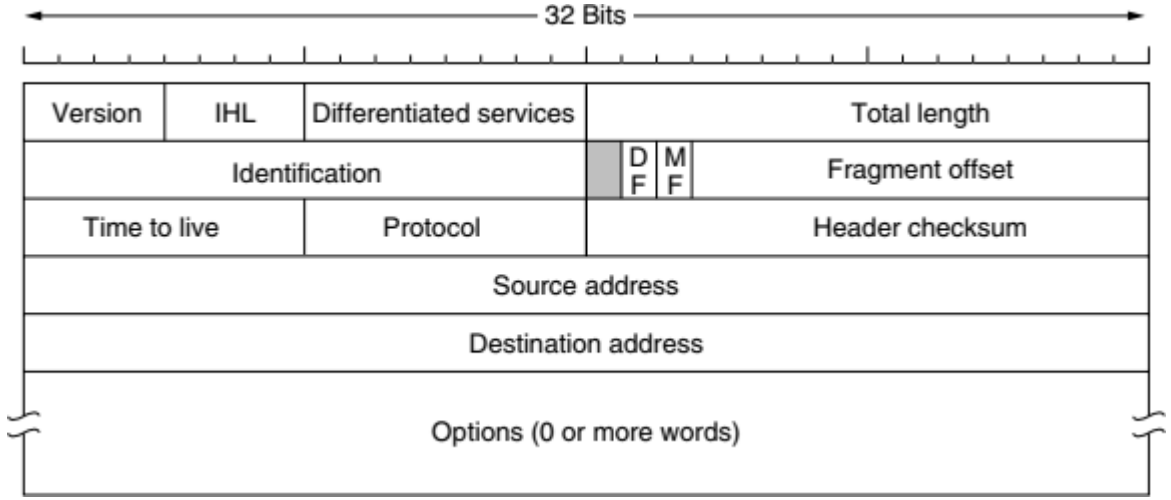
- Traffic Throttling
  - Sending a feedback to sender that are causing the congestion to slow down.
  - Choke packets: The router that felt congested sends a choke packet back to the source host.
  - Explicit Congestion Notification (ECN): A router can tag any packet it forwards to signal that it is experiencing congestion. So that the destination can note that there is congestion and inform the sender when it sends a reply packet.



- Load Shedding
  - Load shedding is a fancy way of saying that when routers are being inundated by packets that they cannot handle, they just throw them away.
  - The key question for a router drowning in packets is which packets to drop.
  - Ans is **wine policy (File Transfer)** where new packets are discarded and **milk policy (Real-time Media)** where old packets are discarded.

IP Protocol

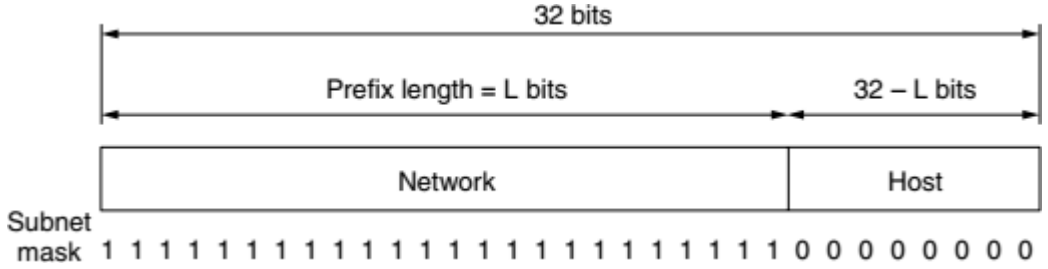
- An IPv4 datagram consists of a header part and a body or payload part. The header has a 20-byte fixed part and a variable-length optional part.



- The Source address and Destination address indicate the IP address of the source and destination network interfaces.

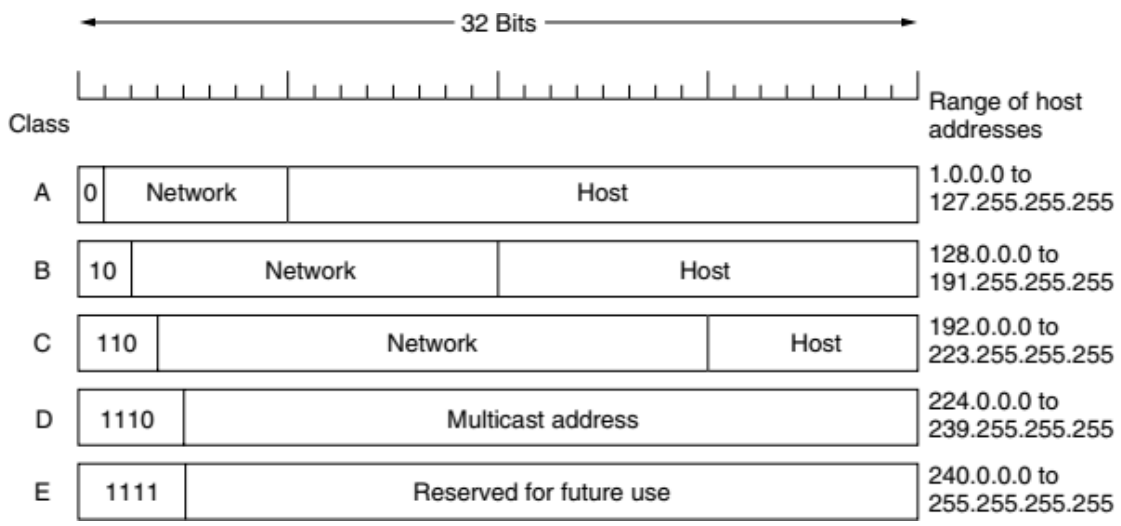
IP Address

- A defining feature of IPv4 is its 32-bit addresses. Every host and router on the Internet has an IP address.
- It is important to note that an IP address does not actually refer to a host. It really refers to a network interface, so if a host is on two networks, it must have two IP addresses.
- IP addresses are hierarchical, unlike Ethernet addresses. Each 32-bit address is comprised of a variable-length network portion in the top bits and a host portion in the bottom bits.
- The network portion has the same value for all hosts on a single network, such as an Ethernet LAN. This is called **prefix.**



IP Address

- Since the prefix length cannot be inferred from the IP address alone, routing protocols must carry the prefixes to routers.
- Sometimes prefixes are simply described by their length, as in a "/16" which is pronounced "slash 16." The same written in dot decimal format is called as subnet mask, i.e., 255.255.0.0.
- In 1993, the IP addresses are divided into 5 classes.



IP Addresses

- The class A allow for up to 128 networks with 16 million hosts each.
- The class B allows for up to 16,384 networks with up to 65,536 hosts each.
- The class C allows for up to 2 million networks with up to 256 hosts each.
- The class D is used for multicasting and class E for experimentation.
- Some of class A, B, & C are used as private addresses, where they can be used in private networks but not in the internet.

Private IP address space	
From	**To**
10.0.0.0	10.255.255.255
172.16.0.0	172.31.255.255
192.168.0.0	192.168.255.255

IP Address : Subnet Masking

- The subnet mask is used by the router to cover up the network address. It shows which bits are used to identify the subnet.
- For example, the subnet mask for class B is 255.255.0.0, i.e., /16 that means the first 16 bits represent the network bits.
- Every network has its own unique address, that can be identified with the help of subnet mask, Like here, class B network has network address 172.20.0.0, which has all zeroes in the host portion of the address.
- To get the network address given the an IP address in the subnet and its subnet mask, we need to perform AND operation on binary representation of them to get network address.
- A bitwise OR between the network address and the inverted subnet mask would give us the broadcast address.

IP Address : Subnet Masking Example

- Given the IP address 172.45.124.89 and subnet mask 255.255.0.0.
- By converting them into binary format we will get IP address as 10101100.00101101.11111100.01011001 and subnet mask as 11111111.11111111.00000000.00000000.
- Upon performing AND operation we will get 10101100.00101101.00000000.00000000, which means 172.45.0.0 is the network address.
- To get the broadcast address, first find the inverted subnet mask, i.e., 00000000.00000000.11111111.11111111.
- Now by performing OR between given IP address and inverted subnet mask, we will get 10101100.00101101.11111111.11111111, i.e., 172.45.255.255.

IP Address: Subnet Masking

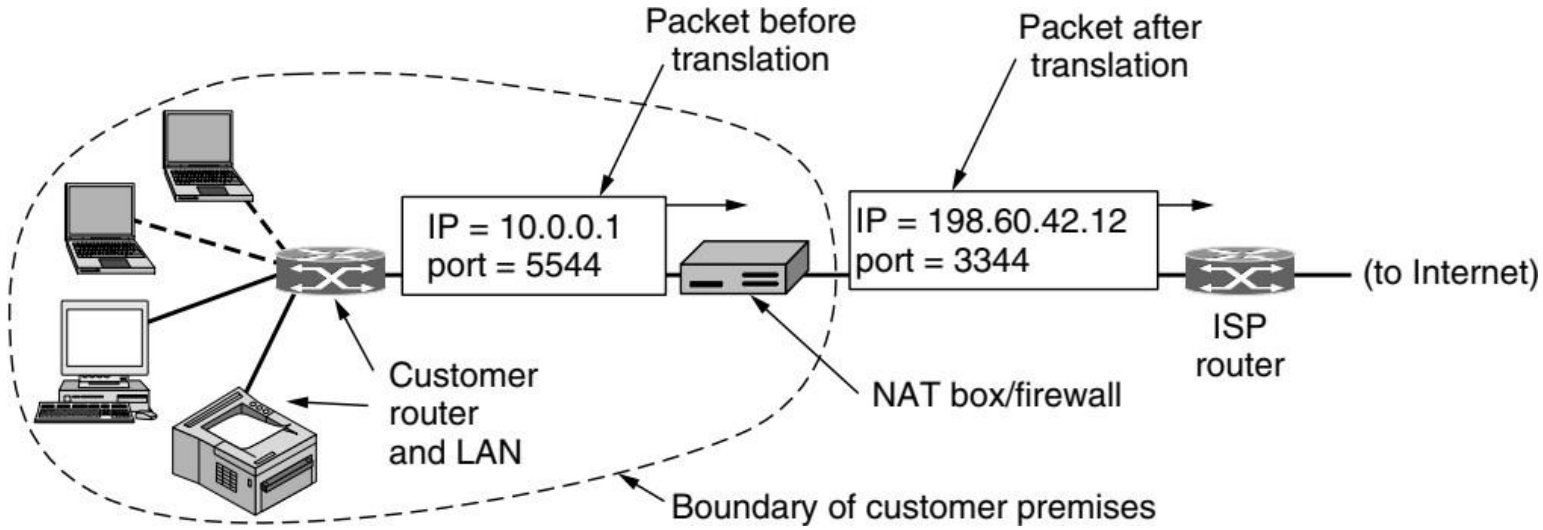
- In other way, one can find out the network address given a IP address in the network by replacing all the host bits with 0's, and broadcast address by replacing all host bits with 1's.
- For example, for the problem discussed in previous slide, i.e., IP address 172.45.124.89 and subnet mask 255.255.0.0, one can find that there are 16 network bits and 16 host bits.
- Hence by taking the binary representation of 172.45.124.89, i.e., 10101100.00101101.11111100.01011001 and replacing the 16 host bits, i.e., the 16 least significant bits with 0's as 10101100.00101101.00000000. 00000000 we can get the network address as 172.45.0.0.
- In the same way we will get the broadcast address of the network by replacing the 16 least significant bits with 1's as 10101100.00101101.11111111.11111111 as 172.45.255.255.

Private Addresses and NAT

- IP addresses are scarce. Only 2^{32} IP addresses available for use, where the number of devices in the internet is more than it.
- A long term solution is IPv6 of size 128, but the transition from IPv4 to v6 is slowly occurring.
- Another solution is using the Network Address Translation with private addresses.
- In NAT, the ISP to assign each home or business network with a single public IP address (or at most, a small number of them) for Internet traffic.
- But, within the customer's private network every computer is assigned with a unique private IP address.
- The unique private IP address is used for internal routing.
- However, when a packet exits the customer network and goes to the ISP, an address translation from the unique internal IP address to the shared public IP address takes place. This translation is called NAT.

NAT

- The operation of NAT is shown in Fig.
- Within the customer premises, every machine has a unique address of the form 10.x.y.z.
- However, before a packet leaves the customer premises, it passes through a NAT box that converts the internal IP source address, 10.0.0.1 port number, 55044 in the figure, to the customer's true IP address, 198.60.42.12 and an unused random port number 53344 in this example.



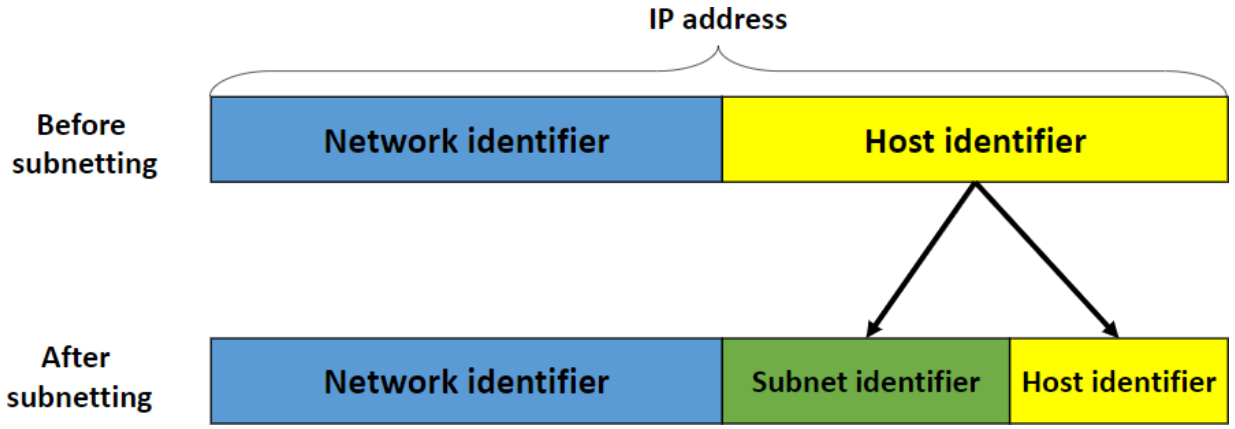
NAT

- NAT box also stores the same information in NAT table.
- when the reply comes back (e.g., from a Web server), it is naturally addressed to 198.60.42.12: 53344.
- By looking at the NAT table the IP address and port number will be re-translated back to 10.0.0.1 & 55044.

Internal Addresses		Public Addresses	
IP Address	Port	IP Address	Port
10.0.0.1	55044	198.60.42.12	53344
10.0.1.54	51024	198.60.42.12	53345
10.0.0.1	56578	198.60.42.12	53348
. . .	. . .	. . .	. . .

Subnetting

- Subnetting is the practice of dividing a bigger network into 2^nsmaller networks.
- It helps you to maximize IP addressing efficiency (Conservation of IP addresses).
- It increases routing efficiency, which helps to enhance the security of the network and reduces the size of the broadcast domain.
- IP Subnetting designates high-order bits from the host as part of the network prefix. This method divides a network into smaller subnets.



Subnetting: Process

1. First find out into how many subnets the given network has to be divided.
2. Approximate the number of subnets to next nearest 2^n value, i.e., for example if a network has to be divided into at least 6 subnets, then it must be divided into 8 subnets.
3. Now find out the number of network bits and number of host bits in the given network using subnet mask.
4. Then mark the “n” number of host bits in the most significant side as subnet bits, and remaining as host bits of each subnet.
5. Now, to get the i_{th}subnet take n bit binary representation of “i-1” and place it in the place of subnet bits.
6. By placing the remaining host bits as 0’s we will get i_{th}sub-network address and by placing 1’s we will get i_{th}sub-network’s broadcast address

Subnetting: Example

- Q. Divide the given network with N/W address as 172.16.4.0 with subnet mask 255.255.253.0 into at least 6 subnets with at least 30 useable IP addresses in each subnet. Find out the 7th subnet range.
- Sol. Given N/W address is 172.16.4.0 ➏ 10101100.00010000.00000100.00000000
- Given subnet mask 255.255.253.0 ➏ 11111111.11111111.11111100.00000000 ➏ /22
- Hence 10101100.00010000.00000100.00000000 (Refer slide-54)
- So in the given network number of N/W bits are 22 and host bits are 32-22=10, i.e., there are 2^{10}=1024 host IP addresses. Ranging from 172.16.4.0 to 172.16.7.255.
- It was asked to divide the given network into 6 subnets with each having 30 usable IP addresses, i.e., including subnet address and broadcast address 32 IP addresses.
- Hence, the given network can be divided into
  1. 8 subnets with 128 IP addresses each (8*128=1024) ➏ 3 subnet bits and 7 host bits
  2. 16 subnets with 64 IP addresses each (16*64=1024) ➏ 4 subnet bits and 6 host bits
  3. 32 subnets with 32 IP addresses each (32*32=1024) ➏ 4 subnet bits and 4 host bits

Subnetting: Example

- So, if we choose option 1, 3 of the host bits in the given network becomes subnet bits.
- 10101100.00010000.00000100.00000000 (Refer slide-54)
- To, get the 7th subnet network address, 7- 1=6 ❼ 110 must be substituted in the subnet bit position, and 0's in host bits position. i.e., 10101100.00010000.00000110.00000000 ❼ 172.16.7.0 is the network address of the 7th subnet.
- In the same way 110 must be substituted in the subnet bit position, and 1's in host bits position. i.e., 10101100.00010000.00000111.01111111 ❼ 172.16.7.127 is the broadcast address of the 7th subnet.

Subnet	Subnet Bits	N/W Address	Broadcast Address
1	1-1=0 ❼ 000	172.16.4.0	172.16.4.127
2	2-1=1 ❼ 001	172.16.4.128	172.16.4.255
3	3-1=2 ❼ 010	172.16.5.0	172.16.5.127
4	011	172.16.5.128	172.16.5.255
5	100	172.16.6.0	172.16.6.127
6	101	172.16.6.128	172.16.6.255
7	110	172.16.7.0	172.16.7.127
8	111	172.16.7.128	172.16.7.255

Supernetting

- Supernetting is the process of summarizing a bunch of contiguous Subnetted networks back in a single large network.
- Supernetting is also known as route summarization and route aggregation.
- There are some points which should be kept in mind while supernetting:
  1. All the Networks should be contiguous and the number of subnets must be 2^m.
  2. The block size of every networks should be equal and must be in form of 2^n.
  3. First Network id should be exactly divisible by whole size of supernet.
- Process:
  - Find out whether the given conditions are satisfied are not, and find out the $m = log_2^{number\_of\_sunets}$.
  - If so, find out the subnet mask, i.e., number of subnet bits “n” in each network (which must be same), and number of host bits “32-n”.
  - Now to get the resultant new subnet mask of the combined supernet by making m number of least significant bits of the network bits into 0’s.

Supernetting : Example

- Combine the subnets 200.1.0.0/24, 200.1.1.0/24, 200.1.2.0/24, 200.1.3.0/24 and form a single supernet.
- Sol. 200.1.0.0 ➡ 11001000.00000001.00000000.00000000
- 200.1.1.0 ➡ 11001000.00000001.00000001.00000000
- 200.1.2.0 ➡ 11001000.00000001.00000010.00000000
- 200.1.3.0 ➡ 11001000.00000001.00000011.00000000
- The given subnets are continuous and number of subnets to be joined is $2^m = 2^2 = 4$. (Cond. 1 Satisfied)
- Since the subnet mask of all subnets is /24 ➡ 255.255.255.0, number of host bits are 8, hence number of hosts are 256 in each of the 4 subnets. (Cond. 2 Satisfied).

Supernetting : Example

- Now, to join the 4 subnets together, the new subnet mask must be created by moving m = 2 network bits into host bits making the resultant subnet mask as /22.
- Hence, the first 22 bits of the all 4 subnet's must be same. (Cond. 3 Satisfied)
- Hence the new subnet mask is /22 ➏ 255.255.253.0
- The supernet N/W address is the N/W address of the first subnet, i.e., 200.1.0.0.
- By following the process discussed in slides 48-49, we can find out the supernet broadcast address as 200.1.3.255.