# UNIT – 3

# NETWORK LAYER

**SYLLABUS:**

**Network Layer:** Design issues, packet switching.

**Routing algorithms:** The Optimality Principle, Shortest Path Algorithm, Flooding, Distance Vector Routing, Link State Routing, Hierarchical Routing, Broad Cast Routing, Multicast Routing.

**Congestion Control Algorithms:** Approaches to Congestion Control, Traffic-Aware Routing, Admission Control.

**Quality of Service:** Application Requirements, Traffic Shaping.

**Internetworking:** Tunneling.

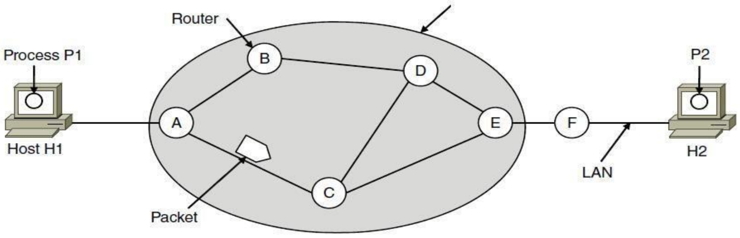
**The Network layer in the Internet:** The IP Version 4 Protocol, IP Addresses, IP Versions 6.

**Internet Control Protocols:** ARP, RARP, ICMP and DHCP.

## NETWORK LAYER DESIGN ISSUES

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

### 1. Store-and-forward packet switching



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

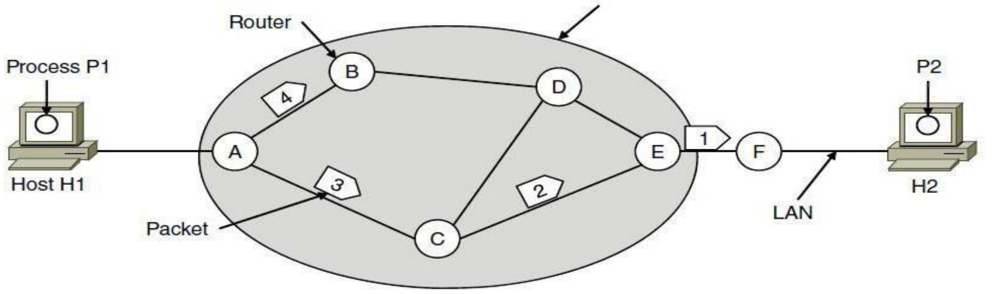
### 2. Services provided to transport layer

The network layer provides services to the transport layer at the network layer/transport layer interface. *The services need to be carefully designed with the following goals in mind:*

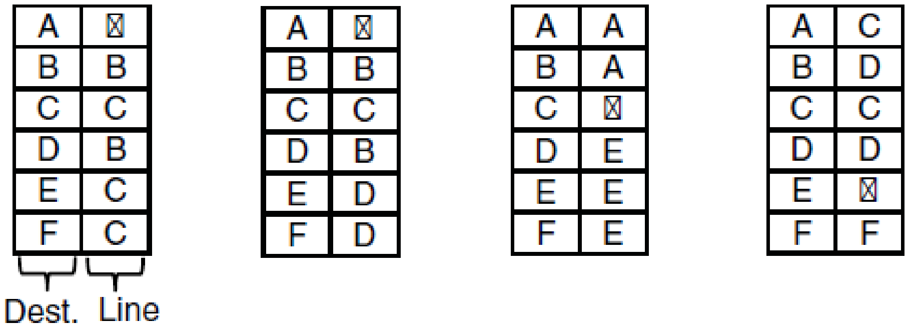
1. Services independent of router technology.
2. Transport layer shielded from number, type, topology of routers.
3. Network addresses available to transport layer use uniform numbering plan - even across LANs and WANs.

### 3. Implementation of connectionless service

If connectionless service is offered, packets are injected into the network individually and routed independently of each other. No advance setup is needed. In this context, the packetsare frequently called **datagrams** (in analogy with telegrams) and the network is called a **datagram network**.



A’s table (initially) A’s table (later) C’s Table E’s
Table



| | |
|---|---|
| A | ☒ |
| B | B |
| C | C |
| D | B |
| E | C |
| F | C |
| Dest. | Line |

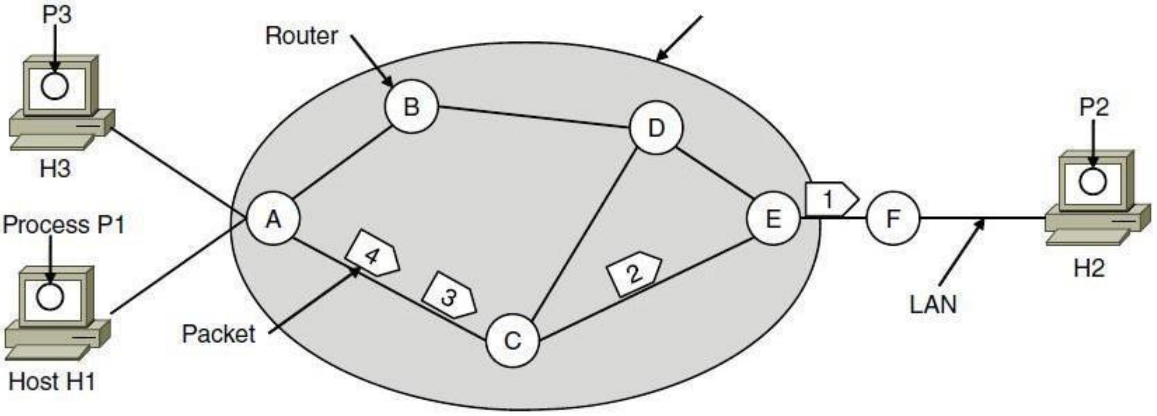
| | |
|---|---|
| A | ☒ |
| B | B |
| C | C |
| D | B |
| E | D |
| F | D |

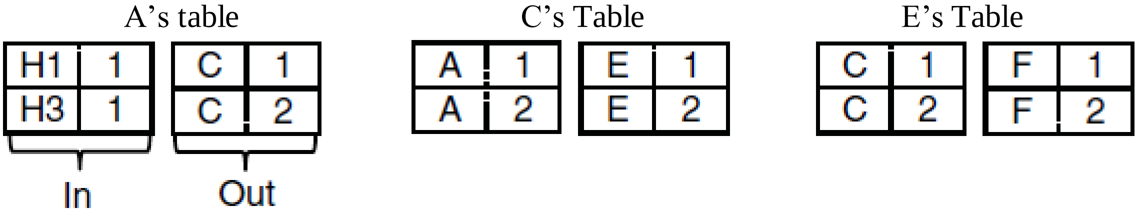
| | |
|---|---|
| A | A |
| B | A |
| C | ☒ |
| D | E |
| E | E |
| F | E |

| | |
|---|---|
| A | C |
| B | D |
| C | C |
| D | D |
| E | ☒ |
| F | F |

- 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. Each table entry is a pair(destination and the outgoing line). Only directly connected lines can be used.
- *A*’s initial routing table is shown in the figure under the label ‘‘initially.’’
- At *A*, packets 1, 2, and 3 are stored briefly, having arrived on the incoming link. Then each packetis forwarded according to *A*’s table, onto the outgoing link to *C* within a new frame. Packet 1 is then forwarded to *E* and then to *F*.
- However, something different happens to packet 4. When it gets to *A* it is sent to router *B*, even though it is also destined for *F*. For some reason (traffic jam along ACE path), *A* decided to sendpacket 4 via a different route than that of the first three packets. Router A updated its routing table, as shown under the label ‘‘later.’’
- The algorithm that manages the tables and makes the routing decisions is called the **routing algorithm**.

## 4. Implementation of connection-oriented service





A’s table

| In | | Out | |
|---|---|---|---|
| H1 | 1 | C | 1 |
| H3 | 1 | C | 2 |

C’s Table

| | | | |
|---|---|---|---|
| A | 1 | E | 1 |
| A | 2 | E | 2 |

E’s Table

| | | | |
|---|---|---|---|
| C | 1 | F | 1 |
| C | 2 | F | 2 |

- 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 iscalled a **VC** (**virtual circuit**), and the network is called a **virtual-circuit network.**

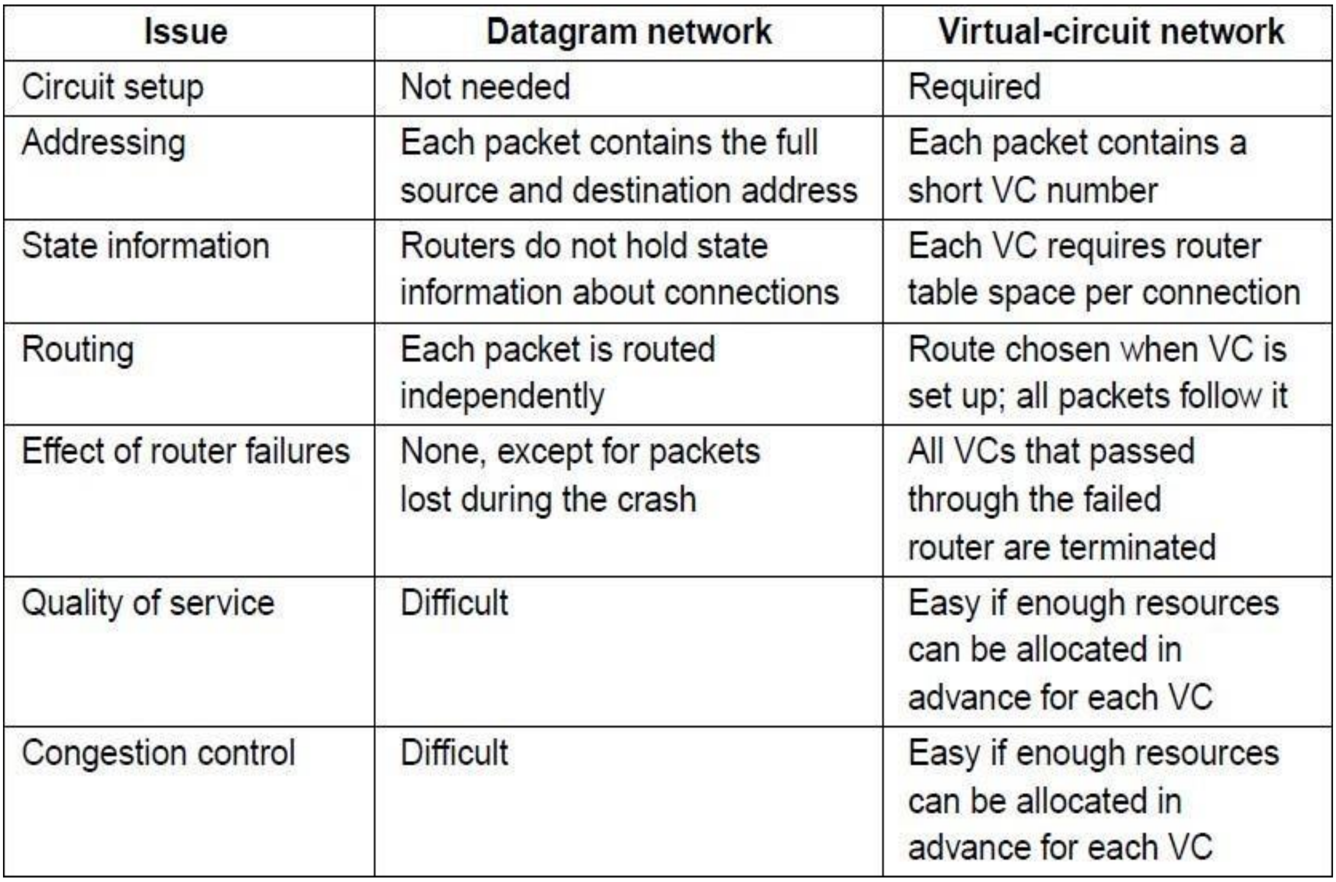
- When a connection is established, a route from the source machine to the destination machine is chosen as part of the connection setup and stored in tables inside the routers. That route is used for all traffic flowing over the connection, exactly the same way that the telephone system works.

- When the connection is released, the virtual circuit is also terminated. With connection-oriented service, each packet carries an identifier telling which virtual circuit it belongs to.

- As an example, consider the situation shown in Figure. Here, host *H1* has established connection1 with host *H2*. This connection is remembered as the first entry in each of the routing tables.

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

5. **Comparison of virtual-circuit and datagram networks**



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

## ROUTING ALGORITHMS

The main function of NL (Network Layer) is routing packets from the source machine to the destination machine.

There are two processes inside router:

- One of them handles each packet as it arrives, looking up the outgoing line to use for it inthe routing table. This process is forwarding.
- The other process is responsible for filling in and updating the routing tables. That is wherethe routing algorithm comes into play. This process is routing.

Regardless of whether routes are chosen independently for each packet or only when newconnections are established, certain properties are desirable in a routing algorithm **correctness,** simplicity, robustness, stability, fairness, optimality

Routing algorithms can be grouped into two major classes:

1. Nonadaptive (Static Routing)
2. Adaptive (Dynamic Routing)

- **Nonadaptive algorithm** do not base their routing decisions on measurements or estimates of the current traffic and topology. Instead, the choice of the route to use to get from I to J is computed in advance, off line, and downloaded to the routers when the network is booted. This procedure is sometimes called static routing.
- Adaptive algorithm, in contrast, change their routing decisions to reflect changes in the topology, and usually the traffic as well.

***Adaptive algorithms differ in***

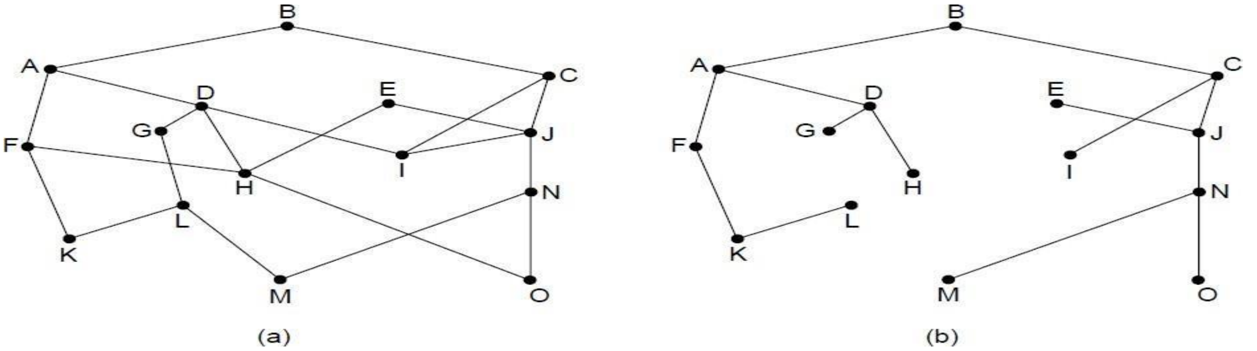
- Where they get their information (e.g., locally, from adjacent routers, or from all routers),
- When they change the routes (e.g., every ΔT sec, when the load changes or when the topology changes), and What metric is used for optimization (e.g., distance, number of hops, or estimated transittime).
- This procedure is called dynamic routing.

## Different Routing Algorithms

- Optimality principle
- Shortest path algorithm (Dijkstra's)
- Flooding
- Distance vector routing
- Link state routing
- Hierarchical Routing

## The Optimality Principle

- One can make a general statement about optimal routes without regard to network topology or traffic. This statement is known as the optimality principle.
- It states that if router J is on the optimal path from router I to router K, then the optimal path from J to K also falls along the same.
- As a direct consequence of the optimality principle, we can see that the set of optimal routes from all sources to a given destination form a tree rooted at the destination. Such a tree is called a **sink tree**.
- The goal of all routing algorithms is to discover and use the sink trees for all routers.

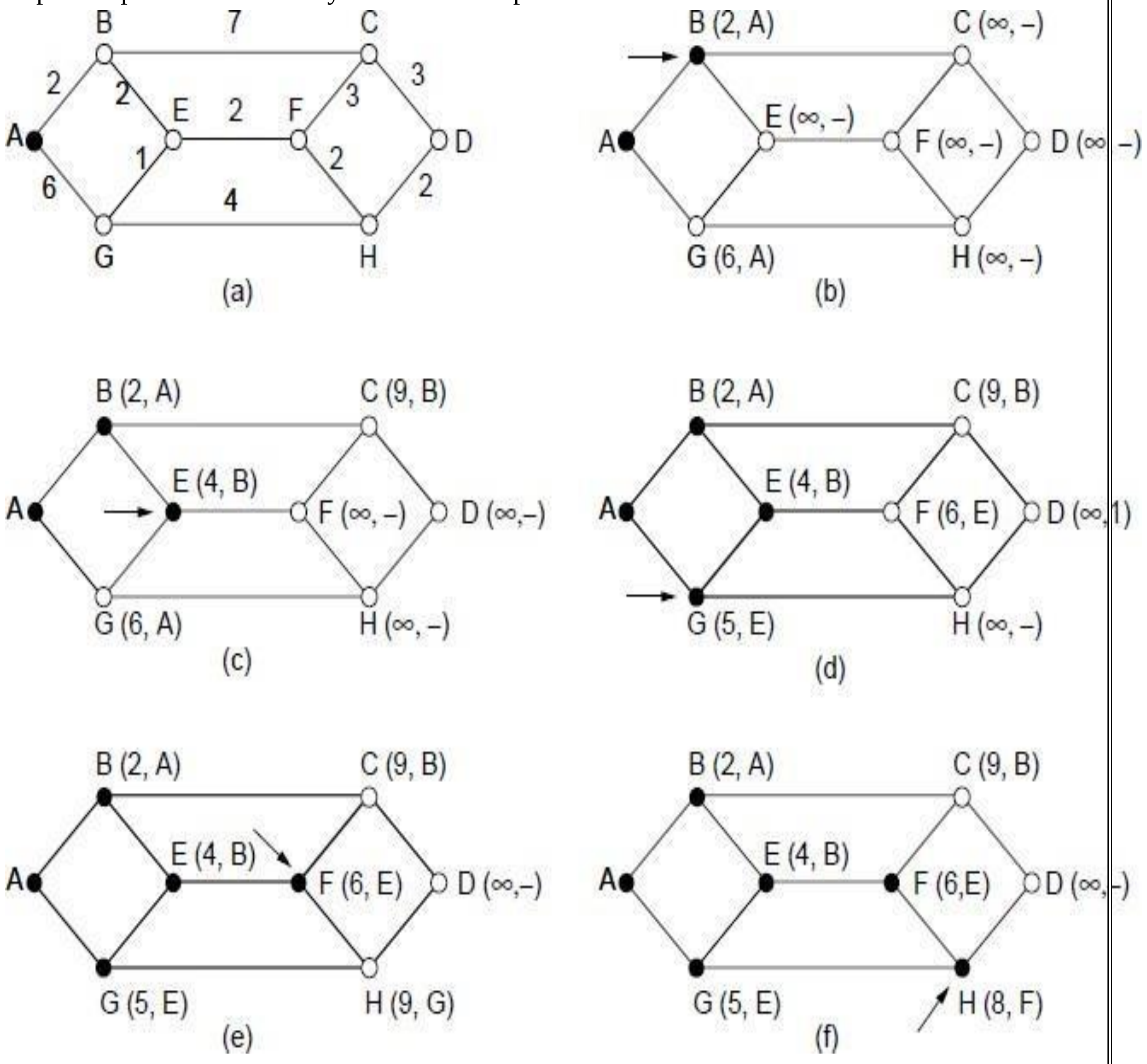


(a) A network. (b) A sink tree for router *B*.

## Shortest Path Routing (Dijkstra's)

- The idea is to build a graph of the subnet, with each node of the graph representing a router and each arc of the graph representing a communication line or link.
- To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph.
- Start with the local node (router) as the root of the tree. Assign a cost of 0 to this node and make it the first permanent node.
- Examine each neighbor of the node that was the last permanent node.
- Assign a cumulative cost to each node and make it tentative.

- Among the list of tentative nodes.
  a) Find the node with the smallest cost and make it Permanent
  b) If a node can be reached from more than one route then select the route with theshortest cumulative cost.
- Repeat steps 2 to 4 until every node becomes permanent.



(a)

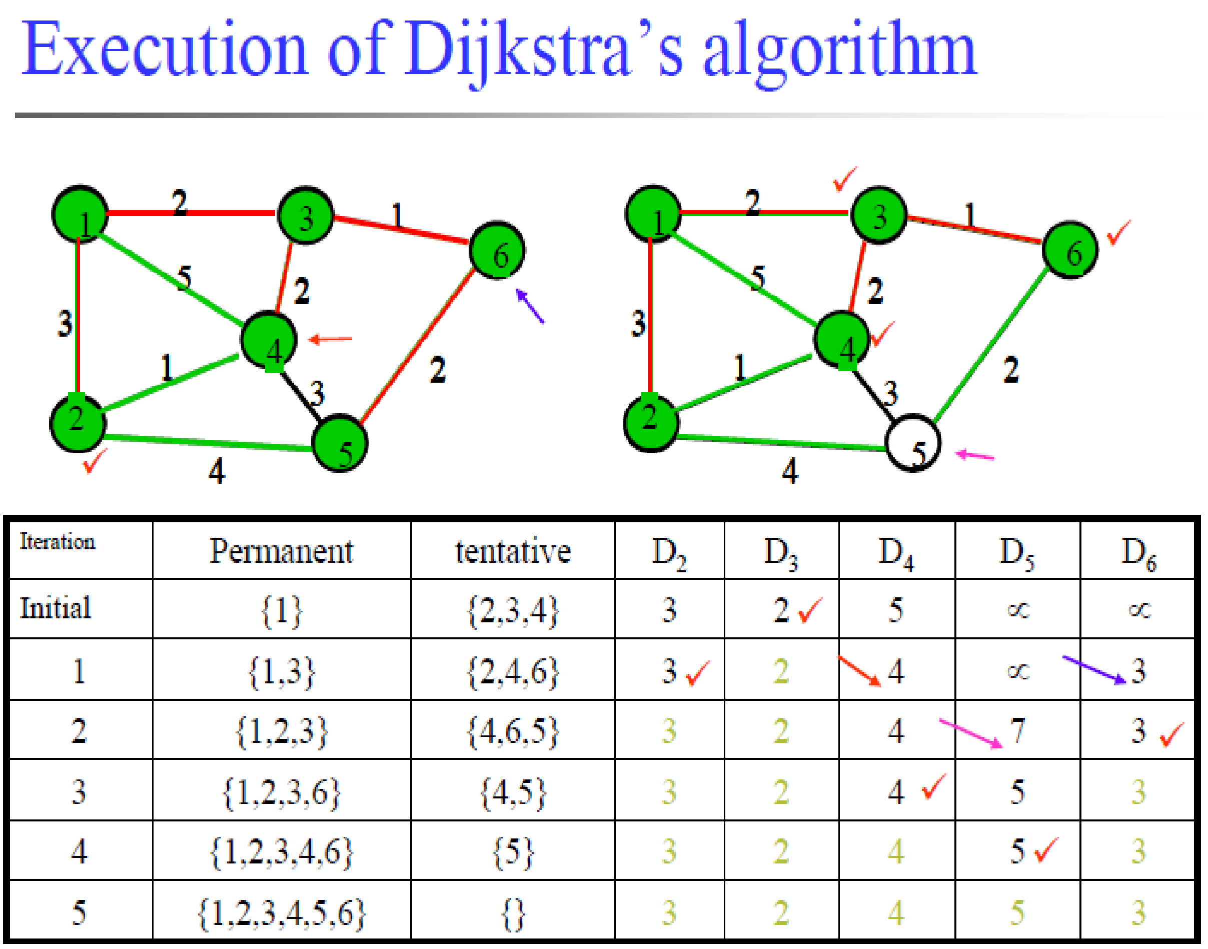
(b)

(c)

(d)

(e)

(f)



# Execution of Dijkstra's algorithm

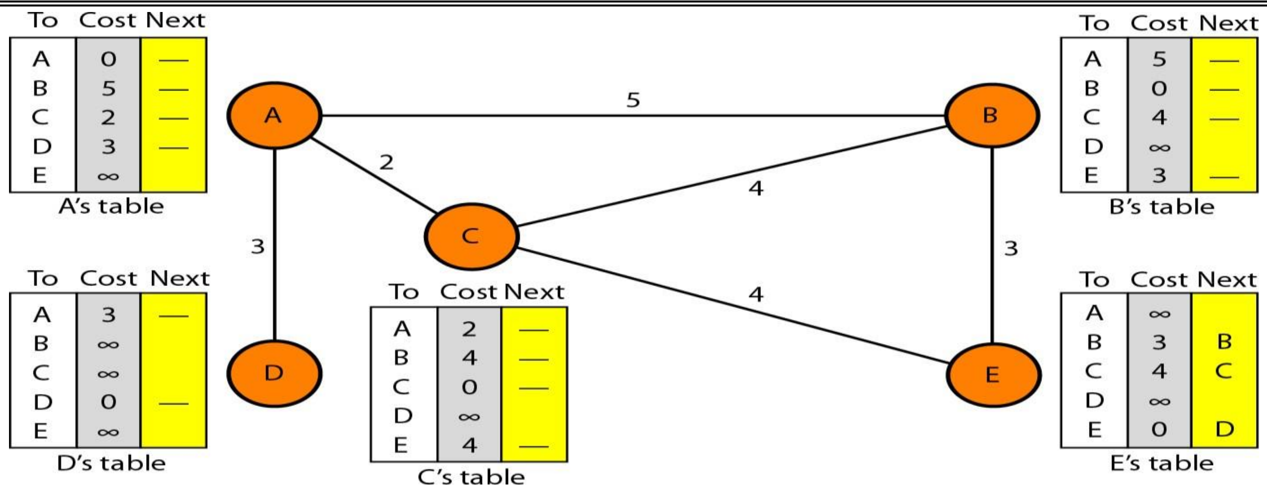
| Iteration | Permanent | tentative | $D_2$ | $D_3$ | $D_4$ | $D_5$ | $D_6$ |
|---|---|---|---|---|---|---|---|
| Initial | {1} | {2,3,4} | 3 | 2 ✓ | 5 | ∝ | ∝ |
| 1 | {1,3} | {2,4,6} | 3 ✓ | 2 | 4 | ∝ | 3 |
| 2 | {1,2,3} | {4,6,5} | 3 | 2 | 4 | 7 | 3 ✓ |
| 3 | {1,2,3,6} | {4,5} | 3 | 2 | 4 ✓ | 5 | 3 |
| 4 | {1,2,3,4,6} | {5} | 3 | 2 | 4 | 5 ✓ | 3 |
| 5 | {1,2,3,4,5,6} | {} | 3 | 2 | 4 | 5 | 3 |

## Distance Vector Routing

- In distance vector routing, the least-cost route between any two nodes is the route with minimum distance. In this protocol, as the name implies, each node maintains a vector (table) of minimum distances to every node.
- Mainly 3 things in this
  a) Initialization
  b) Sharing
  c) Updating

***Initialization***

- Each node can know only the distance between itself and its immediate neighbors, those directlyconnected to it. So for the moment, we assume that each node can send a message to the immediate neighbors and find the distance between itself and these neighbors. Below fig showsthe initial tables for each node. The distance for any entry that is not a neighbor is marked as infinite (unreachable).
- *Initialization of tables in distance vector routing*



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

A's table

| To | Cost | Next |
|---|---|---|
| A | 5 | — |
| B | 0 | — |
| C | 4 | — |
| D | ∞ | |
| E | 3 | — |

B's table

| To | Cost | Next |
|---|---|---|
| A | 3 | — |
| B | ∞ | |
| C | ∞ | |
| D | 0 | — |
| E | ∞ | |

D's table

| To | Cost | Next |
|---|---|---|
| A | 2 | — |
| B | 4 | — |
| C | 0 | — |
| D | ∞ | |
| E | 4 | — |

C's table

| To | Cost | Next |
|---|---|---|
| A | ∞ | |
| B | 3 | B |
| C | 4 | C |
| D | ∞ | |
| E | 0 | D |

E's table

***Sharing***

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

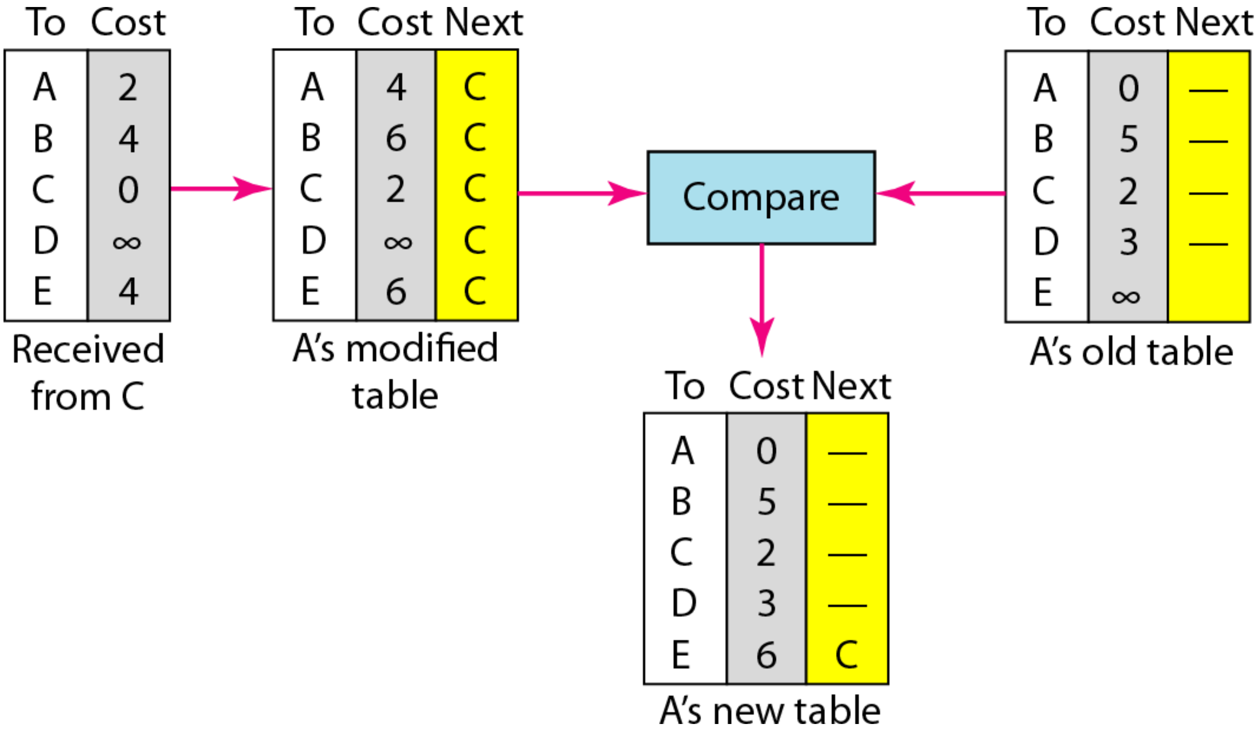
***Updating***

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

1. The receiving node needs to add the cost between itself and the sending node to each value in the second column. (x+y).
2. If the receiving node uses information from any row. The sending node is the next node in the route.
3. The receiving node needs to compare each row of its old table with the corresponding row of the modified version of the received table.
   a) If the next-node entry is different, the receiving node chooses the row with thesmaller cost. If there is a tie, the old one is kept.
   b) If the next-node entry is the same, the receiving node chooses the new row.

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

**Updating in distance vector routing**



Received from C

| To | Cost |
|---|---|
| A | 2 |
| B | 4 |
| C | 0 |
| D | ∞ |
| E | 4 |

A's modified table

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

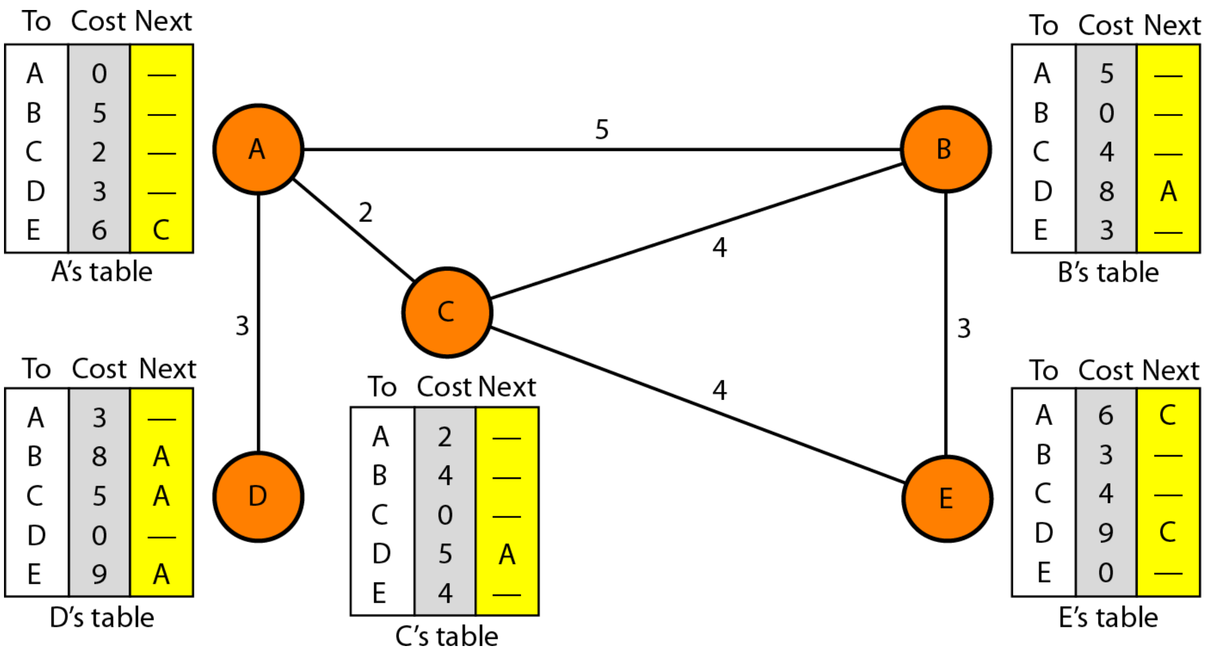
A's old table

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

A's new table

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

**Final Diagram**



A's table

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

B's table

| To | Cost | Next |
|---|---|---|
| A | 5 | — |
| B | 0 | — |
| C | 4 | — |
| D | 8 | A |
| E | 3 | — |

D's table

| To | Cost | Next |
|---|---|---|
| A | 3 | — |
| B | 8 | A |
| C | 5 | A |
| D | 0 | — |
| E | 9 | A |

C's table

| To | Cost | Next |
|---|---|---|
| A | 2 | — |
| B | 4 | — |
| C | 0 | — |
| D | 5 | A |
| E | 4 | — |

E's table

| To | Cost | Next |
|---|---|---|
| A | 6 | C |
| B | 3 | — |
| C | 4 | — |
| D | 9 | C |
| E | 0 | — |

***When to Share***

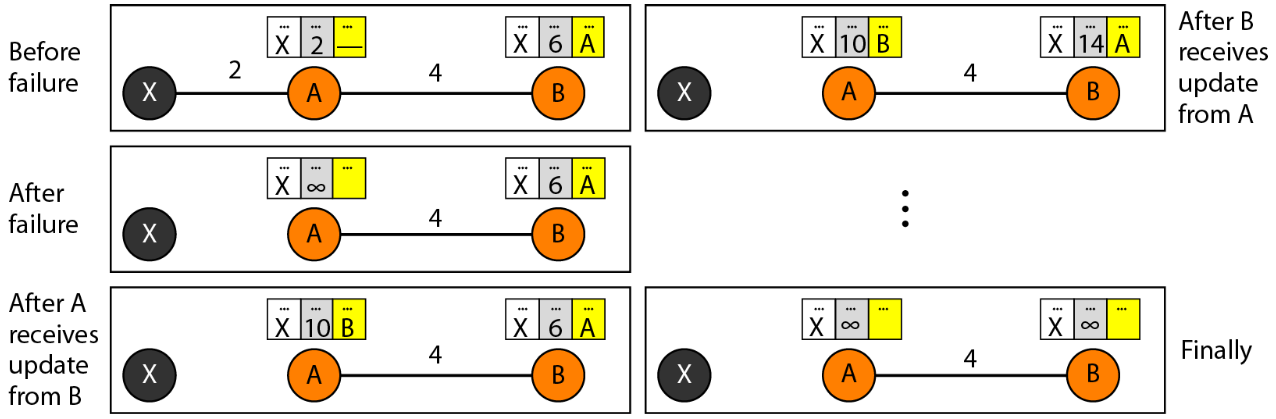
The table is sent both periodically and when there is a change in the table.

**Periodic Update** : A node sends its routing table, normally every 30 s, in a periodic update. The period depends on the protocol that is using distance vector routing.

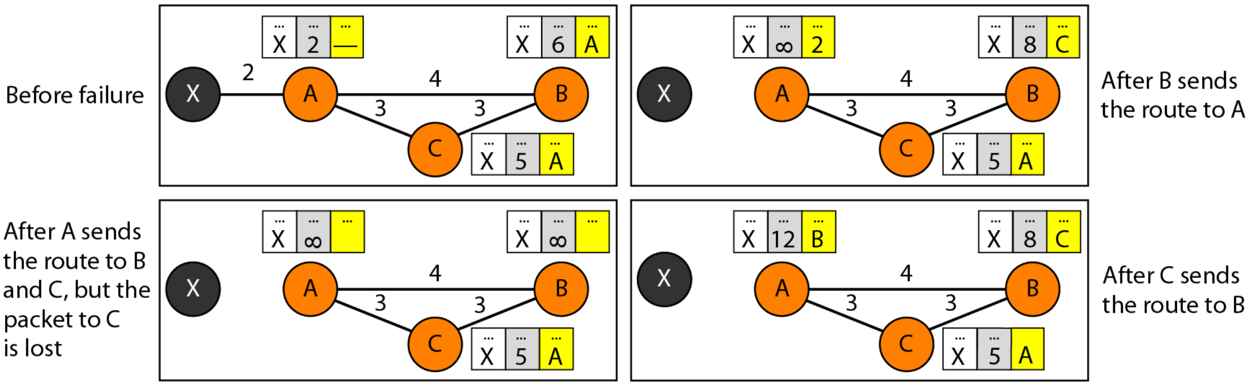
**Triggered Update** : A node sends its two-column routing table to its neighbors anytime there is a change in its routing table. This is called a triggered update. The change can result from the following.

1. A node receives a table from a neighbor, resulting in changes in its own table after updating.
2. A node detects some failure in the neighboring links which results in a distance change to infinity.

*Two-node instability*



*Three-node instability*

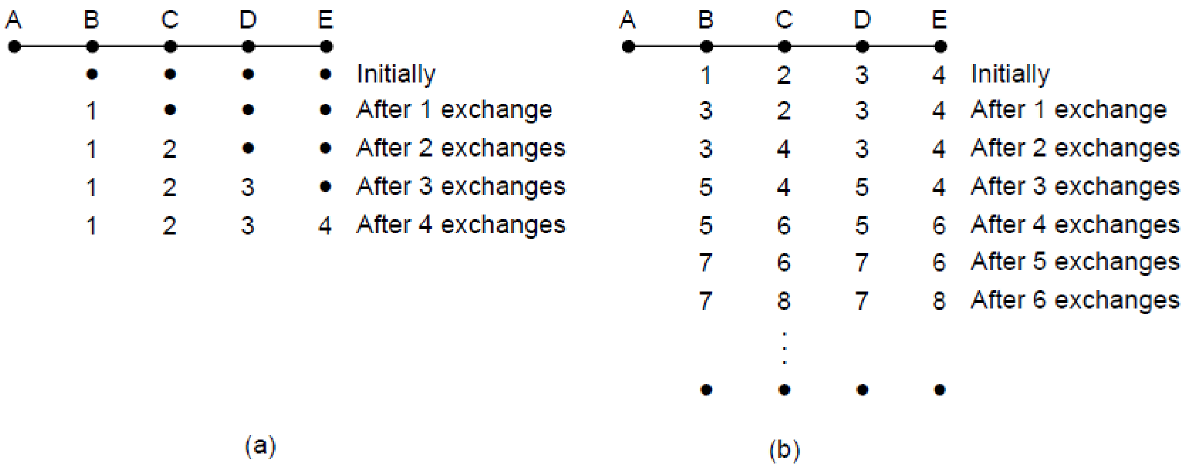

Before failure
X 2 —
X 6 A
X 5 A
2
4
3
3
X
A
B
C
After B sends the route to A
X ∞ 2
X 8 C
After A sends the route to B and C, but the packet to C is lost
X ∞
X ∞
After C sends the route to B
X 12 B

## SOLUTIONS FOR INSTABILITY

- **Defining Infinity:** redefine infinity to a smaller number, such as 100. For our previous scenario, the system will be stable in less than 20 updates. As a matter of fact, most implementations of the distance vector protocol define the distance between each node to be 1 and define 16 as infinity. However, this means that the distance vector routing cannot be used in large systems. The size of the network, in each direction, cannot exceed 15 hops.
- **Split Horizon:** In this strategy, instead of flooding the table through each interface, each node sends **only part of its table** through each interface. If, according to its table, node B thinks that the optimum route to reach X is via A, it does not need to advertise this piece of information to A; the information has come from A (A already knows). Taking information from node A, modifying it, and sending it back to node A creates the confusion. In our scenario, node B eliminates the last line of its routing table before it sends it to A. In this case, node A keeps the value of infinity as the distance to X. Later when node A sends its routing table to B, node B also corrects its routing table. The system becomes stable after the first update: both node A and B know that X is not reachable.
- **Split Horizon and Poison Reverse** Using the split horizon strategy has one drawback. Normally, the distance vector protocol uses a timer, and if there is no news about a route, the node deletes the route from its table. When node B in the previous scenario eliminates the route to X from its advertisement to A, node A cannot guess that this is due to the split horizon strategy (the source of information was A) or because B has not received any news about X recently. The split horizon strategy can be combined with the poison reverse strategy. Node B can still advertise the value for X, but if the source of information is A, it canreplace the distance with infinity as a warning: "Do not use this value; what I know about thisroute comes

from you."

- **The Count-to-Infinity Problem**



| 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)

## Advantages of Distance Vector routing

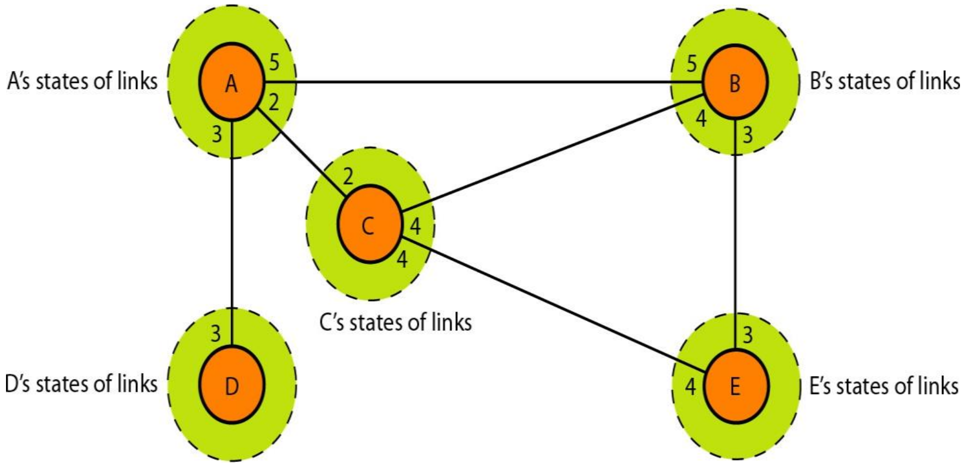
- It is simpler to configure and maintain than link state routing.

## Disadvantages of Distance Vector routing

- It is slower to converge than link state.
- It is at risk from the count-to-infinity problem.
- It creates more traffic than link state since a hop count change must be propagated to all routers and processed on each router. Hop count updates take place on a periodic basis, even if there are no changes in the network topology, so bandwidth-wasting broadcasts still occur.
- For larger networks, distance vector routing results in larger routing tables than link state since each router must know about all other routers. This can also lead to congestion on WAN links.

## LINK STATE ROUTING

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



**Building Routing Tables**

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

- **Creation of Link State Packet (LSP)** A link state packet can carry a large amount of information. For the moment, we assume that it carries a minimum amount of data: the node identity, the list of links, a sequence number, and age. The first two, node identity andthe list of links, are needed to make the topology. The third, sequence number, facilitates flooding and distinguishes new LSPs from old ones. The fourth, age, prevents old LSPs from remaining in the domain for a long time.
- LSPs are generated on two occasions:

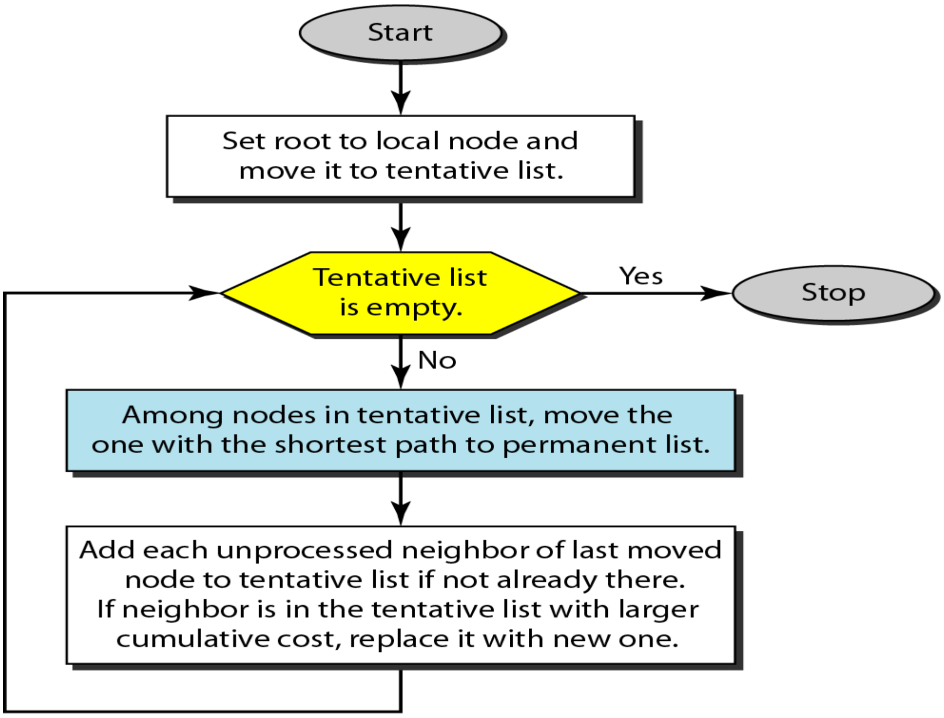
1. When there is a change in the topology of the domainon a periodic basis: The period in this case is much longer compared to distance vector.
2. The timer set for periodic dissemination is normally in the range of 60 min or 2hrs based on the implementation. A longer period ensures that flooding does not create too much trafficon the network.

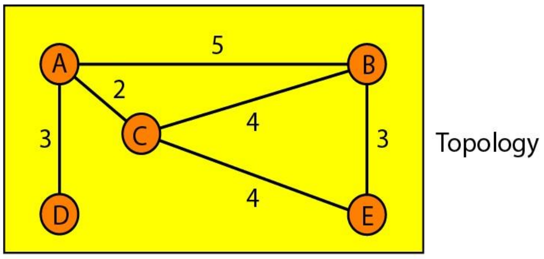
**Flooding of LSPs:**

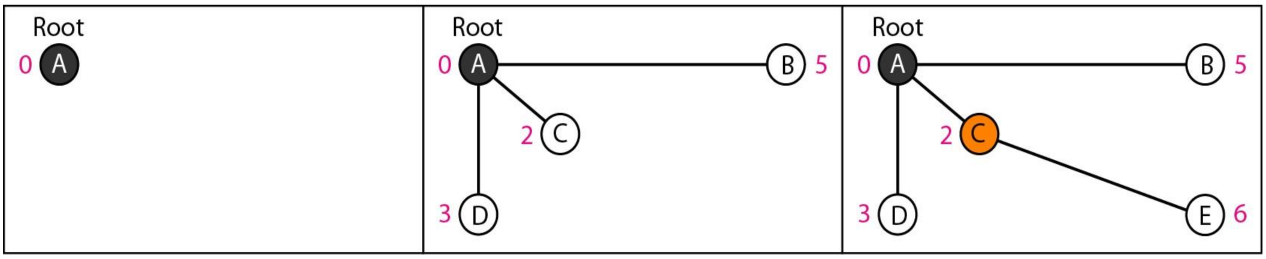
- After a node has prepared an LSP, it must be disseminated to all other nodes, not only to its neighbors. The process is called flooding and based on the following.
- The creating node sends a copy of the LSP out of each interface. A node that receives an LSP compares it with the copy it may already have. If the newlyarrived LSP is older than the one it has (found by checking the sequence number), it discards the LSP. If it is newer, the node does the following:
  a) It discards the old LSP and keeps the new one.
  b) It sends a copy of it out of each interface except the one from which the packet arrived. This guarantees that flooding stops somewhere in the domain (where a node has only one interface).

**Formation of Shortest Path Tree: Dijkstra Algorithm**

- A shortest path tree is a tree in which the path between the root and every other node is the shortest.
- The Dijkstra algorithm creates a shortest path tree from a graph.
- The algorithm divides the nodes into two sets: **tentative and permanent.** It finds the neighbors of a current node, makesthem tentative, examines them, and if they pass the criteria, makes them permanent.


Start
Set root to local node and move it to tentative list.
Tentative list is empty.
Yes
Stop
No
Among nodes in tentative list, move the one with the shortest path to permanent list.
Add each unprocessed neighbor of last moved node to tentative list if not already there.
If neighbor is in the tentative list with larger cumulative cost, replace it with new one.

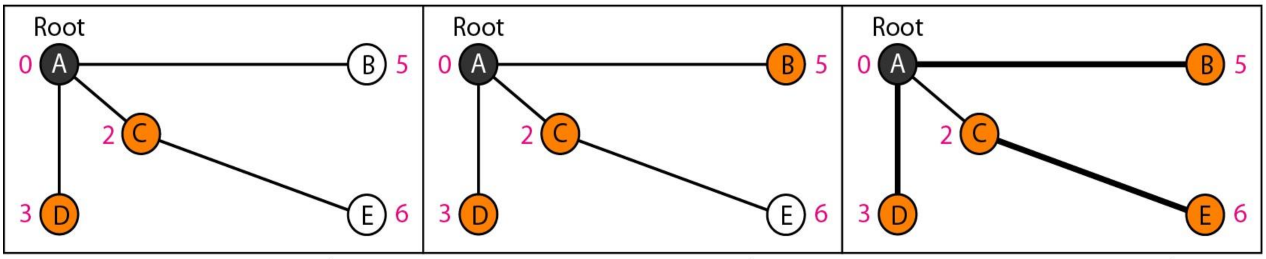




1. Set root to A and move A to tentative list.

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

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



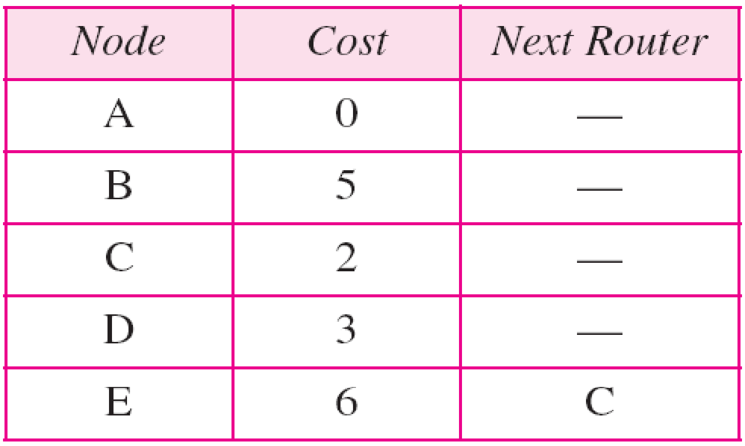
4. Move D to permanent list.

5. Move B to permanent list.

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

**Calculation of a routing table**

***Routing table for node A***



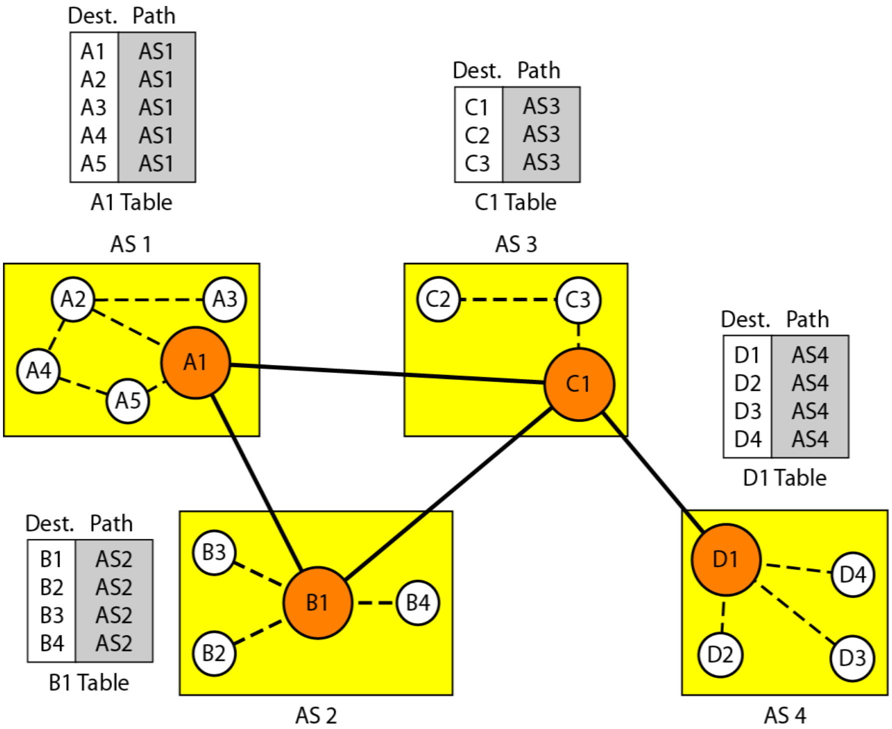
| *Node* | *Cost* | *Next Router* |
|---|---|---|
| A | 0 | — |
| B | 5 | — |
| C | 2 | — |
| D | 3 | — |
| E | 6 | C |

## Path Vector Routing

- Distance vector and link state routing are both intra domain routing protocols. They can be usedinside an autonomous system, but not between autonomous systems.
- These two protocols arenot suitable for inter domain routing mostly because of scalability. Both of these routing protocols become intractable when the domain of operation becomes large.
- Distance vector routing is subject to instability in the domain of operation.
- Link state routing needs a huge amount of resources to calculate routing tables. It also creates heavy traffic because of flooding. There is a need for a third routing protocol which we call path vector routing.
- Path vector routing proved to be useful for inter domain routing. The principle of path vector routing is similar to that of distance vector routing.
- In path vector routing, we assume that there is one node (there can be more, but one is enough for our conceptual discussion) in eachAS that acts on behalf of the entire AS.
- The speaker node in an AS creates a routing table and advertises it to speaker nodes in the neighboring ASs. The idea isthe same as for distance vector routing except that only speaker nodes in each AS can communicate with each other.
- However, what is advertised is different. A speaker node advertises the path, not the metric of the nodes, in its autonomous system or other autonomous systems.

**Initialization**

*Initial routing tables in path vector routing*



| Dest. | Path |
|---|---|
| A1 | AS1 |
| A2 | AS1 |
| A3 | AS1 |
| A4 | AS1 |
| A5 | AS1 |

A1 Table

| Dest. | Path |
|---|---|
| C1 | AS3 |
| C2 | AS3 |
| C3 | AS3 |

C1 Table

| Dest. | Path |
|---|---|
| D1 | AS4 |
| D2 | AS4 |
| D3 | AS4 |
| D4 | AS4 |

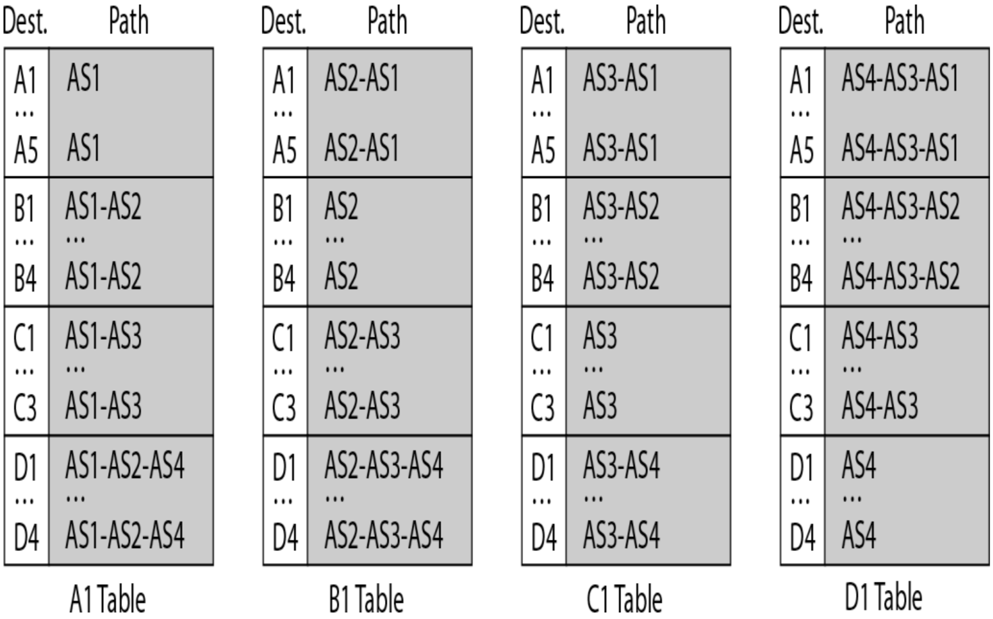
D1 Table

| Dest. | Path |
|---|---|
| B1 | AS2 |
| B2 | AS2 |
| B3 | AS2 |
| B4 | AS2 |

B1 Table

**Sharing**

Just as in distance vector routing, in path vector routing, a speaker in an autonomous system shares its table with immediate neighbors. In Figure, node A1 shares its table with nodes B1 and C1. Node C1 shares its table with nodes D1, B1, and A1. Node B1 shares its table with C1 and A1. Node D1 shares its table with C1.



| Dest. | Path |
|---|---|
| A1<br>⋯<br>A5 | AS1<br><br>AS1 |
| B1<br>⋯<br>B4 | AS1-AS2<br>⋯<br>AS1-AS2 |
| C1<br>⋯<br>C3 | AS1-AS3<br>⋯<br>AS1-AS3 |
| D1<br>⋯<br>D4 | AS1-AS2-AS4<br>⋯<br>AS1-AS2-AS4 |

A1 Table

| Dest. | Path |
|---|---|
| A1<br>⋯<br>A5 | AS2-AS1<br><br>AS2-AS1 |
| B1<br>⋯<br>B4 | AS2<br>⋯<br>AS2 |
| C1<br>⋯<br>C3 | AS2-AS3<br>⋯<br>AS2-AS3 |
| D1<br>⋯<br>D4 | AS2-AS3-AS4<br>⋯<br>AS2-AS3-AS4 |

B1 Table

| Dest. | Path |
|---|---|
| A1<br>⋯<br>A5 | AS3-AS1<br><br>AS3-AS1 |
| B1<br>⋯<br>B4 | AS3-AS2<br>⋯<br>AS3-AS2 |
| C1<br>⋯<br>C3 | AS3<br>⋯<br>AS3 |
| D1<br>⋯<br>D4 | AS3-AS4<br>⋯<br>AS3-AS4 |

C1 Table

| Dest. | Path |
|---|---|
| A1<br>⋯<br>A5 | AS4-AS3-AS1<br><br>AS4-AS3-AS1 |
| B1<br>⋯<br>B4 | AS4-AS3-AS2<br>⋯<br>AS4-AS3-AS2 |
| C1<br>⋯<br>C3 | AS4-AS3<br>⋯<br>AS4-AS3 |
| D1<br>⋯<br>D4 | AS4<br>⋯<br>AS4 |

D1 Table

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

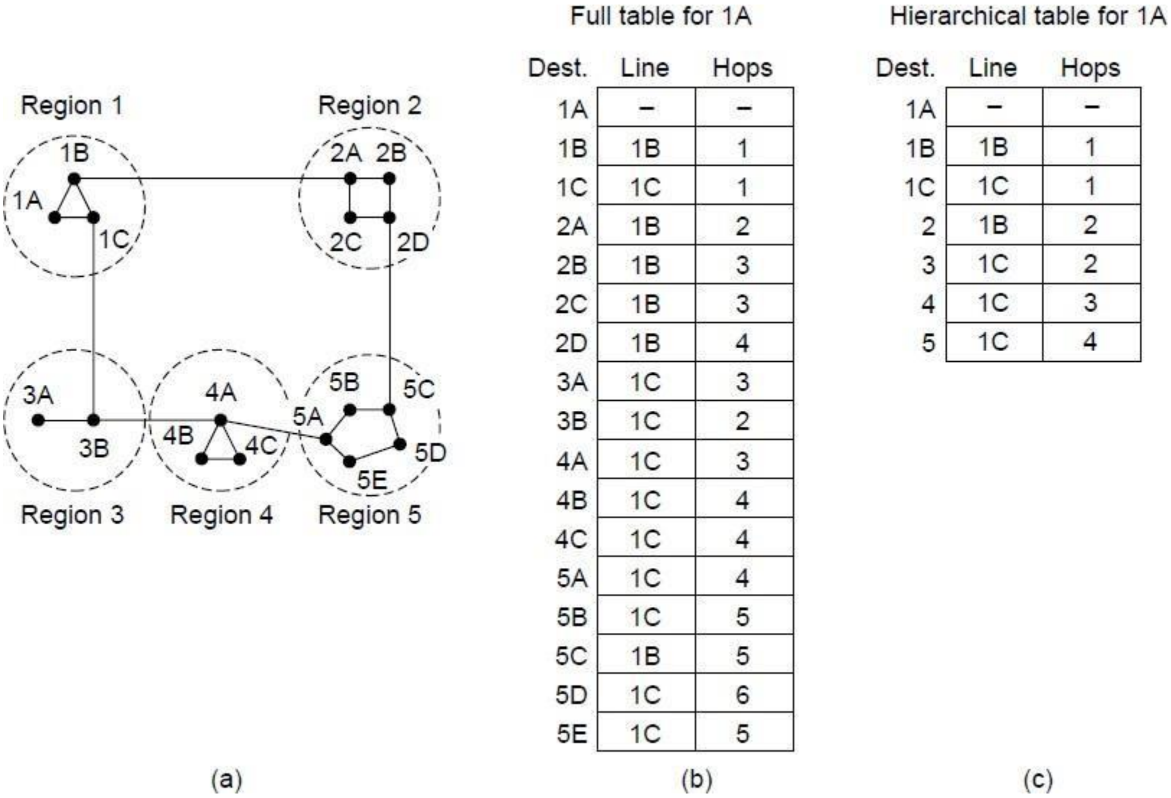
**Loop prevention:** The instability of distance vector routing and the creation of loops can be avoided in path vector routing. When a router receives a message, it checks to see if its AS isin the path list to the destination. If it is, looping is involved and the message is ignored.

**Policy routing:** Policy routing can be easily implemented through path vector routing. When a router receives a message, it can check the path. If one of the AS listed in the path is againstits policy, it can ignore that path and that destination. It does not update its routing table with this path, and it does not send this message to its neighbors.

**Optimum path:** Optimum path in path vector routing is looking for a pathto a destination that is the best for the organization that runs the AS. One system may use RIP, which defines hop count as the metric; another may use OSPF with minimum delay defined as the metric. In our previous figure, each AS may have more than one path to a destination. For example, a path from AS4 to ASI can be AS4-AS3-AS2-AS1, or it can be AS4- AS3-ASI. For the tables, we chose the one that had the smaller number of ASs, but this is not always the case. Other criteria, such as security, safety, and reliability, can also be applied.

## HIERARCHICAL ROUTING

- As networks grow in size, the router routing tables grow proportionally. Not only is router memory consumed by ever-increasing tables, but more CPU time is needed to scan them and more bandwidth is needed to send status reports about them.
- At a certain point, the network may grow to the point where it is no longer feasible for every router to have an entry for every other router, so the routing will have to be done hierarchically, as it is in the telephone network.
- When hierarchical routing is used, the routers are divided into what we will call regions. Each router knows all the details about how to route packets to destinations within its own region but knows nothing about the internal structure of other regions.
- For huge networks, a two-level hierarchy may be insufficient; it may be necessary to group theregions into clusters, the clusters into zones, the zones into groups, and so on, until we run



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

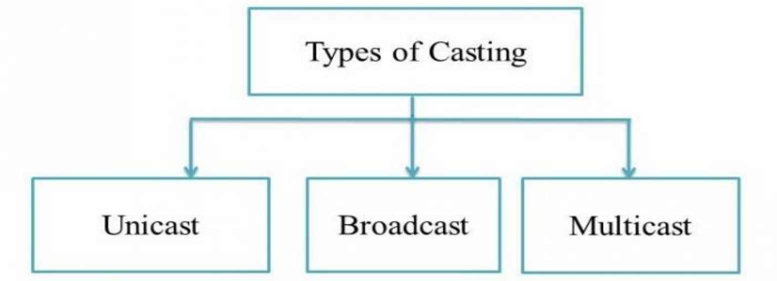
| Dest. | Line | Hops |
|---|---|---|
| 1A | – | – |
| 1B | 1B | 1 |
| 1C | 1C | 1 |
| 2 | 1B | 2 |
| 3 | 1C | 2 |
| 4 | 1C | 3 |
| 5 | 1C | 4 |

outof names for aggregations

- For example, consider a network with 720 routers. If there is no hierarchy, each router needs 720 routing table entries.
- If the network is partitioned into 24 regions of 30 routers each, each router needs 30 local entries plus 23 remote entries for a total of 53 entries.
- If a three-level hierarchy is chosen, with 8 clusters each containing 9 regions of 10 routers, each router needs 10 entries for local routers, 8 entries for routing to other regions within its own cluster, and 7 entries for distant clusters, for a total of 25 entries.
- The optimal number of levels for an N router network is ln N, requiring a total of e ln N entries per router.

## CASTING

- **Casting** in computer networks means transmitting data (stream of packets) over a network. Following are the different types of casting used in networking −
    - Unicast transmission
    - Broadcast transmission
    - Multicast transmission



### Unicast Transmission (One-to-One)

- In Unicast transmission, the data is transferred from a single sender (or a single source host) to a single receiver (or a single destination host).
- The network switches hear the MAC addresses of the devices on the networks to which they are connected. They can then forward packets only onto those networks containing devices with the connected MAC addresses.
- Unicast gradually becomes less efficient as more receivers need to see identical data.

### Example

In the following figure, Host A sends the IP address 11.1.2.2 data to the Host B IP address 20.12.4.3.

- Source Address = IP address of host A is 11.1.2.2
- Destination Address = IP address of host B is 20.12.4.3

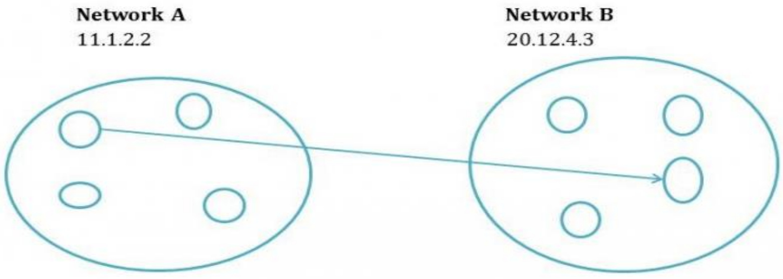


Figure: Unicast

## Broadcast Routing

- Sending a packet to all destinations simultaneously is called **broadcasting**. One broadcasting method that requires no special features from the network is for the source to simply send a distinct packet to each destination.
- Not only is the method wasteful of bandwidth and slow, but it also requires the source to have a complete list of all destinations. This method is not desirable in practice, even though it is widely applicable.
- An improvement is **multidestination routing**, in which 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 that will be needed.
- The router generates 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.
- In effect, the destination set is partitioned among the output lines. After a sufficient number of hops, each packet will carry only one destination like a normal packet.
- Multidestination routing is like using separately addressed packets, except that when several packets must follow the same route, one of them pays full fare and the rest ride free. The network bandwidth is therefore used more efficiently.

## 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.
- This being the case, the router forwards copies of it onto all links except the one it arrived on. If, however, the broadcast packet arrived on a link other than the preferred one for reaching the source, the packet is discarded as a likely duplicate.

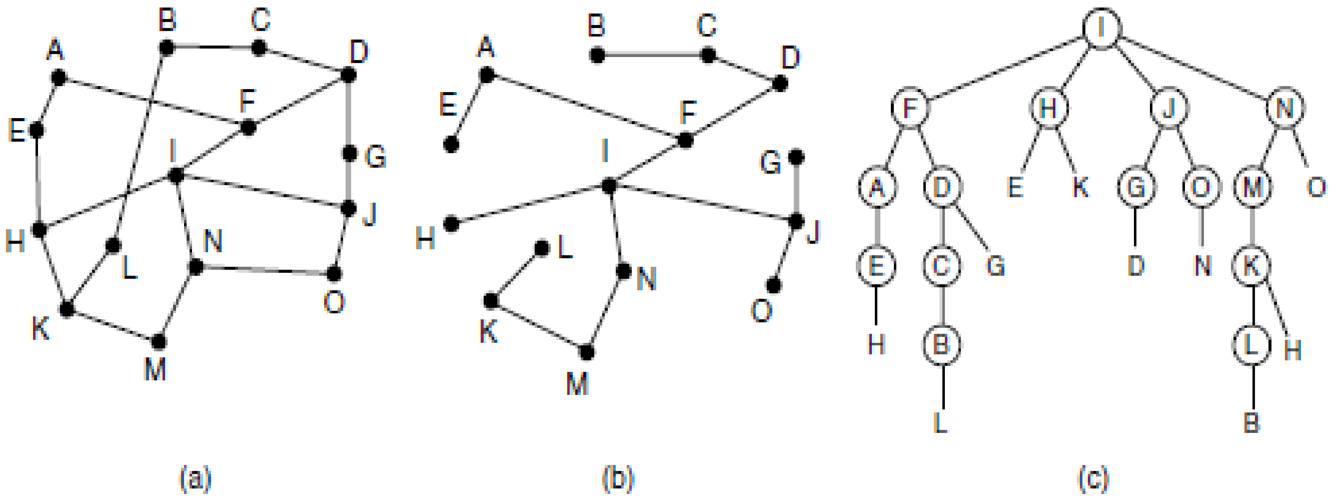


**Figure 5-15.** Reverse path forwarding. (a) A network. (b) A sink tree. (c) The tree built by reverse path forwarding.

- An example of reverse path forwarding is shown in Fig. 5-15. Part (a) shows a network, part (b) shows a sink tree for router *I* of that network, and part (c) shows how the reverse path algorithm works.

- On the first hop, *I* sends packets to *F*, *H*, *J*, and *N*, as indicated by the second row of the tree. Each of these packets arrives on the preferred path to *I* (assuming that the preferred path falls along the sink tree) and is so indicated by a circle around the letter.
- On the second hop, eight packets are generated, two by each of the routers that received a packet on the first hop. As it turns out, all eight of these arrive at previously unvisited routers, and five of these arrive along the preferred line.
- Of the six packets generated on the third hop, only three arrive on the preferred path (at *C*, *E*, and *K*); the others are duplicates. After five hops and 24 packets, the broadcasting terminates, compared with four hops and 14 packets had the sink tree been followed exactly.
- The ***principal advantage*** of reverse path forwarding is that it is efficient while being easy to implement. It sends the broadcast packet over each link only once in each direction, just as in flooding, yet it requires only that routers know how to reach all destinations, without needing to remember sequence numbers (or use other mechanisms to stop the flood) or list all destinations in the packet.
- The broadcast algorithm improves on the behavior of reverse path forwarding. It makes explicit use of the sink tree—or any other convenient spanning tree—for the router initiating the broadcast.
- A spanning tree is a subset of the network that includes all the routers but contains no loops. Sink trees are spanning trees. 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.
- This method makes excellent use of bandwidth, generating the absolute minimum number of packets necessary to do the job. The only problem is that each router must have knowledge of some spanning tree for the method to be applicable.

## MULTICAST ROUTING

- A way to send messages to well-defined groups that are numerically large in size but small compared to the network as a whole. Sending a message to such a group is called **multicasting**, and the routing algorithm used is called **multicast routing**.
- All multicasting schemes require some way to create and destroy groups and to identify which routers are members of a group the best spanning tree to use depends on whether the group is dense, with receivers scattered over most of the network, or sparse, with much of the network not belonging to the group.
- If the group is dense, broadcast is a good start because it efficiently gets the packet to all parts of the network. But broadcast will reach some routers that are not members of the group, which is wasteful.
- The solution is to prune the broadcast spanning tree by removing links that do not lead to members. The result is an efficient multicast spanning tree.

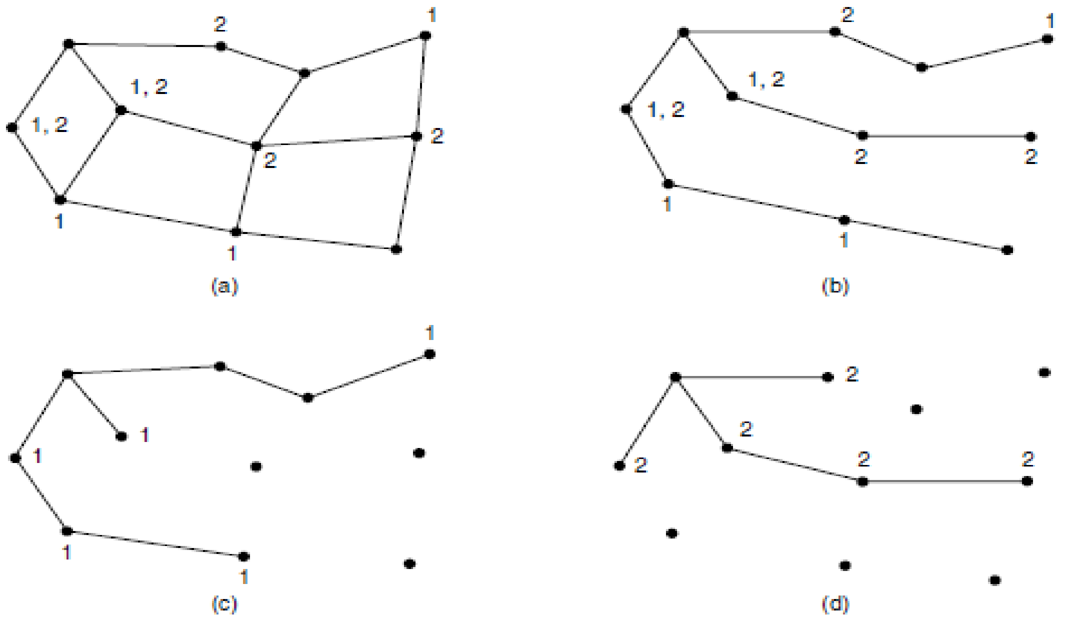


**Figure 5-16.** (a) A network. (b) A spanning tree for the leftmost router. (c) A multicast tree for group 1. (d) A multicast tree for group 2.

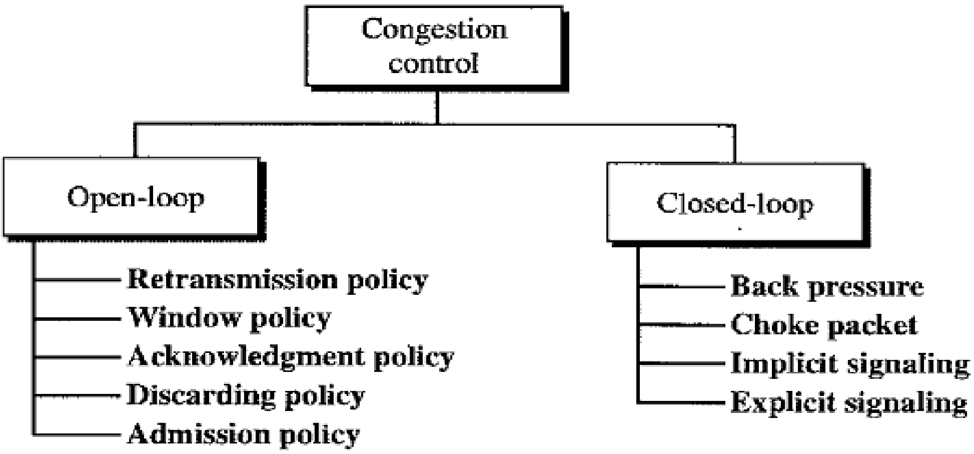
- As an example, consider the two groups, 1 and 2, in the network shown in Fig. 5-16(a). Some routers are attached to hosts that belong to one or both of these groups, as indicated in the figure. A spanning tree for the leftmost router is shown in Fig. 5-16(b).
- This tree can be used for broadcast but is overkill for multicast, as can be seen from the two pruned versions that are shown next. In Fig. 5-16(c), all the links that do not lead to hosts that are members of group 1 have been removed.
- The result is the multicast spanning tree for the leftmost router to send to group 1. Packets are forwarded only along this spanning tree, which is more efficient than the broadcast tree because there are 7 links instead of 10.
- Fig. 5-16(d) shows the multicast spanning tree after pruning for group 2. It is efficient too, with only five links this time. It also shows that different multicast groups have different spanning trees.

## CONGESTION CONTROL

Congestion control refers to techniques and mechanisms that can either prevent congestionbefore it happens or remove congestion after it has happened.

***Congestion control mechanisms can be divided into two categories:***

1. Open-loop congestion control (prevention)
2. Closed-loop congestion control (removal)



## Open-Loop Congestion Control

In open-loop congestion control, policies are applied to prevent congestion before it happens. Here congestion control is handled by either the source or the destination.

### Retransmission Policy

The packet needs to be retransmitted by sender, when a packet is lost or corrupted. Retransmission is sometimes unavoidable. It may increase congestion in the network. The retransmission policy and the retransmission timers must be designed to optimizeefficiency and at the same time prevent congestion.

*Example*: Retransmission policy used by TCP is designed to prevent or alleviate congestion.

### Window Policy

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 is expired several packets will beresent, although some may have arrived safe and sound at the receiver. This duplicationmay make the congestion worse.

- The Selective Repeat window tries to send the specific packets that have been lost or corrupted.

### Acknowledgment Policy

- The acknowledgments are also part of the load in a network. Sending feweracknowledgments means imposing less load on the network.
- If the receiver does not acknowledge every packet it receives, it may slow down thesender and help prevent congestion.
- A receiver may send an acknowledgment only if it has a packet to be sent or a specialtimer expires.

### Discarding Policy

- A good discarding policy by routers may prevent congestion.
- Example: In audio transmission if the policy is to discard less sensitive packets when congestion happens, the quality of sound is still preserved and congestion is prevented.

### Admission Policy

- An admission policy can prevent congestion in virtual-circuit networks.
- Switches first check the resource requirement of a data flow before admitting it to the

network.

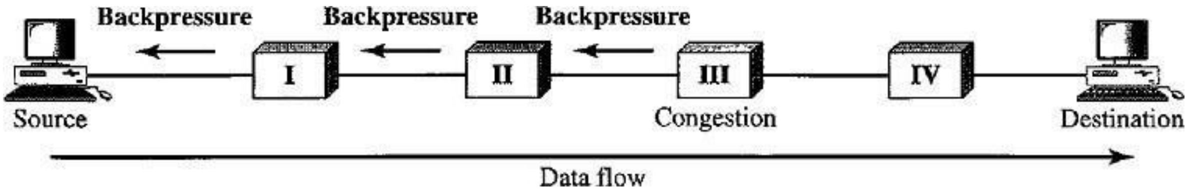
- A router can deny establishing a virtual-circuit connection if there is congestion in the network or if there is a possibility of future congestion.

## Closed-Loop Congestion Control

- Closed-loop congestion control mechanisms try to alleviate congestion after it happens.
- Several mechanisms have been used by different protocols are: Back pressure, Choke packet,Implicit signaling, Explicit signaling.

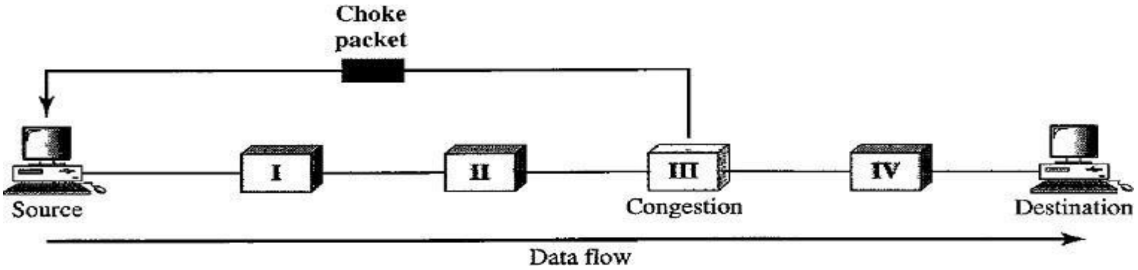
### Backpressure

- In Backpressure mechanism, a congested node stops receiving data from the immediate upstream node.
- This may cause the upstream nodes to become congested and they reject data from their upstream nodes.
- Backpressure is a node-to-node congestion control that starts with a node and propagatesin the opposite direction of data flow to the source.
- The backpressure technique can be applied only to virtual circuit networks, in whicheach node knows the upstream node from which a data flow is coming.



### Choke Packet

- A choke packet is a packet sent by a node to the source to inform that congestion has occurred.
- In the choke packet method, the warning is sent from the router, which has encountered congestion to the source station directly. The intermediate nodes through which the packet has traveled are not warned.



### Implicit Signaling

- In implicit signaling, there is no communication between the congested nodes and thesource. Source guesses that there is congestion somewhere in the network from other symptoms.
- Example: when a source sends several packets and there is no acknowledgment for a while, one assumption is that the network is congested.
- The delay in receiving an acknowledgment is interpreted as congestion in the network and the source should slow down sending speed.

### Explicit Signaling

- The node that experiences congestion can explicitly send a signal to the source ordestination.

- In explicit signaling method, the signal is included in the packets that carry data. Explicit signaling can occur in either the forward or the backward direction.

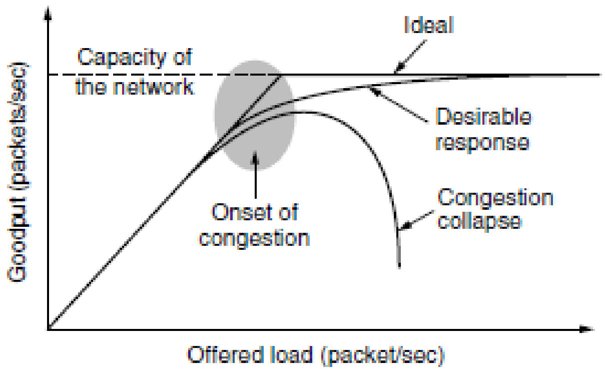


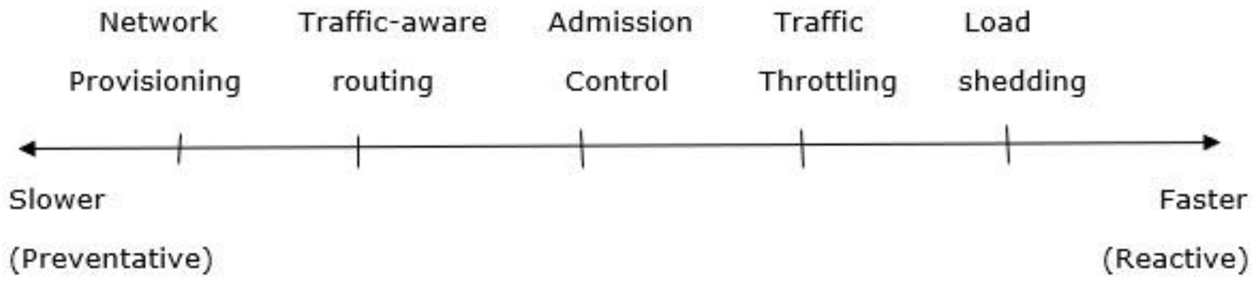
**Figure 5-21.** With too much traffic, performance drops sharply.

- Unless the network is well designed, it may experience a **congestion collapse**, in which performance plummets as the offered load increases beyond the capacity.
- This can happen because packets can be sufficiently delayed inside the network that they are no longer useful when they leave the network.
- For example, in the early Internet, the time a packet spent waiting for a backlog of packets ahead of it to be sent over a slow 56-kbps link could reach the maximum time it was allowed to remain in the network. It then had to be thrown away.
- A different failure mode occurs when senders retransmit packets that are greatly delayed, thinking that they have been lost. In this case, copies of the same packet will be delivered by the network, again wasting its capacity. To capture these factors, the y-axis of Fig. 5-21 is given as **goodput**, which is the rate at which *useful* packets are delivered by the network.

## APPROACHES TO CONGESTION CONTROL

The presence of congestion means the load is greater than the resources available over a network to handle. Generally we will get an idea to reduce the congestion by trying to increase the resources or decrease the load, but it is not that much of a good idea.

There are some approaches for congestion control over a network which are usually applied on different time scales to either prevent congestion or react to it once it has occurred.



Time scale of approaches to congestion control

Let us understand these approaches step wise as mentioned below −

**Step 1** − The basic way to avoid congestion is to build a network that is well matched to the traffic that it carries. If more traffic is directed but a low-bandwidth link is available, definitely congestion occurs.

**Step 2** − Sometimes resources can be added dynamically like routers and links when there is serious congestion. This is called **provisioning**, and which happens on a timescale of months, driven by long-term trends.

**Step 3** − To utilise most existing network capacity, routers can be tailored to traffic patterns making them active during daytime when network users are using more and sleep in different time zones.

**Step 4** − Some of local radio stations have helicopters flying around their cities to report on road congestion to make it possible for their mobile listeners to route their packets (cars) around hotspots. This is called **traffic aware routing.**

**Step 5** − Sometimes it is not possible to increase capacity. The only way to reduce the congestion is to decrease the load. In a virtual circuit network, new connections can be refused if they would cause the network to become congested. This is called **admission control.**

**Step 6** − Routers can monitor the average load, queueing delay, or packet loss. In all these cases, the rising number indicates growing congestion. The network is forced to discard packets that it cannot deliver. The general name for this is **Load shedding.** The better technique for choosing which packets to discard can help to prevent congestion collapse.

## TRAFFIC-AWARE ROUTING

- These schemes adapted to changes in topology, but not to changes in load. The goal in taking load into account when computing routes is to shift traffic away from hotspots that will be the first places in the network to experience congestion.
- Consider the network of Fig. 5-23, which is divided into two parts, East and West, connected by two links, *CF* and *EI*. Suppose that most of the traffic between East and West is using link *CF*, and, as a result, this link is heavily loaded with long delays.
- Including queueing delay in the weight used for the shortest path calculation will make *EI* more attractive. After the new routing tables have been installed, most of the East-West traffic will now go over *EI*, loading this link.
- Consequently, in the next update, *CF* will appear to be the shortest path. As a result, the routing tables may oscillate wildly, leading to erratic routing and many potential problems.

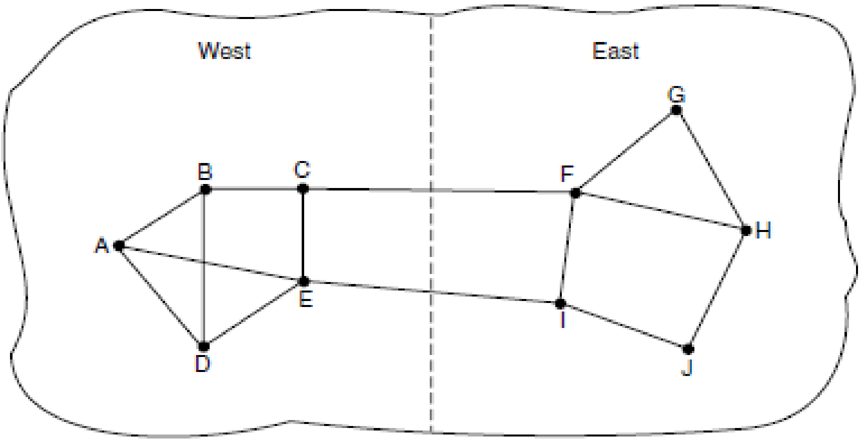


**Figure 5-23.** A network in which the East and West parts are connected by two links.

- If load is ignored and only bandwidth and propagation delay are considered, this problem does not occur. Attempts to include load but change weights within a narrow range only slow down routing oscillations.
- Two techniques can contribute to a successful solution.

  ➔ The first is multipath routing, in which there can be multiple paths from a source to a destination. In our example this means that the traffic can be spread across both of the East to West links.

➔ The second one is for the routing scheme to shift traffic across routes slowly enough that it is able to converge. Adjustments are made outside the routing protocol by slowly changing its inputs which is called **traffic engineering**.

## ADMISSION CONTROL

- One technique that is widely used in virtual-circuit networks to keep congestion at bay is **admission control**. The idea is simple: do not set up a new virtual circuit unless the network can carry the added traffic without becoming congested.
- Thus, attempts to set up a virtual circuit may fail. This is better than the alternative, as letting more people in when the network is busy just makes matters worse. By analogy, in the telephone system, when a switch gets overloaded it practices admission control by not giving dial tones. The trick with this approach is working out when a new virtual circuit will lead to congestion.
- The task is straightforward in the telephone network because of the fixed bandwidth of calls (64 kbps for uncompressed audio). Armed with traffic descriptions, the network can decide whether to admit the new virtual circuit. One possibility is for the network to reserve enough capacity along the paths of each of its virtual circuits that congestion will not occur.
- In this case, the traffic description is a service agreement for what the network will guarantee its users. Even without making guarantees, the network can use traffic descriptions for admission control.
- The task is then to estimate how many circuits will fit within the carrying capacity of the network without congestion. Admission control can also be combined with traffic-aware routing by considering routes around traffic hotspots as part of the setup procedure.

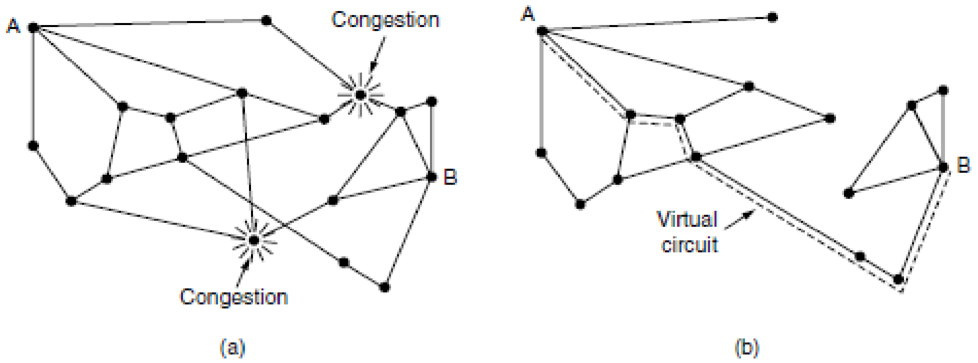


**Figure 5-24.** (a) A congested network. (b) The portion of the network that is not congested. A virtual circuit from *A* to *B* is also shown.

- Suppose that a host attached to router *A* wants to set up a connection to a host attached to router *B*. Normally, this connection would pass through one of the congested routers. To avoid this situation, we can redraw the network as shown in Fig. 5-24(b), omitting the congested routers and all of their lines. The dashed line shows a possible route for the virtual circuit that avoids the congested routers.

## TRAFFIC THROTTLING

- In the Internet and many other computer networks, senders adjust their transmissions to send as much traffic as the network can readily deliver. In this setting, the network aims to operate just before the onset of congestion.

- When congestion is imminent, it must tell the senders to throttle back their transmissions and slow down. This feedback is business as usual rather than an exceptional situation. The term **congestion avoidance** is sometimes used to contrast this operating point with the one in which the network has become (overly) congested.
- Some approaches to throttling traffic that can be used in both datagram networks and virtual-circuit networks. Each approach must solve two problems. First, routers must determine when congestion is approaching, ideally before it has arrived. To do so, each router can continuously monitor the resources it is using.
- Three possibilities are the utilization of the output links, the buffering of queued packets inside the router, and the number of packets that are lost due to insufficient buffering. Averages of utilization do not directly account for the burstiness of most traffic—a utilization of 50% may be low for smooth traffic and too high for highly variable traffic.
- Counts of packet losses come too late. Congestion has already set in by the time that packets are lost. The queueing delay inside routers directly captures any congestion experienced by packets. It should be low most of time, but will jump when there is a burst of traffic that generates a backlog. To maintain a good estimate of the queueing delay, *d*, a sample of the instantaneous queue length, *s*, can be made periodically and *d* updated according to

$$d_{\text{new}} = \alpha d_{\text{old}} + (1 - \alpha)s$$

where the constant α determines how fast the router forgets recent history.

- This is called an **EWMA** (**Exponentially Weighted Moving Average**). It smoothes out fluctuations and is equivalent to a low-pass filter. Whenever *d* moves above the threshold, the router notes the onset of congestion.
- The second problem is that routers must deliver timely feedback to the senders that are causing the congestion. Congestion is experienced in the network, but relieving congestion requires action on behalf of the senders that are using the network.
- To deliver feedback, the router must identify the appropriate senders. It must then warn them carefully, without sending many more packets into the already congested network.

## CHOKE PACKETS

- In this approach, the router selects a congested packet and sends a **choke packet** back to the source host, giving it the destination found in the packet.
- The original packet may be tagged (a header bit is turned on) so that it will not generate any more choke packets farther along the path and then forwarded in the usual way.
- To avoid increasing load on the network during a time of congestion, the router may only send choke packets at a low rate.
- When the source host gets the choke packet, it is required to reduce the traffic sent to the specified destination, for example, by 50%.
- In a datagram network, simply picking packets at random when there is congestion is likely to cause choke packets to be sent to fast senders, because they will have the most packets in the queue.
- The feedback implicit in this protocol can help prevent congestion yet not throttle any sender unless it causes trouble. For the same reason, it is likely that multiple choke packets will be sent to a given host and destination.
- The host should ignore these additional chokes for the fixed time interval until its reduction

in traffic takes effect. After that period, further choke packets indicate that the network is still congested.

- An example of a choke packet used in the early Internet is the SOURCEQUENCH message in which it was generated and the effect it had were not clearly specified. The modern Internet uses an alternative notification design.

## EXPLICIT CONGESTION NOTIFICATION

- When the network delivers the packet, the destination can note that there is congestion and inform the sender when it sends a reply packet. The sender can then throttle its transmissions as before. This design is called **ECN** (**Explicit Congestion Notification**) and is used in the Internet.
- It is a refinement of early congestion signaling protocols, notably the binary feedback scheme of Ramakrishnan and Jain (1988) that was used in the DECNET architecture. Two bits in the IP packet header are used to record whether the packet has experienced congestion.
- Packets are unmarked when they are sent, as illustrated in Fig. 5-25. If any of the routers they pass through is congested, that router will then mark the packet as having experienced congestion as it is forwarded.
- The destination will then echo any marks back to the sender as an explicit congestion signal in its next reply packet. This is shown with a dashed line in the figure to indicate that it happens above the IP level (e.g., in TCP). The sender must then throttle its transmissions, as in the case of choke packets.

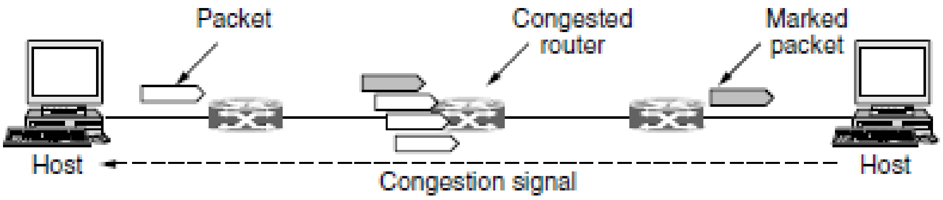


**Figure 5-25.** Explicit congestion notification

## HOP-BY-HOP BACKPRESSURE

- At high speeds or over long distances, many new packets may be transmitted after congestion has been signaled because of the delay before the signal takes effect. An ECN indication will take even longer because it is delivered via the destination.
- Choke packet propagation is illustrated as the second, third, and fourth steps in Fig. 5-26(a).

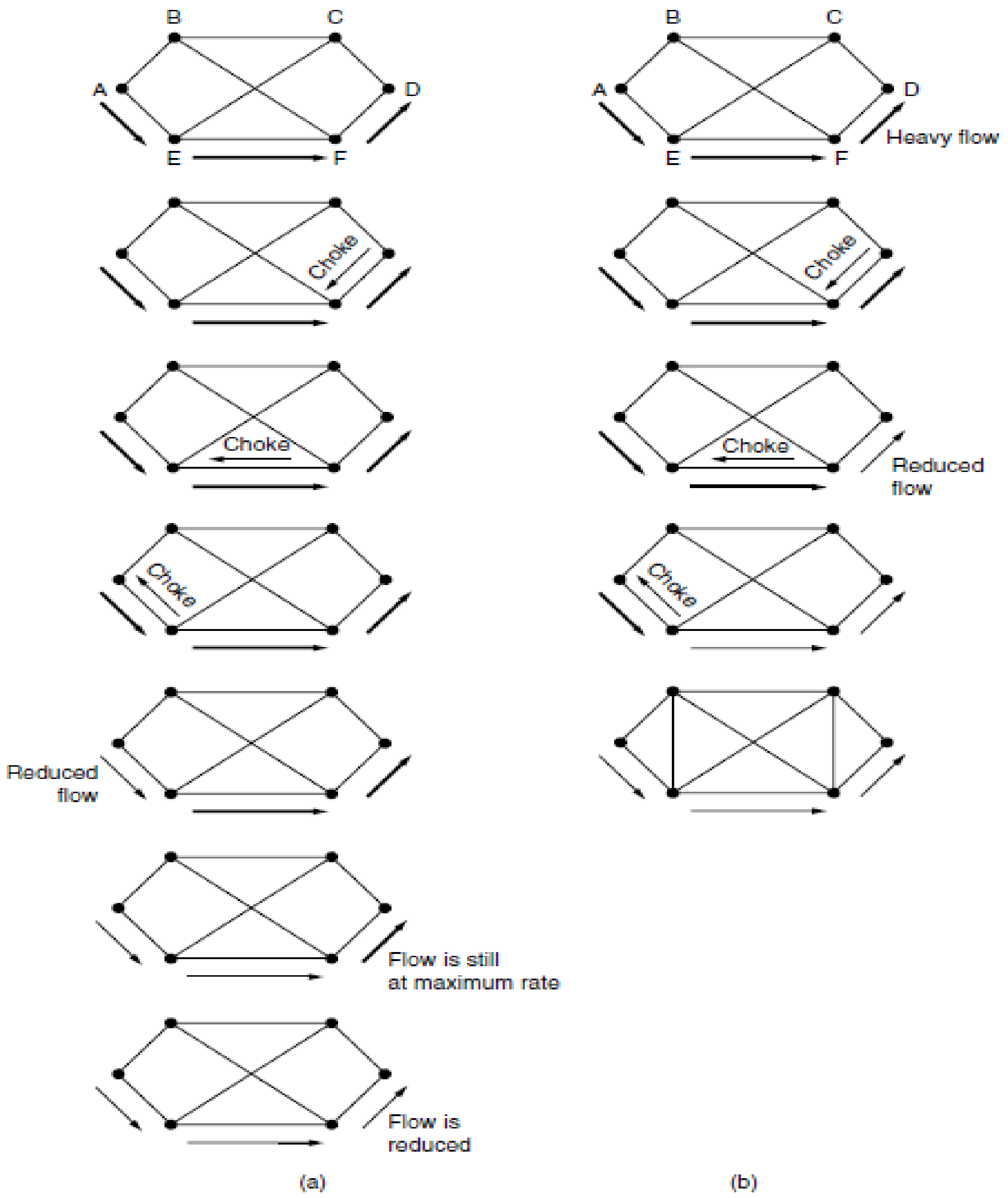


**Figure 5-26.** (a) A choke packet that affects only the source. (b) A choke packet that affects each hop it passes through.

- An alternative approach is to have the choke packet take effect at every hop it passes through, as shown in the sequence of Fig. 5-26(b). Here, as soon as the choke packet reaches *F*, *F* is required to reduce the flow to *D*. Doing so will require *F* to devote more buffers to the connection, since the source is still sending away at full blast, but it gives *D* immediate relief, like a headache remedy in a television commercial. In the next step, the choke packet reaches *E*, which tells *E* to reduce the flow to *F*.
- This action puts a greater demand on *E*'s buffers but gives *F* immediate relief. Finally, the choke packet reaches *A* and the flow genuinely slows down. The net effect of this hop-by-hop scheme is to provide quick relief at the point of congestion, at the price of using up more buffers upstream. In this way, congestion can be nipped in the bud without losing any packets.

## 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. The preferred choice may depend on the type of applications that use the network. For a file transfer, an old packet is worth more than a new one. This is because dropping packet 6 and keeping packets 7 through 10, for example, will only force the receiver to do more work to buffer data that it cannot yet use.
- In contrast, for real-time media, a new packet is worth more than an old one. This is because packets become useless if they are delayed and miss the time at which they must be played out to the user.
- More intelligent load shedding requires cooperation from the senders. An example is packets that carry routing information. These packets are more important than regular data packets because they establish routes; if they are lost, the network may lose connectivity.
- To implement an intelligent discard policy, applications must mark their packets to indicate to the network how important they are. Then, when packets have to be discarded, routers can first drop packets from the least important class, then the next most important class, and so on.

## RANDOM EARLY DETECTION

- To determine when to start discarding, routers maintain a running average of their queue lengths. When the average queue length on some link exceeds a threshold, the link is said to be congested and a small fraction of the packets are dropped at random.
- Picking packets at random makes it more likely that the fastest senders will see a packet drop; this is the best option since the router cannot tell which source is causing the most trouble in a datagram network.
- The affected sender will notice the loss when there is no acknowledgement, and then the transport protocol will slow down. The lost packet is thus delivering the same message as a choke packet, but implicitly, without the router sending any explicit signal.
- RED routers improve performance compared to routers that drop packets only when their buffers are full, though they may require tuning to work well. For example, the ideal number of packets to drop depends on how many senders need to be notified of congestion.
- However, ECN is the preferred option if it is available. It works in exactly the same manner, but delivers a congestion signal explicitly rather than as a loss; RED is used when hosts cannot receive explicit signals.

## QUALITY OF SERVICE

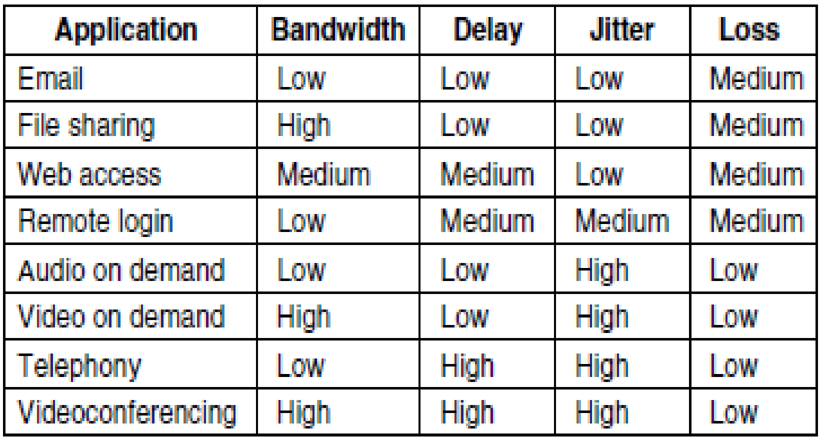
- An easy solution to provide good quality of service is to build a network with enough capacity for whatever traffic will be thrown at it. The name for this solution is **overprovisioning**.
- The resulting network will carry application traffic without significant loss and, assuming a decent routing scheme, will deliver packets with low latency. Quality of service mechanisms let a network with less capacity meet application requirements just as well at a lower cost. Overprovisioning is based on expected traffic.
- All bets are off if the traffic pattern changes too much. With Quality of Service mechanisms, the network can honor the performance guarantees that it makes even when traffic spikes, at the cost of turning down some requests.
- Four issues must be addressed to ensure quality of service:
  1. What applications need from the network.

2. How to regulate the traffic that enters the network.
3. How to reserve resources at routers to guarantee performance.
4. Whether the network can safely accept more traffic.

- No single technique deals efficiently with all these issues. Instead, a variety of techniques have been developed for use at the network (and transport) layer. Practical quality-of-service solutions combine multiple techniques.

## APPLICATION REQUIREMENTS

- A stream of packets from a source to a destination is called a **flow.** A flow might be all the packets of a connection in a connection-oriented network, or all the packets sent from one process to another process in a connectionlessnetwork.
- The needs of each flow can be characterized by four primary parameters: bandwidth, delay, jitter, and loss. Together, these determine the **QoS** (**Quality of Service**) the flow requires.



| Application | Bandwidth | Delay | Jitter | Loss |
|---|---|---|---|---|
| Email | Low | Low | Low | Medium |
| File sharing | High | Low | Low | Medium |
| Web access | Medium | Medium | Low | Medium |
| Remote login | Low | Medium | Medium | Medium |
| Audio on demand | Low | Low | High | Low |
| Video on demand | High | Low | High | Low |
| Telephony | Low | High | High | Low |
| Videoconferencing | High | High | High | Low |

**Figure 5-27.** Stringency of applications' quality-of-service requirements.

- The applications differ in their bandwidth needs, with email, audio in all forms, and remote login not needing much, but file sharing and video in all forms needing a great deal.
- More interesting are the delay requirements. File transfer applications, including email and video, are not delay sensitive. If all packets are delayed uniformly by a few seconds, no harm is done. Interactive applications, such as Web surfing and remote login, are more delay sensitive.
- The variation (i.e., standard deviation) in the delay or packet arrival times is called **jitter**. To accommodate a variety of applications, networks may support different categories of QoS.
- An influential example comes from ATM networks, which were once part of a grand vision for networking but have since become a niche technology.
- They support:

1. Constant bit rate (e.g., telephony).
2. Real-time variable bit rate (e.g., compressed videoconferencing).
3. Non-real-time variable bit rate (e.g., watching a movie on demand).
4. Available bit rate (e.g., file transfer).

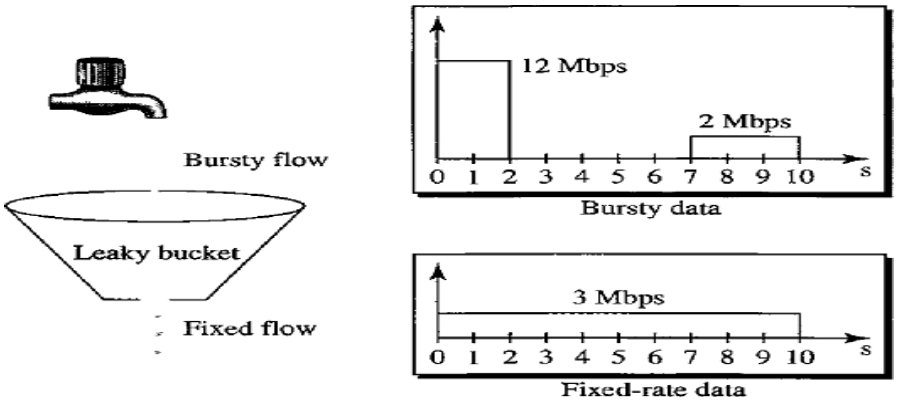
- These categories are also useful for other purposes and other networks. Constant bit rate is an attempt to simulate a wire by providing a uniform bandwidth and a uniform delay. Variable bit rate occurs when video is compressed, with some frames compressing more than others.

# TRAFFIC SHAPING

- Traffic shaping is a mechanism to control the amount of traffic and the rate of the traffic sent to the network.
- Two techniques can shape traffic: **Leaky bucket** and **Token bucket.**

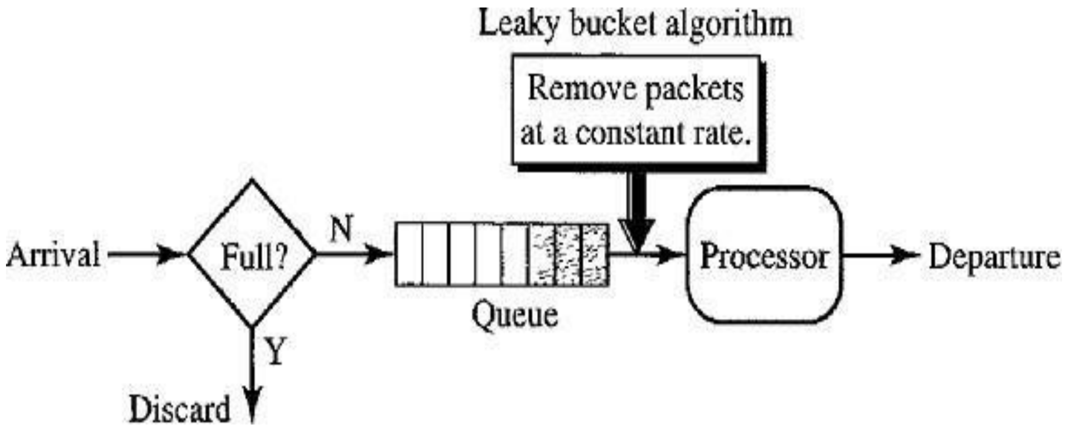
## Leaky Bucket

- If a bucket has a small hole at the bottom, the water leaks from the bucket at a constant rate as long as there is water in the bucket.
- The rate at which the water leaks does not depend on the rate at which the water is input tothe bucket unless the bucket is empty. The input rate can vary, but the output rate remains constant.
- In networking, a technique called leaky bucket can smooth out Bursty traffic. Bursty chunks are stored in thebucket and sent out at an average rate.



- In the above figure, the network has committed a bandwidth of 3 Mbps for a host. The use of the leaky bucketshapes the input traffic to make it conform to this commitment.
- The host sends a burst of data at a rate of 12 Mbps for 2 sec, for a total of 24 Mbits of data.
- The host is silent for 5 sec and then sends data at a rate of 2 Mbps for 3 sec, for a total of 6 Mbits of data.In total the host has sent 30 Mbits of data in l0s.
- The leaky bucket smoothen the traffic by sending out data at a rate of 3 Mbps during the same 10 sec. Without the leaky bucket, the beginning burst may have hurt the network by consuming more bandwidth thanis set aside for this host.
- This way the leaky bucket may prevent congestion.

***Consider the below figure that shows implementation of Leaky Bucket:***



- A FIFO queue holds the packets. If the traffic consists of fixed-size packets the process removes a fixednumber of packets from the queue at each tick of the clock.

- If the traffic consists of variable-length packets, the fixed output rate must be based on the number of bytes orbits.

  The following is an algorithm for variable-length packets:

  - ➔ Initialize a counter to *n* at the tick of the clock.
  - ➔ If *n* is greater than the size of the packet, send the packet and decrement the counter by the packet size. Repeatthis step until *n* is smaller than the packet size.
  - ➔ Reset the counter and go to step 1.

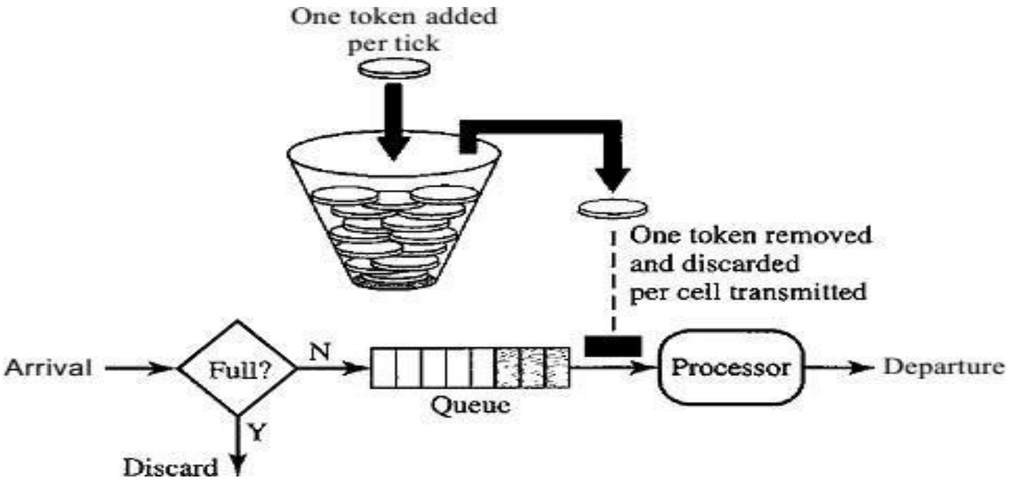
**Problems with Leaky Bucket**

1. The leaky bucket is very restrictive. If a host is not sending for a while, its bucket becomes empty.
2. After some time if the host has bursty data, the leaky bucket allows only an average rate. The time when thehost was idle is not taken into account.

These problems can be overcome by Token bucket algorithm.

## Token Bucket

- The token bucket algorithm allows idle hosts to accumulate credit for the future in the form of tokens.For each tick of the clock, the system sends *n* tokens to the bucket. The system removes one token for every cell (or byte) of data sent.
- Example: If *n* is 100 and the host is idle for 100 ticks, the bucket collects 10,000 tokens.
- Now the host can consume all these tokens in one tick with 10,000 cells, or the host take 1000 ticks with 10cells per tick.
- The host can send bursty data as long as the bucket is not empty. The token bucket can easily be implemented with a counter.
- The token is initialized to zero. Each time a token is added, the counter is incremented by 1. Each time a unit of data is sent, the counter is decremented by 1.When the counter is zero, the host cannot send data.



# THE NETWORK LAYER IN THE INTERNET

Communication at the network layers is host-to-host (Computer-to-Computer).

## LOGICAL ADDRESSING

Logical addressing is implemented by network layer. Logical addressing is a Global Addressing scheme.

Data-link layer handles the addressing problem locally, but if packets passes the network boundary there is a need for logical addressing system to help distinguish source and destination systems.

The network layer adds a header to the packet coming from the upper layer that includes the logical addresses of the sender and receiver.

There are 2 types of addressing mechanisms are present:

1. IPv4 ( IP version4)
2. IPv6 (IP version 6)

# IP ADDRESSES

## IPv4 ADDRESSES

An **IPv4** address is a **32-bit** address that **uniquely** and **universally** defines the connection ofa device to the Internet.

- **Unique:** Two devices on the Internet can never have the same address at the same time.
- **Universal:** The addressing system must be accepted by any host that wants to be connectedto the Internet.

## Address Space

- An address space is the total number of addresses used by the protocol. If a protocol uses **N** bits to define an address, the address space is $2^N$ because each bit canhave two different values (0 or 1) and N bits can have $2^N$ values. IPv4 uses **32**-bit addresses, which means that the address space is $2^{32}$ or **4,294,967,296** (more than **4 billion**).

### Notations

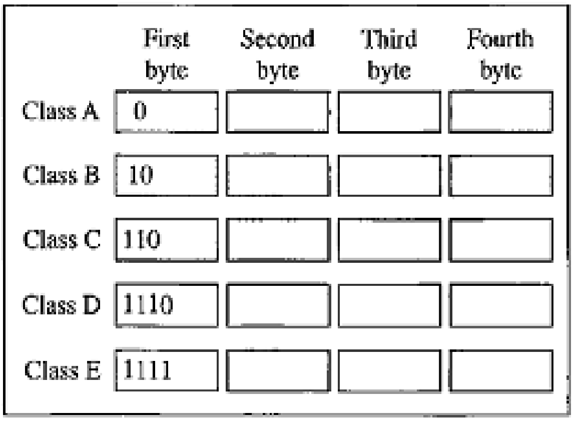
There are two notations to show an IPv4 address: Binary and Dotted-Decimal Notation.

| Binary | Dotted-Decimal |
|---|---|
| • IPv4 address is displayed as 32 bits. Each octet is often referred to as a byte.<br>• It is a 4 byte address<br>Ex: 10000000 00001011 00000011 00011111 | • Internet addresses are written in decimal form with a dot separating the bytes.<br>• Each number in dotted-decimal notation is a value ranging from 0 to 255.<br>Ex: 128.11.3.31 |

## CLASSFUL ADDRESSING

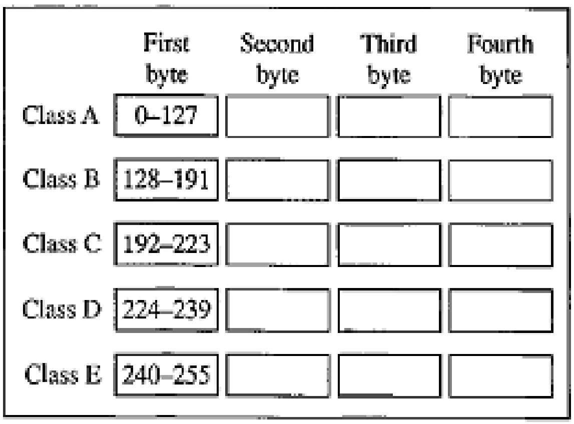
- Initially IPv4 used the concept of Classful addressing.
- In classful addressing, the address space is divided into five classes: A, B, C, D, and E.

- If the address is given in binary notation, the first few bits can immediately tell us the class of the address.
- If the address is given in decimal-dotted notation, the first byte defines the class.



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

a. Binary notation



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

b. Dotted-decimal notation

**Classes and Blocks**

- Each class is divided into a fixed number of blocks.
- Size of the each block is also fixed.

| Class | No of Blocks ($2^n$) | Block Size | Application | % of $\mathbf{2^{32}}$ IP addresses |
|---|---|---|---|---|
| A | 128 **($2^7$)** | 16,777,216 **($2^{21}$)** | Unicast | 50% |
| B | 16,384 **($2^{14}$)** | 65,536 **($2^{16}$)** | Unicast | 25% |
| C | 2,097,152 **($2^{21}$)** | 256 **($2^{24}$)** | Unicast | 12.5% |
| D | 1 | 268,435,456 | Multicast | 6.25% |
| E | 1 | 268,435,456 | Reserved | 6.25% |

**Purpose of classes:**

**Class A** addresses were designed for large organizations with a large number of attached hosts or routers.

**Class B** addresses were designed for midsize organizations with tens of thousands of attached hosts or routers.

**Class C** addresses were designed for small organizations with a small number of attachedhosts or routers.

**Class D** addresses were designed for multicasting.

**Class E** addresses were reserved for future use.

**Problem with above classes and block sizes: A lot of addresses are wasted.**

1. A block in **class A** address is too large for almost any organization. (i.e) most of the addresses in class A were wasted and were not used.
2. A block in **class B** is also very large, probably too large for many of the organizations that received a class B block.
3. A block in **class C** is probably too small for many organizations.
4. **Class D** addresses were designed for multicasting, each address in class D is used to define one group of hosts on the Internet. The Internet authorities wrongly predicted a need for

268,435,456 groups.

5. **Class E** addresses were reserved for future use; only a few addresses were used till now.

**Netid and Hostid**

➔ In classful addressing, an IP address in class A, B, or C is divided into Netid and Hostid.

❖ For Class D, E there were no Netid and Hostid.

❖ In class A, first one byte defines the Netid and three bytes define the Hostid.

❖ In class B, first two bytes define the Netid and two bytes define the Hostid.

❖ In class C, first three bytes define the Netid and one byte defines the Hostid.

**The First and Last address of Each class:**

| Class | First Address | Last Address | Example |
|---|---|---|---|
| A | 0.0.0.0 | 127.255.255.255 | **64.**65.43.21 |
| B | 128.0.0.0 | 191.255.255.255 | **175.112.**1.3 |
| C | 192.0.0.0 | 223.255.255.255 | **202.3.4.**5 |
| D | 224.0.0.0 | 239.255.255.255 | 230.20.45.61 |
| E | 240.0.0.0 | 255.255.255.255 | 252.251.250.21 |

| **Class** | **Range** | **Total IP Addresses** | **Total NID** | **No.of Hosts per N/W** | **Default Subnet Mask** | **DBA** |
|---|---|---|---|---|---|---|
| A | **0-127**<br>**(1-126)** | $2^{31}$ | $2^{7}$ | $2^{24}-2$ | **255.0.0.0** | **nnn.255.255.255** |
| B | **128-191** | $2^{30}$ | $2^{14}$ | $2^{16}-2$ | **255.255.0.0** | **nnn.nnn.255.255** |
| C | **192-223** | $2^{29}$ | $2^{21}$ | $2^{8}-2$ | **255.255.255.0** | **nnn.nnn.nnn.255** |
| D | **224-239** | $2^{28}$ | - | - | - | - |
| E | **240-255** | $2^{28}$ | - | - | - | - |

**1.** In Class A, Networks **0** and **127** are reserved.

**2.** The first address (i.e Network Address) and Last Address (i.e. Directed Broadcast Address) will not be assign to any host. That why the total number of host per each Network is $\mathbf{2^{n}-2}$ where n is no of Host ID bits.

**Mask or Default Mask**

□ A default mask is a 32-bit number made of contiguous 1’s followed by contiguous 0’s.

□ The mask can help us to find the Netid and the Hostid.

□ For example, the mask for a class A address has eight 1s, which means the first 8 bits ofany address in class A define the Netid; the next 24 bits define the Hostid.

The masks for classes A, B, and C are:

| Class | Binary | Dotted-Decimal | CIDR |
|---|---|---|---|
| A | **11111111** 00000000 00000000 00000000 | **255**.0.0.0 | **/8** |
| B | **11111111 11111111** 00000000 00000000 | **255.255**.0.0 | **/16** |
| C | **11111111 11111111 11111111** 00000000 | **255.255.255**.0 | **/24** |

## Special Addresses

- **Limited Broadcast Address (255.255.255.255)**

  This address in the destination field defines that, the packet is broadcasted with in the network. That means the hosts which are connected to the same network are allowed to access the packet. Other packet will not transmit to the Other network.
  Eg: Let **20.1.2.3** is the host connected to NID=20 and it wants to broadcast a packet to all the hosts that are connected to **NID=20** network then it uses **255.255.255.255** as the destination address.
- **Directed Broadcast Address (nnn.255.255.255)**

  This address in the destination field defines that, the packetis transmitted outside of the specified network and broadcast the packet to all the hosts connected that destination network.
  Eg: Let 20.1.2.3 is the host that wants send a packet to all the hosts in **NID=60**, then it fills **60.255.255.255** destination field, then the packet is send to a network contains**NID=60** and all the systems that are connected to it will receive the packet.

**Examples for Determining Classful IP addresses.**

| **IP address** | **CLASS** |
|---|---|
| 1.2.3.4 | A |
| 10.20.30.40 | A |
| 140.2.7.8 | B |
| 175.165.47.98 | B |
| 200.20.50.64 | C |
| 230.5.6.8 | D |
| 251.0.1.5 | E |
| 300.0.5.6 | Invalid |

**Subnetting**

- Subnetting is a process of dividing a large block into smaller contiguous groups and assigns each group to smaller networks (subnets) or share a part of the addresses with neighbors.
- Subnetting increases the number of 1's in the mask.

**Supernetting**

- In supernetting, an organization can combine several blocks to create a larger range of addresses. Supernetting decreases the number of 1's in the mask.

Example: an organization that needs 1000 addresses can be granted four contiguous class C blocks. The organization can then use these addresses to create one supernetwork.

**Address Depletion**

- The number of available IPv4 addresses is decreasing as the number of internet users are increasing.
- We have run out of class A and B addresses, and a class C block is too small for most midsize organizations.
- One solution that has alleviated the problem is the idea of Classless Addressing.

## CLASSLESS ADDRESSING

### Purpose

- Classless addressing was designed and implemented to overcome address depletion andgive more organizations access to the Internet.
- In this scheme, there are no classes, but the addresses are still granted in blocks.

### Address Blocks

- In classless addressing, when a small or large entity, needs to be connected to the Internet,it is granted a block of addresses.
- The size of the block (the number of addresses) varies based on the nature and size of theentity.

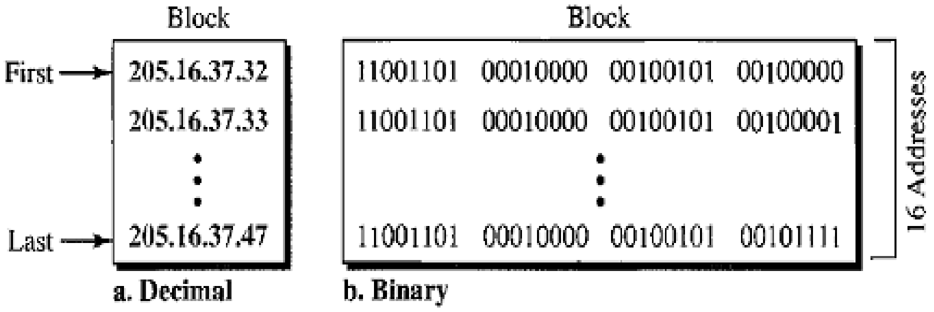
Example:

- For a large organization may be given thousands of addresses
- For a house two addresses are sufficient
- An Internet service provider may be given hundreds of thousands based on the number ofcustomers it may serve.

### Restrictions on classless address blocks

1. The addresses in a block must be contiguous, one after another.
2. The number of addresses in a block must be a power of 2 (1, 2, 4, 8, ... ).
3. The first address must be evenly divisible by the number of addresses.

Consider the below figure for classless addressing that shows a block of addresses, in both binary anddotted-decimal notation, granted to a small business that needs 16 addresses.



205.16.37.32
205.16.37.33
205.16.37.47

11001101 00010000 00100101 00100000
11001101 00010000 00100101 00100001
11001101 00010000 00100101 00101111

a. Decimal b. Binary

It satisfies all 3 restrictions:

- The addresses are contiguous.
- The number of addresses is a power of 2 (16 = $\mathbf{2^4}$).
- The first address is divisible by 16. The first address, when converted to a decimalnumber, is 3,440,387,360, which when divided by 16 results in 215,024,210.

### Mask

A mask is a 32-bit number in which the ***n*** leftmost bits are 1's and the **32 - *n*** rightmost bits are 0's, where ***n= 0 to 32.***

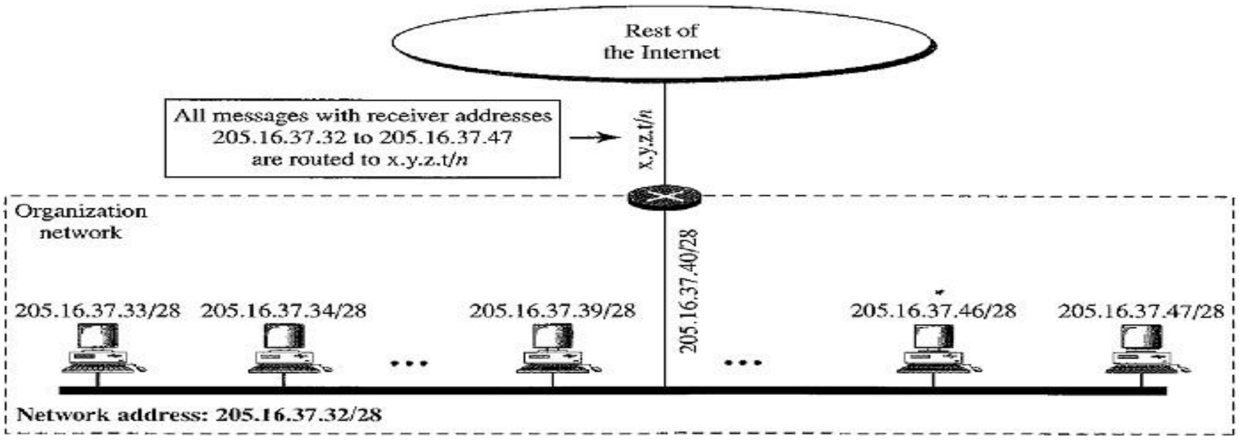
In 1Pv4 addressing, a block of addresses can be defined as **x.y.z.t/n** in which **x.y.z.t** definesone of the addresses and the ***/n*** defines the mask. ***/n*** is called as CIDR notation.

- **First Address** in the block can be found by setting the rightmost **32 - *n*** bits to 0's.
- **Last Address** in the block can be found by setting the rightmost **32 - *n*** bits to 1's.
- **Number of Addresses** in the block can be found by using the formula $\boldsymbol{2^{32-n}}$.

- **Network Address** is the first address in the block and defines the organization network. Usually the first address is used by routers to direct the message sent to the organization from the outside.

Example: **205.16.37.39/28** or **11001101 00010000 00100101 00100111**

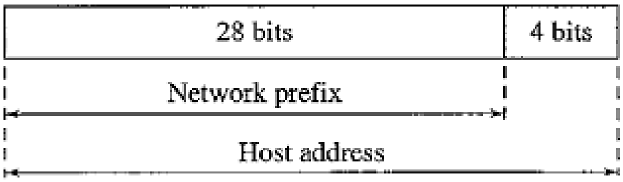
- First address: 11001101 000100000100101 0010000 or 205.16.37.**32**
- Last address: 11001101 00010000 001001010010 1111 or 205.16.37.**47**
- Number of Addresses: $\mathbf{2^{32-28} = 2^{4} = 16.}$
- Network Address (First Address) 11001101 000100000100101 0010000 or 205.16.37.32



**Netid and Hostid**

- The ***n*** leftmost bits of the address **x.y.z.t/n** define the **network address** or **prefix**.
- The **(32 – *n*)** rightmost bits define the particular **suffix** or **host address** (computer orrouter) connected to the network.

**205.16.37.39/28** or **11001101 00010000 00100101 0010 0111**



## INTERNETWORKING

There are many different networks exists such as LANs, MANs, and WANs, PANs. Two or more such networks are connected to form an **Internetwork** also called as an **Internet.**

- Internetworking combines the smaller networks into the larger networks.
- Large networks are much more valuable than small networks because they allow many more connections, so there always will be an incentive to combine smaller networks.
- The purpose of joining all these networks is to allow users on any one of the network to communicate with users on all the other networks.

**Problems With Datalink Layer**

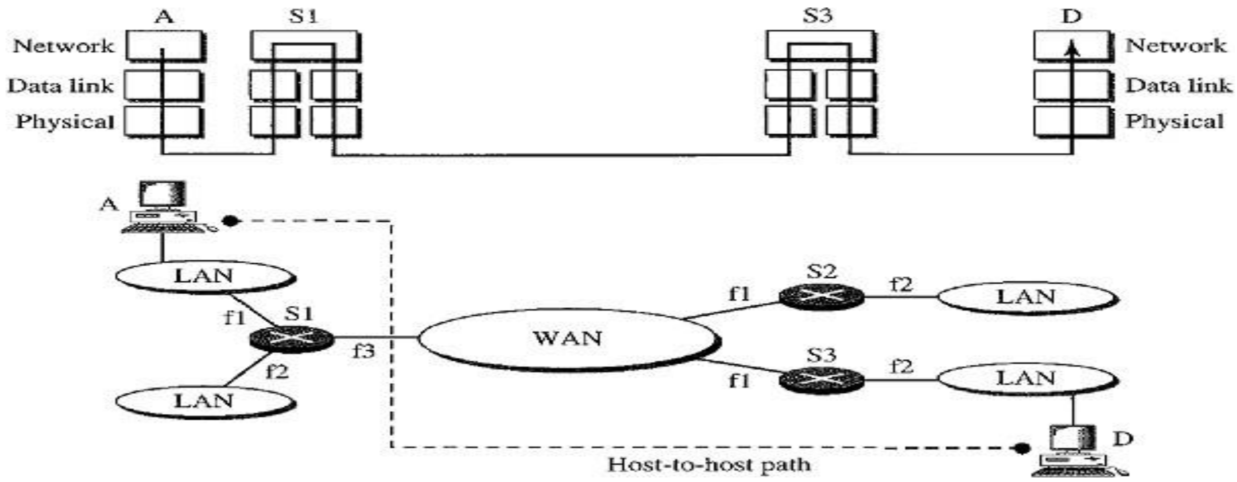
- The frame in datalink layer does not carry any routing information.
- The frame contains the MAC address of source and the MAC address of destination.
- For a LAN or a WAN, delivery means carrying the frame through one link only.

**Need for Network Layer**

- To solve the problem of delivery through several links, the network was designed.
- The network layer is responsible for host-to-host delivery and for routing the packets through the routers or switches.

**Network Layer at Source**

- The network layer at the source is responsible for creating a packet from the data coming from another protocol such as a transport layer protocol or a routing protocol.
  - The header of the packet contains the logical addresses of the source and destination.
  - The network layer is responsible for checking its routing table to find the routing information such as the outgoing interface of the packet or the physical address of the next node.
  - If the packet is too large then the packet is fragmented.



**Network Layer at Router**

- The network layer at the switch or router is responsible for routing the packet.
- When a packet arrives, the router or switch consults its routing table and finds theinterface from which the packet must be sent.
- After some changes made in the packet Header, the routing information is passed to the Data-link layer again.

**Network Layer at Destination**

- The network layer at the destination is responsible for address verification; it makes sure that the destination address on the packet is the same as the address of the host.If the packet is a fragment, the network layer waits until all fragments have arrived, and then reassembles them and delivers the reassembled packet to the transport layer.

**<u>Issues with Internetworking</u>**

When packets sent by a source on one network must transit one or more foreign networks before reaching the destination network, many problems can occur at the interfaces between networks.

- **Delays:** The source needs to be able to address the destination, the packets would cross from a connectionless network to a connection-oriented network. This may require that a new connection be set up on short notice, which injects a delay, and much overhead if the connection is not used for many more packets.
- **Maximum Transfer Unit Size:** The differing max packet sizes used by different networks can be a major nuisance. How do you pass an 8000-byte packet through a network whose maximum size is 1500 bytes?
- **Quality of Service:** If one network has strong QoS and the other offers best effort service, it will be impossible to make bandwidth and delay guarantees for real-time

trafficend to end.

□ **Security**: security mechanisms are problematic in internetworking because packet has to travel through a lot of intermediate network before reaching to destination.

□ **Accounting:** Differences in accounting can lead to unwelcome bills when normal usage suddenly becomes expensive, as roaming mobile phone users with data plans have discovered.

## THE NETWORK LAYER IN THE INTERNET

❖ In the network layer, the Internet can be viewed as a collection of networks or **Autonomous Systems** that are interconnected.

❖ Internet Protocol (IP) is the network layer protocol that was designed for internetworking. IP provide a best-effort (i.e., not guaranteed) way to transport packets from source to destination, without regard to whether these machines are on the same network or whether there are other networks in between them.

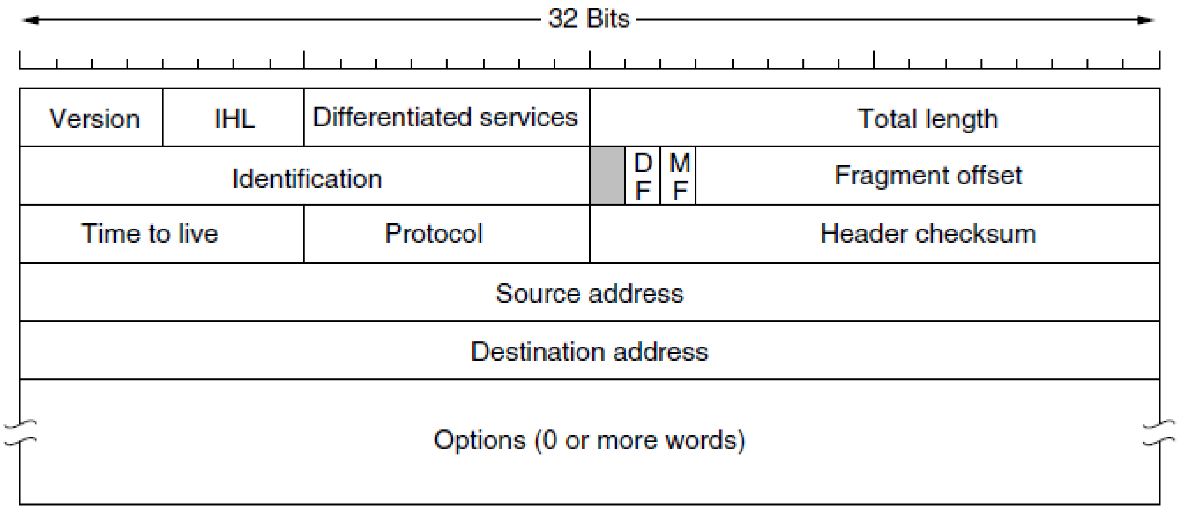
❖ At present there are two versions of IP are used:

1. IPv4
2. IPv6

## IP Version 4 Protocol (IPv4)

❖ 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 bits are transmitted from left to right and top to bottom, with the high-order bit of theVersion field going first.



### Fields of IPv4:

**1. Version (VER) – 4 bits**

□ It defines the version of the IPv4 protocol. Currently 4th version of IPv4 is using.

**2. Header length (IHL) – 4 bits**

□ It defines the total length of the datagram header in 4-byte words.

□ The length of the header is variable between 20 and 60 bytes.

□ When there are no options, the header length is 20 bytes, and the value of this field is 5(5x 4 = 20).

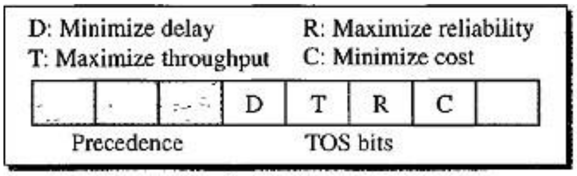
- When the option field is at its maximum size, the value of this field is 15 (15 x 4 = 60).

### 3. Services

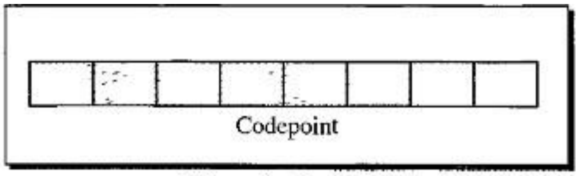
IETF has changed the interpretation and name of this 8-bit field.

Previously this field is called **Service Type** but now the name changed to **DifferentiatedServices**.

**Service Type**



Service type



Differentiated services

In this interpretation, the first 3 bits are called precedence bits. The next 4 bits are called typeof service (TOS) bits, and the last bit is not used.

#### i. Precedence

- It is a 3-bit subfield ranging from 0 to 7 (000 to 111).
- The precedence defines the priority of the datagram in issues such as congestion.
- If a router is congested and needs to discard some datagrams, those datagrams withlowest precedence are discarded first.
- Some datagrams in the Internet are more important than others.

**For example** a datagram used for network management is much more urgent and important than a datagram containing optional information for a group.

#### ii. Type of Service (TOS)

It is a 4-bit subfield with each bit having a special meaning. Out of 4 bits only one bit will have the value of 1.

| TOS Bits | Description |
|---|---|
| 0000 | Normal (default) |
| 0001 | Minimize Cost |
| 0010 | Maximize Reliability |
| 0100 | Maximize Throughput |
| 1000 | Minimize Delay |

#### iii. Differentiated Services

In this interpretation, the first 6 bits make up the code-point subfield, and the last 2 bits arenot used. The code-point subfield can be used in two different ways:

a. When the 3 rightmost bits are 0’s, the 3 leftmost bits are interpreted the same as the precedence bits in the service type interpretation.

b. When the 3 rightmost bits are not all Os, the 6 bits define 64 services based on the priority assignment by the Internet or local authorities.

c. The bottom 2 bits are used to carry explicit congestion notification information, such as whether the packet has experienced congestion.

| Category | Code-point | Assigning Authority | No of service | Numbers |
|---|---|---|---|---|
| 1 | XXXXX0 | Internet | 32 | 0,2,4,6,8,……60,62 |
| 2 | XXXX11 | Local | 16 | 3,7,11,15,……59,63 |
| 3 | XXXX01 | Temporary or Experiment | 16 | 1,5,9,13,17...……,61 |

**Total length**

This is a 16-bit field that defines the total length (**header plus data**) of the IPv4 datagram inbytes. Total length of IPv4 is 65,535 ($2^{16}$-1).

Length of data = Total length - headerlength

**Identification (16 bits)**

- It identifies a datagram originating from the source host.
- The combination of the identification and source IPv4 address must **uniquely** define a datagram as it leaves the source host.
- When a datagram is fragmented, all fragments have the same identification number the same as the original datagram. All fragments having the same identification value must beassembled into one datagram.
- The identification number helps the destination in reassembling the datagram.

**Flags (3 bits)**

The first bit is **Reserved**.

The second bit is called the **Do Not Fragment** bit.

- If its value is 1, the machine (Router) must not fragment the datagram.
- If its value is 0, the datagram can be fragmented if necessary.
- By marking the datagram with the DF bit, the sender knows it will either arrive in onepiece, or an error message will be returned to the sender.

The third bit is called the **More Fragment** bit.

- If its value is 1, it means the datagram is not the last fragment.
- If its value is 0, it means this is the last or only fragment.

**Fragmentation offset (13 bits)**

- It shows the relative position of this fragment with respect to the whole datagram.
- It is the offset of the data in the original datagram measured in units of 8 bytes.

**Time To Live - TTL (8 bits)**

A datagram has a limited lifetime in its travel through an internet.This field can be used in two ways:

i. This field was originally designed to hold a timestamp, which was decremented by each visited router. The datagram was discarded when the value became zero.
ii. This field is used mostly to control the maximum number of hops (routers) visited by the datagram. Each router that processes the datagram decrements this number by 1. The router discards the

datagram, if **TTL=0**.

iii. When a source host sends the datagram, it stores a number in TTL field. This value is approximately 2 times the maximum number of routes between any two hosts.

**Protocol (8 bits)**

- This field defines the higher-level protocol that uses the services of the IPv4 layer.
- An IPv4 datagram can encapsulate data from several higher-level protocols such as TCP, UDP, ICMP, and IGMP.
- This field specifies the final destination protocol to which the IPv4 datagram is delivered.

**Checksum (16 bits)**

The checksum in the IPv4 packet covers only the header, not the data. There are two reasons:

i. All higher-level protocols that encapsulate data in the IPv4 datagram have a checksum field that covers the whole packet. The checksum for the IPv4 datagram does not have to check the encapsulated data.

ii. The header of the IPv4 packet changes with each visited router, but the data do not changes. So the checksum includes only the part that has changed.

**Options**

Options are not required for a datagram. They can be used for network testing and debugging. The Options field is padded out to a multiple of 4 bytes.

- The Security option tells how secret the information is. For Example a military router might use this field to specify not to route packets through certain countries the military considers to be bad.
- The Strict source routing option gives the complete path from source to destination as a sequence of IP addresses. The datagram is required to follow that exact route. It is most useful for system managers who need to send emergency packets when the routing tables have been corrupted, or for making timing measurements.
- The Loose source routing option requires the packet to traverse the list of routers specified, in the order specified, but it is allowed to pass through other routers on the way.
- The Record route option tells each router along the path to append its IP address to the Options field. This allows system managers to track down bugs in the routing algorithms.
- The Timestamp option is like the Record route option, except that in addition to recording its 32-bit IP address, each router also records a 32-bit timestamp. This option is mostly useful for network measurement.

**Source Address (32 bits) & Destination Address (32 bits)**

- These two fields define the IPv4 address of the Source and Destination respectively.
- These fields must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.

**Disadvantages of IPv4**

1. Despite all short-term solutions, such as subnetting, classless addressing, and NAT, address depletion is still a long-term problem in the Internet.
2. The Internet must accommodate real-time audio and video transmission. This type of transmission requires minimum delay strategies and reservation of resources not provided in the IPv4 design.
3. The Internet must accommodate encryption and authentication of data for some applications. No

encryption or authentication is provided by IPv4.

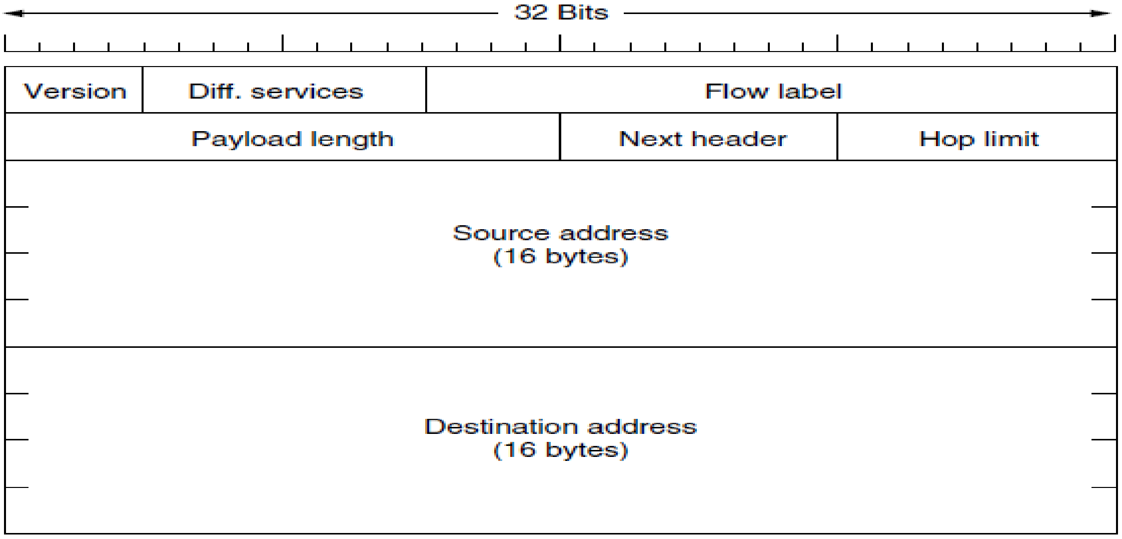
## IP Version 6 (IPv6)

- IPV6 is introduced to overcome the deficiencies of IPv4.
- IPv6 is also called as IPng (Internetworking Protocol next generation).
- In IPv6, the Internet protocol was extensively modified to accommodate the growth of theInternet. Packet format, Length of IP address, ICMP, IGMP, ARP, RARP, RIP routing protocol are also modified in IPv6.

## Advantages of IPv6

- **Larger address space** An IPv6 address is 128 bits long whereas IPv4 is 32-bit address.
- **Better header format** IPv6 uses a new header format in which options are separated from the base header. when options are needed it is inserted between the base header and the upper-layer data.
- **New options** IPv6 has new options to allow for additional functionalities.
- **Allowance for extension** IPv6 is designed to allow the extension of the protocol if required by new technologies or applications.
- **Resource Allocation** In IPv6 the type-of-service field has been removed, but a mechanism called **flow label** has been added to enable the source to request special handling of the packet. This mechanism can be used to support traffic such as real-time audio and video.
- **More Security** The encryption and authentication options in IPv6 provide confidentiality and integrity of the packet.

## IPv6 Header format



**Version (4-bit)**

- This field defines the version number of the IP. For IPv6, the value is 6. Routers will beable to examine this field to tell what kind of packet they have.

**Differentiated services**

- The Differentiated services field (originally called Traffic class) is used to distinguish the class of service for packets with different real-time delivery requirements.
- It is used with the differentiated service architecture for quality of service in the samemanner as the field of the same name in the IPv4 packet.
- The low-order 2 bits are used to signal explicit congestion indications, again in the sameway as with IPv4.

**Flow Label**

- A sequence of packets, sent from a particular source to destination that needs special handling by routers is called a F**low of packets.**
- The combination of the source address and the value of the **Flow Label** uniquely define aflow of packets. To a router, a flow is a sequence of packets that share the same characteristics such as traveling the same path, using the same resources, having the same kind of security etc. A router that supports the handling of flow labels has a flow label table. The table has anentry for each active flow label. Each entry defines the services required by the corresponding flow label.
- When a router receives a packet it consults the flow label table instead of consulting the routing table and going through a routing algorithm to define the address of the next hop, it can easily look in a flow label table for the next hop.
- This mechanism speed up the processing of a packet by a router.

**Payload length (16 bit or 2 Byte)**

- Payload length field defines the length of the IP datagram excluding the base header(40Bytes) (i.e) the 40 header bytes are no longer counted as the total length.
- That means, we can transfer upto 65,535 bytes of payload data plus 40 bytes of Header.

**Next header (8-bit)**

- The next header is an 8-bit field defining the header that follows the base header in the datagram.
- The next header is either optional extension headers used by IP or the header of TCP orUDP encapsulated packet.
- That is, if this header is the last IP header, the Next header field tells which transport protocol handler (e.g., TCP, UDP) to pass the packet to.

**Hop limit (8 bit)**

- Hop limit field serves the same purpose as the TTL field in IPv4. The Hop limit field is usedto keep packets from living forever. The value of this field that is decremented on each hop.

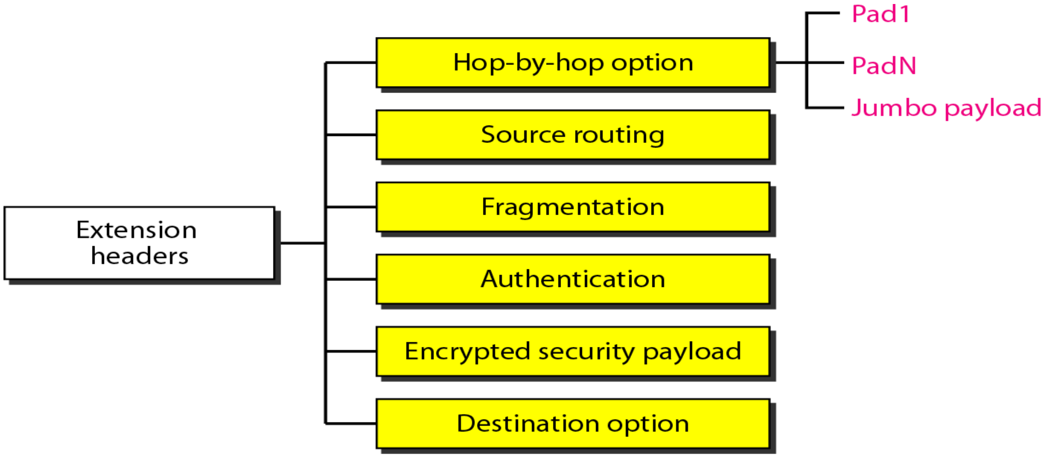
**Source address (128-bit or 16 Byte) and Destination Address (128 bit or 16 Byte)**

- The source address field is a 16-byte (128-bit) Internet address that identifies the original source of the datagram.
- The destination address field is a 16-byte (128-bit) Internet address that usually identifies the final destination of the datagram. If source routing is used, this field contains the address of the next

router.

## Extension Headers

- To give greater functionality to the IP datagram, the base header can be followed by up to six types of extension headers. These headers are used to provide extra information. All the extension Headers are optional.
- Some of the headers have a fixed format; others contain a variable number of variable-lengthoptions. For these, each item is encoded as a (**Type, Length, Value**) tuple.
  - Type is a 1-byte field telling which option this is.
  - Length is also a 1 byte field. It tells how long the value is (0 to 255 bytes).
  - Value is any information required, up to 255 bytes.

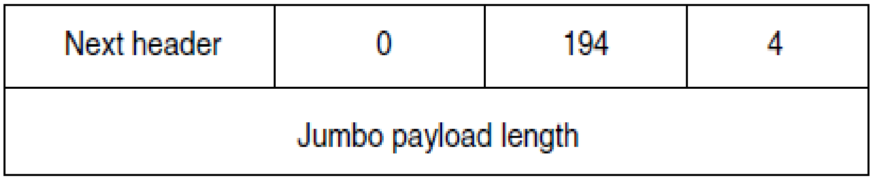


### Hop-by-Hop Option

The hop-by-hop option is used when the source needs to pass information to all routersvisited by the datagram.

Only three options have been defined: Padl, PadN, and jumbo payload.

- The Padl option is 1 byte long and is designed for 1 byte alignment purposes.
- PadN is used when 2 or more bytes is needed for alignment.
- The jumbo payload (Jumbograms) option is used to define a payload longer than 65,535 bytes.



| | | | |
|---|---|---|---|
| Next header | 0 | 194 | 4 |
| Jumbo payload length | | | |

- The first byte of Extension header tells what kind of header comes next.
- The second byte tells how long the hop-by-hop header is in bytes, excluding the first 8 bytes, which are mandatory.
- The next 2 bytes indicate that this option defines the datagram size (code 194) and that the size is a 4-byte number.
- The last 4 bytes give the size of the datagram. Sizes less than 65,536 bytes are not permitted and will result in the first router discarding the packet and sending back an ICMP error message.

Datagrams using this header extension are called **Jumbograms**.

- The use of jumbograms is important for supercomputer applications that must transfer gigabytes of data efficiently across the Internet.

**Fragmentation**

- In IPv4, the source or a router is required to fragment if the size of the datagram is largerthan the MTU of the network over which the datagram travels.
- In IPv6, only the original source can fragment. A source must use a path MTU discoverytechnique to find the smallest MTU supported by any network on the path.
- The source then fragments using this knowledge.

**Source Routing**

- Extension header combines the concepts of the strict source route and theloose source route options of IPv4.

**Authentication**

- Extension header has a dual purpose: it validates the message sender and ensures the integrity of data.
- **Encrypted Security Payload (ESP)** an extension that provides confidentiality and guards against eavesdropping. That means, encrypted security payload encrypts the contents of a packet so that only the intended recipient can read it. These headers use the cryptographic techniques.
- **Destination Option** is used when the source needs to pass information to the destination only. Intermediate routers are not permitted access to this information.

## PACKET FRAGMENTATION

Each network or link imposes some maximum size on its packets. These limits have various causes, among them

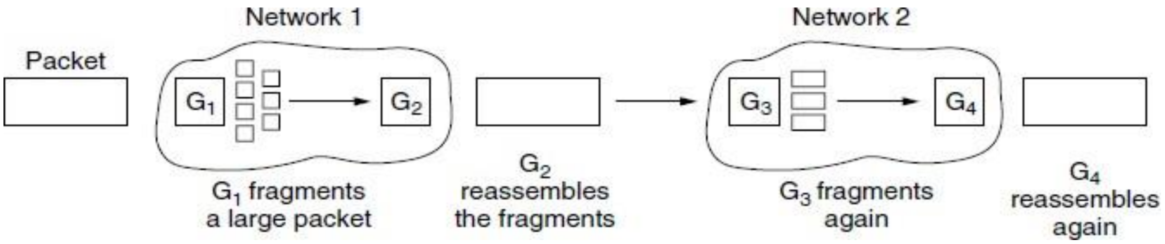
1. Hardware (e.g., the size of an Ethernet frame).
2. Operating system (e.g., all buffers are 512 bytes).
3. Protocols (e.g., the number of bits in the packet length field).
4. Compliance with some international standard.
5. Desire to reduce error-induced retransmissions to some level.
6. Desire to prevent one packet from occupying the channel too long.

- Maximum payload for Ethernet is1500 bytes and for 802.11 it is 2272 bytes. IP allowspackets size as big as 65,515 bytes.
- Hosts usually prefer to transmit large packets because this reduces packet overheads suchas bandwidth wasted on header bytes.
- An internetworking problem appears when a large packet wants to travel through anetwork whose maximum packet size is too small.
- The solution to the problem is to allow routers to break up packets into **fragments**,sending each fragment as a separate network layer packet.

## Fragmentation Techniques:

1. Transparent Fragmentation
2. Non-Transparent Fragmentation
3. Path MTU Discovery

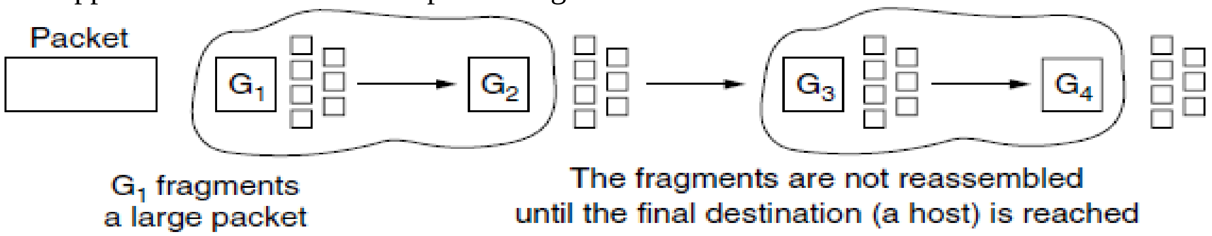
### Transparent Fragmentation



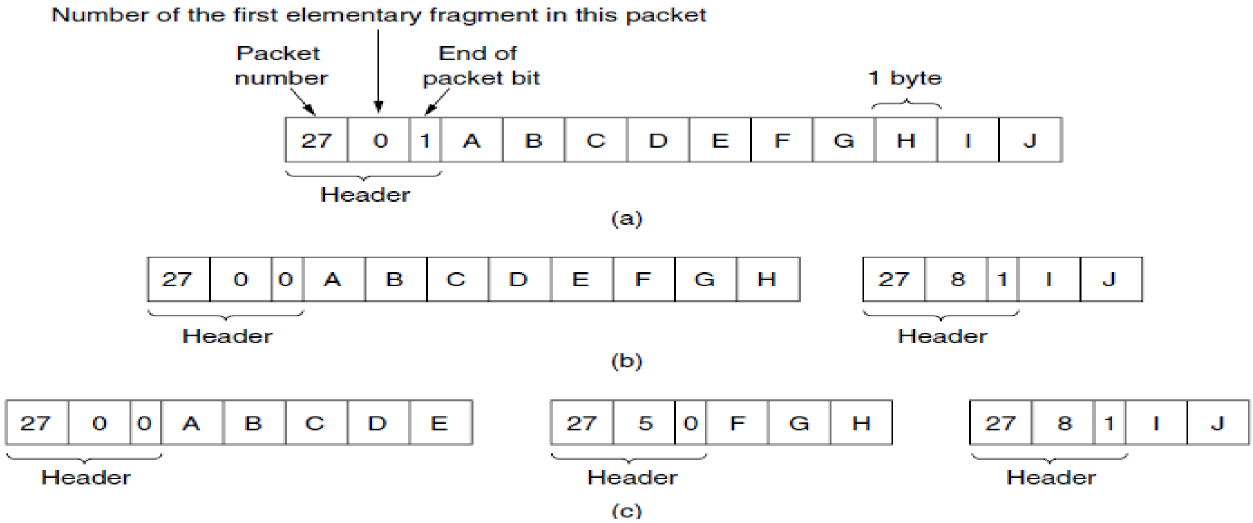
- In the above figure, when an oversized packet arrives at G1, the router breaks it up into fragments. Each fragment is addressed to the same exit router, G2, where the pieces are recombined.
- In this way, passage through the small-packet network is made transparent. Subsequent networks are not even aware that fragmentation has occurred.
- In this type of fragmentation the exit router must know when it has received all the pieces, so either a count field or an ‘‘end of packet’’ bit must be provided. Bcause all packets must exit via the same router so that they can be reassembled.
- This approach is called **Transparent fragmentation.**
- **Problem:** As the packet may pass through a series of small packet networks and need to be repeatedly fragmented and reassembled, the router has to spend a lot of time for fragmentation process only. This will lead to performance issues.

### Non-Transparent Fragmentation

- It provides the solution for transparent fragmentation, such that instead reassembling at the intermediate router; the fragmented packets are reassembled at the destination host only.
- Thisapproach is called Non-transparent fragmentation.


Packet
$G_1$
$G_2$
$G_3$
$G_4$
$G_1$ fragments a large packet
The fragments are not reassembled until the final destination (a host) is reached

- In the above figure, once a packet has been fragmented, each fragment is treated as the original packet. The routers pass the fragment and reassembly is performed only at the destination host.
- **Advantage**: It requires routers to do less work. The fragments be numbered in such away that the original data stream can be reconstructed. The design used by IP is to give every fragment a packet number (carried on all packets), an absolute byte offset within the packet, and a flag indicating whether it is the end of the packet.



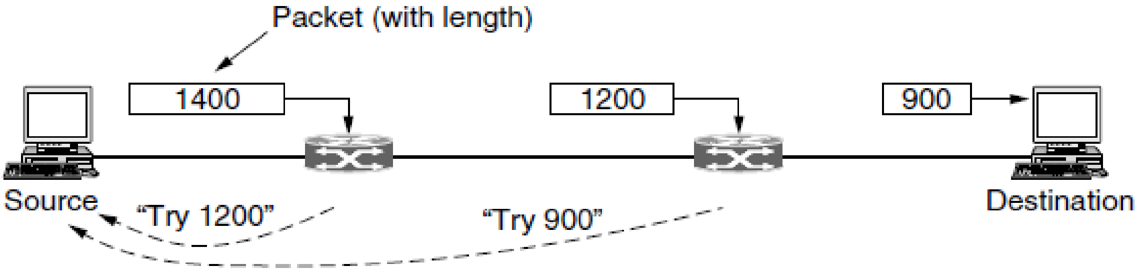
**Example**:

- Fragments can be placed in a buffer at the destination in the right place for reassembly,even if they arrive out of order.
- Fragments can also be fragmented if they pass over a network with a yet smaller MTU.
- If all fragments were not received, retransmissions of the packet can be fragmented intodifferent pieces.
- The destination uses the packet number and fragment offset to place the data in the rightposition and the end-of-packet flag to determine when it has the complete packet.

**Problem:** Fragmentation is detrimental to performance because, it will have the header overheads, and if any of a packet fragment lost then it consider that entire packet is lost. Hence we need to retransmit the entire packet rather than lost fragment. This problem is solved by Path MTU Discovery method

**Path MTU Discovery**

- Each IP packet is sent with its header bits set to indicate that no fragmentation is allowed tobe performed. If a router receives a packet that is too large, it generates an error packet, returns it to the source and drops the packet.
- When the source receives the error packet, it uses the information inside to refragment thepacket into pieces that are small enough for the router to handle. If a router further down the path has an even smaller MTU, the process is repeated.

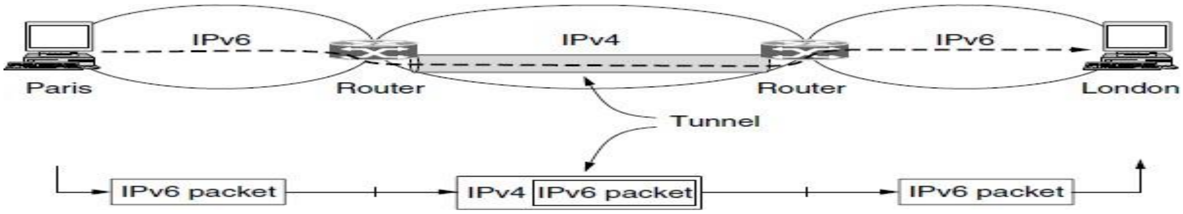


**Advantage**: The source is now know what length packet to send. If the routes and path MTU change, new error packets will be triggered and the source will adapt to the new path.

**Disadvantage:** There may be added startup delays simply to send a packet. More than one round-trip delay may be needed to probe the path and find the MTU before any data is delivered to the destination.

## TUNNELING

- Tunneling is a strategy used when two computers using IPv6 want to communicate with each other and the packet must pass through a region that uses IPv4.
- Consider the below figure that describes the concept of tunneling: Here the source and destination hosts are on the same type of network (i.e. IPv6), but there is a different network in between (i.e. IPv4).



**Example:**

- An international bank with an IPv6 network in Paris, an IPv6 network in London and connectivity between the offices via the IPv4 Internet.
- To send an IP packet to a host in the London office, a host in the Paris office constructs the packet containing an IPv6 address in London, and sends it to the multiprotocol router that connects the Paris IPv6 network to the IPv4 Internet.
- When this router gets the IPv6 packet, it encapsulates the packet with an IPv4 header addressed to the IPv4 side of the multiprotocol router that connects to the London IPv6 network. That is, the router puts a (IPv6) packet inside a (IPv4) packet.
- When this wrapped packet arrives, the London router removes the original IPv6 packet and sends it onward to the destination host.
- The path through the IPv4 Internet can be seen as a big tunnel extending from one multiprotocol router to the other. The IPv6 packet just travels from one end of the tunnel to the other end of the tunnel.
- Only the multiprotocol routers have to understand both IPv4 and IPv6 packets.
- Tunneling is widely used to connect isolated hosts and networks using other networks. The network that results is called an **overlay network,** since it has effectively been overlaid on the base network.

**Application of Tunneling:**

- A Virtual Private Network is an overlay that is used to provide a measure of security.

## INTERNET CONTROL MESSAGE PROTOCOL (ICMP)

- The IP provides unreliable and connectionless datagram delivery. (i.e.) IP does not provideany Error control mechanisms and it does not provide any host management queries.
- ICMP has been designed to compensate for the above two deficiencies. It is a companion tothe IP protocol.

**Types of Messages**

ICMP messages are divided into two broad categories:

1. Error-Reporting Messages
2. Query Messages

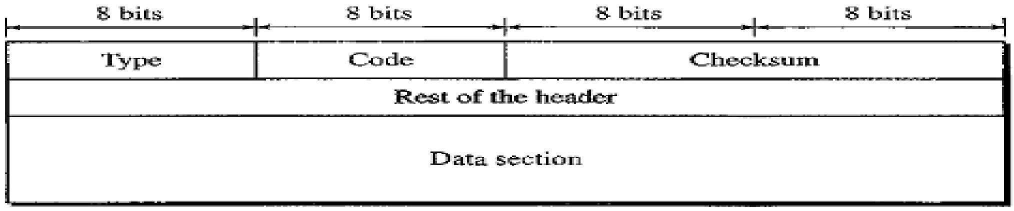
- **Error-reporting messages** report problems that a router or a host (destination) may encounter when it processes an IP packet.
- **Query messages** occur in pairs, help a host or a network manager get specific information from a router or another host.

**Example:** nodes can discover their neighbors, and also hosts can discover and learn aboutrouters on their network, and routers can help a node redirect its messages.

**Message Format**

An ICMP message has an 8-byte header and a variable-size data section.

The general format of the header is different for each message type, the first 3 fields Type, Code, Checksum are common to all. These common fields consisting of 4 bytes.

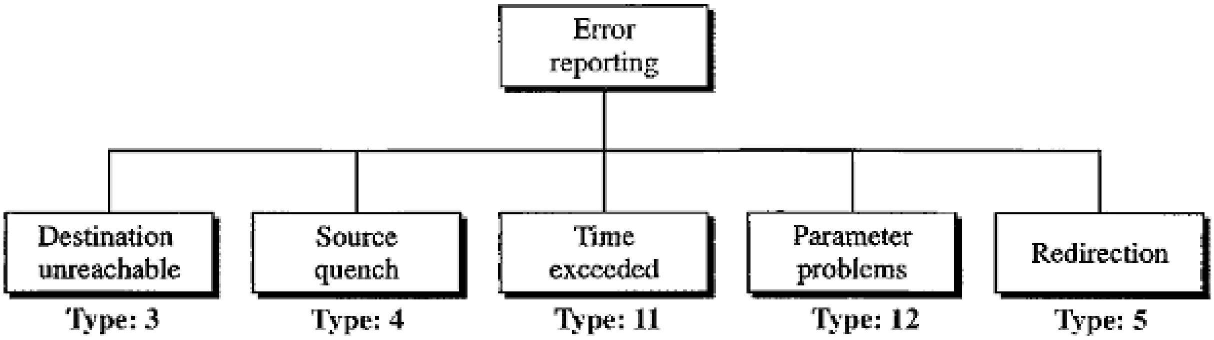


- **ICMP Type** defines the type of the message.
- **Code** field specifies the reason for the particular message type.
- **Rest of the header** is specific for each message type.
- **Checksum** is calculated over the entire message (header and data).
- **Data section** in Error messages carries information for finding the original packet that had the error. **Data Section** in Query messages the data section carries extra information based on the type of the query.

**Error Reporting**

- The main responsibilities of ICMP are to report errors. ICMP does not correct errors.
- Error messages are always sent to the original source because the only informationavailable in the datagram about the route is the source and destination IP addresses.
- ICMP uses the source IP address to send the error message to the original source of the datagram.

**ICMP handles 5 types of errors:**



**Destination Unreachable (Type 3)**

- When a router cannot route a datagram or a host cannot deliver a datagram then thedatagram is discarded.
- The router or the host sends a destination-unreachable message back to the source hostthat initiated the datagram.
- Note that destination-unreachable messages can be created by either a router or thedestination host.

**Source Quench (Type 4)**

- IP does not have a flow control mechanism embedded in the protocol.
- The lack of flow control can create major problems such as Congestion in routers or thedestination host. When there is a congestion the router or host may discard the packets.
- When a router or host discards a datagram due to congestion, it sends a source-quenchmessage to the sender of the datagram.

  **Source Quench message has two purposes.**

i. It informs the source that the datagram has been discarded.
ii. It warns the source that there is congestion somewhere in the path and that the sourceshould slow down (quench) the sending process.

**Redirection (Type 5)**

- Routing is dynamic. Routing table will be updated by routers. Host does not involve inthe process of updation of routing tables.
- The hosts usually use static routing. Routing table has a limited number of entries.
- Host usually knows the IP address of the default router only. For this reason, the host maysend a datagram, which is destined for another network to the wrong router.
- In this case the router that receives the datagram will forward the datagram to the correctrouter.
- To update the routing table of the host, it sends a redirection message to the host.

**Time Exceeded (Type 11)**

- Each datagram contains a field called Time To Live (TTL).
- When a datagram visits a router, the value of TTL field is decremented by 1.
- When the TTL value reaches 0 the router discards the datagram.
- When the datagram is discarded, a time-exceeded message must be sent by the router tothe original source.

- A time-exceeded message is also generated when not all fragments that make up amessage arrive at the destination host within a certain time limit.

**Parameter Problem (Type 12)**

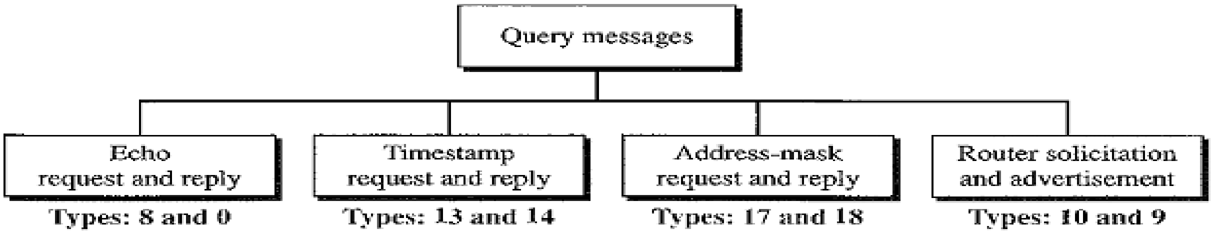
- If a router or the destination host discovers an ambiguous or missing or illegal value in any field of the datagram, it discards the datagram and sends a parameter-problem message back to the source.
- There are some cases where ICMP error messages will not be generated.

i. No ICMP error message will be generated in response to a datagram carrying an ICMPerror message.
ii. No ICMP error message will be generated for a fragmented datagram that is not the firstfragment.
iii. No ICMP error message will be generated for a datagram having a **Multicast Address**.
iv. No ICMP error message will be generated for a datagram having a special address such as
127.0.0.0 or 0.0.0.0.

**Query Messages**

- In this type of ICMP message, a node sends a message that is answered in a specificformat by the destination node.
- A query message is encapsulated in an IP packet, which in turn is encapsulated in a datalink layer frame.

Here no bytes of the original IP are included in the message.



**Echo Request and Echo Reply**

- The echo-request and reply messages can be used to determine if there is communicationat the IP level.
- These are used for diagnostic purpose, network managers and users utilize this pair ofmessages to identify network problems.
- ICMP messages are encapsulated in IP datagrams.
- The receipt of an echo-reply message by the machine that sent the echo request is proof that the IP protocols in the sender and receiver are communicating with each other using the IP datagram.

Example: **ping** command.

**Timestamp Request and Reply**

- Two machines (hosts or routers) can use the timestamp request and timestamp reply messages to determine the round-trip time needed for an IP datagram to travel between them. It can also be used to synchronize the clocks in two machines.

**Address-Mask Request and Reply**

- A host may know its IP address, but it may not know the corresponding mask.
- To obtain its mask, a host sends an address-mask-request message to a router on the LAN.

- **Ex:** A host IP address is 159.31.17.24, but it doesn't know its corresponding mask **/24**.
- If the host knows the address of the router, it sends the request directly to the router.
- If it does not know, it broadcasts the message.
- The router receiving the address-mask-request message responds with an address-mask-reply message, providing the necessary mask for the host.

### Router Solicitation and Advertisement

- The router-solicitation and router-advertisement messages can help whether the router isfunctioning or not.
- A host can broadcast (or) multicast a router-solicitation message.
- Routers that receive the solicitation message broadcast their routing information using therouter-advertisement message.
- In router advertisement message it announces its own presence and all routers on thenetwork which it is aware of.

### Debugging Tools

- There are several tools that can be used in the Internet for debugging There are two tools that are used for ICMP debugging: **ping** and **traceroute.Ping**
- Ping program to find if a host is alive and responding.
- The source host sends ICMP echo-request messages (type: 8, code: 0); the destination, if alive, responds with ICMP echo-reply messages.
- The ping program sets the identifier field in the echo-request and echo-reply message and starts the sequence number from 0; this number is incremented by 1 each time a new message is sent.
- Ping can calculate the round-trip time. It inserts the sending time in the data section of the message. When the packet arrives, it subtracts the arrival time from the departure time to get the round-trip time (RTT).

**Example: ping** program to test the server fhda.edu. The result is shown below:

```
$ ping thda.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
--- fhda.edu ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time xxxxx ms
RTT min/avg/max = 1.911/1.955/2.04 ms.
```

- **ttl=62**: means that the packet cannot travel more than 62 hops.
- ping defines the number of data bytes as 56 and the total number of bytes as 84. It is obvious that if we add 8 bytes of ICMP header and 20 bytes of IP header to 56, the result is 84.

### Traceroute

- The traceroute program in UNIX or tracert in Windows can be used to trace the route of apacket from the source to the destination.

- The program elegantly uses two ICMP messages, **Time Exceeded** and **DestinationUnreachable**, to find the route of a packet.
- Traceroute finds the routers along the path from the host to a destination IP address.
- It finds this information without any kind of privileged network support.
- The method is simply to send a sequence of packets to the destination, first with a TTL of1, then a TTL of 2, 3, and so on.
- The counters on these packets will reach zero at successive routers along the path.
- These routers will each obediently send a TIME EXCEEDED message back to the host.
- From those messages, the host can determine the IP addresses of the routers along thepath, as well as keep statistics and timings on parts of the path.
- Trace-route is perhaps the most useful network debugging tool of all time.
- This is a program at the application level that uses the services of UDP.

## ADDRESS RESOLUTION PROTOCOL (ARP)

- **ARP** protocol is used to identify the physical address of a target machine by using the logicaladdress of the target machine.
- Simply we can define ARP as it maps 32-bit IPv4 address (Logical Address) to a 48-bit MACaddress (Physical address).

### Need for Physical Address

- Anytime a host or a router has an IP datagram to send to another host or router, it has thelogical (IP) address of the receiver.
- The logical (IP) address is obtained in two ways:
  i. If the sender is the host then logical address is obtained from DNS.
  ii. If the sender is router then logical address is obtained from a routing table.
- But the IP datagram must be encapsulated in a frame to be able to pass through thephysical network.
- This means that the sender needs the physical address of the receiver.
- In order to know the physical address of the receiver the sender uses ARP protocol.

### Process of ARP

- The sender knows the IP address of the target. The host or the router sends an ARP query packet.
- IP asks ARP to create an ARP request packet. The packet includes the **Physical address** and **IP addresses** of the **Sender** and the **IP address** of the **Receiver**.
- The target physical address field is filled with all 0's. Because the sender does not know the physical address of the receiver and the query is broadcast over the network.
- Every host or router on the network receives and processes the ARP query packet, but only the intended recipient recognizes its IP address and sends back an ARP reply packet whereas the remaining devices discard the packet.
- The reply packet contains the recipient's IP and physical addresses.
- The packet is unicast directly to the sender by using the physical address received in thequery packet.

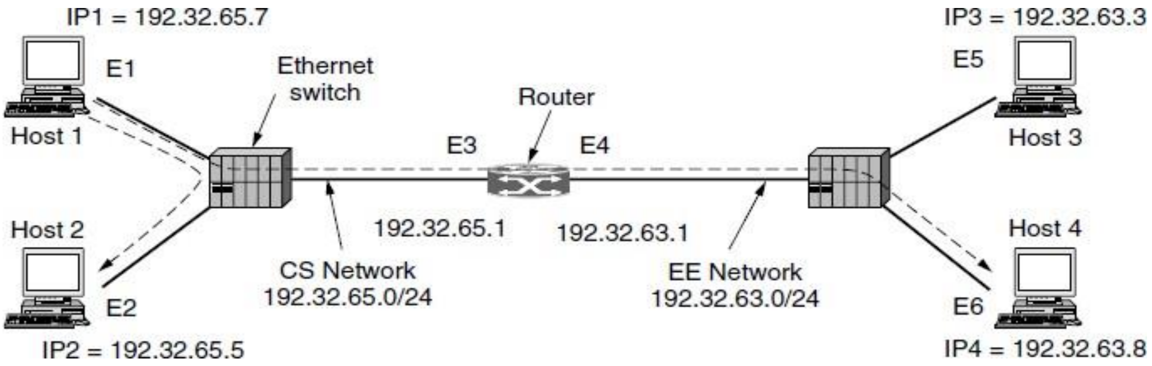
- The sender receives the reply message. It now knows the physical address of the target machine.
- The IP datagram that carries data for the target machine is now encapsulated in a frameand datagram is unicasted to the destination.

**Example:**

The below figure illustrates the process of ARP protocol, it contains 4 host computers, 2 switches, 1 router each contains IP addresses and Physical (Ethernet addresses).

Consider the below setup belongs to a small university with two /24 networks.

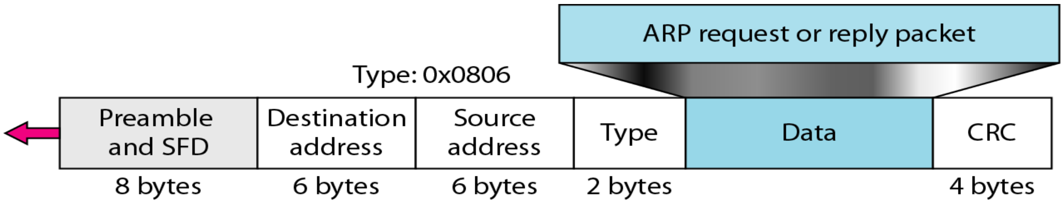
- One network (CS) is a switched Ethernet in the Computer Science Dept. It has the prefix 192.32.65.0/24.
- The other LAN (EE), also switched Ethernet, is in Electrical Engineering and has the prefix 192.32.63.0/24.
- The two LANs are connected by an IP router.
- Each machine on an Ethernet and each interface on the router has a unique Ethernetaddress, labeled E1 through E6, and a unique IP address on the CS or EE network.



- Host1 has a packet that needs to be delivered to another Host4 with IP address 192.32.63.8.
- Host1 doesn’t know the physical address of 192.32.63.8.
- Host1 broadcast ARP request packet to ask for physical address of 192.32.63.8.
- This packet is received by every system on the physical network, but only Host4 will respond by sending ARP reply packet that includes its physical address E6 (Ex: A4:6E:F4:59:83:AB).
- After receiving the response packet from Host4, the Host1 can send all the packets it has for the destination (Host4) by using the physical address it received.

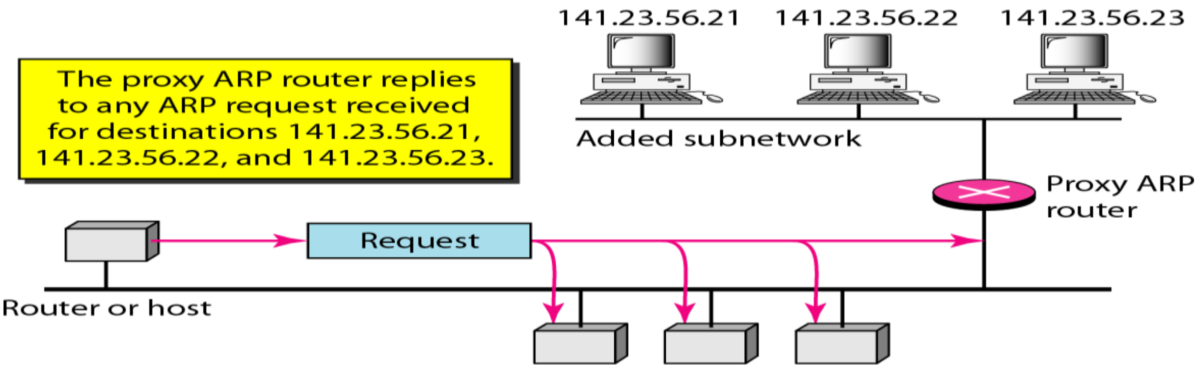
**ARP Encapsulation in FRAME**

An ARP packet is encapsulated directly into a data link frame (i.e. Ethernet frame). Note that the type field indicates that the data carried by the frame are an ARP packet.



**ProxyARP**

- A technique called **Proxy ARP** is used to create a subnetting effect.
- A proxy ARP is an ARP that acts on behalf of a set of hosts.
- Whenever a router running a proxy ARP receives an ARP request looking for the IP address of one of these hosts, the router sends an ARP reply announcing its own hardware(physical) address.
- After the router receives the actual IP packet, it sends the packet to the appropriate host orrouter.


141.23.56.21
141.23.56.22
141.23.56.23
The proxy ARP router replies to any ARP request received for destinations 141.23.56.21, 141.23.56.22, and 141.23.56.23.
Added subnetwork
Proxy ARP router
Request
Router or host

**In the above figure:** The ARP installed on the right-hand host will answer only to an ARP request with a target IP address of 141.23.56.23.

- The administrator may need to create a subnet without changing the whole system to recognize subnetted addresses. One solution is to add a router running a proxy ARP.
- In this case, the router acts on behalf of all the hosts installed on the subnet.
- When it receives an ARP request with a target IP address that matches the address of one of its protégés (141.23.56.21, 141.23.56.22, or 141.23.56.23), it sends an ARP reply and announces its hardware address as the target hardware address.
- When the router receives the IP packet, it sends the packet to the appropriate host.

**Gratuitous ARP**

- Once a machine has run ARP, it caches the result in case it needs to contact the same machine shortly. Next time it will find the mapping in its own cache, thus eliminating theneed for a second broadcast.
- In many cases, Destination hosts will need to send back a reply, forcing it to run ARP to determine the sender's Ethernet address.
- This ARP broadcast can be avoided by having sender include its IP-to-Ethernet mapping in the ARP packet.
- When the ARP broadcast arrives at Destination Host, the pair (192.32.65.7, E1) is enteredinto Destination host ARP cache. In fact, All machines on the Ethernet can enter this mapping into their ARP caches.
- To allow mappings to change, when a host is configured to use a new IP address but keeps its old Ethernet address entries in the ARP cache should time out after a few minutes.
- One way to help keep the cached information updated and to optimize performance, everymachine broadcast its mapping when it is configured.
- This broadcast is generally done in the form of an ARP looking for its own IP address.
- There should not be a response from other machine, because the broadcast is done to make or update an entry in everyone's ARP cache.

- This is known as a **Gratuitous ARP**.

## REVERSE ADDRESS RESOLUTION PROTOCOL (RARP)

- RARP finds the logical address for a machine that knows only its physical address.
- Each host or router is assigned one or more logical (IP) addresses, which are unique andindependent of the physical (hardware) address of the machine.
- To create an IP datagram, a host or a router needs to know its own IP address.
- IP address of a machine is usually read from its configuration file stored on a disk file.
- The machine can get its physical address by reading its NIC, which is unique locally.
- It can then use the physical address to get the logical address by using the RARPprotocol.
- A RARP request is created and broadcast on the local network. Another machine on thelocal network that knows all the IP addresses will respond with a RARP reply.

**Problems with RARP**

- Broadcasting is done at the data link layer.
- The physical broadcast address, all 1's in the case of Ethernet (i.e FF:FF:FF: FF:FF:FF),it doesn't pass the boundaries of a network. (i.e.) if an administrator has several networks or several subnets, it needs to assign a RARP server for each network or subnet.

## BOOTSTRAP Protocol (BOOTP)

- BOOTP is a client/server protocol designed to provide physical address to logical addressmapping.
- BOOTP is an application layer protocol. The administrator may put the client and theserver on the same network or on different networks.
- BOOTP messages are encapsulated in a UDP packet, and the UDP packet itself isencapsulated in an IP packet.

**Advantage of BOOTP over RARP**

- The client and server are application-layer processes. As in other application-layer processes,a client can be in one network and the server in another, separated by several other networks.

## DYNAMIC HOST CONFIGURATION PROTOCOL (DHCP)

- DHCP is a static and dynamic configuration protocol whereas BOOTP is a staticconfiguration protocol only.

**Why DHCP?**

- When a client requests its IP address, the BOOTP server consults a table that matches thephysical address of the client with its IP address.
- This implies that the binding between the physical address and the IP address of the clientalready exists. The binding is predetermined.
- The binding or mapping between the physical address and IP addresses is static and fixedin a table

until changed by the administrator. BOOTP is a static configuration protocol.

***There are situations where BOOTP fails to handle:***

i. What if a host moves from one physical network to another.

ii. What if a host wants a temporary IP address.

DHCP has been devised to provide **Static** and **Dynamic Address Allocation** that can bemanual or automatic.

**Static Address Allocation**

- In this capacity DHCP acts as BOOTP.
- It is backward compatible with BOOTP, which means a host running the BOOTP clientcan request a static address from a DHCP server.
- A DHCP server has a database that statically binds physical addresses to IP addresses.

**Dynamic Address Allocation**

- DHCP has a second database with a pool of available IP addresses. This second database makes DHCP dynamic.
- When a DHCP client requests a temporary IP address, the DHCP server goes to the pool of available (unused) IP addresses and assigns an IP address for a negotiable period of time.
- When a DHCP client sends a request to a DHCP server, the server first checks its static database.
- If an entry with the requested physical address exists in the static database, the permanent IP address of the client is returned.
- If the entry does not exist in the static database, the server selects an IP address from the available pool, assigns the address to the client, and adds the entry to the dynamicdatabase.
- The dynamic aspect of DHCP is needed when a host moves from network to network oris connected and disconnected from a network then DHCP provides temporary IP addresses for a limited time.

**Temporary Addresses**

- The addresses assigned from the pool are temporary addresses.
- The DHCP server issues a lease for a specific time. When the lease expires, the client must either stop using the IP address or renew the lease.
- The server has the option to agree or disagree with the renewal. If the server disagrees, the client stops using the address.

**Advantages of DHCP over BOOTP**

- One major problem with the BOOTP protocol is that the table mapping. The IP addressesto physical addresses needs to be manually configured.
- This means the administrator needs to manually enter the changes every time there is achange in a physical address or IP address.
- DHCP allows both manual and automatic configurations.
- Static addresses are created manually whereas dynamic addresses are createdautomatically.