UNIT-1

NETWORK LAYER- I

Prof. Veena Gadad, dept of CSE, RVCE

- Network layer design issues
- Store and Forward packet Switching
- Services Provided to the Transport Layer.
- Implementation of Connectionless Service.
- Implementation of Connection-Oriented Service.
- Comparison of Virtual Circuit and Datagram Subnets.
- Routing algorithms:
  - Shortest Path Routing,
  - Flooding, Distance Vector Routing,
  - Link state Routing,
  - Hierarchical Routing,
  - Broadcast Routing,
  - Multicast Routing
- Congestion Control Algorithms
- Quality Of Service

Network layer

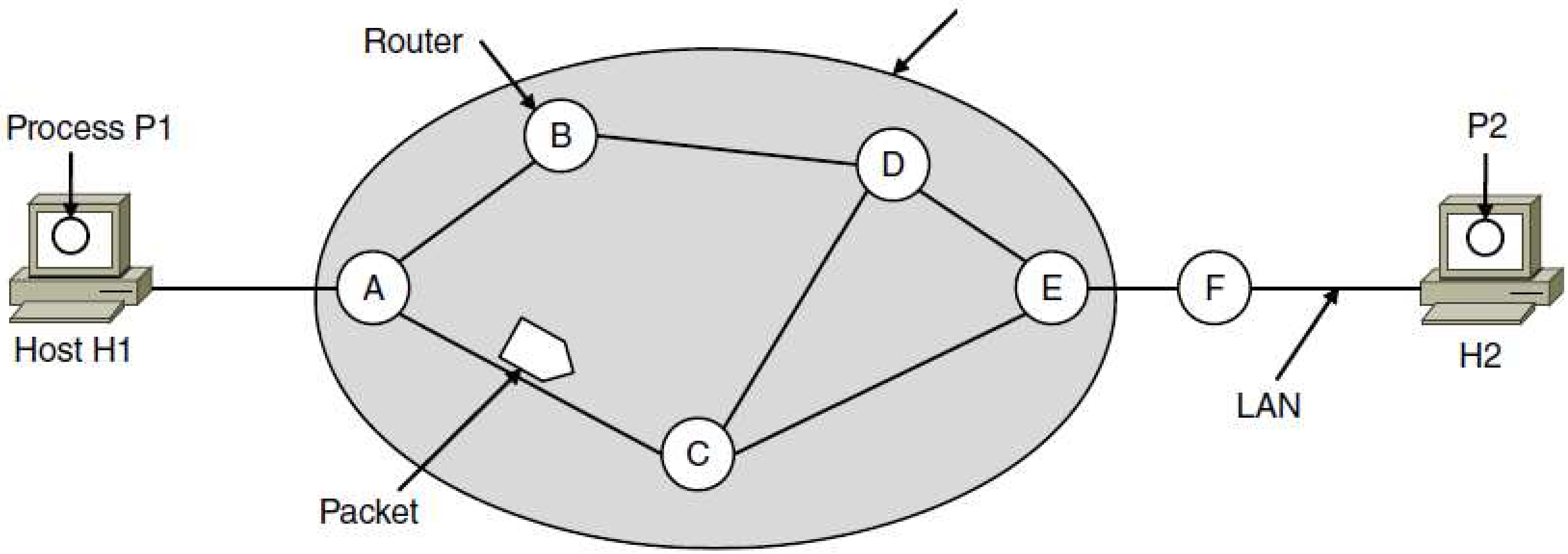
- The network layer is concerned with getting packets from source all the way to destination across many intermediate hops.
- Network layer is responsible for moving packets from source to destination in contrast to data link layer which is responsible for moving frames from one end of the wire to another.
- Network layer should know the topology of the subnet- set of all routers and appropriate paths.

Store and forward packet switching

- It is a telecommunication technique in which information is sent to an intermediate station where it is kept and sent at a later time to the final destination or to another intermediate station.
- The intermediate station, or node in a networking context, verifies the integrity of the message before forwarding it.
- Network layer works in this context(Data link layer operates as cut through switching)

Environment of network layer protocol

- The major components are ISP's equipments connected by transmission lines.



- The end systems are known as customers equipments.

How network layer operates?

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

Network layer design issues

- When network layer is designed there are certain issues that the designers must be concerned with to provide service to transport layer.
- The services are provided with following goals in mind:
  - The services should be independent of router technology.
  - The transport layer should be shielded from the number, type and topology of routers present.
  - The network addresses made available to the transport layer should use a uniform numbering plan, even across LANs and WANs.

The functions of this layer include :

1. Routing:

- The process of transferring packets received from the Data Link Layer of the source network to the Data Link Layer of the correct destination network is called routing.
- Involves decision making at each intermediate node on where to send the packet next so that it eventually reaches its destination.
- The node which makes this choice is called a router.
- For routing we require some mode of addressing which is recognized by the Network Layer.
- This addressing is called IP addressing

2. Inter-networking - The network layer is the same across all physical networks (such as Token-Ring and Ethernet).

- Thus, if two physically different networks have to communicate, the packets that arrive at the Data Link Layer of the node which connects these two physically different networks, would be stripped of their headers and passed to the Network Layer.
- The network layer would then pass this data to the Data Link Layer of the other physical network..

3. Congestion Control

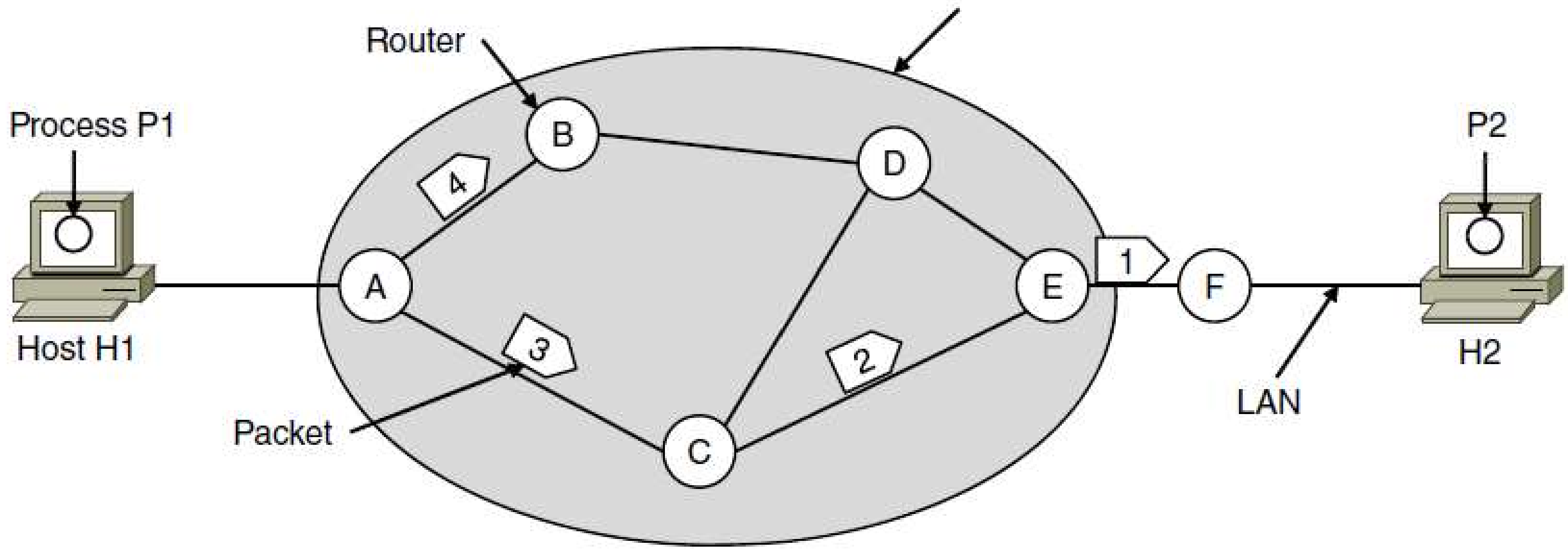
- If the incoming rate of the packets arriving at any router is more than the outgoing rate, then congestion is said to occur.
- Congestion may be caused by many factors.
  - If suddenly, packets begin arriving on many input lines and all need the same output line, then a queue will build up.
  - If there is insufficient memory to hold all of them, packets will be lost.
  - if routers have an infinite amount of memory, congestion gets worse, because by the time packets reach to the front of the queue, they have already timed out (repeatedly), and duplicates have been sent.
  - All these packets are dutifully forwarded to the next router, increasing the load all the way to the destination.
  - Another reason for congestion are slow processors. If the router's CPUs are slow at performing the bookkeeping tasks required of them, queues can build up, even though there is excess line capacity.
  - Similarly, low-bandwidth lines can also cause congestion.

Implementation of Connectionless service

- With this service, packets are injected into the network individually and routed independently of each other.
- No advance setup is needed.
- Packets are frequently called **datagrams.**
- The network is called a **datagram network.**

Working

- Suppose P1 wants to send long message to P2, it sends message to transport layer with instruction to deliver it.
- It prepends a transport header to the front of the message and hands the result to the network layer, probably just another procedure within the operating system.



- Assuming 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 using some point-to-point protocol.
- Every router has an internal table telling it where to send packets for each of the possible destinations.
- Each table entry is a pair consisting of a destination and the outgoing line to use for that destination.

- Initial routing tables

A's table (initially)

Dest.	Line
A	☒
B	B
C	C
D	B
E	C
F	C

A's table (later)

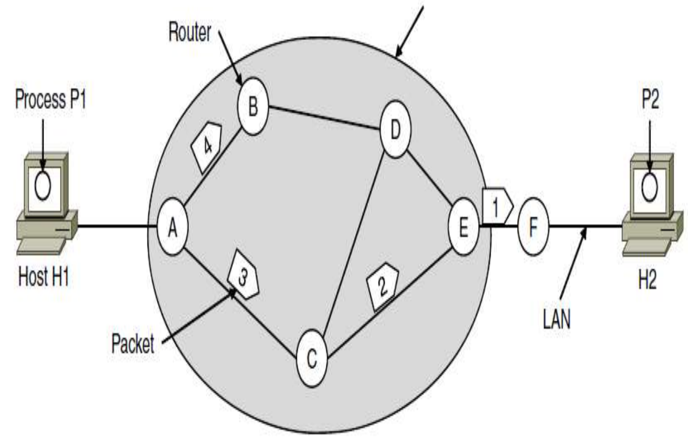
A	☒
B	B
C	C
D	B
E	D
F	D

C's Table

A	A
B	A
C	☒
D	E
E	E
F	E

E's Table

A	C
B	D
C	C
D	D
E	☒
F	F



- At A, packets 1, 2, and 3 are stored briefly, having arrived on the incoming link and had their checksums verified.
- Then each packet is forwarded according to A's table, onto the outgoing link to C within a new frame.

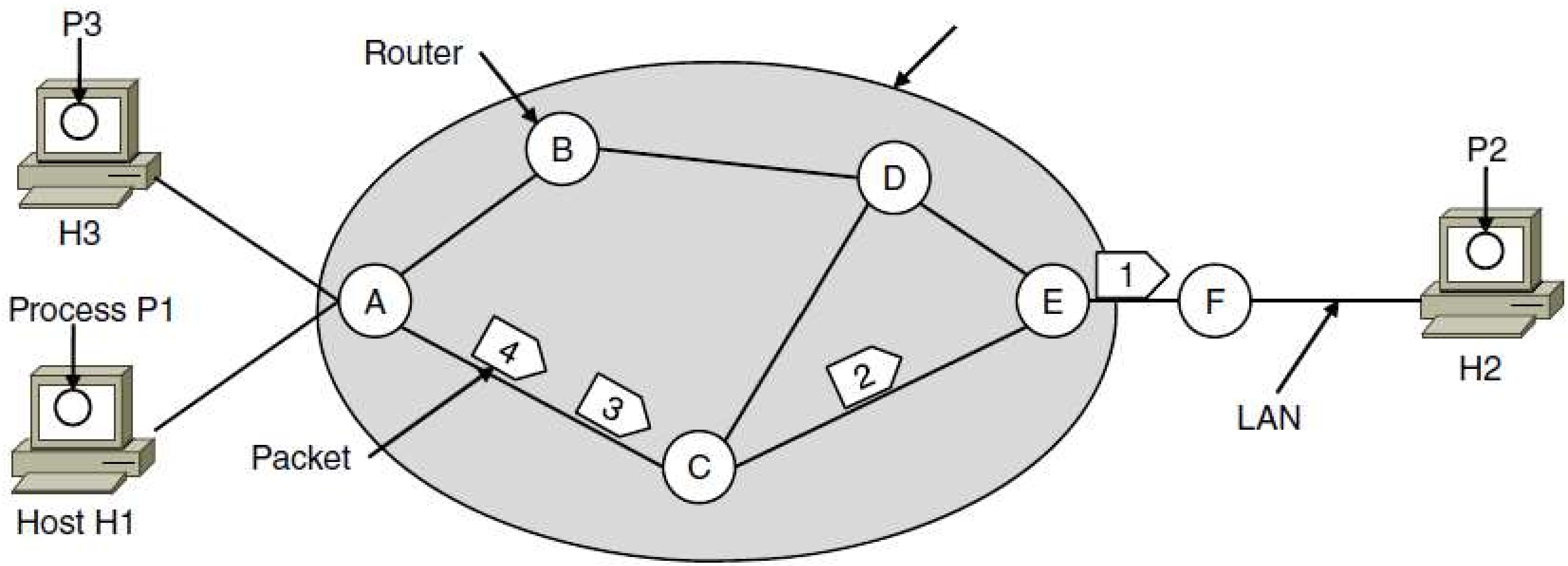
- If at all there is congestion in the path, the router may take a decision to take an alternative path, this happens to packet 4.
- The algorithm that manages the tables and makes the routing decisions is called the **routing algorithm.**

Implementation of Connection-Oriented Service

- The connection oriented service is called Virtual Circuit and the network is called virtual circuit network.

- The idea behind virtual circuits is to avoid having to choose a new route for every packet sent.
- 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.
- 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.

Consider the following network



- Host H1 has established connection 1 with host H2. This connection is remembered as the first entry in each of the routing tables.

- The first line of A’s table says that if a packet bearing connection identifier 1 comes in from H1, it is to be sent to router C and given connection identifier 1.

A’s table

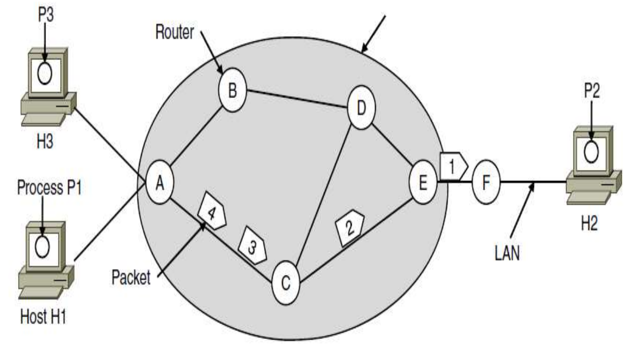
In		Out	
H1	1	C	1
H3	1	C	2

C’s Table

A	1	E	1
A	2	E	2

E’s Table

C	1	F	1
C	2	F	2



- The first entry at C routes the packet to E, also with connection identifier 1.
- H3 to H2: H3 chooses identifier 1, A chooses identifier 2 to avoid conflict.
- This is called as **label switching- MPLS**

Comparison of VC 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

Definition:

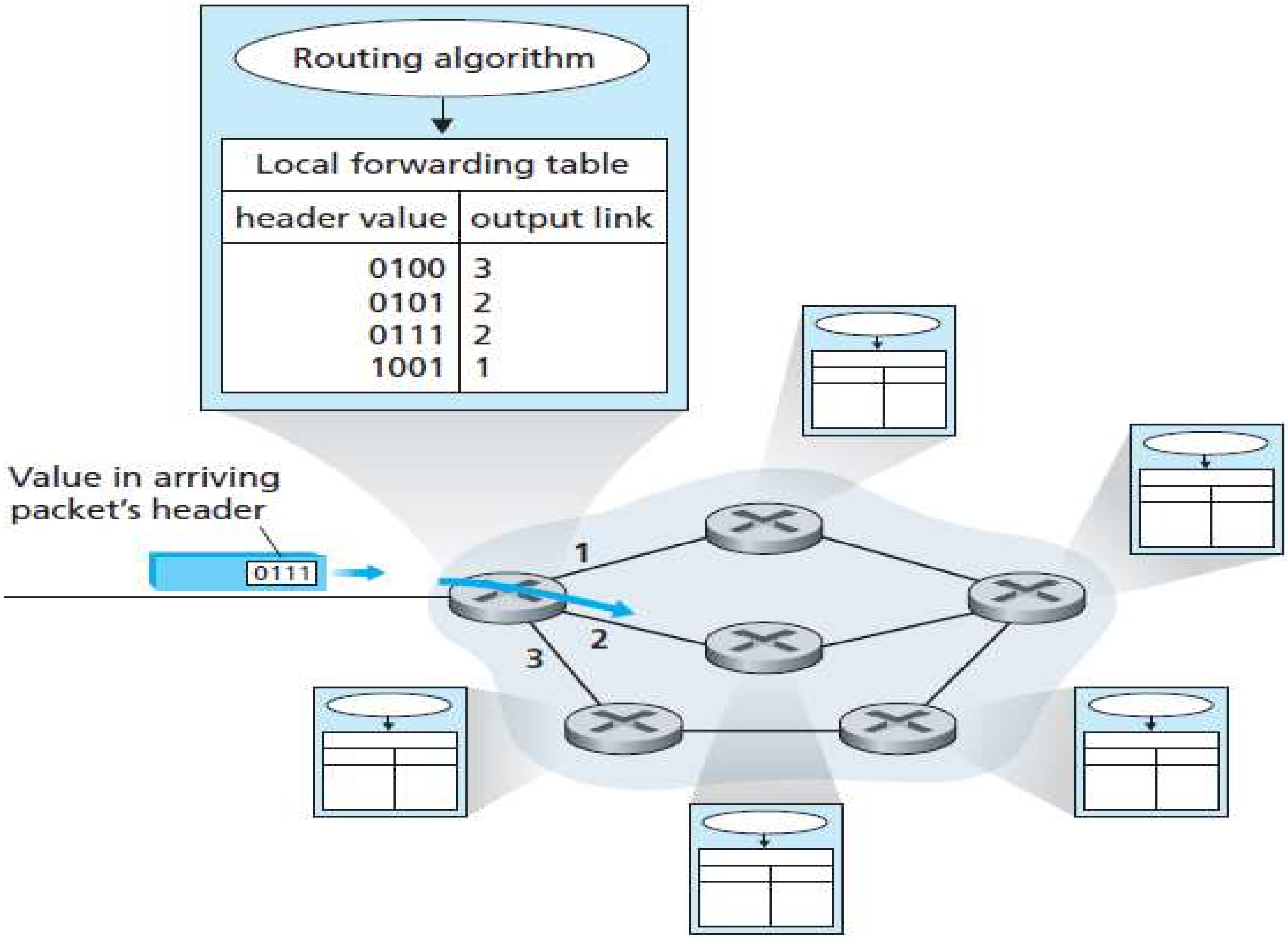
- Routing algorithm- is part of the network layer software responsible for deciding which output line an incoming packet should be transmitted on.
- For datagram subnets this decision is to be made every time for new packet.
- For VC subnets, routing decisions are made only once when a new VC is established
  - Also called **session routing-** the route remains in force for an entire session- e.g. VPN

Routing vs Forwarding

- The role of the network layer is thus deceptively simple—to move packets from a sending host to a receiving host.
- To do so, two important network-layer functions can be identified:
  - *Forwarding*. When a packet arrives at a router's input link, the router must move the packet to the appropriate output link.
  - *Routing*. The network layer must determine the route or path taken by packets as they flow from a sender to a receiver. It includes updating and filling of forwarding tables.

- *Forwarding* refers to the router-local action of transferring a packet from an input link interface to the appropriate output link interface.
- *Routing* refers to the network-wide process that determines the end-to-end paths that packets take from source to destination.
- Every router has a **forwarding table.**
- The value stored in the forwarding table entry for that header indicates the router's outgoing link interface to which that packet is to be forwarded.

Routing algorithms determine values in forwarding tables



- The routing algorithm may be centralized or decentralized.
  - An algorithm executing on a central site and downloading routing information to each of the routers.
  - A piece of the distributed routing algorithm running in each router.
- Every router receives routing protocol messages, which are used to configure its forwarding table.
- Every router has two processes and an operating system to care of routing and forwarding.

Properties of routing algorithm:

Correctness, simplicity, robustness, stability, fairness, and efficiency.

- *Robustness*: once the network is put to air it is expected to run continuously for years without system wide failures.
- *Stability*: A stable algorithm reaches equilibrium and stays there. It should converge quickly too, since communication may be disrupted until the routing algorithm has reached equilibrium.
- *Fairness and efficiency*: Minimizing the mean packet delay and maximizing total network throughput are important to send traffic effectively.

Classification of routing algorithms

- Routing algorithms can be grouped into two major classes: nonadaptive and adaptive.
- *Nonadaptive algorithms:* do not base their routing decisions on any measurements or estimates of the current topology and traffic.
  - The route is computed in advance, offline and downloaded to the routers when the network is booted.(also called as static routing).
  - These algorithms do not respond to failures and are useful when the routing choice is clear.

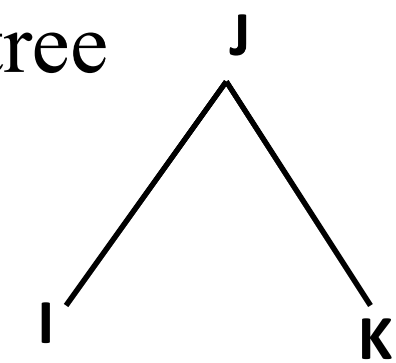
- Adaptive algorithms, in contrast, change their routing decisions to reflect changes in the topology, and sometimes changes in the traffic as well.
  - Also called as dynamic routing algorithms.
- These algorithms differ in
  - Where they get their information (e.g., locally, from adjacent routers, or from all routers)
  - When they change the routes (e.g., when the topology changes, or every ΔT *seconds as the load changes)*
  - What metric is used for optimization (e.g., distance, number of hops, or estimated transit time).

Optimality Principle(Stated by Bellman 1957)

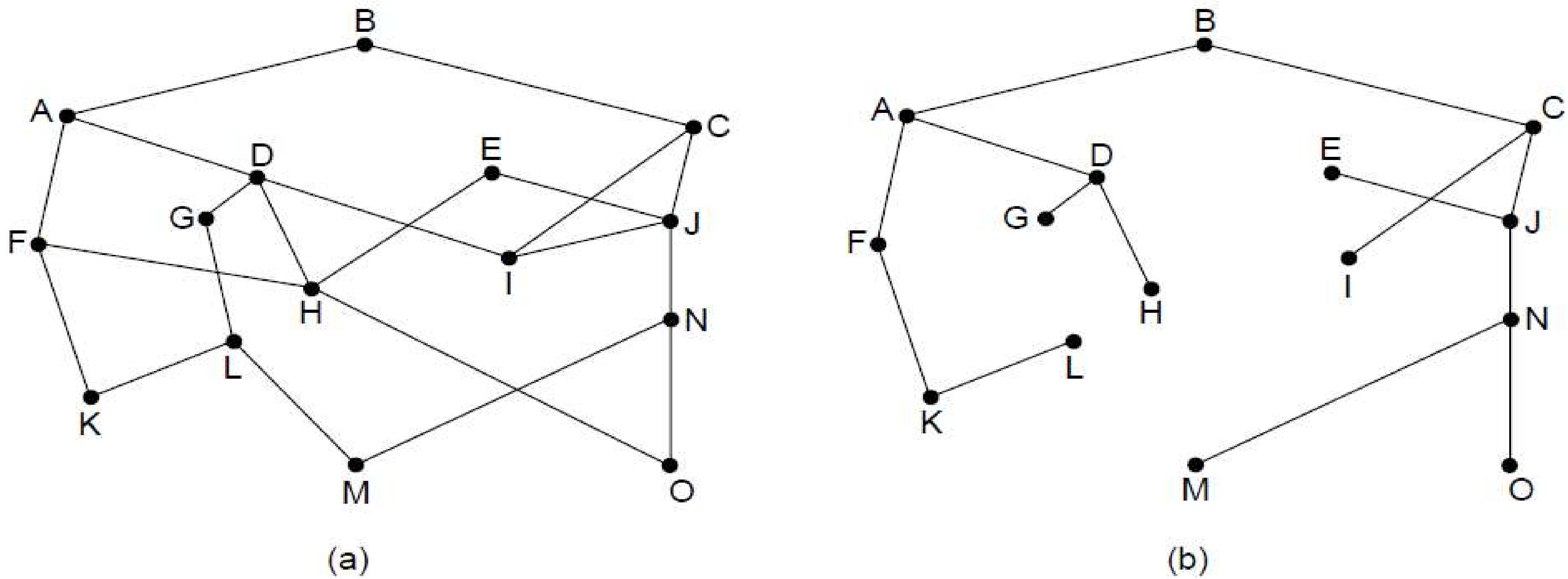
Statement:

"If router J is on the optimal path from router I to router K, then the optimal path from J to K also falls along the same route."

- As a direct consequence of the optimality principle, 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**



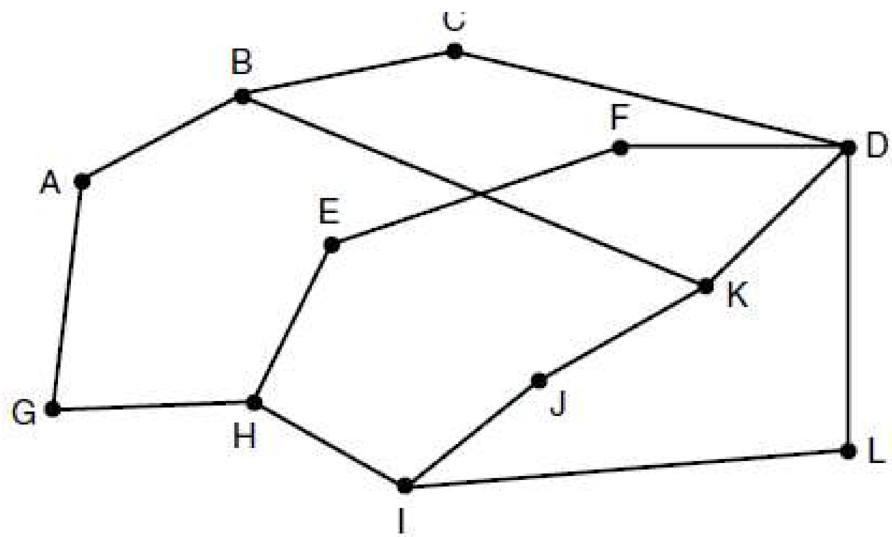
- The figure shows the network and sink tree



- A sink tree is not necessarily unique; other trees with the same path lengths may exist.
- A sink tree is directed acyclic graph with no loops.

- It is essential to determine a sink tree for a given network so each packet will be delivered within a finite and bounded number of hops.
- The optimality principle and the sink tree provide a benchmark against which other routing algorithms can be measured.

Draw a sink tree for router C in the following network for a group with members at routers A, B, C, D, E, F, I, and K.



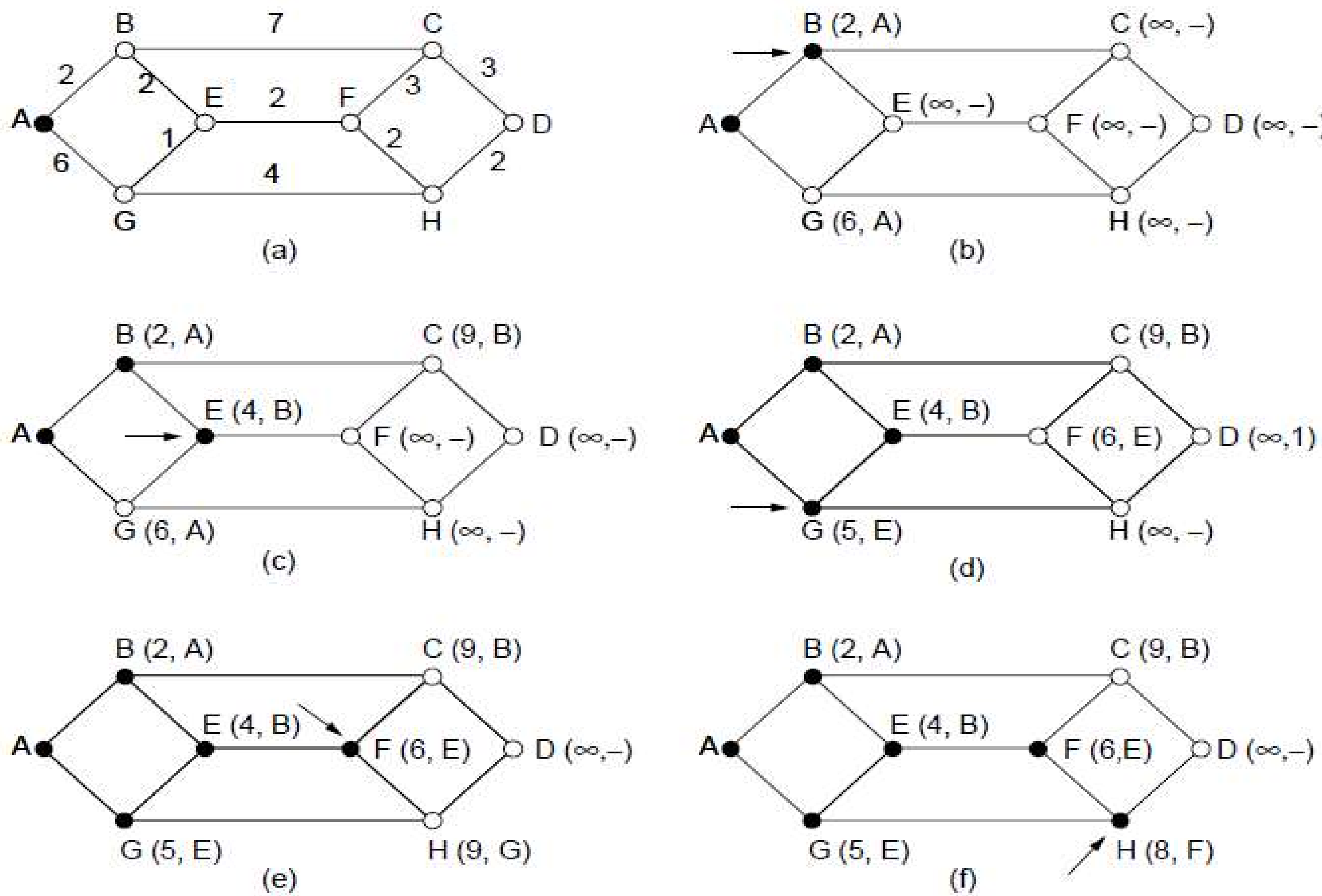
Routing Algorithms

1. Shortest Path
2. Flooding
3. Distance Vector Routing
4. Link State Routing
5. Hierarchical Routing
6. Broadcast Routing
7. Multicast Routing

Shortest Path Algorithm

- To compute optimal paths from a given network by constructing a graph with each node representing a router and an edge representing a communication link.
- *Metric Used to compute optimal path*:
  - No of hops from source to destination.
  - Geographic distance (in km or m).
  - Mean delay of standard test packet (measured by hourly run).
  - Bandwidth.
  - Average traffic.
  - Communication cost.
  - Measured delay.
- The routing algorithm computes "shortest" path according to any one of a metric or a combination of metrics

Dijkstra's shortest path algorithm



- Each node is labeled (in parentheses) with its distance from the source node along the best known path.
- The distances must be non-negative, as they will be if they are based on real quantities like bandwidth and delay.
- 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.

Flooding

- For every router to take routing decisions the complete picture of the network is essential.
- A simple technique is flooding in which every incoming packet is sent out on every outgoing line except the one it arrived on.
- This mechanism generates vast no of duplicate packets, so there should be some mechanism to damp the process.

Measures taken to prevent large number of packets being generated.

1. To add hop counter in the header of the packet.
   - The hop count gets decremented by one each time the packet moves to next hop.
   - The packet is dropped from the network when the hop counter becomes 0.
   - This hop counter is initialized to the length of path from source to destination, if unknown it is assigned with full diameter of the network.

This may result is exponential number of duplicate packets as routers may simply forward packets .

2. Routers should keep track of flooded packets.

- This is to avoid routers flooding packets for the second time.
- The source router puts its address and sequence number into the packet header and sends the packet to all of its neighbors.
- Each node maintains a list of the source address and sequence number of each broadcast packet it has already received, duplicated, and forwarded
- When a node receives a broadcast packet, it first checks whether the packet is in this list.
- If so, the packet is dropped; if not, the packet is duplicated and forwarded to all the node's neighbors (except the node from which the packet has just been received).
- To prevent the list from growing without bound, each list should be augmented by a counter, k, meaning that all sequence numbers through k have been seen.

Applications of flooding algorithm

1. It ensures that a packet is delivered to every node in the network- an effective way of broadcasting information e.g. wireless network.
2. It is tremendously robust. e.g., in a military network located in a war zone, flooding will find a path if one exists, to get a packet to its destination.
3. Flooding also requires little in the way of setup.
   – The routers only need to know their neighbors- this serves as building block for other routing algorithm.
4. Flooding can also be used as a metric against which other routing algorithms can be compared.

Distance Vector routing (Dynamic routing algorithm)

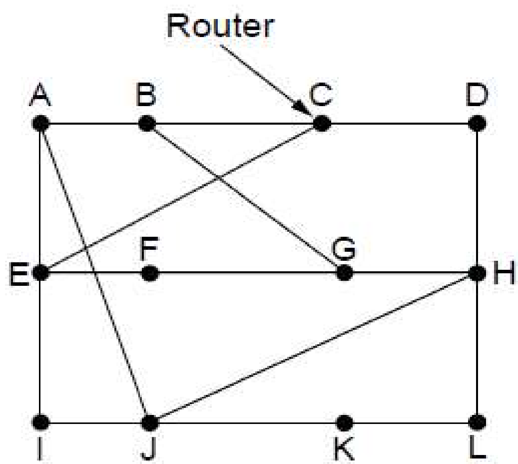
- The algorithm operates by having each router maintain a table (i.e., a vector) giving the best known distance to each destination and which link to use to get there.
- These tables are updated by exchanging information with the neighbors.
- Eventually, every router knows the best link to reach each destination.
- The algorithm is called by different name: Bellman Ford routing algorithm.

- The table is indexed by router in the network.
- Each entry has two parts the preferred outgoing line to use for that destination and an estimate of the distance to that destination.
- The distance might be measured as the number of hops or using another metric.
- The router is assumed to know the “distance” to each of its neighbors.
- Assuming that delay is used as a metric and that the router knows the delay to each of its neighbors.
- Once every *T msec, each router sends to* each neighbor a list of its estimated delays to each destination.
- It also receives a similar list from each neighbor.

Consider the following network in (a),

The first 4 columns in (b) are delays received from J's neighbours.

JA- 8, JI-10, JH-12, JK-6.(msec)

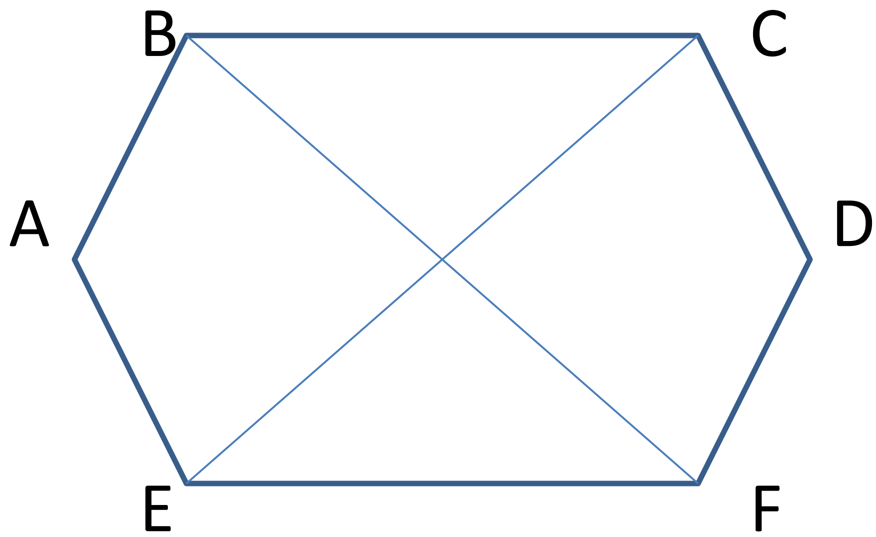


(a)

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

Vectors received from J's four neighbors

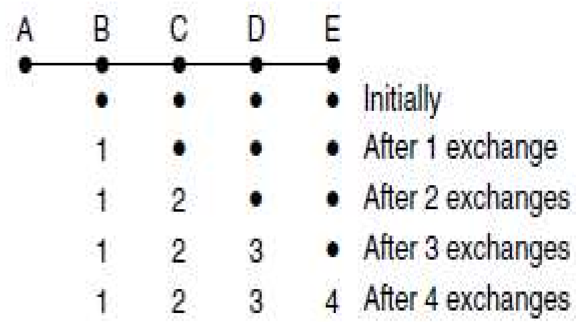
(b)



- Distance vector routing is used, and the following vectors have just come in to router C: from B: (5, 0, 8, 12, 6, 2); from D: (16, 12, 6, 0, 9, 10); and from E: (7, 6, 3, 9, 0, 4).
- The cost of the links from C to B, D, and E, are 6, 3, and 5, respectively. What is C's new routing table? Give both the outgoing line to use and the cost.

The count to infinity problem

- The settling of routes to best paths across the network is called **convergence.**
- Distance vector converges slowly in practice i.e., it reacts rapidly to good news, but leisurely to bad news.
- Consider five node(linear network). Metric used is number of hops.



A is down initially, when A is up.

- The good news is spreading at the rate of one hop per exchange.
- For a network whose longest path is of length N hops, with N exchanges everyone will know about newly revived links and routers.

Consider the case when A goes down.

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
		⋮			
	•	•	•	•	

This problem is known as the **count-to-infinity problem.**

Link State Routing

- Distance vector routing was used in the ARPANET until 1979, when it was replaced by link state routing.
- Variants of link state routing called IS-IS and OSPF are the routing algorithms that are most widely used inside large networks and the Internet today.
- Link state routing has 5 steps:
  - Discover its neighbors and learn their network addresses.
  - Set the distance or cost metric to each of its neighbors.
  - Construct a packet telling all it has just learned.
  - Send this packet to and receive packets from all other routers.
  - Compute the shortest path to every other router.

- The complete topology is distributed to every router.
- Then Dijkstra’s algorithm can be run at each router to find the shortest path to every other router.

Learning about the neighbours

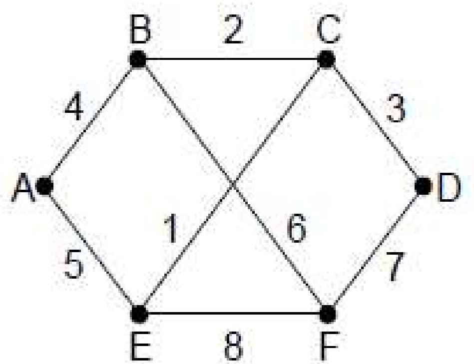
- When a router is booted, its first task is to learn who its neighbors are.
- It accomplishes this goal by sending a special HELLO packet on each point-to-point line.
- The router on the other end is expected to send back a reply giving its name.
- Care is taken to make sure that the names are globally unique.

Setting Link Costs

- The link state routing algorithm requires each link to have a distance or cost metric for finding shortest paths.
- The cost to reach neighbors can be set automatically, or configured by the network operator.
- A common choice is to make the cost inversely proportional to the bandwidth of the link.
- For example, 1-Gbps Ethernet may have a cost of 1 and 100-Mbps Ethernet a cost of 10. This makes higher-capacity paths better choices.
- Other metrics could be used such as distance (computed using ECHO messages and RTT)

Building Link State Packets

- Link state packet starts with the identity of the sender, followed by a sequence number and age (to be described later) and a list of neighbors.
- The cost to each neighbor is also given.



(a)

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

(b)

When to build link state packets?

1. Build packets periodically, that is, at regular intervals.
2. Build when some significant event occurs, such as a line or neighbor going down or coming back up again or changing its properties appreciably.

Distributing the Link State Packets

- Every routers must get all of the link state packets quickly and reliably.

Flooding

- To keep the flood in check, each packet contains a sequence number that is incremented for each new packet sent.
- Routers keep track of all the (source router, sequence) pairs they see.

Problems:

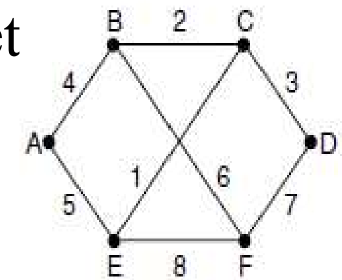
1. Size of sequence number- usually set to 32 bits.
2. Router crash- It may lose track of its sequence numbers.
3. Corruption of sequence numbers

- As a solution to these problems “age” field is included after sequence number and it is decremented once per second.
- When the age hits zero, the information from that router is discarded.
- The *Age field is also* decremented by each router during the initial flooding process, to make sure no packet can get lost and live for an indefinite period of time.
- A packet whose age is zero is discarded.

Some refinements:

- A link state packet is not immediately put to queue, instead it is put into holding area to wait for next link state packet, if both are same, older is discarded and new one is flooded.
- Every link state packet is acknowledged.

Data structure used by router B to store link state packet



Source	Seq.	Age	Send flags A	Send flags C	Send flags F	ACK flags A	ACK flags C	ACK flags 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	

Computing new routes

- Dijkstra's algorithm is run locally to construct the shortest paths to all possible destinations.
- The results of this algorithm tell the router which link to use to reach each destination.
- This information is installed in the routing tables, and normal operation is resumed.

Comparison LS and DV routing algorithms.

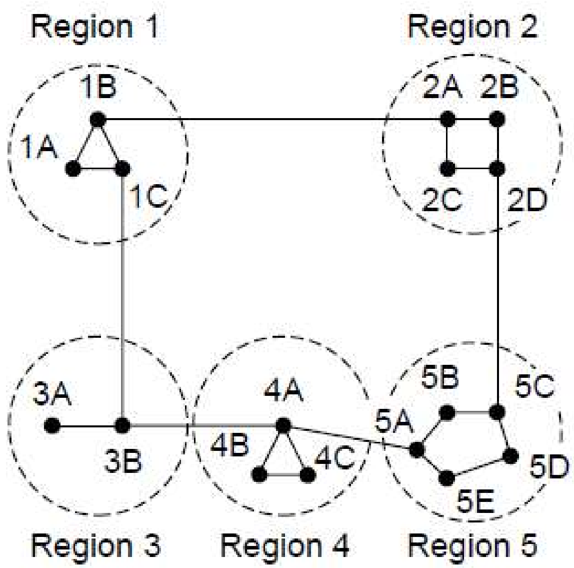
Parameter	Link State	Distance Vector
Memory	**More** A network of n routers with k neighbours the memory to store is proportional to kn	**Less** Only neighbours
Computation time	Increases with large networks	less
Convergence	Moderate	Converges slowly.

Protocols IS-IS (Intermediate System-Intermediate System) and OSPF(Open Shortest Path First) use this link state algorithm

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.
- The routing will have to be done hierarchically, as it is in the telephone network.

- With this routing, the routers are divided into **regions**
  - Each router knows all the details about how to route packets to destinations within its own region unaware of other regions.
- For larger networks regions are grouped into **clusters.**
- Clusters may be grouped to **zones**.
- Consider a large network- two-level hierarchy with five regions.



(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

The full routing table for router 1A has 17 entries, with hierarchical routing the table entries get reduced to 7.

(c)

Disadvantage:

- Increased path length.
- For example, the best route from *1A to 5C is via region 2,* but with hierarchical routing all traffic to region 5 goes via region 3, because that is better for most destinations in region 5.

How many levels of Hierarchy?

- If there are 720 routers- with no hierarchy 720 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.

- In general the optimal number of levels for an N router network is ln N, requiring a total of e ln N entries per router.
- For hierarchical routing with 4800 routers, what region and cluster sizes should be chosen to minimize the size of the routing table for a three-layer hierarchy?
- A good starting place is the hypothesis that a solution with k clusters of k regions of k routers is close to optimal, which means that k is about the cube root of 4800 (around 16). Use trial and error to check out combinations where all three parameters are in the general vicinity of 16.

- The minimum occurs at 15 clusters, each with 16 regions, each region having 20 routers, or one of the equivalent forms, e.g., 20 clusters of 16 regions of 15 routers. In all cases the table size is 15 + 16 + 20 = 51.

Broadcasting

- In some applications, hosts need to send messages to many or all other hosts, e.g., distributing whether reports, stock market updates of live radio programs.

Defn: Sending a packet to all destinations simultaneously is called **broadcasting.**

Various methods for broadcasting:

1. Send distinct packets to each destination.
2. Multidestination routing.
3. Flooding
4. Reverse path forwarding

1. Send distinct Packet to each destination

- The source to simply send a distinct packet to each destination.
- 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.

2. Multidestination routing

- 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.

- 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.
- Multidestination routing is like using separately addressed packets, except that when several packets must follow the same route.

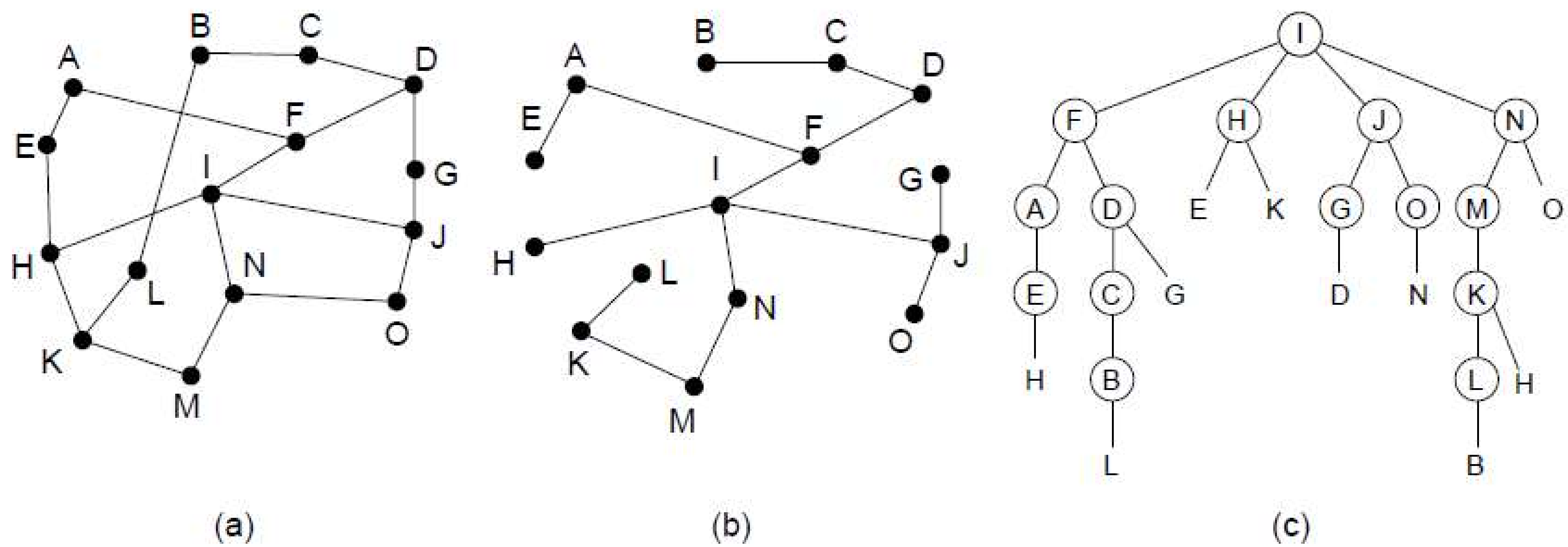
Flooding

- The algorithm uses sequence number per source to use links efficiently with a decision rule at routers that is relatively simple.

Reverse Path Forwarding(RPF)

- 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.
- It means 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 the broadcast packet arrived on a link other than the preferred one for reaching the source, the packet is discarded as a likely duplicate.

- 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, Each of these packets arrives on the preferred path to I.
- On the second hop, eight packets are generated, two by each of the routers that received a packet on the first hop.
- 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.
- With sink tree after five hops and 24 packets, the broadcasting terminates.

Advantages of reverse path forwarding vs flooding:

- The broadcast packet is sent only once in each direction.
- It requires the routers know how to reach all destinations without needing to remember sequence numbers or list all destinations in the packet.
- Easy to implement.

Using spanning tree

- To explicitly use he 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.*