ADITYA ENGINEERING COLLEGE (A)

Computer Networks

By

Dr. M. Vamsi Krishna
Professor

Dept of Computer Science and Engineering
Aditya Engineering College(A)
Surampalem.

Unit - IV

- **The 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 Networks,
- **Routing Algorithms:** The Optimality principle, Shortest path, Flooding, Distance vector, Link state, Hierarchical,
- **Congestion Control algorithms:** General principles of congestion control, Congestion prevention polices, Approaches to Congestion Control, Traffic Aware Routing, Admission Control, Traffic Throttling, Load Shedding,
- **Traffic Control Algorithm:** Leaky bucket & Token bucket.
- **Internet Working:** Network layer in the internet,
- **IP protocols:** IP Version 4,IP Version 6,Transition from IPV4 to IPV6, Comparison of IPV4 & IPV6,
- **Internet control protocols**: ICMP, ARP, DHCP

Network Layer

- The network layer is concerned with getting packets from the source all the way to the destination.
- To achieve its goals, the network layer must know about the topology of the network (i.e., the set of all routers and links)

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 Networks

Store and Forward Packet Switching

- The major components of the network are the ISP's equipment (routers connected by transmission lines), shown inside the shaded oval, and the customers' equipment, shown outside the oval.

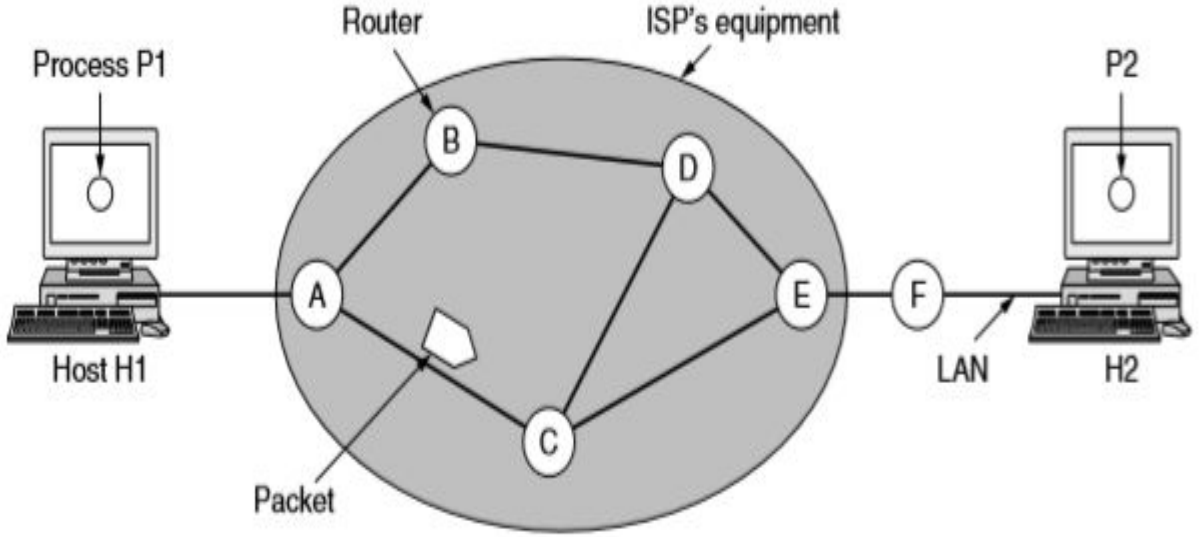


Figure 5-1. The environment of the network layer protocols.

- *Host H1 is directly connected to one of the ISP's routers, A, perhaps as a home computer that is plugged into a DSL modem.*
- *In contrast, H2 is on a LAN, which might be an office Ethernet, with a router, F, owned and operated by the customer.*
- *This router has a leased line to the ISP's equipment. We have shown F as being outside the oval because it does not belong to the ISP.*

This equipment is used as follows.

- 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.

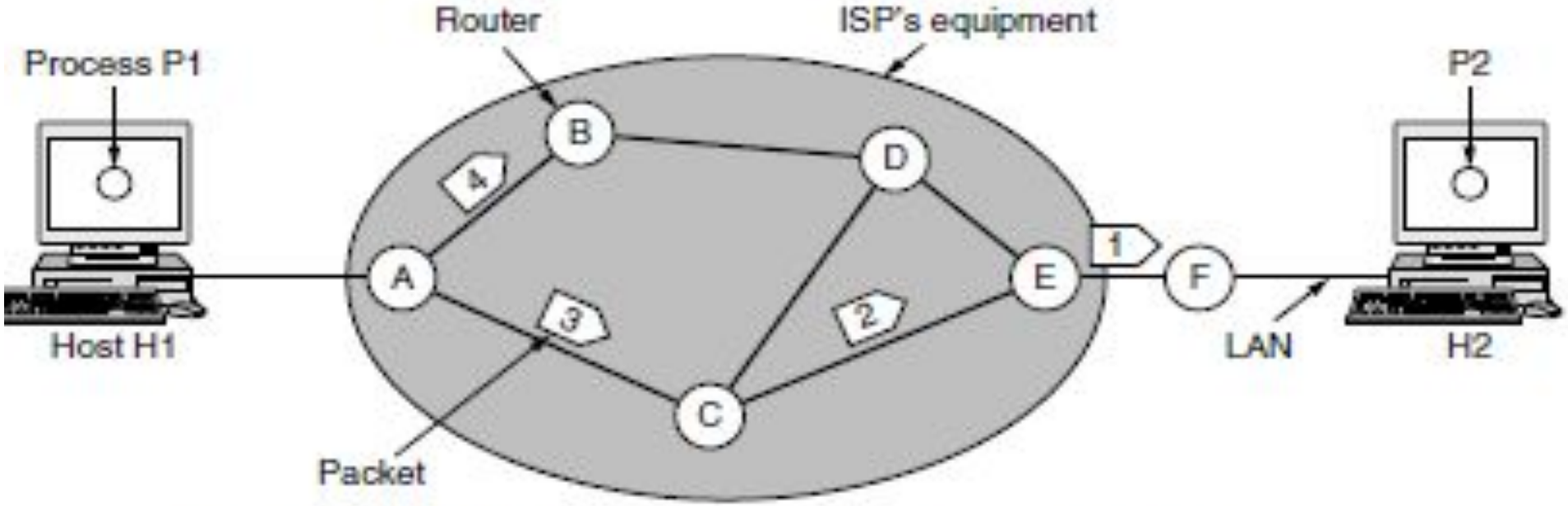
Services Provided to the Transport Layer

- The network layer provides services to the transport layer at the network layer/transport layer interface. Goals of the services:

1. The services 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:

- If connectionless service is offered, packets are injected into the network individually and routed independently of each other.
- The packets are frequently called **datagrams** and the network is called a datagram network.
- If connection-oriented service is used, a path from the source router all the way to the destination router must be established before any data packets can be sent. This connection is called Virtual Circuit and the network is called Virtual Circuit Network.
- The algorithm that manages the tables and makes the routing decisions is called the **routing algorithm**



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	B
F	B

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

Figure 5-2. Routing within a datagram network.

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

- *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 using some point-to-point protocol, for example, PPP.*
- *At this point the ISP takes over. 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. Only directly connected lines can be used.*
- *For example, in Figure, A has only two outgoing lines—to B and to C—so every incoming packet must be sent to one of these routers, even if the ultimate destination is to some other router.*
- *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 and had their checksums verified. Then each packet is forwarded according to A's table, onto the outgoing link to C within a new frame.*
- *Packet 1 is then forwarded to E and then to F. When it gets to F, it is sent within a frame over the LAN to H2.*
- *Packets 2 and 3 follow the same route.*
- *However, something different happens to packet 4. When it gets to A it is sent to router B, even though it is also destined for F.*
- *For some reason, A decided to send packet 4 via a different route than that of the first three packets.*
- *Perhaps it has learned of a traffic jam somewhere along the ACE path and updated its routing table, as shown under the label ''later.''*
- *The algorithm that manages the tables and makes the routing decisions is called the **routing algorithm**.*

Implementation of Connection-Oriented Service:

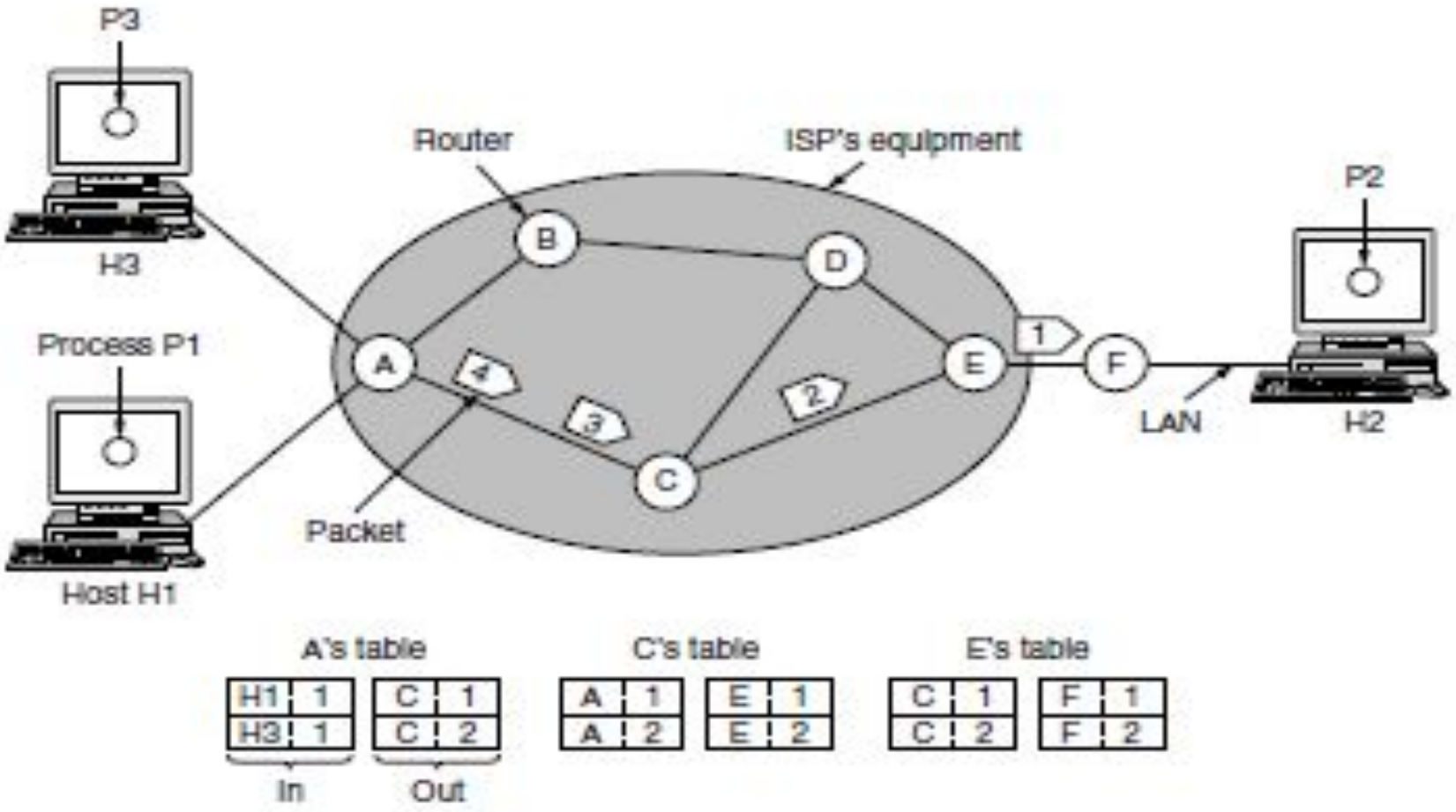


Figure 5-3. Routing within a virtual-circuit network.

- *For connection-oriented service, we need a virtual-circuit network.*
- *The idea behind virtual circuits is to avoid having to choose a new route for every packet sent.*
- *Instead, 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 Fig. 5-3. Here, 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.*
- *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 (because it is initiating the connection and this is its only connection) 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 traffic for 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 (MultiProtocol Label Switching)**.*

Comparison of Virtual Circuits 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 Optimality principle
- Shortest path
- Flooding
- Distance vector
- Link state
- Hierarchical

Routing Algorithms

- The algorithm that manages the tables and makes the routing decisions is called the **routing algorithm.**
- Correctness, simplicity, robustness, stability, fairness, and efficiency are desirable properties in a routing algorithm.
- 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.
- Instead, the choice of the route *is computed in advance, offline,* and downloaded to the routers when the network is booted. This procedure is sometimes called **static routing.**
- **Adaptive algorithms** change their routing decisions to reflect changes in the topology, and sometimes changes in the traffic as well.
- These algorithms get information locally, from adjacent routers, or from all routers), when they change the routes.
- That's why there are also known as Dynamic routing algorithms.

BASIS FOR COMPARISON	STATIC ROUTING	DYNAMIC ROUTING
Configuration	It is Manually configure.	It is automatically configure with the help of routing protocols.
Routes	Routes are user defined.	Routes are updated according to change in topology.
Implemented in	It is implemented in Small networks.	It is implemented in Large networks.
Link failure	Link failure obstructs the rerouting.	Link failure doesn't affect the rerouting.
Security	Static routing provides high security.	Dynamic routing is less secure due to sending broadcasts and multicasts.
Additional resources	Additional resources are not required in static routing.	Dynamic routing needs additional resources to store the information.

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 route.
- The set of optimal routes to a particular node forms a sink tree.
- 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.
- 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.

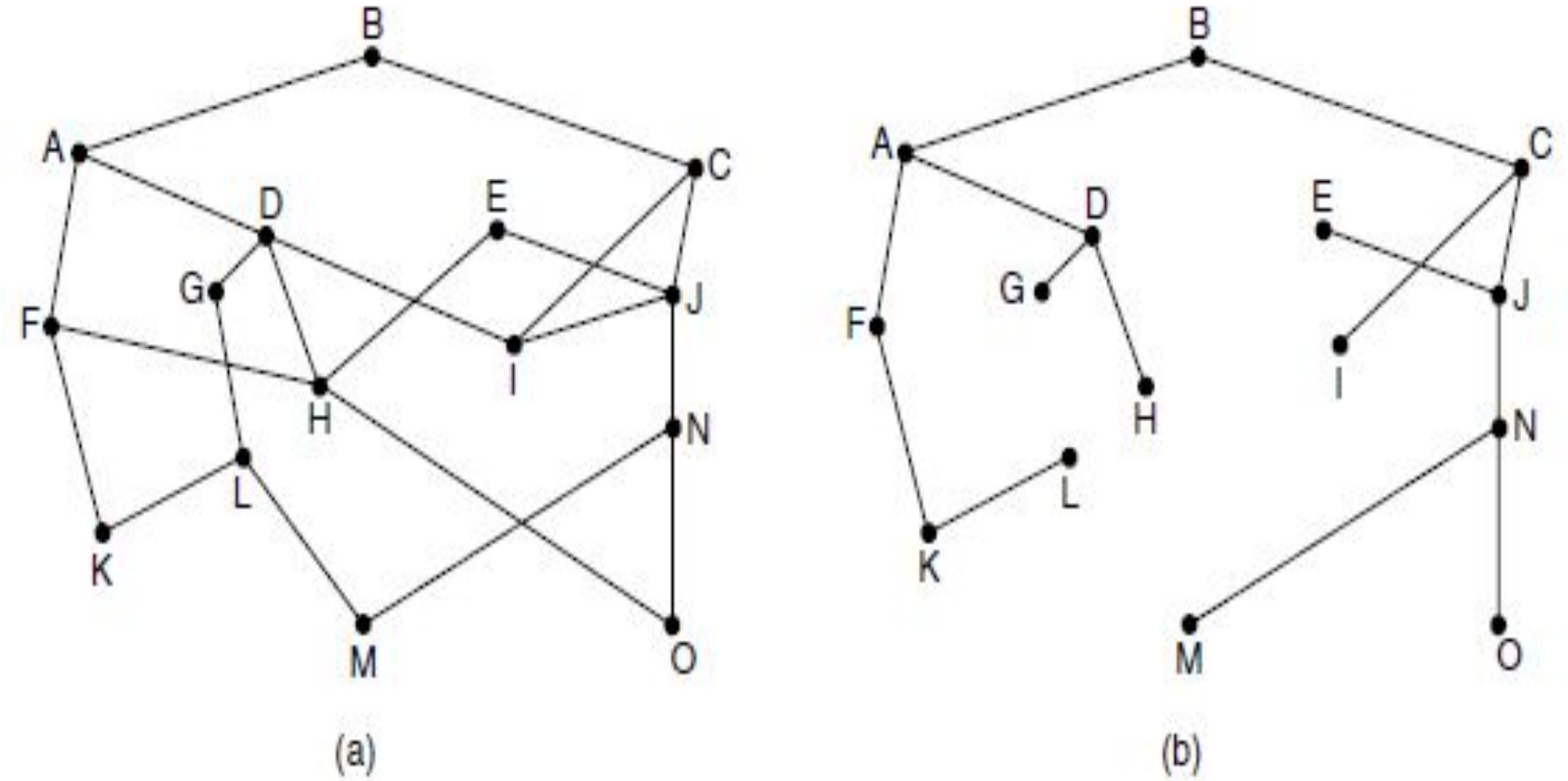


Figure 5-6. (a) A network. (b) A sink tree for router *B*.

Shortest Path Algorithm

- The idea is to build a graph of the network, with each node of the graph representing a router and each edge 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.
- Dijkstra finds the shortest paths between a source and all destinations in the network.
- 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.

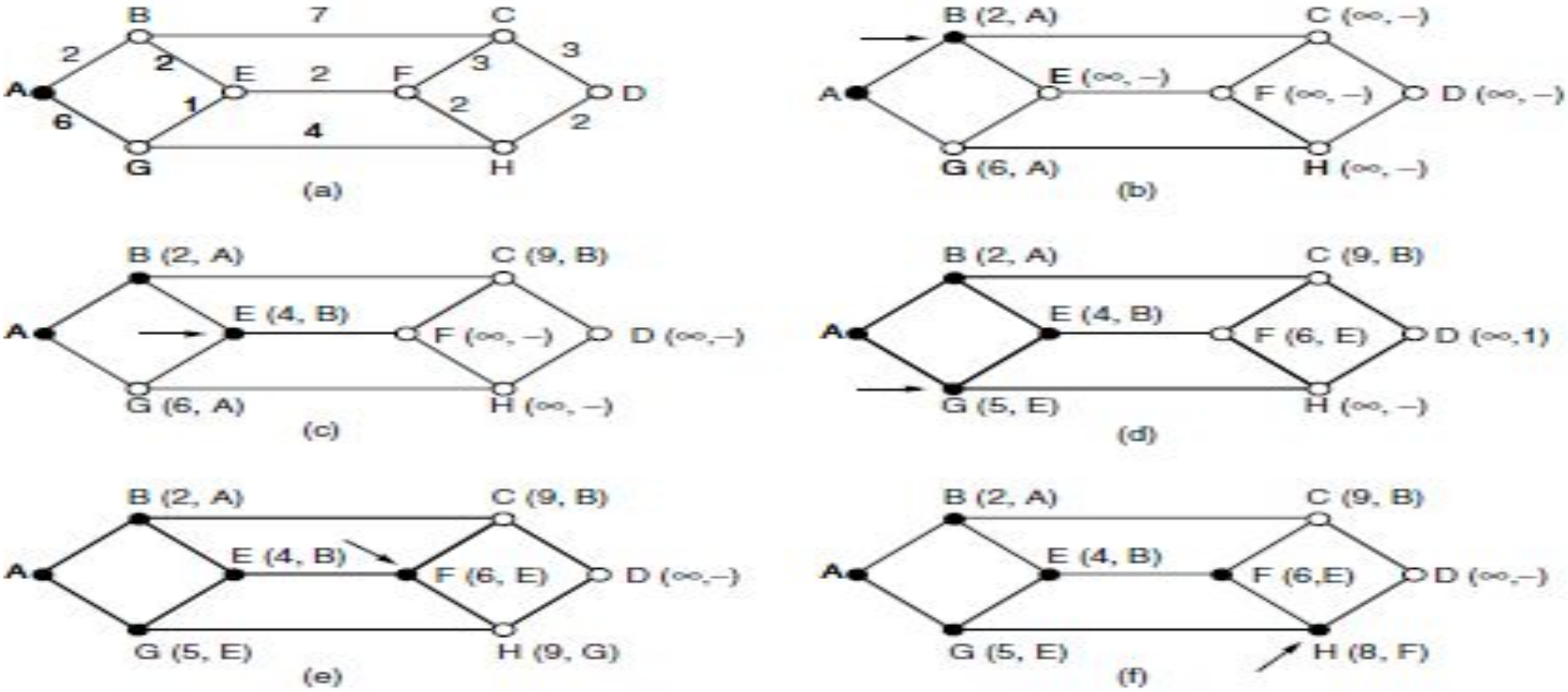


Figure 5-7. The first six steps used in computing the shortest path from *A* to *D*. The arrows indicate the working node.

Flooding

- Flooding is a local technique in which every incoming packet is sent out on every outgoing line except the one it arrived on.
- Packet send by node to every neighbor.
- Flooding generates vast numbers of duplicate packets that reach the destination.
- All nodes will be visited.
- All possible routes will be tried.
- How to eliminate duplicates?

  --Hop Counter.

  --Sequence Number.

Hop Counter:

- Hop counter contained in the header of each packet is decremented at each hop and the packet will be discarded when the counter reaches zero.
- Ideally, the hop counter should be initialized to the length of the path from source to destination.

Sequence Number:

- Source router will put a sequence number in each packet it receives from its hosts.
- Each router then needs a list per source router telling which sequence numbers originating at that source have already been seen.
- If an incoming packet is on the list, it is not flooded.

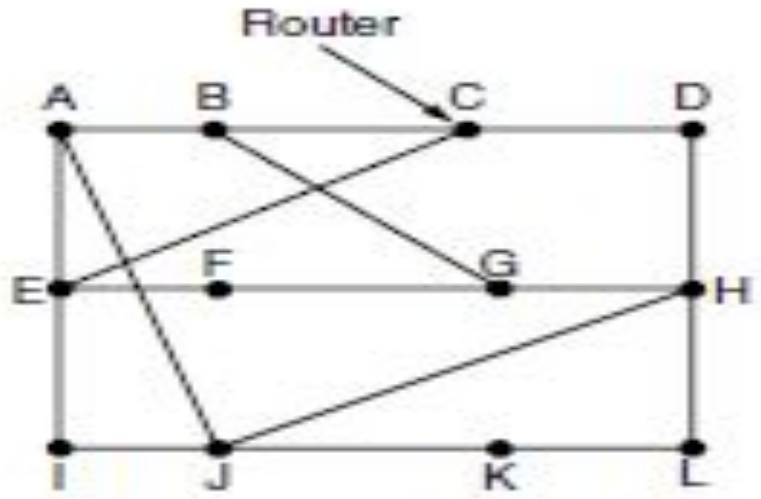
Distance Vector Routing

- Computer networks generally use dynamic routing algorithms that are more complex than flooding, but more efficient because they find shortest paths for the current topology.
- A distance vector routing algorithm operates by having each router maintain a table (i.e., a vector) giving the best-known distance to each destination and which link to use to get there. These tables are updated by exchanging information with the neighbors.
- Eventually, every router knows the best link to reach each destination.
- Distance vector routing algorithm is also known as **Bellman-Ford routing algorithm.**
- In routing table, 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 propagation delay.
- Part (a) shows a network. The first four columns of part (b) show the delay vectors received from the neighbors of router *J*. A claims to have a 12-msec delay to B, a 25-msec delay to C, a 40- msec delay to D, etc.
- Suppose that J has measured or estimated its delay to its neighbors, A, I, H, and K, as 8, 10, 12, and 6 msec, respectively.
- Consider how J computes its new route to router G. It knows that it can get to A in 8 msec, and furthermore 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.
- From J to G JA=8

JB= A -□JA+AB=8+12=20 JC= A□JA+AC=8+25=33
H □JH+HB=12+31=43 H□JH+HC=12+19=31
I □JI+IB=10+36=46 I□JI+IC=10+18=28
K □JK+KB=6+28=34 K□JK+KC=8+36=42



(a)

New estimated delay from J

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	

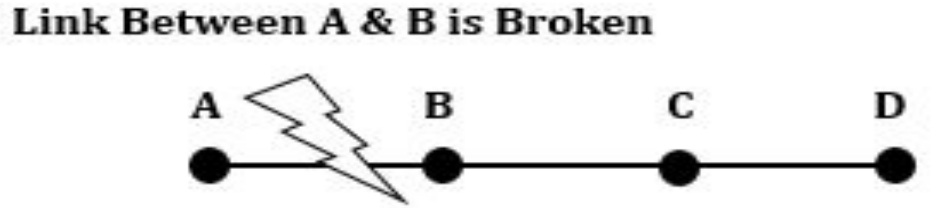
Vectors received from J's four neighbors

(b)

Figure 5-9. (a) A network. (b) Input from *A*, *I*, *H*, *K*, and the new routing table for *J*.

The Count – to – infinity Problem

- Distance vector routing is useful as a simple technique by which routers can collectively compute shortest paths, but it has a serious drawback in practice: although it converges to the correct answer, it may do so slowly.
- In particular, it reacts rapidly to good news, but leisurely to bad news.



	A	B	C	D
A	0,-	1,A	2,B	3,C
B	1,B	0,-	1,B	2,C
C	2,B	1,C	0,-	1,C
D	3,B	2,C	1,D	0,-

- Imagine that the link between A and B is cut.
- At this time, B corrects its table.
- After a specific amount of time, routers exchange their tables, and so B receives C's routing table.
- Since C doesn't know what has happened to the link between A and B, it says that it has a link to A with the weight of 2 (1 for C to B, and 1 for B to A -- it doesn't know B has no link to A).
- B receives this table and thinks there is a separate link between C and A, so it corrects its table and changes infinity to 3 (1 for B to C, and 2 for C to A, as C said).
- Once again, routers exchange their tables.
- When C receives B's routing table, it sees that B has changed the weight of its link to A from 1 to 3, so C updates its table and changes the weight of the link to A to 4 (1 for C to B, and 3 for B to A, as B said).
- This process loops until all nodes find out that the weight of link to A is infinity.

	B	C	D
Sum of Weight to A after link cut	∞, A	2, B	3, C
Sum of Weight to A after 1st updating	3, C	2, B	3, C
Sum of Weight to A after 2nd updating	3, C	4, B	3, C
Sum of Weight to A after 3rd updating	5, C	4, B	5, C
Sum of Weight to A after 4th updating	5, C	6, B	5, C
Sum of Weight to A after 5th updating	7, C	6, B	7, C
Sum of Weight to A after nth updating	.....	....	....
∞	∞	∞	∞

Link State Routing

- *Distance vector routing was used in the ARPANET until 1979, when it was replaced by link state routing.*
- *The primary problem that caused its demise was that the algorithm often took too long to converge after the network topology changed (due to the count-to-infinity problem).*
- *Consequently, it was replaced by an entirely new algorithm, now called link state routing*
- *The idea behind link state routing is fairly simple and can be stated as five parts.*

- *Each router must do the following things to make it work:*

1. *Discover its neighbours and learn their network addresses.*
2. *Set the distance or cost metric to each of its neighbours.*
3. *Construct a packet telling all it has just learned.*
4. *Send this packet to and receive packets from all other routers.*
5. *Compute the shortest path to every other router.*

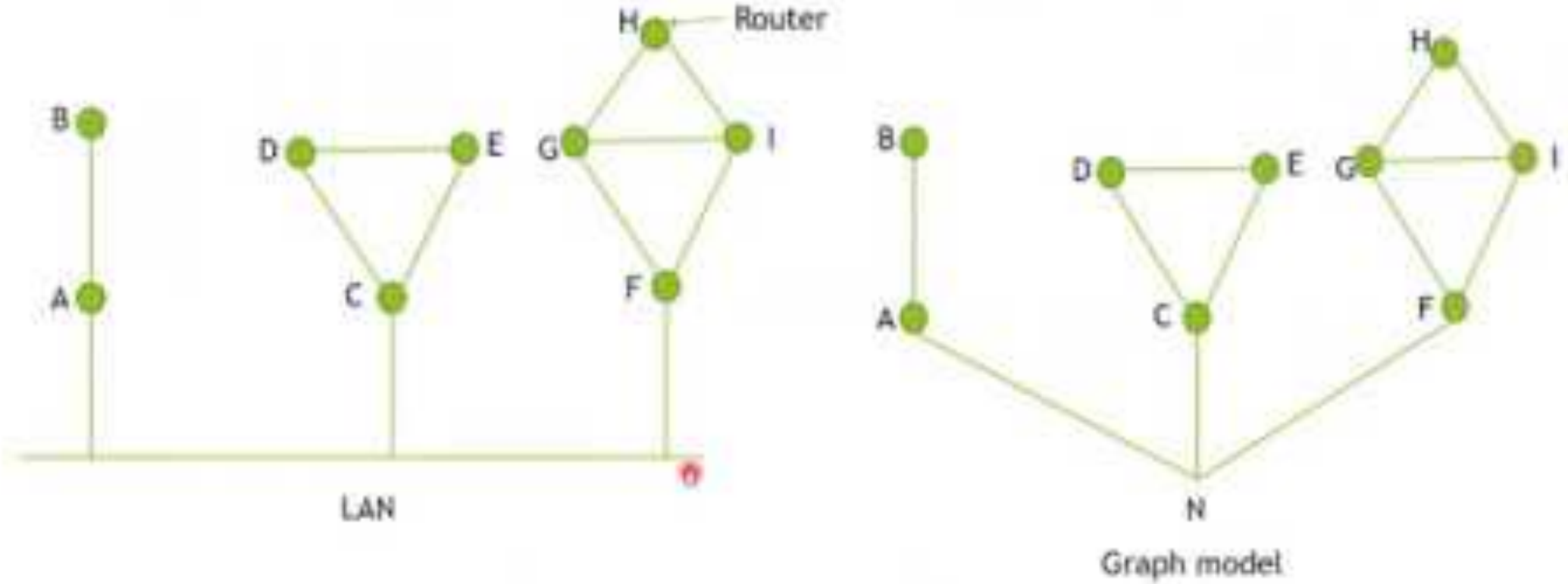
- *In effect, the complete topology is distributed to every router.*
- *Then Dijkstra's algorithm can be run at each router to find the shortest path to every other router.*

Step 1: Learning about neighbours

- *When a router is booted, its first task is to learn who its neighbour's 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.*
- *These names must be globally unique because when a distant router later hears that three routers are all connected to F, it is essential that it can determine whether all three mean the same F.*

- *When two or more routers are connected by a broadcast link (e.g., a switch, ring, or classic Ethernet), the situation is slightly more complicated.*
- *Figure illustrates a broadcast LAN to which three routers, A, C, and F, are directly connected.*
- *Each of these routers is connected to one or more additional routers, as shown.*
- *The broadcast LAN provides connectivity between each pair of attached routers.*
- *However, modelling the LAN as many point-to-point links increases the size of the topology and leads to wasteful messages.*
- *A better way to model the LAN is to consider it as a node itself, as shown in Fig.*
- *Here, we have introduced a new, artificial node, N, to which A, C, and F are connected.*
- *One designated router on the LAN is selected to play the role of N in the routing protocol.*
- *The fact that it is possible to go from A to C on the LAN is represented by the path ANC here.*

Discover its neighbours and learn their network addresses (Learning about the neighbours)



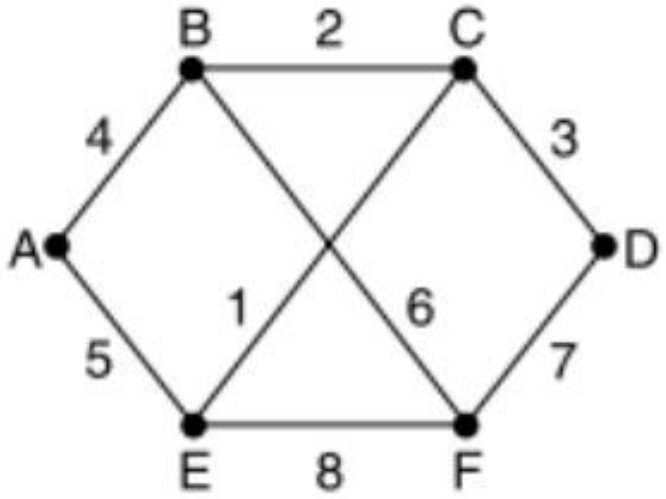
- When a router is booted it sends a special HELLO packet to each of its neighbours.

Step 2: *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 neighbour's can be set automatically or configured by the network operator.*

Step 3: *Building Link State Packets*

- *Once the information needed for the exchange has been collected, the next step is for each router to build a packet containing all the data.*
- *The packet starts with the identity of the sender, followed by a sequence number and age and a list of neighbours. The cost to each neighbour is also given.*
- *The corresponding link state packets for all six routers are shown in Figure.*
- *Building the link state packets is easy. The hard part is determining when to build them.*
- *One possibility is to build them periodically, that is, at regular intervals.*
- *Another possibility is to build them when some significant event occurs, such as a line or neighbours going down or coming back up again or changing its properties appreciably.*



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

Step 4: *Distributing the Link State Packets*

- *The trickiest part of the algorithm is distributing the link state packets. All of the routers must get all of the link state packets quickly and reliably.*
- *If different routers are using different versions of the topology, the routes they compute can have inconsistencies such as loops, unreachable machines, and other problems.*
- *The fundamental idea is to use flooding to distribute the link state packets to all routers.*
- *To keep the flood in check, each packet contains a sequence number that is incremented for each new packet sent. Routers keep track of all the (source router, sequence) pairs they see.*
- *When a new link state packet comes in, it is checked against the list of packets already seen. If it is new, it is forwarded on all lines except the one it arrived on. If it is a duplicate, it is discarded.*
- *If a packet with a sequence number lower than the highest one seen so far ever arrives, it is rejected as being obsolete as the router has more recent data.*

- *This algorithm has a few problems, but they are manageable.*
- *First, if the sequence numbers wrap around, confusion will reign. The solution here is to use a 32-bit sequence number. With one link state packet per second, it would take 137 years to wrap around, so this possibility can be ignored.*
- *Second, if a router ever crashes, it will lose track of its sequence number. If it starts again at 0, the next packet it sends will be rejected as a duplicate.*
- *Third, if a sequence number is ever corrupted and 65,540 is received instead of 4 (a 1-bit error), packets 5 through 65,540 will be rejected as obsolete, since the current sequence number will be thought to be 65,540.*
- *The solution to all these problems is to include the age of each packet after the sequence number and decrement it once per second. When the age hits zero, the information from that router is discarded.*
- *Normally, a new packet comes in, say, every 10 sec, so router information only times out when a router is down (or six consecutive packets have been lost, an unlikely event).*
- *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 to this algorithm make it more robust.*
- *When a link state packet comes in to a router for flooding, it is not queued for transmission immediately.*
- *Instead, it is put in a holding area to wait a short while in case more links are coming up or going down.*
- *If another link state packet from the same source comes in before the first packet is transmitted, their sequence numbers are compared. If they are equal, the duplicate is discarded. If they are different, the older one is thrown out.*
- *To guard against errors on the links, all link state packets are acknowledged.*

- *The data structure used by router B for the network shown in Fig. 5-12(a) is depicted in Fig. 5-13. Each row here corresponds to a recently arrived, but as yet not fully processed, link state packet.*
- *The table records where the packet originated, its sequence number and age, and the data.*
- *In addition, there are send and acknowledgement flags for each of B's three links (to A, C, and F, respectively).*
- *The send flags mean that the packet must be sent on the indicated link. The acknowledgement flags mean that it must be acknowledged there.*

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	

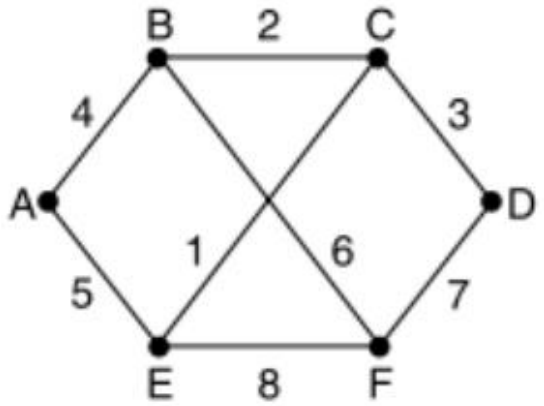


Figure 5-13. The packet buffer for router *B* in Fig. 5-12(a).

Step 5: Computing New Routes

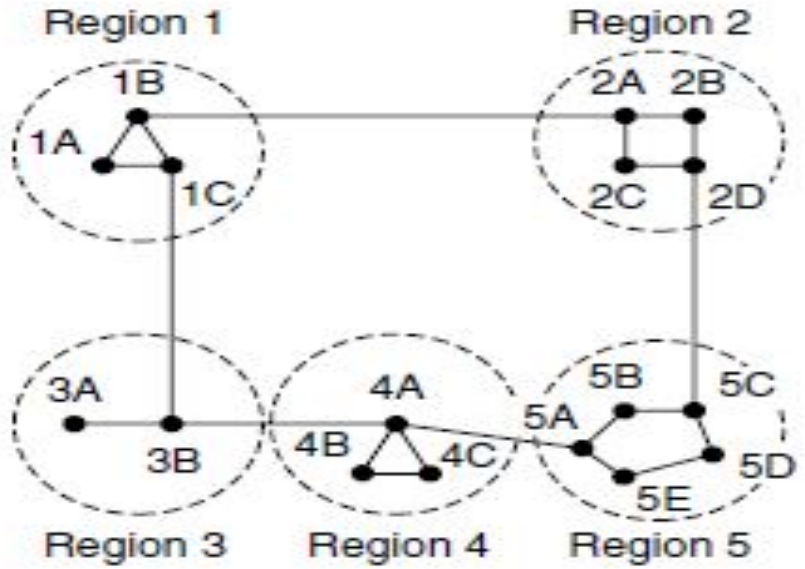
- *Once a router has accumulated a full set of link state packets, it can construct the entire network graph because every link is represented.*
- *Every link is, in fact, represented twice, once for each direction. The different directions may even have different costs.*
- *The shortest-path computations may then find different paths from router A to B than from router B to A.*
- *Now Dijkstra's algorithm can be 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.*
- *A general comment on routing algorithms is also in order.*
- *Link state, distance vector, and other algorithms rely on processing at all the routers to compute routes. Problems with the hardware or software at even a small number of routers can wreak havoc across the network.*

Hierarchical Routing:

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

- Figure 5-14 gives a quantitative example of routing in a two-level hierarchy with five regions.
- The full routing table for router 1A has 17 entries, as shown in Fig. 5-14(b).
- When routing is done hierarchically, as in Fig. 5-14(c), there are entries for all the local routers, as before, but all other regions are condensed into a single router, so all traffic for region 2 goes via the 1B-2A line, but the rest of the remote traffic goes via the 1C-3B line.
- Hierarchical routing has reduced the table from 17 to 7 entries.
- There is a penalty to be paid: 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.

- 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.
- **Kamoun and Kleinrock (1979) discovered that the optimal number of levels for an N router network is ln N requiring a total of e ln N entries per router.**



(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)

Figure 5-14. Hierarchical routing.

Broadcast Routing

- In some applications, hosts need to send messages to many or all other hosts.
- For example, a service distributing weather reports, stock market updates, or live radio programs might work best by sending to all machines and letting those that are interested read the data.
- Sending a packet to all destinations simultaneously is called **broadcasting.**
- The first broadcasting method is simply sending distinct packet to each destination.
- This method wastes the bandwidth and it is slow , but it also requires the source to have a complete list of all destinations.
- The second broadcast method is **multidestination routing**, in which each packet contains a list of destinations.

- The router generates a new copy of the packet for each output line.
- The next broadcast routing technique is flooding.
- Flooding generates too many packets and consumes too much bandwidth.
- The fourth broadcast algorithm uses sink trees for initiating broadcast.
- The idea for broadcasting is **reverse path forwarding**.
- 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.

- Sink trees are spanning trees.
- A spanning tree is a subset of the network that includes all the routers but contains no loops.
- 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 principal advantage of reverse path forwarding is that it is efficient while being easy to implement.

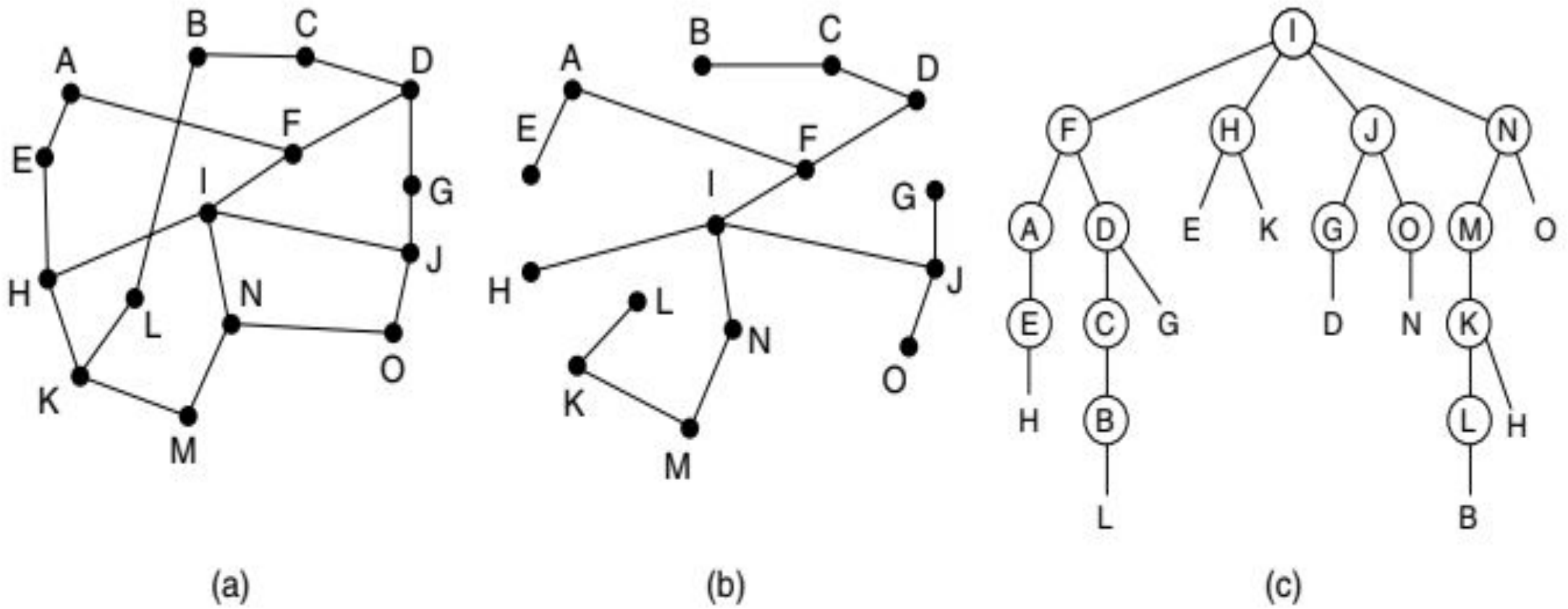


Figure 5-15. Reverse path forwarding. (a) A network. (b) A sink tree. (c) The tree built by reverse path forwarding.

- 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 and five of these arrive along the preferred line.
- After five hops and 24 packets, the broadcasting terminates, compared with four hops and 14 packets had the sink tree been followed exactly.

Multicast Routing:

- Broadcasting a packet is wasteful, if the receivers are not supposed to see it.
- Sending a message to a group is called multicasting, and the routing algorithm used is called multicast routing.
- Multicasting requires group management, need to create and destroy groups.
- To do multicast routing each router computes a spanning tree covering all other routers.
- In the below figure we have two groups group1 and group2.
- Some routers are attached to one or more groups.
- Figure (b) is spanning tree for the left most router.

- When a process sends a multicast packet to a group, the first router examines its spanning tree and prunes it, removing all lines that do not lead to hosts that are members of the group.
- In our example, Figure (c) shows the pruned spanning tree for group 1.
- Figure(d) shows the pruned spanning tree for group 2. Multicast packets are forwarded only along the appropriate spanning tree.

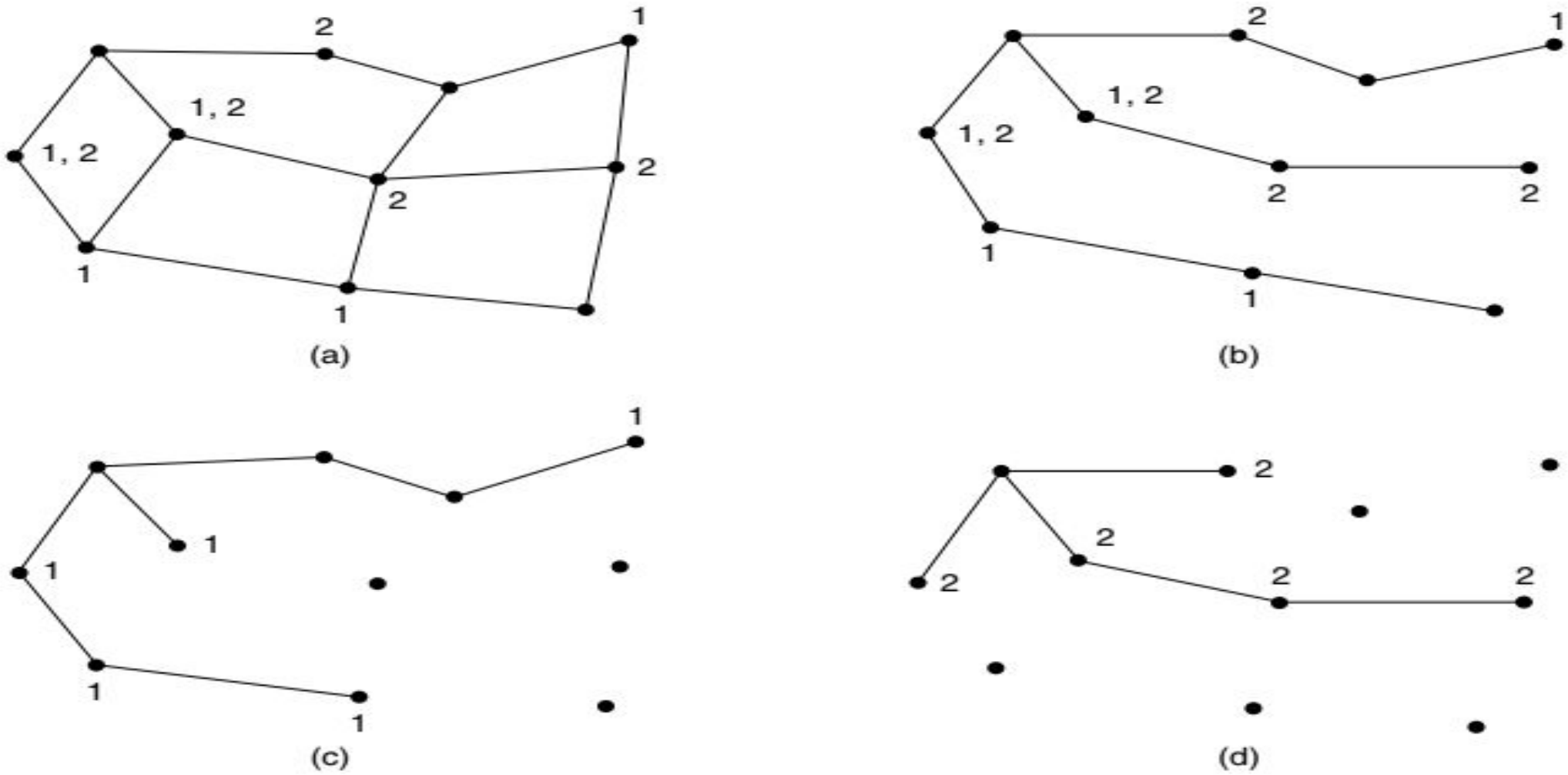


Figure 5-16. (a) A network. (b) A spanning tree for the leftmost router. (c) A multicast tree for group 1. (d) A multicast tree for group 2.

Congestion Control Algorithms

- *General principles of congestion control*
- *Congestion prevention polices*
- *Approaches to Congestion Control*
- *Traffic Aware Routing*
- *Admission Control*
- *Traffic Throttling*
- *Load Shedding*
- *Traffic Control Algorithm*
- *Leaky bucket & Token bucket.*

Congestion Control Algorithms:

- When too many packets are present in the subnet, performance degrades. This situation is called congestion.
- Congestion can be brought on by several factors. If suddenly, streams of packets begin arriving on three or four input lines and all need the same output line, a queue will build up. If there is insufficient memory to hold all of them, packets will be lost.
- Slow processors can also cause congestion.
- Similarly, low-bandwidth lines can also cause congestion.

General Principles of Congestion Control:

- Complex problems of Computer Networks can be viewed from a control theory point of view. This approach leads to dividing all solutions into two groups: open loop and closed loop.

Open Loop Congestion Control

- Open loop congestion control policies are applied to prevent congestion before it happens. The congestion control is handled either by the source or the destination.
- In contrast, **closed loop** solutions are based on the concept of a feedback loop. This approach has three parts when applied to congestion control:

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

Monitor the system to detect when and where congestion occurs

- The percentage of all packets discarded for lack of buffer space, the average queue lengths.
- The number of packets that time out and are retransmitted, the average packet delay, and the
- Standard deviation of packet delay.

Pass this information to places where action can be taken

- Send a packet to the traffic source or sources, announcing the problem.
- A bit or field can be reserved in every packet for routers to fill in whenever congestion gets above some threshold level.
- Sending probe packets.

Adjust system operation to correct the problem:

- The presence of congestion means that the load is (temporarily) greater than the resources (in part of the system) can handle.
- Two solutions come to mind: increase the resources or decrease the load.
- Increasing the resources is not always possible, the only way then to beat back the congestion is to decrease the load.

Congestion Prevention Policies:

Policies adopted by open loop congestion control :

Retransmission Policy : It is the policy in which retransmission of the packets are taken care. To prevent congestion, retransmission timers must be designed to prevent congestion and also able to optimize efficiency.

Window Policy : The type of window at the sender side may also affect the congestion. Selective repeat window should be adopted as it sends the specific packet that may have been lost because GoBackN increases duplication of packets.

Discarding Policy : A good discarding policy adopted by the routers will help the routers from preventing congestion.

Acknowledgment Policy : Since acknowledgement are also the part of the load in network, the acknowledgment policy imposed by the receiver may also affect congestion.

Admission Policy : In admission policy a mechanism should be used to prevent congestion.

Closed loop congestion control:

- It is used to treat or alleviate congestion after it happens. Several techniques are used by different protocols; some of them are:

Backpressure :

- Backpressure is a node-to-node congestion control technique that propagate in the opposite direction of data flow. The backpressure technique can be applied only to virtual circuit where each node has information of its above upstream node.

Choke Packet Technique :

- Choke packet technique is applicable to both virtual networks as well as datagram subnets. A choke packet is a packet sent by a node to the source to inform it of congestion. Whenever the resource utilization exceeds the threshold value which is set by the administrator, the router directly sends a choke packet to the source giving it a feedback to reduce the traffic.

Implicit signaling:

- There is no communication between the congested nodes and the source. The source guesses that there is congestion in a network.

Explicit signaling:

- If a node experiences congestion it can explicitly sends a packet to the source or destination to inform about congestion. The difference between choke packet and explicit signaling is that the signal is included in the packets that carry data rather than creating different packet.

Approaches to Congestion Control

- *The presence of congestion means that the load is (temporarily) greater than the resources (in a part of the network) can handle.*
- *Two solutions come to mind: increase the resources or decrease the load.*

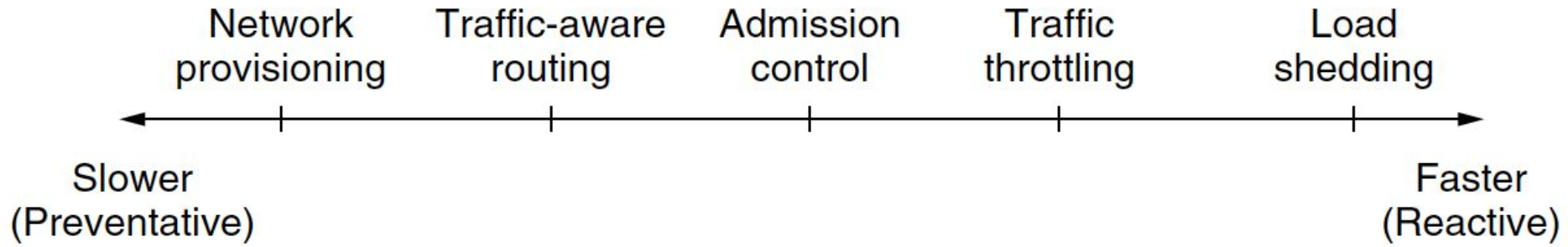


Figure 5-22. Timescales of approaches to congestion control.

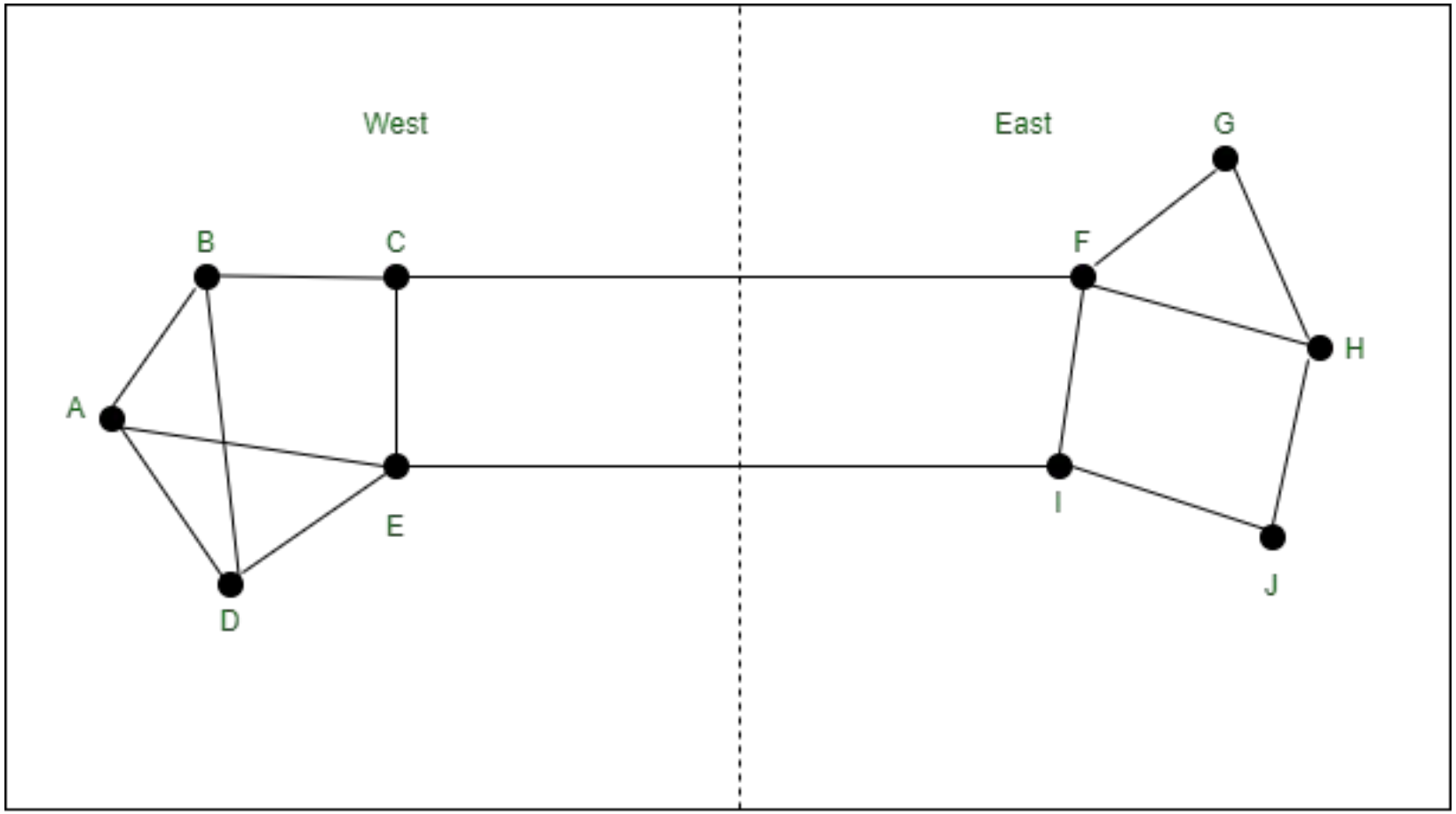
Network Provisioning

- *The most basic way to avoid congestion is to build a network that is well matched to the traffic that it carries.*
- *If there is a low-bandwidth link on the path along which most traffic is directed, congestion is likely.*
- *Sometimes resources can be added dynamically when there is serious congestion, for example, turning on spare routers or enabling lines that are normally used only as backups (to make the system fault tolerant) or purchasing bandwidth on the open market.*
- *More often, links and routers that are regularly heavily utilized are upgraded at the earliest opportunity. This is called* ***provisioning*** *and happens on a time scale of months, driven by long-term traffic trends.*

Traffic - Aware Routing

- *To make the most of the existing network capacity, routes can be tailored to traffic patterns that change during the day as network users wake and sleep in different time zones.*
- *For example, routes may be changed to shift traffic away from heavily used paths by changing the shortest path weights.*
- *Some 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**. Splitting traffic across multiple paths is also helpful.*

- *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.*


West
East
G
B
C
F
H
A
E
I
J
D

- *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.*
- *Given these difficulties, in the Internet routing protocols do not generally adjust their routes depending on the load.*
- *Instead, adjustments are made outside the routing protocol by slowly changing its inputs.*
- *This is called **traffic engineering**.*

Admission Control

- *Sometimes it is not possible to increase capacity of a network.*
- *The only way then to beat back 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**.*
- *Traffic is often described in terms of its rate and shape. The problem of how to describe it in a simple yet meaningful way is difficult because traffic is typically bursty.*

- *For example, traffic that varies while browsing the Web is more difficult to handle than a streaming movie with the same long-term throughput because the bursts of Web traffic are more likely to congest routers in the network.*
- *A commonly used descriptor that captures this effect is the **leaky bucket** or **token bucket**.*
- *Admission control can also be combined with traffic-aware routing by considering routes around traffic hotspots as part of the setup procedure.*

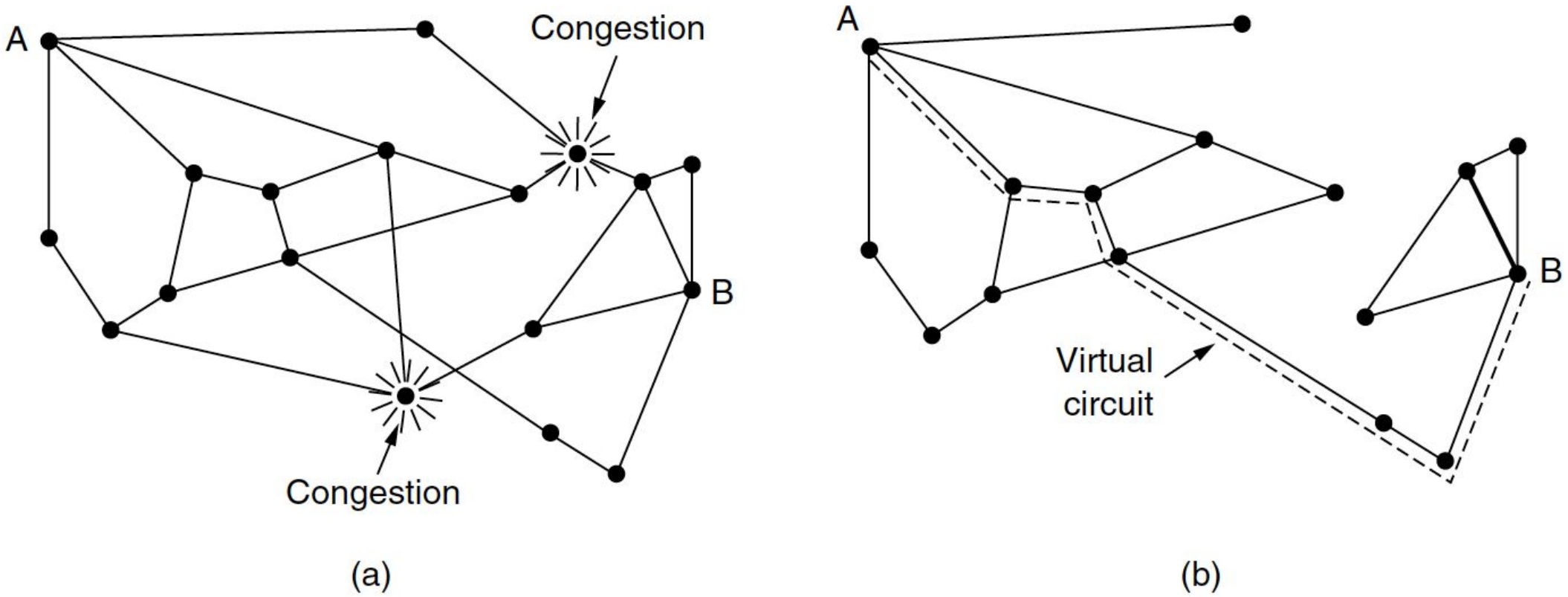


Figure 5-24. (a) A congested network. (b) The portion of the network that is not congested. A virtual circuit from *A* to *B* is also shown.

- *Consider the network illustrated in Fig. 5-24(a), in which two routers are congested, as indicated.*
- *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.*
- *Shaikh et al. (1999) gave a design for this kind of load-sensitive routing.*

Traffic Throttling

- *At a finer granularity, when congestion is imminent the network can deliver feedback to the sources whose traffic flows are responsible for the problem.*
- *The network can request these sources to throttle their traffic, or it can slow down the traffic itself.*
- *Two difficulties with this approach are how to identify the onset of congestion, and how to inform the source that needs to slow down.*
- *To tackle the first issue, routers can monitor the average load, queueing delay, or packet loss. In all cases, rising numbers indicate growing congestion.*
- *To tackle the second issue, routers must participate in a feedback loop with the sources.*

- *For a scheme to work correctly, the time scale must be adjusted carefully.*
- *If every time two packets arrive in a row, a router yells STOP and every time a router is idle for 20 μsec, it yells GO, the system will oscillate wildly and never converge.*
- *On the other hand, if it waits 30 minutes to make sure before saying anything, the congestion-control mechanism will react too sluggishly to be of any use.*
- *Delivering timely feedback is a nontrivial matter. An added concern is having routers send more messages when the network is already congested.*

- *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.*
- *Of these possibilities, the second one is the most useful.*

- *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.*
- *Different schemes use different feedback mechanisms.*

Choke Packets

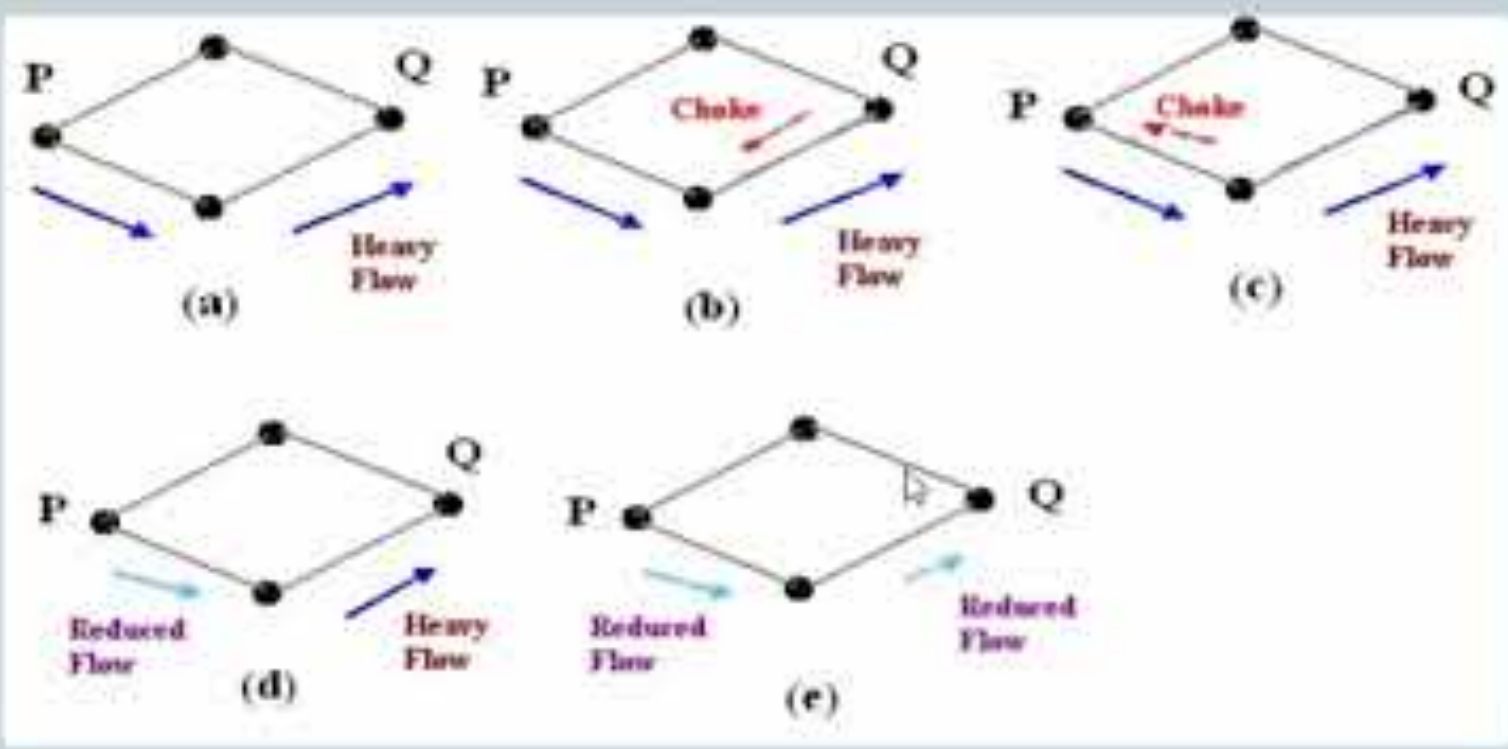
- *The most direct way to notify a sender of congestion is to tell it directly.*
- *In this approach, the router selects a congested packet and sends a choke packet back to the source host, giving it the destination found in the packet.*
- *The original packet may be tagged (a header bit is turned on) 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%.*

Explicit Congestion Notification

- *Instead of generating additional packets to warn of congestion, a router can tag any packet it forwards (by setting a bit in the packet's header) to signal that it is experiencing congestion.*
- *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.*

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 alternative approach is to have the choke packet take effect at every hop it passes through,*
- *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.*
- *The idea is discussed in detail by Mishra et al. (1996).*



Depicts the functioning of choke packets,
(a) Heavy traffic between nodes P and Q,
(b) Node Q sends the Choke packet to P,
(c) Choke packet reaches P,
(d) P reduces the flow and send a reduced flow out,
(e) Reduced flow reaches node Q.

- *Finally, when all else fails, the network is forced to discard packets that it cannot deliver.*
- *The general name for this is **load shedding**.*
- *A good policy for choosing which packets to discard can help to prevent congestion collapse.*

Load Sheading

- *When none of the above methods make the congestion disappear, routers canbring out the heavy artillery:* ***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 term comes from the world of electrical power generation, where it refers to the practice of utilities intentionally blacking out certain areas to save the entire grid from collapsing on hot summer days when the demand for electricity greatly exceeds the supply.*

- *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.*
- *Of course, unless there is some significant incentive to avoid marking every packet as VERY IMPORTANT—NEVER, EVER DISCARD, nobody will do it.*

Random Early Detection

- *Dealing with congestion when it first starts is more effective than letting it gum up the works and then trying to deal with it.*
- *This observation leads to an interesting twist on load shedding, which is to discard packets before all the buffer space is really exhausted.*
- *A popular algorithm for doing this is called RED (Random Early Detection) (Floyd and Jacobson, 1993).*
- *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:

Techniques for Achieving Good Quality of Service:

Overprovisioning:

- An easy solution to provide good quality of service is to build a network with enough capacity(buffer space , Bandwidth) for whatever traffic will be thrown at it. The name for this solution is **overprovisioning.**
- The trouble with this solution is that it is expensive.

Buffering

- Flows can be buffered on the receiving side before being delivered. Buffering them does not affect the reliability or bandwidth, and increases the delay, but it smooths out the jitter.
- For audio and video on demand, jitter is the main problem, so this technique helps a lot.

Traffic Shaping:

- Traffic shaping smooths out the traffic on the server side, rather than on the client side and regulates the average *rate (and burstiness) of data transmission*

Traffic Shaping:

- Traffic Shaping is a mechanism to control the amount and the rate of the traffic sent to the network.
- Approach of congestion management is called Traffic Shaping.
- Traffic shaping helps to regulate average rate and burstiness of of data transmission and reduces congestion.
- Traffic shaping, also known as packet shaping, is a congestion management method that regulates network data transfer by delaying the flow of less important or less desired packets.

- It is used to optimize network performance by prioritizing certain flows and ensuring the traffic rate does’nt exceed the bandwidth limit.
- When a connection is set up, the user and the subnet (i.e the customer and carrier) agree on a certain traffic pattern called shape for that circuit. Sometimes this is called a service level agreement.
- Such agreements are not so important for file transfers but are of great importance for real-time data, such as audio and video connections, which have stringent quality of-service requirements.
- Monitoring a traffic flow is called traffic policing.

Traffic Shaping Algorithms:

- Leaky Bucket Algorithm
- Token Bucket Algorithm

Leaky Bucket Algorithm

- It was proposed by Turner in 1986
- It is single-server queuing system with constant service time.
- Water coming out is in some fixed rate and if bucket will be full, we will stop pouring it it.
- The input rate can vary, but the output rate remains constant.
- Similarly, in networking, a host is allowed to put one packet per clock tick onto network.
- This mechanism turns an uneven flow of packets from host into an even flow of packets onto network.
- It smoothens the bursty traffic and reduces the chances of congestion.

Leaky Bucket Algorithm

Let us consider an example to understand

- Imagine a bucket with a small hole in the bottom. No matter at what rate water enters the bucket, the outflow is at constant rate. When the bucket is full of water additional water entering spills over the sides and is lost.
- Similarly, each network interface contains a leaky bucket, and the following **steps** are involved in leaky bucket algorithm:

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

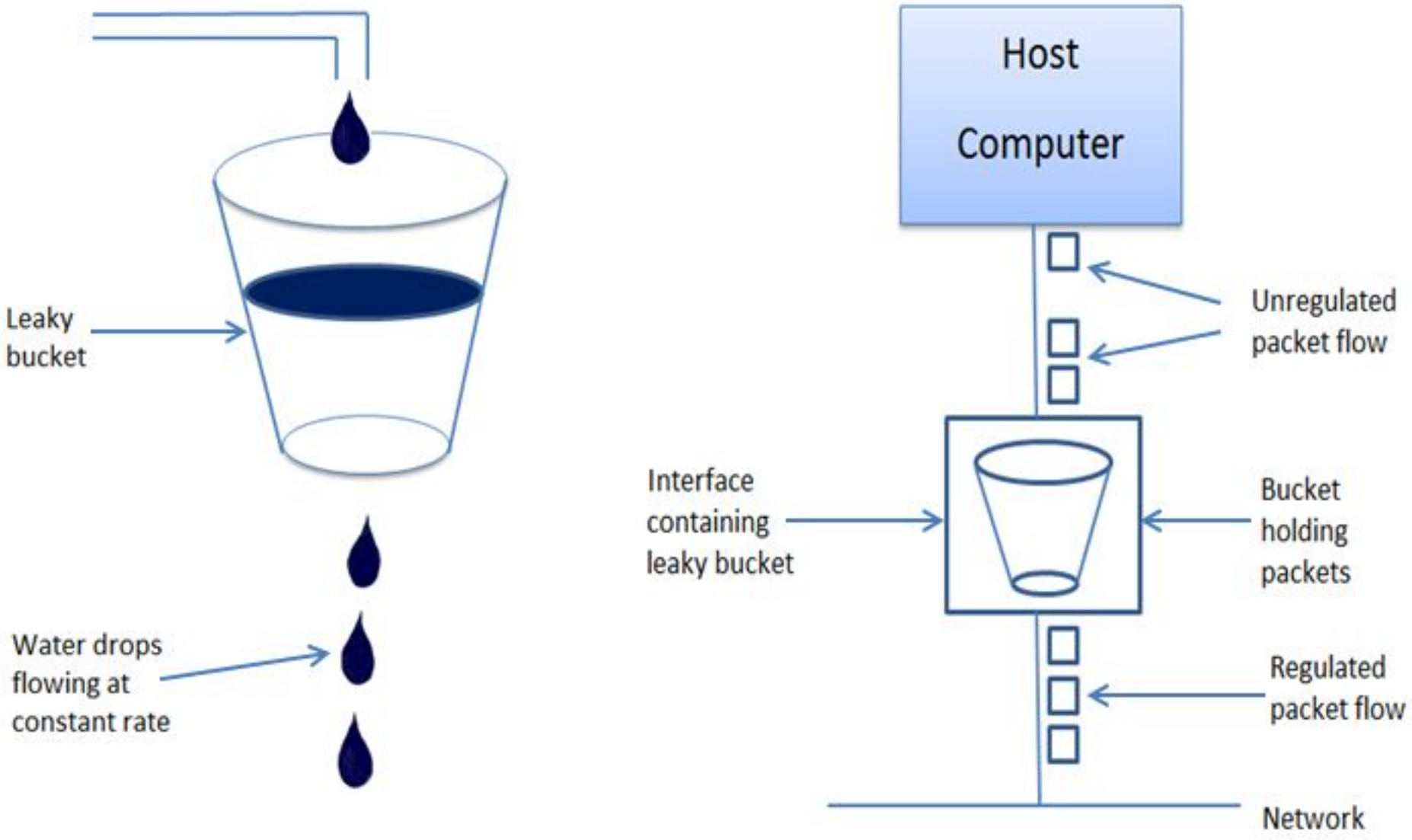
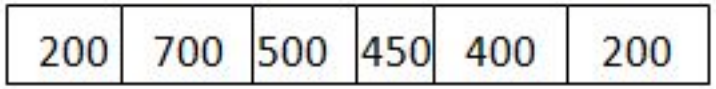


Fig: Leaky Bucket Algorithm

Algorithm

- Initialize a counter to n at a tick of the clock.
- If n is greater than the size of the packet, send the packet and decrement the counter by packet size else reject the packet.
- Repeat this until all packets have been processed.

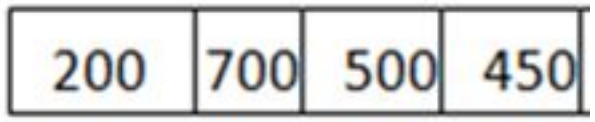
Example

- Let n=1000
- Packet= 

- Since n > size of the packet at the head of the Queue, i.e., n > 200
  Therefore, n = 1000-200 = 800
  Packet size of 200 is sent into the network.

- Now, agai 

at the head of the Queue, i.e., n > 400
  Therefore, n = 800-400 = 400
  Packet size of 400 is sent into the network.

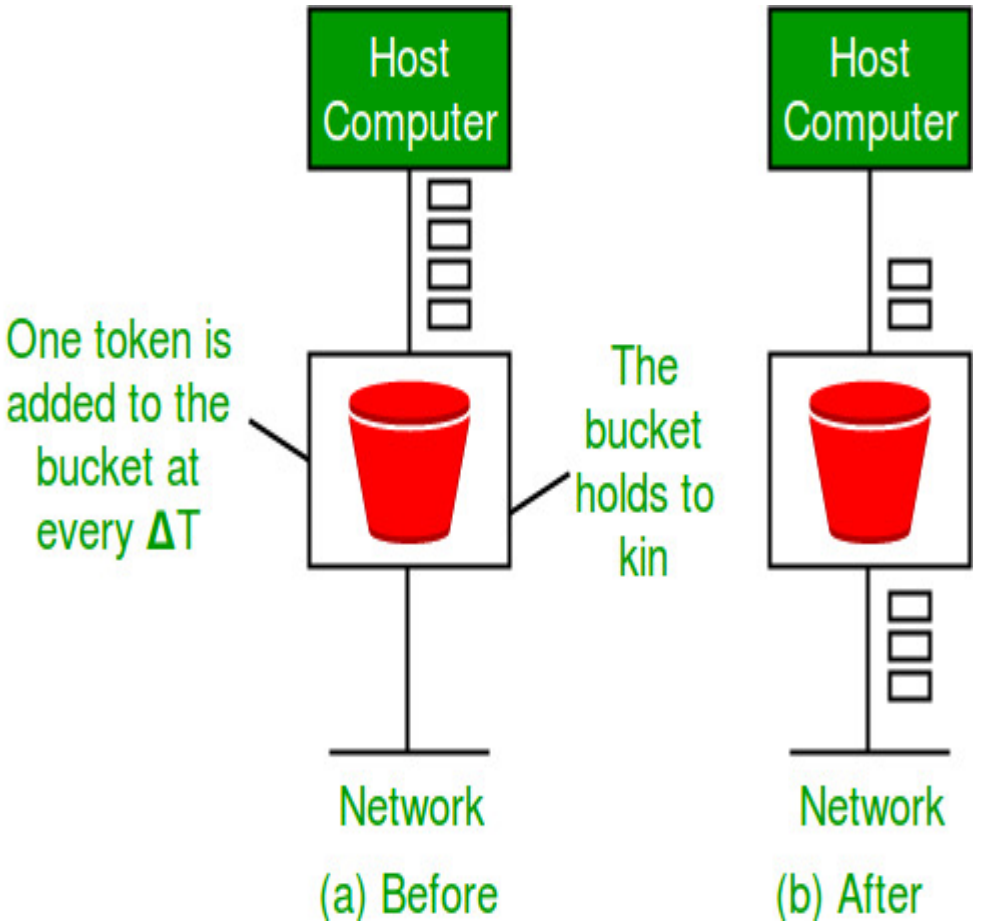


- Since, n < size of the packet at the head of the Queue, i.e. n < 450
  Therefore, the procedure is stopped.

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.
- **Steps** of this algorithm can be described as follows:
- In regular intervals tokens are thrown into the bucket. *f*
- The bucket has a maximum capacity. *f*
- 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.

- Let's understand with an example,
- In figure (A) we see a bucket holding three tokens, with five packets waiting to be transmitted. For a packet to be transmitted, it must capture and destroy one token. In figure (B) We see that three of the five packets have gotten through, but the other two are stuck waiting for more tokens to be generated.


Host Computer
One token is added to the bucket at every ΔT
The bucket holds to kin
Network
(a) Before
Host Computer
Network
(b) After

Ways in which token bucket is superior to leaky bucket:

The leaky bucket algorithm controls the rate at which the packets are introduced in the network, but it is very conservative in nature.

- Some flexibility is introduced in the token bucket algorithm.
- In the token bucket, algorithm tokens are generated at each tick (up to a certain limit).
- For an incoming packet to be transmitted, it must capture a token and the transmission takes place at the same rate.
- Hence some of the busty packets are transmitted at the same rate if tokens are available and thus introduces some amount of flexibility in the system.

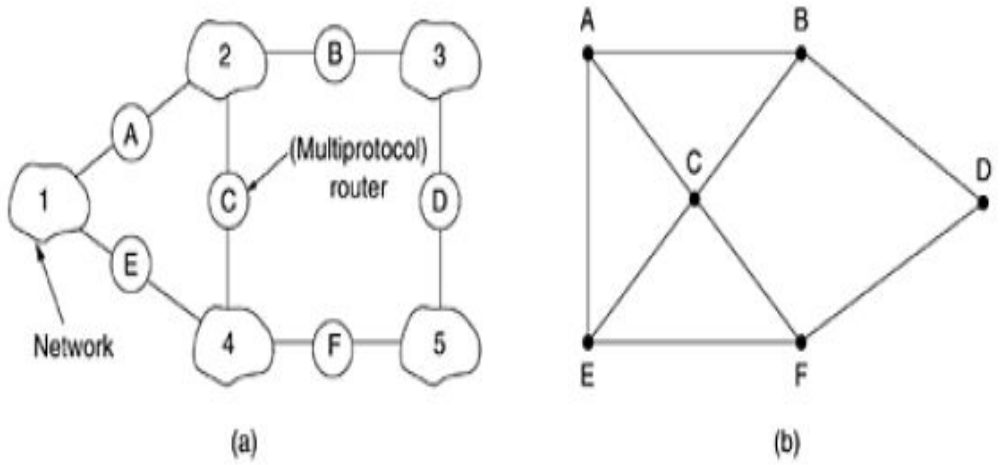
Internetworking

- Network layer in the internet
- **IP protocols:**
  - IP Version 4
  - IP Version 6
  - Transition from IPV4 to IPV6
  - Comparison of IPV4 & IPV6
- **Internet control protocols**:
  - ICMP
  - ARP
  - DHCP

Internetworking

- Routing through an internetwork is similar to routing within a single subnet, but with some added complications.
- For example, *B in Fig. 5*-49(a) can directly access *A and C via network 2 and also D via network 3. This leads to the* graph of Fig. 5-49(b).

Figure 5-49. (a) An internetwork. (b) A graph of the internetwork.



- Once the graph has been constructed, known routing algorithms, such as the distance vector and link state algorithms, can be applied to the set of multiprotocol routers.
- This gives a two level routing algorithm: within each network an **interior gateway protocol is used, but** between the networks, an **exterior gateway protocol is used.** In fact, since each network is independent, they may all use different algorithms.
- Because each network in an internetwork is independent of all the others, it is often referred to as an **Autonomous System (AS).**

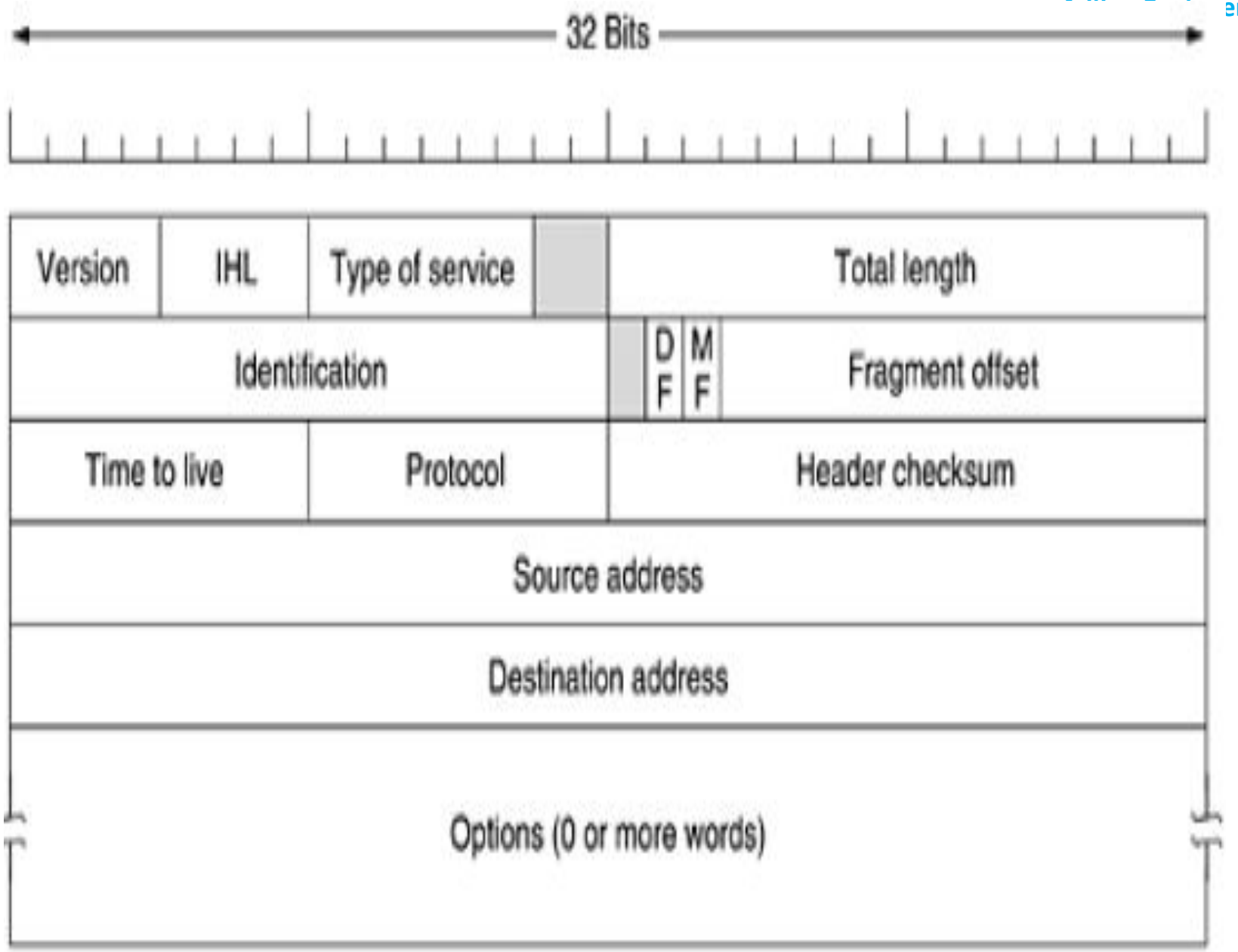
The Network Layer in the Internet:

Internet Protocol:

- IPv4 is a connectionless protocol used for packet-switched networks.
- Internet Protocol Version 4 (IPv4) is the fourth revision of the Internet Protocol and a widely used protocol in data communication over different kinds of networks.
- It provides a logical connection between network devices by providing identification for each device.
- IPv4 uses 32-bit (4 byte) addressing, which gives 2^{32} addresses.
- IPv4 addresses are written in the dot-decimal notation, which comprises of four octets of the address expressed individually in decimal and separated by periods, for instance, 192.168.1.5

Class	Address range	Supports
Class A	1.0.0.1 to 126.255.255.254	Supports 16 million hosts on each of 127 networks.
Class B	128.1.0.1 to 191.255.255.254	Supports 65,000 hosts on each of 16,000 networks.
Class C	192.0.1.1 to 223.255.254.254	Supports 254 hosts on each of 2 million networks.
Class D	224.0.0.0 to 239.255.255.255	Reserved for multicast groups.
Class E	240.0.0.0 to 254.255.255.254	Reserved for future use, or research and development purposes.

Ranges 127.x.x.x are reserved for the loopback or localhost, for example, **127.0.0.1** is the loopback address. Range **255.255.255.255** broadcasts to all hosts on the local network.


32 Bits
Version
IHL
Type of service
Total length
Identification
D F
M F
Fragment offset
Time to live
Protocol
Header checksum
Source address
Destination address
Options (0 or more words)

IPv4 Frame Format

- **VERSION:** Version of the IP protocol (4 bits), which is 4 for IPv4
- **IHL:** IP header length (4 bits), which is the number of 32 bit words in the header. The minimum value for this field is 5 and the maximum is 15.
- **Type of service:** Low Delay, High Throughput, Reliability (8 bits)
- **Total Length:** Length of header + Data (16 bits), which has a minimum value 20 bytes and the maximum is 65,535 bytes.
- **Identification:** Unique Packet Id for identifying the group of fragments of a single IP datagram (16 bits)
- **Flags:** 3 flags of 1 bit each : reserved bit (must be zero), do not fragment flag, more fragments flag (same order)

- **Fragment Offset:** Represents the number of Data Bytes ahead of the particular fragment in the particular Datagram. Specified in terms of number of 8 bytes, which has the maximum value of 65,528 bytes.
- **Time to live:** Datagram's lifetime (8 bits), It prevents the datagram to loop through the network by restricting the number of Hops taken by a Packet before delivering to the Destination.
- **Protocol:** Name of the protocol to which the data is to be passed (8 bits)
- **Header Checksum:** 16 bits header checksum for checking errors in the datagram header.

- *The **Source address** and **Destination address** indicate the IP address of the source and destination network interfaces.*
- *The **Options** field was designed to provide an escape to allow subsequent versions of the protocol to include information not present in the original design, to permit experimenters to try out new ideas, and to avoid allocating header bits to information that is rarely needed.*
- *The options are of variable length. Each begins with a 1-byte code identifying the option.*
- *Some options are followed by a 1-byte option length field, and then one or more data bytes. The Options field is padded out to a multiple of 4 bytes.*

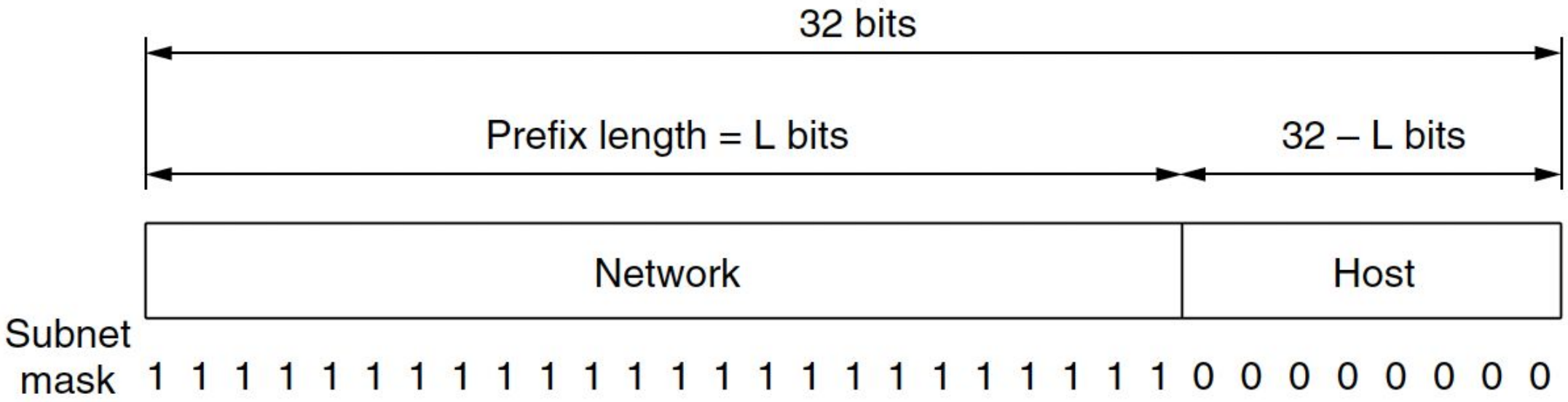
Option Name	Type	Description
End of option	0	This is used to tell “end of options”
No operation	1	This is used to align subsequent option in 32 bit boundary
Loose Source and Record Route	131	It lets the originating system to set the intermediate devices/routers the packet should take to reach its destination. Also the route taken is recorded in the option field. The destination host must use this reverse path noted in this option
Strict Source and Record Route	137	It lets the originating system to set the number of devices/routers the packet should take to reach its destination. Also the route taken is recorded in the option field. The destination host must use this reverse path noted in this option
Record Route	7	This is used to record route a packet takes to reach its destination
Stream Identifier	136	
Internet Timestamp	68	This option records the time that reach system takes to process the datagram
Router Alert	148	Packets containing this option should be look more closely by the router
Probe MTU	11	This is issued to provide Path MTU however it has been declared obsolete
Reply MTU	12	Same is above however this is now obsolete
Traceroute	82	This is used to trace the path to the destination host

IP Address

- *A defining feature of IPv4 is its 32-bit addresses. Every host and router on the Internet has an IP address that can be used in the Source address and Destination address fields of IP packets*
- *It is important to note that an IP address does not actually refer to a host.*
- *It really refers to a network interface, so if a host is on two networks, it must have two IP addresses*

Prefixes

- *IP addresses are hierarchical, unlike Ethernet addresses.*
- *Each 32-bit address is comprised of a variable-length network portion in the top bits and a host portion in the bottom bits.*
- *The network portion has the same value for all hosts on a single network, such as an Ethernet LAN.*
- *This means that a network corresponds to a contiguous block of IP address space.*
- *This block is called a **prefix**.*


32 bits
Prefix length = L bits
32 – L bits
Network
Host
Subnet mask
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0

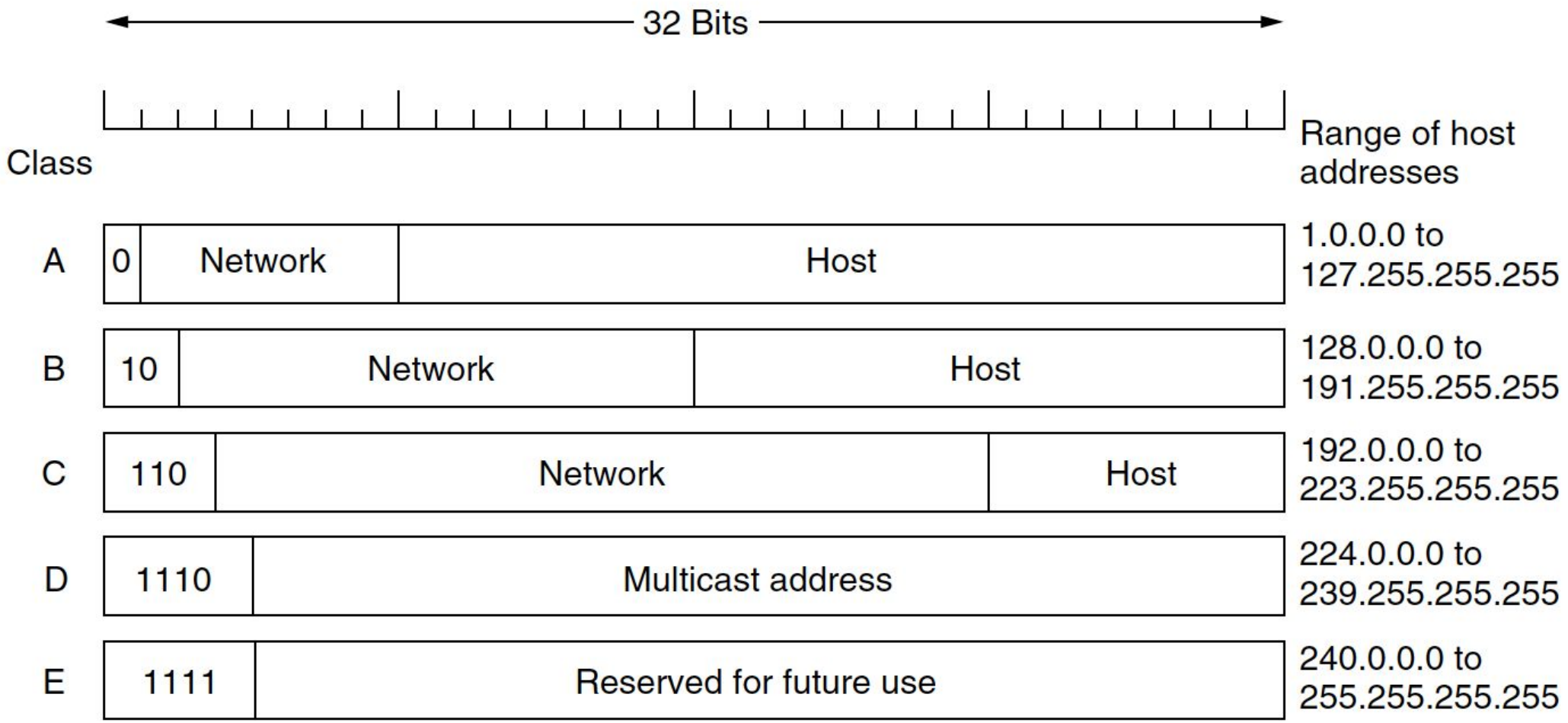
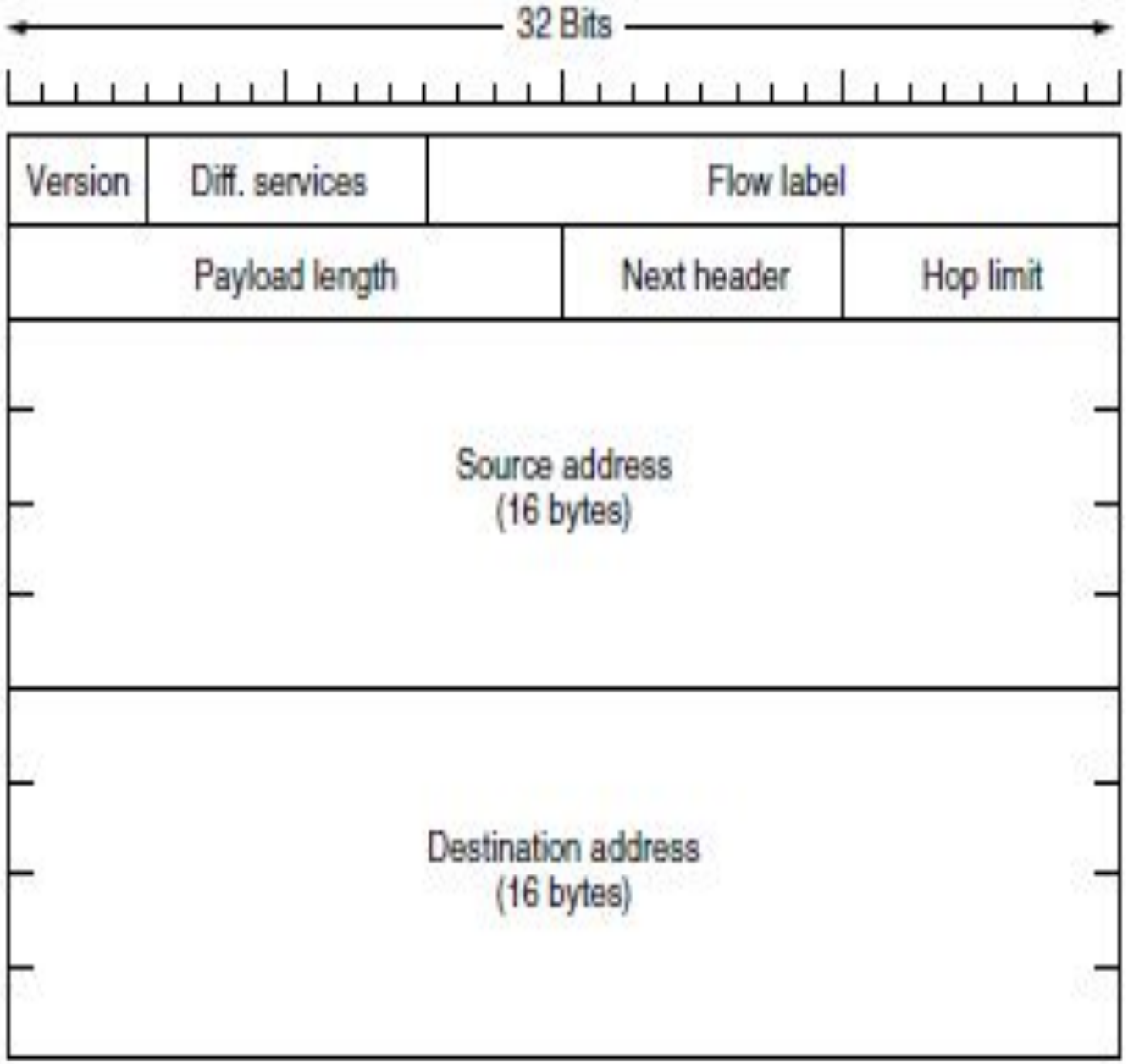


Figure 5-53. IP address formats.

IP Version 6:

- It uses 128-bit addresses.
- Goals of IPv6 are: Support billions of hosts, Reduce the size of the routing tables, Provide better security (authentication and privacy), Pay more attention to the type of service, particularly for real-time data, Allow the protocol to evolve in the future, Permit the old and new protocols to coexist for years.


32 Bits
Version
Diff. services
Flow label
Payload length
Next header
Hop limit
Source address
(16 bytes)
Destination address
(16 bytes)

IPv6 Header Format:

- **Version (4-bits) :** Indicates version of Internet Protocol which contains bit sequence 0110.
- **Differentiated services field (originally called Traffic class):** *used to* distinguish the class of service for packets with different real-time delivery.
- **Flow Label (20-bits) :** Flow Label field is used by source to label the packets belonging to the same flow in order to request special handling by intermediate IPv6 routers.
- **Payload Length (16-bits) :** It is a 16-bit (unsigned integer) field, indicates total size of the payload which tells routers about amount of information a particular packet contains.

- **Next Header (8-bits) :** Next Header indicates type of extension header(if present) immediately following the IPv6 header.
- **Hop Limit (8-bits) :** Hop Limit field is same as TTL in IPv4 packets. It indicates the maximum number of intermediate nodes IPv6 packet is allowed to travel.
- **Source Address (128-bits) :** Source Address is 128-bit IPv6 address of the original source of the packet.
- **Destination Address (128-bits) :** Destination Address field indicates the IPv6 address of the final destination(in most cases).

Transition from IPv4 to IPv6

- *A new notation has been devised for writing 16-byte addresses.*
- *They are written as eight groups of four hexadecimal digits with colons between the groups, like this:*
- *8000:0000:0000:0000:0123:4567:89AB:CDEF*
- *Since many addresses will have many zeros inside them, three optimizations have been authorized.*
- *First, leading zeros within a group can be omitted, so 0123 can be written as 123.*
- *Second, one or more groups of 16 zero bits can be replaced by a pair of colons.*
- *Thus, the above address now becomes*
- *8000::123:4567:89AB:CDEF*
- *Finally, IPv4 addresses can be written as a pair of colons and an old dotted decimal number, for example:*
- *::192.31.20.46*

Extension Headers

- *Some of the missing IPv4 fields are occasionally still needed, so IPv6 introduces the concept of (optional) extension headers. These headers can be supplied to provide extra information, but encoded in an efficient way.*
- *Six kinds of extension headers are defined at present.*
- *Each one is optional, but if more than one is present they must appear directly after the fixed header, and preferably in the order listed.*

Extension header	Description
Hop-by-hop options	Miscellaneous information for routers
Destination options	Additional information for the destination
Routing	Loose list of routers to visit
Fragmentation	Management of datagram fragments
Authentication	Verification of the sender's identity
Encrypted security payload	Information about the encrypted contents

Figure 5-57. IPv6 extension headers.

- *Some of the headers have a fixed format; others contain a variable number of variable-length options.*
- *For these, each item is encoded as a (Type, Length, Value) tuple.*
- *The Type is a 1-byte field telling which option this is. The Type values have been chosen so that the first 2 bits tell routers that do not know how to process the option what to do.*
- *The choices are:*
- *skip the option;*
- *discard the packet;*
- *discard the packet and send back an ICMP packet;*
- *and discard the packet but do not send ICMP packets for multicast addresses (to prevent one bad multicast packet from generating millions of ICMP reports).*

- *The Length is also a 1-byte field. It tells how long the value is (0 to 255 bytes). The Value is any information required, up to 255 bytes.*
- *The hop-by-hop header is used for information that all routers along the path must examine.*
- *So far, one option has been defined: support of datagrams exceeding 64 KB.*
- *The format of this header is shown in Fig. 5-58. When it is used, the Payload length field in the fixed header is set to 0.*

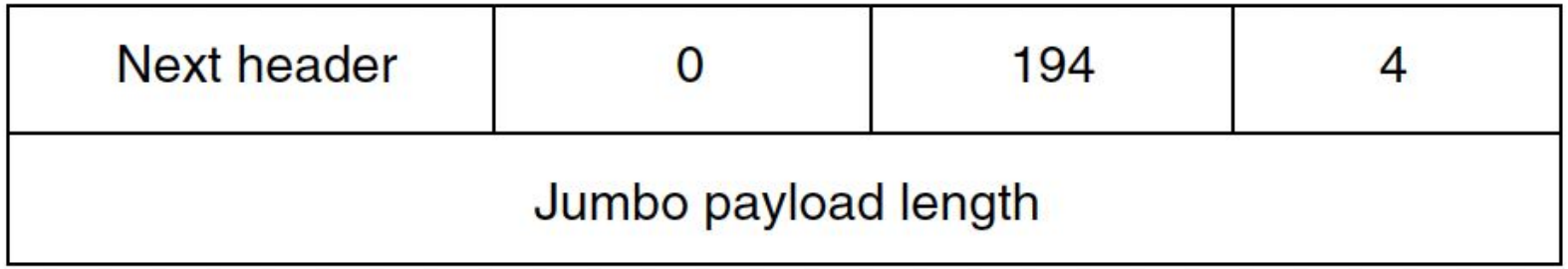


Figure 5-58. The hop-by-hop extension header for large datagrams (jumbograms).

Internet Control Protocols

- ICMP
- ARP
- DHCP

Internet Control Protocols

- *In addition to IP, which is used for data transfer, the Internet has several companion control protocols that are used in the network layer.*
- *They include :*
- *ICMP*
- *ARP*
- *DHCP*

ICMP (Internet Control Message Protocol)

- To handle error and control messages, IP depends on Internet Control Message Protocol(ICMP).
- It is a supporting protocol and used by networks devices like routers for sending the error messages and operations information.
- **Destination un-reachable :** This message is generated by the host to inform the client that the destination is unreachable for some reason.
- **Time exceeded message :** When some fragments are lost in a network then the holding fragment by the router will be dropped then ICMP will take source IP from discarded packet and informs to the source.
- **Parameter problem :** Whenever packets come to the router then calculated header checksum should be equal to received header checksum then only packet is accepted by the router.

- **Source quench**: It is a message request to decrease traffic rate for messages sending to the host(destination).
- **Redirection message :** Redirect requests data packets be sent on an alternate route.
- The **ECHO** and **ECHO REPLY** messages are sent by hosts to see if a given destination is reachable and currently alive. Upon receiving the ECHO message the destination is expected to send back an ECHO REPLY message. These messages are used in the **ping** utility that checks if a host is up and on the Internet.
- The **TIMESTAMP REQUEST** and **TIMESTAMP REPLY** messages are similar, except that the arrival time of the message and the departure time of the reply are recorded in the reply. This facility can be used to measure network performance.
- The **ROUTER ADVERTISEMENT** and **ROUTER SOLICITATION** messages are used to let hosts find nearby routers. A host needs to learn the IP address of at least one router to be able to send packets off the local network.

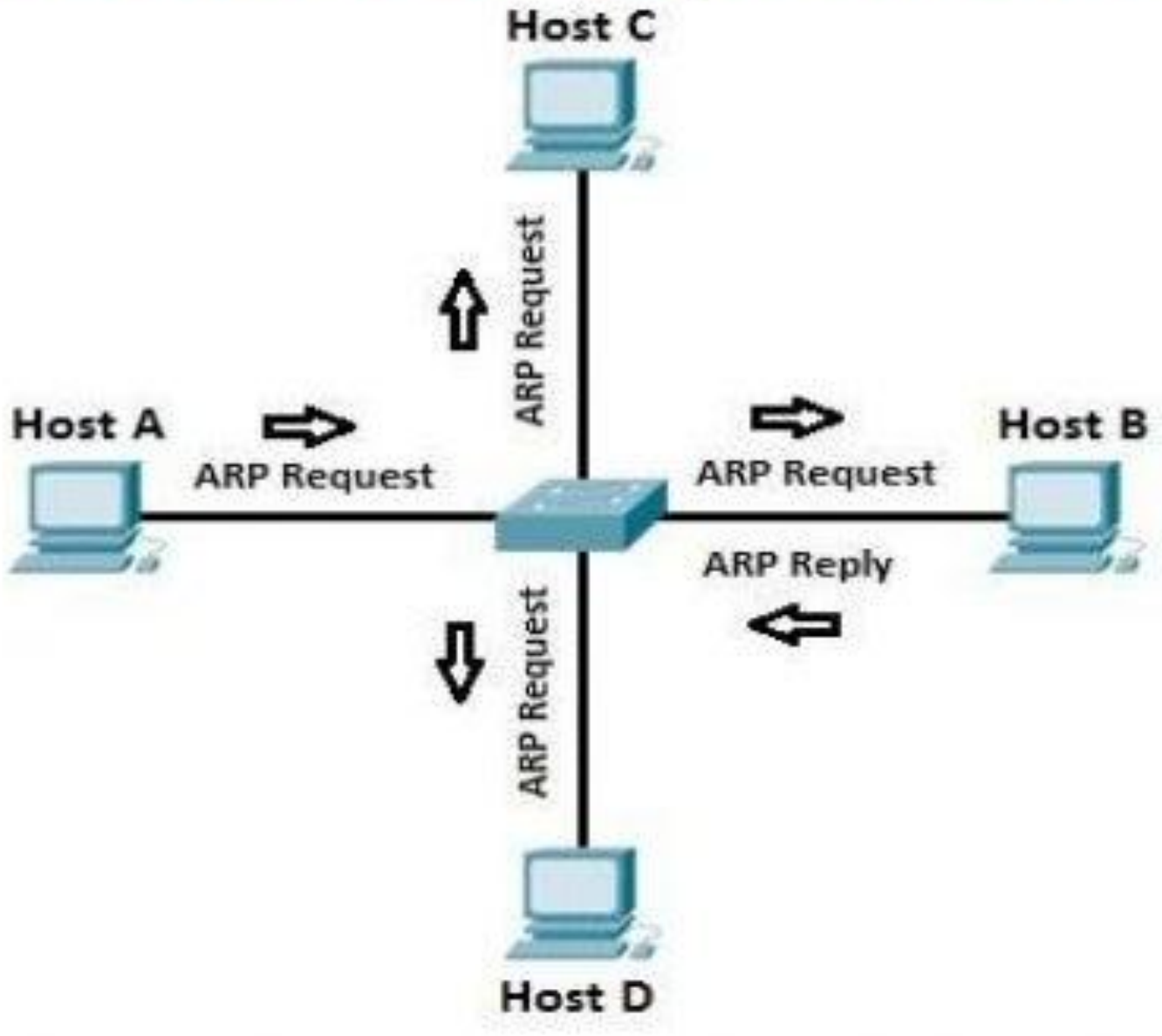
ICMP Message Types:

Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo and echo reply	Check if a machine is alive
Timestamp request/reply	Same as Echo, but with timestamp
Router advertisement/solicitation	Find a nearby router

ARP—The Address Resolution Protocol

- ARP is a network protocol used to find out the hardware (MAC) address of a device from an IP address.
- It is used when a device wants to communicate with some other device on a local network.
- The sending device uses ARP to translate IP addresses to MAC addresses.
- The device sends an ARP request message containing the IP address of the receiving device.
- All devices on a local network segment see the message, but only the device that has that IP address responds with the ARP reply message containing its MAC address.

- The sending device now has enough information to send the packet to the receiving device.
- ARP request packets are sent to the broadcast addresses (FF:FF:FF:FF:FF:FF for the ethernet broadcasts and 255.255.255.255 for the IP broadcast).


Host C
ARP Request
Host A
ARP Request
ARP Request
Host B
ARP Reply
ARP Request
Host D

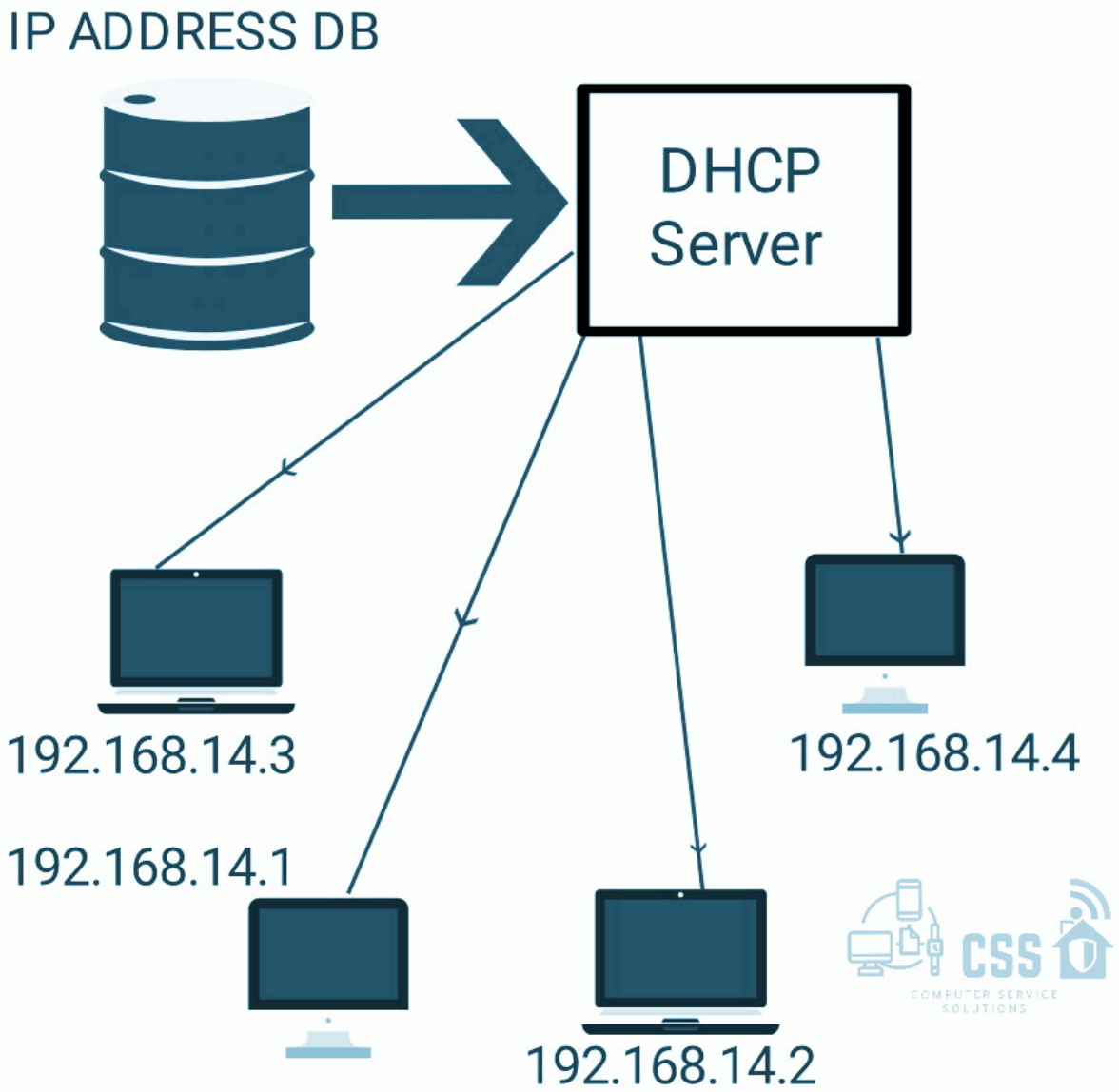
- Lets say that Host A wants to communicate with host B.
- Host A knows the IP address of host B, but it doesn't know the host B's MAC address.
- In order to find out the MAC address of host B, host A sends an ARP request, listing the host B's IP address as the destination IP address and the MAC address of FF:FF:FF:FF:FF:FF
- Switch will forward the frame out to all interfaces except the incoming interface.
- Each device on the segment will receive the packet, but because the destination IP address is host B's IP address, only host B will reply with the ARP reply packet, listing its MAC address.
- Host A now has enough information to send the traffic to host B.
- All operating systems maintain ARP caches that are checked before sending an ARP request message.
- One can display ARP entries in windows by using the arp – a command.

DHCP

- DHCP stands for Dynamic Host Configuration Protocol.
- It is basically used to configure the host dynamically.
- It temporarily binds IP address & other configuration parameters to DHCP Client and provides framework for passing configuration information to hosts.
- It was designed basically to provide the computers with temporary address.
- It is well adapted to situations where hosts move from one location to another or are regularly connecting and disconnecting.

Characteristics of DHCP

- Centralized IP address administration.
- Supports multiple servers.
- Provides dynamic assignment of IP Address.
- Also allows static assignment.
- Doesn’t interact with DNS (Domain Name service)


IP ADDRESS DB
DHCP Server
192.168.14.3
192.168.14.4
192.168.14.1
192.168.14.2

BOOTP/DHCP MESSAGE FORMAT

OpCode (1 = Req, 2 = Reply)	Hardware Type (1 = Ethernet)	Hardware Address Length	Hop Count
Number of Seconds		Unused (in BOOTP) Flags (in DHCP)	
Transaction ID			
Client IP address			
Your IP address			
Server IP address			
Gateway IP address			
Client hardware address (16 bytes)			
Server host name (64 bytes)			
Boot file name (128 bytes)			
Options			

(There are >100 different options)

13