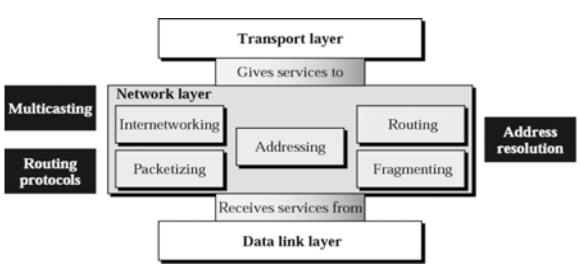
Unit - III
Network Layer

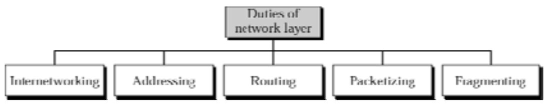
Learning Objectives

- Store-and-Forward Packet Switching
- Services Provided to the Transport Layer
- Implementation of Connectionless Service
- Implementation of Connection-Oriented Service
- Comparison of Virtual-Circuit and Datagram Subnets
- Routing algorithms
- Congestion Control and quality of service

Position of network layer



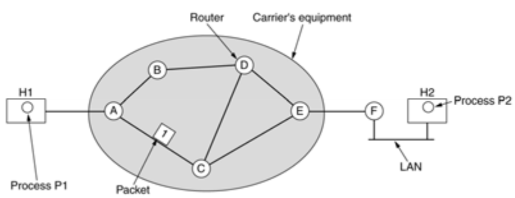
Position of network layer



 U3.4

Store-and-Forward Packet Switching

The environment of the network layer protocols.



 U3.5

Store-and-Forward Packet Switching

The network layer services have been designed with the following goals :

- 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.

 U3.6

Store-and-Forward Packet Switching

Two classes of service provided by the Network Layer.

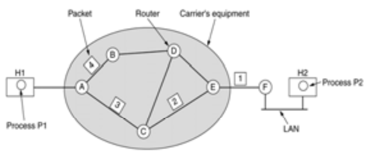
- Connectionless
- Connection-oriented

Routing

If connectionless service is offered, packets are injected into the subnet individually and routed independently of each other. No advance setup is needed. In this context, the packets are frequently called datagrams and the subnet is called datagram subnet.

Routing

Routing within a diagram subnet.



A's table initially (Dest)	(Line)	A's table later (Dest)	(Line)	C's table (Dest)	(Line)	E's table (Dest)	(Line)
A	–	A	–	A	A	A	C
B	B	B	B	B	A	B	D
C	C	C	C	C	–	C	C
D	B	D	B	D	D	D	D
E	C	E	B	E	E	E	–
F	C	F	B	F	E	F	F

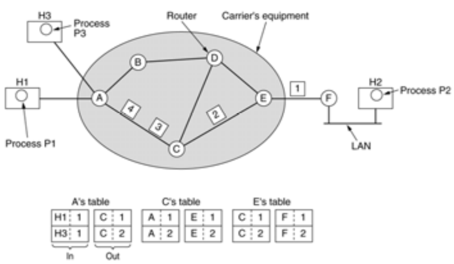
Dest. Line

Routing

If connection-oriented is used, a path from the source to the destination router must be established before any data packets can be sent. This connection is called a VC (virtual circuit).

Routing

Routing within a virtual-circuit subnet.



Routing

Distinguish between virtual circuit and datagram type of routing?

In case of virtual circuit, a session is established between source and destination. At the beginning of the session, route is decided for all the packets to be sent for that session. In datagram type of routing, each packet is independently routed.

Comparison of Virtual-Circuit and Datagram Subnet

Issue	Datagram subnet	Virtual-circuit subnet
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 Algorithms

- The Optimality Principle
- Shortest Path Routing
- Flooding
- Distance Vector Routing
- Link State Routing
- Hierarchical Routing
- Broadcast Routing
- Multicast Routing
- Routing for Mobile Hosts
- Routing in Ad Hoc Networks

Routing Algorithms

- The main function of the network layer is routing packets from the source machine to the destination machine. The algorithms that choose the routes and the data structures that they use are a major area of the network layer design.
- The routing algorithm is that part of the network layer software responsible for deciding which output line an incoming packet should be terminated on.
- If the subnet uses datagrams internally, this decision must be made a new for arriving packets since the best route may have changed since last time.

Routing Algorithms

- If the subnet uses virtual circuits internally, routing decision are made only when a new virtual circuit is being set up.
- Therefore, data packets just follow the previously established route.
- This case is sometimes called session routing because a route remains in force for an entire user session.

Routing Algorithms

- Routing algorithms can be grouped into two major classes : Non adaptive and adaptive.
- Non adaptive algorithms do not base their routing decisions on measurements or estimates of the current traffic and topology. This procedure is sometimes called static routing.
- Adaptive algorithms, in contrast, change their routing decisions to reflect changes in the topology, and usually the traffic as well.

Routing Algorithms

List out the advantages and disadvantages of fixed routing.

- The advantages of fixed routing are as follows.
- The routes are always fixed and hence the routing overhead is minimum.
- The routing is dependent on network topology, i.e., static in nature.
- Routing is same for datagram and virtual circuit type of services.

The major disadvantages are:

- Lack of flexibility.
- The system is not robust. In case of link failure or node failure, the system cannot recover.
- Congestion may occur on a particular route.

Routing Algorithms

What is flooding? Why flooding technique is not commonly used for routing?

Flooding is one type of non-adaptive routing technique where no network information is used. In case of flooding as each node receives a packet, it is re-transmitted or forwarded to all the links connected to the node (except the link through which the packet has arrived).

Flooding is not commonly used for routing for the following reasons:

- Flooding leads to unbounded number of packets
- May lead to congestion in the network
- A number of copies of the same packet is delivered at the destination node

 U3.19

Routing Algorithms

Define Autonomous Systems.

A routing domain generally is considered a portion of an internet under common administrative authority that is regulated by a particular set of administrative guidelines. Routing domains are also called **autonomous systems**.

 U3.20

Routing Algorithms

Differentiate between Link State and Distance Vector routing algorithms.

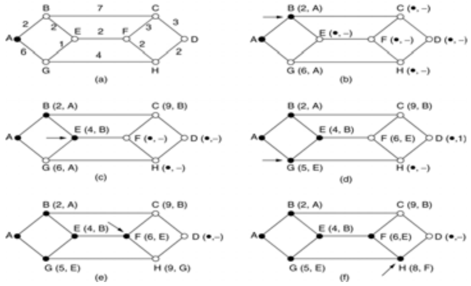
Link-state algorithms (also known as shortest path first algorithms) flood routing information to all nodes in the internetwork. Each router, however, sends only the portion of the routing table that describes the state of its own links. In link-state algorithms, each router builds a picture of the entire network in its routing tables.

Distance vector algorithms (also known as Bellman-Ford algorithms) call for each router to send all or some portion of its routing table, but only to its neighbors. In essence, link-state algorithms send small updates everywhere, while distance vector algorithms send larger updates only to neighboring routers. *Distance vector* algorithms know only about their neighbors.

 U3.21

Shortest Path Routing

The first 5 steps used in computing the shortest path from A to D.



Distance Vector Routing

- Distance vector routing algorithms operate by having each router maintain a table (i.e. a vector) giving the best known distance to each destination and which line to use to there. These tables are updated by exchanging information with the neighbors.
- The distance vector routing algorithm is sometimes called by other names, most commonly the distribute Bellman-Ford routing algorithm and the Ford-Fulkerson algorithm.

Distance Vector Routing

- In distance vector routing, each router maintains a routing table indexed by, and containing one entry for, each router in the subnet. This entry contains two parts : the proffered outgoing line to use for that destination and an estimate of the time or distance to that destination.
- The metric used might be number of hops, time delay in milliseconds, total number of packets queued along the path, or something similar.``

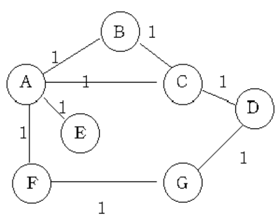
Distance Vector Routing

Distance-Vector Routing

- Each node constructs a one-dimensional array containing the "distances"(costs) to all other nodes and distributes that vector to its immediate neighbors.
- The starting assumption for distance-vector routing is that each node knows the cost of the link to each of its directly connected neighbors.
- A link that is down is assigned an infinite cost.

Distance Vector Routing

Example.



Distance Vector Routing

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

Table : Initial distances stored at each node(global view).

We can represent each node's knowledge about the distances to all other nodes as a table

Distance Vector Routing

Note that each node only knows the information in one row of the table.

1. Every node sends a message to its directly connected neighbors containing its personal list of distance. (for example, **A** sends its information to its neighbors **B,C,E**, and **F**.)

2. If any of the recipients of the information from **A** find that **A** is advertising a path shorter than the one they currently know about, they update their list to give the new path length and note that they should send packets for that destination through **A**. (node **B**learns from **A** that node **E** can be reached at a cost of 1; **B** also knows it can reach **A** at a cost of 1, so it adds these to get the cost of reaching **E** by means of **A**. **B** records that it can reach **E** at a cost of 2 by going through **A**.)

Distance Vector Routing

3. After every node has exchanged a few updates with its directly connected neighbors, all nodes will know the least-cost path to all the other nodes.

4. In addition to updating their list of distances when they receive updates, the nodes need to keep track of which node told them about the path that they used to calculate the cost, so that they can create their forwarding table. (for example, **B** knows that it was **A** who said " I can reach **E** in one hop" and so **B** puts an entry in its table that says " To reach **E**, use the link to **A**.)

Distance Vector Routing

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

Table: final distances stored at each node (global view).

Distance Vector Routing

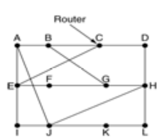
Destination	Cost	NextHop
A	1	A
C	1	C
D	2	C
E	2	A
F	2	A
G	3	A

Table : shows the complete routing table maintained at node B for the network

 U3.31

Distance Vector Routing

(a) A subnet. (b) Input from A, I, H, K, and the new routing table for J.



(a)

To	A	I	H	K	New estimated delay from J	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	New routing table for J	

Vectors received from J's four neighbors

(b)

 U3.32

Distance Vector Routing

A	B	C	D	E	
	•	•	•	•	Initially
	1	•	•	•	After 1 exchange
	1	2	•	•	After 2 exchanges
	1	2	3	•	After 3 exchanges
	1	2	3	4	After 4 exchanges

(a)

A	B	C	D	E	
	1	2	3	4	Initially
	3	2	3	4	After 1 exchange
	3	4	3	4	After 2 exchanges
	5	4	5	4	After 3 exchanges
	5	6	5	6	After 4 exchanges
	7	6	7	6	After 5 exchanges
	7	8	7	8	After 6 exchanges
		⋮			
	•	•	•	•	

(b)

The count-to-infinity problem.

 U3.33

Link State Routing

Link State Routing

- Every node knows how to reach its directly connected neighbors, and if we make sure that the totality of this knowledge is disseminated to every node, then every node will have enough knowledge of the network to determine correct routes to any destination.

 U3.34

Link State Routing

Reliable Flooding is the process of making sure that all the nodes participating in the routing protocol get a copy of the link-state information from all the other nodes. As the term " flooding" suggests, the basic idea is for a node to send its link-state information out on all of its directly connected links, with each node that receives this information forwarding it out on all of its link. This process continues until the information has reached all the nodes in the network.

 U3.35

Link State Routing

- **Link State Packet (LSP)** contains the following information**:**
  - The ID of the node that created the LSP;
  - A list of directly connected neighbors of that node, with the cost of the link to each one;
  - A sequence number;
  - A time to live (TTL) for this packet.

 U3.36

Link State Routing

- Flooding works in the following way. When a node X receives a copy of an LSP that originated at some other node Y, it checks to see if it has already stored a copy of an LSP from Y. If not, it stores the LSP. If it already has a copy, it compares the sequence numbers; if the new LSP has a larger sequence number, it is assumed to be the more recent, and that LSP is stored, replacing the old one. The new LSP is then forwarded on to all neighbors of X except the neighbor from which the LSP was just received.

 U3.37

Link State Routing

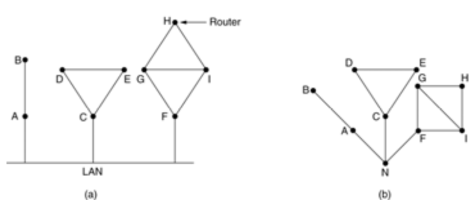
Each router must do the following:

1. Discover its neighbors, learn their network address.
2. Measure the delay or cost to each of its neighbors.
3. Construct a packet telling all it has just learned.
4. Send this packet to all other routers.
5. Compute the shortest path to every other router.

 U3.38

Learning about the Neighbors

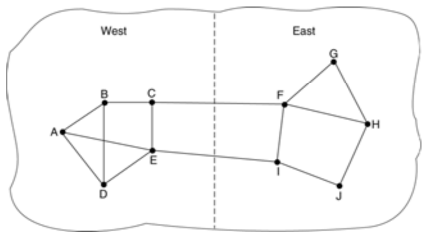
Learn about neighbours



 U3.39

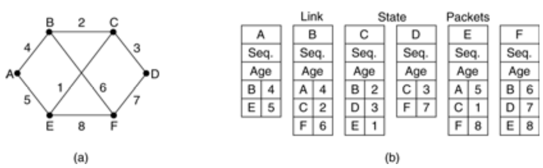
Measuring Line Cost

A subnet in which the East and West parts are connected by two lines.



 U3.40

Building Link State Packets



(a) A subnet. (b) The link state packets for this subnet.

 U3.41

Distributing the Link State Packets

The packet buffer for router B in the previous slide

			Send flags			ACK flags			
Source	Seq.	Age	A	C	F	A	C	F	Data
A	21	60	0	1	1	1	0	0	
F	21	60	1	1	0	0	0	1	
E	21	59	0	1	0	1	0	1	
C	20	60	1	0	1	0	1	0	
D	21	59	1	0	0	0	1	1	

 U3.42

Hierarchical Routing

When hierarchical routing is used, the routers are divided into what we call regions, with each router knowing all the details about how to route packets to destinations with in own region, but knowing nothing about the internal structure of other regions.

When different networks are interconnected, it is natural to regard each one as a separate region in order to free the routers in one network from having to know the topological structure of other ones.

 U3.43

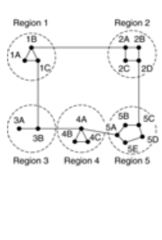
Hierarchical Routing

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.

 U3.44

Hierarchical Routing

Hierarchical routing.



(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)

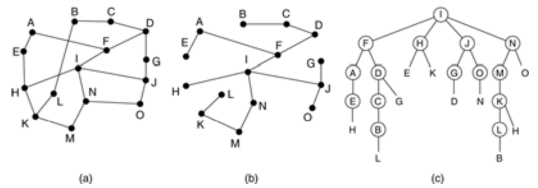
Hierarchical table for 1A

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

(c)

 U3.45

Broadcast Routing



Reverse path forwarding. (a) A subnet. (b) a Sink tree. (c) The tree built by reverse path forwarding.

 U3.46

Broadcast Routing

- In some applications, hosts need to send messages to many or all other hosts.
- For example, a service distributing weather reports, stock market updates or live radio programs.
- Sending a packet to all destinations simultaneously is called broadcasting.
- Various methods have been proposed for doing it :

 U3.47

Broadcast Routing

- One broadcasting method that requires no special features from the subnet is for the source to simply send a distinct packet to each destinations.
- Not only is the method wasteful of bandwidth, but it also requires the source to have a complete list of all destinations.
- In practice this may be the only possibility, but it is the least desirable of the methods.

 U3.48

Broadcast Routing

•Flooding is another method.

•Although Flooding is ill-suited for ordinary point-to-point communication, for broadcasting it might rate serious consideration.

•The problem with flooding as a broadcast technique is the same problem it has a point-to-point routing algorithms: it generates too many packets and consumes too much bandwidth.

Broadcast Routing

•A third algorithm is multi destination routing.

•If this method is used, each packet contains either a list of destinations or a bit map indicating the desired destinations.

•When a packet arrives at a router, the router checks all the destinations to determine the set of output lines what will be needed.

•The router regenerates a new copy of the packet for each output line to be used and includes in each packet only those destinations that are to use the line.

Broadcast Routing

•Fourth broadcast algorithm makes explicit use of the sink tree for the router initiating the broadcast- or any other convenient spanning tree for that matter.

•A spanning subtree is a subset of the subnet that includes all the routers but contains no loops.

•If each router knows which of its lines belong to the spanning tree lines except the one copy of an incoming broadcast packet onto all the spanning tree lines except the one is arrived on.

•This method makes excellent use of bandwidth, generating the absolute minimum number of packets necessary to do the job.

Broadcast Routing

•Our last broadcast algorithm is an attempt to approximate the behavior of the previous one.

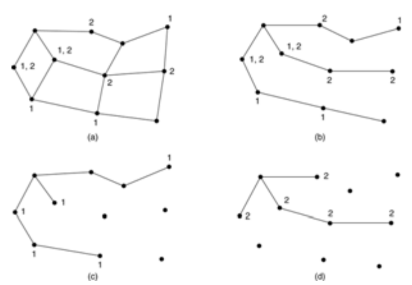
•The idea is called, reverse path forwarding, is remarkably simple once it has been pointed out..

•When a broadcast packet arrives at a router, the router checks to see if the packet arrived on the line that is normally used for sending broadcast packets to 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 the router.

•This being the case, the router forwards copies of it onto all lines except the one it arrived on.

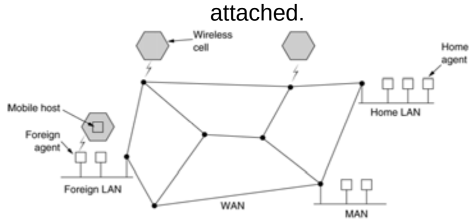
Multicast Routing



(a) A network. (b) A spanning tree for the leftmost router.
(c) A multicast tree for group 1. (d) A multicast tree for group 2.

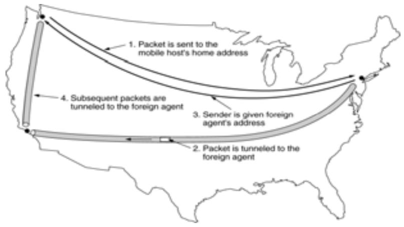
Routing for Mobile Hosts

A WAN to which LANs, MANs, and wireless cells are attached.



Routing for Mobile Hosts (2)

Packet routing for mobile users.



 U3.55

Conclusion

- Comparison of Virtual-Circuit and Datagram Subnets ids of mail and phone systems
- Routing algorithms can be classified as satic and dynamic
- Link sate and distance vector routing are commonly used in dynamic
- Routing in mobile is used for AD-HOC networks

 U3.56

Topic

Congestion Control

 U3.57

Learning Objectives

- To discuss General Principles of Congestion Control
- Congestion Prevention Policies
- Congestion Control in Virtual-Circuit Subnets
- Congestion Control in Datagram Subnets
- Load Shedding
- Jitter Control

 U3.58

Congestion

- An important issue in a packet-switched network is congestion.
- Congestion in a network may occur if the load on the network – the number of packets sent to the network – is greater than the capacity of the network – the number of packets a network can handle.
- Congestion control refers to the mechanisms and techniques to control the congestion and keep the load below the capacity.

 U3.59

Congestion Control

- Congestion control refers to techniques and mechanisms that can either prevent congestion, before it happens, or remove congestion, after it has happened.
- In general, we can divide congestion control mechanisms into two broad categories :
  - ✓ Open-Loop Congestion Control
  - ✓ Closed-Loop Control

 U3.60

Congestion Control

Open-Loop Congestion Control

- Policies are applied to prevent congestion before it happens.
- In these mechanism, congestion control is handled by either the source or destination.

 U3.61

Congestion Control

Open-Loop Congestion Control

Different policies that can prevent congestion

1. Retransmission policy
2. Window Policy
3. Acknowledgment Policy
4. Discarding Policy
5. Admission Policy

 U3.62

Congestion Control

Retransmission Policy

- Retransmission is sometime unavoidable.
- If the sender feels that a sent packet is lost or corrupted, the packet needs to be transmitted.
- Retransmission may increase congestion in the network.

 U3.63

Congestion Control

Window Policy

- This type of window at the sender may also effect congestion.
- The Selective repeat window is better than the Go-Back-N window for congestion control.
- In the Go-Back-N window, when the timer for a packet times out, several packets may be resent – This duplication may make the congestion.
- In Selective Repeat window, tries to send the specific packets that have been lost or damaged.

 U3.64

Congestion Control

Acknowledgment Policy

- The ACK policy imposed by the receiver may also affect congestion.
- If the receiver doe not ACK every packet it receives, it may slow down the sender and help prevent congestion.
- Several approaches are used in this case.
- A receiver may send an ACK only if it has a packet to be sent or a special timer expires.
- A receiver may decide to ACK only N packet at a time.
- Sending fewer ACK means imposing less load on network.

 U3.65

Congestion Control

Discarding Policy

- A good discarding policy by the routers may prevent congestion and at the same time may not hard the integrity of the transmission.
- For example, in audio transmission, if the policy is to discard less sensitive packets when congestion is likely to happen, the quality of sound is still preserved and congestion is prevented.

 U3.66

Congestion Control

Admission Policy

- An admission policy, which is a quality-of-service mechanism, can also prevent congestion in virtual circuit networks.
- Switches in a flow first check the resource requirement of a flow before admitting it to the network.
- A router can deny establishing a virtual circuit connection if there is congestion in the network or if there is a possibility of future congestion.

 U3.67

Congestion Control

Closed-Loop Congestion Control

- This mechanism try to alleviate congestion after it happens.
- Several mechanism have been used by different protocols:
  - Backpressure
  - Choke Packet
  - Implicit Signaling
  - Explicit Signaling

 U3.68

Congestion Control

Backpressure

- The technique of backpressure refers to a congestion control mechanism in which a congested node stops receiving data from the immediate upstream node.
- It is a node-to-node congestion control that starts with a node and propagates, in the opposite direction of data flow to the source.
- It can be applied only to virtual circuit networks, in which each node knows the upstream node from which a flow of data is coming.

 U3.69

Congestion Control

Choke Packet

- It is a packet sent by a node to the source to inform it of congestion.
- Choke packet scheme is a close loop mechanism where each link is monitored to examine how much utilization is taking place.
- If the utilization goes beyond a certain threshold limit, the link goes to a warning and a special packet, called **choke packet** is sent to the source.
- On receiving the choke packet, the source reduced the traffic in order to avoid congestion.

 U3.70

Congestion Control

The congestion control in the choke packet scheme can be monitored in the following manner : -

- Each link is monitored to estimate the level of utilization.
- If the utilization crosses a certain threshold limit, the link goes to a warning state and a choke packet is send to the source.
- On receiving the choke packet, the source reduces the transmitting limit to a certain level (say, by 50%).
- If still warning state persists, more choke packets are sent further reducing the traffic. This continues until the link recovers from the warning state.
- If no further choke packet is received by the source within a time interval, the traffic is increased gradually so that the system doesn't go to congestion state again.

 U3.71

Congestion Control

Implicit Signaling

- There is no communication between the congested node and the source.
- The source guesses that there is a congestion somewhere in the network from other symptoms.
- For example, when a source sends a several packets and there is no ACK for a while, one assumption is that the network is congested.
- The delay in receiving an ACK is interpreted as congestion in the network.
- The source should slow down.

 U3.72

Congestion Control

Explicit Signaling

- The node that experiences congestion can explicitly send a signal to the source or destination.
- Explicit signaling is differ than choke packet –
  - ✓In a choke packet method, a separate packet is used for this purpose
  - ✓In a explicit signaling, the signal is included in the packets that carry data.

Congestion Control

Explicit Signaling

- Explicit Signaling can occur in either the forward or backward direction :-

Backward Signaling – A bit can be set in a packet moving in the direction opposite to the congestion. This bit can warn the source that there is congestion and that it needs to slow down to avoid the discarding of packets.

Forward Signaling – A bit can be set in a packet moving in the direction of the congestion. This bit can warn the destination that there is congestion. Receiver can use ACK policies , slow down the ACK, to alleviate the congestion.

Congestion Prevention Policies

Layer	Policies
Transport	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy • Timeout determination
Network	• Virtual circuits versus datagram inside the subnet • Packet queueing and service policy • Packet discard policy • Routing algorithm • Packet lifetime management
Data link	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy

Conclusion

- Congestion Prevention Policies are diffent at different layers
- Congestion Control in Virtual-Circuit Subnets
- Congestion Control in Datagram Subnets
- Choke Packets can be sent and load can be shed
- Jitter can be high or low

Topic

Quality Of Service

Quality of Service

- To discuss the requirements of each networks
- Techniques for Achieving Good Quality of Service
- Integrated Services
- Differentiated Services
- Label Switching and MPLS

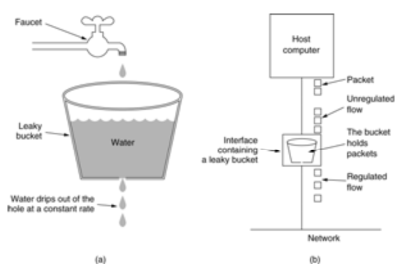
Requirements

How stringent the quality-of-service requirements are.

Application	Reliability	Delay	Jitter	Bandwidth
E-mail	High	Low	Low	Low
File transfer	High	Low	Low	Medium
Web access	High	Medium	Low	Medium
Remote login	High	Medium	Medium	Low
Audio on demand	Low	Low	High	Medium
Video on demand	Low	Low	High	High
Telephony	Low	High	High	Low
Videoconferencing	Low	High	High	High

 U3.79

The Leaky Bucket Algorithm



(a) A leaky bucket with water. (b) a leaky bucket with packets.

 U3.80

The Leaky Bucket Algorithm

The following steps are performed:

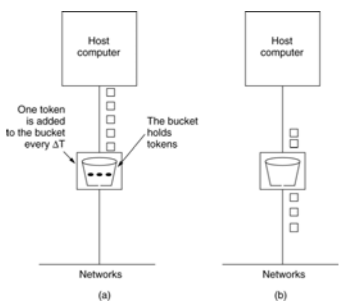
- When the host has to send a packet, the packet is thrown into the bucket.
- The bucket leaks at a constant rate, meaning the network interface transmits packets at a constant rate.
- Bursty traffic is converted to a uniform traffic by the leaky bucket.
- In practice the bucket is a finite queue that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

 U3.81

The Leaky Bucket Algorithm

- This arrangement can be simulated in the operating system or can be built into the hardware.
- Implementation of this algorithm is easy and consists of a finite queue.
- Whenever a packet arrives, if there is room in the queue it is queued up and if there is no room then the packet is discarded.

The Token Bucket Algorithm



(a) Before. (b) After.

The Token Bucket Algorithm

- The leaky bucket algorithm described above, enforces a rigid pattern at the output stream, irrespective of the pattern of the input.
- For many applications it is better to allow the output to speed up somewhat when a larger burst arrives than to loose the data.
- Token Bucket algorithm provides such a solution. In this algorithm leaky bucket holds token, generated at regular intervals.

The Token Bucket Algorithm

Main steps of this algorithm can be described as follows:

- In regular intervals tokens are thrown into the bucket.
- The bucket has a maximum capacity.
- If there is a ready packet, a token is removed from the bucket, and the packet is send.
- If there is no token in the bucket, the packet cannot be send.

The Token Bucket Algorithm

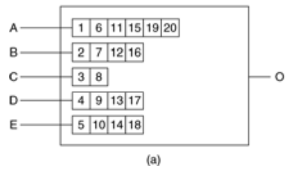
- The token bucket algorithm is less restrictive than the leaky bucket algorithm, in a sense that it allows bursty traffic.
- However, the limit of burst is restricted by the number of tokens available in the bucket at a particular instant of time.
- The implementation of basic token bucket algorithm is simple; a variable is used just to count the tokens.
- This counter is incremented every t seconds and is decremented whenever a packet is sent.
- Whenever this counter reaches zero, no further packet is sent.

Admission Control

An example of flow specification.

Parameter	Unit
Token bucket rate	Bytes/sec
Token bucket size	Bytes
Peak data rate	Bytes/sec
Minimum packet size	Bytes
Maximum packet size	Bytes

Packet Scheduling

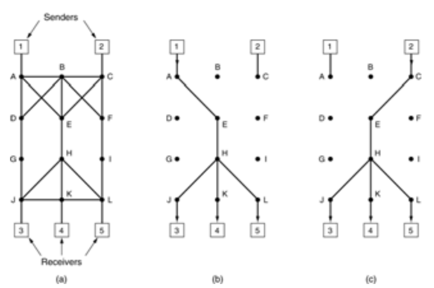


Packet	Finishing time
C	8
B	16
D	17
E	18
A	20

(b)

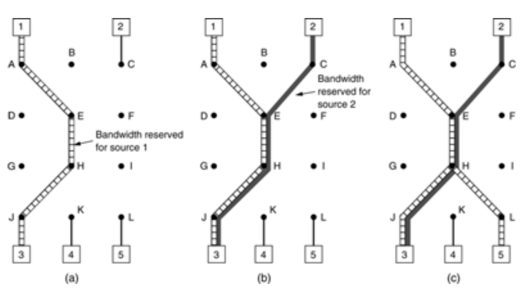
(a) A router with five packets queued for line O.
(b) Finishing times for the five packets.

RSVP-The ReSerVation Protocol



(a) A network, (b) The multicast spanning tree for host 1.
(c) The multicast spanning tree for host 2.

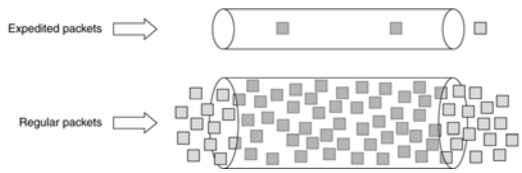
RSVP-The ReSerVation Protocol



(a) Host 3 requests a channel to host 1. (b) Host 3 then requests a second channel, to host 2. (c) Host 5 requests a channel to host 1.

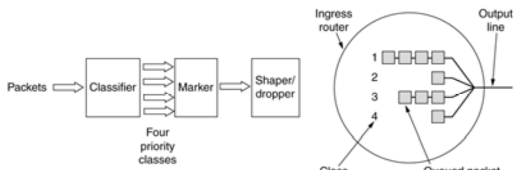
Expedited Forwarding

Expedited packets experience a traffic-free network.



Assured Forwarding

A possible implementation of the data flow for assured forwarding.



Conclusion

- Network differ in various ways
- Multiple networks are connected together and problems can occur
- Techniques for Achieving Good Quality of Service
- Services can be Integrated Services and differtiated services

Topic

Internetworking

Internetworking

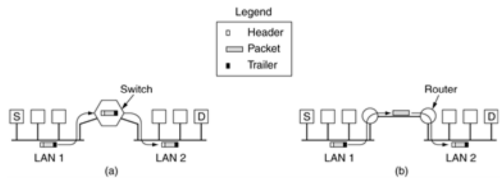
- How Networks Differ
- How Networks Can Be Connected
- Concatenated Virtual Circuits
- Connectionless Internetworking
- Tunneling
- Internetwork Routing
- Fragmentation

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

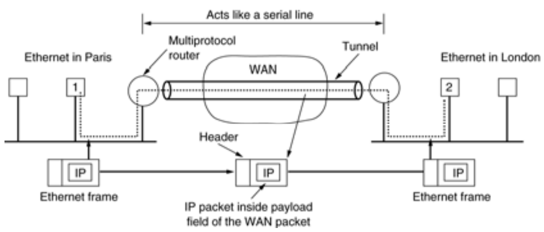
How Networks Can Be Connected

(a) Two Ethernets connected by a switch.
(b) Two Ethernets connected by routers.



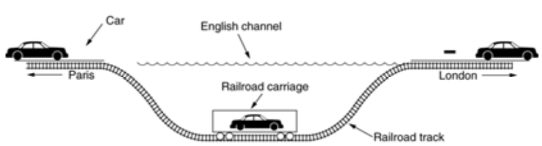
Tunneling

Tunneling a packet from Paris to London.



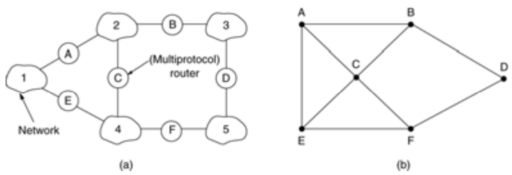
Tunneling

Tunneling a car from France to England.



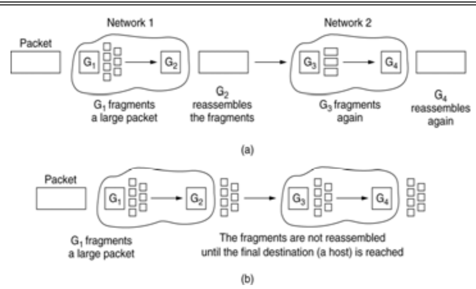
Internetwork Routing

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



 U3.100

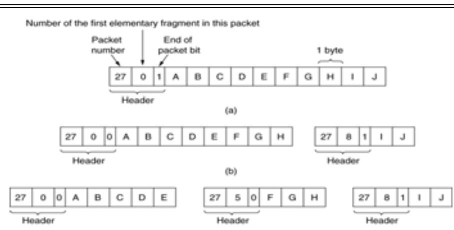
Fragmentation



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

 U3.101

Fragmentation



 U3.102

Internetworks

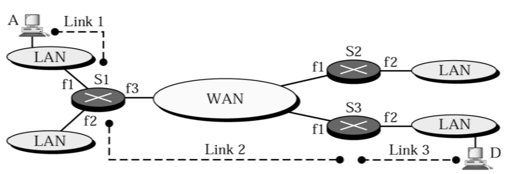
Need For Network Layer

Internet As A Packet-Switched Network

Internet As A Connectionless Network

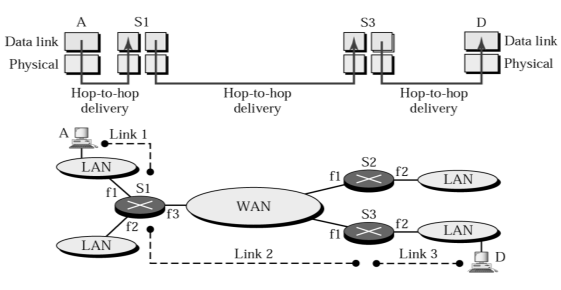
 U3.103

Internetwork



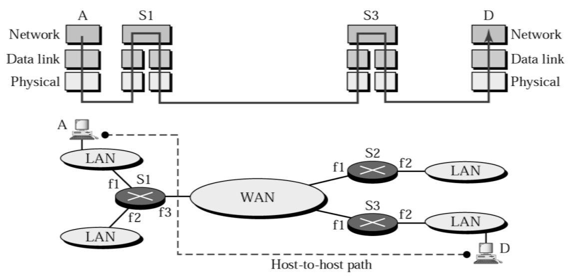
 U3.104

Links in an internetwork



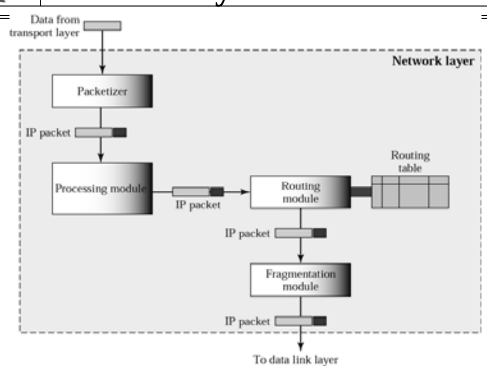
 U3.105

Network layer in an internetwork



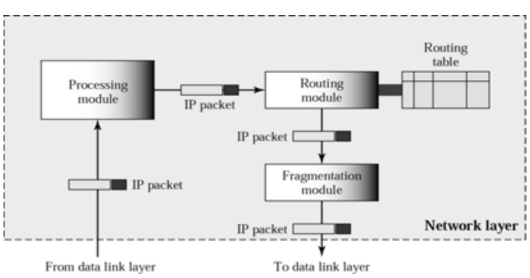
 U3.106

Network layer at the source



 U3.107

Network layer at a router



 U3.108

Conclusion

- Different networks can be fitnessed by tunneling
- When Networks Differ fragmentation may be called for.

 U3.109

Topic

Addressing

 U3.110

Learnig Objective

- ***Internet Address***
- ***Classful Addressing***
- ***Subnetting***
- ***Supernetting***
- ***Classless Addressing***
- ***Dynamic Address Configuration***
- ***Network Address Translation***

 U3.111

Note:

An IP address is a 32-bit address.

 U3.112

Note:

The IP addresses are unique and universal.

 U3.113

10000000 00001011 00000011 00011111

128.11.3.31

 U3.114

Note:

The binary, decimal, and hexadecimal number systems are reviewed in Appendix B.

 U3.115

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

a. 10000001 00001011 00001011 11101111

b. 11111001 10011011 11111011 00001111

Solution

We replace each group of 8 bits with its equivalent decimal number (see Appendix B) and add dots for separation:
a. 129.11.11.239
b. 249.155.251.15

 U3.116

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

a. 111.56.45.78

b. 75.45.34.78

Solution

We replace each decimal number with its binary equivalent (see Appendix B):

a. 01101111 00111000 00101101 01001110
b. 01001011 00101101 00100010 01001110

 U3.117

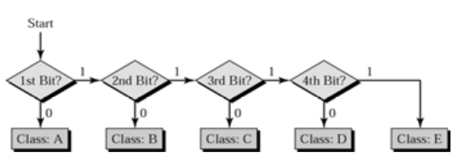
Note:

In classful addressing, the address space is divided into five classes: A, B, C, D, and E.

 U3.118

	First byte	Second byte	Third byte	Fourth byte
Class A	**0**			
Class B	**10**			
Class C	**110**			
Class D	**1110**			
Class E	**1111**			

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Vishal Jain U3.119



© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Vishal Jain U3.120

Find the class of each address:

a. **0**0000001 00001011 00001011 11101111

b. **1111**0011 10011011 11111011 00001111

Solution

See the procedure in Figure 19.11.

a. The first bit is 0; this is a class A address.
b. The first 4 bits are 1s; this is a class E address.

Finding the class in decimal notation

	First byte	Second byte	Third byte	Fourth byte
Class A	**0 to 127**			
Class B	**128 to 191**			
Class C	**192 to 223**			
Class D	**224 to 239**			
Class E	**240 to 255**			

Find the class of each address:

a. **227**.12.14.87

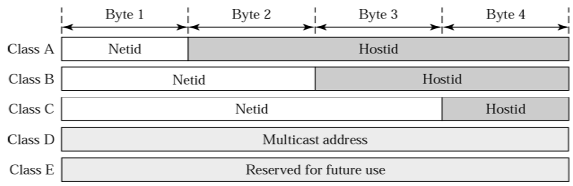
b. **252**.5.15.111

c. **134**.11.78.56

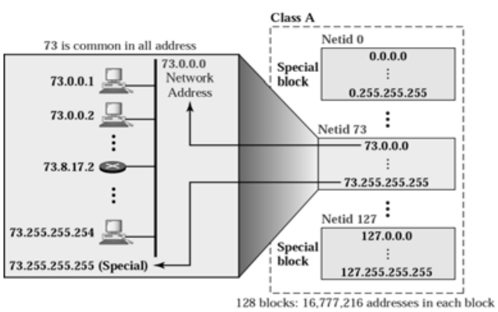
Solution

a. The first byte is 227 (between 224 and 239); the class is D.
b. The first byte is 252 (between 240 and 255); the class is E.
c. The first byte is 134 (between 128 and 191); the class is B.

Netid and hostid



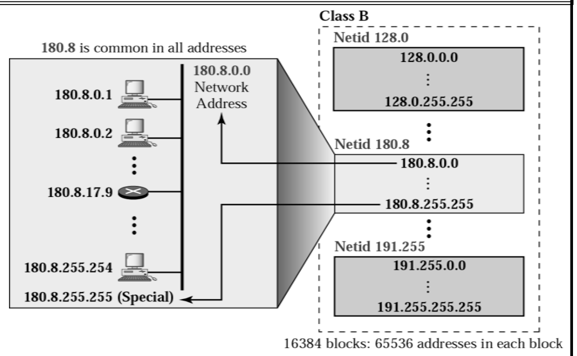
Blocks in class A



Note:

Millions of class A addresses are wasted.

Blocks in class B



 U3.127

Note:

Many class B addresses are wasted.

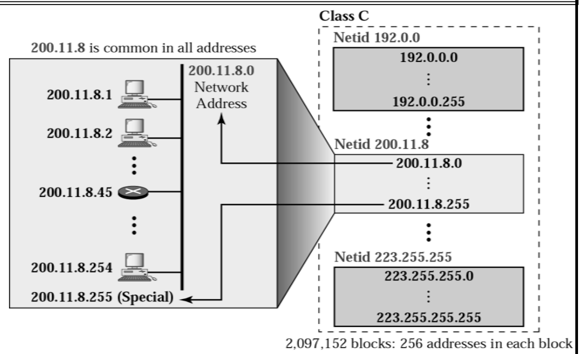
 U3.128

Note:

The number of addresses in class C is smaller than the needs of most organizations.

 U3.129

Blocks in class C



 U3.130

Note:

In classful addressing, the network address is the one that is assigned to the organization.

 U3.131

Given the address 23.56.7.91, find the network address.

Solution

The class is A. Only the first byte defines the netid. We can find the network address by replacing the hostid bytes (56.7.91) with 0s. Therefore, the network address is 23.0.0.0.

 U3.132

Given the address 132.6.17.85, find the network address.

Solution

The class is B. The first 2 bytes defines the netid. We can find the network address by replacing the hostid bytes (17.85) with 0s. Therefore, the network address is 132.6.0.0.

Given the network address 17.0.0.0, find the class.

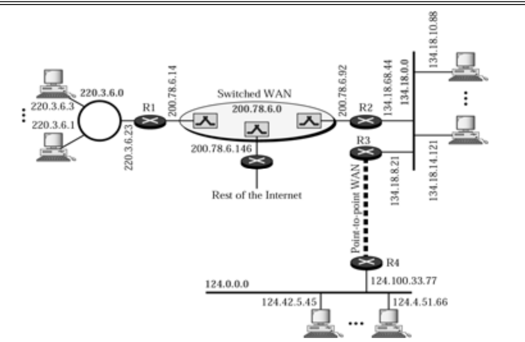
Solution

The class is A because the netid is only 1 byte.

Note:

A network address is different from a netid. A network address has both netid and hostid, with 0s for the hostid.

Sample internet



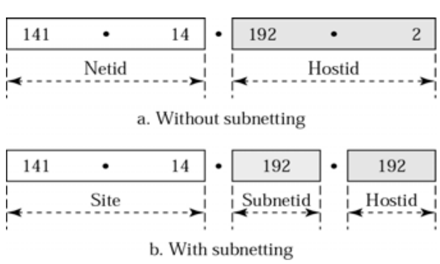
 U3.136

Note:

IP addresses are designed with two levels of hierarchy.

 U3.137

Addresses in a network with and without subnetting



 U3.138

Conclusion

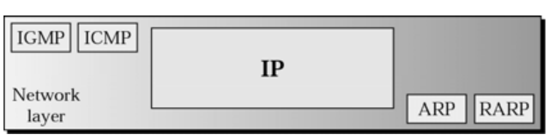
- The IP address is 32 bits
- There are 5 IP addresses. Classes A, B and C differ in number of hosts. Class D for multicasting, Class E is reserved.
- Subnetting divides one large network into several smaller ones.

Topic

Network Layer Protocols:

ARP, IPv4, ICMPv4, IPv6, and ICMPv6

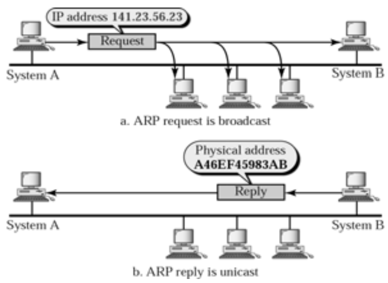
Protocols at network layer



ARP

- *Mapping*
- *Packet Format*
- *Encapsulation*
- *Operation*

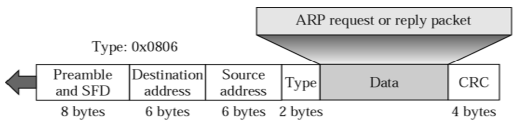
ARP operation



ARP packet

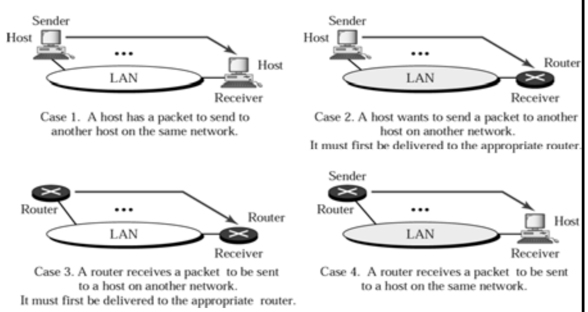
Hardware Type		Protocol Type
Hardware length	Protocol length	Operation Request 1, Reply 2
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled in a request)		
Target protocol address (For example, 4 bytes for IP)		

Encapsulation of ARP packet



 U3.145

Four cases using ARP



 U3.146

Note:

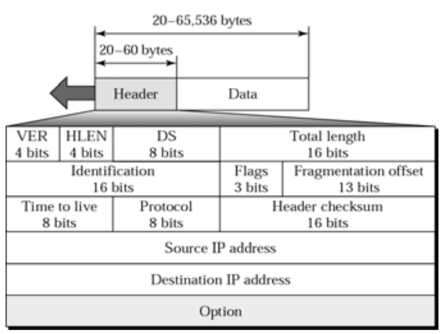
An ARP request is broadcast; an ARP reply is unicast.

 U3.147

IP

- ***Datagram***
- ***Fragmentation***

IP datagram



IP datagram

A brief description of each of the fields are given below:

VER (4 bits): Version of the IP protocol in use (typically 4).

HLEN (4 bits): Length of the header, expressed as the number of 32-bit words. Minimum size is 5, and maximum 15.

Total Length (16 bits): Length in bytes of the datagram, including headers. Maximum datagram size is (216) 65536 bytes.

Service Type (8 bits): Allows packet to be assigned a priority. Router can use this field to route packets. Not universally used.

IP datagram

Time to Live (8 bits): Prevents a packet from traveling forever in a loop. Senders sets a value, that is decremented at each hop. If it reaches zero, packet is discarded.

• **Protocol:** Defines the higher level protocol that uses the service of the IP layer

• **Source IP address (32 bits):** Internet address of the sender.

• **Destination IP address (32 bits):** Internet address of the destination.
• **Identification, Flags, Fragment Offset:** Used for handling fragmentation.
• **Options (variable width):** Can be used to provide more functionality to the IP datagram

IP datagram

Header Checksum (16 bits):

o Covers only the IP header.

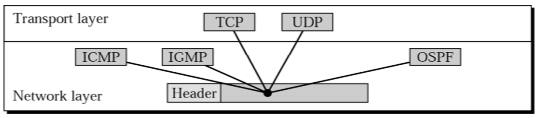
Steps:

o Header treated as a sequence of 16-bit integers
o The integers are all added using ones complement arithmetic
o Ones complement of the final sum is taken as the checksum
o Datagram is discarded in case of mismatch in checksum values

Note:

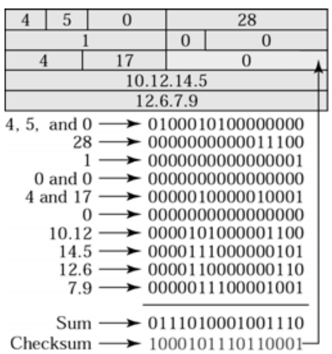
The total length field defines the total length of the datagram including the header.

Multiplexing



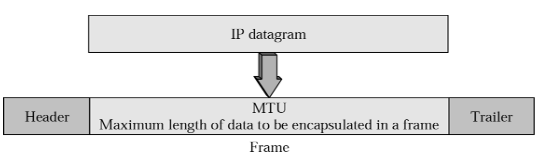
 U3.154

Example of checksum calculation



 U3.155

MTU



 U3.156

MTU

- Each network imposes a limit on maximum size, known as maximum transfer unit (MTU) of a packet because of various reasons.
- One approach is to prevent the problem to occur in the first place, i.e. send packets smaller than the MTU.
- Second approach is to deal with the problem using fragmentation. When a gateway connects two networks that have different maximum and or minimum packet sizes, it is necessary to allow the gateway to break packets up into fragments, sending each one as an internet packet.
- The technique is known as fragmentation.

 U3.157

MTU

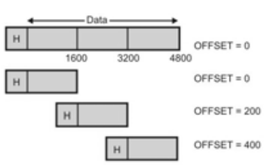
The following fields of an IP datagram are related to fragmentation:

- **Identification:** A16-bit field identifies a datagram originating from the source host.
- **Flags:** There are 3 bits, the first bit is reserved, the second bit is do not fragment bit, and the last bit is more fragment bit.
- **Fragmentation offset:** This 13-bit field shows the relative position of the segment with respect to the complete datagram measured in units of 8 bytes.

 U3.158

Fragmentation example

Fragmentation example, where a packet is fragmented into packets of 1600 bytes. So, the offset of the second fragmented packet is 1600/8 = 200 and the offset of the third fragmented packet is 400 and so on.



 U3.159

ICMP

- Introduction
- Messages
- Debugging Tools
- ICMP Package

 U3.160

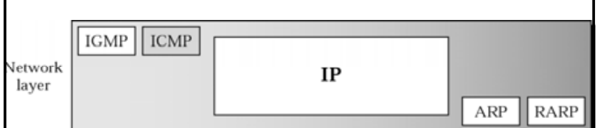
ICMP

- To make efficient use of the network resources, IP was designed to provide unreliable and connectionless best-effort datagram delivery service. As a consequence, IP has no error-control mechanism and also lacks mechanism for host and management queries. A companion protocol known as *Internet Control Message Protocol* (ICMP), has been designed to compensate these two deficiencies.

 U3.161

ICMP

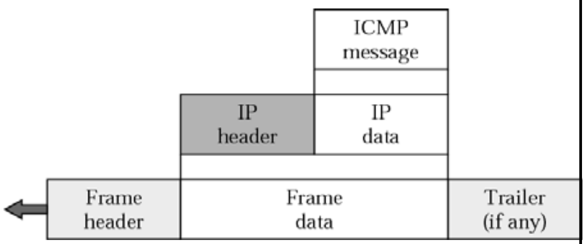
- **Position of ICMP in the network layer**



 U3.162

ICMP

•**ICMP encapsulation**



 U3.163

ICMP

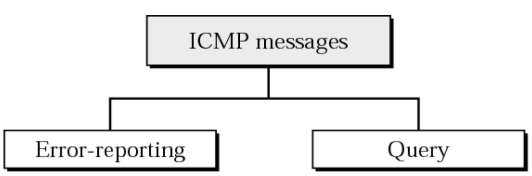
•**ICMP Messages**

ICMP messages are divided into error-reporting messages and query messages. The error-reporting messages report problems that a router or a host (destination) may encounter. The query messages get specific information from a router or another host.

 U3.164

ICMP

•**ICMP Messages**



 U3.165

ICMP

ICMP messages

Category	Type	Message
Error-reporting messages	3	Destination unreachable
	4	Source quench
	11	Time exceeded
	12	Parameter problem
	5	Redirection

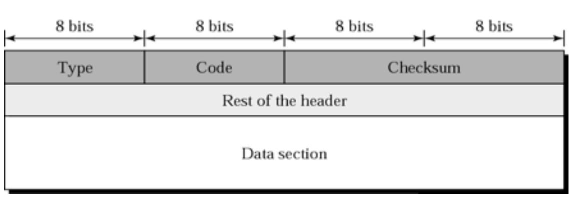
ICMP

Message Format

An ICMP message has an 8-byte header and a variable-size data section. Although the general format of the header is different for each message type, the first 4 bytes are common to all.

ICMP

Message Format



ICMP

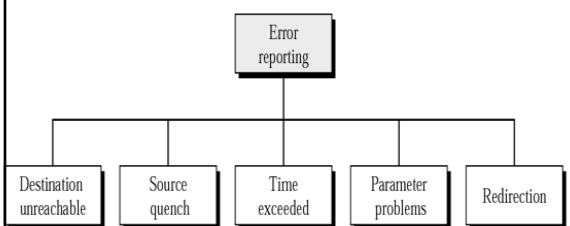
ERROR REPORTING

IP, as an unreliable protocol, is not concerned with error checking and error control. ICMP was designed, in part, to compensate for this shortcoming. ICMP does not correct errors, it simply reports them.

ICMP always reports error messages to the original source.

 U3.169

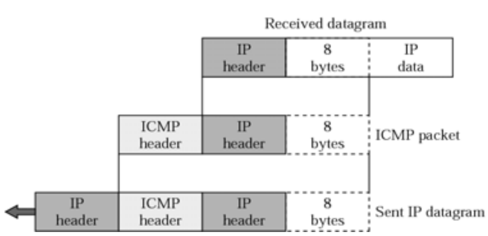
ICMP



 U3.170

ICMP

Contents of data field for the error messages



 U3.171

ICMP

Destination-unreachable format

Type: 3	Code: 0 to 15	Checksum
Unused (All 0s)		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

ICMP

Destination-unreachable format

Destination-unreachable messages with codes 2 or 3 can be created only by the destination host.

Other destination-unreachable messages can be created only by routers.

ICMP

Source-quench format

Type: 4	Code: 0	Checksum
Unused (All 0s)		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

ICMP

Source-quench format

A source-quench message informs the source that a datagram has been discarded due to congestion in a router or the destination host.

The source must slow down the sending of datagrams until the congestion is relieved.

 U3.175

ICMP

Source-quench format

One source-quench message is sent for each datagram that is discarded due to congestion.

 U3.176

ICMP

Time-exceeded message

Whenever a router decrements a datagram with a time-to-live value to zero, it discards the datagram and sends a time-exceeded message to the original source.

 U3.177

ICMP

Time-exceeded message

When the final destination does not receive all of the fragments in a set time, it discards the received fragments and sends a time-exceeded message to the original source.

 U3.178

ICMP

Time-exceeded message

code 0 is used only by routers to show that the value of the time-to-live field is zero. Code 1 is used only by the destination host to show that not all of the fragments have arrived within a set time.

 U3.179

ICMP

Time-exceeded message format

Type: 11	Code: 0 or 1	Checksum
Unused (All 0s)		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

 U3.180

ICMP

Time-exceeded message format

Type: 11	Code: 0 or 1	Checksum
Unused (All 0s)		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

ICMP

Parameter-problem

A parameter-problem message can be created by a router or the destination host.

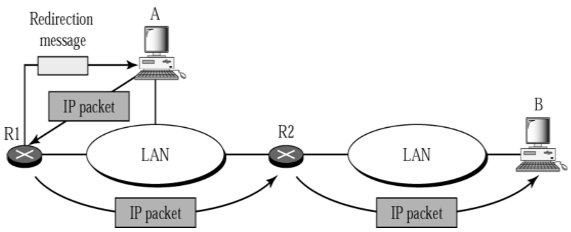
ICMP

Parameter-problem message format

Type: 12	Code: 0 or 1	Checksum
Pointer	Unused (All 0s)	
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

ICMP

Redirection concept



ICMP

Redirection concept

A host usually starts with a small routing table that is gradually augmented and updated. One of the tools to accomplish this is the redirection message.

ICMP

Redirection message format

Type: 5	Code: 0 to 3	Checksum
IP address of the target router		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

ICMP

Redirection message

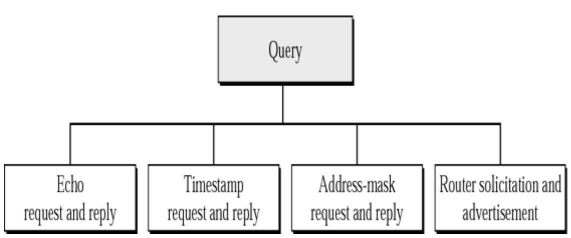
A redirection message is sent from a router to a host on the same local network.

ICMP

QUERY Messages

ICMP can also diagnose some network problems through the query messages, a group of four different pairs of messages. In this type of ICMP message, a node sends a message that is answered in a specific format by the destination node.

ICMP



ICMP

An echo-request message can be sent by a host or router. An echo-reply message is sent by the host or router which receives an echo-request message.

ICMP

Echo-request and echo-reply messages can be used by network managers to check the operation of the IP protocol.

ICMP

Echo-request and echo-reply messages can test the reachability of a host. This is usually done by invoking the ping command.

ICMP

Echo-request and echo-reply messages

8: Echo request
0: Echo reply

Type: 8 or 0	Code: 0	Checksum
Identifier		Sequence number
Optional data Sent by the request message; repeated by the reply message		

ICMP

Timestamp-request and timestamp-reply

ICMP

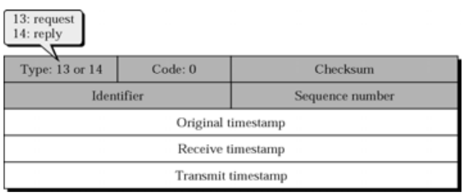
Timestamp-request and timestamp-reply messages can be used to calculate the round-trip time between a source and a destination machine even if their clocks are not synchronized.

ICMP

The timestamp-request and timestamp-reply messages can be used to synchronize two clocks in two machines if the exact one-way time duration is known.

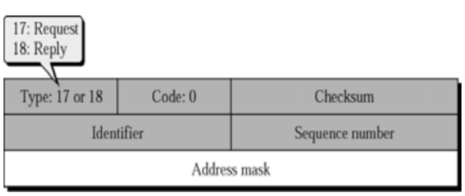
ICMP

Timestamp-request and timestamp-reply message format



ICMP

Mask-request and mask-reply message format



ICMP

Router-solicitation message format

Type: 10	Code: 0	Checksum
Identifier		Sequence number

 U3.199

ICMP

Router-advertisement message format

Type: 9	Code: 0	Checksum
Number of addresses	Address entry size	Lifetime
Router address 1		
Address preference 1		
Router address 2		
Address preference 2		
⋮		

 U3.200

ICMP

Debugging Tools

Two tools that use ICMP for debugging:
1. ping
2. traceroute.

 U3.201

ICMP

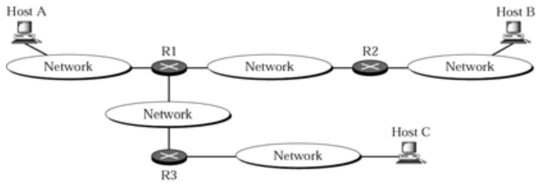
Example

$ ping fhda.edu

PING fhda.edu (153.18.8.1) 56 (84) bytes of data.
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=0 ttl=62 time=1.91 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=1 ttl=62 time=2.04 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=2 ttl=62 time=1.90 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=3 ttl=62 time=1.97 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=4 ttl=62 time=1.93 ms

ICMP

The traceroute program operation



ICMP

Example

We use the traceroute program to find the route from the computer voyager.deanza.edu to the server fhda.edu. The following shows the result:

$ traceroute fhda.edu

traceroute to fhda.edu (153.18.8.1), 30 hops max, 38 byte packets
1 Dcore.fhda.edu (153.18.31.254) 0.995 ms 0.899 ms 0.878 ms
2 Dbackup.fhda.edu (153.18.251.4) 1.039 ms 1.064 ms 1.083 ms
3 tiptoe.fhda.edu (153.18.8.1) 1.797 ms 1.642 ms 1.757 ms

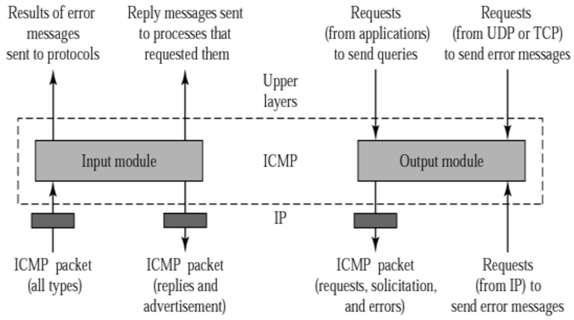
ICMP

ICMP Package

To give an idea of how ICMP can handle the sending and receiving of ICMP messages, an ICMP package made of two modules: an input module and an output module.

ICMP

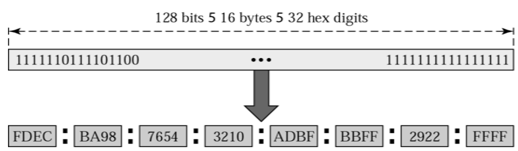
ICMP Package



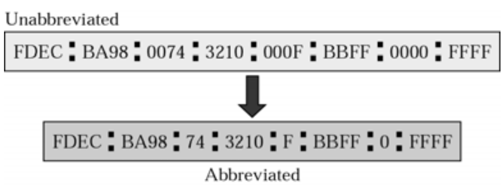
IPv6

- ***IPv6 Addresses***
- ***Categories of Addresses***
- ***IPv6 Packet Format***
- ***Fragmentation***
- ***ICMPv6***
- ***Transition***

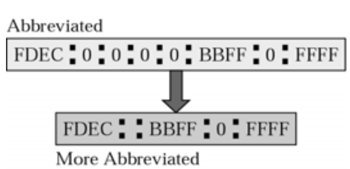
IPv6 address



Abbreviated address



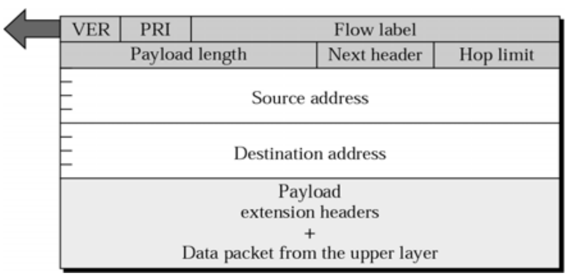
Abbreviated address with consecutive zeros

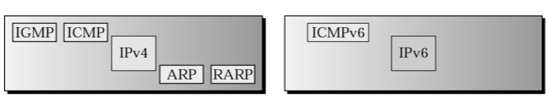


CIDR address

FDEC : 0 : 0 : 0 : 0 : BBFF : 0 : FFFF/60

Format of an IPv6 datagram





Network layer in version 4 Network layer in version 6

IPV6 Protocol

Comparison between IPv4 and IPv6 packet header

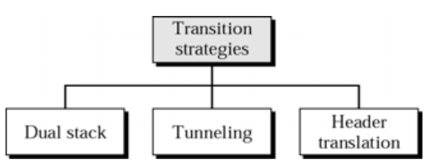
Comparison
1. The header length field is eliminated in IPv6 because the length of the header is fixed in this version.
2. The service type field is eliminated in IPv6. The priority and flow label fields together take over the function of the service type field.
3. The total length field is eliminated in IPv6 and replaced by the payload length field.

 U3.214

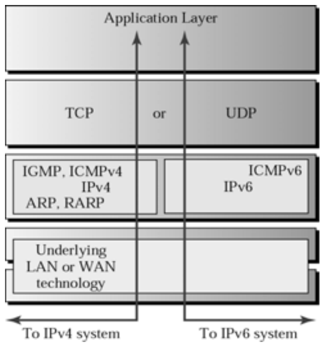
IPV6 Protocol

4. The identification, flag, and offset fields are eliminated from the base header in IPv6. They are included in the fragmentation extension header.
5. The TTL field is called hop limit in IPv6.
6. The protocol field is replaced by the next header field.
7. The header checksum is eliminated because the checksum is provided by upper layer protocols; it is therefore not needed at this level.
8. The option fields in IPv4 are implemented as extension headers in IPv6.

 U3.215

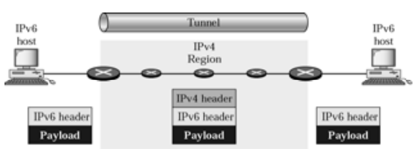


 U3.216



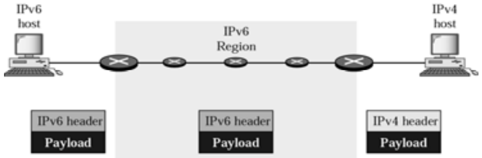
 U3.217

Tunneling



 U3.218

Header translation



 U3.219

Conclusion

- ARP request is broadcast and reply is unicast
- IP is unreliable and connectionless protocol responsible for source to destination.
- Packet in IP layer is called datagram
- Datagram header consists of a 20-60bytes and data
- ICMP messages are encapsulated in IP datagrams.

Review Questions

1. In IPv4, if the fragment offset has a value of 100, it means that _______.

A) the datagram has not been fragmented
B) the datagram is 100 bytes in size
C) the first byte of the datagram is byte 100
D) the first byte of the datagram is byte 800

2. In IPv4, what is the length of the data field given an HLEN value of 12 and total length value of 40,000?

A) 39,988
B) 40,012
C) 40,048
D) 39,952

Review Questions

3. In IPv4, what is needed to determine the number of the last byte of a fragment?

A) Identification number
B) Offset number
C) Total length
D) (b) and (c)

4. The IPv4 header size _______.

A) is 20 to 60 bytes long
B) is always 20 bytes long
C) is always 60 bytes long
D) depends on the MTU

Review Questions

5. In IPv6, the _______ field in the base header restricts the lifetime of a datagram.
A) version
B) next-header
C) hop limit
D) neighbor-advertisement

6. The ________ protocol is the transmission mechanism used by the TCP/IP suite.
A) ARP
B) IP
C) RARP
D) none of the above

Review Questions

7. IP is _________ datagram protocol.
A) an unreliable
B) a connectionless
C) both a and b
D) none of the above

8. The term ________ means that IP provides no error checking or tracking. IP assumes the unreliability of the underlying layers and does its best to get a transmission through to its destination, but with no guarantees.
A) reliable delivery
B) connection-oriented delivery
C) best-effort delivery
D) none of the above

Review Questions

9. The IPv4 header size _______.
A) is 20 to 60 bytes long
B) is 20 bytes long
C) is 60 bytes long
D) none of the above

10. IPv4, when a datagram is encapsulated in a frame, the total size of the datagram must be less than the _______.
A) MUT
B) MAT
C) MTU
D) none of the above

Review Questions

1. What is purpose of ARP. Why is ARP request broadcast? Why is ARP reply unicast ?
2. Why does IP Checksum cover just the header?
3. Explain Routing for mobile Hosts.
4. If the frame offset has value 100, then what can you say about the sequence number of first byte of the datagram.
5. Describe various messages in ICMP
6. What strategies have been devised from transition of IPV4 to IPV6.

 U3.226

Review Questions

9. What do you mean by Tunneling
10. What are various classes of IP Address ?
11. What are various techniques in achieving good quality control in Datagrams and virtual subnets ?
12. Explain Count to infinity problem in Distance vector routing protocol.
13. In IPV4 , what is length of data field given an HLEN value of 12, and total length value of 40,000.

 U3.227

Review Questions

1. Define two type of routing protocols. Discuss any two algo in deatil?
2. Define link state routing algorithm in detail. Alsop explain all its steps in detail
3. What is routing for mobile host.
4. How to networks differ? Compare and contrast them
5. How fragmentation is done on different packet sizes?

 U3.228

Review Questions

6. Discuss the services provided in detail.
7. How does choke packet work in system toreduce congestion
8. Write and explain frame format for IP in detail
9. Write messages of ICMP in detail
!0. What ir role of ARP. Discuss its feature in detail.

 U3.229

Recommended reading

1. Tanenbaum , A computer Networks: Prentice Hall
2. Stallings , High speed Networks :Printice Hall
3. Comer D. Computer Networks: Printice hall
4. Kurose, J and ross , Computer Networking : Addison Wesley

 U3.230