NETWORK LAYER

MODULE3

CHAPTER 5
THE NETWORK LAYER

I. Routing Algorithm
II. Network Layer Design Issues
III. Congestion Control Algorithms
IV. QOS Control Algorithms

Network layer design Issues:

The Network layer is concerned with getting packets from the source all the way to destination.

Network layer design Issues:

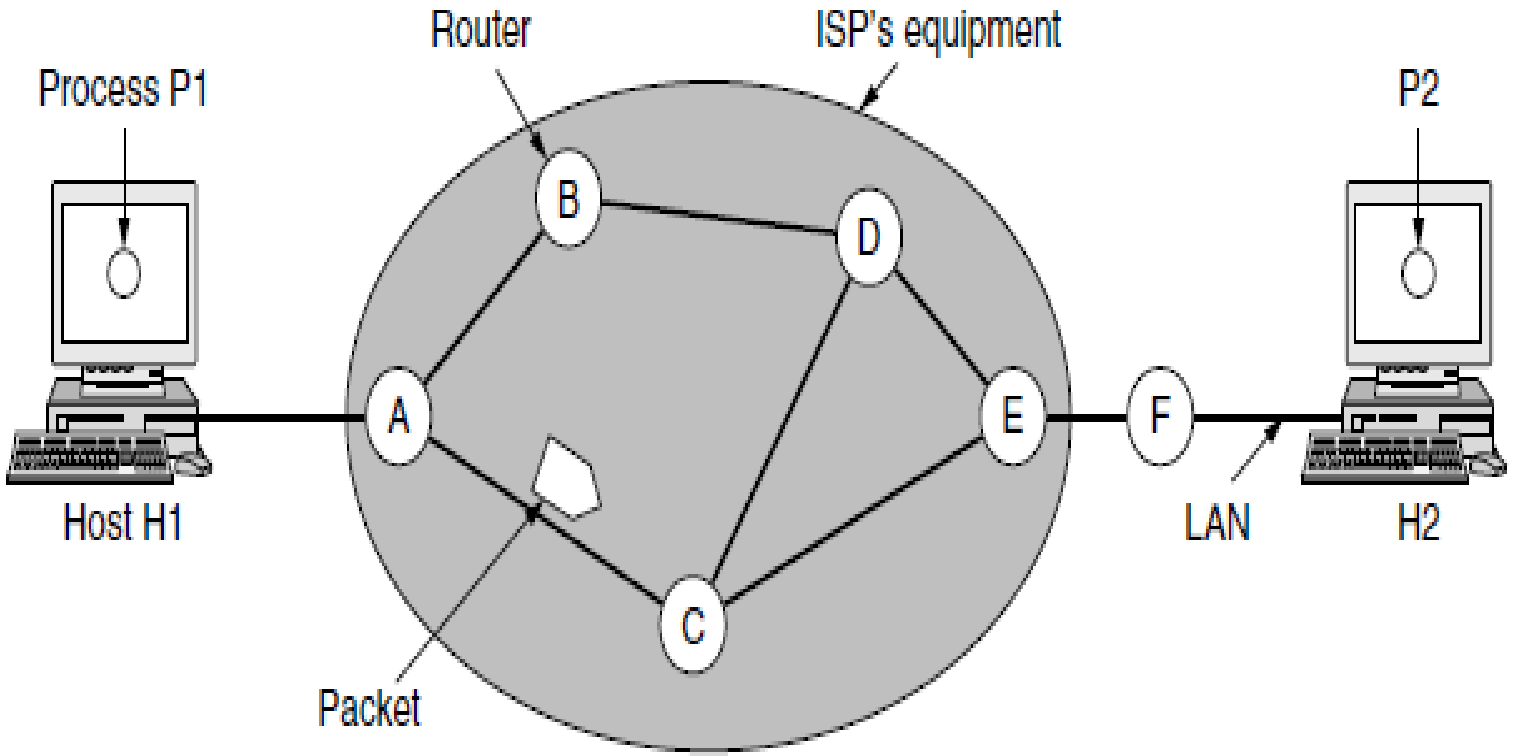
1.Service provided to the transport layer

2. the internal design of the network

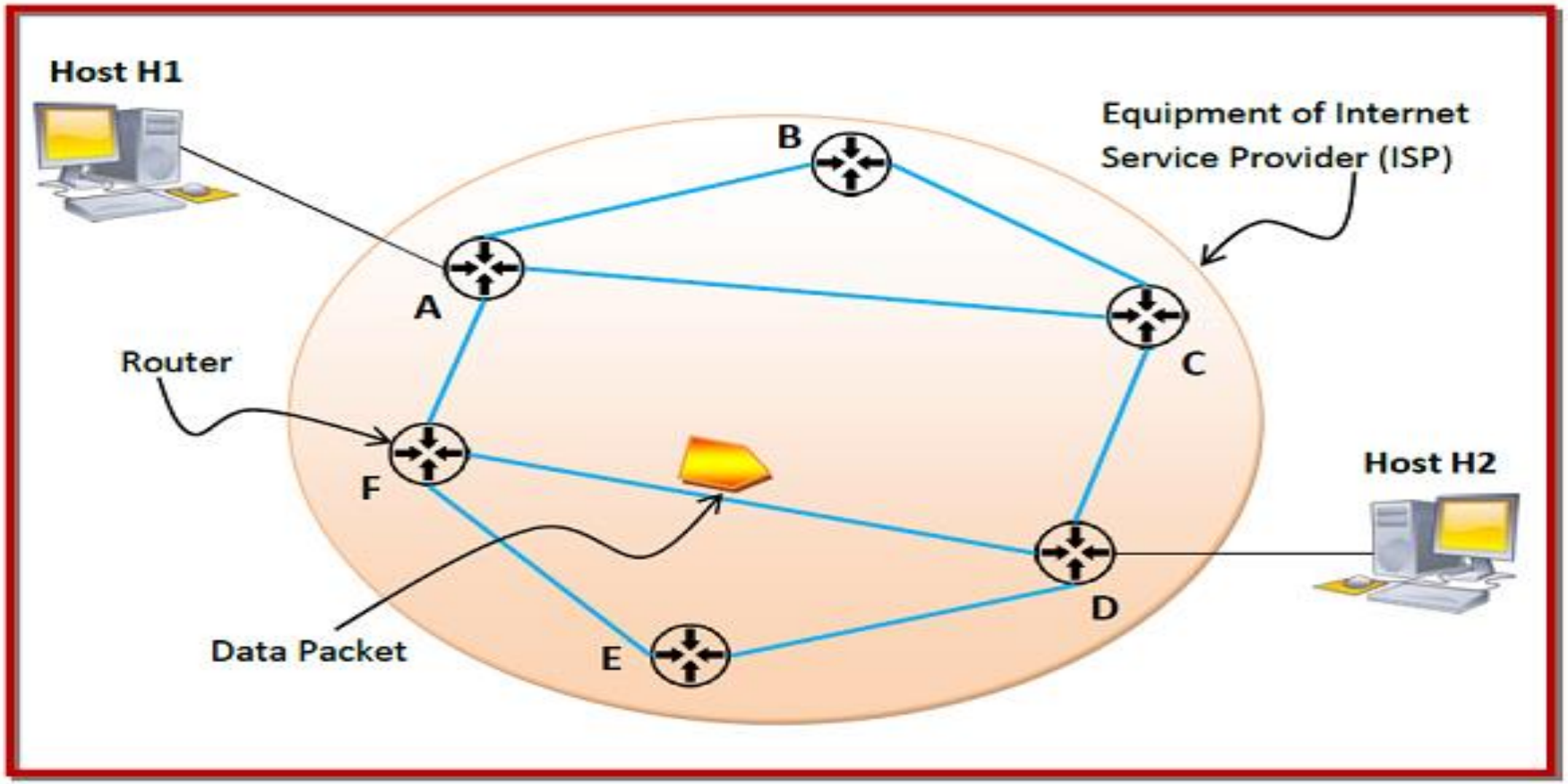
3.Implementation of Connectionless service

4.Implementation of Connection oriented service

The environment of the network layer protocols



Store-and-Forward Packet Switching



Store-and-Forward Packet Switching

- In the above diagram, we can see that the Internet Service Provider (ISP) has six routers (A to F) connected by transmission lines shown in blue lines.
- There are two hosts, host H1 is connected to router A, while host H2 is connected to router D.
- Suppose that H1 wants to send a data packet to H2. H1 sends the packet to router A.
- The packet is stored in router A until it has arrived fully. Router A verifies the checksum using CRC (cyclic redundancy check) code. If **there is a CRC error**, the packet is discarded, otherwise it is transmitted to the next hop, here router F.
- The same process is followed by router F which then transmits the packet to router D. Finally router D delivers the packet to host H2.

Store-and-Forward Packet Switching

- The node which has a packet to send, delivers it to the nearest node, i.e. router. The packet is stored in the router until it has fully arrived and its checksum is verified for error detection.
- Once, this is done, the packet is transmitted to the next router.
- The same process is continued in each router until the packet reaches its destination.

Services provided to the transport layer

- **Logical Addressing** – Network layer adds header to incoming packet which includes logical address to identify sender and receiver.
- **Routing** – It is the mechanism provided by Network Layer for routing the packets to the final destination in the fastest possible and efficient way.
- **Flow control** – This layer routes the packet to another way, If too many packets are present at the same time preventing bottlenecks and congestion.

Services provided to the transport layer

- **Breaks Large Packets** – Breaks larger packets into small packets.
- **Connection Oriented service** – It is a network communication mode, where a communication session is established before any useful data can be transferred and where a stream of data is delivered in the same order as it was sent.
- **Connectionless Service** – It is a data transmission method used in packet switching networks by which each data unit is individually addressed and routed based on information carried in each unit, rather than in the setup information of a prearranged, fixed data channel as in connection-oriented communication.

Services provided to the transport layer

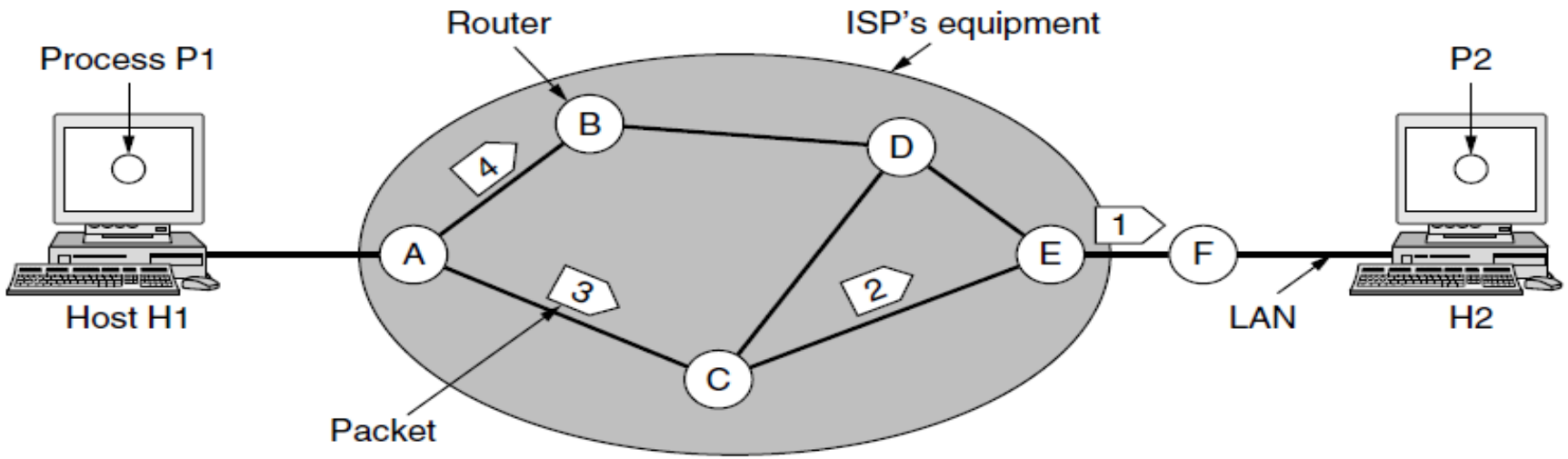
- **Datagram** – A datagram is a basic transfer unit associated with a packet-switched network. The delivery, arrival time and order of arrival need not be guaranteed by the network.

- **A virtual circuit** – It is a means of transporting data over a packet switched computer network in such a way that it appears as though there is a dedicated physical layer link between the source and destination end system of this data.

Implementation of Connectionless Service(Datagram Switching)

- If connectionless service is offered, packets are injected into the network individually and routed independently of each other.
- No advance setup is needed. In this context, the packets are frequently **called datagrams** (in analogy with telegrams) and the network is called a datagram network.
- **IP (Internet Protocol),** which is the basis for the entire Internet, is the dominant example of a connectionless network service. Each packet carries a destination IP address that routers use to individually forward each packet.

Implementation of Connection oriented Service



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

Implementation of Connection oriented Service

- 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 a VC (virtual circuit), in analogy with the physical circuits set up by the telephone system, and the network is called a virtual-circuit network.
- An example of aconnection-oriented network service is MPLS (MultiProtocol Label Switching).It is used within ISP networks in the Internet, with IP packets wrapped in an MPLS header having a 20-bit connection identifier or label.

Implementation of Connection-Oriented Service(virtual circuit switching , label switching)

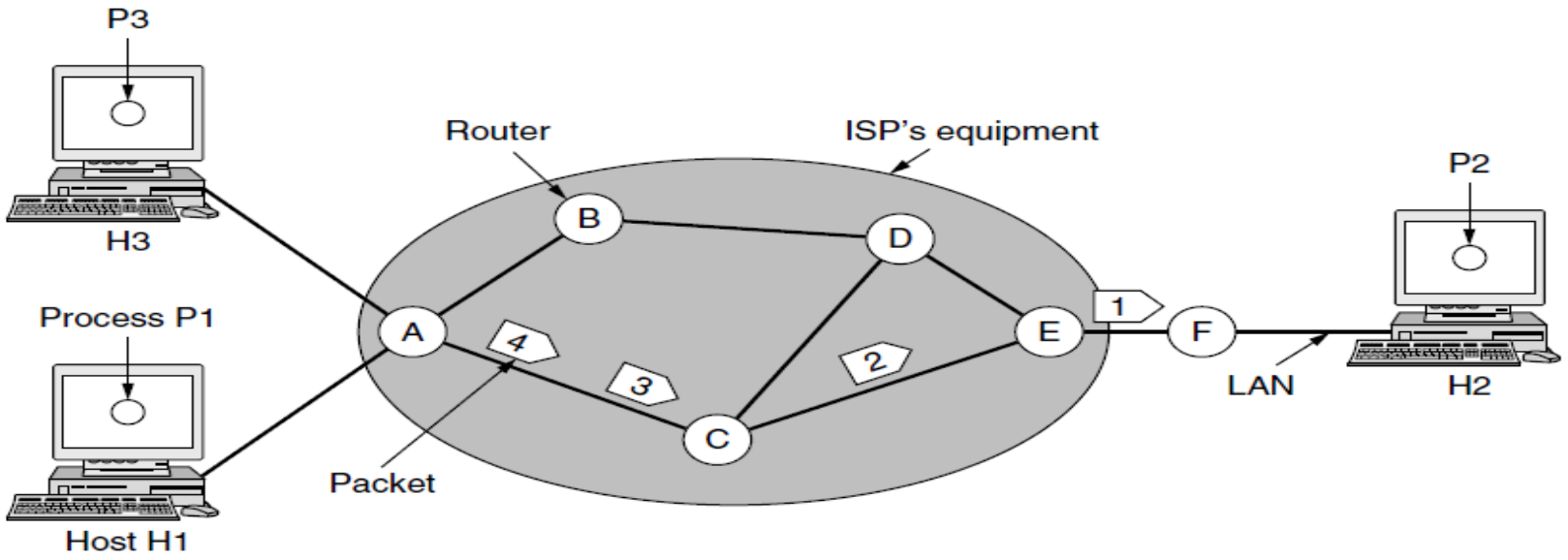


Figure 5-3. Routing within a virtual-circuit network.

Comparison of Virtual-Circuit and Datagram Networks

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

Figure 5-4. Comparison of datagram and virtual-circuit networks.

ROUTING ALGORITHMS

Characteristics of Routing algorithms:

- Correctness, simplicity, robustness, stability, fairness and efficiency.
- The routing algorithm should be able to cope with changes in the topology and traffic without requiring all jobs in all hosts to be aborted.
- Stability is also an important goal for the routing algorithm.
- Minimizing the mean packet delay is an obvious candidate to send traffic through the network effectively, but so is maximizing total network throughput.

Types of Routing :

- Routing algorithms can be grouped into two major classes:
  1.Nonadaptive

  2.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 to use to get from *I* to *J* (for all *I* and *J*) is computed in advance, offline, and downloaded to the routers when the network is booted. This procedure is sometimes called **static routing.**
- Because it does not respond to failures, static routing is mostly useful for situations in which the routing choice is clear

Types of Routing :

- Adaptive algorithms, in contrast, change their routing decisions to reflect changes in the topology, and sometimes changes in the traffic as well.
- These dynamic routing 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), and what metric is used for optimization (e.g., distance, number of hops, or estimated transit time)

The Optimality Principle-

about optimal routes without regard to network topology or traffic

- This statement is known as the **optimality principle** (Bellman,1957)
- 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.

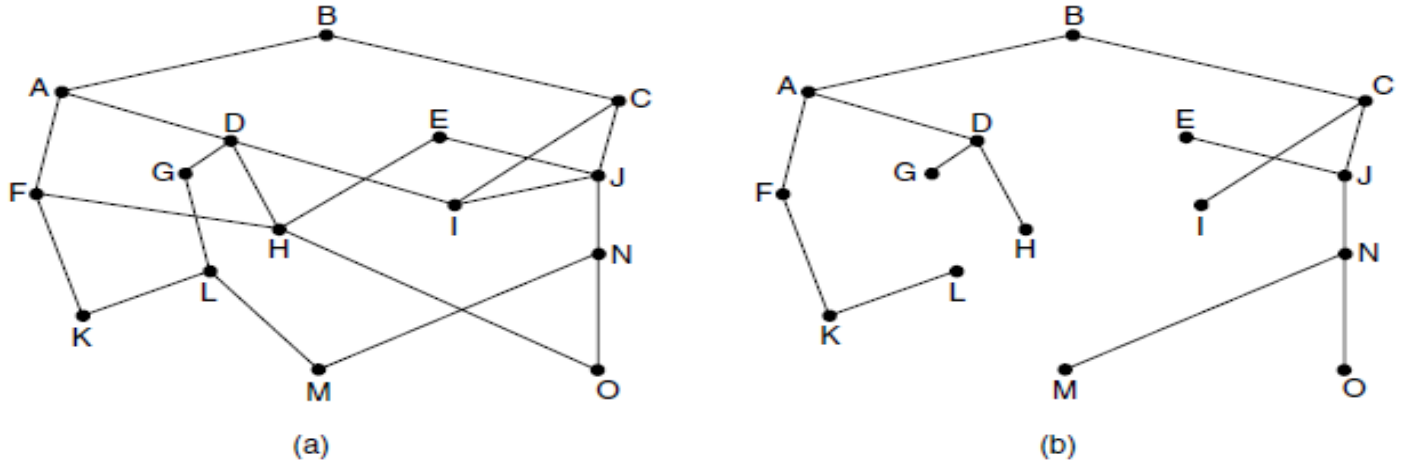


Figure 5-6. (a) A network. (b) A sink tree for router *B*.

The Optimality Principle-

about optimal routes without regard to network topology or traffic

- 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 and is illustrated in Fig. 5-6(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.
- **Refer to algorithm in 5-8 Dijkstra's for shortest path**

Shortest Path Algorithm

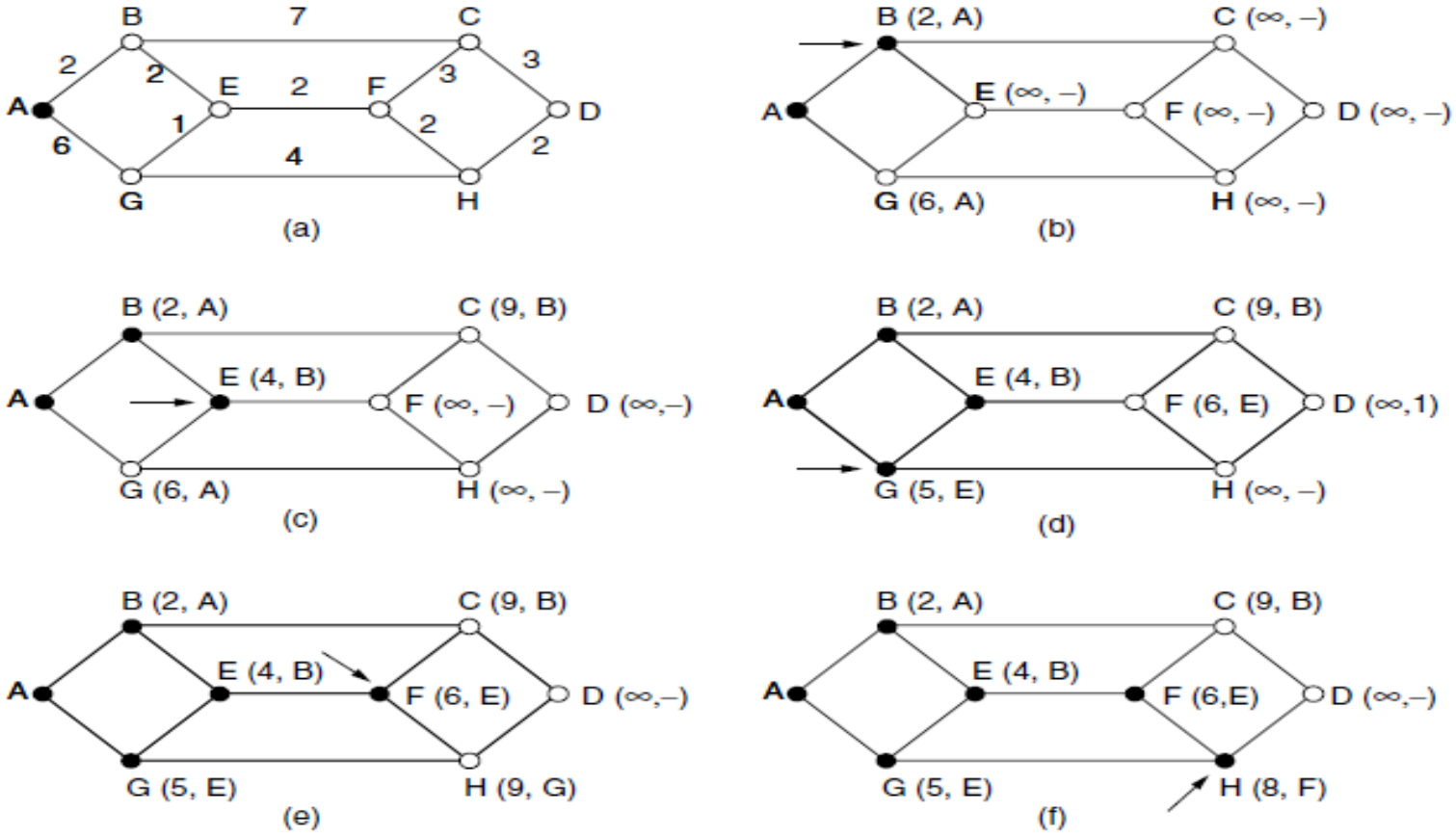


Figure 5-7. The first six steps used in computing the shortest path from *A* to *D*. The arrows indicate the working node.

FLOODING

- A simple local technique is **flooding**, in which every incoming packet is sent out on every outgoing line except the one it arrived on.
- Flooding obviously generates vast numbers of duplicate packets, in fact, an infinite number unless some measures are taken to damp the process.
- One such measure is to have a hop counter contained in the header of each packet that is decremented at each hop, with the packet being discarded when the counter reaches zero.
- Ideally, the hop counter should be initialized to the length of the path from source to destination.
- If the sender does not know how long the path is, it can initialize the counter to the worst case, namely, the full diameter of the network.

FLOODING

- Flooding is not practical for sending most packets
- First, it ensures that a packet is delivered to every node in the network. This may be wasteful if there is a single destination that needs the packet, but it is effective for broadcasting information
- Second, flooding is tremendously robust, Even if large numbers of routers are blown to bits (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.
- Flooding always chooses the shortest path because it chooses every possible path in parallel. Consequently, no other algorithm can produce a shorter delay

Distance Vector Routing(Bellman-Ford)

- 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 informationwith the neighbors.
- It was the original ARPANET routing algorithm and was also used in the Internetunder the name RIP.

Distance Vector Routing(Bellman-Ford)

- In distance vector routing, each router maintains a routing table indexed by,and containing one entry for each router in the network. This entry has two parts:

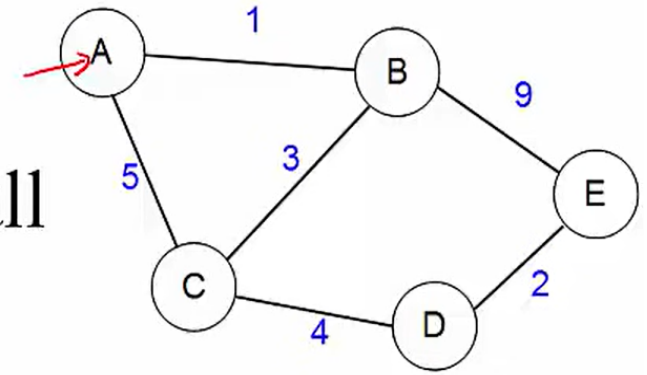
➢ the preferred outgoing line to use for that destination and

➢ an estimate of the distance to that destination.

Problems solved in class:

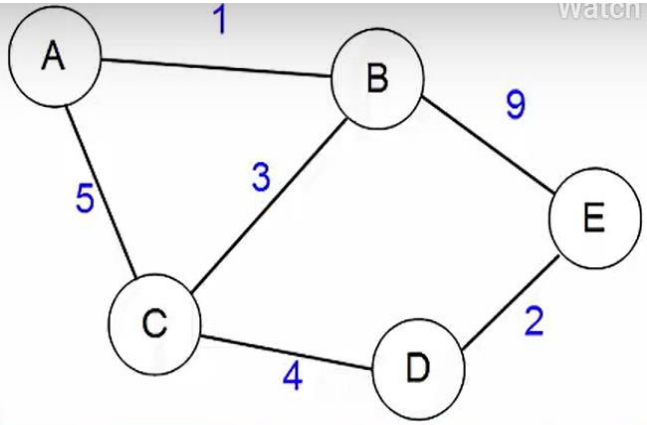
Distance Vector Routing(Bellman-Ford)

- Initial state at a node: distance (cost) to neighbors is known
- Final state at a node: distance (cost) to all nodes is known, and also the next-hop
- Need to handle
  - What information to exchange? (message format)
  - How to act on a message?
  - When to send a message?



Distance Vector Routing(Bellman-Ford)

- Each node maintains a routing table (distance vector)
  - Destination
  - Estimated cost to destination
  - Next hop via which to reach destination
- Initial state: Cost to



Dest	Cost	Next Hop
A	1	A
C	3	C
E	9	E

Initial Routing table at B

Dest	Cost	Next Hop
A	1	A
C	3	C
D	7	C
E	9	E

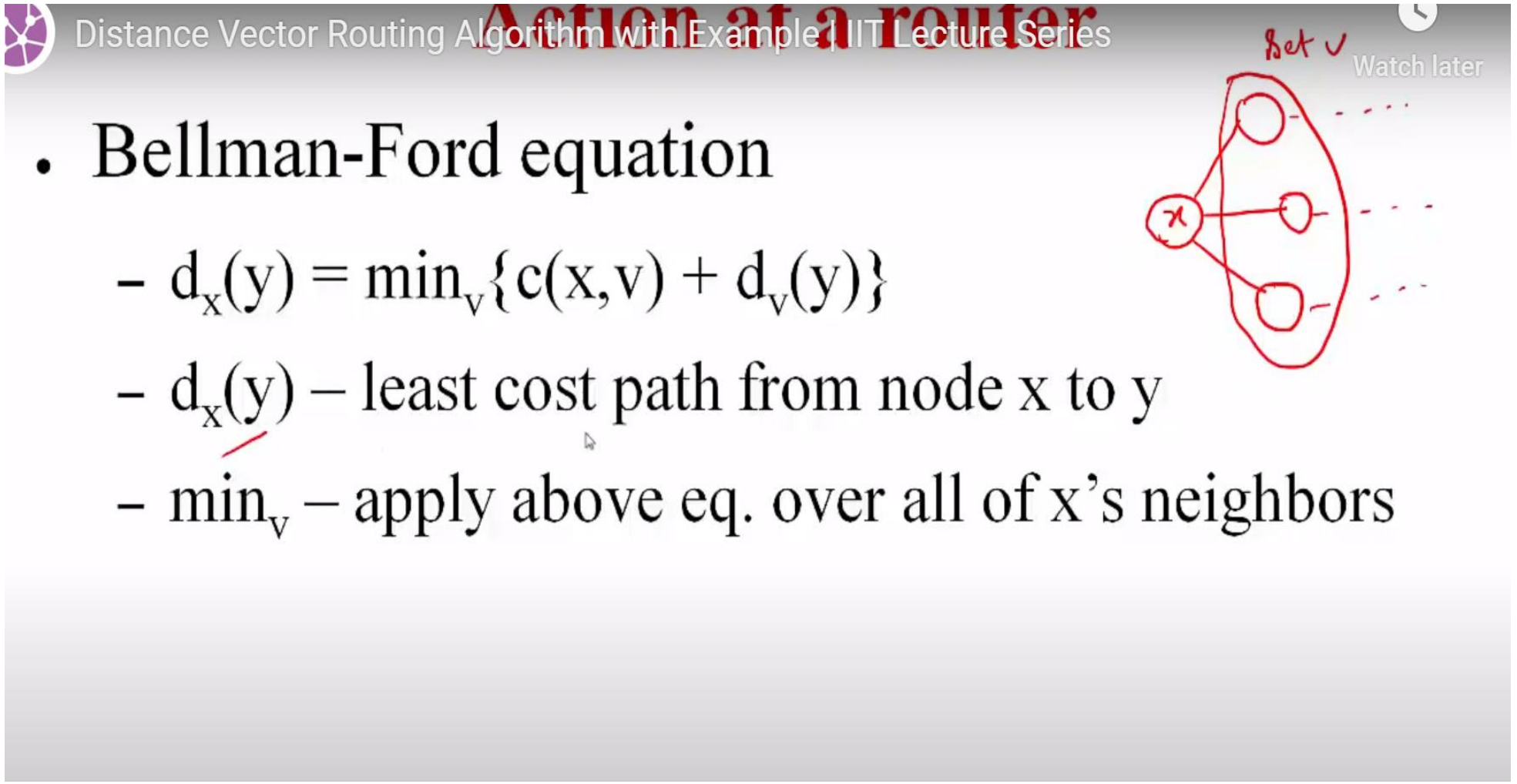
Final Routing table at B

Distance Vector Routing(Bellman-Ford)

Message Content

- Each node exchanges with all its neighbors "Routing Table" info
  - Destination and 'Estimated' cost to destination
  - Next hop information is not shared

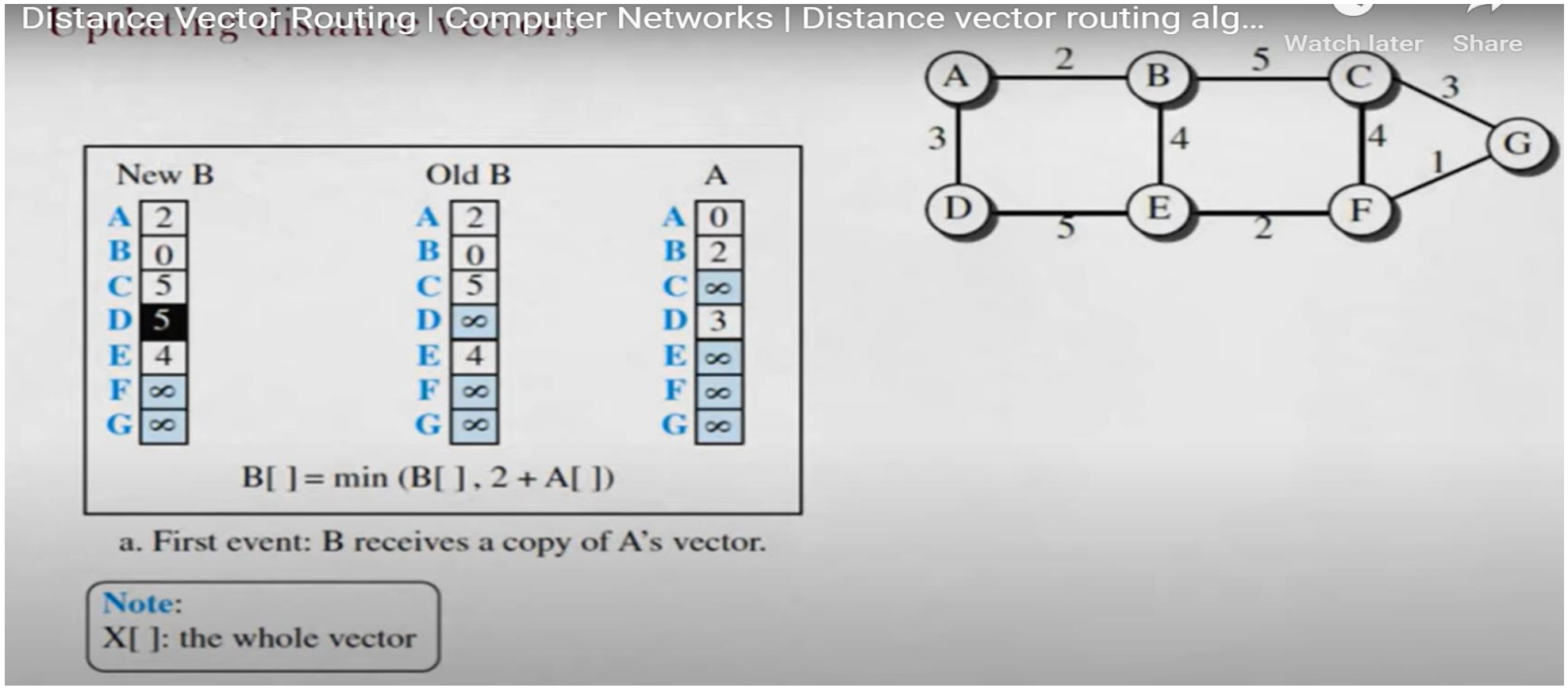
Distance Vector Routing(Bellman-Ford)



Distance Vector Routing(Bellman-Ford)

- On receiving a message from a neighbor v,
  - Update cost (estimate) to destinations based on above Bellman-ford equation; change next hop accordingly
  - For each y (destination in routing table of the received message)
    - $D_x(y) = \min\{\text{current estimate}, c(x,v) + D_v(y)\}$
  - Estimated costs finally converge to optimal cost after series of message exchanges

Distance Vector Routing(Bellman-Ford)



Note: please check for the problem solved in class

Count to infinity problem in Distance Vector routing

- The settling of routes to best paths across the network is called **convergence**.
- 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. Consider a router whose best route to destination *X* is long. If, on the next exchange, neighbor *A* suddenly reports a short delay to *X*, the router just switches over to using the line to *A* to send traffic to *X*. In one vector exchange, the good news is processed.

Count to infinity problem in Distance Vector routing

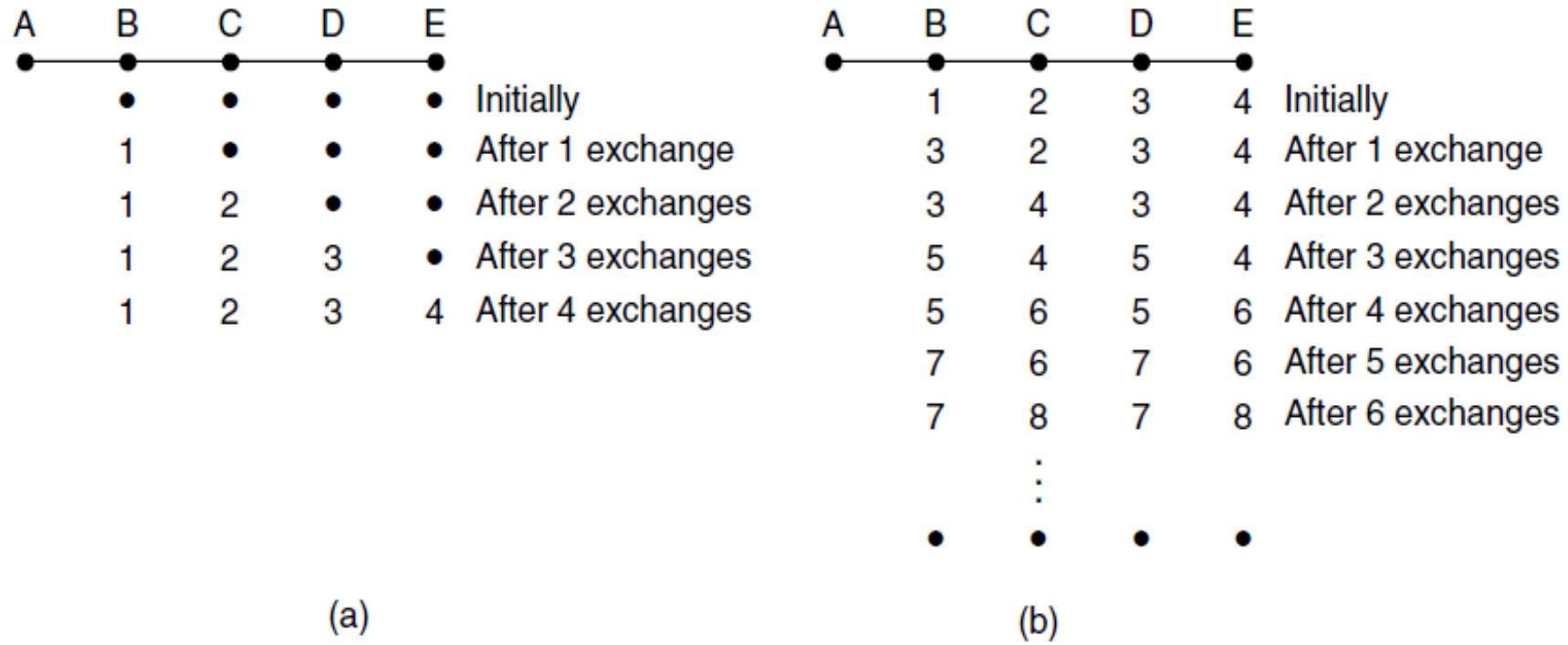


Figure 5-10. The count-to-infinity problem.

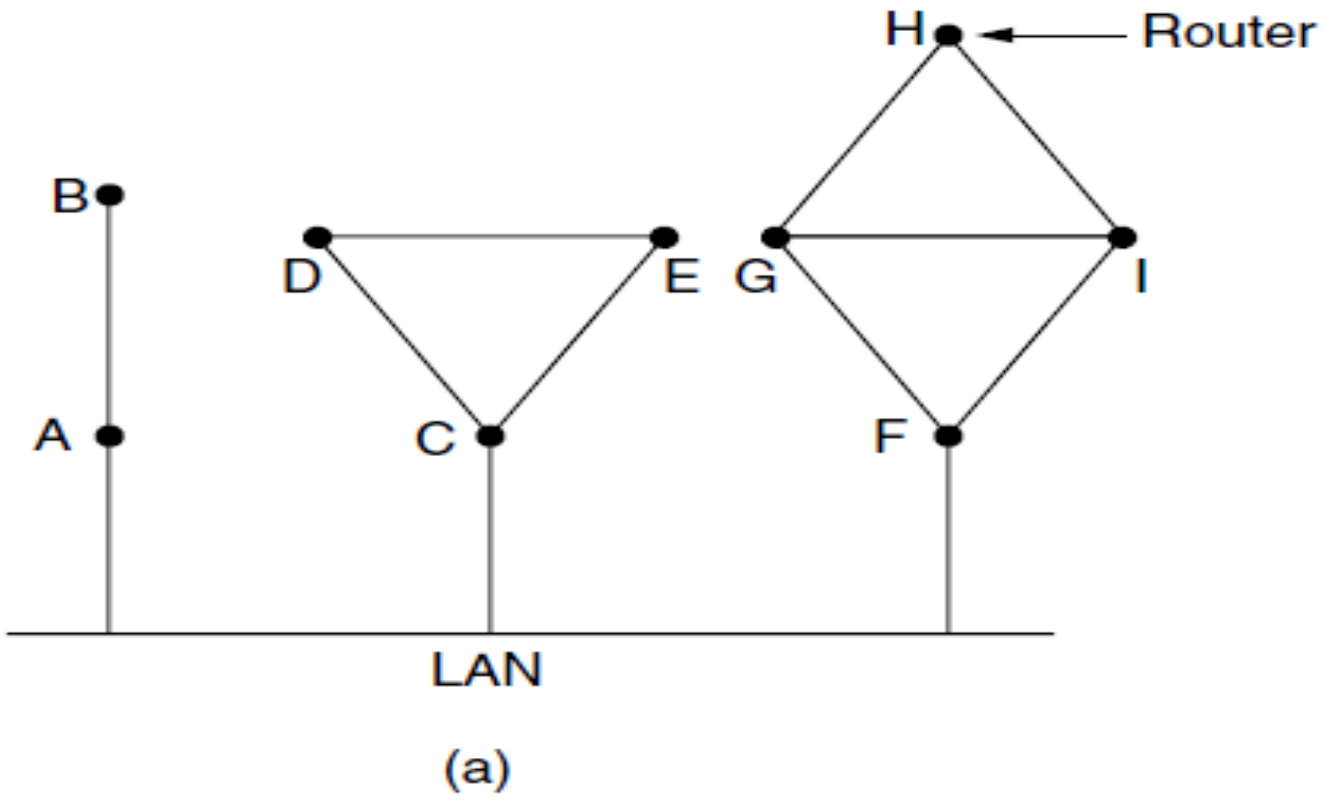
Link State Routing

- Distance vector routing was used in the ARPANET until 1979, when it was replaced by link state routing.
- The primary problem that caused its demise wasthat the algorithm often took too long to converge after the network topologychanged (due to the count-to-infinity problem)
- 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

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

Link State Routing

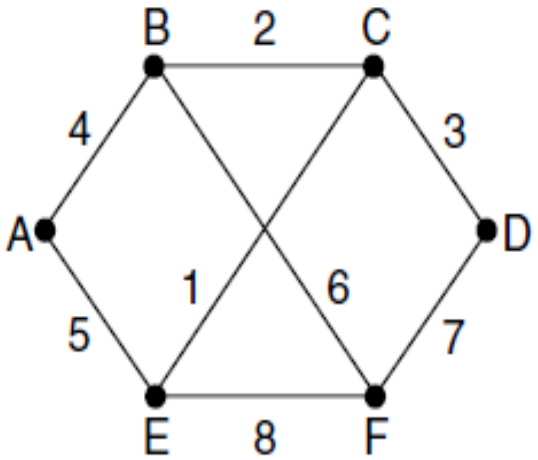


(a)

Link State Routing

- 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**.
- 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.

Link State Routing



(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)

Figure 5-12. (a) A network. (b) The link state packets for this network.

Link State Routing

- The most direct way to determine this delay is to send over the line a special **ECHO** packet that the other side is required to send back immediately. By measuring the round-trip time and dividing it by two, the sending router can get a reasonable estimate of the delay.

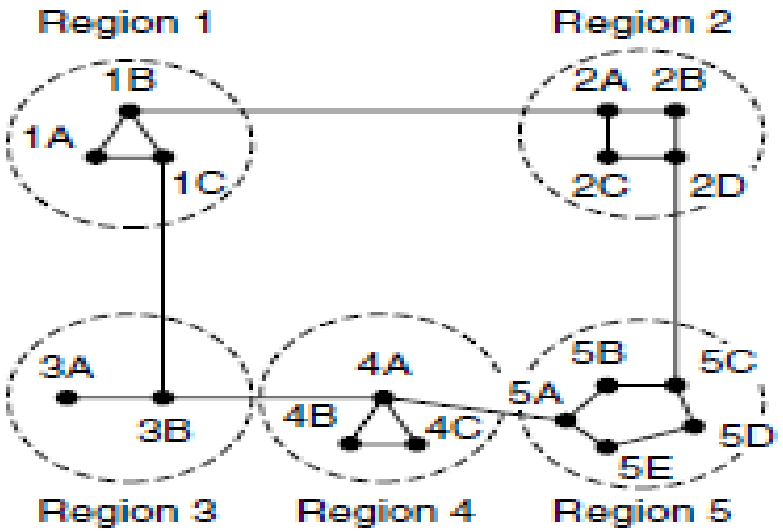
Hierarchical Routing

- As networks grow in size, the router routing tables grow proportionally. Not only is router memory consumed by ever-increasing tables, but more CPU time is needed to scan them and more bandwidth is needed to send status reports about them.
- At a certain point, the network may grow to the point where it is no longer. feasible for every router to have an entry for every other router, so the routing will have to be done hierarchically, as it is in the telephone network.

Hierarchical Routing

- 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.

Hierarchical Routing



(a)

Full table for 1A

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

(b)

Hierarchical table for 1A

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

(c)

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 iscalled **broadcasting**

Broadcast Routing

- An improvement is **multidestination routing**, in which each packet contains either a list of destinations or a bit map indicating the desired destinations. When a packet arrives at a router, the router checks all the destinations to determine the set of output lines that will be needed.
- The router generates a new copy of the packet for each output line to be used and includes in each packet only those destinations that are to use the line.

Broadcast Routing

- The idea for **reverse path forwarding** is elegant and remarkably simple once it has been pointed out (Dalal and Metcalfe, 1978). 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.
- The principal advantage of reverse path forwarding is that it is efficient while being easy to implement. It sends the broadcast packet over each link only once in each direction, just as in flooding, yet it requires only that routers know how to reach all destinations, without needing to remember sequence numbers (or useother mechanisms to stop the flood) or list all destinations in the packet

Broadcast Routing

- 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 oneit arrived on.

Draw Back:

- The only problem is that each router must have knowledge of some spanning tree for the method to be applicable. Sometimes this information is available (e.g., with link state routing, all routers know the complete topology, so they can compute a spanning tree) but sometimesit is not (e.g., with distance vector routing).

Multicast Routing

- Some applications, such as a multiplayer game or live video of a sports eventstreamed to many viewing locations, send packets to multiple receivers. Unlessthe group is very small, sending a distinct packet to each receiver is expensive.On the other hand, broadcasting a packet is wasteful if the group consists of, say,1000 machines on a million-node network, so that most receivers are not interestedin the message (or worse yet, they are definitely interested but are not supposed to see it).

Multicast Routing

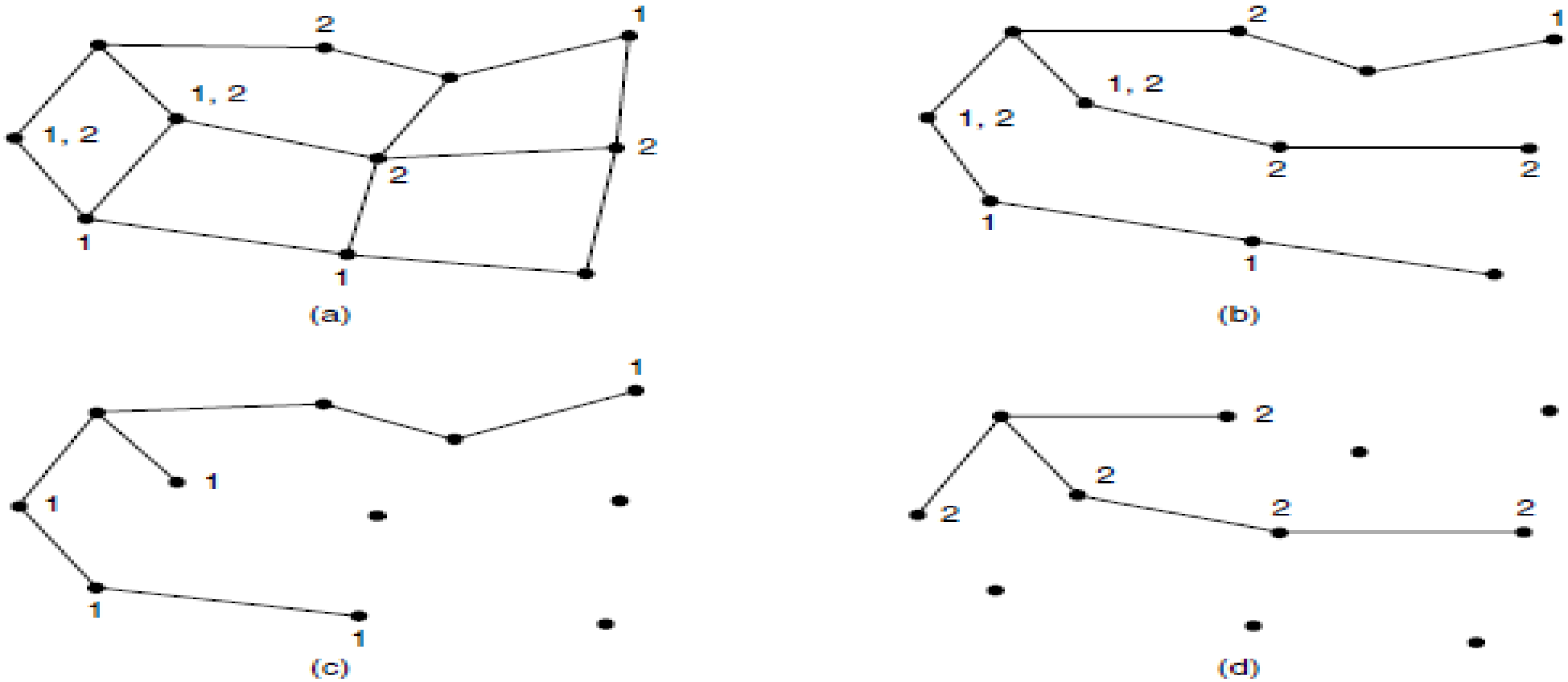


Figure 5-16. (a) A network. (b) A spanning tree for the leftmost router. (c) A multicast tree for group 1. (d) A multicast tree for group 2.

Routing for Mobile Hosts

- An alternative design uses **core-based trees** to compute a single spanning tree for the group (Ballardie et al., 1993). All of the routers agree on a root (called the **core** or **rendezvous point**) and build the tree by sending a packet from each member to the root.

Anycast Routing

- In anycast, a packet is delivered to the nearest member ofa group (Partridge et al., 1993). Schemes that find these paths are called **anycast routing**.

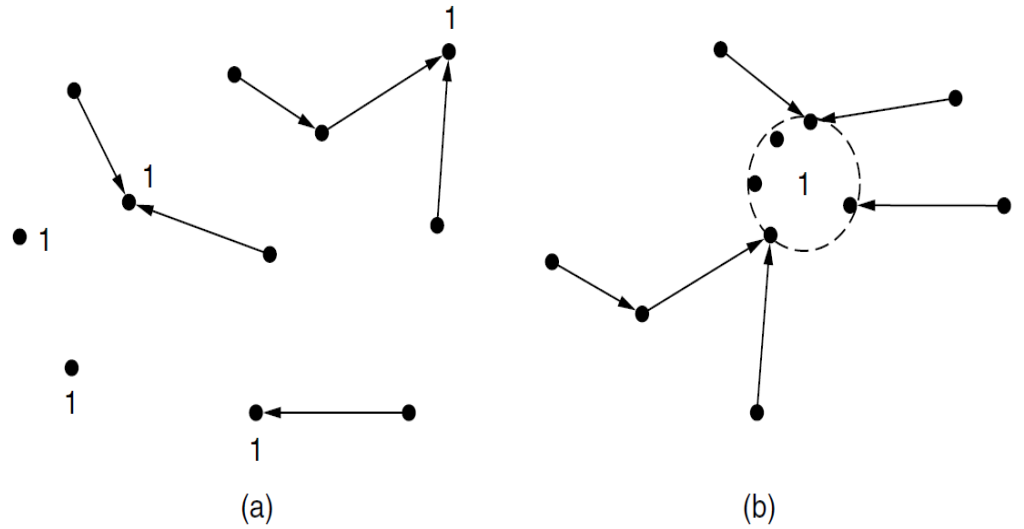


Figure 5-18. (a) Anycast routes to group 1. (b) Topology seen by the routing protocol.

Routing for Mobile Hosts

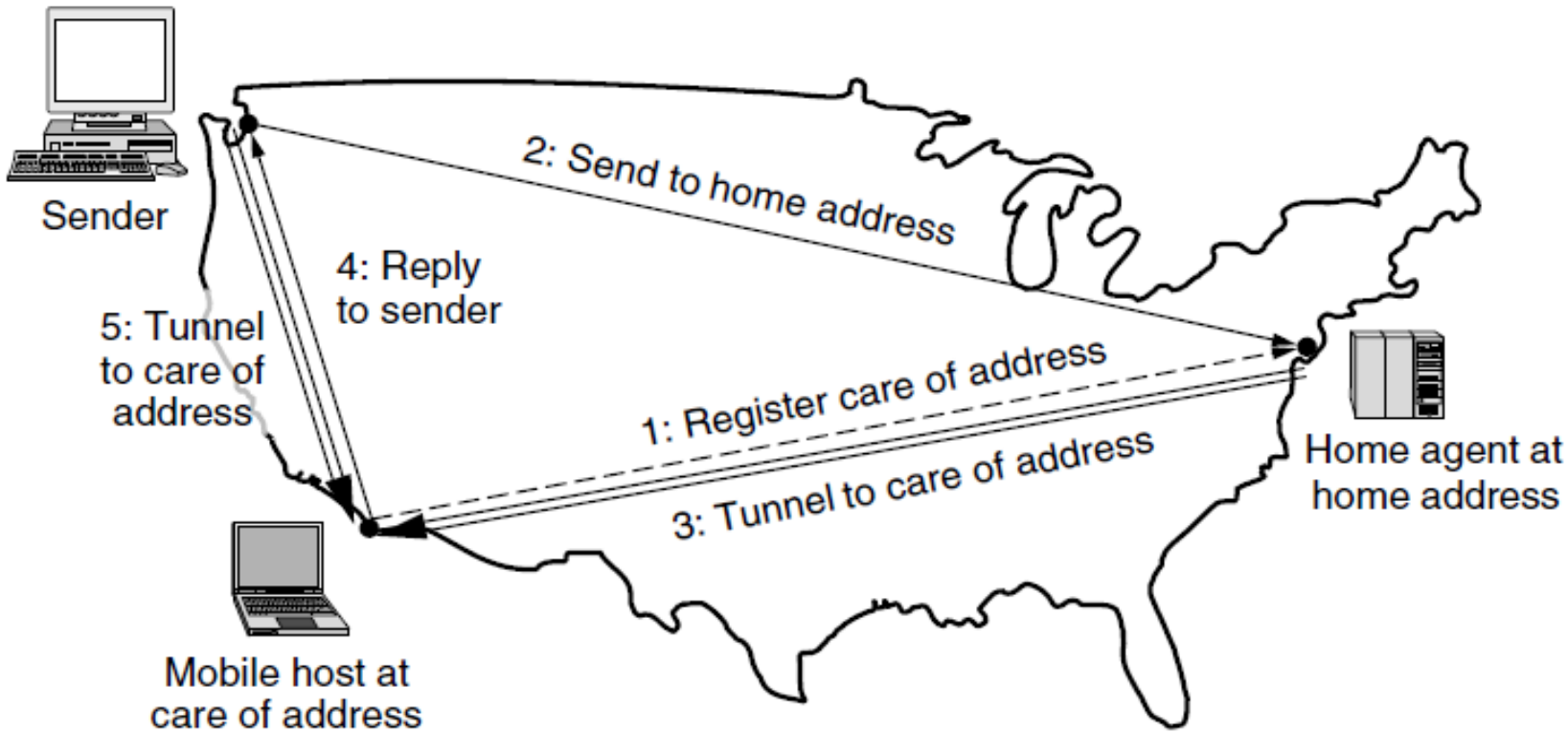


Figure 5-19. Packet routing for mobile hosts.

Routing for Mobile Hosts

- When the encapsulated packet arrives at the care of address, the mobile host unwraps it and retrieves the packet from the sender. The mobile host then sends its reply packet directly to the sender (step 4). The overall route is called **triangle routing** because it may be circuitous if the remote location is far from the home location.

Routing in Ad Hoc Networks

- We have now seen how to do routing when the hosts are mobile but the routers are fixed. An even more extreme case is one in which the routers themselves are mobile
- In all these cases, and others, each node communicates wirelessly and acts asboth a host and a router. Networks of nodes that just happen to be near each otherare called **ad hoc networks** or **MANETs** (**Mobile Ad hoc NETworks**).

Routing in Ad Hoc Networks

- **Route Discovery:**
- In AODV, routes to a destination are discovered on demand, that is, only when a somebody wants to send a packet to that destination
- **Note: Please go through page 385 and 390**

Routing in Ad Hoc Networks

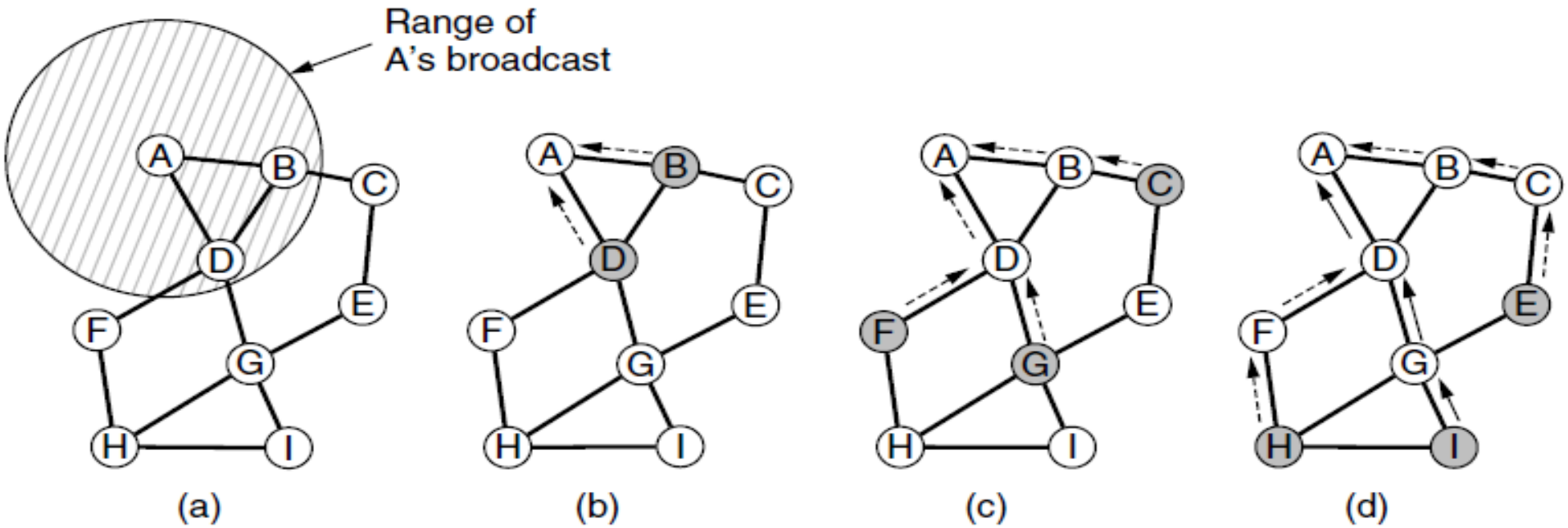


Figure 5-20. (a) Range of *A*'s broadcast. (b) After *B* and *D* receive it. (c) After *C*, *F*, and *G* receive it. (d) After *E*, *H*, and *I* receive it. The shaded nodes are new recipients. The dashed lines show possible reverse routes. The solid lines show the discovered route.

Routing in Ad Hoc Networks

- Suppose that a process at node *A* wants to send a packet to node *I*. The AODV algorithm maintains a distance vector table at each node, keyed by destination, giving information about that destination, including the neighbor to which to send packets to reach the destination. First, *A* looks in its table and does not find an entry for *I*. It now has to discover a route to *I*. This property of discovering routes only when they are needed is what makes this algorithm "on demand."
- **Note: Route request packet**

  Route Reply Packet(go through text book for this part)

  Time to live

CONGESTION CONTROL ALGORITHMS

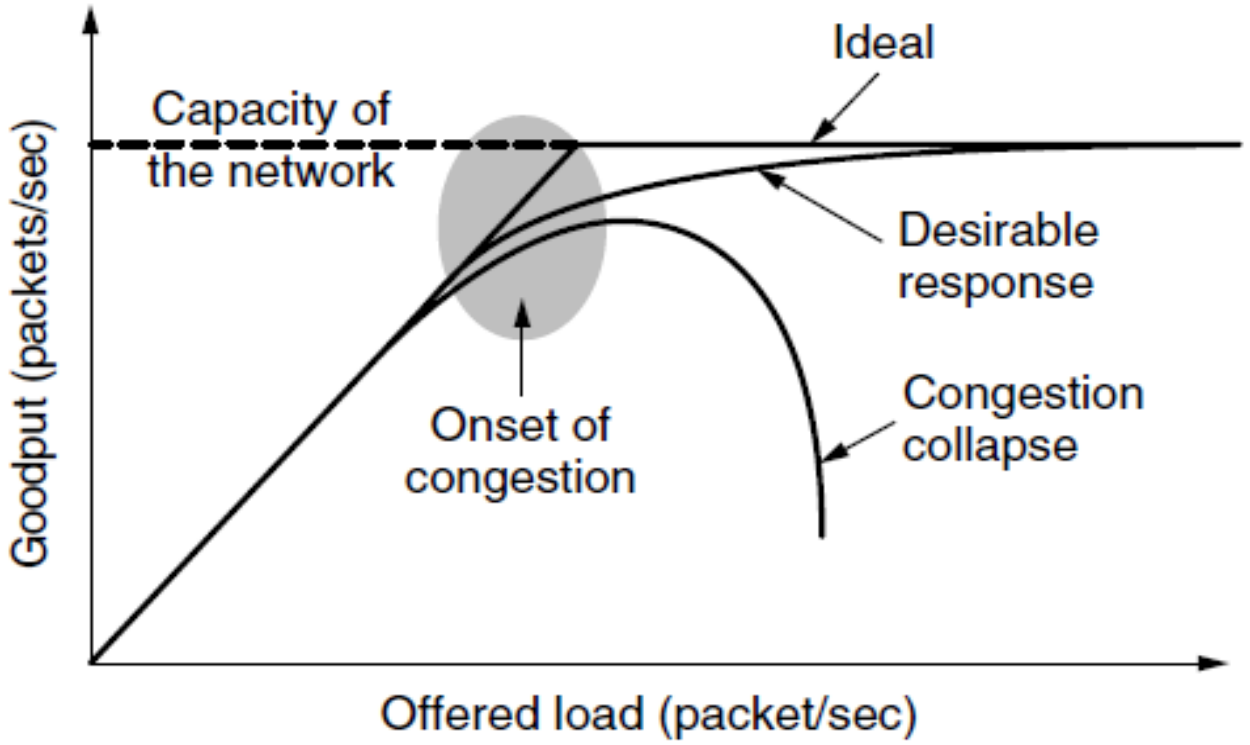


Figure 5-21. With too much traffic, performance drops sharply.

CONGESTION CONTROL ALGORITHMS

Terminologies and definitions:

- Too many packets present in (a part of) the network causes packet delay and loss that degrades performance. This situation is called **congestion**
- Unless the network is well designed, it may experience a **congestion collapse in** which performance plummets as the offered load increases beyond the capacity.
- **Goodput**, which is the rate at which *useful* packets are delivered by the network

Approaches to Congestion Control

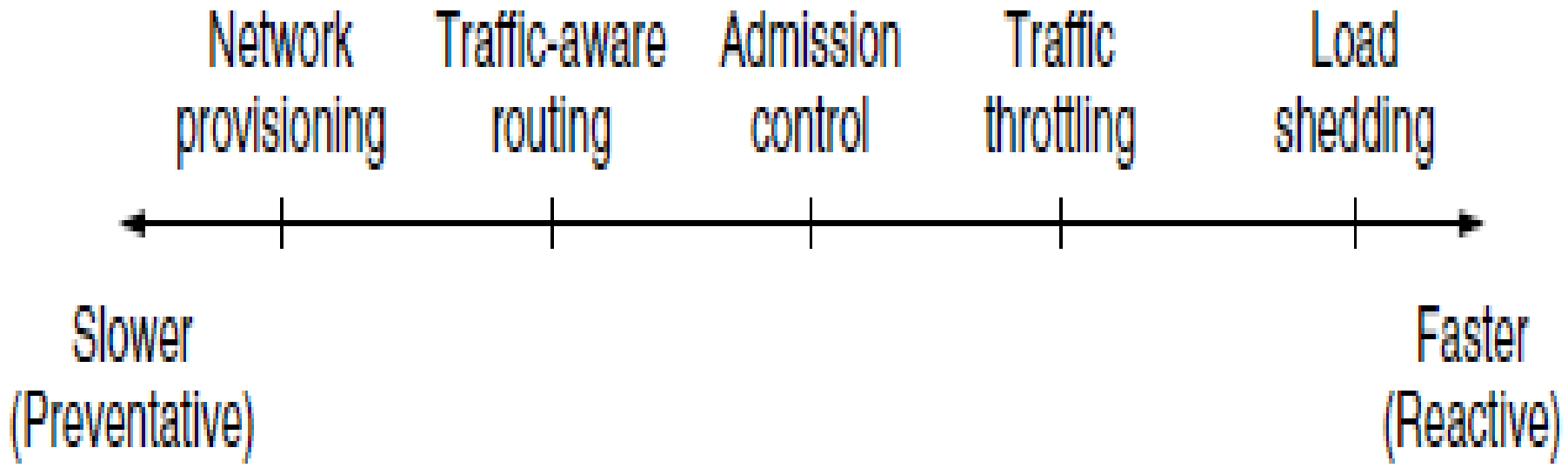


Figure 5-22. Timescales of approaches to congestion control.

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.
- As shown in Fig. 5-22, these solutions are usually applied on different time scales to either prevent congestion or react to it once it has occurred

Approaches to Congestion Control

- The most basic way to avoid congestion is to build a network that is well matched to the traffic that it carries
- Sometimes the 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.

Approaches to Congestion Control

- 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. This is called **traffic-aware routing.**
- However, sometimes it is not possible to increase capacity. 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**.

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 firstissue, routers can monitor the average load, queueing delay, or packet loss. In allcases, rising numbers indicate growing congestion.

Approaches to Congestion Control

- 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
- **Traffic-Aware Routing:**

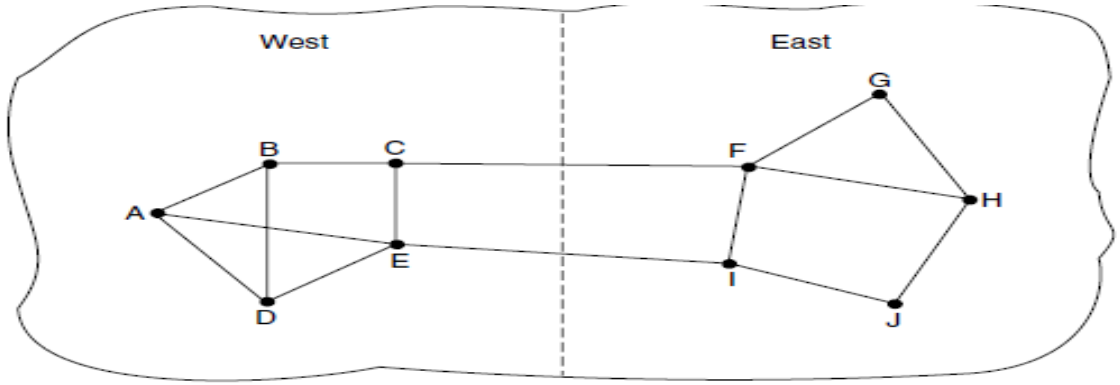


Figure 5-23. A network in which the East and West parts are connected by two links.

Approaches to Congestion Control

- **Admission Control:**The idea is simple: do not set up a new virtual circuit unless the network can carry the added traffic without becoming congestedA commonly used descriptor that captures this effect is the **leaky bucket** or **token bucket**

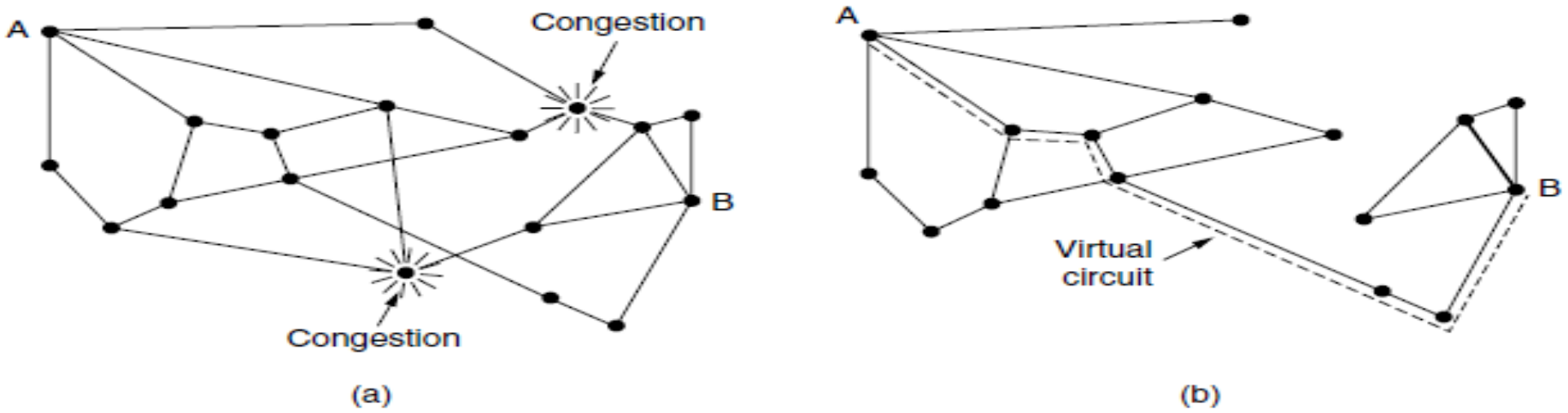


Figure 5-24. (a) A congested network. (b) The portion of the network that is not congested. A virtual circuit from *A* to *B* is also shown.

Approaches to Congestion Control

- 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

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

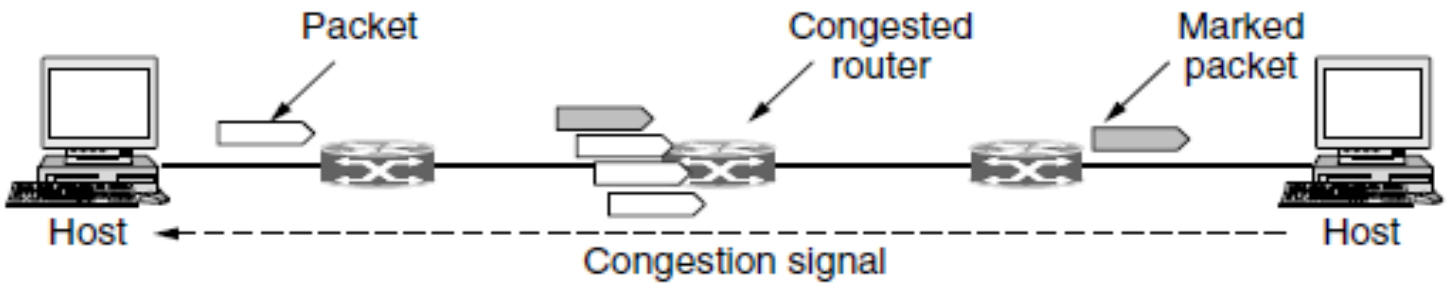


Figure 5-25. Explicit congestion notification

Approaches to Congestion Control

- **Traffic Throttling:** When congestionis imminent, it must tell the senders to throttle back their transmissions and slow down. This feedback is business as usual rather than an exceptional situation.

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 anymore 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

Approaches to Congestion Control

- **RED (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 bufferspace is really exhausted.
- RED routers improve performance compared to routers that drop packets only when their buffers are full, though they may require tuning to work well

QUALITY OF SERVICE

- Four issues must be addressed to ensure quality of service:

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

QUALITY OF SERVICE

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

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

Key terminologies to achieve QoS

- Traffic Shaping
- SLA(service level agreements)
- Traffic policing
- Leaky Buckets and token buckets
- Packet scheduling Algorithms(FIFO,FCFS Weighted fair queuing)

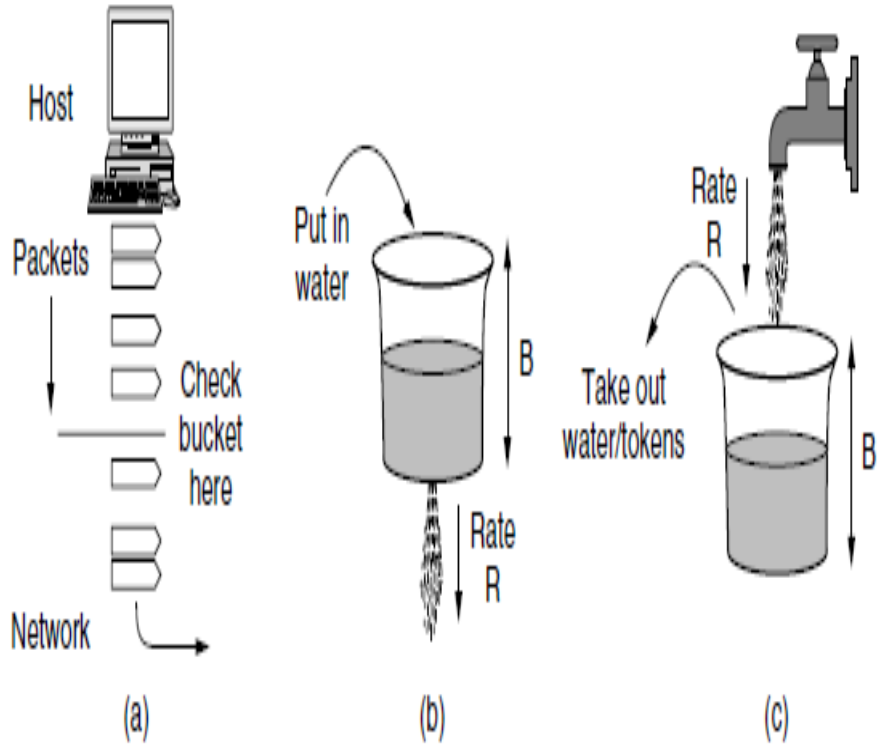


Figure 5-28. (a) Shaping packets. (b) A leaky bucket. (c) A token bucket.