ICS 153

Introduction to Computer Networks

Inst: Chris Davison

cbdaviso@uci.edu

ICS 153
The Network Layer

- Contents:
  - Connection-Oriented and Connectionless Service
  - Routing Algorithms
    - Non-Adaptive Routing
    - Adaptive Routing
  - Congestion Control Algorithms

ICS 153 Homework: Network Layer: Design

- Chapter 5 (Design)
- 1, 3, 4, 6, 10

ICS 153
Network Layer

- Recall
  - – The network layer is responsible for the routing of packets
  - – The network layer is responsible for congestion control

Connection-Orientated and Connectionless Service

- Network layers can offer two types of service to the transport layer:
  - Connection-oriented service
    - Connection setup required before communication begins
    - Network layer provides the Transport layer with a reliable service: in-sequence delivery, flow control
  - Connectionless service
    - No prior connection setup required
    - Packets are stored and forwarded one at a time by IMPs

Connection-Oriented Service

- How to provide connection-oriented service:
  - Set up a route (virtual circuit) between source and destination
  - That route is used for all traffic flowing over the virtual circuit
  - IMP maintains an internal table to tell which outgoing line to forward packet on for each active virtual circuit
  - Packets must contain a virtual circuit number so that each IMP can figure out how to forward them

Connection-Oriented Service: Analogy

- Public telephone Network
  - Set up a virtual circuit (dial a number)
  - Transmit data on the circuit (converse)
  - Close down the virtual circuit (hang up)
- Two users are provided with the illusion of a dedicated point-to-point channel
- Information is delivered to the receiver in the same order in which it is transmitted by the sender

Connectionless Service

- How to provide connectionless service:
  - Send the packet into the network and allow the network to forward it however it likes
  - IMPs maintain routing tables to look up the next IMP for each arriving packet
  - each packet must contain a destination address so the IMPs can make routing decisions

Connectionless Service: Analogy

- Postal Service:
  - Each packet (letter) is transported as an individual entity
  - Each packet (letter) must carry the complete destination address
  - If a packet (letter) is lost, error control is the user’s responsibility
  - Packets (letters) do not necessarily arrive in the order sent

Connectionless vs. Connection-Oriented Services

- Connection Setup Procedure:
  - – Connection-oriented service
    - Explicit setup and tear-down required
    - For short, transaction oriented communication, the delay of connection setup may be expensive
  - – Connectionless Service
    - No setup or tear-down required
    - For long, continuous communication, the overhead of packet headers may be

Connectionless vs. Connection-Oriented Services

- Header Overhead
  - Connection-oriented service
    - Only the virtual circuit number
  - Connectionless service
    - The full destination address is required

Connectionless vs. Connection-Oriented Services

- Message Sequence
  - Connection-oriented service
    - Sequence automatically maintained
  - Connectionless service
    - Destination may have to resequence out-of-sequence messages

Connectionless vs. Connection-Oriented Services

- Vulnerability
  - Connection-oriented service
    - Vulnerable: If an IMP crashes, all virtual circuits passing through it have to be aborted and re-established
  - Connectionless Service
    - Robust: If an IMP goes down, only hosts whose packets were queued at the time of the crash are lost. Other packets will be routed dynamically.

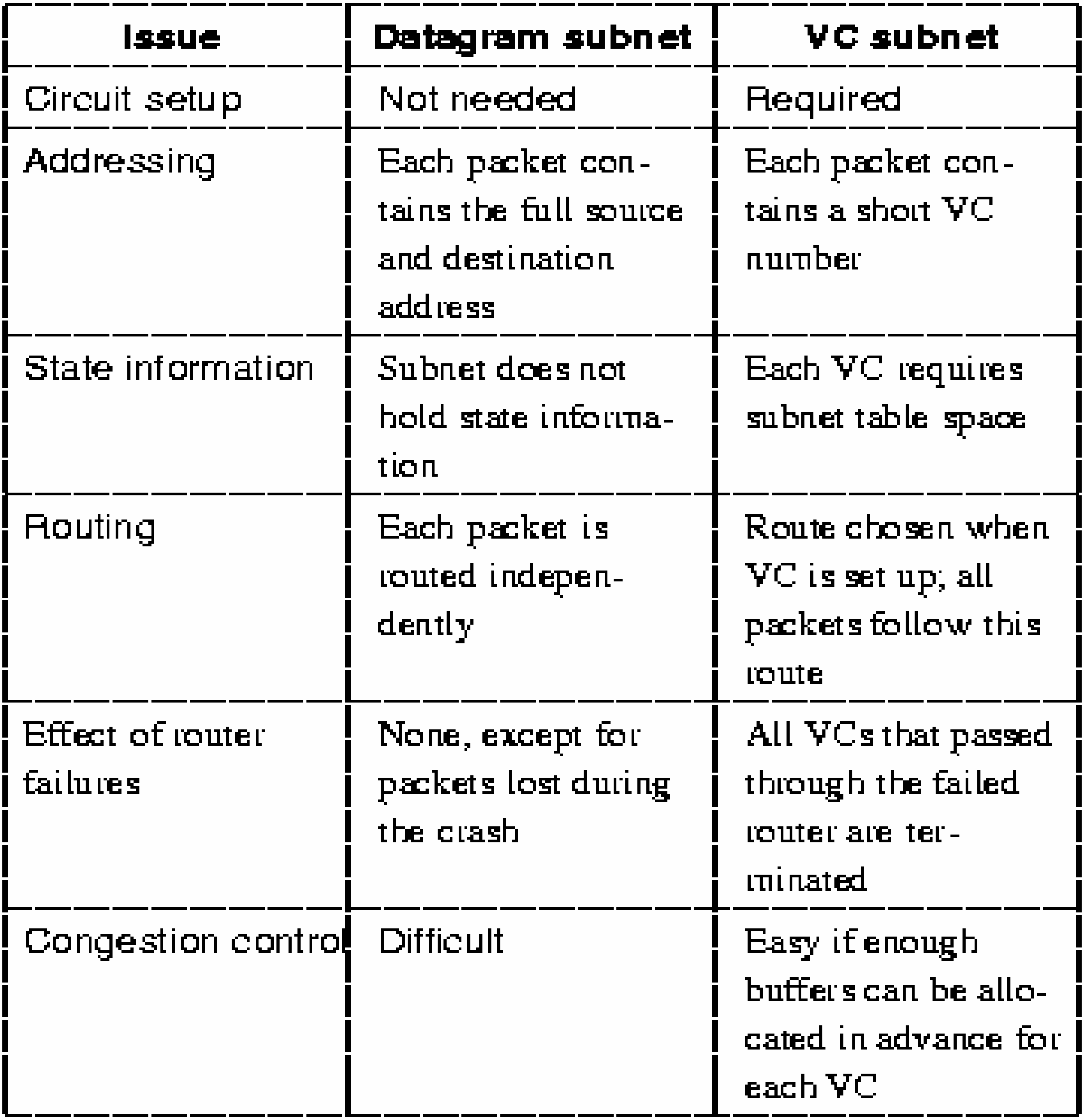
Connectionless vs. Connection-Oriented Services

- Guaranteed service:
  - Connection-oriented service
    - Can provide guarantees on the delays and throughputs of packets being sent
  - Connectionless service
    - It is very difficult or provide guarantees for timely packet delivery

VC vs. Datagram Subnets

- Virtual Circuit subnet
  - pre-established end to end route between sender and receiver
- Datagram subnet
  - each packet independently routed between sender and receiver

Subnet Structure



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

Summary

- Connection-oriented service
  - Is useful for applications which prefer in-sequence delivery of packets. It is also preferable for applications that require guaranteed service
- Connectionless service
  - Provides flexibility in the routing and handling of individual packets and is robust in the face of IMP crashes

Summary

- Where to put the complexity?
  - In connection-oriented service the complexity is at the network (in the subnet).
  - In connectionless service the complexity is in the transport layer (the hosts).

Routing Algorithms

- An IMP executes a routing algorithm to decide which output line an incoming packet should be transmitted on
- In connection-oriented service, the routing algorithm is performed only during connection setup
- In connectionless service, the routing algorithm is performed as each packet arrives

Routing Algorithms

- Two types of Routing algorithms
  - Non-Adaptive (static) Routing Algorithms
  - Adaptive (dynamic)Algorithms
- Hierarchical Routing is used to make these algorithms scale to large networks

Non-Adaptive Routing Algorithms

- Non-adaptive routing algorithms do not base their routing decisions on the current state of the network
- Examples:
  - Shortest Path Routing
  - Flooding

Shortest Path Routing

- For a pair of communicating hosts, there is a shortest path between them
- Shortness may be defined by:
  - Number of IMP hops
  - Geographic distance
  - Link Delay
  - Queue length
  - cost ($) of link

Computing the Shortest Path

- Dijkstra’s Shortest Path Algorithm
  - Draw Nodes as circles. Fill in a circle to mark it as a “permanent node”
  - Set the current node equal to the source node
  - For the current node:
    - Mark the cumulative distance from the current node to each non-permanent adjacent node. Also mark the name of the current node. Do not do this marking if the adjacent node already has a shorter cumulative distance listed.
    - Mark the non-permanent node with the shortest listed cumulative distance as permanent and set the current node equal to it.
    - Repeat

Shortest Path Example

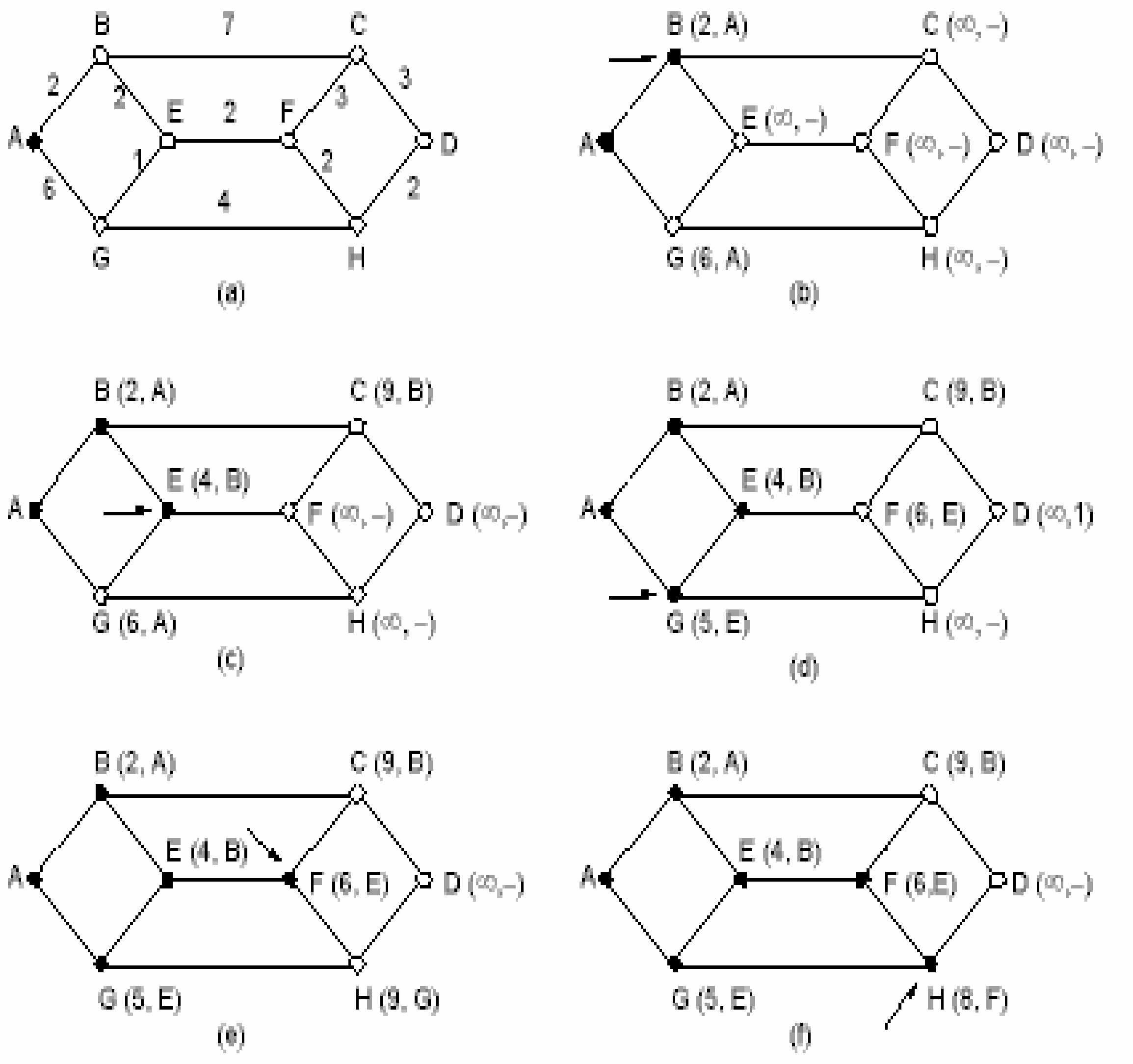


Fig. 5-6. The first five steps used in computing the shortest path from *A* to *D*. The arrows indicate the working node.

Shortest Path Routing

- Non-adaptive, if:
  - geographical distance is used as edge weights
  - maximum link throughputs are used as edge weights
  - number of IMP hops are used as edge weights

Flooding Algorithm

- Every incoming packet is sent out on every outgoing line except the one it arrived on
- Problem:
  - Vast number of duplicated packets

Reducing Flooding Algorithm’s Duplicate Packets

- Solution 1
  - Have a hop counter in the packet header
  - IMPs decrement each arriving packet’s hop counter
  - IMPs discard a packet with hop count = 0
  - Ideally, the hop counter should be initialized to the length of the packet from the source to the destination

Reducing Flooding Algorithm’s Duplicate Packets

- Solution 2
  - Require the first IMP hop to put a sequence number in each packet it receives from its hosts
  - Each IMP maintains a table listing the sequence numbers it has seen from each first-hop IMP. The IMP can then discard packets it has already seen.

Reducing Flooding Algorithm’s Duplicate Packets

- Solution 3
  - Selective Flooding
    - Do not send out packets on every outgoing line
    - Only send packets on outgoing lines that are approximately in the right direction

Flooding: Possible Applications

- Military Applications
  - Large number of IMPs is desirable
  - If one IMP is taken out (bombed?) flooding will still get packets to their destinations
- Distributed Databases
  - Simultaneous updates of multiple databases can be done with a single packet transmission

Flooding: Possible Applications

- Metric
  - By definition, flooding always selects the shortest path

Flow-Based Routing

- Considers topology and *load* to optimize routing.
- If the subnet capacity and the average packet flow are known it is possible to compute average packet delay

Flow Based routing: Problems

- Topology, capacity, traffic patterns must be known in advance.
- Any changes in the above will result in non-optimal routing

Non-Adaptive Algorithms

- Problems:
  - If traffic levels in different parts of the subnet change dramatically and often, non-adaptive routing algorithms are unable to cope with these changes
  - Lots of computer traffic is bursty, but non-adaptive routing algorithms are usually based on average traffic conditions

Adaptive Routing Algorithms

- Three types:
  - Centralized adaptive routing
  - Isolated adaptive routing
  - Distributed adaptive routing

Centralized Adaptive Routing

- Routing table adapts to network traffic
- A routing control center is somewhere in the network
- Periodically, each IMP forwards link status information to the control center
- The center can, with Dijkstra’s shortest path algorithm, computer the best routes
- Best routes are dispatched to each IMP

Problems with Centralized Algorithms

- Vulnerability
  - If the control center goes down, routing becomes non-adaptive
- Scalability
  - The control center must handle a great deal of routing information, especially for larger networks

Isolated Adaptive Routing Algorithms

- Routing decisions are made only on the basis of information available locally in each IMP
- Examples
  - Hot Potato
  - Backward Learning

Hot Potato Routing

- When a packet arrives, the IMP tries to get rid of it as fast as it can by putting it on the output line that has the shortest queue
- Hot Potato does not care where the output line leads
- Not very effective

Backward Learning Routing

- Packet headers include destination and source addresses. They also include a hop counter
- Network nodes, initially ignorant of network topology, acquire knowledge of the network state as packets are handled.

Backward Learning

- Algorithm
  - Routing is originally random
  - A packet with a hop count of one is from a directly connected node, thus, neighboring nodes are identified with their connecting links
  - A packet with a hop count of two is from a source two hops away
  - As packets arrive, the IMP compares the hop count for a given source address with the minimum hop count already registered. If the new one is less, it is substituted for the

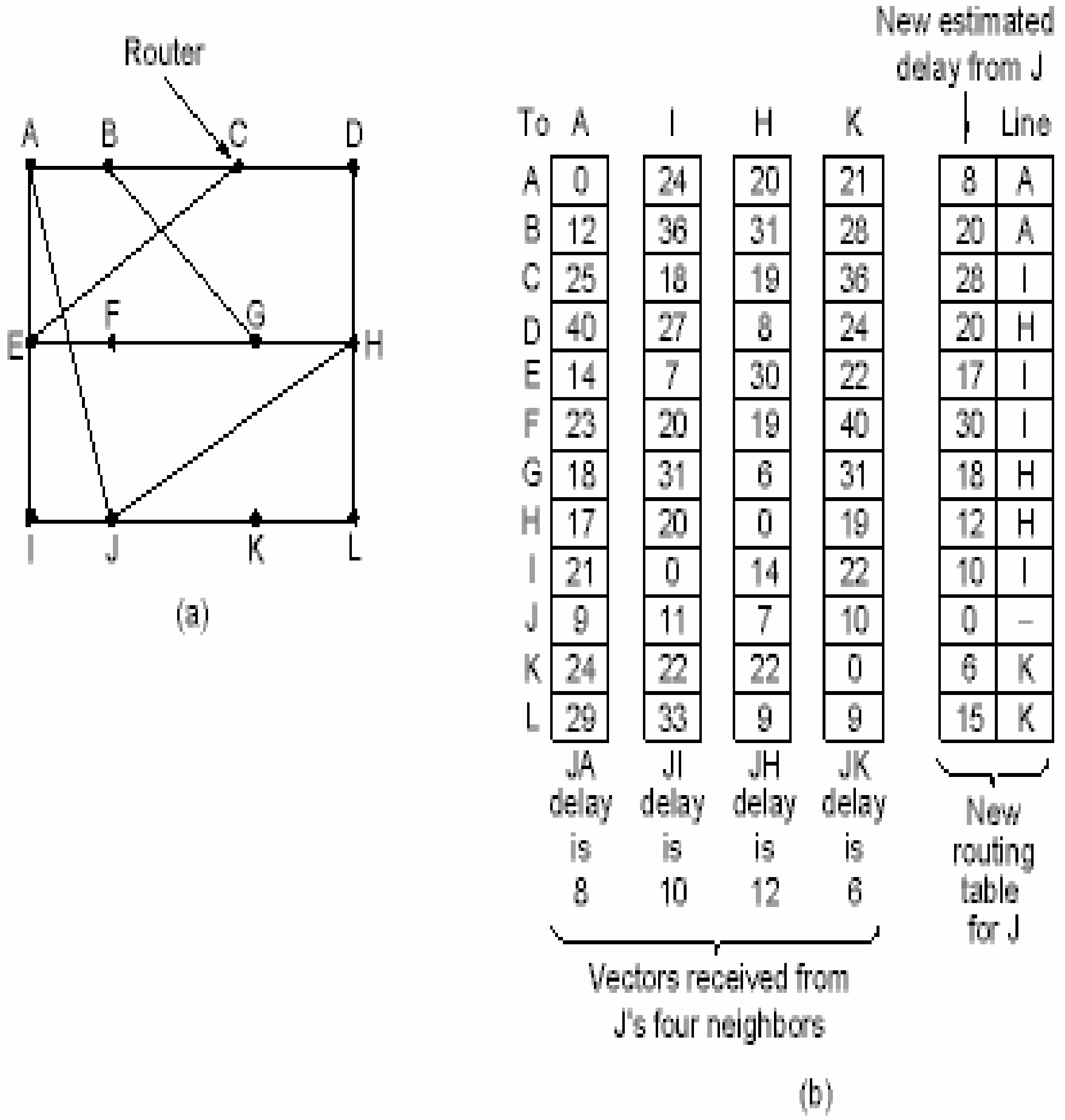
Distributed Routing Algorithms

- Each IMP periodically exchanges routing information (e.g. estimated time delay, queue length, etc.) with its neighbors
- Examples:
  - Distance Vector Routing
  - Link State Routing

Distance Vectors

- Each IMP, or router, maintains lists of best-known distances to all other known routers. The lists are called “vectors”
- Each router is assumed to know the exact distance (in delay, hop count, etc.) to other routers directly connected to it
- Periodically, vectors are exchanged between adjacent routers and each router updates its vectors

Distance Vectors



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

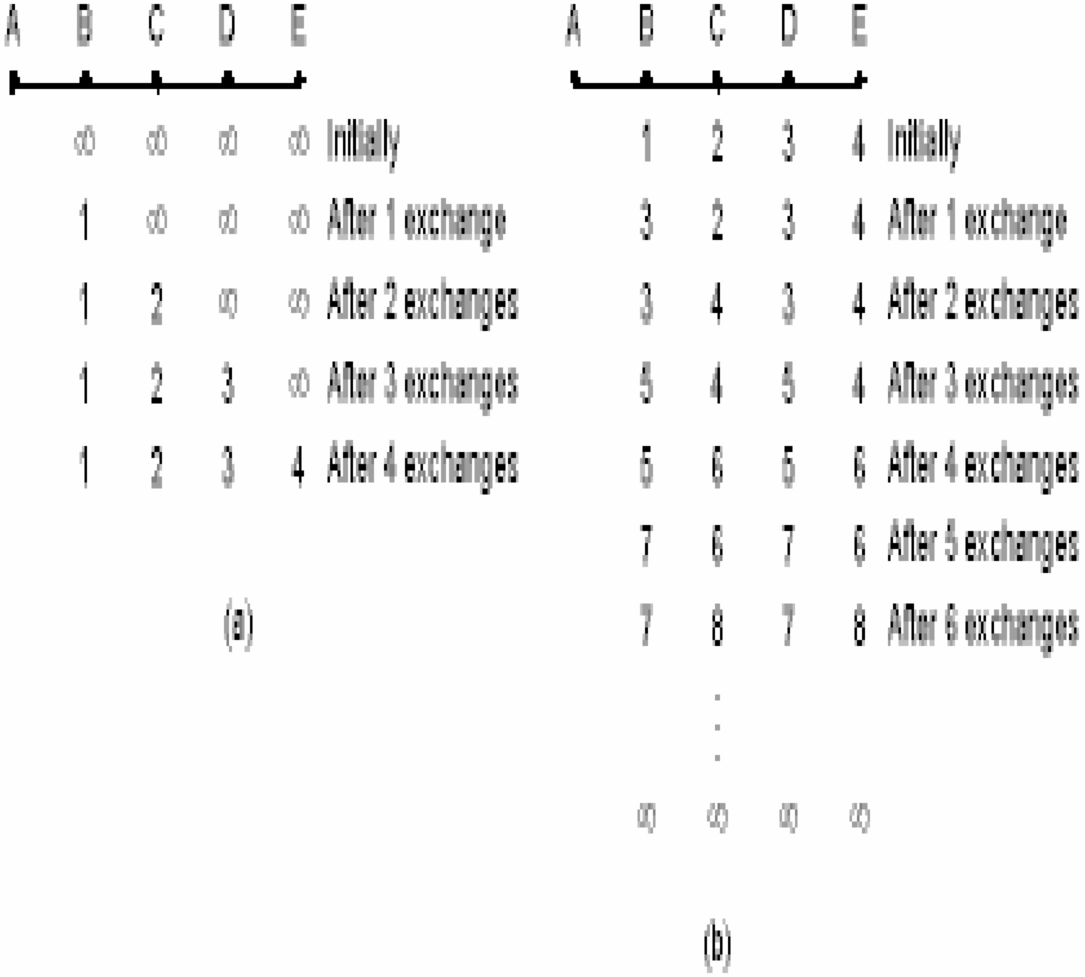
(b)

Fig. 5-10. (a) A subnet. (b) Input from *A*, *I*, *H*, *K*, and the new routing table for *J*.

Distance Vector: Problem

- Count-to-infinity
  - With distance vector routing, good news travels fast, but bad news travels slowly
  - When a router goes down it can take a long time before all the other routers become aware of it

Count-to-Infinity



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

(a)

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

(b)

Fig. 5-11. The count-to-infinity problem.

Link State Routing

- Each router measures the distance (in delay, hop count, etc.) between itself and its adjacent routers
- The router builds a packet containing all these distances. The packet also contains a sequence number and an age field
- Each router distributes these packets using flooding

Link State Routing

- To control flooding, the sequence numbers are used by routers to discard flood packets they have already seen from a given router
- The age field in the packet is an expiration date. It specifies how long the information in the packet is good for
- Once a router receives all the link state packets from the network, it can reconstruct the complete topology and compute a shortest path between itself and any other

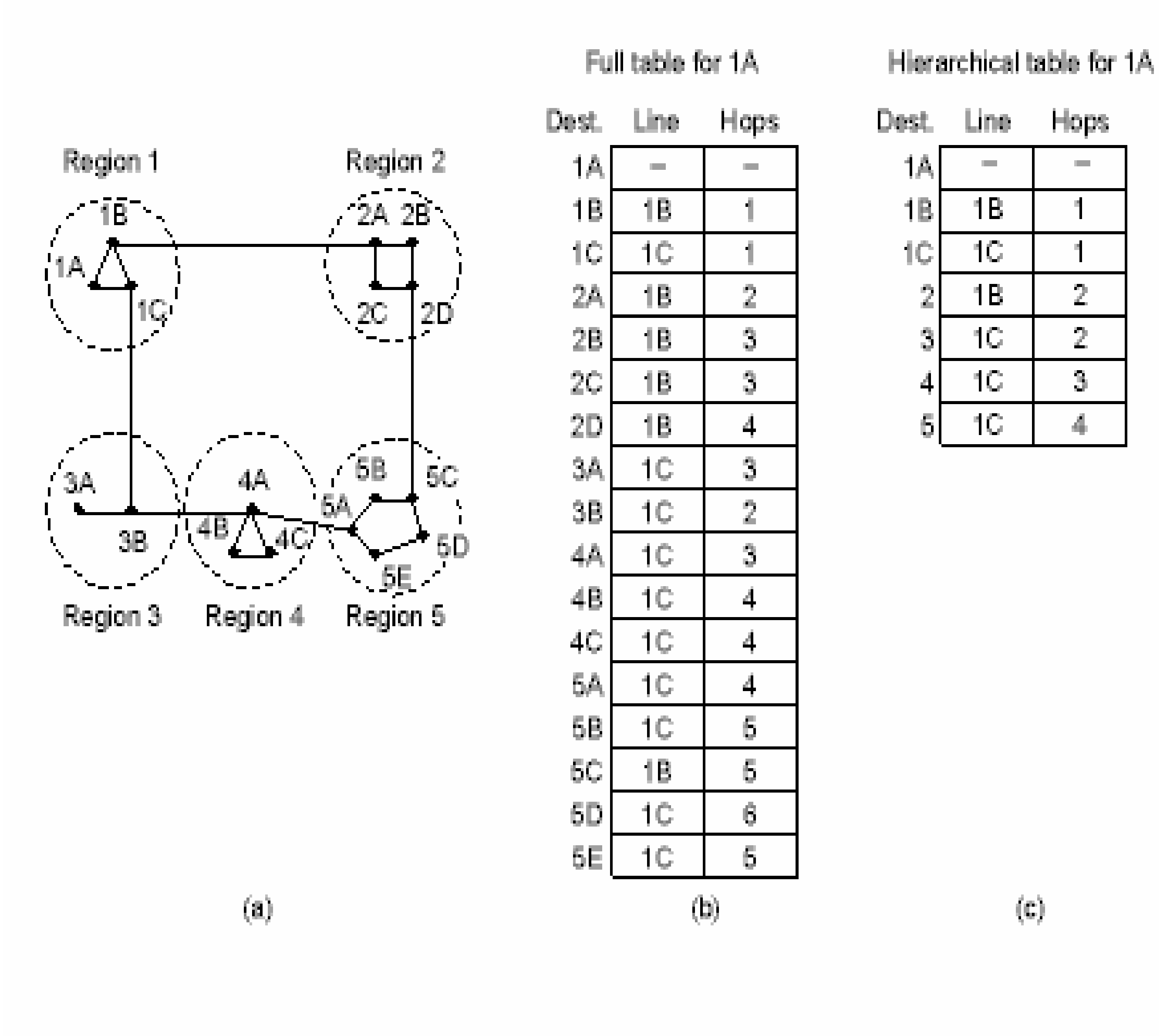
Hierarchical Routing

- All routing algorithms have difficulties as the network becomes large
- For large networks the routing tables grow very quickly, and so does the number of flood packets
- How can this be reduced? Hierarchical routing

Hierarchical Routing

- Segment the network into regions
- Routers in a single region know all the details about other routers in that region, but not of the details about routers in other regions
- Analogy: Telephone Area Codes

Hierarchical Routing Example



Full table for 1A

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

Hierarchical table for 1A

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

Fig. 5-17. Hierarchical routing.

Multicast Routing

- Widely-separated groups of hosts communicating with other groups of hosts
- A host will join a multicast group
- Routers will send multicast addressed packets out on lines only associated with a multicast group

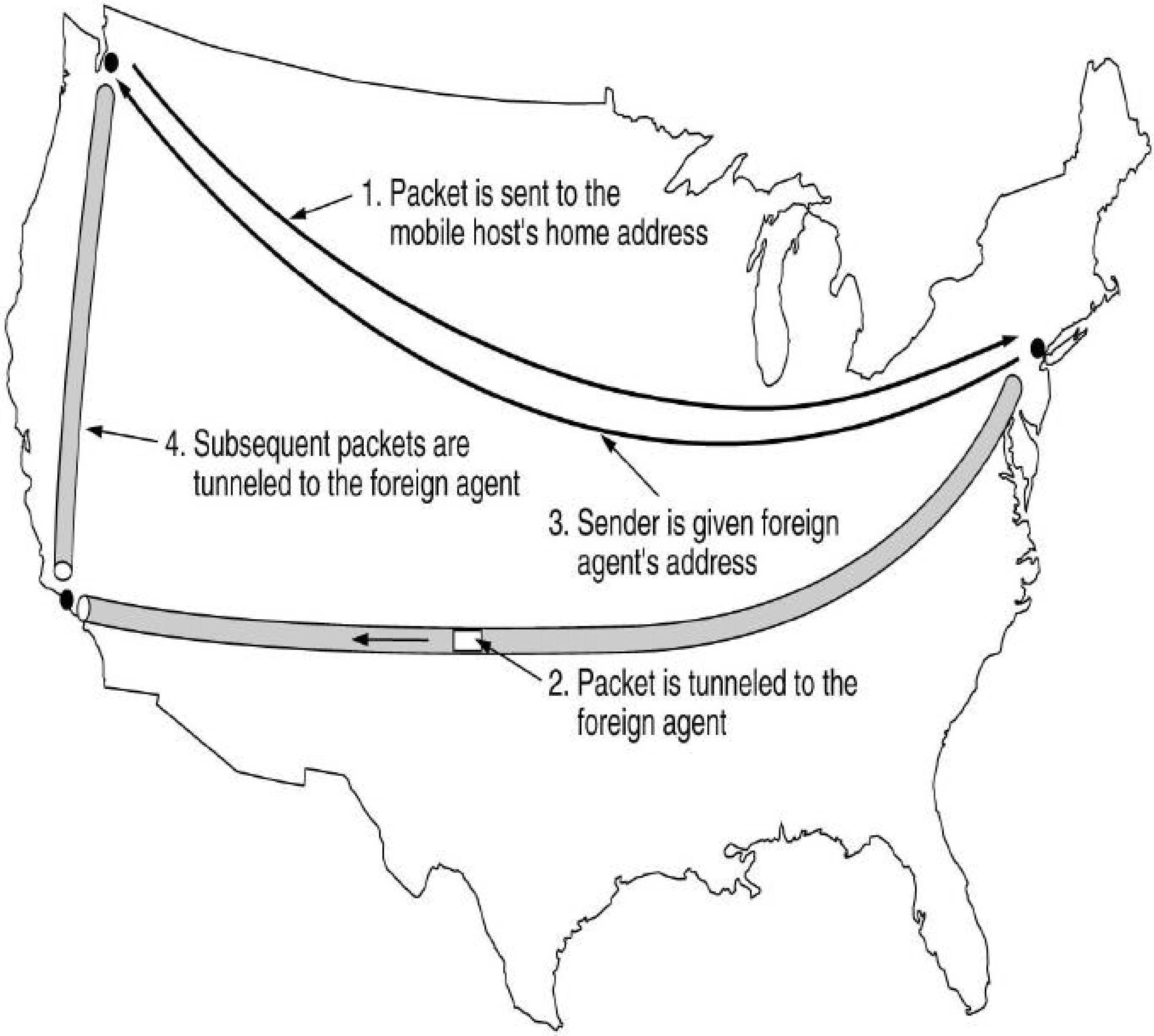
Multicast Routing

- Typical use of multicast routing: Machine Imaging
  - One image server sends a packet to the multicast group address
  - Routers send the packet out on all lines designated to group members
  - Hosts listening to the multicast group will receive the packet

Mobile IP Routing

- Used for mobile computing hosts (PDAs, laptops, etc.)
- Each host has a permanent *home location* (area) and a *home agent* who tracks the host’s whereabouts.
- Every connected LAN (802.11, etc.) is considered an *area*.
- Each area has a *foreign agent* which keeps track of visiting hosts.
  - Hosts must *register* with a foreign agent

Mobile IP Routing


1. Packet is sent to the mobile host's home address
4. Subsequent packets are tunneled to the foreign agent
3. Sender is given foreign agent's address
2. Packet is tunneled to the foreign agent

Congestion

- Too many packets in part of the subnet = Performance Degradation = CONGESTION
- The network layer provides congestion control to ensure timely delivery of packets from source to destination

Congestion Control

- Causes of congestion
- Types of congestion control schemes
- Solving congestion

Causes of Congestion

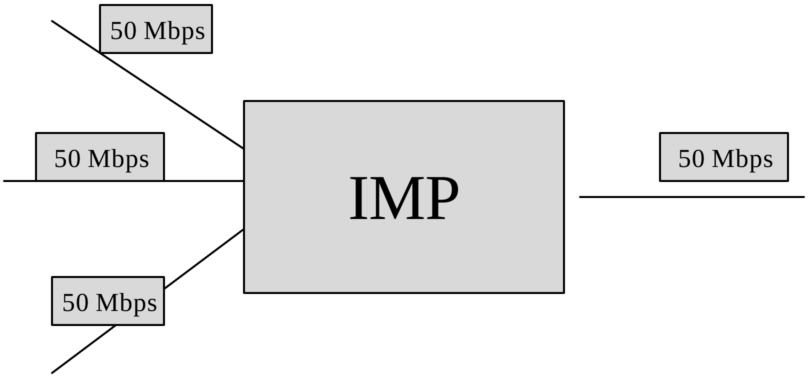
- Exhaustion of buffer space
- Deadlock

- Congestion occurs when the system load is greater than the system resources can handle.

Exhaustion of Buffer Space

- IMPs maintain packet queues (or buffers)
- Buffers fill up if:
  - IMPs are too slow, OR
  - Combined input traffic rate exceeds the outgoing traffic rate
- Insufficient buffer space leads to congestion

Exhaustion of Buffer Space



Deadlock

- The first IMP cannot proceed until the second IMP does something and the second IMP cannot proceed until the first IMP does something
- Both IMPs come to a complete halt and stay that way forever

Types of Deadlock

- Store and Forward Lockup
  - Direct store and forward lockup
  - Indirect store and forward lockup
- Reassembly Lockup

Direct Store and Forward Lockup

- Simplest lockup between two IMPs
- Example
  - Suppose IMP A has five buffers, all of which are queued for output to IMP B
  - Similarly, IMP B has five buffers, all of which are queued for output to IMP A
  - Neither IMP can accept any incoming packets from the other. They are both stuck.

Reassembly Lockup

- In some network layer implementations, the sending IMP must split messages (fragment) into multiple network layer packets
- Receiving IMPs reassemble split up packets into a single packet.
- If the receiving IMP’s buffer fills up with incomplete packets, it cannot reassemble any more packets

Types of Congestion Control

- Preventive
  - The hosts and IMPs attempt to prevent congestion before it can occur
- Reactive
  - The hosts and IMPs respond to congestion after it occurs and then attempt to stop it

Preventive and Reactive Control

- Preventive Techniques:
  - Resource reservation
  - Leaky/Token bucket
  - Isarithmic control
- Reactive Techniques
  - Load shedding
  - Choke Packets

Load Shedding

- When an IMP becomes inundated with packets, it simply drops some.

Intelligent Load Shedding

- Discarding packets does not need to be done randomly
- IMP should take other information into account:
- Possibilities
  - Tail dropping
  - Priority discarding
  - Age based discarding

Tail dropping

- When the buffer fills and a packet segment is dropped, drop all the rest of the segments from that packet, since they will be useless anyway
- Only works with IMPs that segment and reassemble packets

Priority Discarding

- Sources specify the priority of their packets
- When a packet is discarded, the IMP chooses a low priority packet
- Requires hosts to participate by labeling their packets with priority levels

Age Biased Discarding

- When the IMP has to discard a packet it discards either:
  - Oldest (milk)
    - Multimedia
  - Newest (wine)
    - FTP

Choke Packets

- Each IMP monitors the utilization of each of its output lines
- Associated with each line us a variable *u,* which reflects the utilization of that line
- Whenever *u* moves above a given threshold, the output line enters a “warning state”
- Each newly arriving packet checks if its output line is in the warning state
- If so, the IMP sends a choke packet back to the source

Choke Packets

- The data packet is tagged (by setting a bit in its header) so that it will not generate any more choke packets at downstream IMPs
- When the source host receives the choke packet, it is required to reduce its traffic generation rate to the specified destination by *x* percent.
- Since other packets aimed a the same destination are probably already on their way to the congested loation, the source host should ignore choke packets for that destination for a fixed time interval. After that, it resumes its response to choke packets.

Isarithmic Control

- Each host is initially allocated a pool of permits
- Each outgoing packet is required to acquire a permit at the host before it is transmitted
- If no permit is available, the packet waits in the host until a permit becomes available
- Once a permit is aquired, the packet is transmitted and begins its journey to the destination accompanied by the permit. The source node

Isarithmic Control

- At IMPs the travelling packet is not subject to isarithmic control
- At the destination host, the permit is freed and added to the destination host’s permit pool
- This pool controls the transmission of packets from the destination host

These rules guarantee that the number of packets in the network will never exceed the number of permits initially

Isarithmic Control: Problem

- If a host doesn’t send any packets, i.e. it just holds onto its permits, then the network will not be fully utilized
- Solution:
  - Place a limit on the number of permits a host may keep. If the host has more than its allowed number of permits, it must transmit the excess permits to other hosts by piggybacking them onto other data packets or by using special permit-transport packets.

Resource Reservation

- For connection-oriented networks only
- During connection setup:
  - Request resource (e.g., buffers space, connection bandwidth) from the network
  - If the network has enough available resources to support the new connection, the connection will be established
  - Otherwise, the connection will be rejected

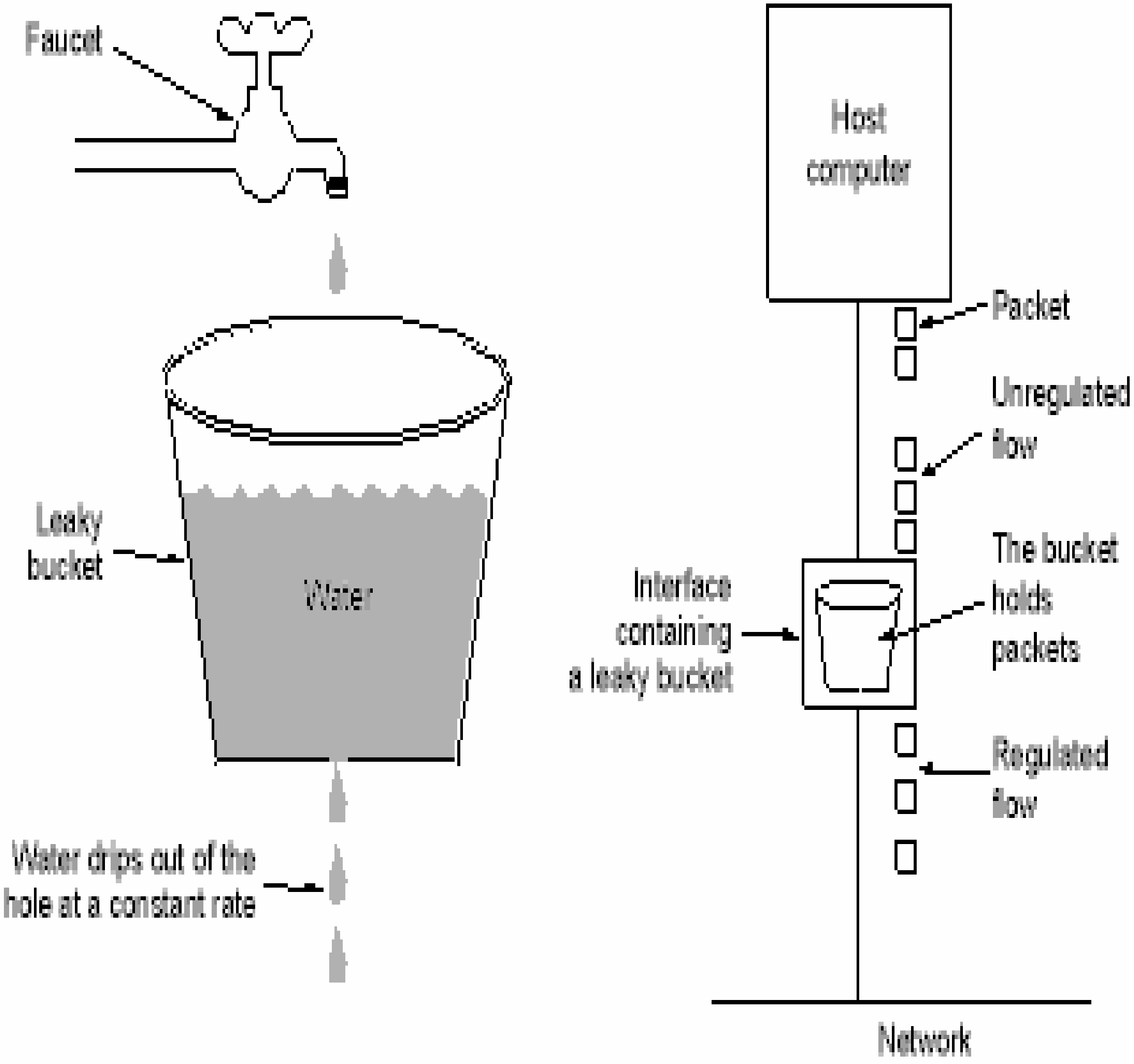
Resource Reservation

- Once a connection is accepted, the host must use only the amount of resources reserved. It may not use more than that.
- Problem:
  - Malicious hosts that attempt to use more resources than they reserve

Leaky Bucket

- Used in conjunction with resource reservation to police the host’s reservation
- At the host-to-network, allow packets into the network at a constant rate
- Packets may be generated in a bursty manner, but after they pass through the leaky bucket, they enter the network evenly spaced

Leaky Bucket: Analogy



Leaky Bucket

- The leaky bucket is a “traffic shaper”: It changes the characteristics of the packet stream
- Traffic shaping makes more manageable and predictable network usage
- Usually the network tells the leaky bucket the rate at which it can send packets when the connection is established

Leaky Bucket: Problem

- In some cases, we may want to allow short bursts of packets to enter the network without throttling them
- For this purpose we use a modified leaky bucket known as: Token Bucket

Token Bucket

- The bucket holds tokens instead of packets
- Tokens are generated and placed into the token bucket at a constant rate
- When a packet arrives at the token bucket, it is transmitted if there is a token available. Otherwise it is discarded (or buffered until a token becomes available).
- The token bucket has a fixed size, so when it becomes full, subsequently generated tokens are discarded.

Token Bucket: Analogy

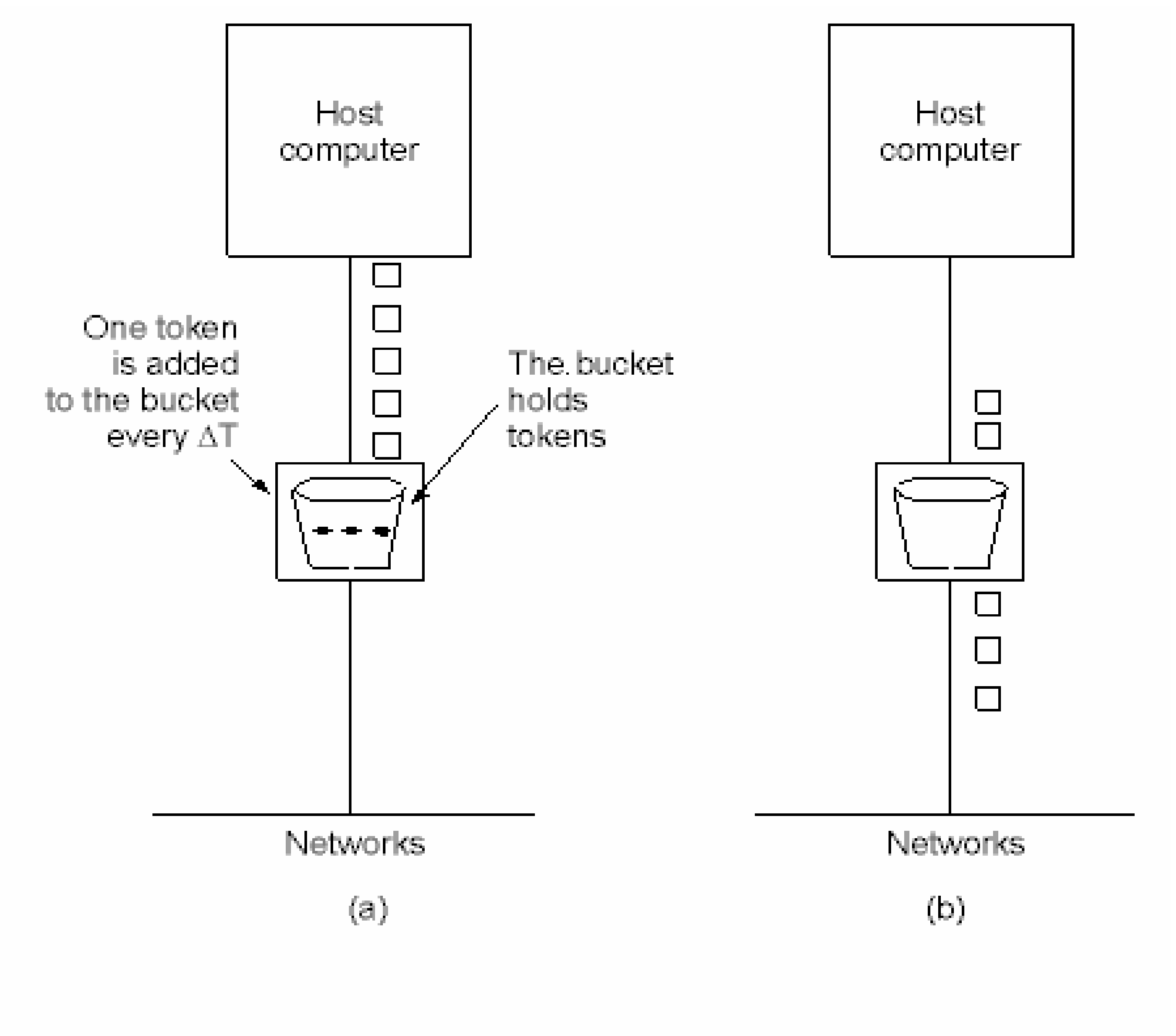


Fig. 5-26. The token bucket algorithm. (a) Before. (b) After.

Internetworking

- internet - Connecting (internetworking) two or more networks together.
  - Various hardware and software combinations
  - Various protocol combinations
    - Appletalk
    - TCP/IP
    - SNA
- Internet - Worldwide internet that connects universities, etc.

Internetworking

- Repeater - physical layer device that amplifies weak signals

- Bridge - store and forward data link layer device

- Router - IMP network layer store and forward device