MLR Institute of Technology

Unit-III

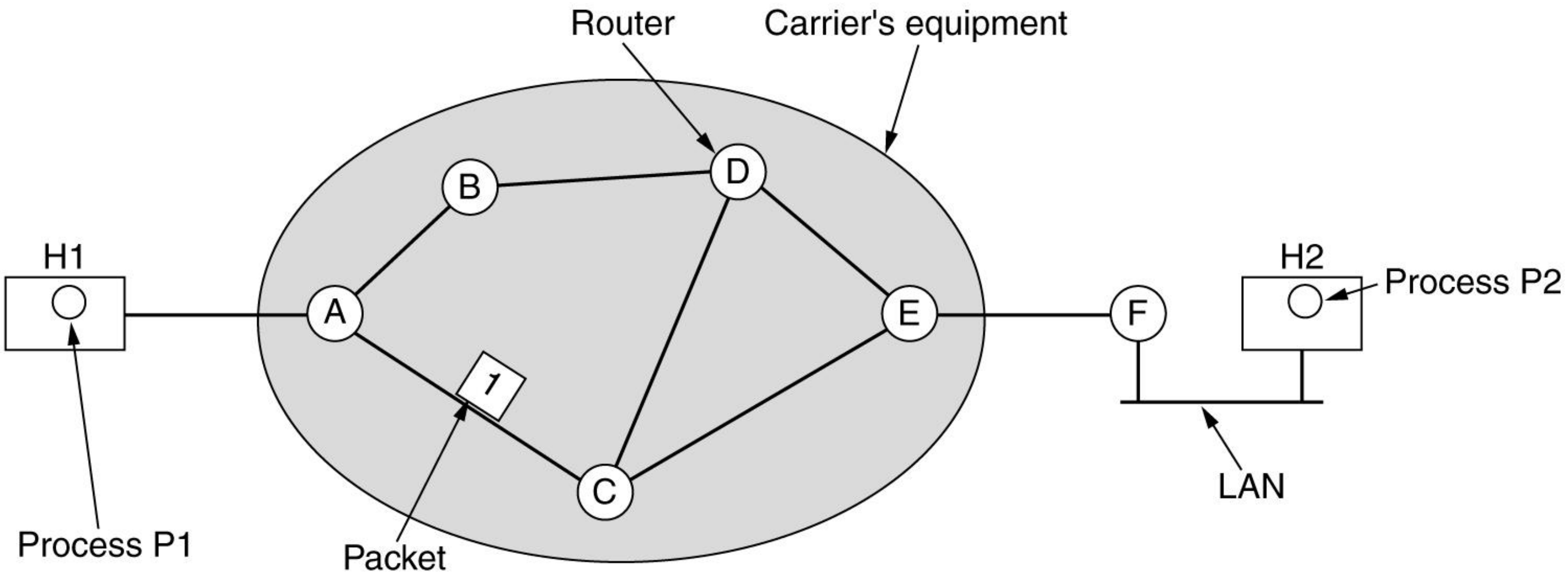
The Network Layer

Faculty: V.HEMA
Dept: CSE
EmpID:MLRIT1662

Network Layer Design Isues

- Store-and-Forward Packet Switching
- Services Provided to the Transport Layer
- Implementation of Connectionless Service
- Implementation of Connection-Oriented Service
- Comparison of Virtual-Circuit and Datagram Subnets

Store-and-Forward Packet Switching



The environment of the network layer protocols.

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 carrier.
- The packet is stored there until it has fully arrived so the checksum can be verified. Then it is forwarded to the next router along the path until it reaches the destination host, where it is delivered.
- This mechanism is store-and-forward packet switching, as we have seen in previous chapters.

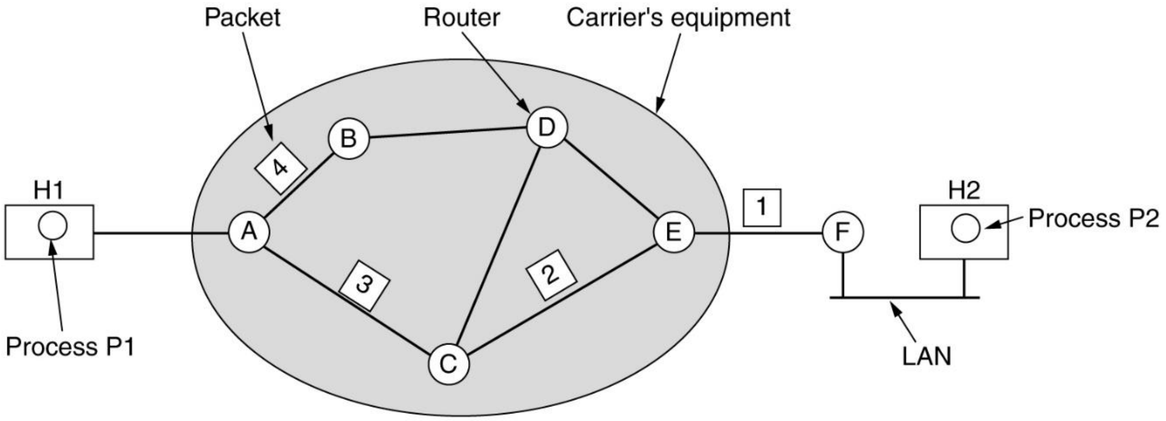
Services Provided to the Transport Layer

1. The service should be independent of the router technology.

2. The transport layer should be shielded from the number, type, and topology of the routers present.

3. The network addresses made available to the transport layer should use a uniform numbering plan, even across LANs and WANs.

Implementation of Connectionless Service



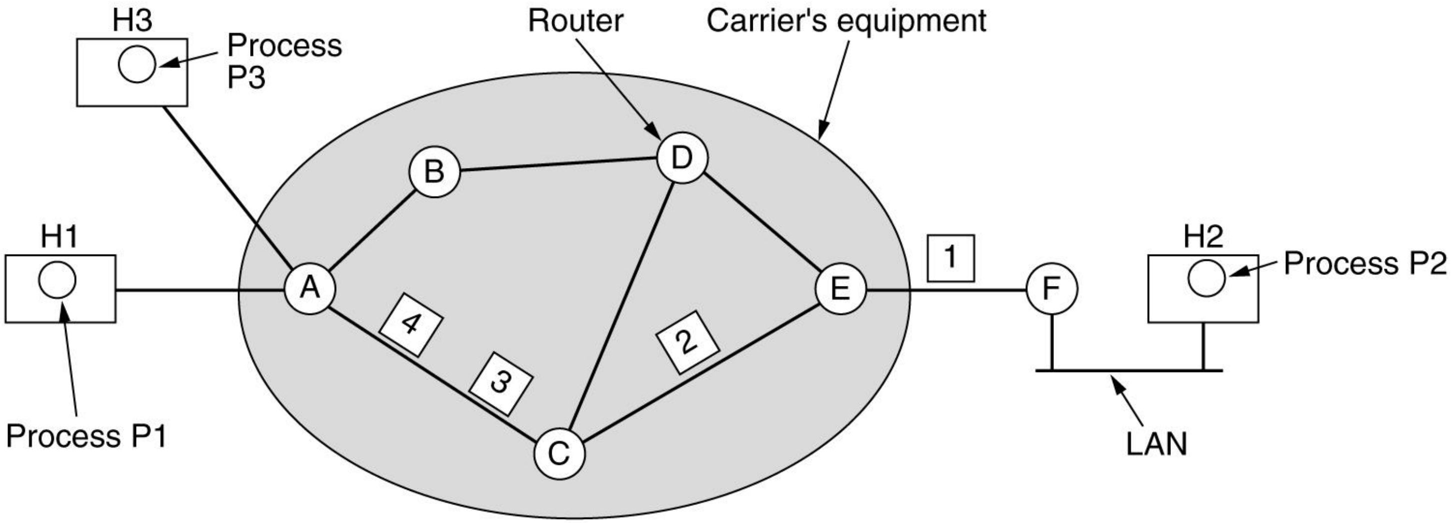
A's table			C's table		E's table	
initially		later				
A	–	–	A	A	A	C
B	B	B	B	A	B	D
C	C	C	C	–	C	C
D	B	B	D	D	D	D
E	C	B	E	E	E	–
F	C	B	F	E	F	F
Dest.	Line					

Routing within a datagram subnet.

Implementation of Connectionless Service

- In this context, the packets are frequently called **datagrams** (in analogy with telegrams) and
- the subnet is called a **datagram subnet**.

Implementation of Connection-Oriented Service



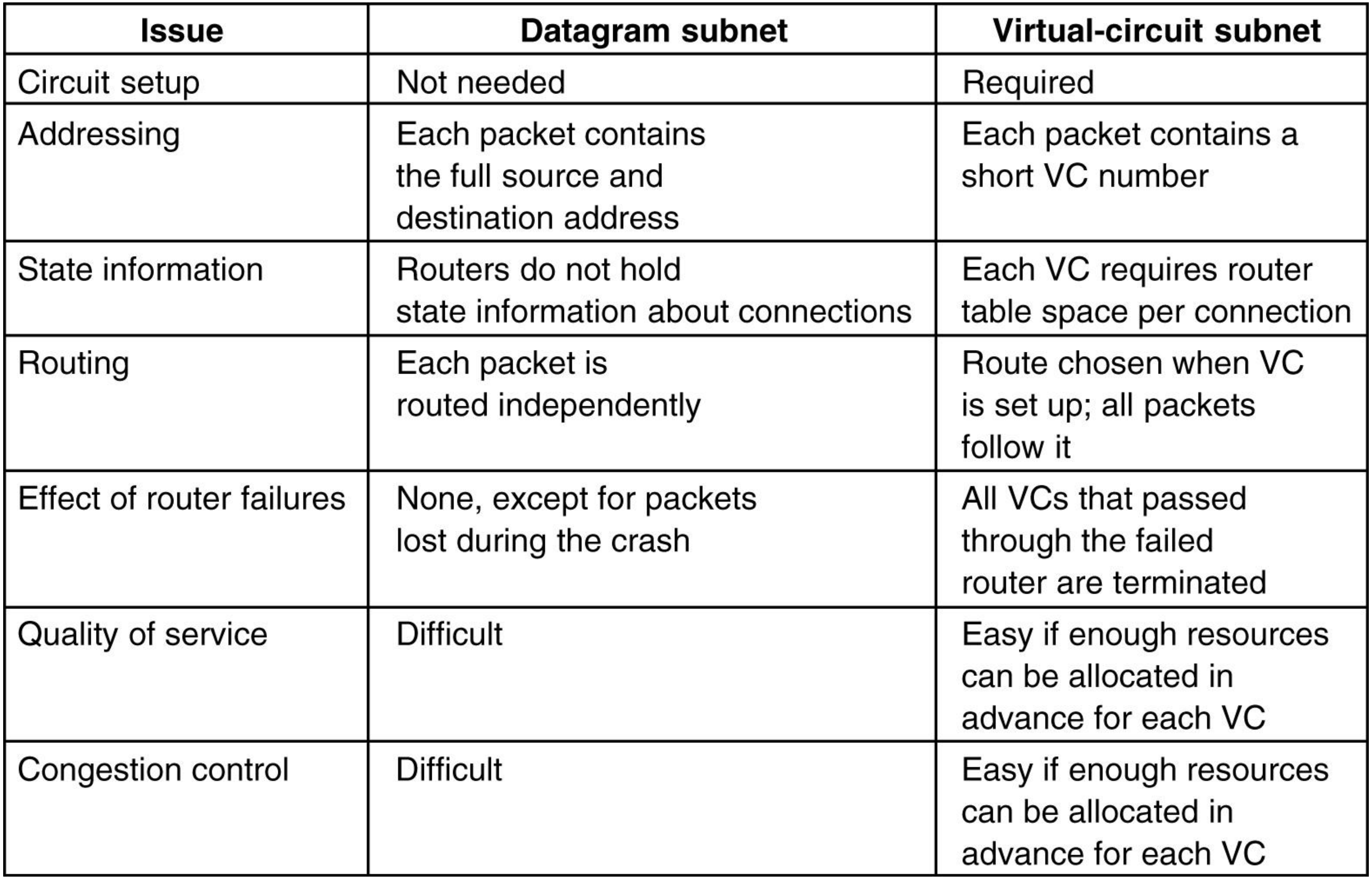
A's table		C's table		E's table	
In	Out				
H1 1	C 1	A 1	E 1	C 1	F 1
H3 1	C 2	A 2	E 2	C 2	F 2

Routing within a virtual-circuit subnet.

Implementation of Connection-Oriented Service

- In connection-oriented service a path from the source router to the destination router must be established before any data packets can be sent.
- This connection is called a **VC** (**virtual circuit**), in analogy with the physical circuits set up by the telephone system, and
- the subnet is called a **virtual-circuit subnet**.

Comparison of Virtual-Circuit and Datagram Subnets



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

Routing Algorithms

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

Routing Algorithms

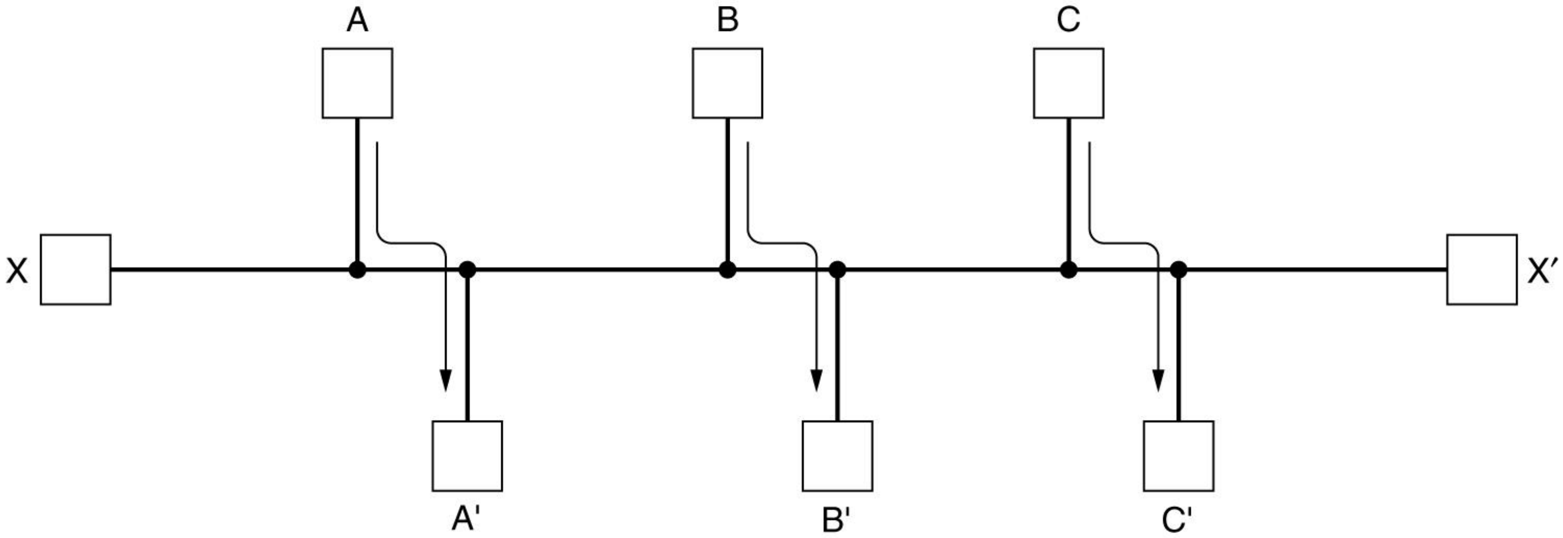
The **routing algorithm** is that part of the network layer software responsible for deciding which output line an incoming packet should be transmitted.

- If the subnet uses datagrams internally, this decision must be made anew for every arriving data packet since the best route may have changed since last time.
- If the subnet uses virtual circuits internally, routing decisions are made only when a new virtual circuit is being set up. Thereafter, data packets just follow the previously-established route.
- The latter case is sometimes called **session routing** because a route remains in force for an entire user session (e.g., a login session at a terminal or a file transfer).

The properties desirable in a routing algorithm are:

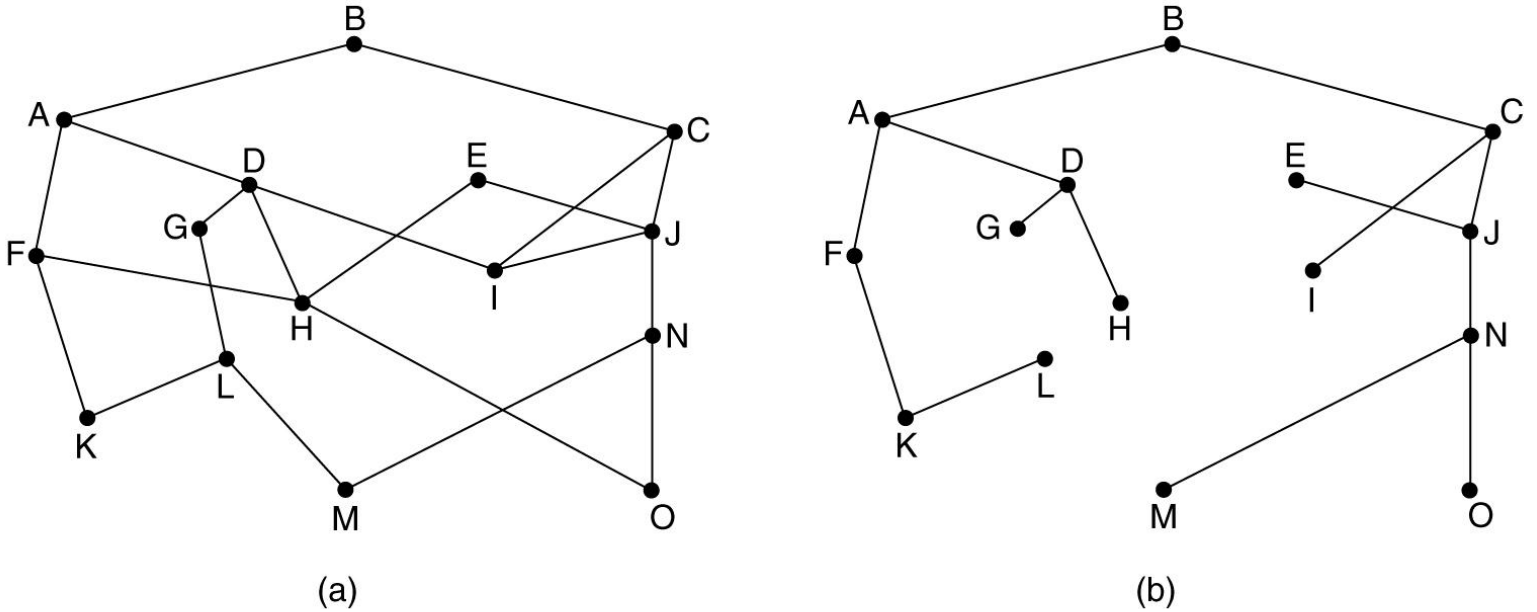
1. Correctness.
2. Simplicity.
3. Robustness.
4. Stability.
5. Fairness.
6. Optimality.

Routing Algorithms (2)



Conflict between fairness and optimality.

The Optimality Principle



(a) A subnet. (b) A sink tree for router B.

The Optimality Principle

- **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 route.
- 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 sink tree not necessarily be unique, other trees with the same path lengths may exists.
- The goal of all routing algorithms is to discover and use the sink tree for all routers.
- Since a sink tree is indeed a tree, it does not contain any loops, so each packet will be delivered within a finite and bounded number of hops.

Network Performance Measures

- Two Performance Measures
  - Quantity of Service (Throughput)
    - How much data travels across the net?
    - How long does it take to transfer long files?
  - Quality of Service (Average packet delay)
    - How long does it take for a packet to arrive at its destination?
    - How responsive is the system to user commands?
    - Can the network support real-time delivery such as audio and video?

Types of Routing Algorithms

- Nonadaptive (static)
  a) Do not use measurements of current conditions
  b) Static routes are downloaded at boot time
- Adaptive Algorithms
  a) Change routes dynamically
    a) Gather information at runtime
      a) locally
      b) from adjacent routers
      c) from all other routers
    b) Change routes
      a) Every delta T seconds
      b) When load changes
      c) When topology changes

MLR Institute of Technology

Shortest Path Routing (a nonadaptive routing algorithm)

- Find the shortest path from a specified source to all other destinations in the network.
- Given a network topology and a set of weights describing the cost to send data across each link in the network
- Shortest path algorithm first developed by E. W. Dijkstra

Shortest Path Routing

- Dijkstra (1959) shortest path alg. Between two nodes.

- Each node is labelled (in parenthesis) with its distance from the sourse node along the best known path.
- Initially, no paths are known , so

Shortest Path Routing
(a nonadaptive routing algorithm)

Mark the source node as permanent.

Designate the source node as the working node.

Set the tentative distance to all other nodes to infinity.

While some nodes are not marked permanent

> Compute the tentative distance from the source to all nodes adjacent to the working node. If this is shorter than the current tentative distance replace the tentative distance of the destination and record the label of the working node there.
>
> Examine ALL tentatively labeled nodes in the graph. Select the node with the smallest value and make it the new working node. Designate the node permanent.

Shortest Path Routing

- Each node is labeled (in parentheses) with its distance from the source node along the best known path.
- Initially, no paths are known, so all nodes are labeled with infinity. As the algorithm proceeds and paths are found, the labels may change, reflecting better paths.
- A label may be either tentative or permanent.
- Initially, all labels are tentative.
- When it is discovered that a label represents the shortest possible path from the source to that node, it is made permanent and never changed thereafter.

Example of Shortest Path Routing

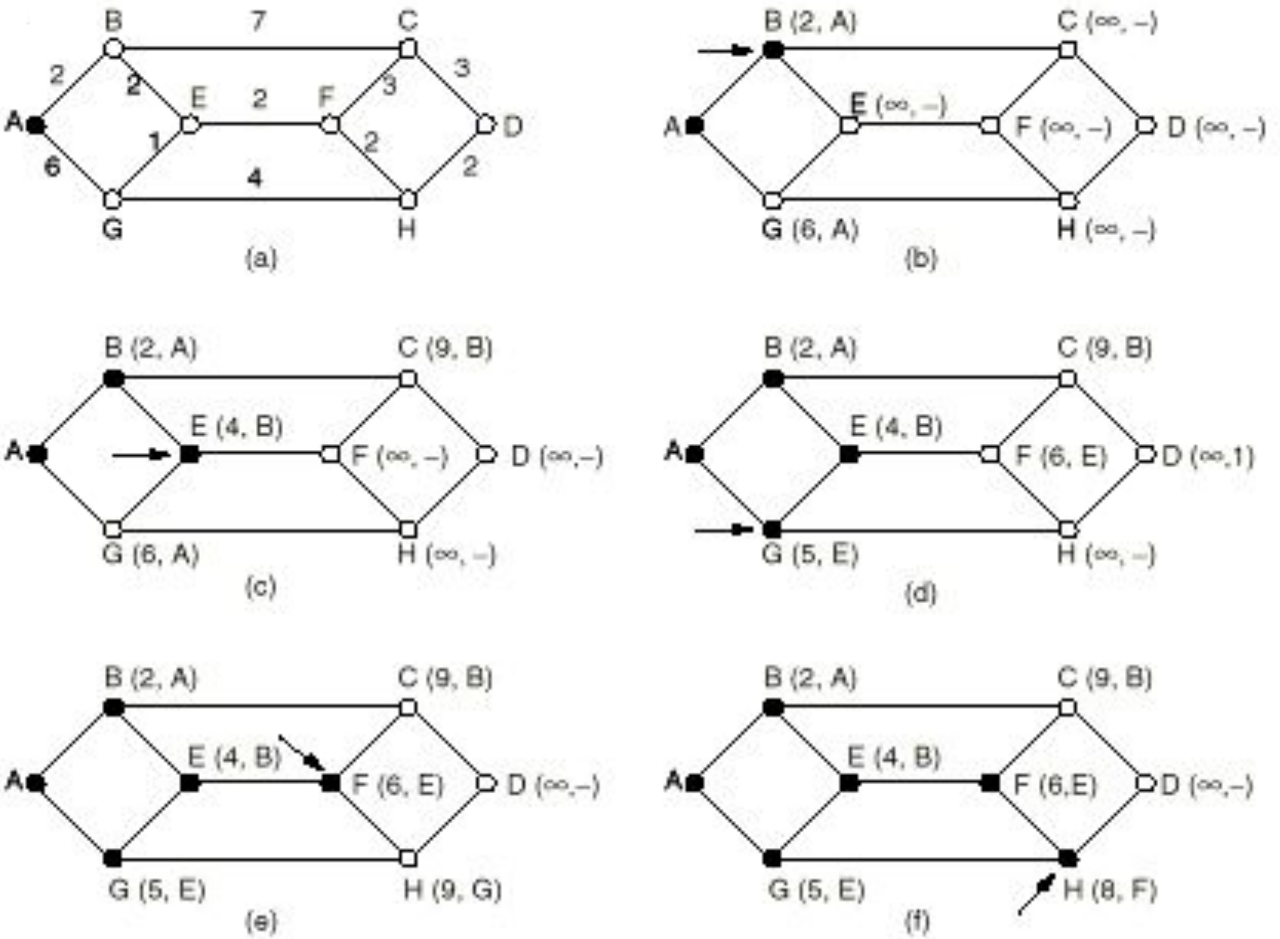


Fig. 5-6. The first five steps used in computing the shortest path from *A* to *D*. The arrows indicate the working node.

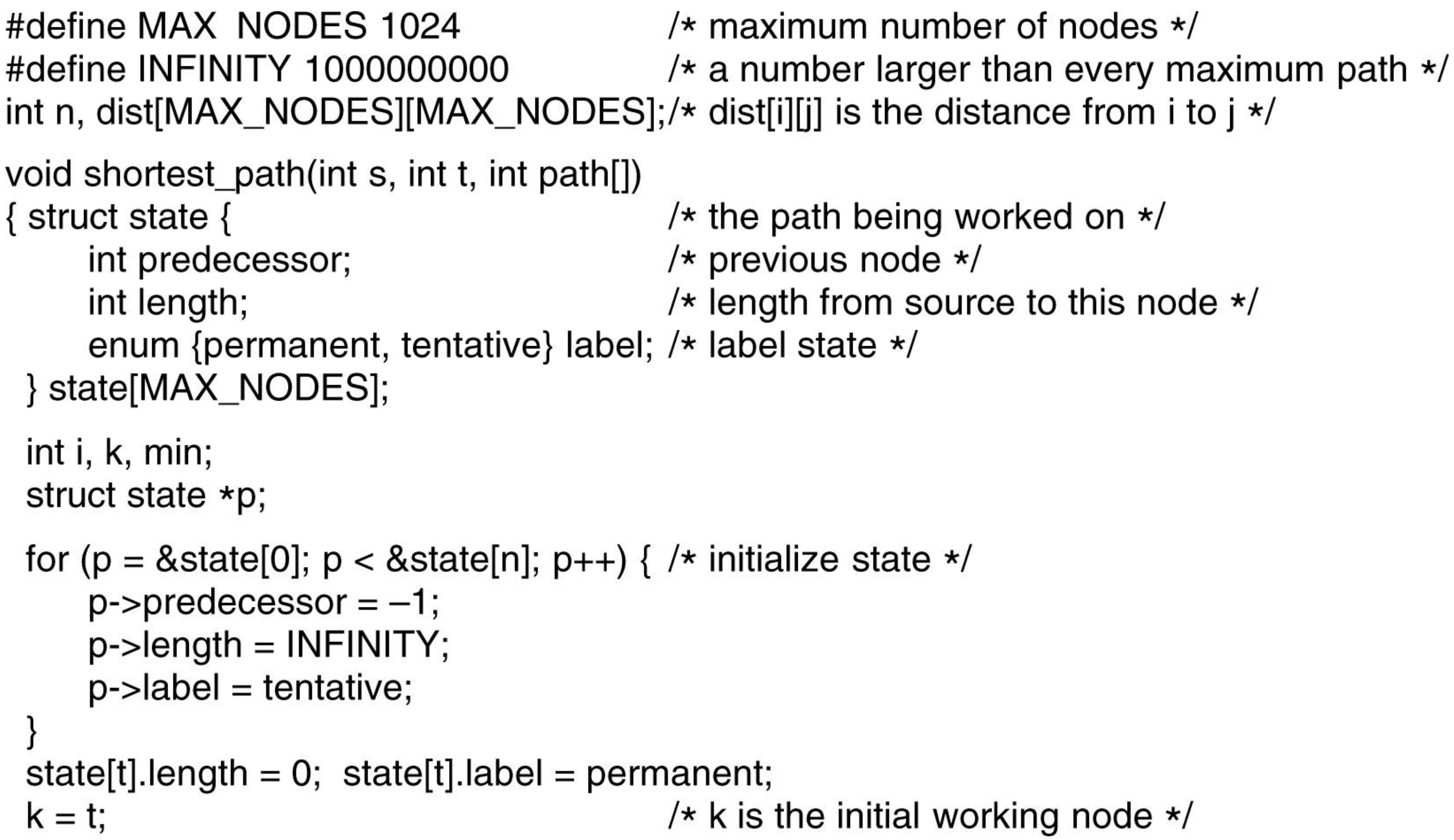
Shortest Path Routing

- We want to find the shortest path from *A* to *D*.
- We start out by marking node *A* as permanent, indicated by a filled-in circle.
- Then we examine, in turn, each of the nodes adjacent to *A* (the working node), relabeling each one with the distance to *A*.
- Whenever a node is relabeled, we also label it with the node from which the probe was made so that we can reconstruct the final path later.
- Having examined each of the nodes adjacent to *A*, we examine all the tentatively labeled nodes in the whole graph and make the one with the smallest label permanent, This one becomes the new working node.

- We now start at *B* and examine all nodes adjacent to it. If the sum of the label on *B* and the distance from *B* to the node being considered is less than the label on that node, we have a shorter path, so the node is relabeled.

- After all the nodes adjacent to the working node have been inspected and the tentative labels changed if possible, the entire graph is searched for the tentatively-labeled node with the smallest value. This node is made permanent and becomes the working node for the next round.

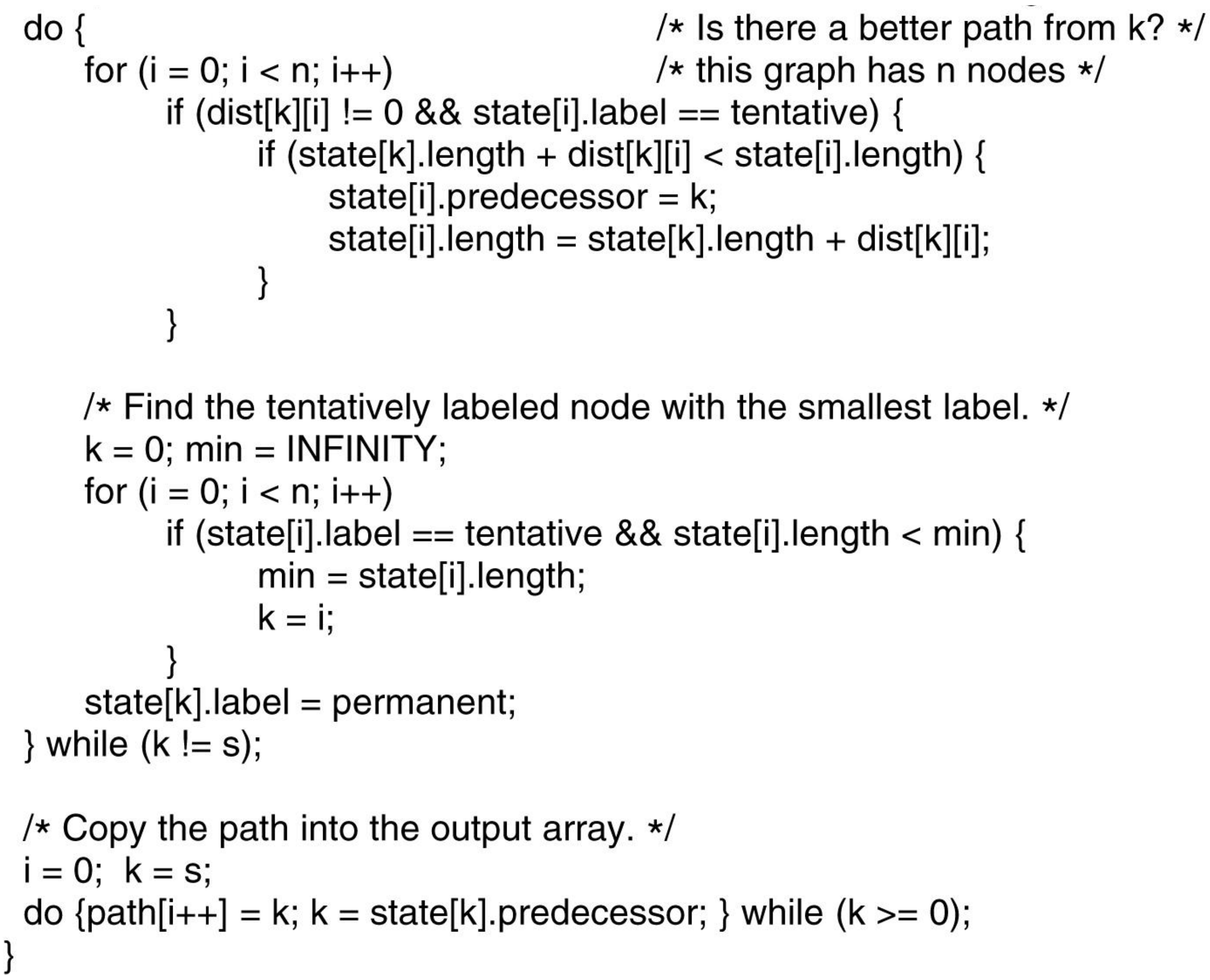
Flooding



```
#define MAX  NODES 1024                  /* maximum number of nodes */
#define INFINITY 1000000000              /* a number larger than every maximum path */
int n, dist[MAX_NODES][MAX_NODES];/* dist[i][j] is the distance from i to j */

void shortest_path(int s, int t, int path[])
{ struct state {                         /* the path being worked on */
        int predecessor;                 /* previous node */
        int length;                      /* length from source to this node */
        enum {permanent, tentative} label; /* label state */
  } state[MAX_NODES];

  int i, k, min;
  struct state *p;

  for (p = &state[0]; p < &state[n]; p++) { /* initialize state */
        p->predecessor = –1;
        p->length = INFINITY;
        p->label = tentative;
  }
  state[t].length = 0;  state[t].label = permanent;
  k = t;                                 /* k is the initial working node */
```

Dijkstra's algorithm to compute the shortest path through a graph.

Flooding (2)



```
do {                                            /* Is there a better path from k? */
    for (i = 0; i < n; i++)                     /* this graph has n nodes */
         if (dist[k][i] != 0 && state[i].label == tentative) {
              if (state[k].length + dist[k][i] < state[i].length) {
                   state[i].predecessor = k;
                   state[i].length = state[k].length + dist[k][i];
              }
         }

    /* Find the tentatively labeled node with the smallest label. */
    k = 0; min = INFINITY;
    for (i = 0; i < n; i++)
         if (state[i].label == tentative && state[i].length < min) {
              min = state[i].length;
              k = i;
         }
    state[k].label = permanent;
} while (k != s);

/* Copy the path into the output array. */
i = 0;  k = s;
do {path[i++] = k; k = state[k].predecessor; } while (k >= 0);
}
```

Dijkstra's algorithm to compute the shortest path through a graph.

Flooding
(a nonadaptive routing algorithm)

a) Brute force routing
   a) Every incoming packet is sent on every outgoing line
   b) Always finds the shortest path quickly
   c) Also finds many long paths
   d) Time to live is set to size of subnet

b) Selective Flooding
   a) Flood only in the direction of the destination

c) Practical in a few settings
   a) Military Applications
   b) Distributed Databases
   c) Metric for comparison

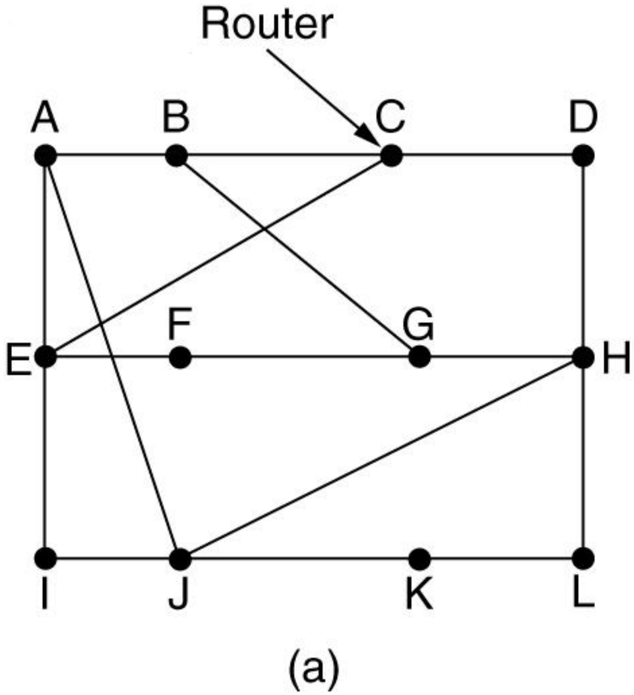
Distance Vector Routing (an adaptive routing algorithm)

- It is also known as Bellman-Ford Routing Or Ford Fulkerson Algorithm
- It is the Original ARPANET routing algorithm
- Previously used on Internet wit the name RIP.
- Early version of DecNet and Novell's IPX
- AppleTalk and Cisco routers use improved versions of this algorithm

Distance Vector Routing
(an adaptive routing algorithm)

- **Distance vector routing** algorithms operate by having each router maintain a table (i.e, a vector) giving the best known distance to each destination and which line to use to get there.
- These tables are updated by exchanging information with the neighbors.
- Neighboring routers periodically exchange information from their routing tables.
- Routers replace routes in their own routing tables anytime that neighbors have found better routes.
- Information provided from neighbors
  a) Outgoing line used for destination
  b) Estimate of time or distance
    a) can be number of hops, time delay, packet queue length, etc.

Distance Vector Routing



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

Vectors received from J's four neighbors

New routing table for J

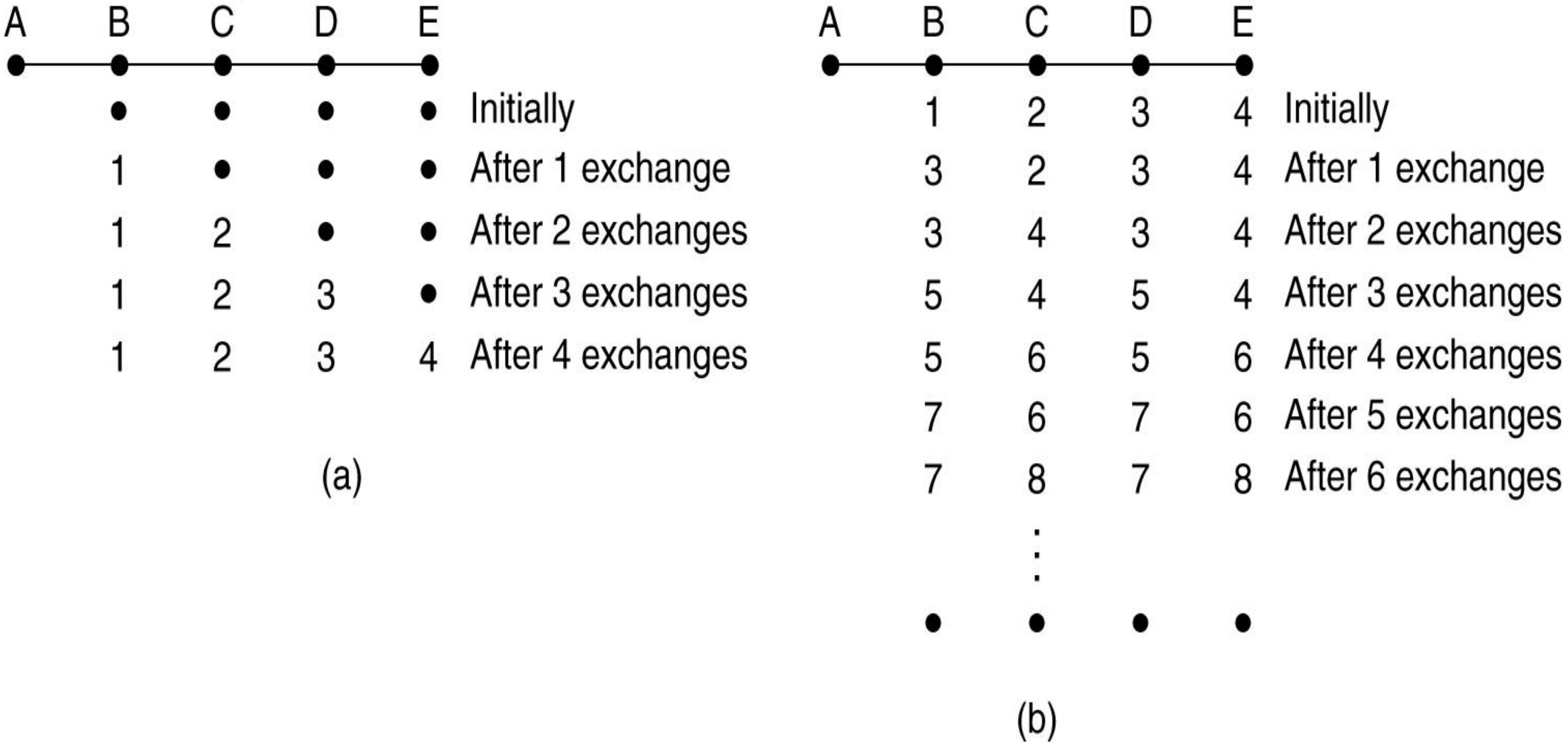
(b)

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

Distance Vector Routing

- Consider how *J* computes its new route to router *G*. It knows that it can get to *A* in 8 msec, and *A* claims to be able to get to *G* in 18 msec, so *J* knows it can count on a delay of 26 msec to *G* if it forwards packets bound for *G* to *A*.
- Similarly, it computes the delay to *G* via *I*, *H*, and *K* as 41 (31 + 10), 18 (6 + 12), and 37 (31 + 6) msec, respectively.
- The best of these values is 18,
- so it makes an entry in its routing table that the delay to *G* is 18 msec
- and that the route to use is via *H*.
- The same calculation is performed for all the other destinations, with the new routing table shown in the last column of the figure.

Distance Vector Routing (2)



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

(a)

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

(b)

The count-to-infinity problem.

Distance Vector Routing- The Count to infinity Problem

- Distance vector routing works in theory but has a serious drawback in practice: It reacts rapidly to good news, but leisurely to bad news.
- To see how fast good news propagates, consider the five-node (linear) subnet of fig (a),
- where the delay metric is the number of hops. Suppose *A* is down initially and all the other routers know this. In other words, they have all recorded the delay to *A* as infinity.
- When *A* comes up, the other routers learn about it via the vector exchanges.
- At the time of the first exchange,
- *B* learns that its left neighbor has zero delay to *A*. *B* now makes an entry in its routing table that *A* is one hop away to the left. All the other routers still think that *A* is down.

Link State Routing

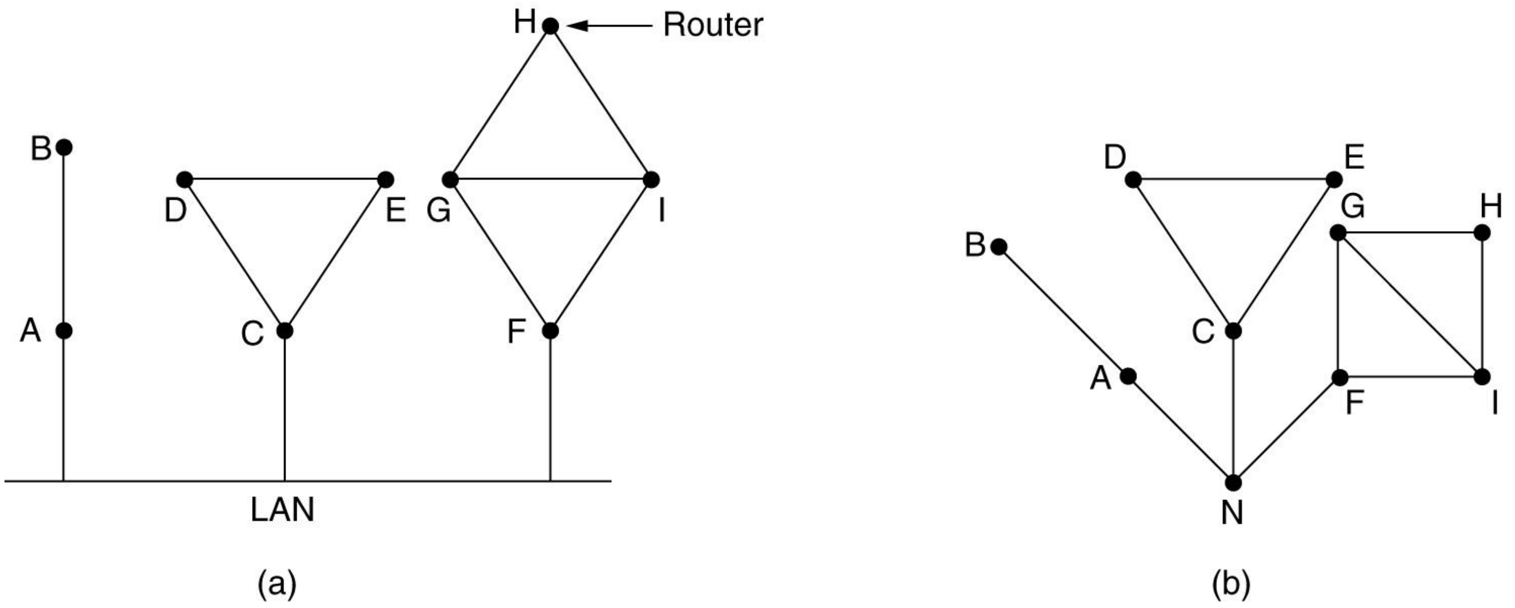
Each router must do the following five steps:

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

1). Discovering Your Neighbors

a) Send "Hello" packet on each point-to-point line. Destination node replies with its address.

Learning about the Neighbors



(a) Nine routers and a LAN. (b) A graph model of (a).

2.) Measuring Line Cost

- Send an “ECHO” packet over the line.
- Destination is required to respond to “ECHO” packet immediately.
- Measure the time required for this operation.
- Question: Should we measure just the time it takes to transmit the packet, or should we include the time that the packet waits in the queue?

Argument 2:

- We should include the time that the packet spends in the queue, as this provides a more accurate picture of the real delays.
- We should only include the transmission times, otherwise the network is likely to oscillate between preferred paths.

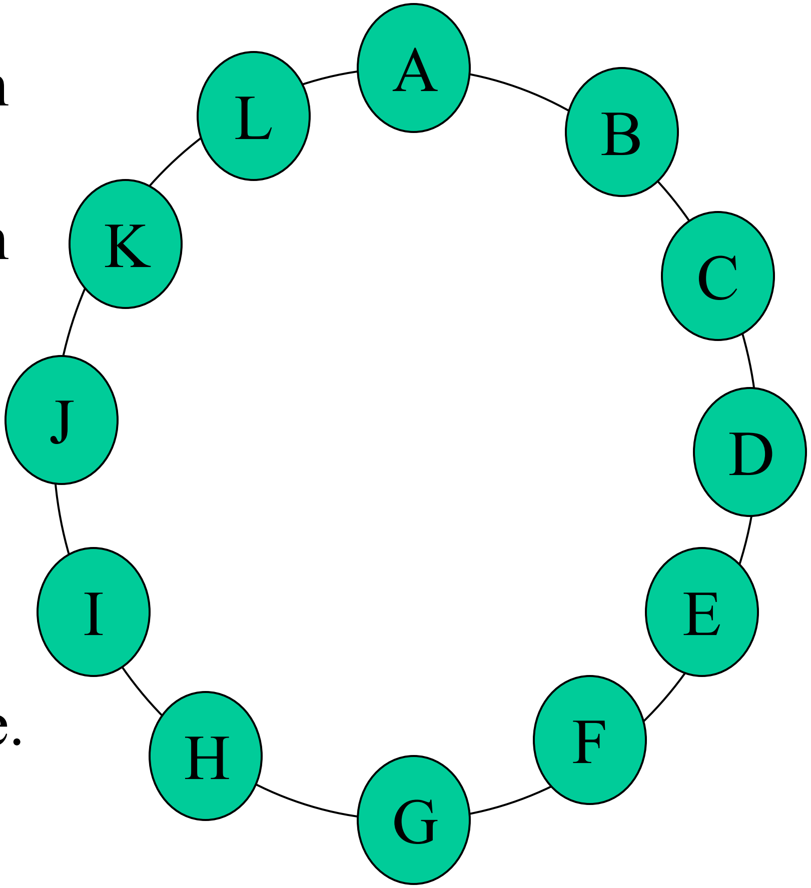
- If only band width is considered (load is ignored), this problem does not occur.

- To avoid oscillations in the choice of best path, it may be wise to distribute the load over multiple lines, with some known fraction going over each line.

Oscillating Paths

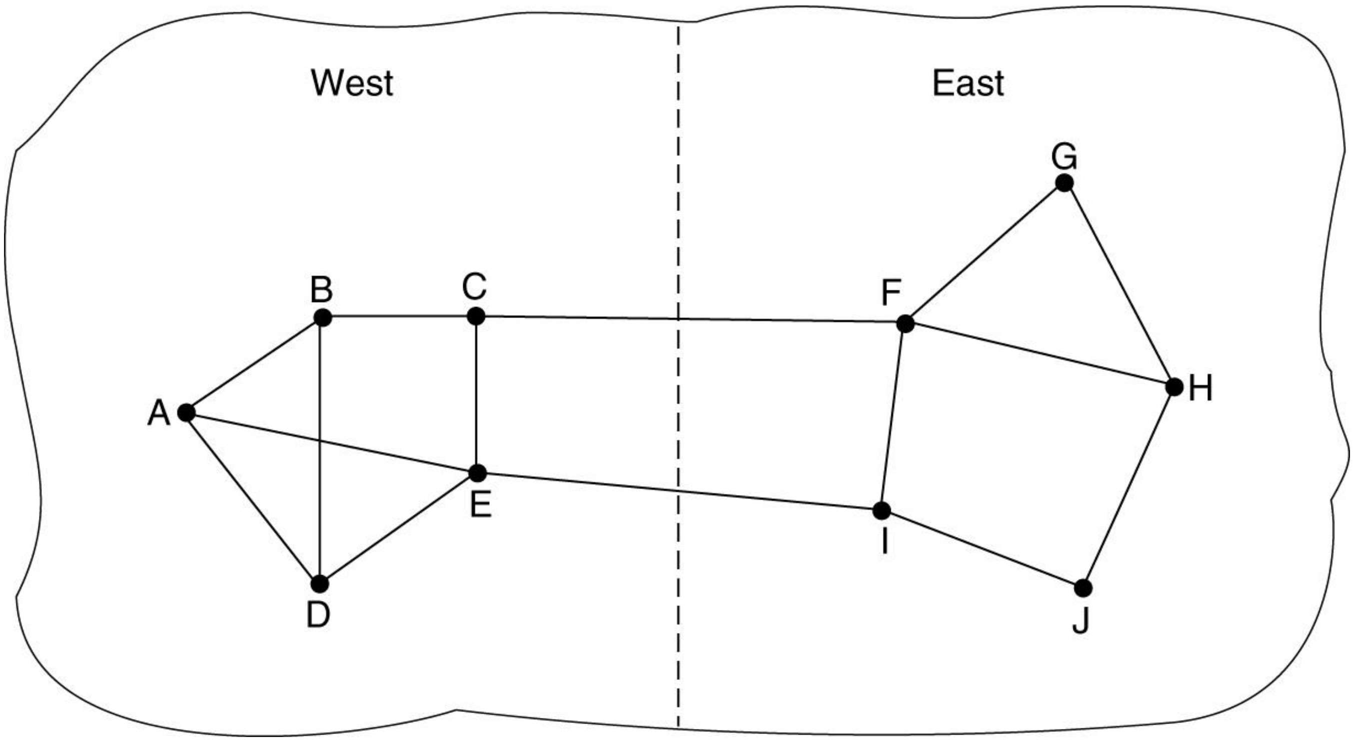
Consider the situation where all nodes are sending to destination A.

Each node must determine to either route clockwise or counter clockwise.



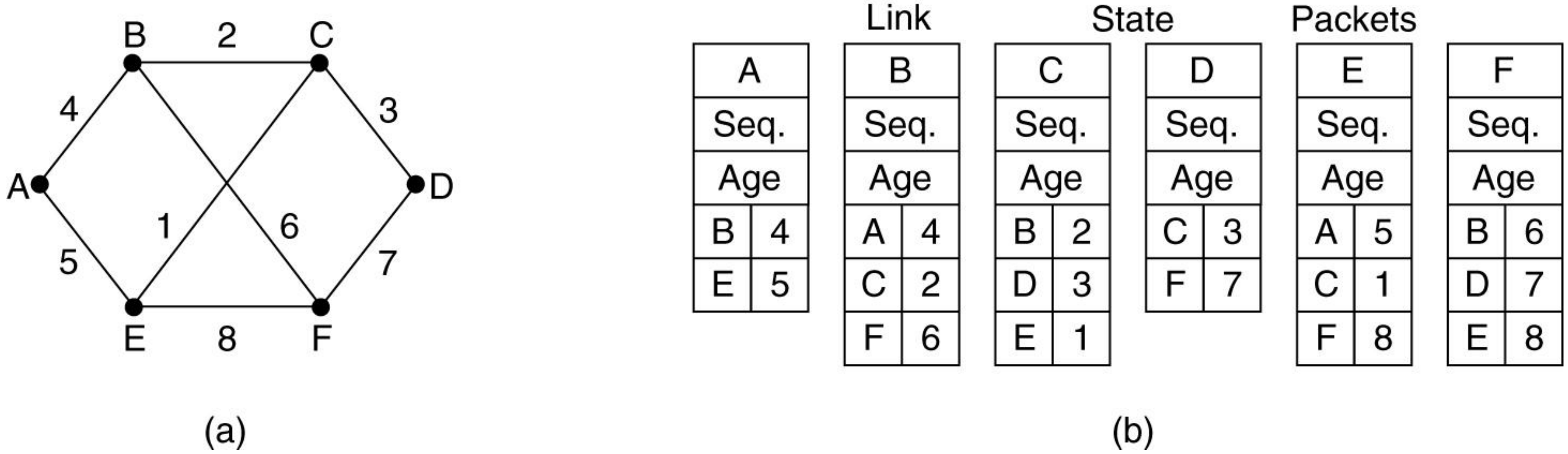
The cost of routing clockwise is the number of other nodes routing clockwise.

Measuring Line Cost



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

Building Link State Packets



Link State Packets

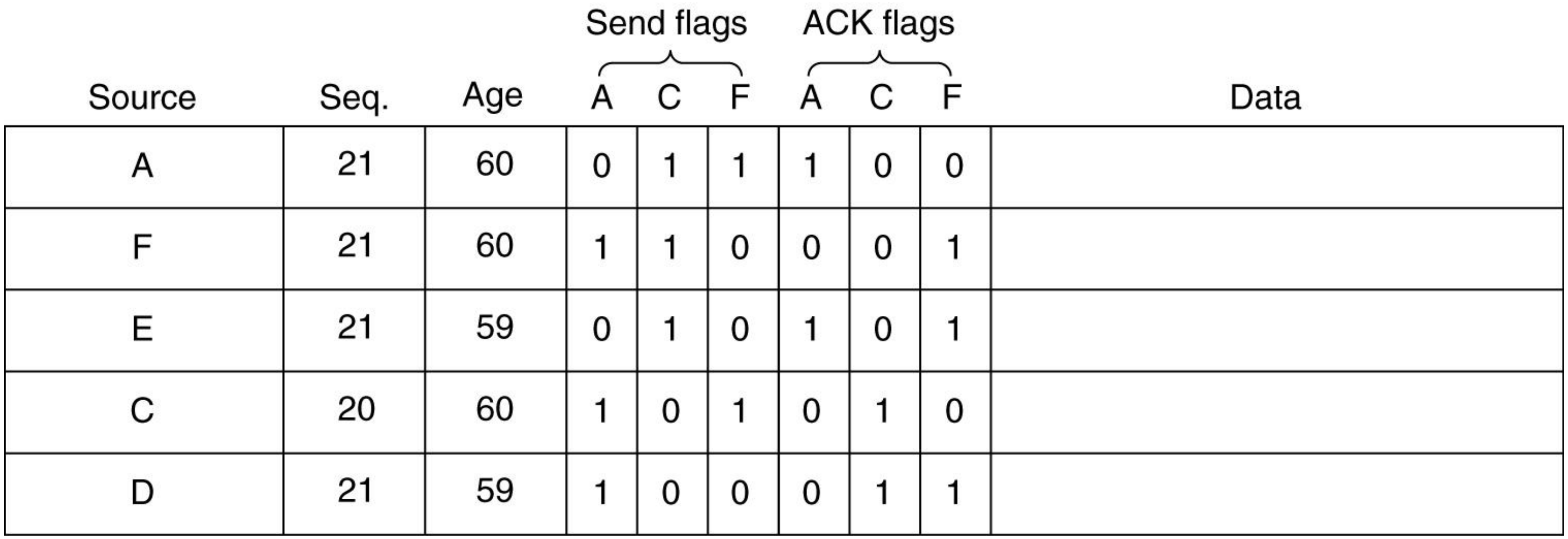
A		B		C		D		E		F	
Seq.		Seq.		Seq.		Seq.		Seq.		Seq.	
Age		Age		Age		Age		Age		Age	
B	4	A	4	B	2	C	3	A	5	B	6
E	5	C	2	D	3	F	7	C	1	D	7
		F	6	E	1			F	8	E	8

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

Distributing the Link State Packets

- Use selective flooding
- Sequence numbers prevent duplicate packets from being propagated
- Lower sequence numbers are rejected as obsolete
- This algorithm has few problems, but manageable.
- First: If the sequence number wrap around, (use 32-bit Sequence Number, one link per second, it would take 137 years to wrap around.).
- Second: If a router ever crashes, it will lose track of its Sequence Number. If its starts again at 0 , the next packet will be rejected as a duplicate.
- Third: if a sequence number is ever corrupted.
- The solution to all these problems is to indicate age of each packet after the sequence number and decrement it once per second. When the age hits zero , the information from the router is discarded.

Distributing the Link State Packets



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

The packet buffer for router B in the previous slide (Fig. 5-13).

Computing the New Routes

- Dijkstra’s Shortest Path algorithm is used to determine the shortest path to each destination.

Hierarchical Routing

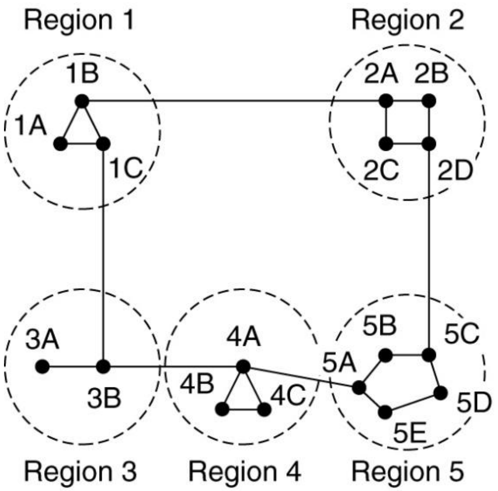
- Addresses the growth of routing tables
- Routers are divided into regions
- Routers know the routes for their own regions only
- Works like telephone routing
- Possible hierarchy

  a) city, state, country, continent

- Optimal number of levels for an N router subnet is lnN

- It may be required to group the
- Regions into clusters,
- Clusters into zones,
- Zones into groups and so on…

Hierarchical Routing



(a)

Full table for 1A

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

(b)

Hierarchical table for 1A

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

(c)

Hierarchical routing.

Hierarchical Routing

- The full routing table for router 1A has 17 entries.
- When routing has done hierarchically, there are 7-entries.

When a single network becomes very large,
Ex: consider a subnet with 720-routers.

- If there is no hierarchy, each router should need 720-routing table entries.
- If subnet is partitioned into 24-regions of 30-routers each, two level hierarchy. Each router needs 30-local entries and 23-remote entries for a total of 53-entries.

- If three-level hierarchy is chosen, with 8-clusters, each containing 9-regions of 10-routers each.
- Each router needs 10-entries for local router, 8-entries for routing to other regons within its own cluster, and 7-entries for distant clusters. For a total of (10+8+7) 25 entries.

Broadcast Routing

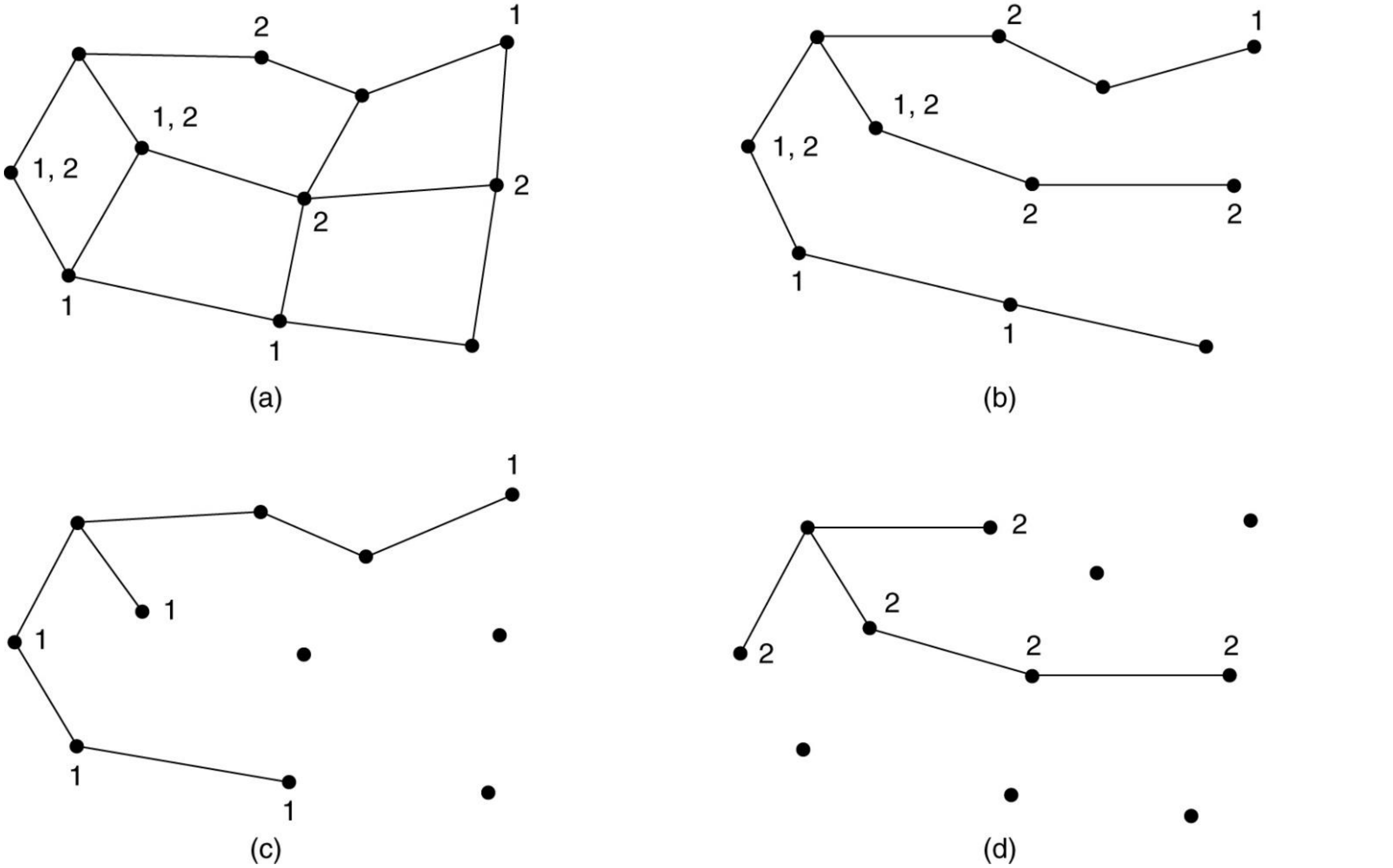
- Sending a packet to all destinations simultaneously is called Broadcasting.

- Many methods are there:

- 1. sourse to simply send a packet to each destination.
- 2. flooding is another method.

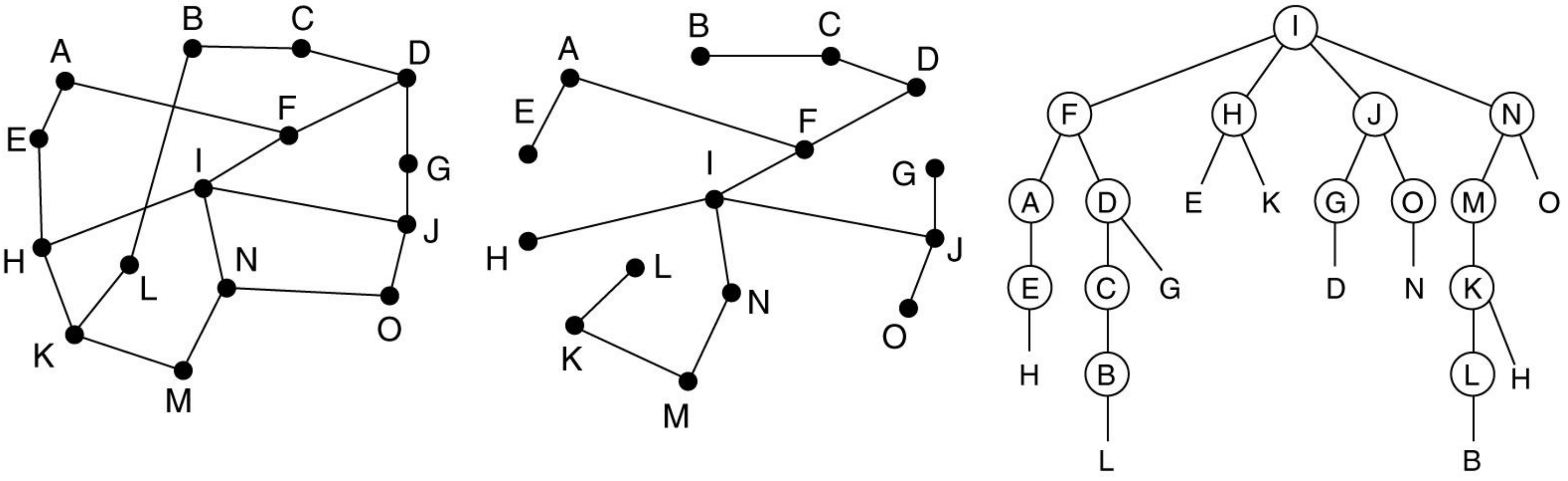
- It generates too many packets and consumes too much space.

Multicast Routing



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

Multi destination Routing



(a) (b) (c)

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

Routing for Mobile Hosts

- Millions of people use computers while on go, from the truly mobile situations with a wireless device in moving cars, to nomadic situations in which laptop computers are used in a series of a different location.

- We use the term mobile hosts to mean either category, as distinct from stationary hosts that never move.

- The mobile hosts introduce a new complication to route packets to the mobile hosts, the network first has to find it.

Routing for Mobile Hosts….

Assumed model :

- The model of the world that we will consider is one in which all hosts are assumed to have a permanent home location that never changes.

- Each host has a permanent home address that can be used to determine home location.

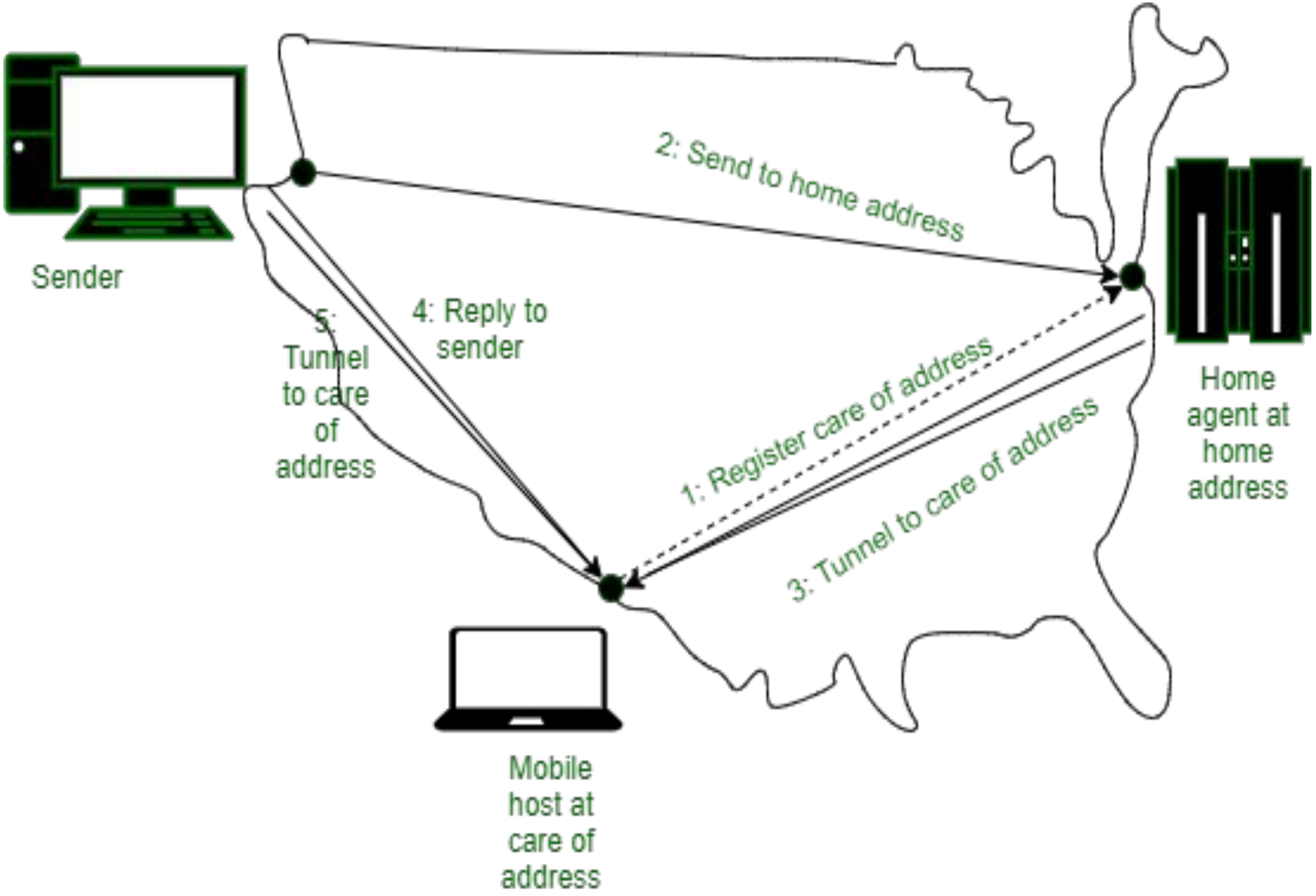
- Like the telephone number 1-212-5551212 indicates the United States (country code 1) and Manhattan (212).

Routing for Mobile Hosts….

Features :

- The basic idea used for mobile routing on the internet and cellular network is for the mobile hosts to tell the host at the home location.
- This host, which acts on behalf of the mobile host called a home agent.
- Once it knows where the mobile host currently located, it can forward packets so that they are delivered. The figure shows mobile routing in action.
- The local address called care-of address.
- Once it has this address it can tell its home agent where it is now. It does this by sending registration message to home agent with care of address.

Routing for Mobile Hosts….


Sender
2: Send to home address
4: Reply to sender
5: Tunnel to care of address
1: Register care of address
3: Tunnel to care of address
Home agent at home address
Mobile host at care of address

Description of Diagram :

- The message is shown with a dashed line in the figure indicate that it is a control message, not a data message.

- The sender sends a data packet to the mobile host using its permanent address.

- This packet is routed by the network to the host home location because the home addresses belong there.

- It encapsulates the packet with a new header and sends this bundle to the care-of address.

- This mechanism is called tunneling.

- It is very important on the internet, so we will look at it in more detail later.

Description of Diagram :

- When the encapsulated packet arrives at the care-of address, the mobile host unwraps it and retrieves the packet from the sender.

- The overall route is called triangle routing because it way is circuitous if the remote location is far from the home location.

- As part of the step, 4 senders learns the current care-of address.

- Subsequent packets can be routed directly to the mobile host by tunneling them to the care-of address (step 5) bypassing the home location.

- If connectivity lost for any reason as the mobile moves, the home address can always be used to reach the mobile.

Routing in Ad Hoc Networks

- **Challenges:**

  – Dynamic topology

  – Unreliable links

  – Limited resources (battery, processing power)

  – Low link bandwidth

  – Security

  – No default router available

Routing in Ad Hoc Networks

No physical links:

– Wireless links created and destroyed as nodes move

– Frequent disconnections and partitions

Congestion control algorithms

- Congestion
- Congestion control
- Causes of congestion
- Principles of congestion control
- Approaches to Congestion Control
- Traffic-Aware Routing
- Admission Control
- Traffic Throttling
- Load Shedding

Congestion control algorithms….

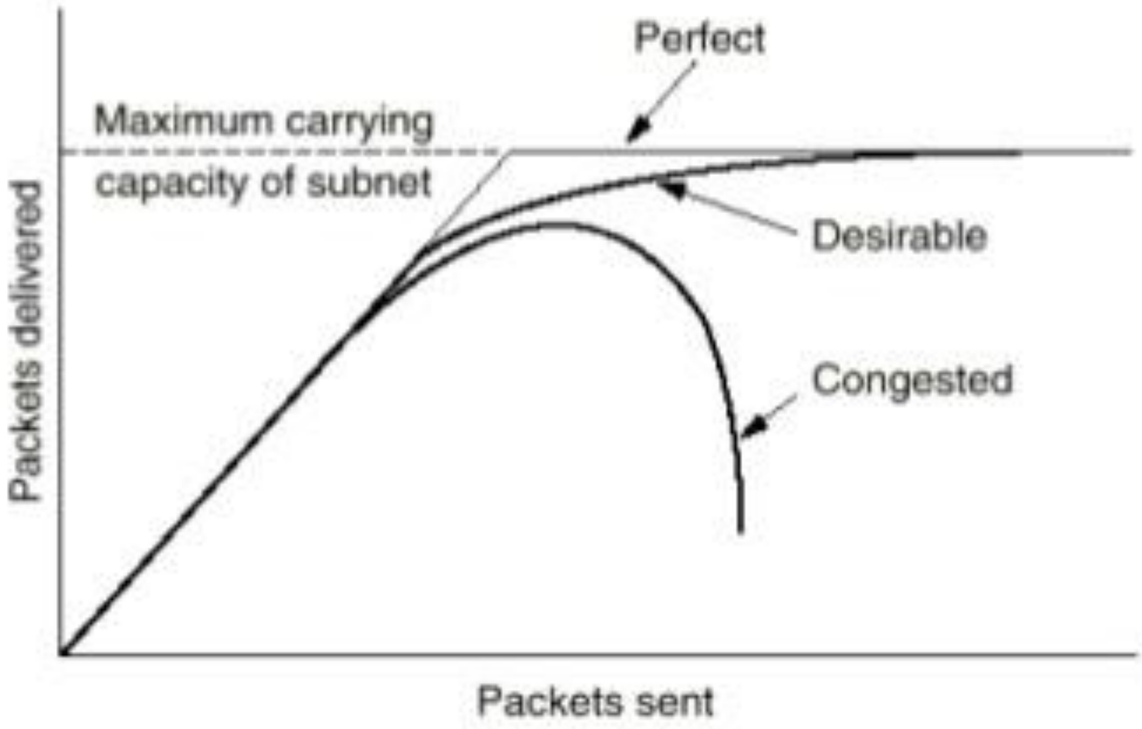
Congestion:

“ Congestion in a network may occur if the load on the network-the number of packets sent to the network-is greater than the capacity of the network-the number of packets that the network can handle.”

Congestion control:

Refers to the mechanisms and techniques to control the congestion and keep the load below the capacity.

Congestion control algorithms….



When too much traffic is offered, congestion sets in and performance degrades sharply.

Congestion control algorithms….

Causes of congestion:

Congestion occurs when a router receives data faster than it can send it.

Insufficient bandwidth
Slow hosts
Data simultaneously arriving from multiple lines destined for the same outgoing line.

The system is not balanced.

Correcting the problem at one router will probably just move the bottleneck to another router.

Congestion control algorithms….

Congestion Control vs Flow Control:

Congestion Control :

Controls the traffic throughout the network.

Flow Control:

Controls point to point traffic between sender and receiver.
eg: A fast host sending to a slow host.

General Principles of Congestion Control

1. Monitor the system
   a) detect when and where congestion occurs.
2. Pass information to where action can be taken.
3. Adjust system operation to correct the problem.

Approaches to Congestion Control

Two solutions possible:

1)Increase resources

2)Decrease load

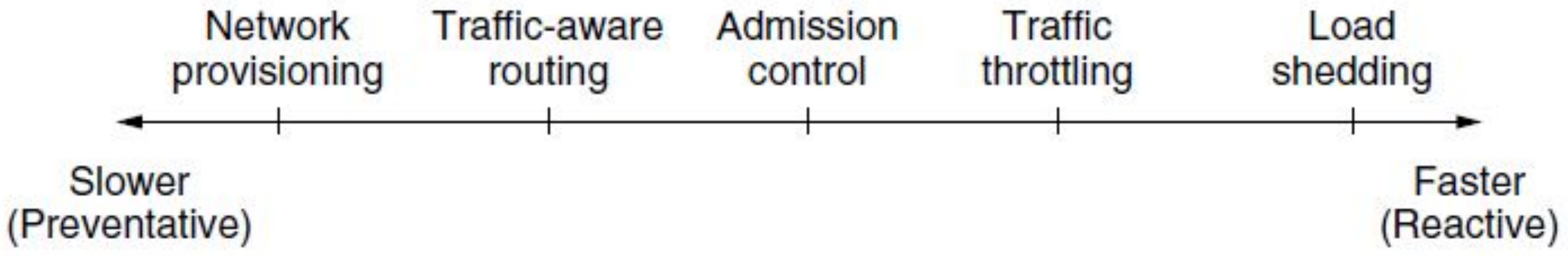
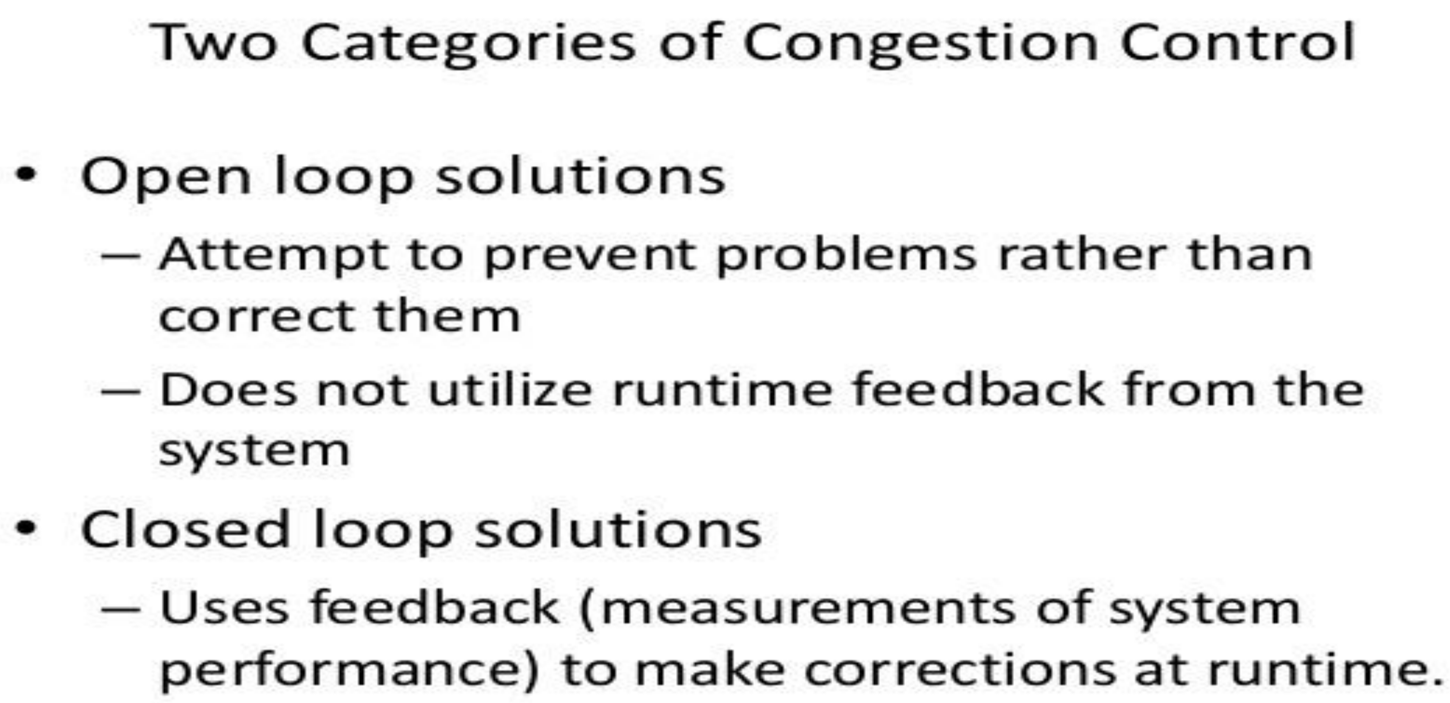


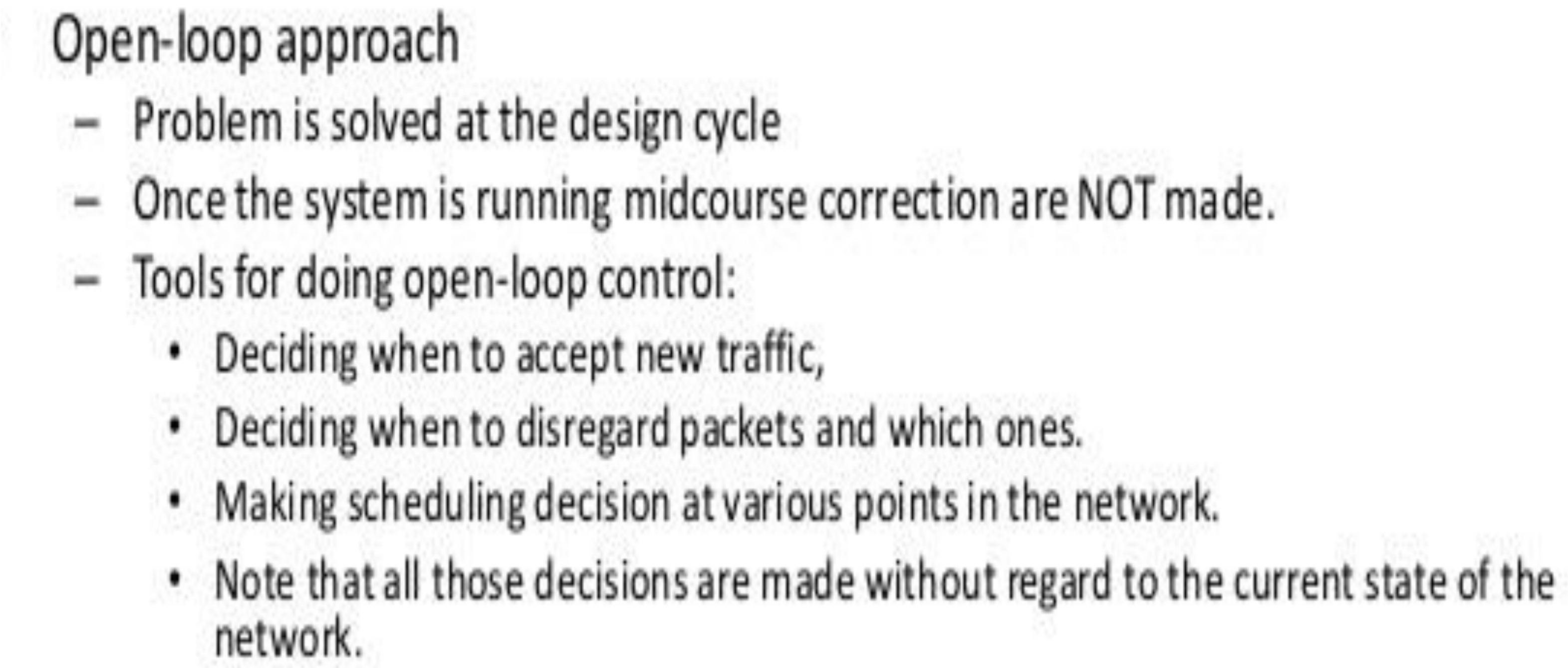
Fig:Timescales of approaches to congestion control



Two Categories of Congestion Control

- Open loop solutions
  - Attempt to prevent problems rather than correct them
  - Does not utilize runtime feedback from the system
- Closed loop solutions
  - Uses feedback (measurements of system performance) to make corrections at runtime.

Open loop: Prevent congestion before it happens
Closed loop: Remove congestion after it happens

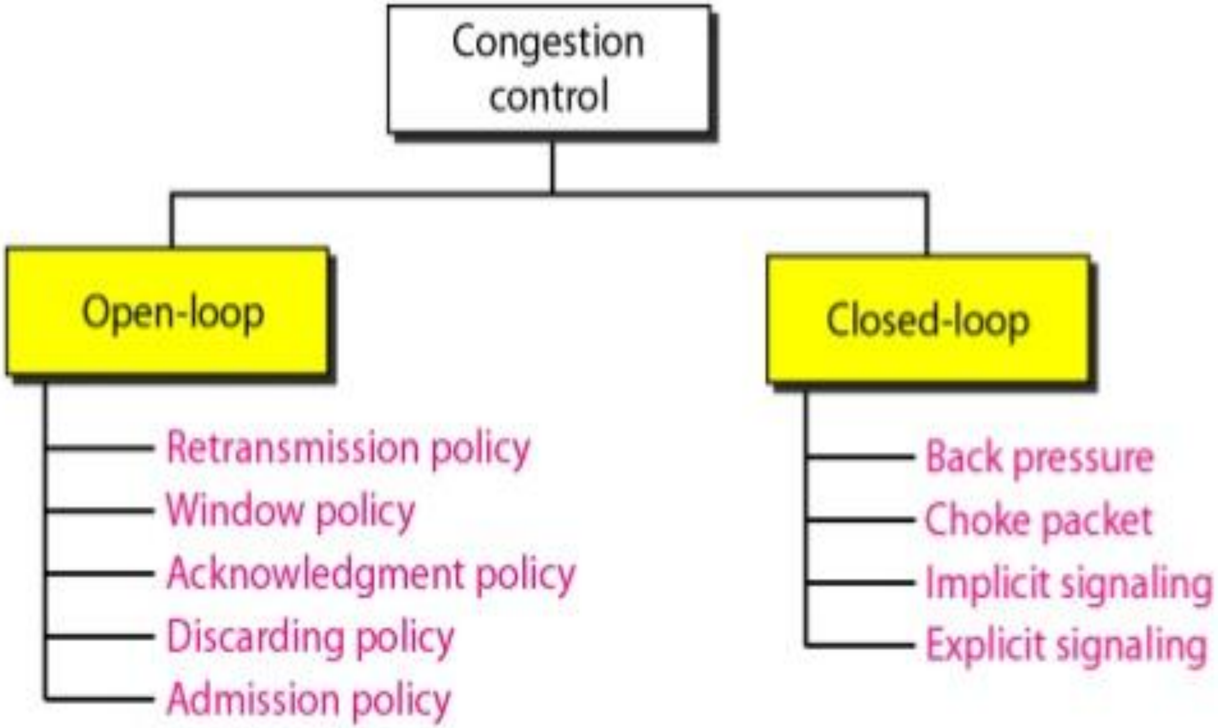


Open-loop approach

- Problem is solved at the design cycle
- Once the system is running midcourse correction are NOT made.
- Tools for doing open-loop control:
  - Deciding when to accept new traffic,
  - Deciding when to disregard packets and which ones.
  - Making scheduling decision at various points in the network.
  - Note that all those decisions are made without regard to the current state of the network.

Closed-loop approach

- It is based on the principle of feedback-loop. The approach has three parts when applied to congestion control:
  1. Monitor the system to detect when and where congestion occurs,
  2. Pass this information tot places where action can be taken
  3. Adjust system operation to correct the problem.



Congestion control

Open-loop

- Retransmission policy
- Window policy
- Acknowledgment policy
- Discarding policy
- Admission policy

Closed-loop

- Back pressure
- Choke packet
- Implicit signaling
- Explicit signaling

Retransmission Policy:
Packet can be retransmit to the source again.

Window policy:

- Selective-reject window method.

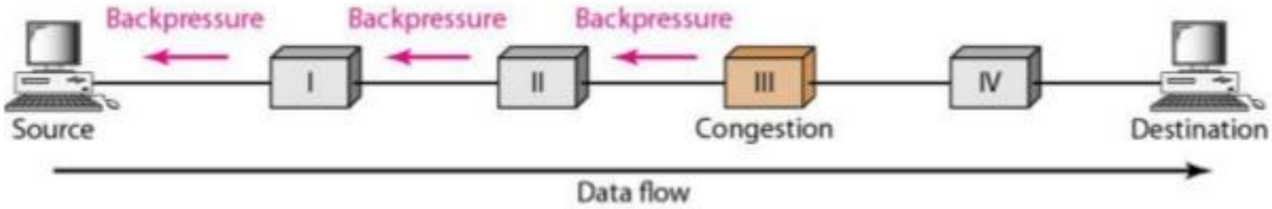
Acknowledgement policy : Receiver sends the acknowledgment.

Discarding Policy: Router discards less sensitive packets when congestion likely to happened.

Admission policy: Quality of service mechanisms.

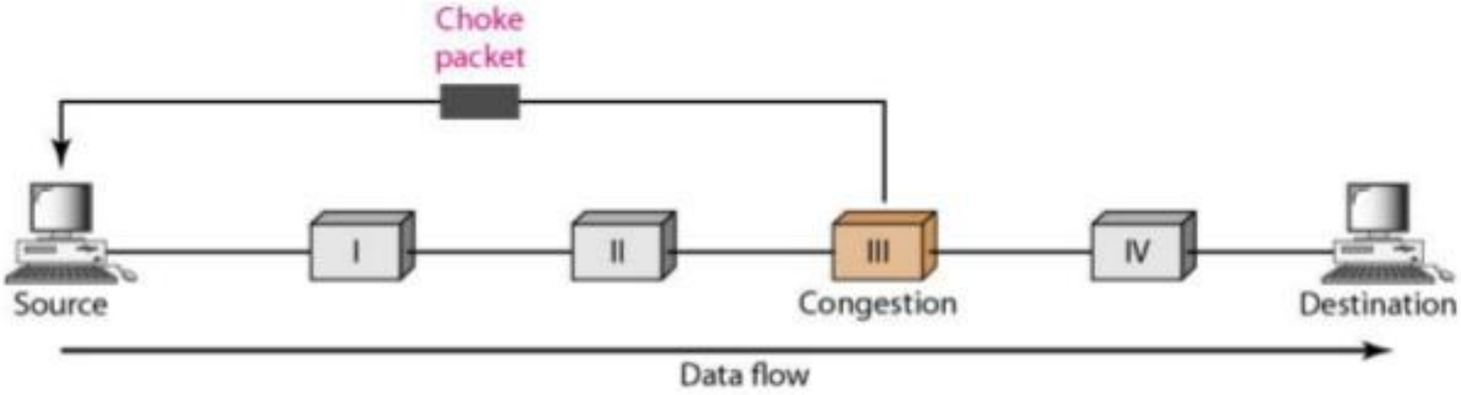
Warning Bit/ Backpressure

- A special bit in the packet header is set by the router to warn the source when congestion is detected.
- The bit is copied and piggy-backed on the ACK and sent to the sender.
- The sender monitors the number of ACK packets it receives with the warning bit set and adjusts its transmission rate accordingly.



Choke Packets

- A more direct way of telling the source to slow down.
- A choke packet is a control packet generated at a congested node and transmitted to restrict traffic flow.



Implicit Signaling:

- Source guesses there is a congestion in network when it does not receive any acknowledgment.
- Source becomes slow down.

Explicit Signaling:

- Congestion node sending direct signal to source or destination.
- Differ from the choke packet.
- Forward or backward direction.

Quality of Service Mechanism:

Flow characteristics:

Reliability—

Delay—

Jitter—

Bandwidth—

Improve Quality of service:

Scheduling—FIFO queuing, Priority queuing, Weighted fair queuing.

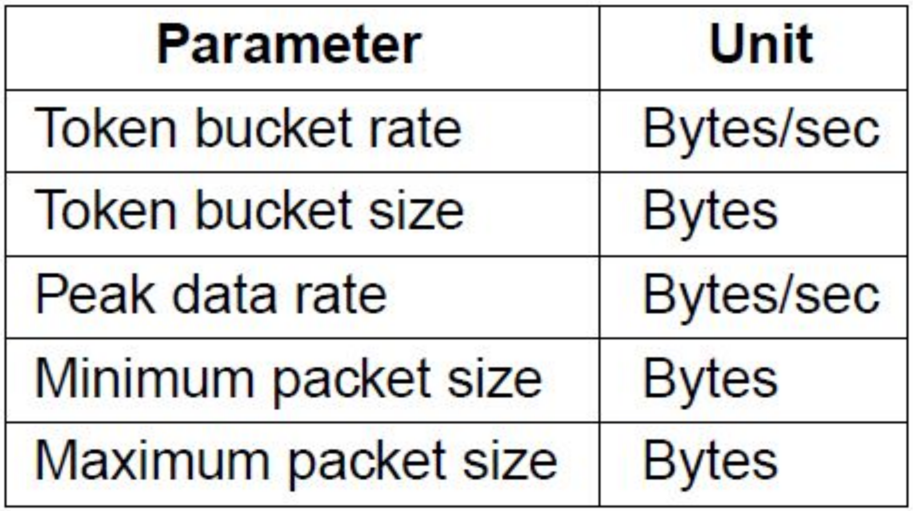
Traffic Shaping— Leaky bucket, Token Bucket.

Resource reservation– Resources are reserved to improve the QoS— Buffer, bandwidth, CPU Time and so on..

Admission Control– Mechanism used by router or switch.

- ❖Accept or reject a flow based on predefined parameters called flow specifications.
- ❖Before a router accepts a flow for processing, It checks the flow specifications(bandwidth, buffer size,CPU speed) and its previous communications to other flows can handle the new flow before admission of data.

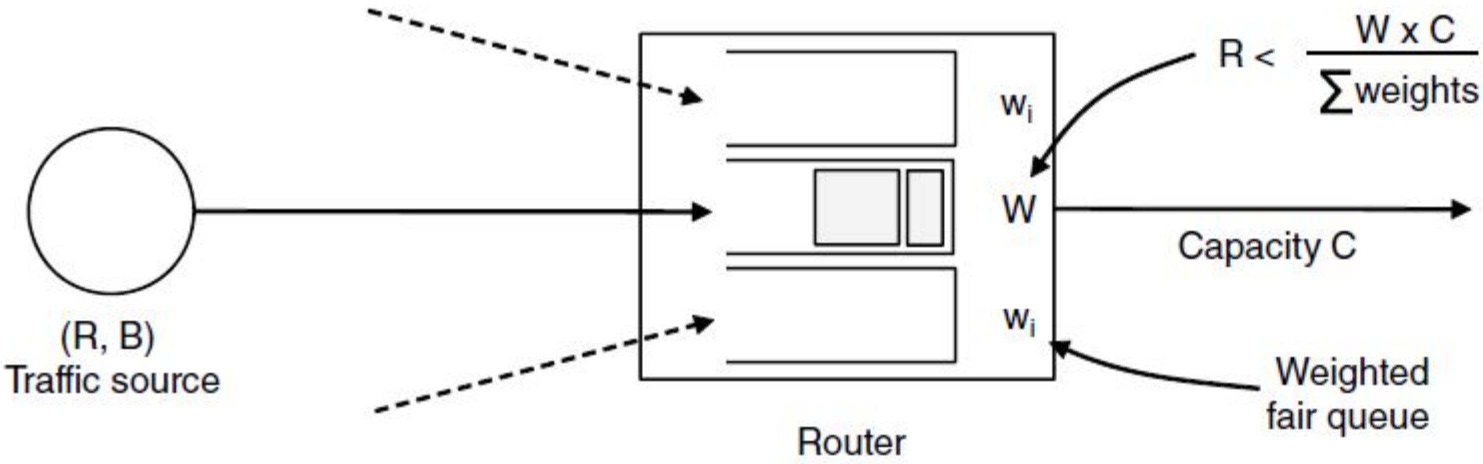
Admission Control (1)



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

An example flow specification

Admission Control (2)



Bandwidth and delay guarantees with token buckets and WFQ.

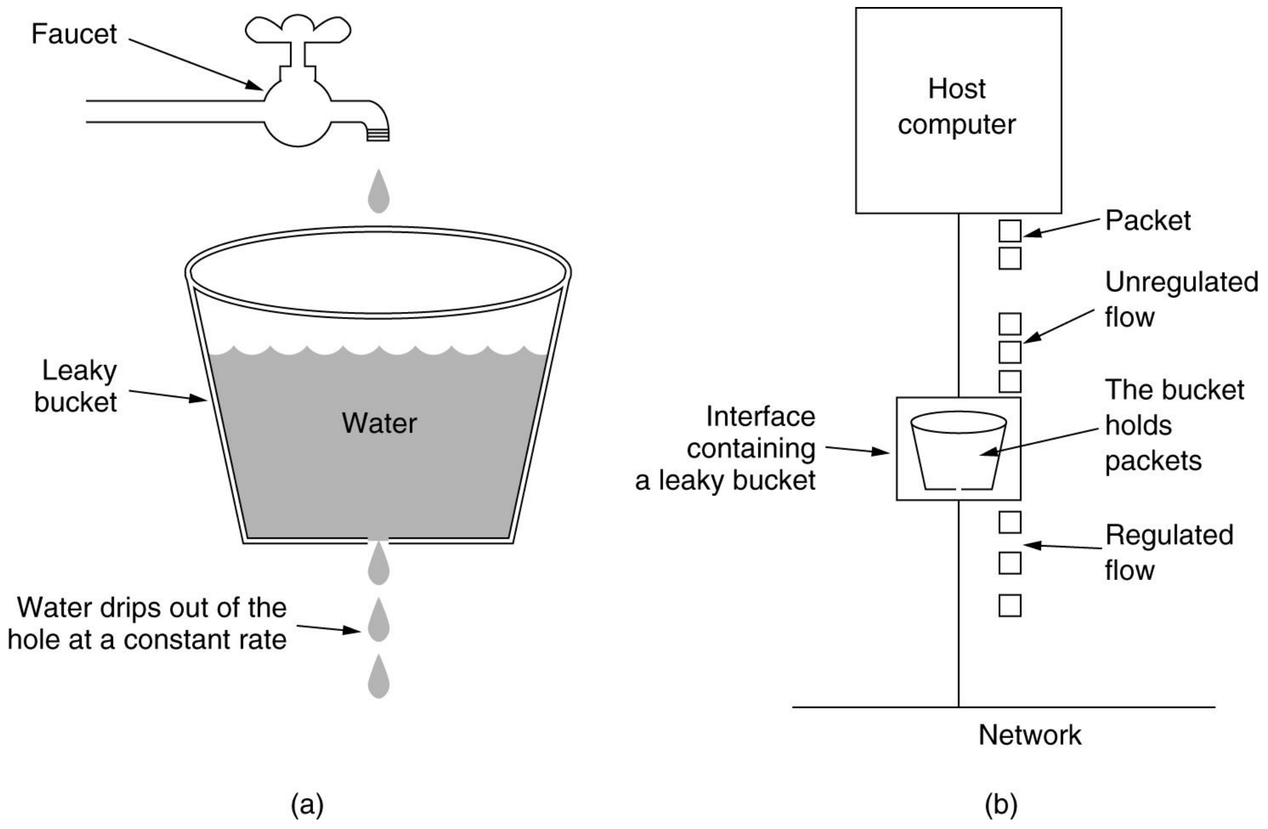
Traffic Shaping

- Traffic in data networks is **bursty** – typically arrives at non-uniform rates as the traffic rate varies.
- **Traffic shaping** is a technique for regulating the average rate and burstiness of a flow of data that enters the network.
- When a flow is set up, the user and the network agree on a certain traffic pattern (shape).
- Sometimes this agreement is called an **SLA** (**Service Level Agreement**).

Traffic Shaping

- ❖Leaky Bucket
- ❖Token Bucket

The Leaky Bucket Algorithm



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

Leaky bucket

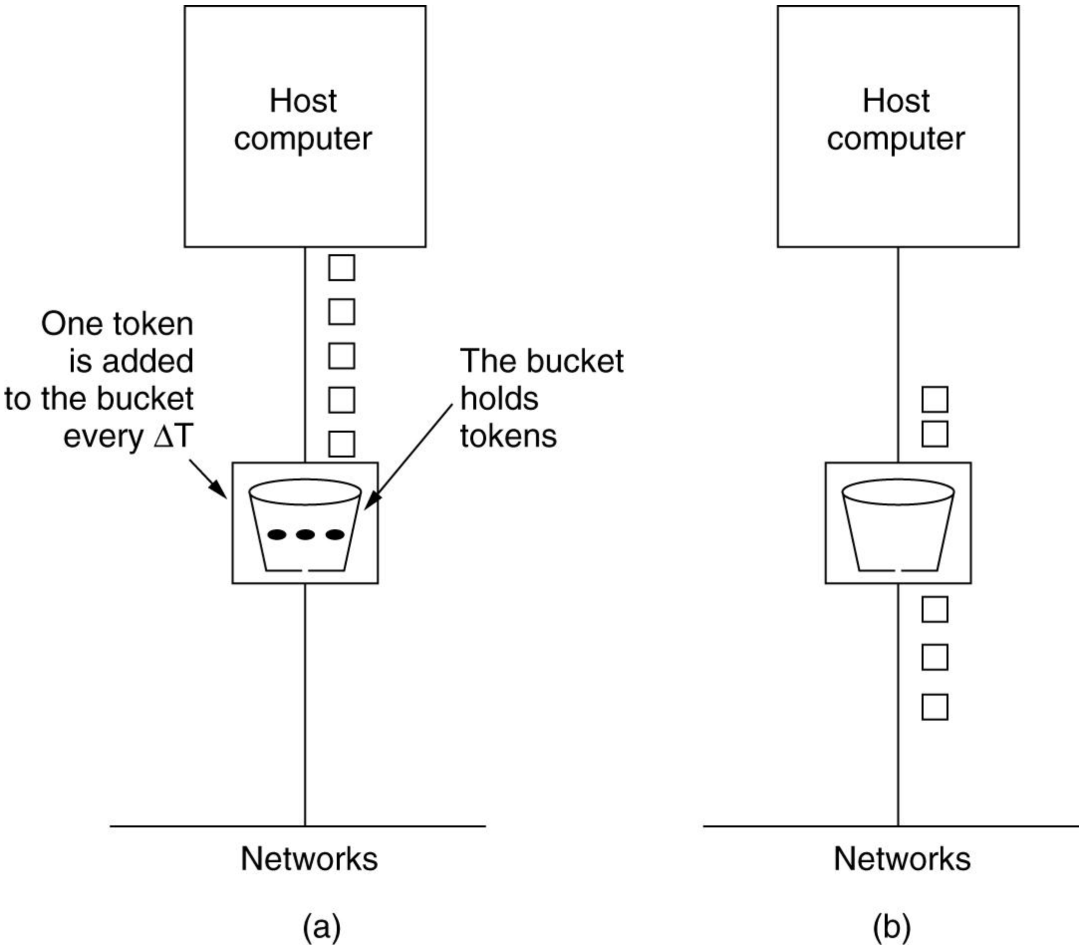
Steps:

- When host wants to send a packet, packet is thrown into the bucket.

- The bucket leaks at a constant rate, means the network interface transmits the packet at a constant rate.

- Bursty traffic is converted to a uniform packet by a leaky bucket.

The Token Bucket Algorithm



(a) Before. (b) After.

Token Bucket Algorithm...

Need of token bucket Algorithm:-

- The leaky bucket algorithm enforces output pattern at the average rate, no matter how bursty the traffic is.

- So in order to deal with the bursty traffic we need a flexible algorithm so that the data is not lost.

- One such algorithm is token bucket algorithm.

Token Bucket Algorithm...

Steps: of this algorithm can be described as follows:

In regular intervals tokens are thrown into the bucket. Ƒ

The bucket has a maximum capacity. Ƒ

If there is a ready packet, a token is removed from the bucket, and the packet is sent.

If there is no token in the bucket, the packet cannot be sent.

Token Bucket

- Allow the output is vary depending on the size of the burst.

- In this algorithm the bucket holds token to transmits the packet, the host must capture and destroy one token.

- Tokens are generated by a clock at the rate of one token every sec.

- Idle hosts can capture and save up tokens(up to the max. size of the bucket) in order to send larger bursts later.

Token Bucket Algorithm

- Burst length – *S* sec.
- Maximum output rate – *M* bytes/sec
- Token bucket capacity – *B* bytes
- Token arrival rate – *R* bytes/sec

- An output burst contains a maximum of **(*B* + *RS*)** bytes.
- The number of bytes in a maximum speed burst of length ***S*** seconds is ***MS***.
- Hence, we have: $\boldsymbol{B + RS = MS}$
- This equation can be solved to get $\boldsymbol{S = B/(M - R)}$

Internetworking

❖It is the practice of interconnecting multiple computer networks.

❖Routing between two networks is called internetworking.

❖Same networks or different networks.

❖Networks can be considered different based on various parameters such as,

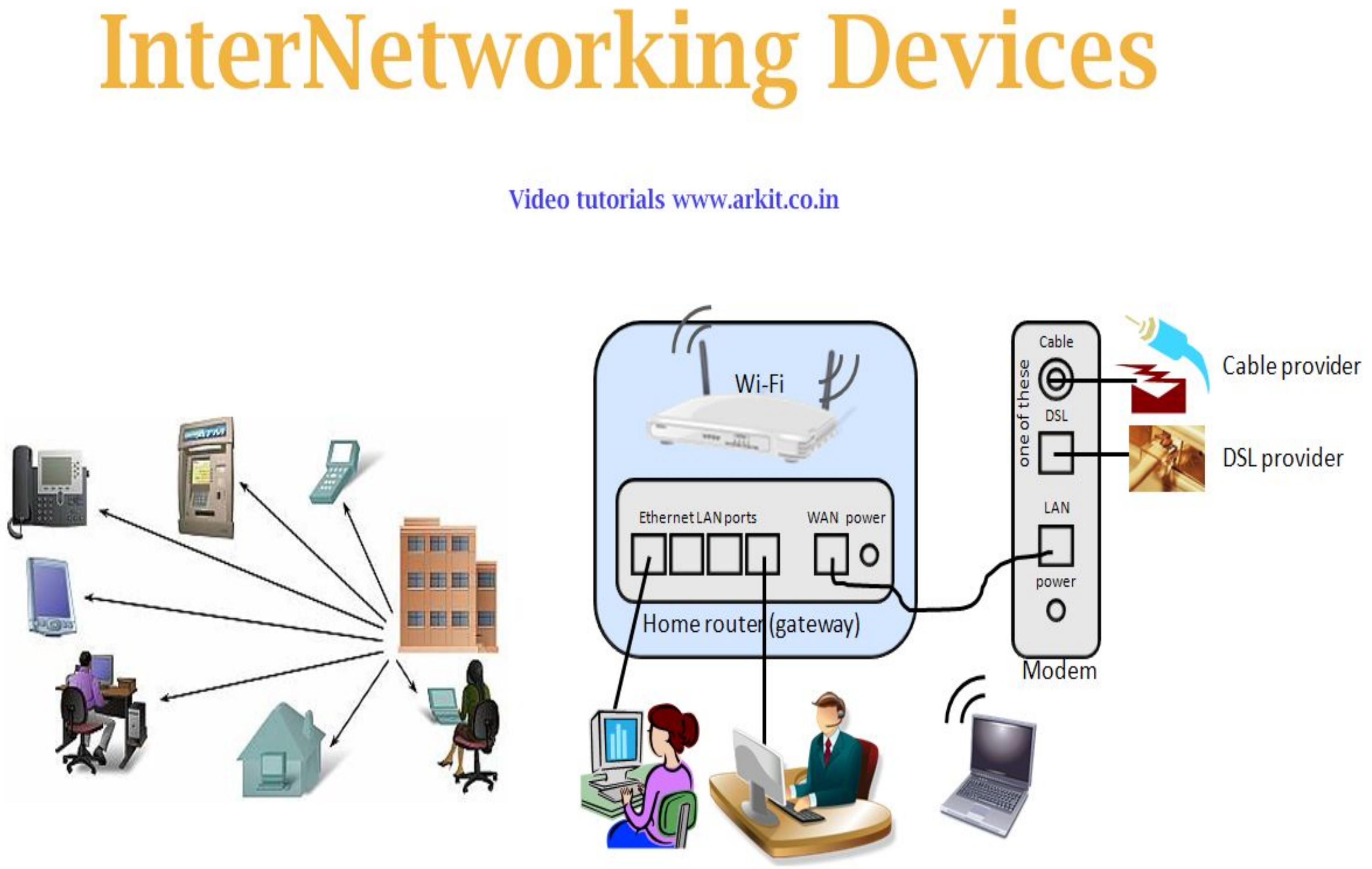
- Protocol
- Topology
- Addressing scheme

In Internetworking, routers have knowledge of each other’s address beyond them.

□They can be statically configured go on different network or they can learn by using internetworking routing protocols.

- Tunneling
- Internetworking Routing Protocols

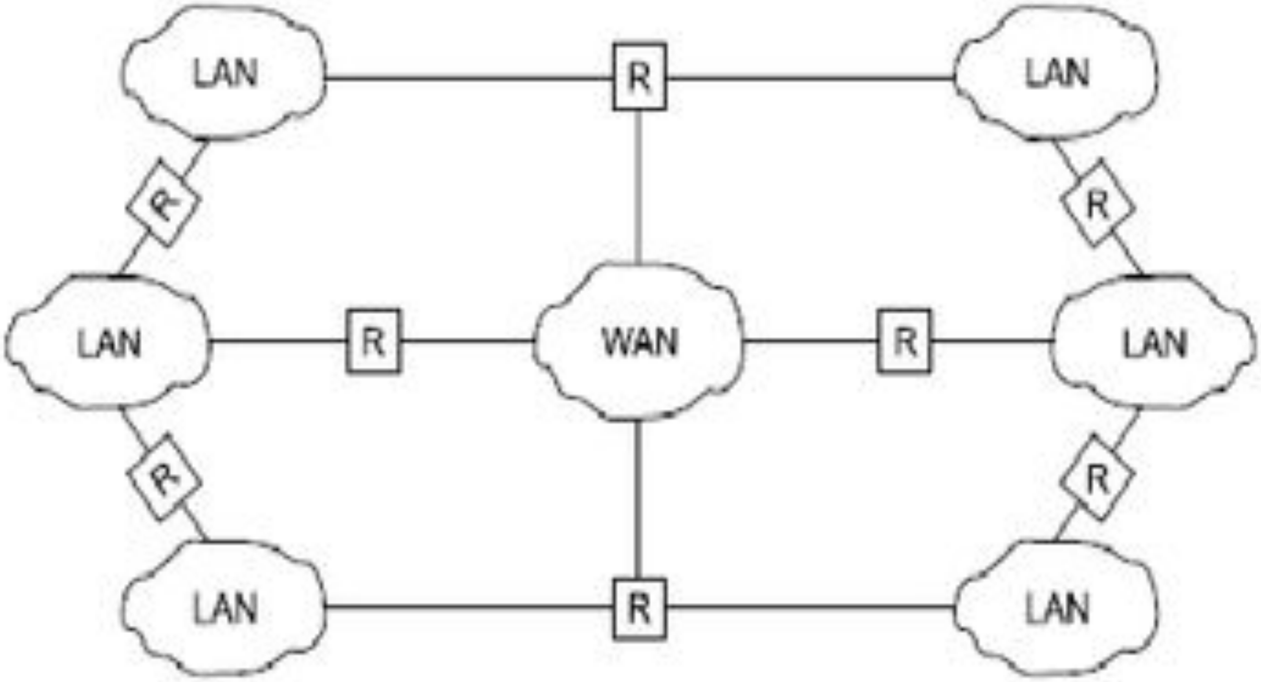
Internetworking



InterNetworking Devices

Video tutorials www.arkit.co.in

Internetworking



Network Layer in the Internet

❖IPV4

❖IPV6

What is an IP?

 An IP stands for internet protocol. An IP address is assigned to each device connected to a network.

Each device uses an IP address for communication.

It also behaves as an identifier as this address is used to identify the device on a network.

Network Layer in the Internet…

What is IPv4?

❖IPv4 is a version 4 of IP.

❖It is a current version and the most commonly used IP address.

❖It is a 32-bit address written in four numbers separated by 'dot', i.e., periods.

❖This address is unique for each device.

Network Layer in the Internet

An IP address consists of two parts.

- ❖ The first one is a network address.
- ❖ The other one is a host address.

Eg: 66.94. 29.13

Network Part

This part specifies the unique number assigned to your network. It also identifies the class of network assigned. In Fig, the network part takes up two bytes of the IP address.

Host Part

This is the part of the IP address that you assign to each host. It uniquely identifies this machine on your network. Note that for each host on your network, the network part of the address will be the same, but the host part must be different.

Network Layer in the Internet…

- ❖The above example represents the IP address in which each group of numbers separated by periods is called an **Octet**.

- ❖Each number in an octet is in the range from 0-255.

- ❖This address can produce 4,294,967,296 possible unique addresses.

- ❖In today's computer network world, computers do not understand the IP addresses in the standard numeric format as the computers understand the numbers in binary form only.

- ❖The binary number can be either 1 or 0.

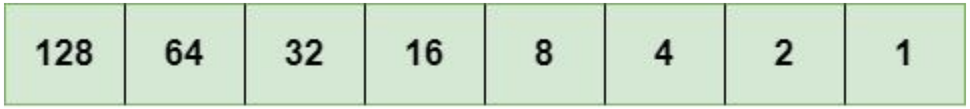
- ❖The IPv4 consists of four sets, and these sets represent the octet.

- ❖The bits in each octet represent a number.

- ❖Each bit in an octet can be either 1 or 0. If the bit is the 1, then the number it represents will count, and if the bit is 0, then the number it represents does not count.

Network Layer in the Internet…

Representation of 8 Bit Octet

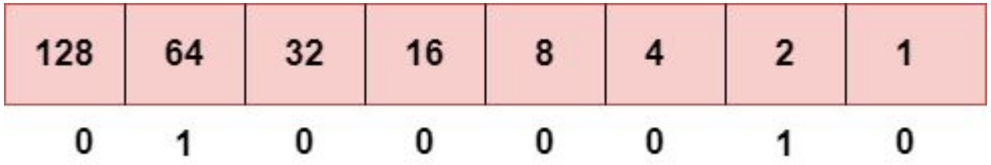


128	64	32	16	8	4	2	1

The above representation shows the structure of 8- bit octet.

Now, we will see how to obtain the binary representation of the above IP address, i.e., 66.94.29.13

Step 1: First, we find the binary number of 66.



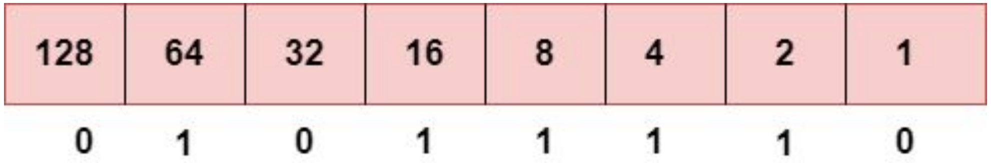
128	64	32	16	8	4	2	1
0	1	0	0	0	0	1	0

Network Layer in the Internet…

To obtain 66, we put 1 under 64 and 2 as the sum of 64 and 2 is equal to 66 (64+2=66), and the remaining bits will be zero, as shown above.

Therefore, the binary bit version of 66 is 01000010.

Step 2: Now, we calculate the binary number of 94.



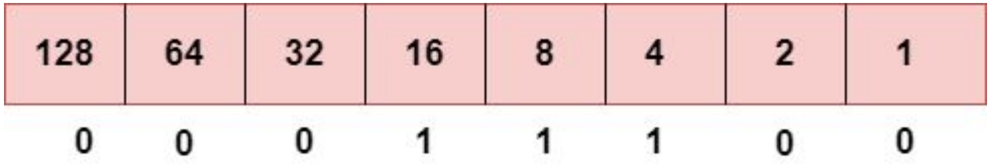
128	64	32	16	8	4	2	1
0	1	0	1	1	1	1	0

To obtain 94, we put 1 under 64, 16, 8, 4, and 2 as the sum of these numbers is equal to 94, and the remaining bits will be zero.

Therefore, the binary bit version of 94 is 01011110.

Network Layer in the Internet…

Step 3: The next number is 29.

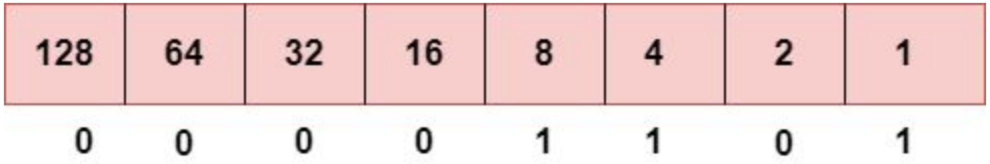


128	64	32	16	8	4	2	1
0	0	0	1	1	1	0	0

To obtain 29, we put 1 under 16, 8, 4, and 1 as the sum of these numbers is equal to 29, and the remaining bits will be zero.

Therefore, the binary bit version of 29 is 00011101.

Step 4: The last number is 13.



128	64	32	16	8	4	2	1
0	0	0	0	1	1	0	1

To obtain 13, we put 1 under 8, 4, and 1 as the sum of these numbers is equal to 13, and the remaining bits will be zero.

Therefore, the binary bit version of 13 is 00001101.

Network Layer in the Internet…

Drawback of IPv4

IPv4 produces 4 billion addresses, which are not enough for each device connected to the internet on a planet.

so it gave rise to the development of the next generation of IP addresses, i.e., IPv6.

What is IPv6?

❖IPv4 produces 4 billion addresses, and the developers think that these addresses are enough, but they were wrong.
❖IPv6 is the next generation of IP addresses.
❖The main difference between IPv4 and IPv6 is the address size of IP addresses.
❖The IPv4 is a 32-bit address, whereas IPv6 is a 128-bit hexadecimal address.
❖IPv6 provides a large address space, and it contains a simple header as compared to IPv4.

Network Layer in the Internet…

It provides transition strategies that convert IPv4 into IPv6, and these strategies are as follows:

Dual stacking: It allows us to have both the versions, i.e., IPv4 and IPv6, on the same device.

Tunneling: In this approach, all the users have IPv6 communicates with an IPv4 network to reach IPv6.

Network Address Translation: The translation allows the communication between the hosts having a different version of IP.

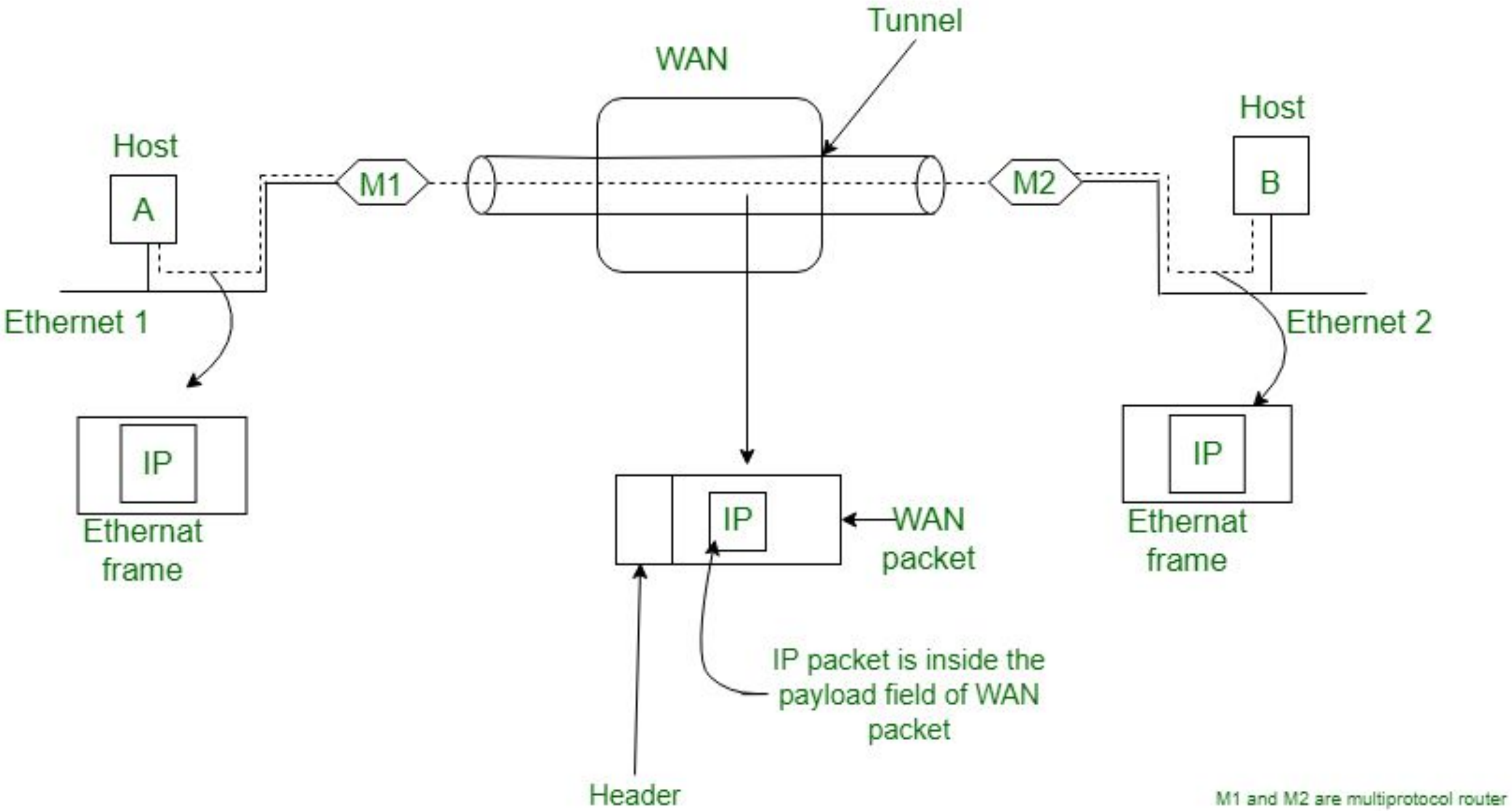
Network Layer in the Internet…

Tunneling:

A technique of internetworking called **Tunneling** is used when source and destination networks of same type are to be connected through a network of different type.

let us consider an Ethernet to be connected to another Ethernet through a WAN as:

Network Layer in the Internet…



Tunneling

Network Layer in the Internet…

Tunneling:

In this particular example, the IP packet does not have to deal with WAN. the host A and B also do not have to deal with the WAN. The multiprotocol routers M1 and M2 will have to understand about IP and WAN packets. Therefore, the WAN can be imagined to be equivalent to a big tunnel extending between multiprotocol routers M1 and M2 and the technique is called Tunneling

Network Layer in the Internet…

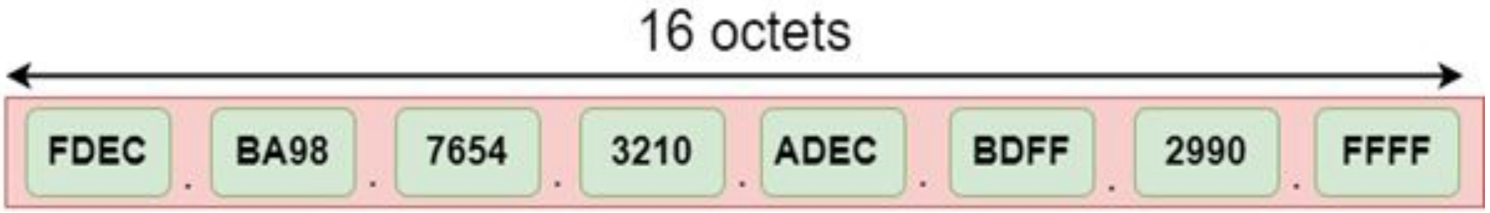
- ❖This hexadecimal address contains both numbers and alphabets.
- ❖Due to the usage of both the numbers and alphabets, IPv6 is capable of producing over 340 undecillion ($3.4*10^{38}$) addresses.
- ❖IPv6 is a 128-bit hexadecimal address made up of 8 sets of 16 bits each, and these 8 sets are separated by a colon or dot.

The address format of IPv4:

192 168 2 33□4 Octets

Network Layer in the Internet…

The address format of IPv6:

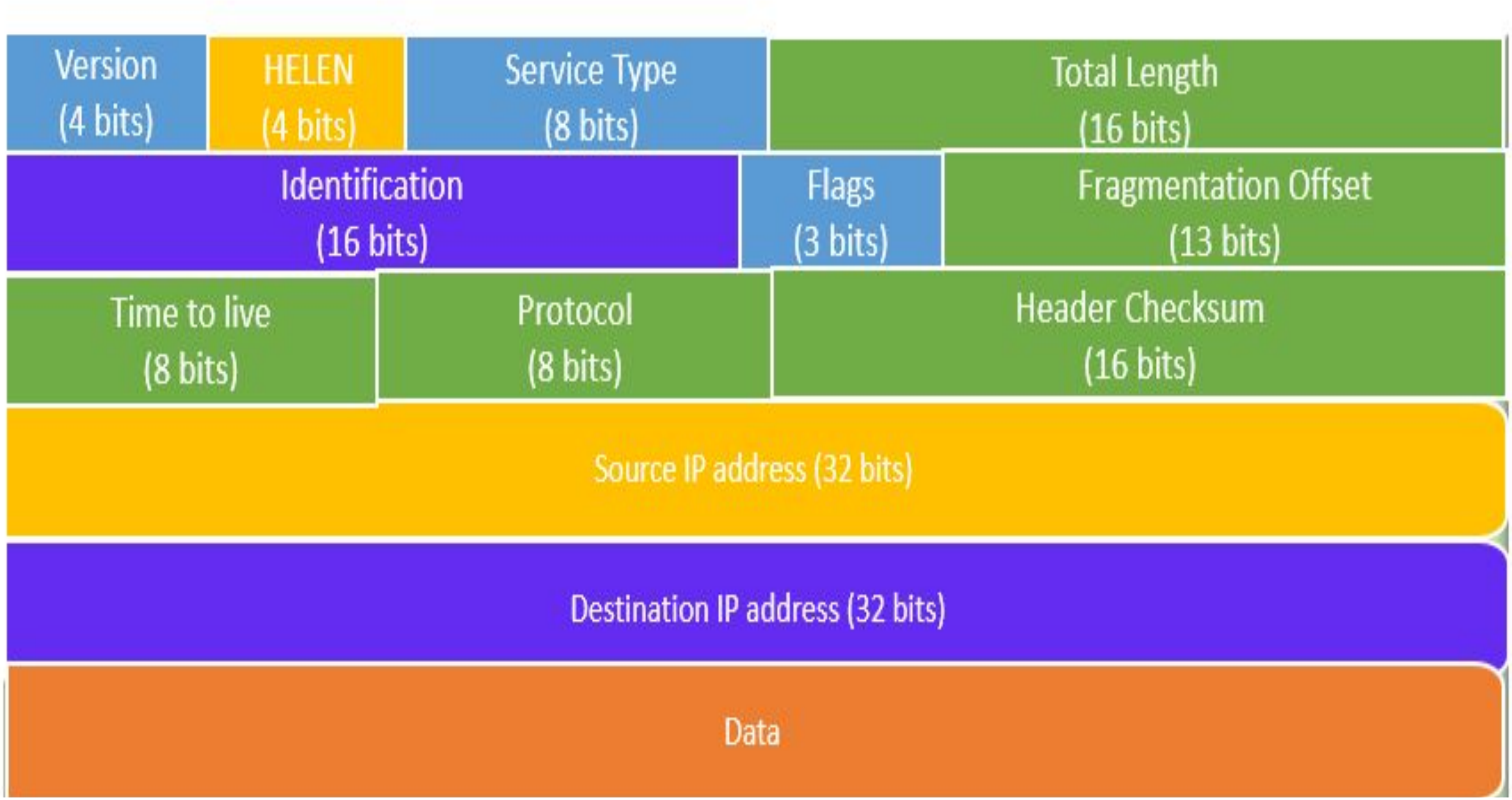


- ❖The above diagram shows the address format of IPv4 and IPv6.
- ❖An IPv4 is a 32-bit decimal address. It contains 4 octets or fields separated by 'dot', and each field is 8-bit in size.
- ❖The number that each field contains should be in the range of 0-255.
- ❖Whereas an IPv6 is a 128-bit hexadecimal address.
- ❖It contains 8 fields separated by a colon, and each field is 16-bit in size.

MLR Institute of Technology

Network Layer in the Internet…

IPV4 Header Format:



Version (4 bits) | HELEN (4 bits) | Service Type (8 bits) | Total Length (16 bits)

Identification (16 bits) | Flags (3 bits) | Fragmentation Offset (13 bits)

Time to live (8 bits) | Protocol (8 bits) | Header Checksum (16 bits)

Source IP address (32 bits)

Destination IP address (32 bits)

Data

Network Layer in the Internet…

IPV4 Header Format:

Following are various components/fields of IP packet header:

Version: The first IP header field is a 4-bit version indicator. In IPv4, the value of its four bits is set to 0100, which indicates 4 in binary.

Internet Header Length: Internet header length, shortly known as IHL, is 4 bits in size. It is also called HELEN (Header Length).

- This IP component is used to show how many 32-bit words are present in the header.

Network Layer in the Internet…

IPV4 Header Format:

Type of Service: Type of Service is also called Differentiated Services Code Point or DSCP.

- This field is provided features related to the quality of service for data streaming or VoIP calls.

Total length: The total length is measured in bytes.
Header+Data
The minimum size of an IP datagram is 20 bytes and the maximum, it can be 65535 bytes . HELEN and Total length can be used to calculate the dimension of the payload.

- All hosts are required to be able to read 576-byte datagrams. However, if a datagram is too large for the hosts in the network, the fragmentation method is widely used.

Network Layer in the Internet…

IPV4 Header Format:

Identification: Identification is a packet that is used to identify fragments of an IP datagram uniquely.

- Some have recommended using this field for other things like adding information for packet tracing, etc.

IP Flags: Flag is a three-bit field that helps you to control and identify fragments.

The following can be their possible configuration:
Bit 0: is reserved and has to be set to zero
Bit 1: means do not fragment
Bit 2: means more fragments.

Network Layer in the Internet…

IPV4 Header Format:

Fragment Offset: Fragment Offset represents the number of Data Bytes ahead of the particular fragment in the specific Datagram.

- It is specified in terms of the number of 8 bytes, which has a maximum value of 65,528 bytes.

Time to live: It is an 8-bit field that indicates the maximum time the Datagram will be live in the internet system. The time duration is measured in seconds, and when the value of TTL is zero, the Datagram will be erased.

- Every time a datagram is processed its TTL value is decreased by one second.
- TTL are used so that datagrams are not delivered and discarded automatically.
- The value of TTL can be 0 to 255.

Network Layer in the Internet…

IPV4 Header Format:

Protocol: This IPv4 header is reserved to denote that internet protocol is used in the Datagram. For Example, 6 number digit is mostly used to indicate TCP, and 17 is used to denote the UDP protocol.

Header Checksum: The next component is a 16 bits header checksum field, which is used to check the header for any errors. The IP header is compared to the value of its checksum. When the header checksum is not matching, then the packet will be discarded.

Source Address: The source address is a 32-bit address of the source used for the IPv4 packet.

Network Layer in the Internet…

IPV4 Header Format:

Destination address: The destination address is also 32 bit in size stores the address of the receiver.

(IP Options: It is an optional field of IPv4 header used when the value of IHL (Internet Header Length) is set to greater than 5.

Data: This field stores the data from the protocol layer, which has handed over the data to the IP layer.

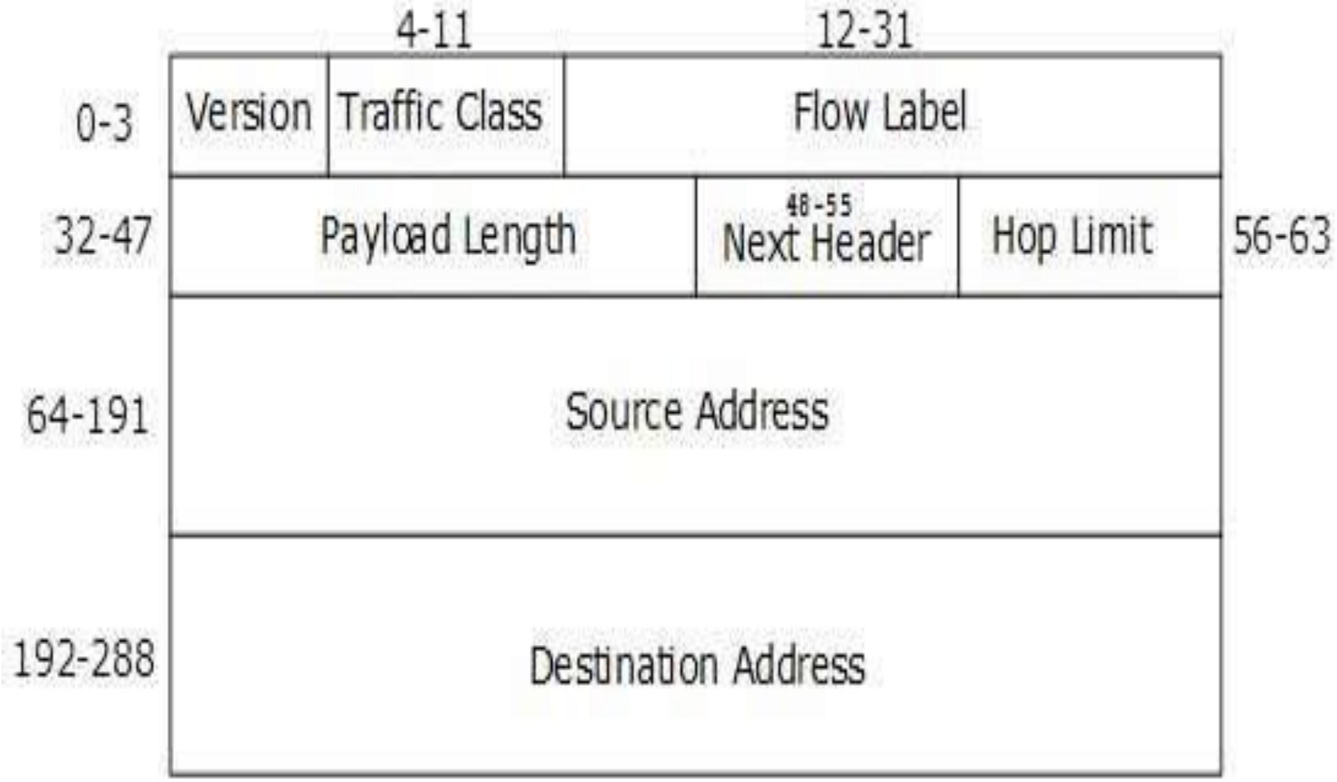
Network Layer in the Internet…

IPV6 Header Format:

- Fixed Header

- Extension Headers

Fixed Header:

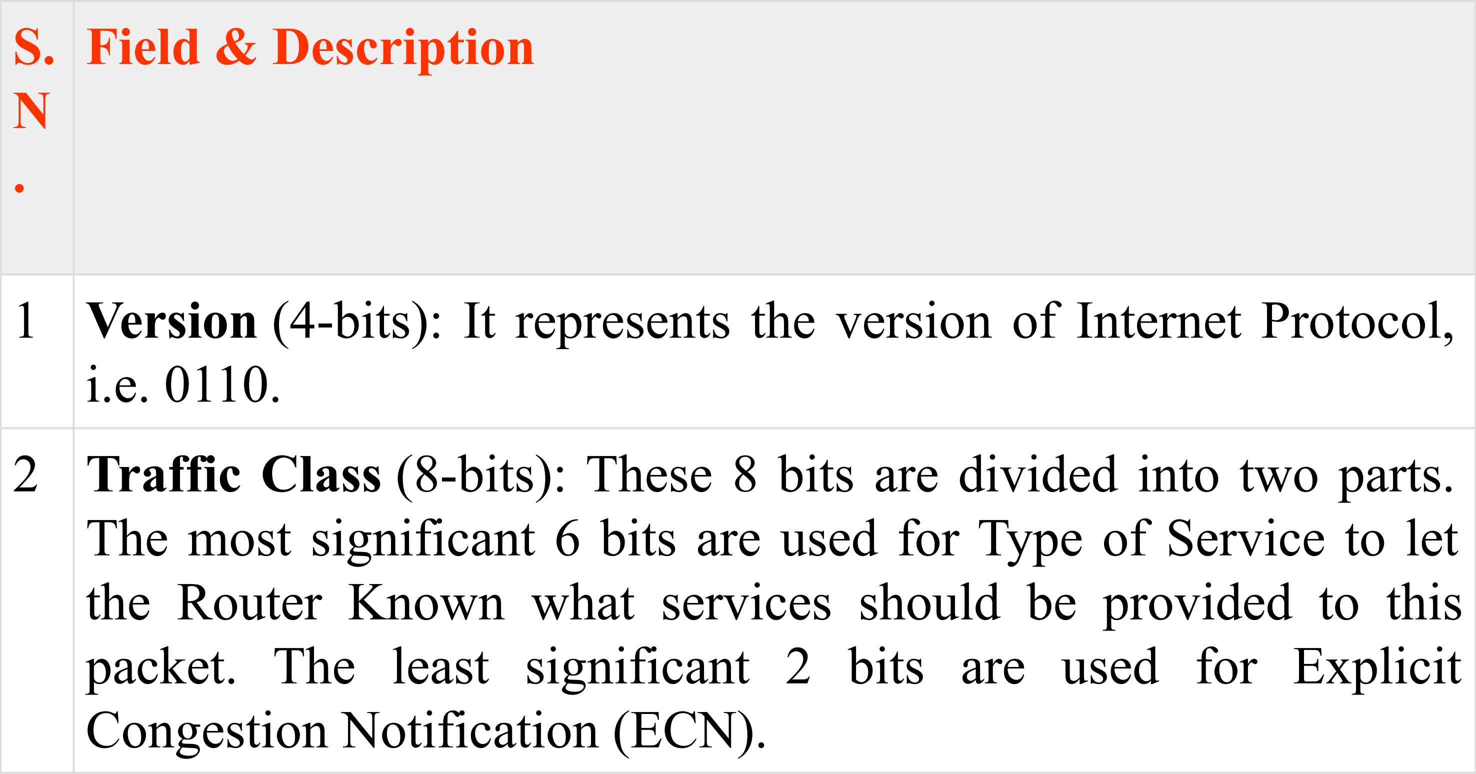


		4-11	12-31		
0-3	Version	Traffic Class	Flow Label		
32-47	Payload Length		48-55 Next Header	Hop Limit	56-63
64-191	Source Address				
192-288	Destination Address				

Network Layer in the Internet…

Fixed Header:

IPv6 fixed header is 40 bytes long and contains the following information.



S.N.	Field & Description
1	**Version** (4-bits): It represents the version of Internet Protocol, i.e. 0110.
2	**Traffic Class** (8-bits): These 8 bits are divided into two parts. The most significant 6 bits are used for Type of Service to let the Router Known what services should be provided to this packet. The least significant 2 bits are used for Explicit Congestion Notification (ECN).

Network Layer in the Internet…

Fixed Header:

3	**Flow Label** (20-bits): This label is used to maintain the sequential flow of the packets belonging to a communication.
4	**Payload Length** (16-bits): This field is used to tell the routers how much information a particular packet contains in its payload. Payload is composed of Extension Headers and Upper Layer data. With 16 bits, up to 65535 bytes can be indicated.
5	**Next Header** (8-bits): This field is used to indicate either the type of Extension Header.

Network Layer in the Internet…

Fixed Header:

6	**Hop Limit** (8-bits): This field is used to stop packet to loop in the network infinitely. This is same as TTL in IPv4. The value of Hop Limit field is decremented by 1 as it passes a link (router/hop). When the field reaches 0 the packet is discarded.
7	**Source Address** (128-bits): This field indicates the address of originator of the packet.
8	**Destination Address** (128-bits): This field provides the address of intended recipient of the packet.

Network Layer in the Internet…

Extension Header:

- In IPv6, the Fixed Header contains only that much information which is necessary, avoiding those information which is either not required or is rarely used. All such information is put between the Fixed Header and the Upper layer header in the form of Extension Headers. Each Extension Header is identified by a distinct value.

- When Extension Headers are used, IPv6 Fixed Header's Next Header field points to the first Extension Header. If there is one more Extension Header, then the first Extension Header's 'Next-Header' field points to the second one, and so on. The last Extension Header's 'Next-Header' field points to the Upper Layer Header. Thus, all the headers points to the next one in a linked list manner.

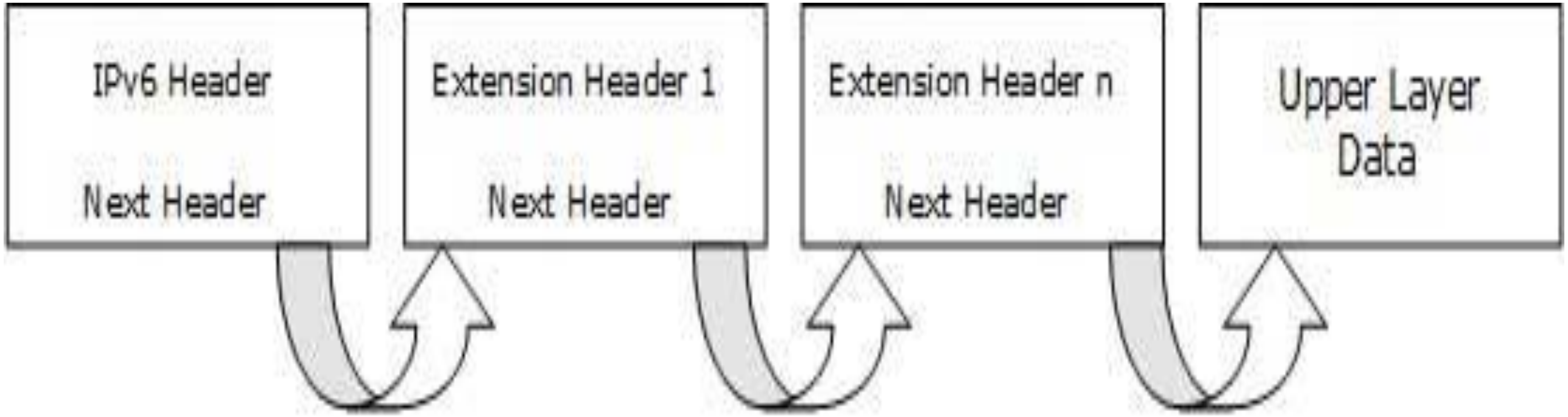
Network Layer in the Internet…

Extension Header:

These headers:

1. should be processed by First and subsequent destinations.
2. should be processed by Final Destination.

Extension Headers are arranged one after another in a linked list manner, as depicted in the following diagram:



Network Layer in the Internet…

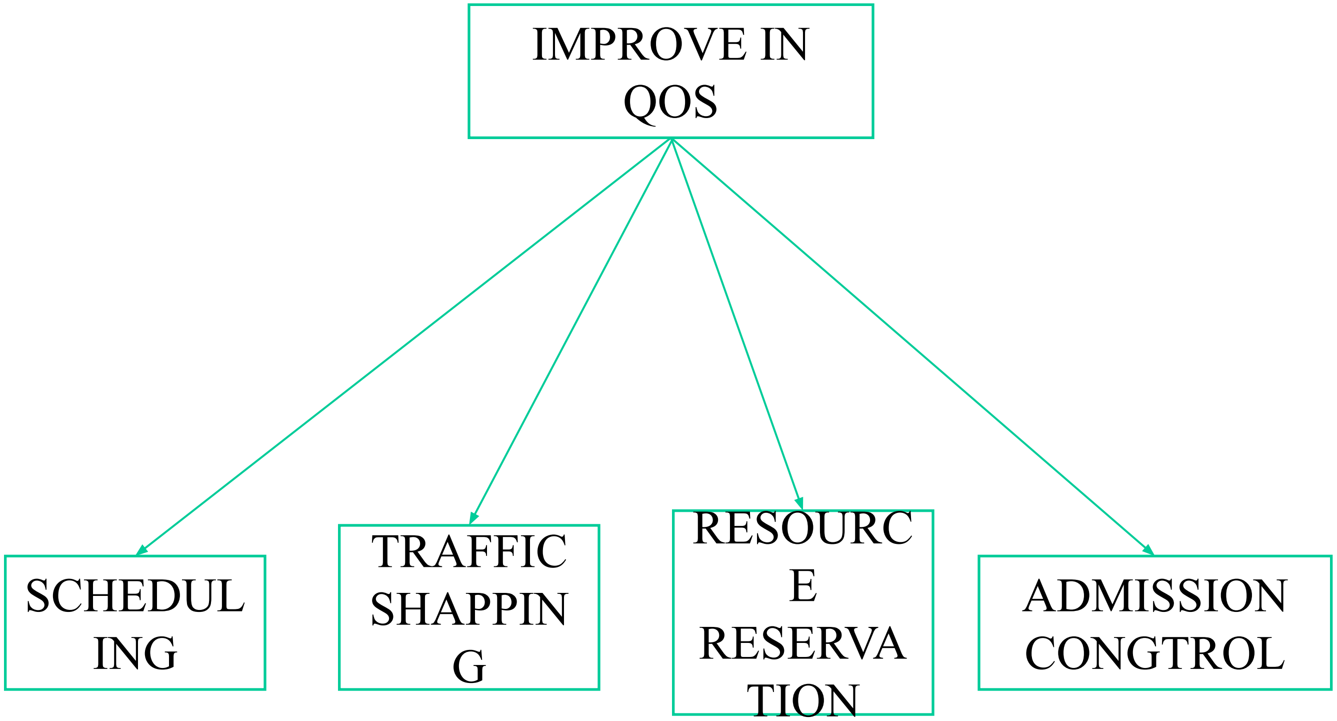
Differences between IPv4 and IPv6

	Ipv4	Ipv6
Address length	IPv4 is a 32-bit address.	IPv6 is a 128-bit address.
Fields	IPv4 is a numeric address that consists of 4 fields which are separated by dot (.).	IPv6 is an alphanumeric address that consists of 8 fields, which are separated by colon.
Classes	IPv4 has 5 different classes of IP address that includes Class A, Class B, Class C, Class D, and Class E.	IPv6 does not contain classes of IP addresses.
Number of IP address	IPv4 has a limited number of IP addresses.	IPv6 has a large number of IP addresses.
VLSM	It supports VLSM (Virtual Length Subnet Mask). Here, VLSM means that Ipv4 converts IP addresses into a subnet of different sizes.	It does not support VLSM.

Address configuration	It supports manual and DHCP configuration.	It supports manual, DHCP, auto-configuration, and renumbering.
Address space	It generates 4 billion unique addresses	It generates 340 undecillion unique addresses.
End-to-end connection integrity	In IPv4, end-to-end connection integrity is unachievable.	In the case of IPv6, end-to-end connection integrity is achievable.
Security features	In IPv4, security depends on the application. This IP address is not developed in keeping the security feature in mind.	In IPv6, IPSEC is developed for security purposes.
Address representation	In IPv4, the IP address is represented in decimal.	In IPv6, the representation of the IP address in hexadecimal.

Fragmentation	Fragmentation is done by the senders and the forwarding routers.	Fragmentation is done by the senders only.
Packet flow identification	It does not provide any mechanism for packet flow identification.	It uses flow label field in the header for the packet flow identification.
Checksum field	The checksum field is available in IPv4.	The checksum field is not available in IPv6.
Transmission scheme	IPv4 is broadcasting.	On the other hand, IPv6 is multicasting, which provides efficient network operations.
Encryption and Authentication	It does not provide encryption and authentication.	It provides encryption and authentication.
Number of octets	It consists of 4 octets.	It consists of 8 fields, and each field contains 2 octets. Therefore, the total number of octets in IPv6 is 16.

Quality of Service



IMPROVE IN QOS

SCHEDULING

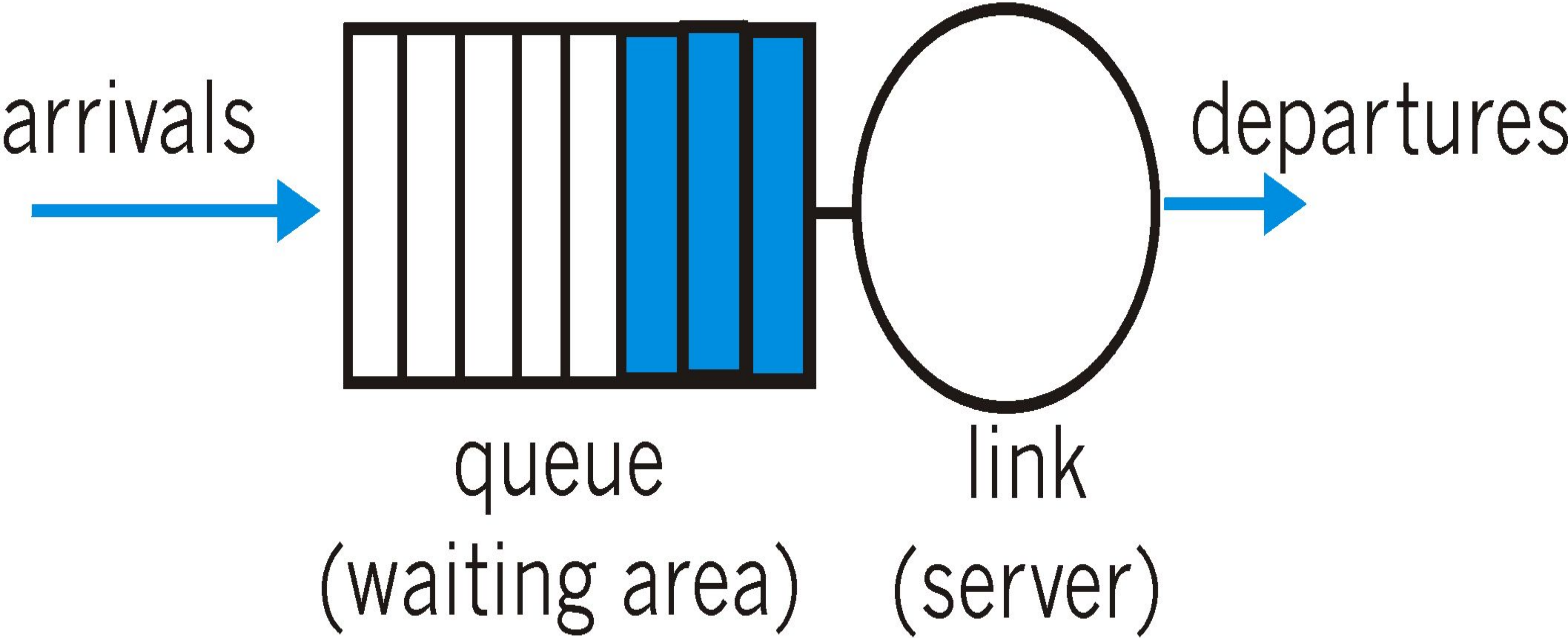
TRAFFIC SHAPPING

RESOURCE RESERVATION

ADMISSION CONGTROL

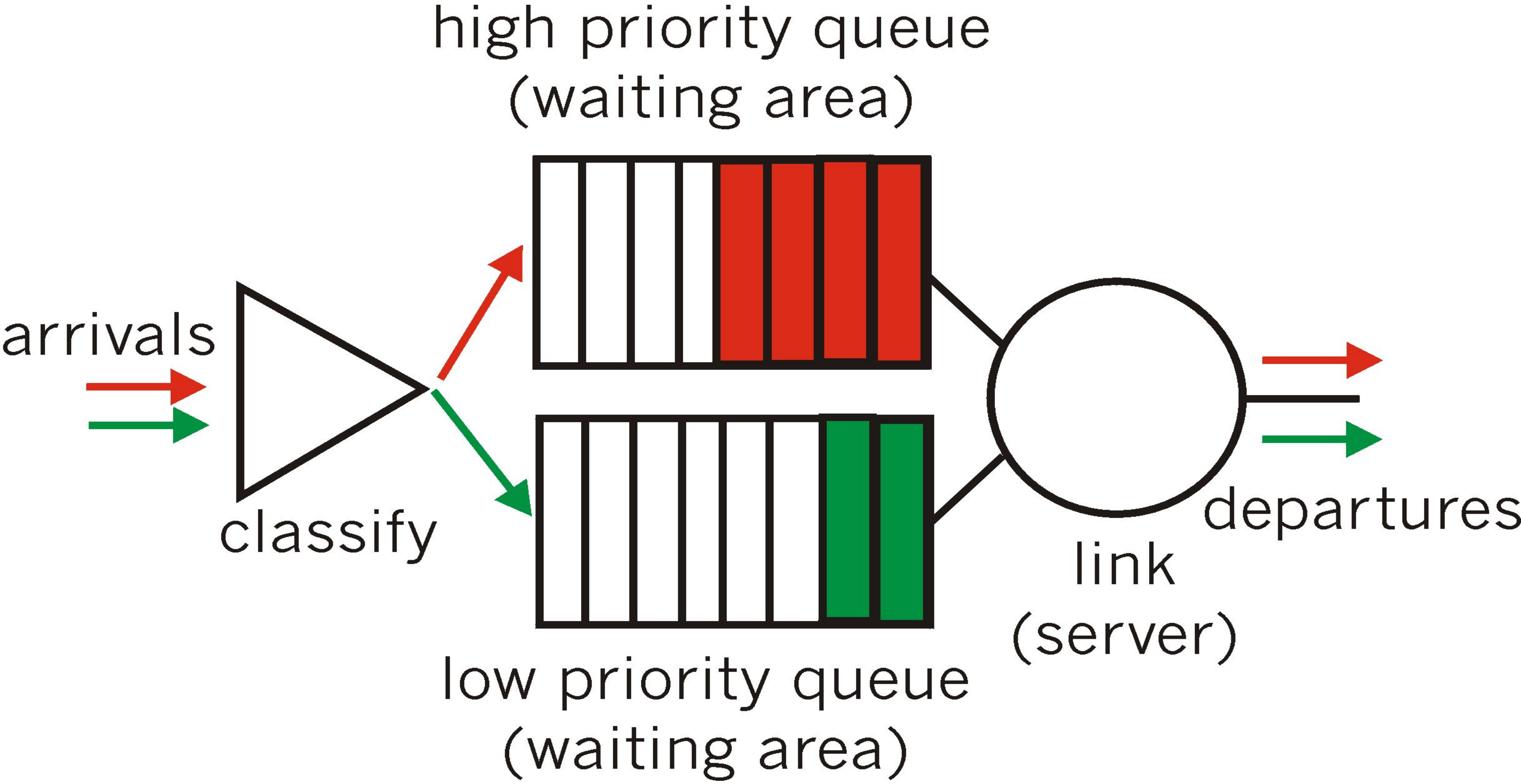
SCHEDULING

1. FCFS SCHEDULING



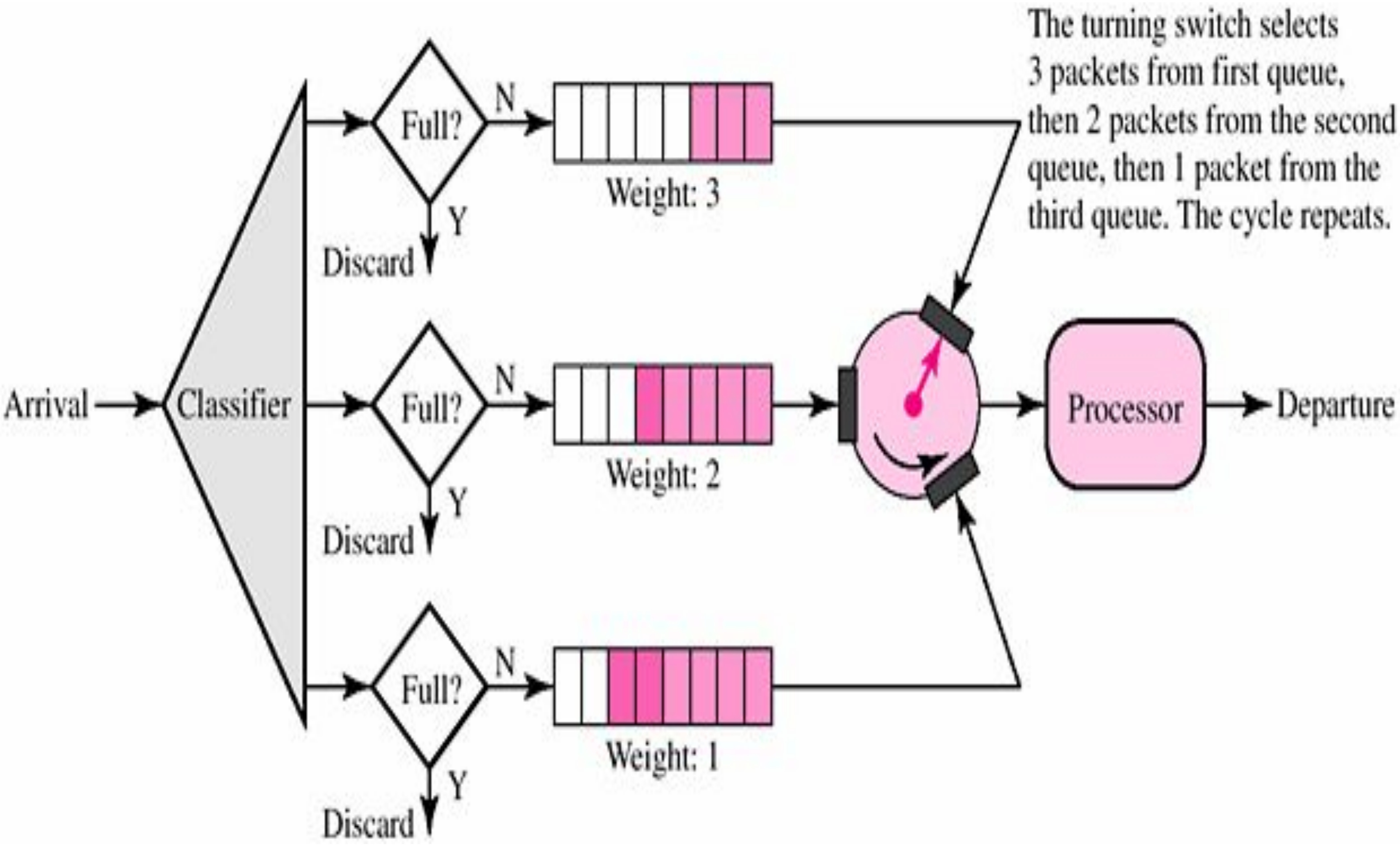
SCHEDULING

2. PRIORITY BASED SCHEDULING



SCHEDULING

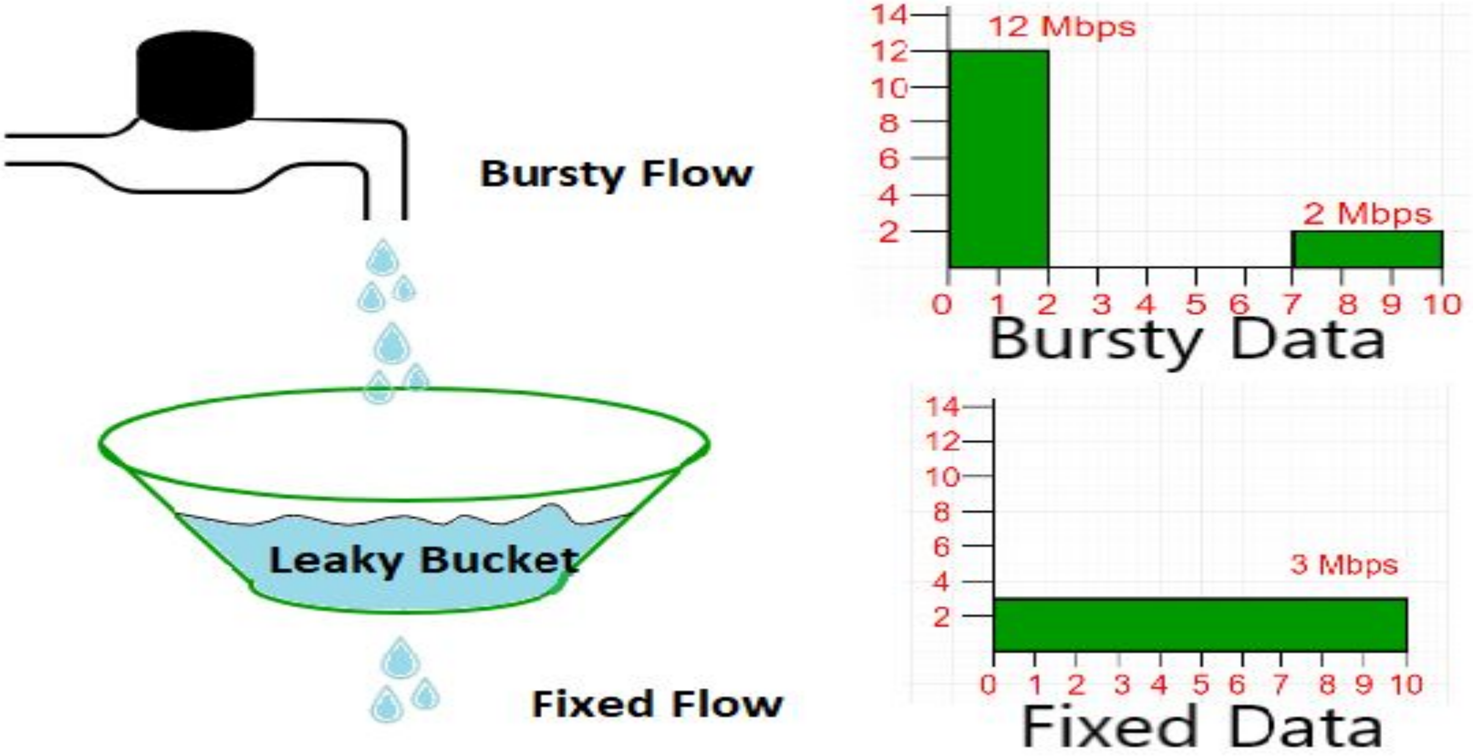
3. WEIGHED FAIR QUEUE SCHEDULING



The turning switch selects 3 packets from first queue, then 2 packets from the second queue, then 1 packet from the third queue. The cycle repeats.

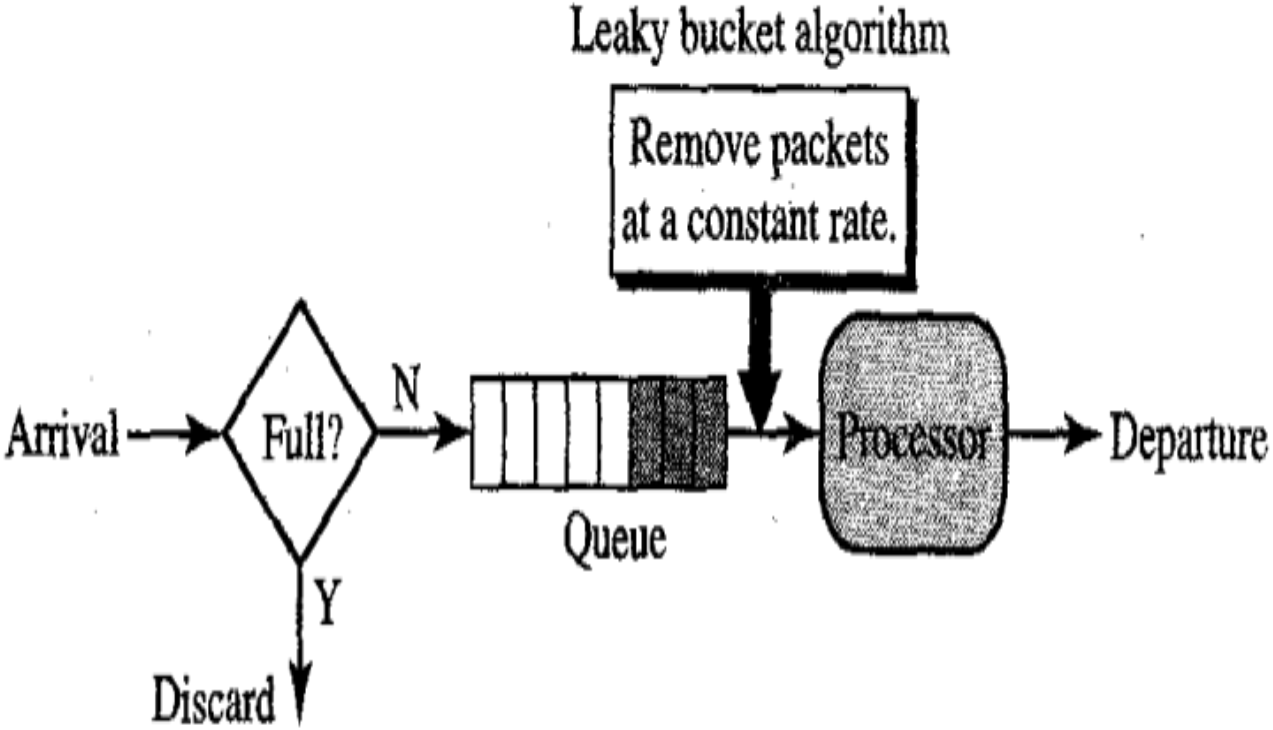
TOKEN SHAPPING

1. LEACKY BUCKET MECHANISM



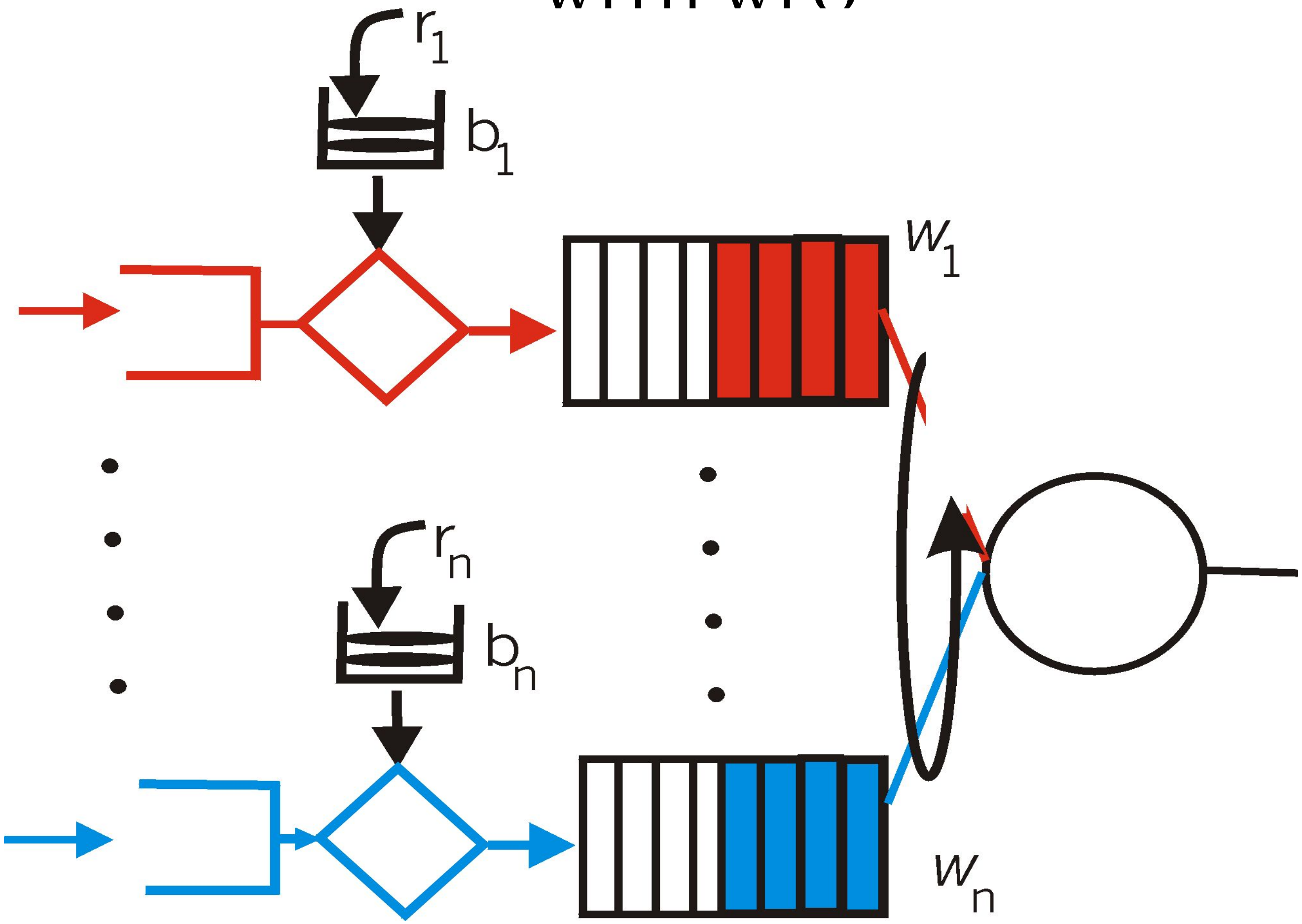
TOKEN SHAPPING

1. LEACKY BUCKET MECHANISM



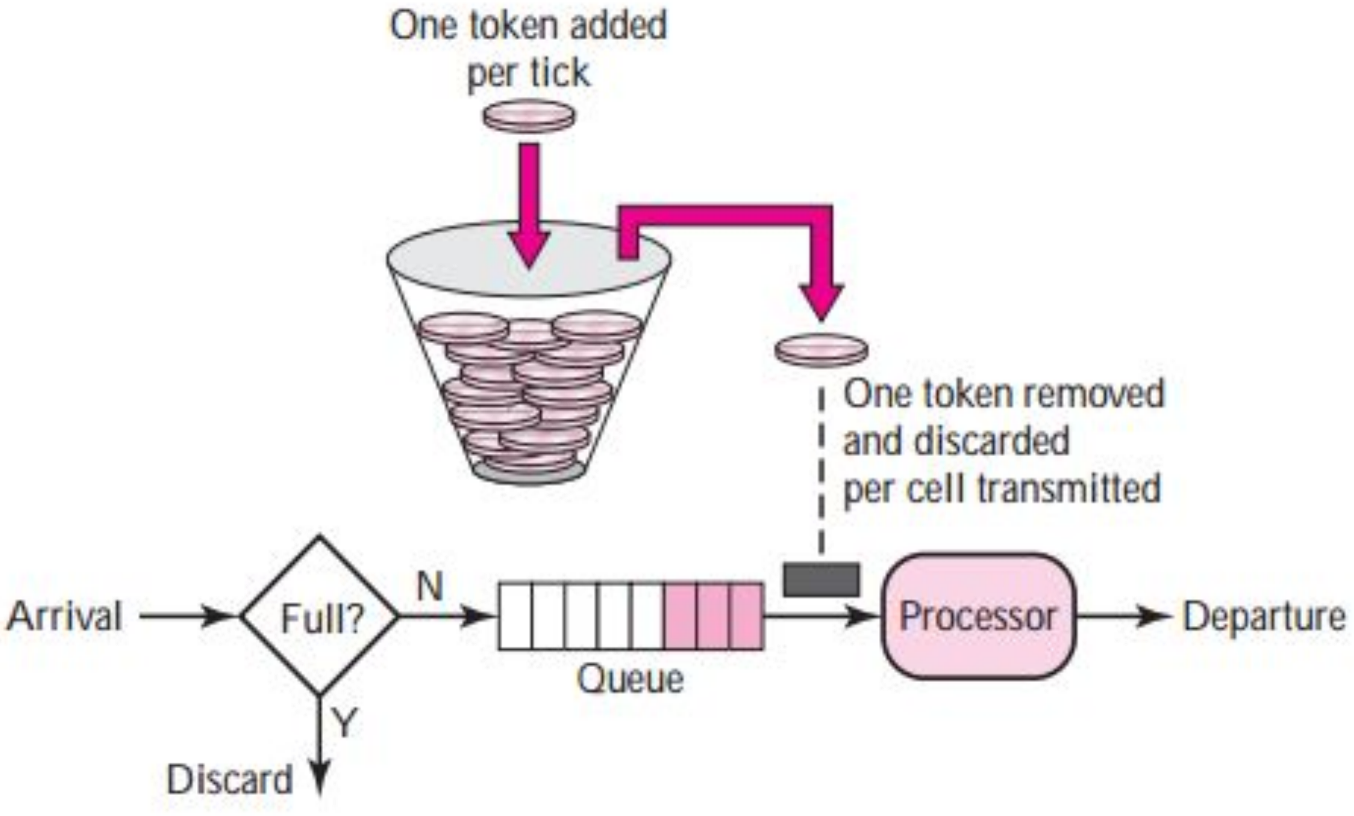
TOKEN SHAPPING

N MULTIPLEXED LEACKY BUCKET FLOW WITH WFQ



TOKEN SHAPPING

2. TOKEN BUCKET MECHANISM

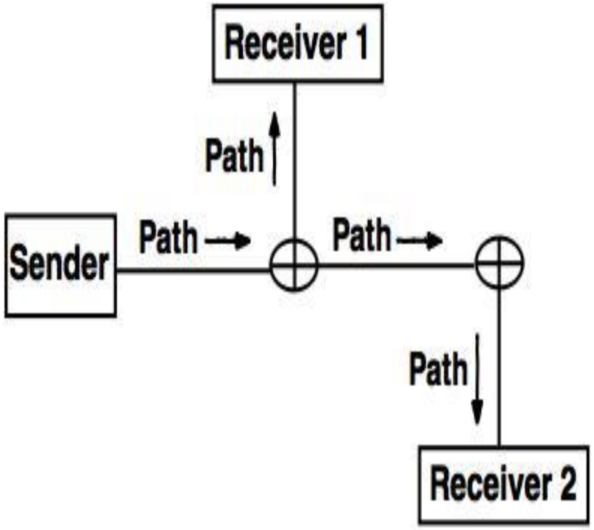


RESOURCE RESERVATION

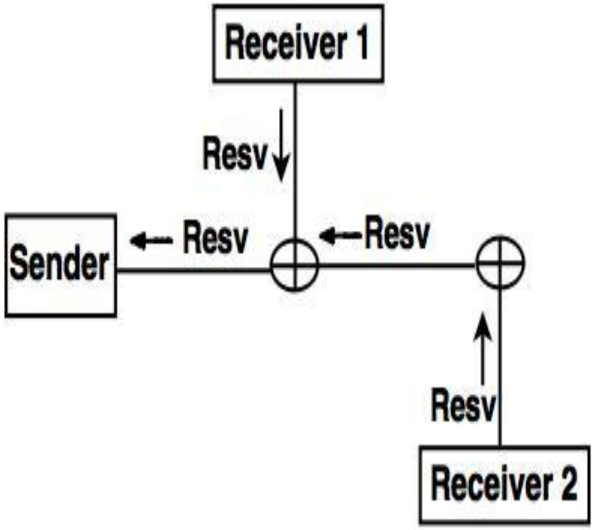
1. RESOURCE RESERVATION PROTOCOL

FIRST STEP: PATH MESSAGING

SECOND STEP: RESERVATION MESSAGING



Path messages



Resv Messages