The Network Layer

Chapter 5

Routing	Control plane
Forwarding	Data plane

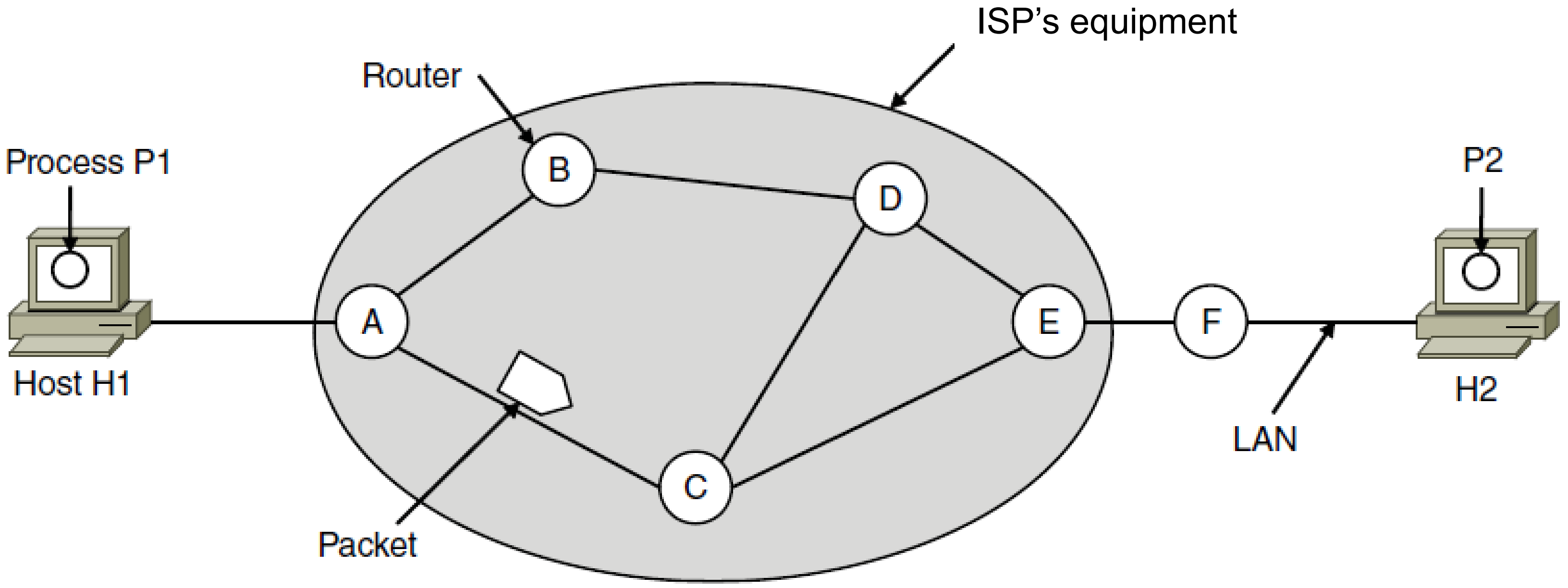
The Network Layer

- The lowest layer that deals with end-to-end transmission.
- Routing over a known subnet G(N, A)
  - Avoid overloading a network, by congestion control
  - Internetworking

Network Layer Design Issues

- Store-and-forward packet switching
- Services provided to transport layer
- Implementation of connectionless service
- Implementation of connection-oriented service
- Comparison of virtual-circuit and datagram networks

Store-and-Forward Packet Switching



The environment of the network layer protocols.

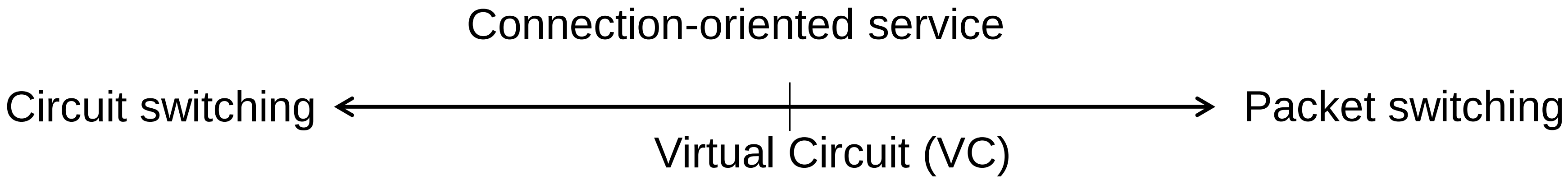
Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

Services Provided to the Transport Layer

1. Services independent of router technology.
2. Transport layer shielded from number, type, topology of routers.
3. Network addresses available to transport layer use uniform numbering plan
   – even across LANs and WANs

Service Provided to the Transport Layer

- Two camps on whether the network should provide connection-oriented service or connectionless service:
  - **Connectionless service** (such as Internet):
    - Let the host do all processing of packet reordering, error control, flow control, etc.
    - Each packet carries the full destination address.
    - Network services simply with primitive SEND PACKET and RECEIVE PACKET and little else.
  - **Connection oriented service** (such as ATM, MPLS, VLAN)
    - Let the subnet provide a reliable, connection oriented service.
    - Quality of service (QoS) can be guaranteed, for real time traffic such as voice and video



Connection-oriented service

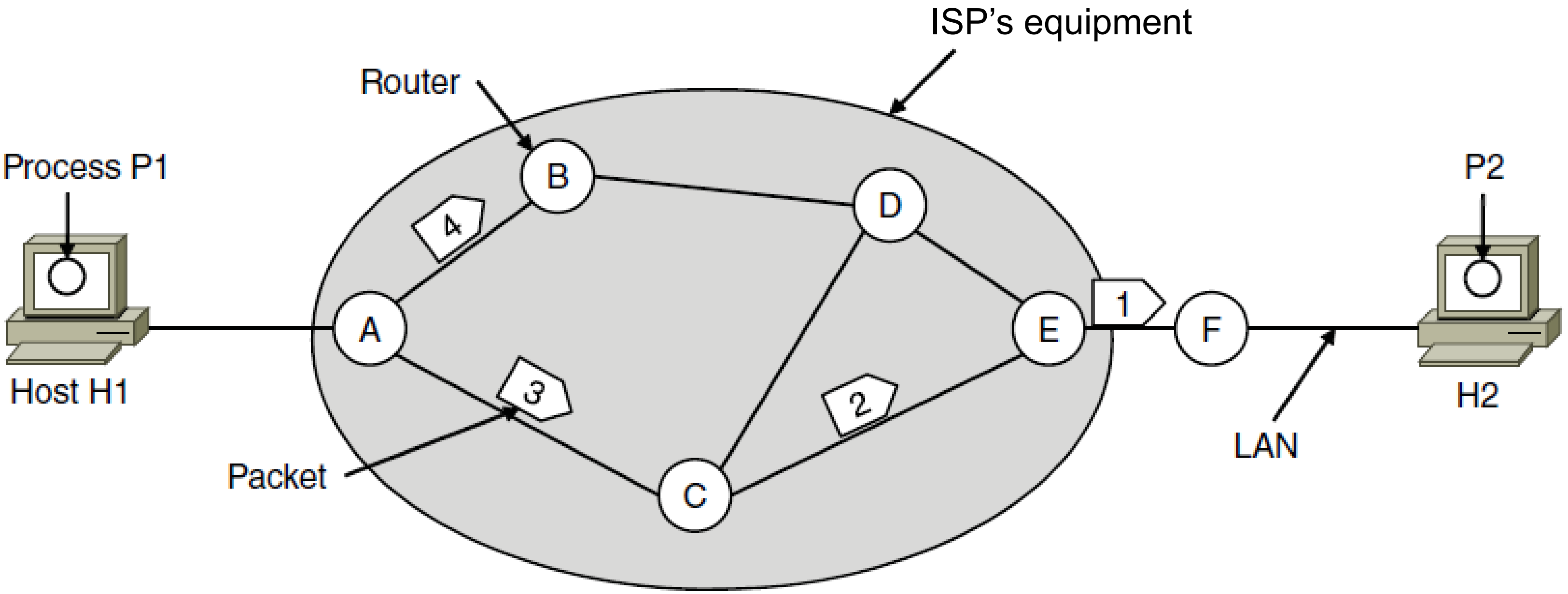
Circuit switching

Packet switching

Virtual Circuit (VC)

- **Datagram** subnet
  - If connection service is offered, packets are injected into the subnet individually and routed independently.
- **Virtual circuit (VC)**subnet
  - If connection oriented service is used, a path from the source router to the destination router must be established before any packet can be sent.

Implementation of Connectionless Service



A's table (initially)

A	–
B	B
C	C
D	B
E	C
F	C

Dest. Line

A's table (later)

A	–
B	B
C	C
D	B
E	B
F	D 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

Routing within a datagram network

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

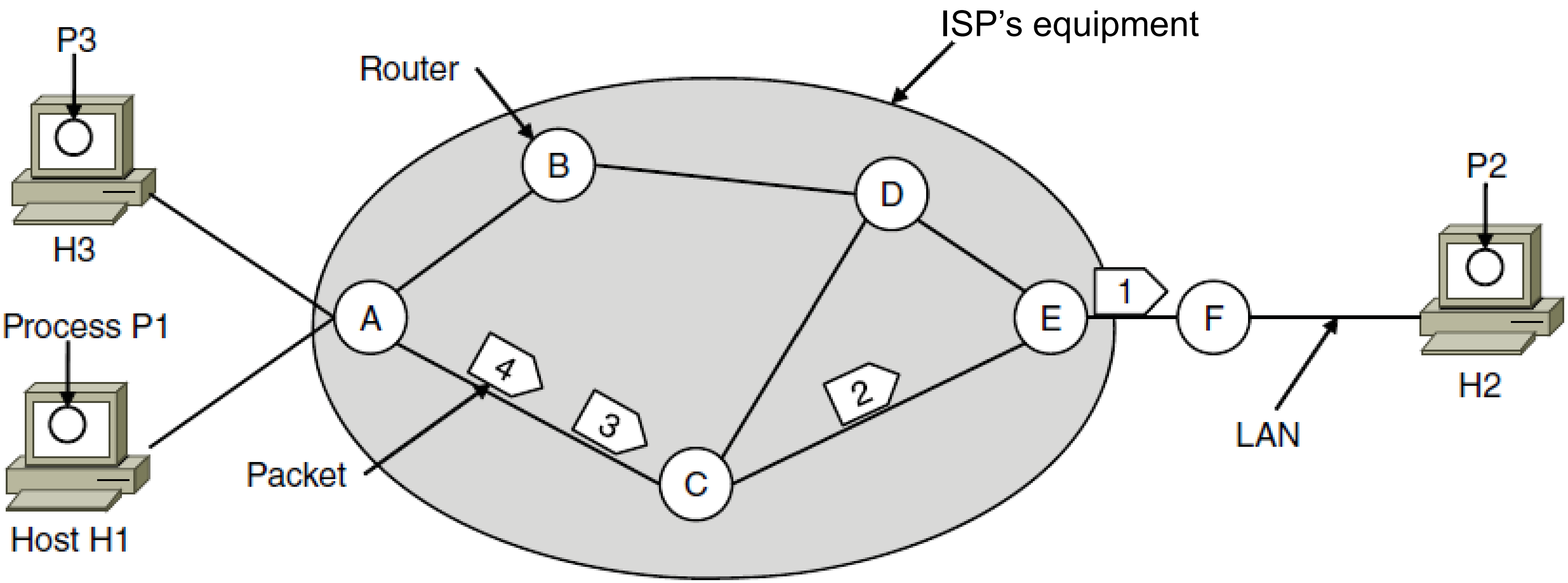
Routing algorithms

a) Every router has an internal table telling where to send packets for each possible destination.

- Each table entry is a pair consisting of a destination and the outgoing line to use for that destination.

b) The algorithm that manages the tables and makes the routing decision is called the **routing algorithm**.

Implementation of Connection-Oriented Service



A's table

H1	1	C	1
H3	1	C	2

In Out

C's Table

A	1	E	1
A	2	E	2

E's Table

C	1	F	1
C	2	F	2

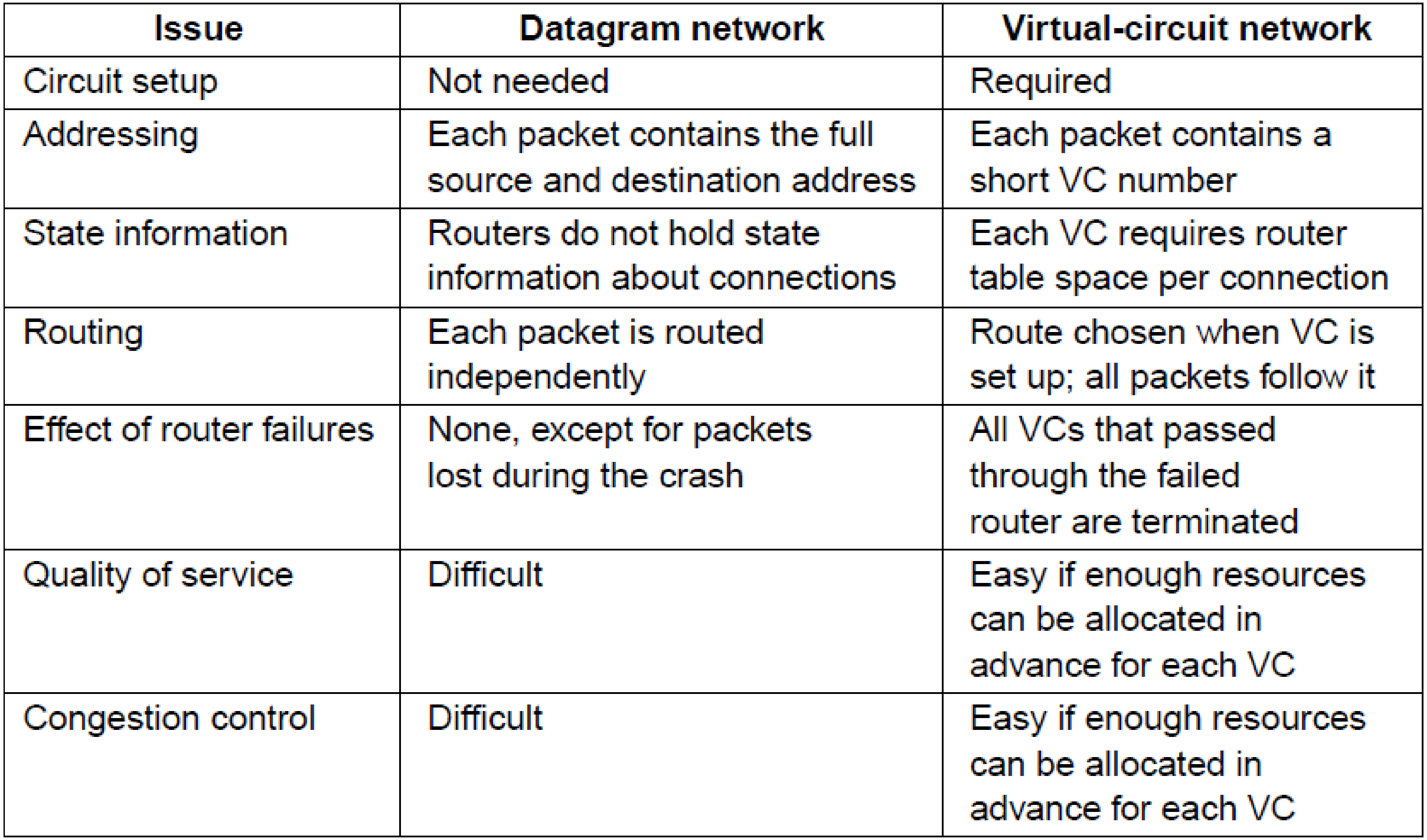
Routing within a virtual-circuit network

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

(Continue)

- A virtual-circuit subnet is needed for connection-oriented service.
- 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.
- With connection-oriented service, each packet carries an identifier telling which virtual circuit it belongs to.
- …..Label Switching (MPLS vs SD-WAN ?)

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

Comparison of datagram and virtual-circuit networks

Routing Algorithms (1)

- Optimality principle
- Shortest path algorithm
- Flooding
- Distance vector routing
- Link state routing
- Routing in ad hoc networks

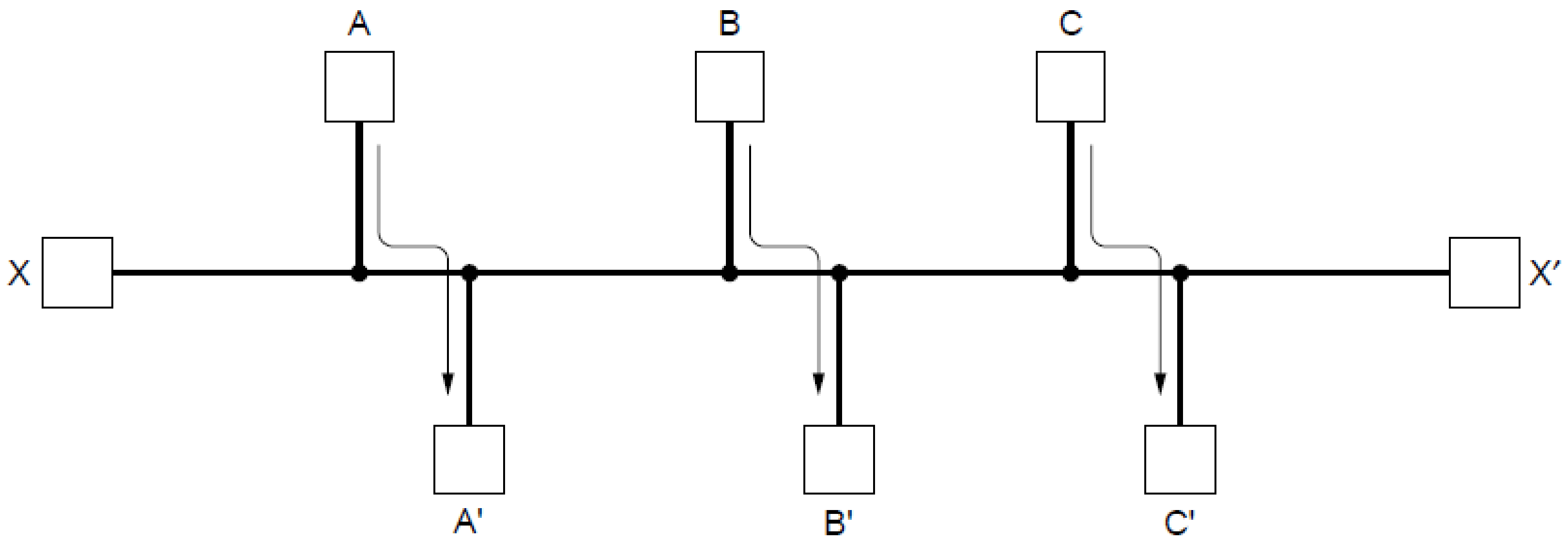
Routing Algorithms (2)

- Broadcast routing
- Multicast routing
- Anycast routing
- Routing for mobile hosts
- Routing in ad hoc networks

Cont.

- The routing algorithm is part of the network layer software responsible for deciding which output line an incoming packet should be transmitted on.
  - If the subnet uses datagram internally, this decision must be made anew for every arriving data packet.
  - If the subnet uses virtual circuits internally, routing decisions are made only when a new virtual circuit is being set up (called **session routing**).
- One router can be viewed as having two processes inside it.
  - One handles each packet as it arrives, looking up the outgoing line to use for it in the routing table.---**Forwarding**
  - The other is responsible for filling and updating the routing tables.---The **routing** algorithm.
- Certain properties desired in the routing algorithm:
  - Correctness, simplicity, robustness, stability, fairness, and efficiency.

Fairness vs. Efficiency



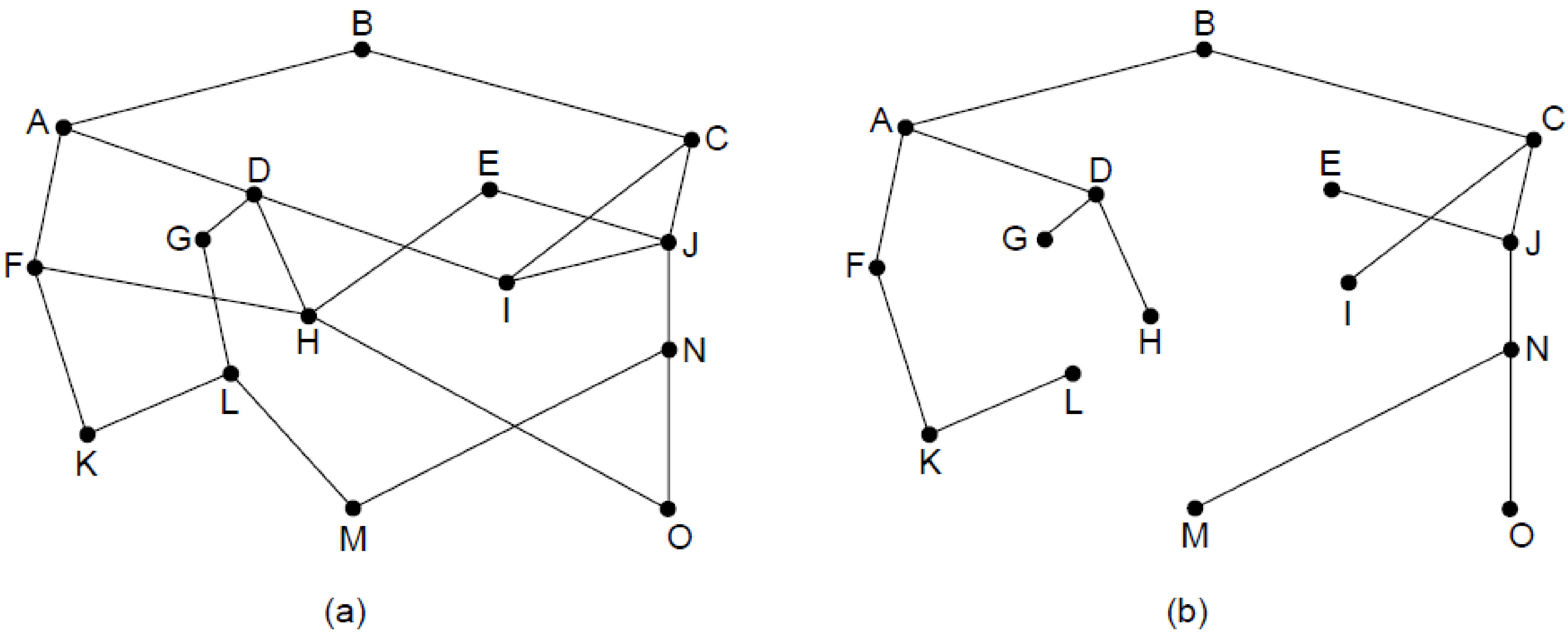
Network with a conflict between fairness and efficiency.

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

Cont.

- Two classes of routing algorithms:
  - **Nonadaptive** algorithms → **Static routing**
  - **Adaptive algorithms**, to reflect changes in the topology, and usually the traffic as well. → **Dynamic routing**
    - Where they get their information (e.g., locally, from adjacent routers, or from all routers).
    - When they changes the routes (e.g., every T sec, when the load changes or when the topology changes).
    - What metric is used for optimization (e.g., distance, number of hops, or estimated transit time).

The Optimality Principle

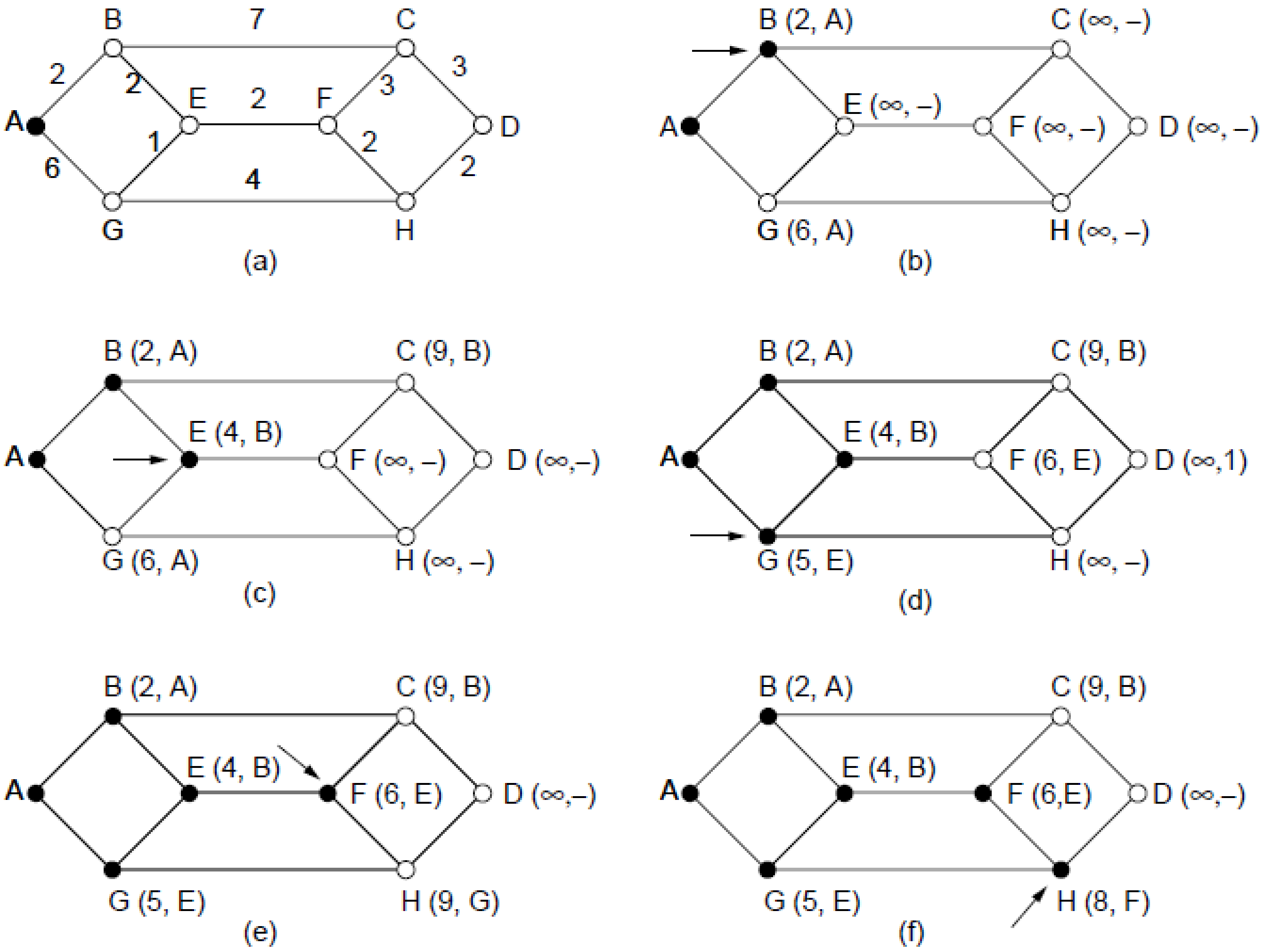


(a) A network. (b) A sink tree for router *B.*

Cont.

- The **optimality principle**:
  - 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. (Proof by contradiction).
- The optimal routes from all sources to a given destination form a tree rooted at the destination. Such a tree is called a **sink tree**.
  - A sink tree is not necessarily unique.
  - The goal of all routing algorithms is to discover and use the sink trees for all routers.
  - Since a sink tree is a tree, it does not contain any loops, so each packet will be delivered within a finite and number of hops.

Shortest Path Algorithm (1)



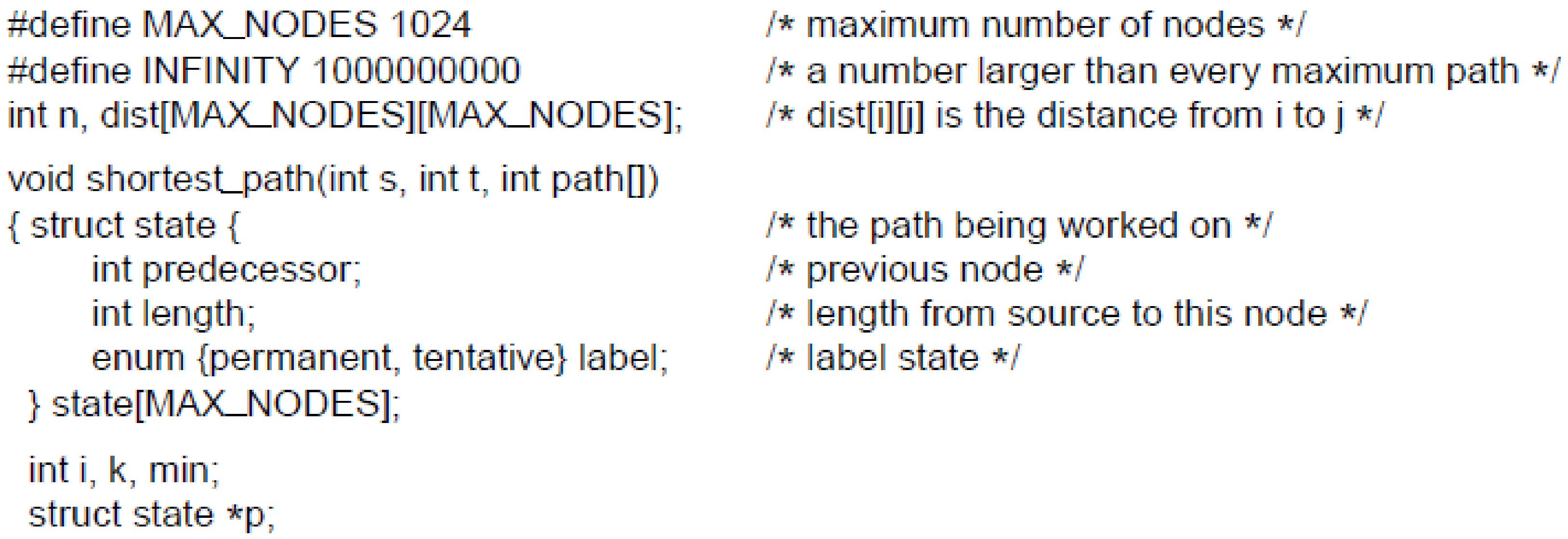
The first five steps used in computing the shortest path from *A to D.* The arrows indicate the working node

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

Cont.

- A subnet is represented as a *graph* (*node*, *edge*), i.e. *G*(*N*,*E*).
  - Each edge is associated with a cost, which is a function of the distance, bandwidth, average traffic, communication cost, mean queue length, measured delay and other factors.
  - Several algorithms can be used for computing the shortest path between two nodes of a known graph.
- The Dijkstra algorithm:
  - Each node is labeled 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 changes
      - 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.

Shortest Path Algorithm (2)



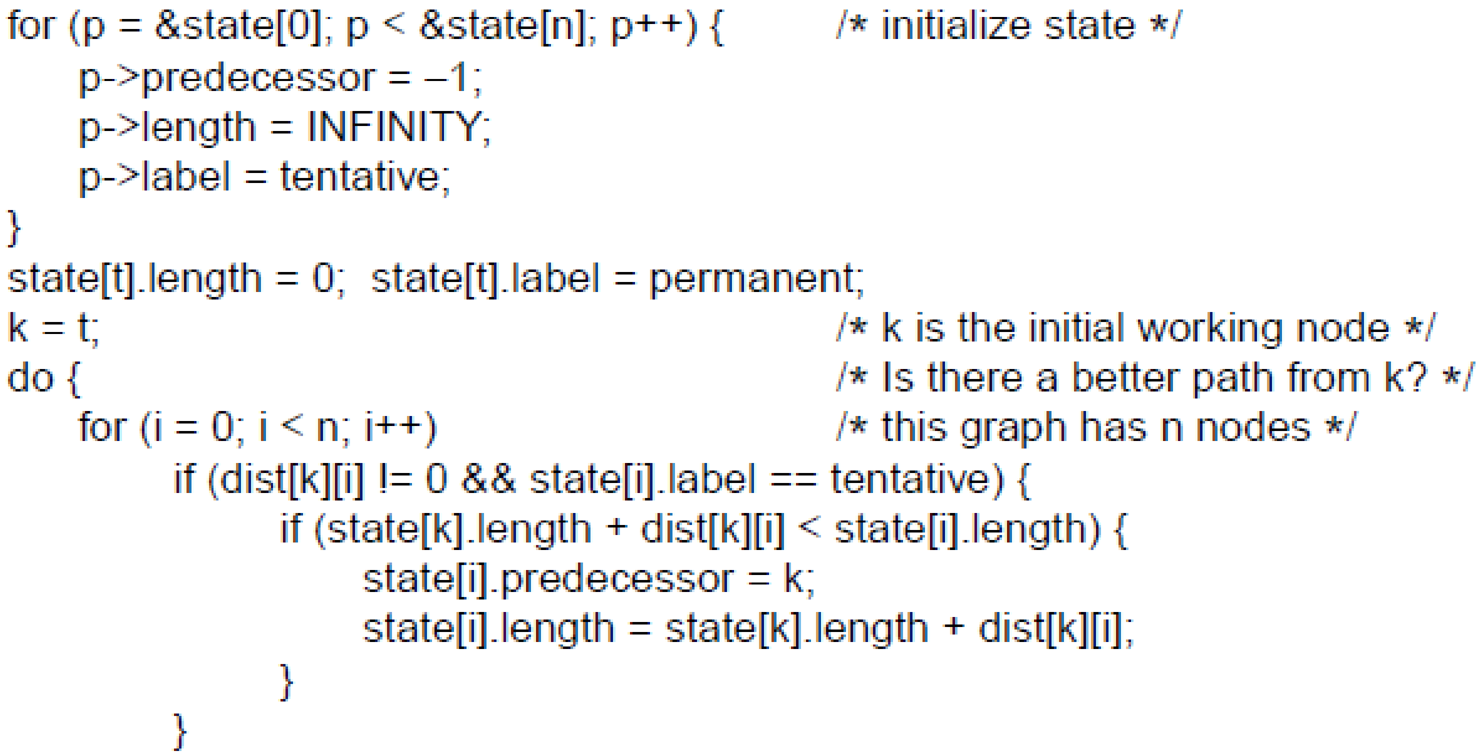
```
#define MAX_NODES 1024                    /* maximum number of nodes */
#define INFINITY 1000000000               /* a number larger than every maximum path */
int n, dist[MAX_NODES][MAX_NODES];        /* dist[i][j] is the distance from i to j */

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

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

. . .

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

Shortest Path Algorithm (3)

. . .



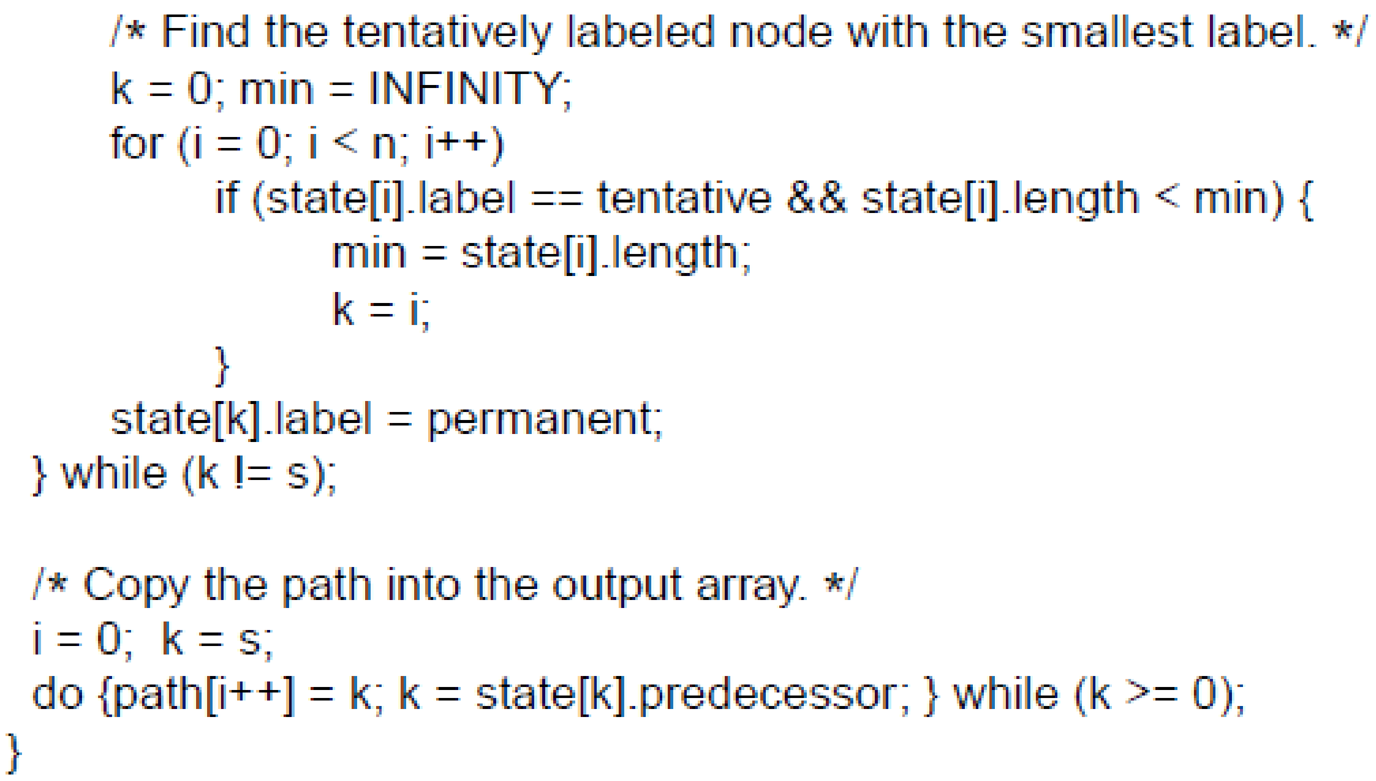
```
for (p = &state[0]; p < &state[n]; p++) {        /* initialize state */
     p->predecessor = –1;
     p->length = INFINITY;
     p->label = tentative;
}
state[t].length = 0;  state[t].label = permanent;
k = t;                                            /* k is the initial working node */
do {                                              /* Is there a better path from k? */
     for (i = 0; i < n; i++)                      /* this graph has n nodes */
          if (dist[k][i] != 0 && state[i].label == tentative) {
               if (state[k].length + dist[k][i] < state[i].length) {
                    state[i].predecessor = k;
                    state[i].length = state[k].length + dist[k][i];
               }
          }
```

. . .

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

Shortest Path Algorithm (4)

. . .



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

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

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

Flooding

- By flooding, every incoming packet is sent out on every outgoing line except the one it arrived on. To avoid generating vast number of duplicate packets, two approaches can be exploited:
  - Using a **hop counter** contained in the header of each packet. The counter is decremented at each hop, with the packet being discarded when the counter reaches zero.
    - How long is the path ?
  - Keeping track of which packets have been flooded, to avoid sending them out a second time.
    - It necessitates the source router putting a **sequence number** in each packet it receives from its hosts.
    - Each router needs a list per source router telling which sequence numbers originating at that source have already been seen.
    - Using a counter for each list can prevent the list from growing without bound.

Flooding (2)

- Flooding is not practical, but has some uses:
  - In military applications, for its robustness.
  - In distributed database applications, for database update concurrently.
  - In wireless networks
  - A metric, against which other routing algorithms can be compared.
  - In the setup of flooding, the routers only need to know their neighbors.

Dynamic Routing Algorithms

- Two most popular dynamic routing algorithms:
  1. **Distance vector routing**, or called **Bellman -Ford** routing algorithm or **Ford-Fulkerson** algorithm.
     - Bellman, R. E., “Dynamic Programming”, *Princeton University Press*, Princeton, N. J., 1957.
     - Ford, L. R. Jr., and Fulkerson, D. R., “Flows in Networks”, *Princeton University Press*, Princeton, N. J., 1962
  2. **Link state routing**, based on the **Dijkstra’s** algorithm.

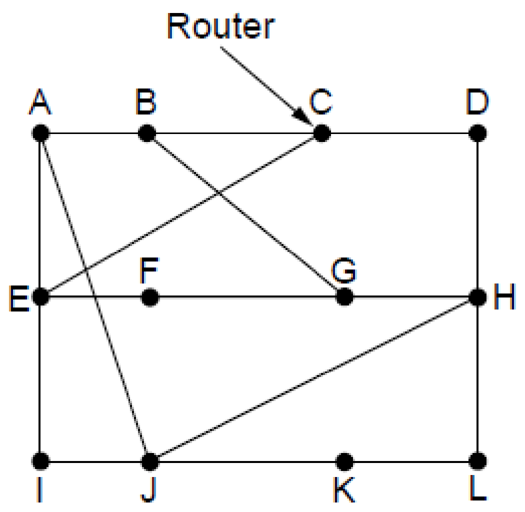
Distance Vector Routing

- The distance vector routing algorithm was the original ARPANET routing algorithm and was also used in the Internet under the name RIP (routing information protocol).
  - Each router maintains a routing table with one entry for each router in the subnet.
    - The entry contains two parts: the preferred outgoing line to use for that destination and an estimate of the time or distance to that destination.
  - The router is assumed to know the “distance” to each of its neighbors.
  - …….

Cont.

- Once every *T* msec each router sends to each neighbor a list of its estimated delays to each destination. It also receives a similar list from each neighbor.
  - Suppose that *Xi* is *X*'s estimate of how long it takes to get to router *i*.
  - If the router knows that the delay to *X* is *m* msec, it also knows that it can reach router *i* via *X* in *Xi*+*m* msec.
  - By performing this calculation for each neighbor, a router updates the best estimate and the corresponding line in its routing table.

Distance Vector Routing



(a)

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

Vectors received from J's four neighbors

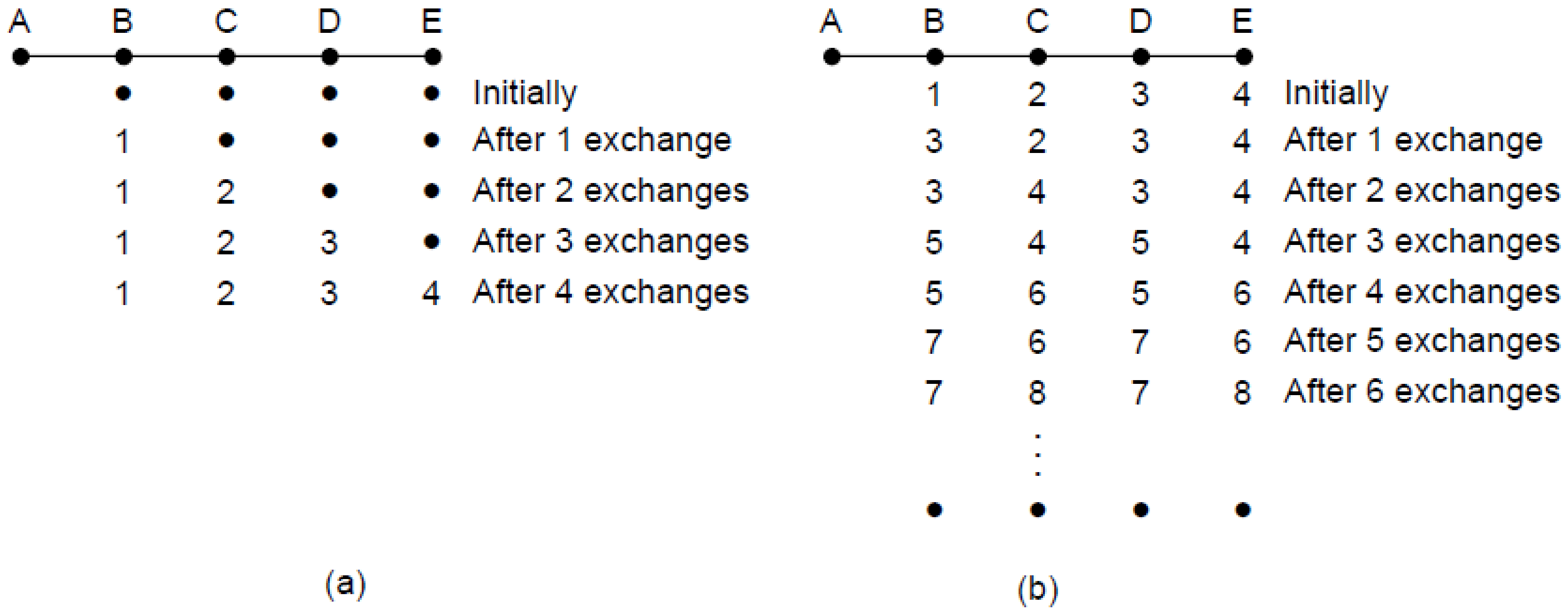
(b)

(a) A network.

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

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

The Count-to-Infinity Problem



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

(a)

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

(b)

The count-to-infinity problem

Cont.

- Good News propagating in a linear subnet (a)
  - Suppose *A* is down initially and all the other routers have all recorded the delay to *A* as infinity.
  - When *A* comes up, the other routers learn about it via the vector exchanges.
  - Results: The good news is spreading at the rate of one hop per exchange.
- Bad News (b)
  - All lines and routers are initially up.
  - Suddenly *A* goes down, or alternatively, the line between *A* and *B* is cut.
  - .. It raises the **count-to-infinity** problems.

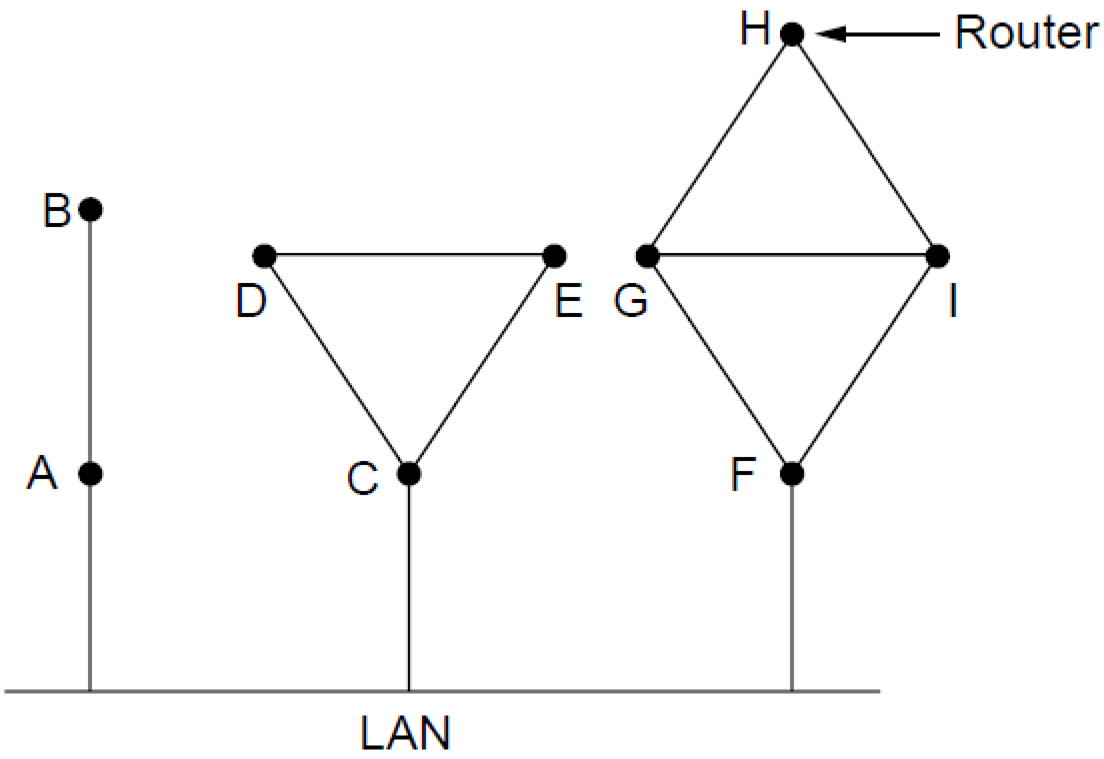
Link State Routing

- Dynamic routing algorithm
- Link state routing replaced distance vector routing in the ARPANET in 1979. For two primary problems:
    - The delay metric was queue length. However, line bandwidth was not taken into account when choosing routes. (Initially, all lines were 56 kbps!).
    - The algorithm often took too long to converge, due to the count-to-infinity problem.

Link State Routing

1. Discover neighbors, learn network addresses.
2. Set distance/cost metric to each neighbor.
3. Construct packet telling all learned.
4. Send packet to, receive packets from other routers.
5. Compute shortest path to every other router.

Learning about the Neighbors (1)



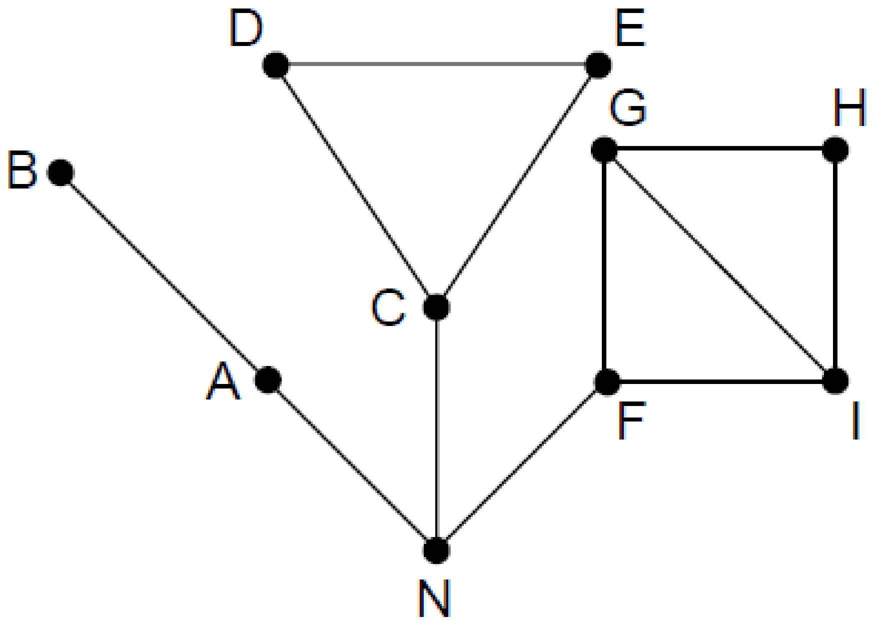
Nine routers and a broadcast LAN.

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

(Continue)

1. Send a special HELLO packet on each point-to-point line to learn who its neighbors are.
2. When two or more routers are connected by a broadcast link (LAN):
   - One solution: Model the LAN as many point-to-point links, which increases the size of the topology and hence leads to waseful message.
   - A better way to model the LAN is to consider it as a node itself.
     - It is represented as an artificial node *N* in Fig. 5-11(b).
     - One **designated router** on the LAN is selected to play the role of *N* in the routing protocol.

Learning about the Neighbors (2)

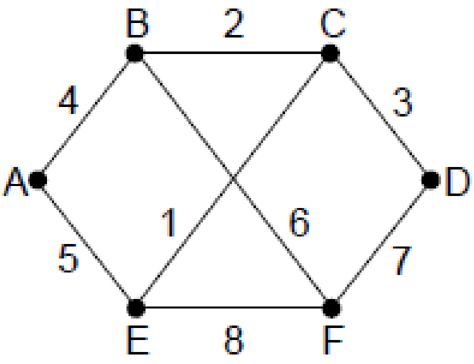


A graph model of previous slide.

Line Cost

- The link state routing algorithm requires each link to have a distance or cost metric for finding shortest paths.
  - A common choice is to make the cost inversely proportional to the bandwidth of the link.
  - If the cost metric is the delay of the links, the most direct way is to send a special **ECHO** packet over the line that the other side is required to send back immediately.

Building Link State Packets



(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

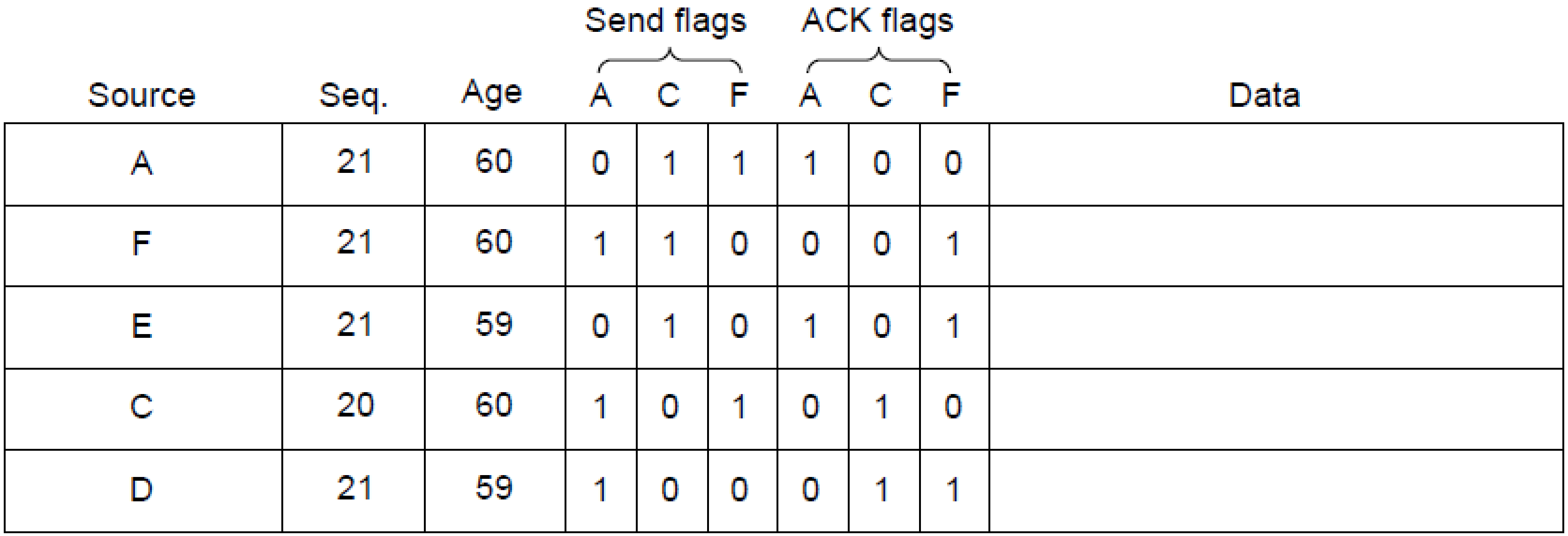
(b)

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

Cont.

- The packet starts with the *identity of the sender*, followed by *a sequence number and age* and *a list of neighbors*.
  - For each neighbor, the delay to that neighbor is given.
- The hard part is determining <u>when to build them</u>:
  - periodically, or
  - when some significant event occurs.

Distributing the Link State Packets



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

The packet buffer for router *B* in previous slide

Cont.

- Issue: how to distribute the link state packets *reliable*, to avoid inconsistencies, loops, unreachable machines, and other problems.
- The fundamental idea is to use *flooding* to distribute the link state packets. 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 packet 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 it has a sequence number lower than the highest one seen so far, it is rejected as being obsolete.

Cont.

- Problems with the algorithm:
  - If the sequence numbers wrap around,...
    - The solution is to use a 32-bit sequence number.
  - If a router ever crashes,
  - If a sequence number is ever corrupted,...
- 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.
  - The age field of a packet is also decremented by each router during the initial flooding process.

Cont.

- Refinements to this algorithm:
  - When a link state packet comes in to a router for flooding, it is not queued for transmission immediately but put in a holding area to wait for a short while.
    - If another link state packet from the same source comes in before the packet is transmitted, their sequence number 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 router-router lines, all link state packets are acknowledged.
  - When a line goes idle, the holding area is scanned in round-robin order to select a packet or acknowledgement to send.

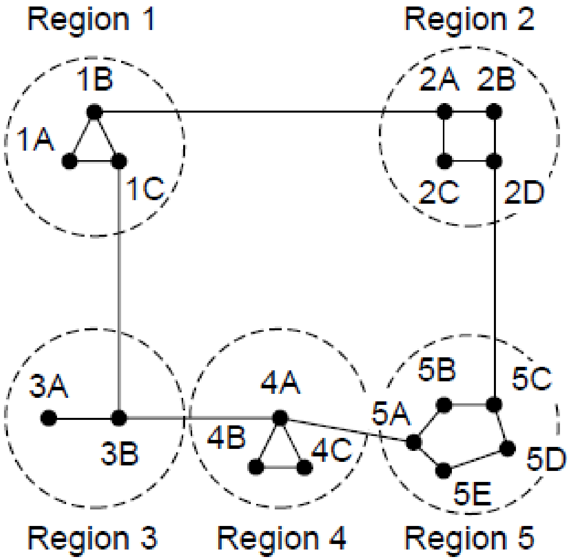
Compute the New Routes

a) Now G(N,A) is available. Dijkstra's algorithm can be run locally to construct the shortest paths to all possible destinations.

- – Link state routing does not suffer slow convergence problems, but requires more memory and computation.
- – For a subnet with *n* routers, each of which has k neighbors, the memory required to store the input data is proportional to *kn*. For large subnets, this can be a problem.
  - Problems with hardware or software can wreak havoc with this algorithm....

b) Practical examples of link state routing: OSPF and IS-IS

Hierarchical Routing



(a)

Full table for 1A

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

(b)

Hierarchical table for 1A

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

(c)

Hierarchical routing.

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

Cont.

- As networks grow in size, the router routing tables grow proportionally.
  - Issues then arise, on router memory, CPU time to scan them, and bandwidth to send status reports
- In hierarchical routing, the routers are divided into **regions**, with each router knowing all the details about how to route packets to destinations within its own region, but knowing nothing about the internal structure of other regions.
  - For huge networks, a two-level hierarchy may be insufficient; it may be necessary to group the regions into clusters, the clusters into zones, the zones into groups, and so on.
  - The penalty to be paid is in the form of increased path length.

Kamoun&Kleinrock’s result: The optimal number of levels for an N router network is *ln N*, requiring a total of *e ln N* entries per router.

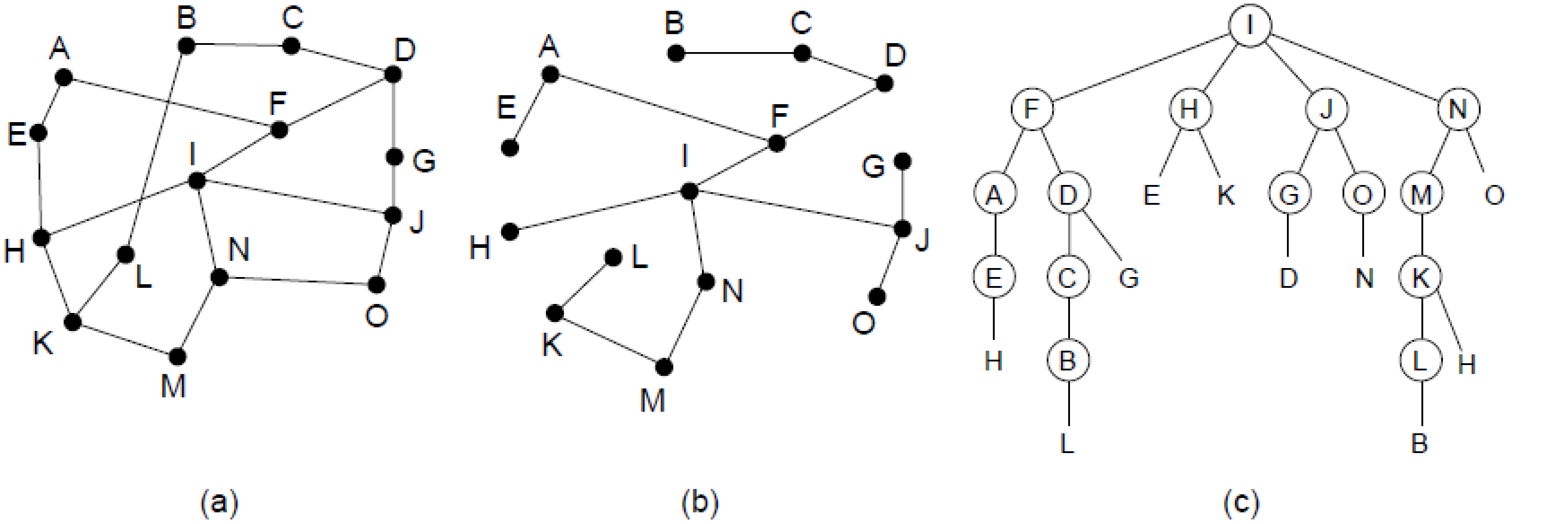
Broadcast Routing

- Sending a packet to all destinations simultaneously is called **broadcasting**.
  - The source simply sends distinct packet to each destination, necessitating no special features from the subnet.
    - Issues: wasteful of bandwidth and slow, the requirement of a complete list of all destinations.
  - Flooding
    - Issues: It generates too many packets and consumes too much bandwidth. (It is good for broadcasting but requires sequence number!)
  - Multidestination routing, by which each packet either contains either a list of destinations or a bit map indicating the desired destinations.
    - 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.
    - After a sufficient number of hops, each packet will carry only one destination and can be treated as a normal packet.

Cont

- Make use of the sink tree for the router initiating the broadcast, or any other convenient spanning tree.
  - A **spanning tree** is a subset of the subnet that includes all the routers but contains no loop.
    - 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.
    - Each router must have knowledge of some spanning tree for it to be applicable, e.g. with the link state routing.
- **Reverse path forwarding**, attempting to approximate the behavior of the previous one even when the routers do not know anything at all about spanning trees.
  - When a broadcast packet arrives at a router, the router check to see if the packet arrived on the line that is normally used for sending packets to the source of the broadcast.
    - If so, the router forwards copies of it onto all lines except the one it arrives on.
    - If not, the packet is discarded as a likely duplicate.

Broadcast Routing



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

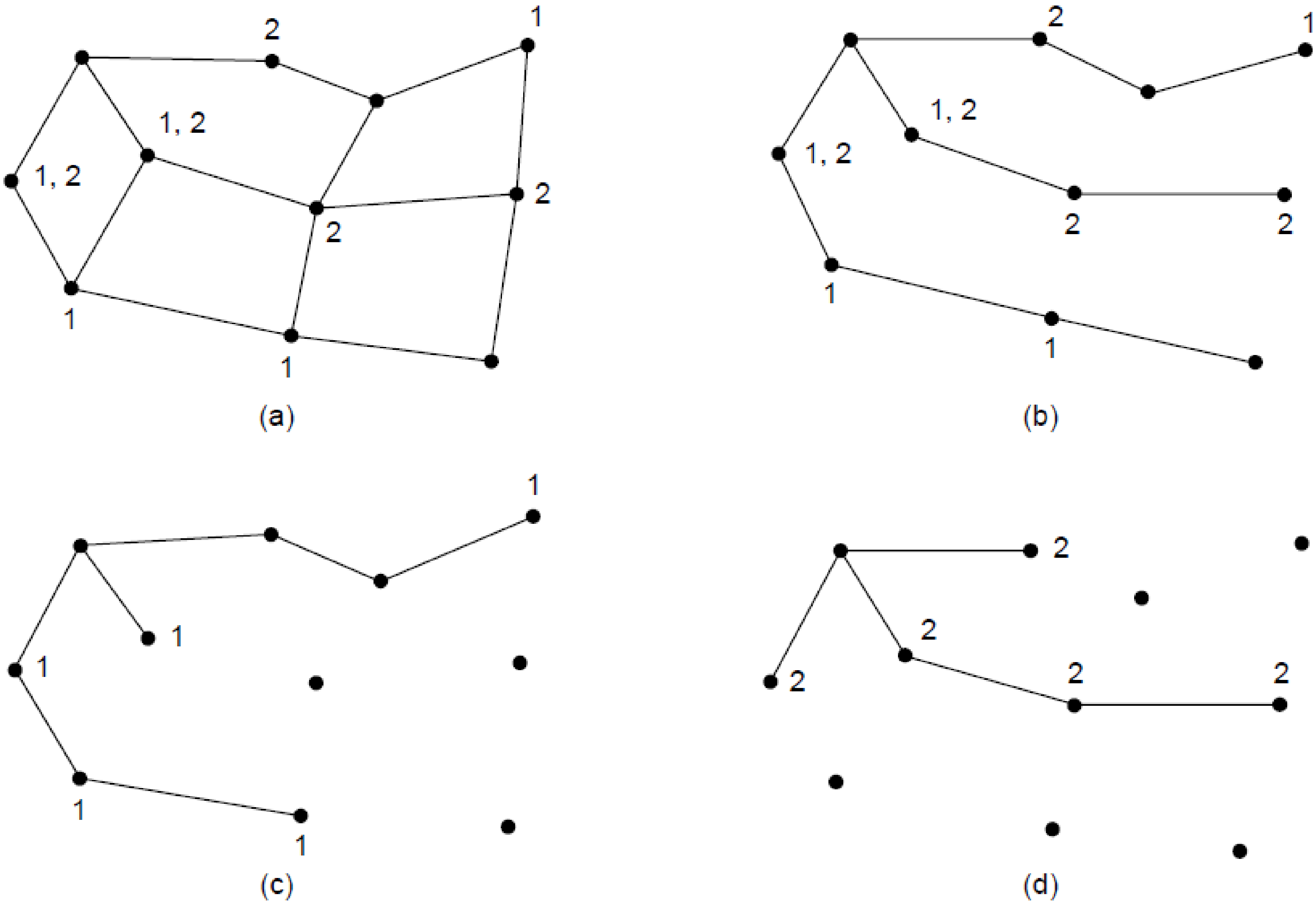
Cont.

- The principal advantage of reverse path forwarding is that it is both reasonably efficient and easy to implement.
  - It does not require routers to know about spanning trees.
  - It does not have the overhead of a destination list or bit map in each broadcast packet as does multidestination addressing
  - It does not require any special mechanism to stop the process, as flooding does.

Multicast Routing

- *Multicasting*: To send messages to well-defined groups that are numerically large in size but small compared to the network as a whole.
  - Multicasting requires group management.
    - Some way is needed to create and destroy groups, and for some processes to join and leave groups.
  - Assume that each group is identified by a multicast address and that routers know the groups to which they belong.
- Multicasting routing, based on the broadcast schemes
  - Send packets along a spanning tree covering all other routers in the subnet.
  - The best spanning tree to use depends on whether the group is dense or sparse.

Multicast Routing (1)



(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.

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

Cont.

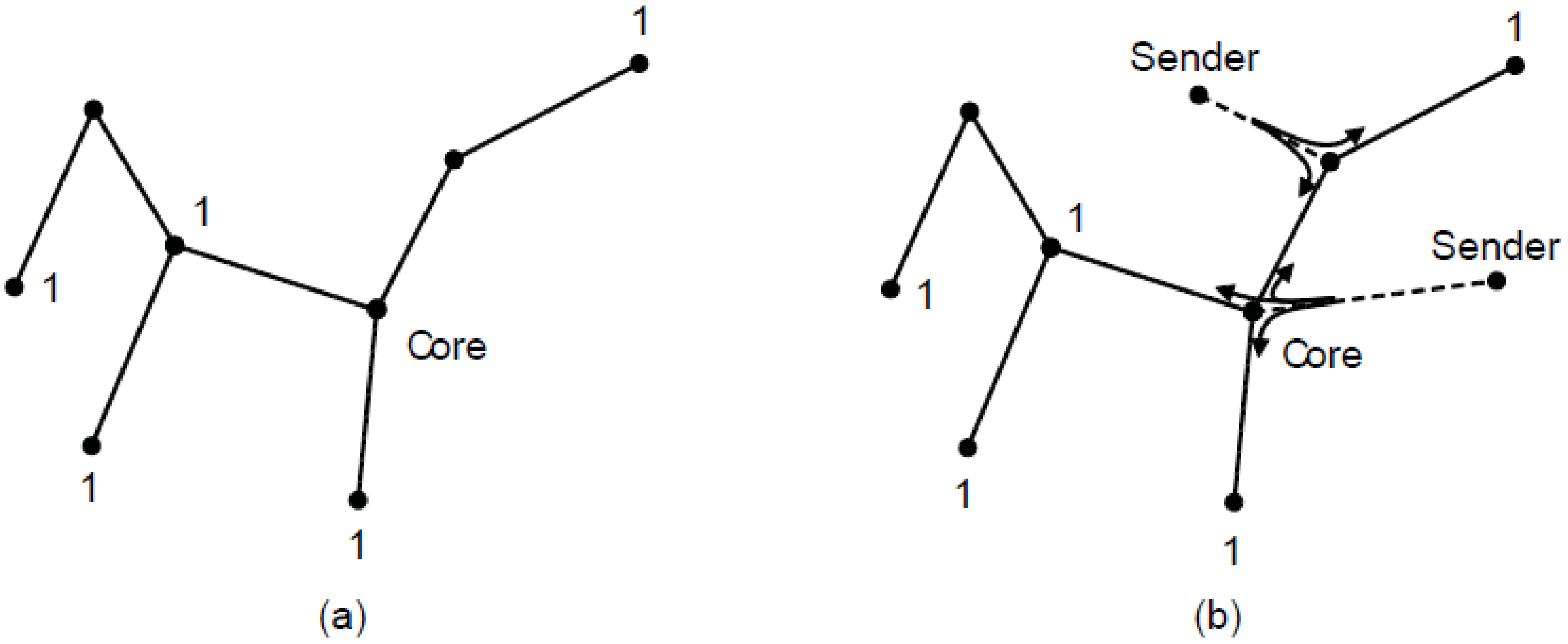
- To prune the spanning tree:
  - If link state routing is used and each router is aware of the complete subnet topology, including which hosts belong to which groups. Each router can then construct its own pruned spanning tree for each sender to the group in question.
    - Example, MOSPF (Multicast OSPF, 1994)
  - If distance vector routing is used, the basic algorithm is reverse path forwarding.
    - Whenever a router with no hosts interested in a particular group and no connection to other routers receives a multicast message for that group, it responds with a PRUNE message, telling the sender not to send it any more multicast for that group.
    - ...
    - The spanning tree is recursively pruned. (DVMRP, Distance Vector Multicast Routing protocol, works this way, 1988)

Cont.

- One disadvantage of this algorithm is that it scales poorly to large networks.
  - Suppose that each group has an average of *m* members. Then at each router, *m* pruned trees must be stored per group.
- **Core-base tree**: An alternative design.
  - A single spanning tree per group is computed, with the root near (the core) the middle of the group.
    - To send a multicast message, a host sends it to the core, which then does the multicast along the spanning tree.
    - Reduce the storage costs from *m* trees to one tree per group.

a) Shared tree approaches like core-base trees are used for multicasting in sparse groups in the Internet (such as PIM, Protocol Independent Multicast, 2006)

Multicast Routing (2)



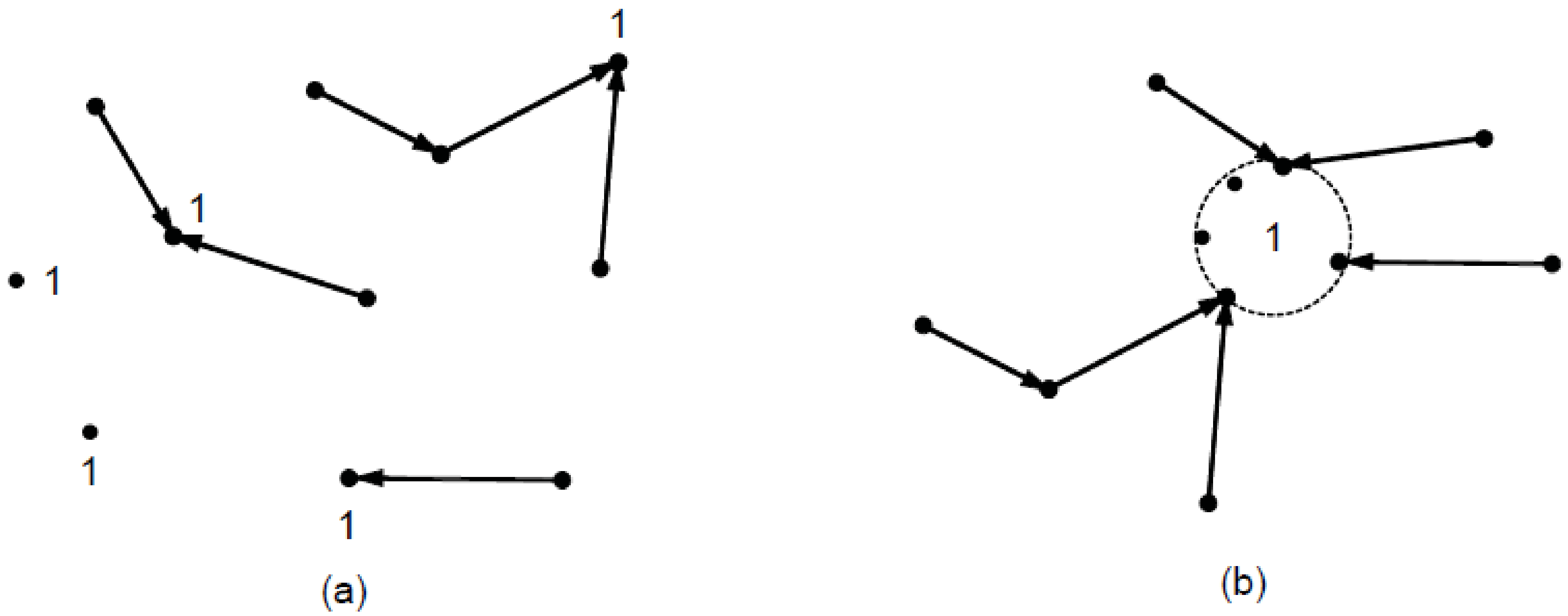
(a) Core-based tree for group 1.

(b) Sending to group 1.

Anycast

- A packet is delivered to the nearest member of a group.
  - Schemes that find these paths are called **anycast routing**.
  - Anycast is used in the Internet as part of DNS.

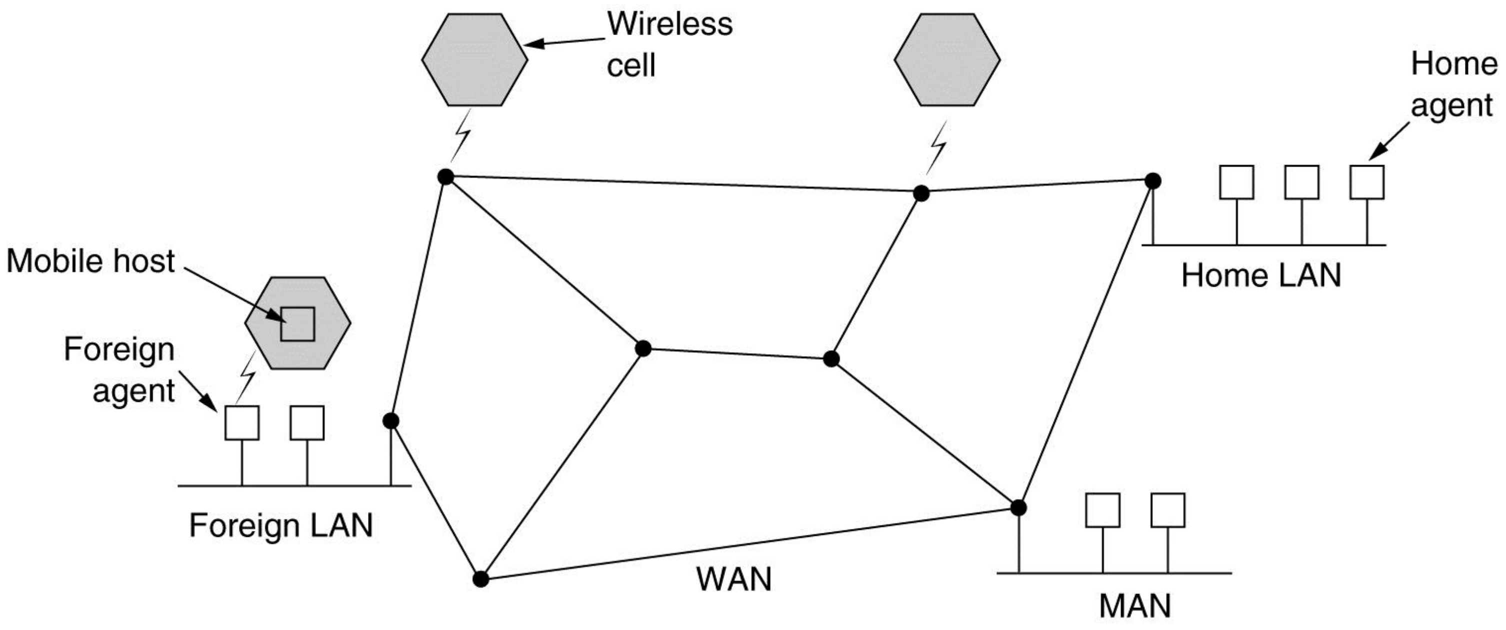
Anycast Routing



(a) Anycast routes to group 1.

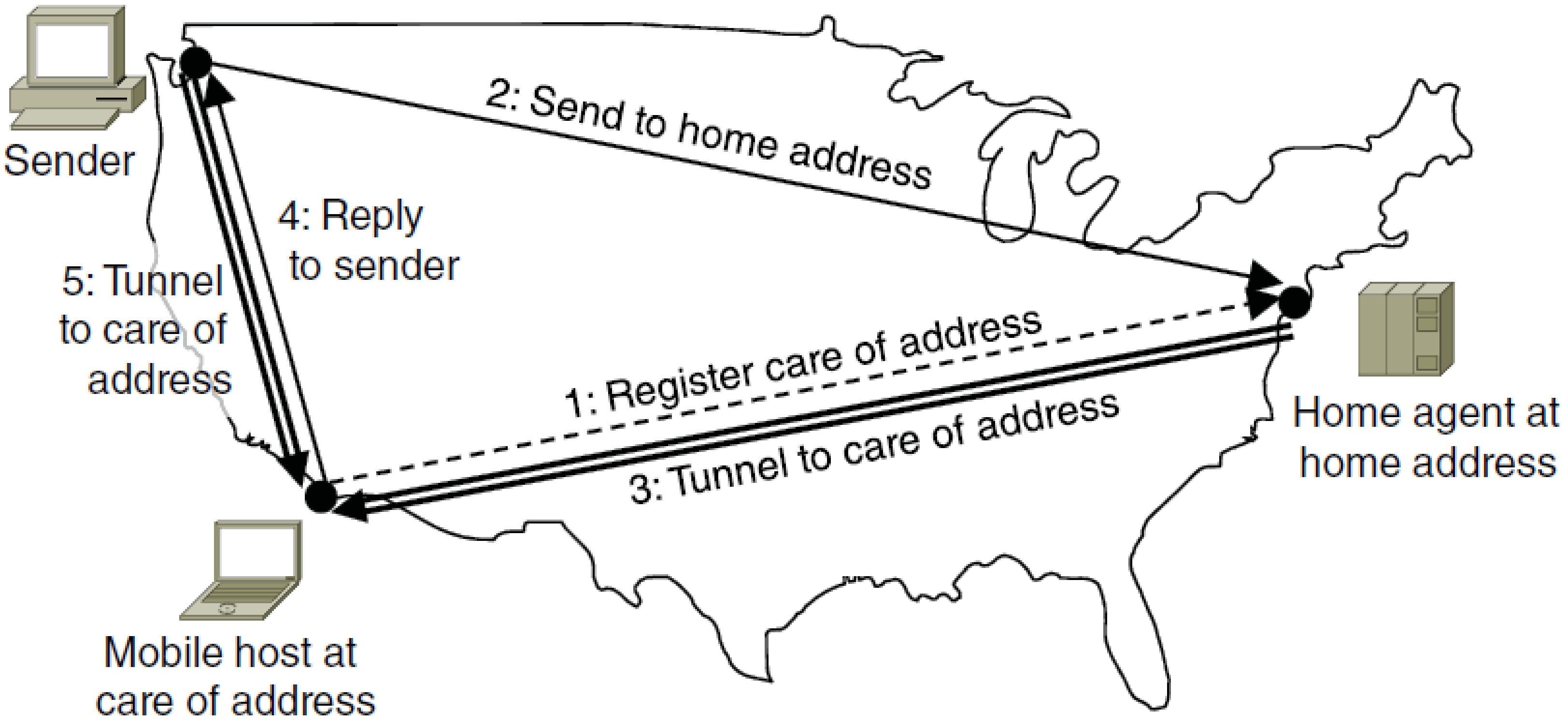
(b) Topology seen by the routing protocol.

Routing for Mobile Hosts (in 4th ed)



A WAN to which LANs, MANs, and wireless cells are attached.

Routing for Mobile Hosts


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

Packet routing for mobile hosts

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

Cont.

- The Mobile Hosts(MH) introduce a new complication: to route a packet to a mobile host, the network first has to find it.
- MHs are assumed to have a permanent **home location** that never changes.
  - MHs have *a permanent* ***home address*** that can be used to determine their home locations.
  - ***Home agent***, the host records where a MH is and acts on behalf of the MH. ← HLR, Home Location Register in cellular networks.

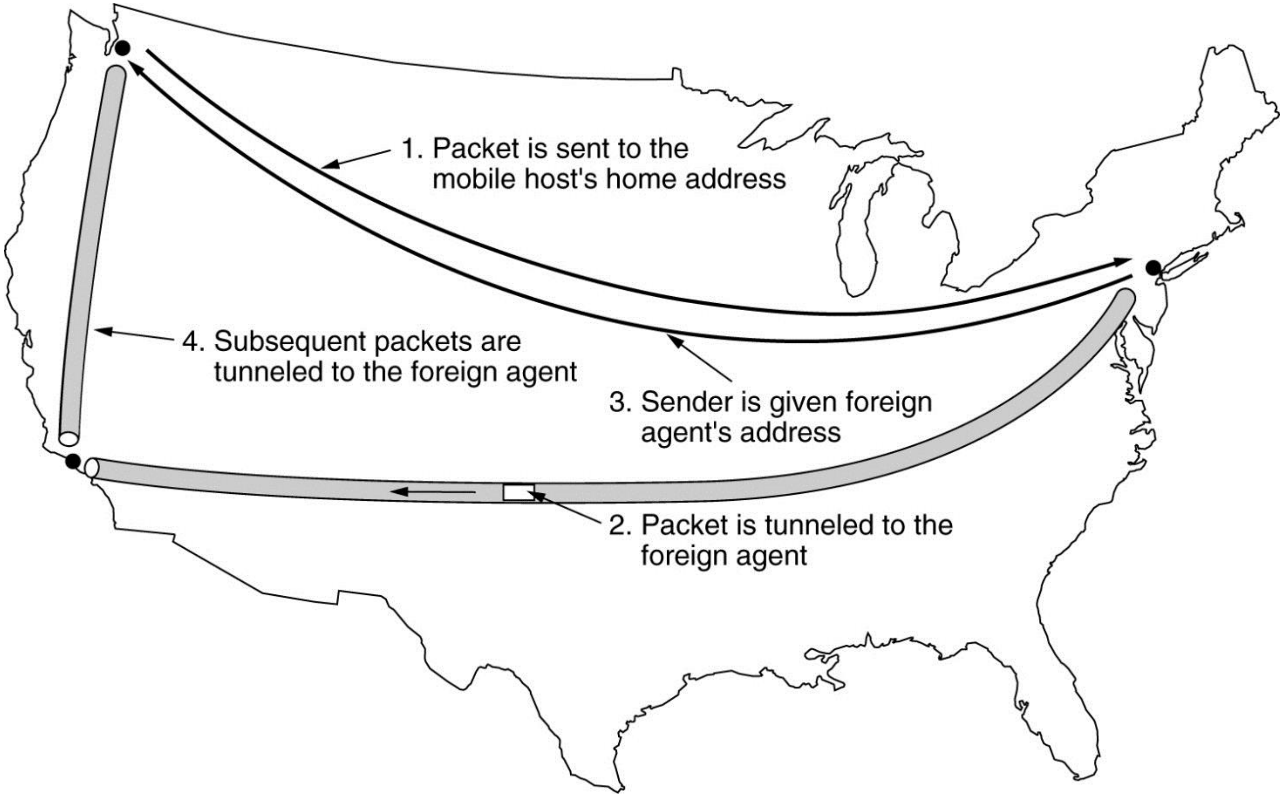
Cont.

- The MH in a foreign area must acquire a local network address, called a **care of address**, before it can use the network.
    - Once the MH has this address, it can tell its home agent where it is now, by sending a registration message to the home agent, in Step 1.

Triangle routing:

- When a packet is sent to an MH,
    - it is routed to the user's home location because that is where the home address belongs, in step 2.
        - Packets sent to the home location of the MH is intercepted by the home agent.
    - The home agent then does two things:
        - First, it encapsulates the packet in the payload field of an outer packet and send it to the care of address, in step 3, through **tunneling.**
            - After getting the encapsulated packet, the MH removes the original packet from the payload field and sends its reply packet directly to the sender, in step 4.

Routing for Mobile Hosts (2)(in 4th ed)


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

Packet routing for mobile users.

Routing in Ad Hoc Networks

Possibilities when the routers are mobile:

1. Military vehicles on battlefield.
   - No infrastructure.
2. A fleet of ships at sea.
   - All moving all the time
3. Emergency workers at an earthquake site
   - The infrastructure destroyed.
4. A gathering of people with notebook computers.
   - In an area lacking 802.11.

Cont.

- Each node acts both as a router and a host.
- Networks of nodes that happen to be near each other are called **ad hoc** networks or **MANET**s (Mobile Ad hoc NETworks).
  - No fixed topology, fixed and known neighbors, fixed relationship between IP address and location,…

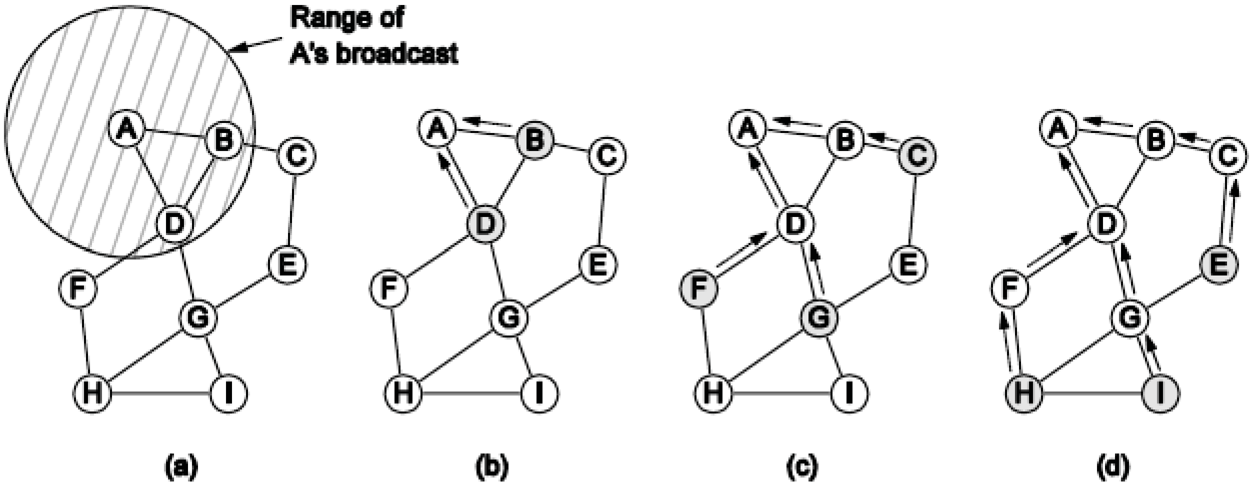
(Cont.) AODV—Ad hoc On-demand Distance Vector routing algorithm

- It is based on the Bellman-Ford distance vector algorithm but adapted to work in mobile environment, taking into account the limited bandwidth and low battery life.
- It is an on-demand algorithm.
  - Routes to a destination are discovered only when somebody wants to send a packet to that destination.

Route Discovery

- An ad hoc network can be described by a graph of nodes. Two nodes are connected (i.e., have an arc between them in the graph) if they can communicate directly using their radios.
- The AODV algorithm maintains a table at each node, keyed by destination, giving information about that destination, including which neighbor to send packets to in order to reach the destination.
  - Suppose that *A* looks in its table and does not find an entry for *I*. It now has to discover a route to *I* ---- “on demand”.

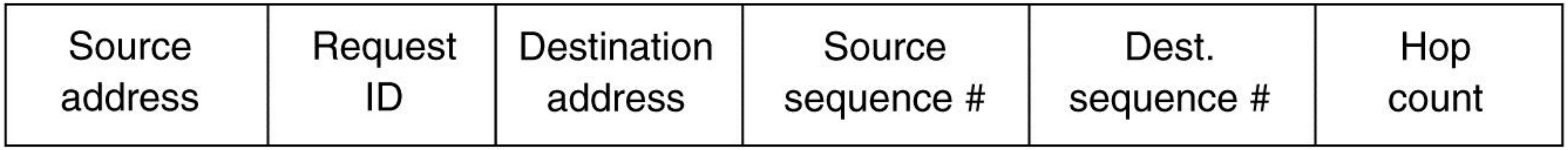
Routing in Ad Hoc Networks



(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.

Route Discovery (2)



Source address	Request ID	Destination address	Source sequence #	Dest. sequence #	Hop count

Format of a ROUTE REQUEST packet.

Cont.

- The *Source address* and *Request ID* fields uniquely identify the ROUTE REQUEST packet to allow nodes to discard any duplicates they may receive.
- Each node also maintains a *sequence* counter incremented whenever a ROUTE REQUEST is sent (or a reply to someone else's ROUTE REQUEST).
- The *Hop count* will keep track of how many hops the packet has made. It is initialized to 0.

Cont.

When a ROUTE REQUEST packet arrives at a node, it is processed as follows:

1. The (*Source address*, *Request ID*) pair is looked up in a local history table to see if this packet has already been seen and processed.
   - If it is a duplicate, it is discarded and processing stops.
   - If it is not a duplicate, the pair is entered into the history table so future duplicates can be rejected, and processing continues.
2. The receiver looks up the destination in its route table. If a **fresh** route to the destination is known, a ROUTE REPLY packet is sent back to the source telling it how to get to the destination.
   - Fresh means that the *Destination sequence number* stored in the routing table is greater than or equal to the *Destination sequence number* in the ROUTE REQUEST packet.
   - If it is less, the stored route is older than the previous route the source had for the destination, so step 3 is executed.
3. Since the receiver does not know a fresh route to the destination, it increments the *Hop count* field and rebroadcasts the ROUTE REQUEST packet.
   - It also extracts the data from the packet and stores it as a new entry in its reverse route table. This information will be used to construct the reverse route so that the reply can get back to the source later.
   - A timer is also started for the newly-made reverse route entry. If it expires, the entry is deleted.

Route Discovery (3)

Source address	Destination address	Destination sequence #	Hop count	Lifetime

Format of a ROUTE REPLY packet.

Cont.

- When the destination receives the ROUTE REQUEST, it builds a ROUTE REPLY packet.
  - The *Source address*, *Destination address*, and *Hop count* are copied from the incoming packet, but the *Destination sequence number* taken from its counter in memory.
  - The *Hop count* field is set to 0.
  - The *Lifetime* field controls how long the route is valid.
- The packet is unicast to the node that the ROUTE REQUEST packet came from. It then follows the reverse path to the source.
  - At each node, *Hop count* is incremented so the node can see how far from the destination it is.

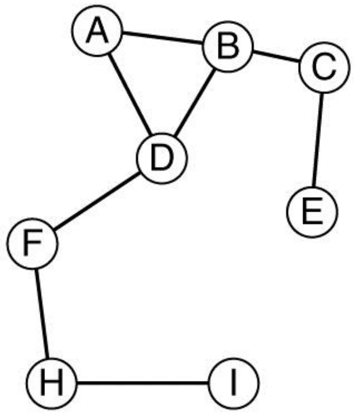
Cont.

- At each intermediate node on the way back, the packet is inspected. It is entered into the local routing table as a route to the destination (I) if one or more of the following three conditions are met:
  - No route to the destination (I) is known.
  - The sequence number for the destination (I) in the ROUTE REPLY packet is greater than the value in the routing table.
  - The sequence numbers are equal but the new route is shorter.
- Nodes that got the original ROUTE REQUEST packet but were not on the reverse path discard the reverse route table entry when the associated timer expires.
- In a large network, the algorithm generates many broadcasts, even for destinations that are close by.
  - Using "*Time to live*" to control the distance of each broadcast.

Route Maintenance

Dest.	Next hop	Distance	Active neighbors	Other fields
A	A	1	F, G	
B	B	1	F, G	
C	B	2	F	
E	G	2		
F	F	1	A, B	
G	G	1	A, B	
H	F	2	A, B	
I	G	2	A, B	

(a)



(b)

(a) D's routing table before G goes down.

(b) The graph after G has gone down.

Cont.

- Because nodes can move or switched off, the topology change spontaneously.
  - Broadcast a *Hello* message periodically to learn of the change.
  - Similarly, if a node tries to send a packet to a neighbor that does not respond, it learns that the neighbor is no longer available.
- Purge routes that are no longer work:
  - For each possible destination, each node (N) keeps track of its neighbors that have fed it a packet for that destination during the last ΔT seconds.
    - These are called N's **active neighbors** for that destination.
    - N does this by having a routing table keyed by destination and containing the outgoing node to use to reach the destination, the hop count to the destination, the most recent destination sequence number, and the list of active neighbors for that destination.
  - When any of N's neighbor becomes unreachable, it checks its routing table to see which destinations have routes using the now-gone neighbor. For each of these routes, the active neighbors are informed that their route via N is now invalid and must be purged from their routing tables.
    - The active neighbors then tell their active neighbors, and so on, recursively, until all routes depending on the now-gone node are purged from all routing tables.

Other ad hoc routing schemes

- Dynamic Source Routing (DSR), Johnson et al., 2001
- Greedy Perimeter Stateless Routing (GPSR), Karp and Kung, 2000
  - Based on geography
- ……

Congestion Control

1899 Horsey Horseless

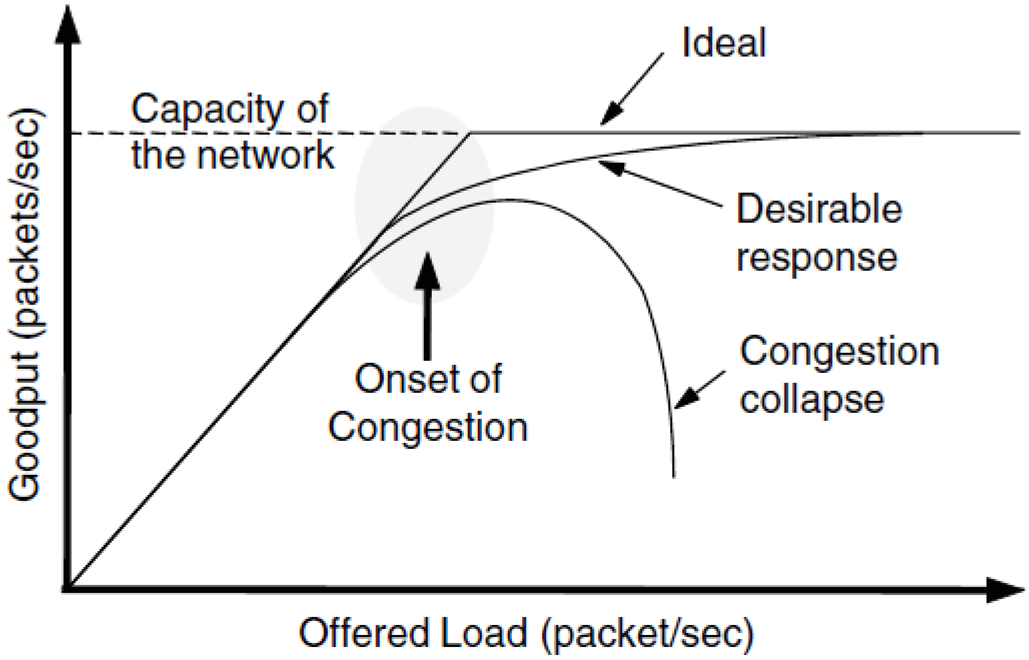
Congestion Control Algorithms

- Congestion:
  - *When* too many packets are present in (a part of) the subnet, performance degrades.
- When a queue builds up at some node for an output line, adding more memory may reduce packet loss probability.
  - If routers have an infinite amount of memory, congestion gets worse.
    - By the time packets get to the front of the queue, they have already timed out (repeatedly) and duplicates have been sent.
- Slow processors and/or low-bandwidth lines may cause congestion.

Cont.

- Difference between congestion control and flow control:
  - Congestion control
    - Make sure the subnet is able to carry the offered traffic.
    - It is a global issue, involving the behavior of all the hosts, all the routers, the store-and-forwarding processing within the routers, and all the other factors that tend to diminish the carrying capacity of the subnet.
  - Flow control
    - Relate to the point-to-point traffic between a given sender and a given receiver.
    - Make sure that a faster sender can not continually transmit data faster than the receiver can absorb it.

Congestion Control Algorithms (2)



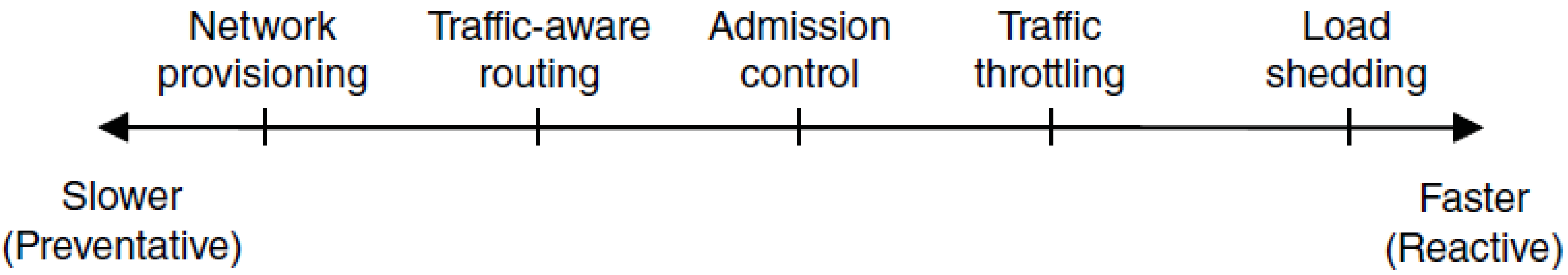
When too much traffic is offered, congestion sets in and performance degrades sharply.

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

Congestion Control Algorithms (1)

- Approaches to congestion control
- Traffic-aware routing
- Admission control
- Traffic throttling
- Load shedding

Approaches to Congestion Control



Timescales of approaches to congestion control

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

Congestion control

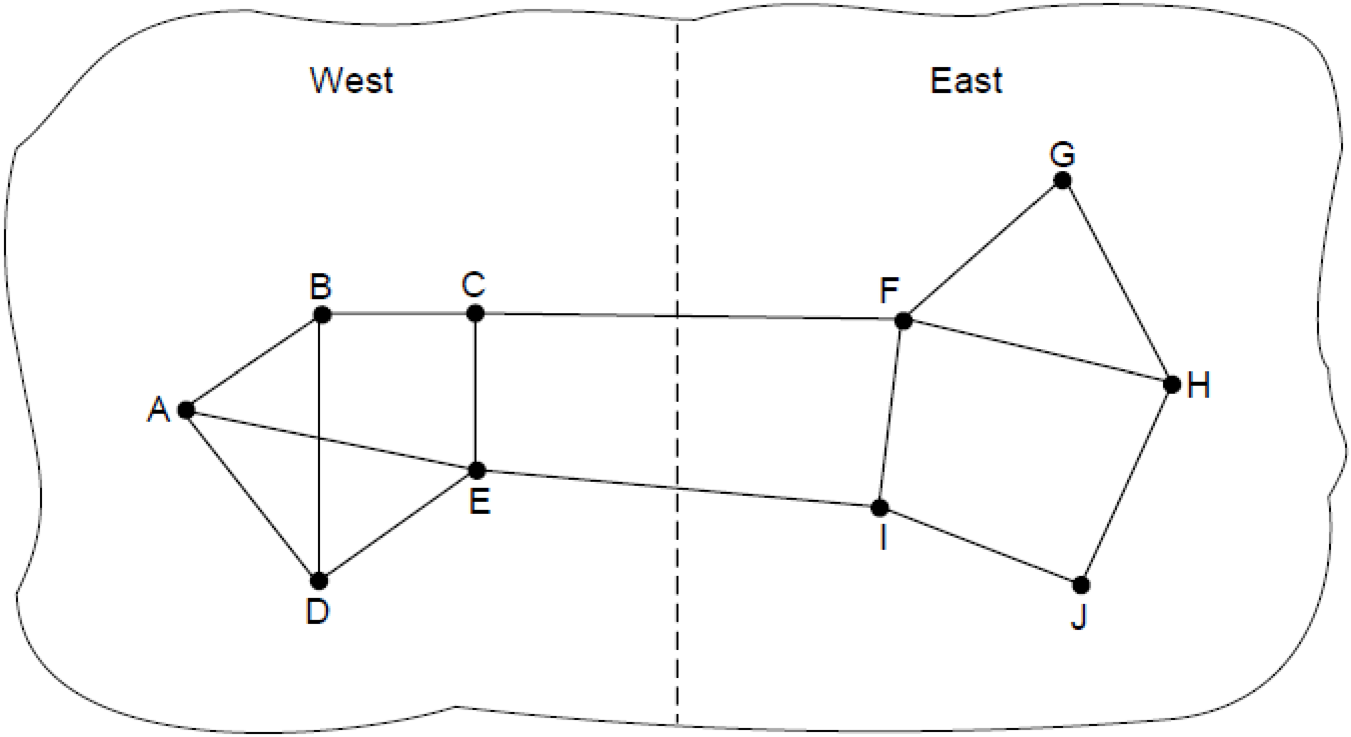
a) Two solutions to congestion:

- Increase the resource
  - Preventive approaches in a longer timescale.
- Decrease the load
  - React to it once it has occurred, taking effects in a shorter time scale.

(cont)

- Provisioning
  - Upgrade links and routers regularly according to utilization at the earliest months.
- Traffic-aware routing
  - Dynamic/adaptive routing
- Admission control
- Traffic throttling
  - The network delivers feedback to the sources whose traffic flows are responsible for the problem.
    - Two issues:
      - How to identify the onset of congestion and
      - How to inform the source that needs to slow down.
- Load shedding
  - The network discards packets that it cannot deliver.

Traffic-Aware Routing



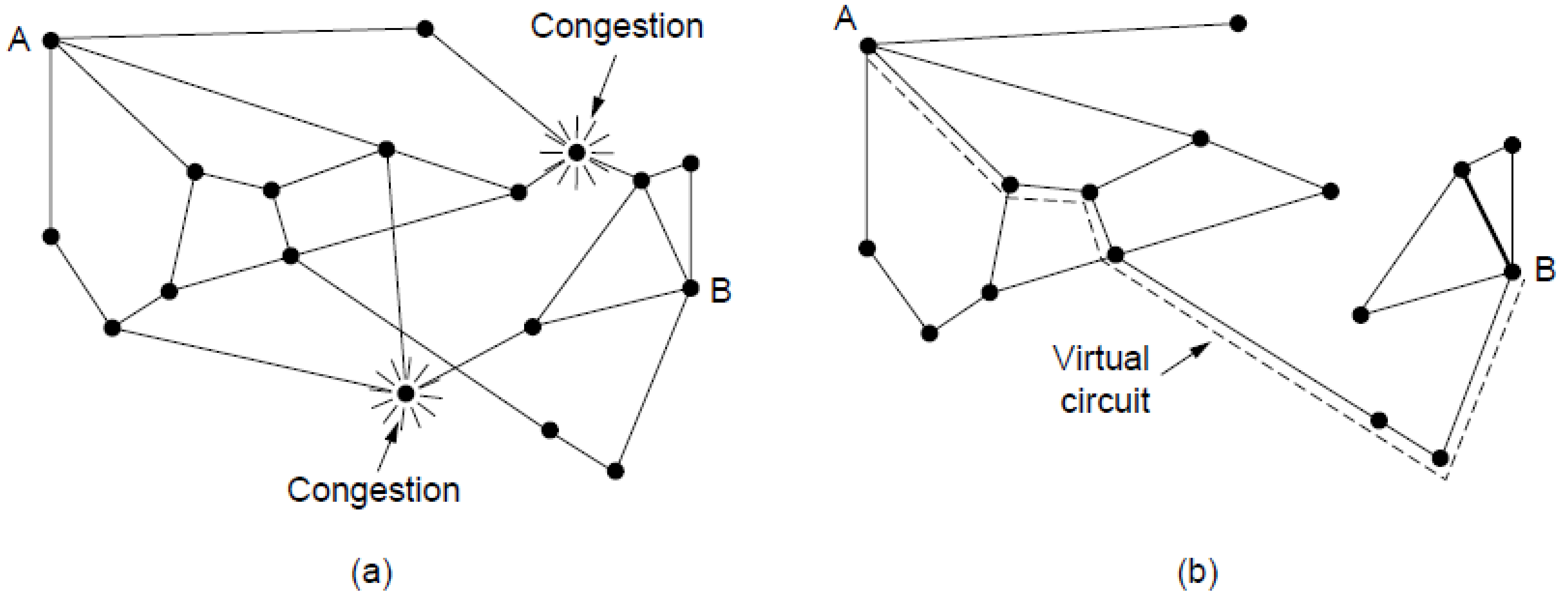
A network in which the East and West parts are connected by two links. (One path vs multiple path routing)

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

Admission Control

- A technique widely used in virtual-circuit networks
  - Do not set up a new virtual circuit unless the network can carry the added traffic without becoming congested.
- Traffic descriptor
  - A commonly used one 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.
  - E.g. Load-sensitive routing

Admission control



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

Traffic throttling

- Congestion avoidance:
  - Determine when congestion is approaching.
    - By monitoring the utilization of the output line, the buffering of queued packets insides the routers, and/or the number of packets that are lost due to insufficient buffering.
    - E.g, Exponentially Weighted Moving Average:

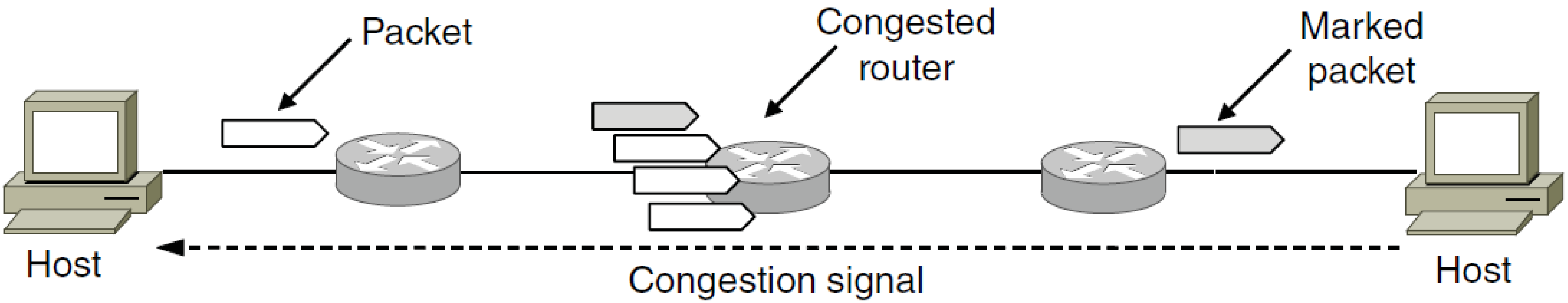
      $$d_{new}=\alpha d_{old}+(1-\alpha)s$$

      where d is the estimate of the queueing delay, s is a sample of the instantaneous queue length/time, and the constant *a* $\in[0,1]$ determines how fast the router forgets recent history.
  - Routers deliver timely feedback to the senders that are causing the congestion:
    - Possible methods: Choke packets, explicit congestion notification, hop-by-hop backpressure.

Choke Packet

- The router sends a **choke packet** back to the source host directly.
  - The original packet is tagged (a header bit is turned on) so that it will not generate any more choke packets further along the path and is then forwarded in the usually way.
- When the source host gets the choke packet, it is required to reduce the traffic sent to the specified destination by *X* percent.
  - The host should ignore choke packets referring to that destination for a fixed interval.
  - After that period has expired, the host listens for more choke packets for another interval.
    - If one arrives, the line is still congested. The host reduces the flow still more and begins ignoring choke packets again.
    - If no choke packets arrive during the listening period, the host may increase the flow again.
- E.g. the SOURCE QUENCH message in the early Internet.

Traffic Throttling (2)



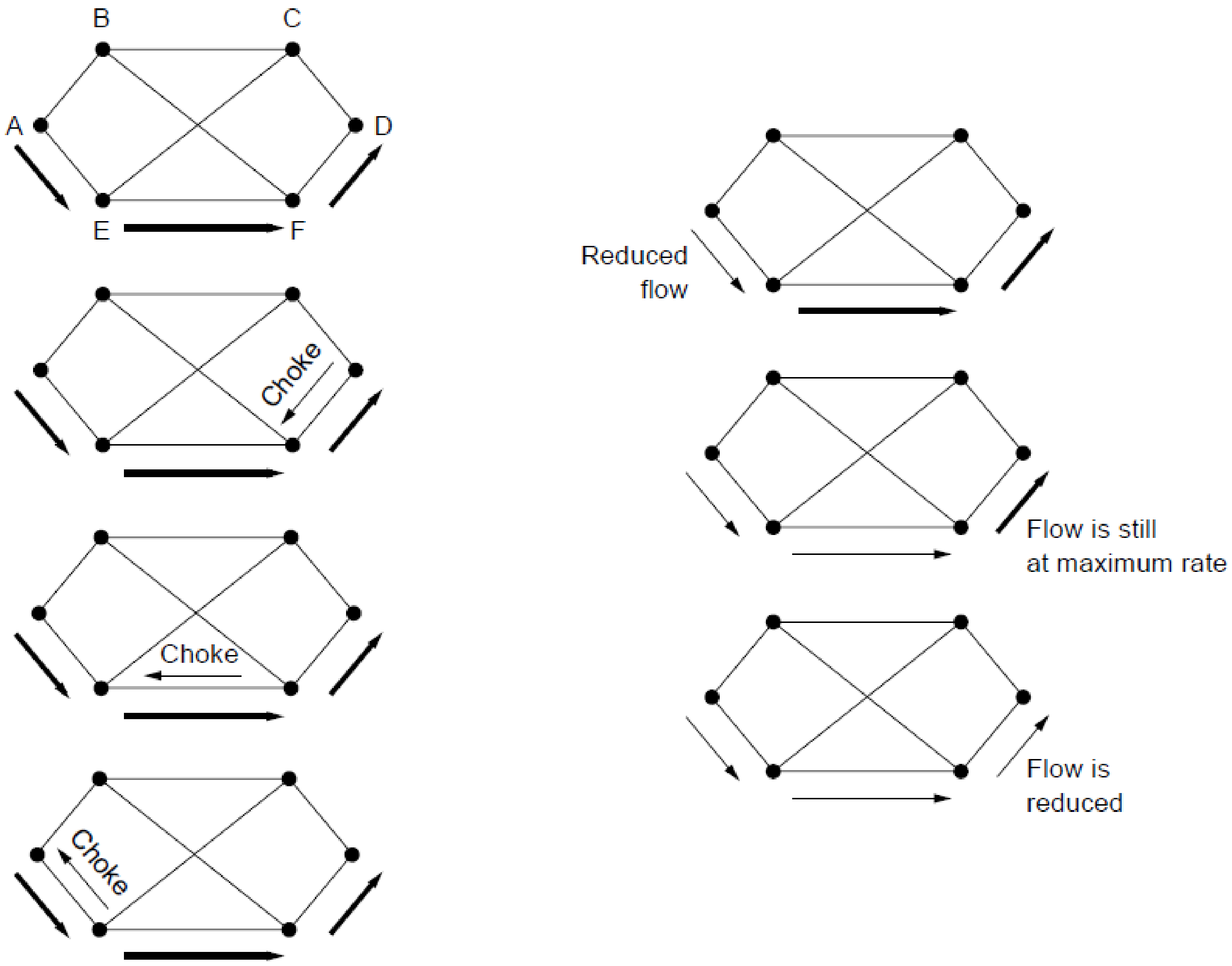
Explicit congestion notification

Explicit congestion notification

- The router signals the warning state of congestion by setting a special bit in the packet header.
- When the packet arrived at the destination, the transport entity copied the bit into the next acknowledgement/reply sent back to the source.
- The source then cut back on traffic.

- Advantage: No additional packets are injected into the network.

Hop-by-Hop Backpressure (1)



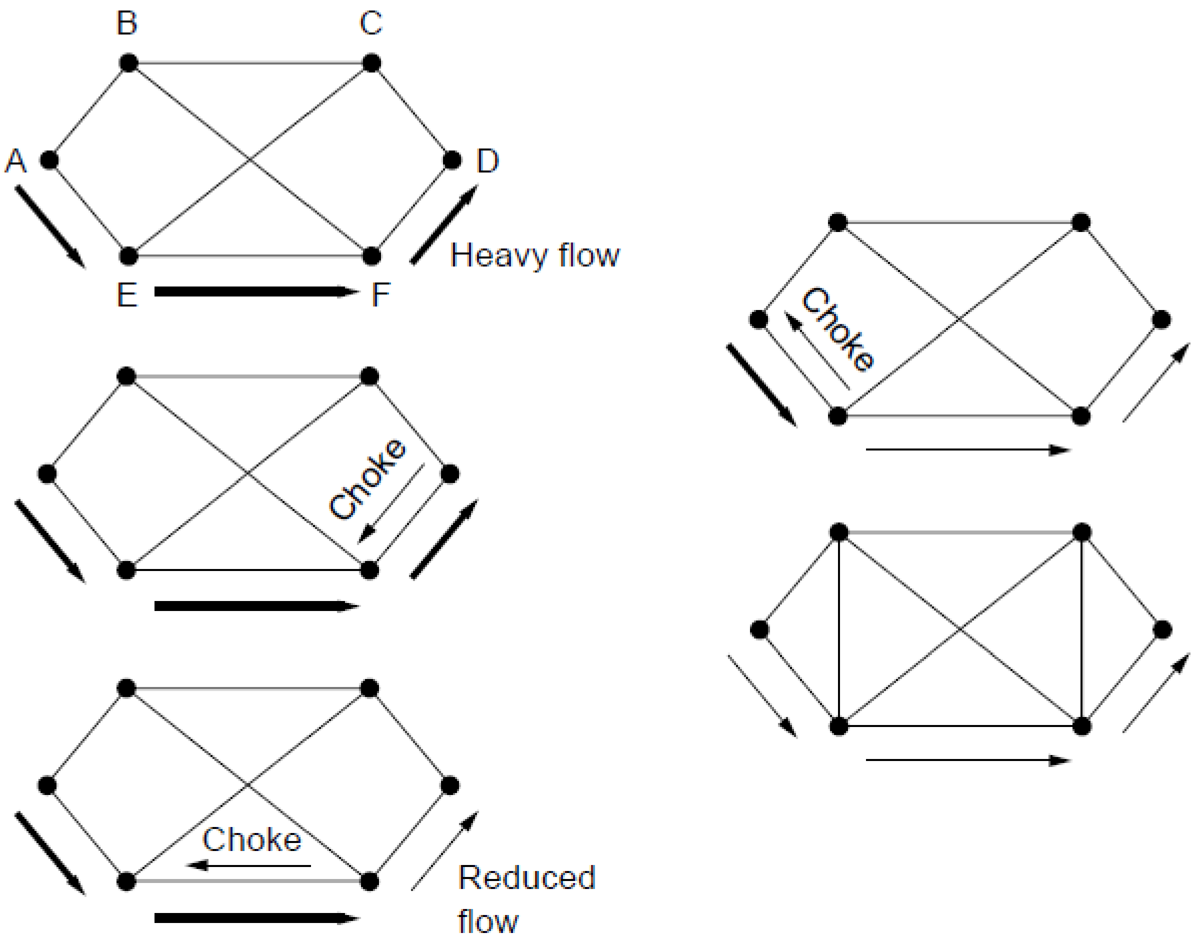
A choke packet that affects only the source..

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

Hop-by-Hop Backpressure

- At high speeds and over long distances, sending a choke/ECN packet to the source hosts does not work well because the reaction is slow.
- 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.

Hop-by-Hop Backpressure (2)



A choke packet that affects each hop it passes through.

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

Load Shedding

- **Load Shedding**: when routers are being inundated by packets that they can not handle, they just throw them away.
  - A router drowning in packets can just pick packets at random to drop, or
  - which packet to discard may depend on applications running:
    - For file transfer, an old packet is worth more than a new one. (**wine**)
    - For multimedia, a new packet is more important than an old one. (**milk**)
  - To implement an intelligent discard policy, applications must mark their packets in priority class to indicate how important they are.
  - Another option is to allow hosts to exceed the limits specified in the agreement negotiated when the virtual circuit was set up, but subject to the condition that all excess traffic be marked as low priority.

Random Early Detection (RED)

- Dealing with congestion after it is first detected is more effective than letting it gum up the networks and then trying to deal with it.
  - RED: Discard packets before all buffer space is really exhausted.
  - In some transport protocols (including TCP), the response to lost packets is for the source to slow down:
    - The reasoning is that TCP was designed for wired networks and wired networks are very *reliable*, so lost packets are mostly due to *buffer overruns rather than transmission errors*.
    - To work well with TCP in wireless networks, transmission errors due to *noise on the air link* must recover at the data link layer.

    - ECN (explicit signal) ↔ RED (Implicit signaling)

Quality of Service

- Overprovisioning, a simple but expensive solution
- Others:
  - Four issues must be addressed:

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.

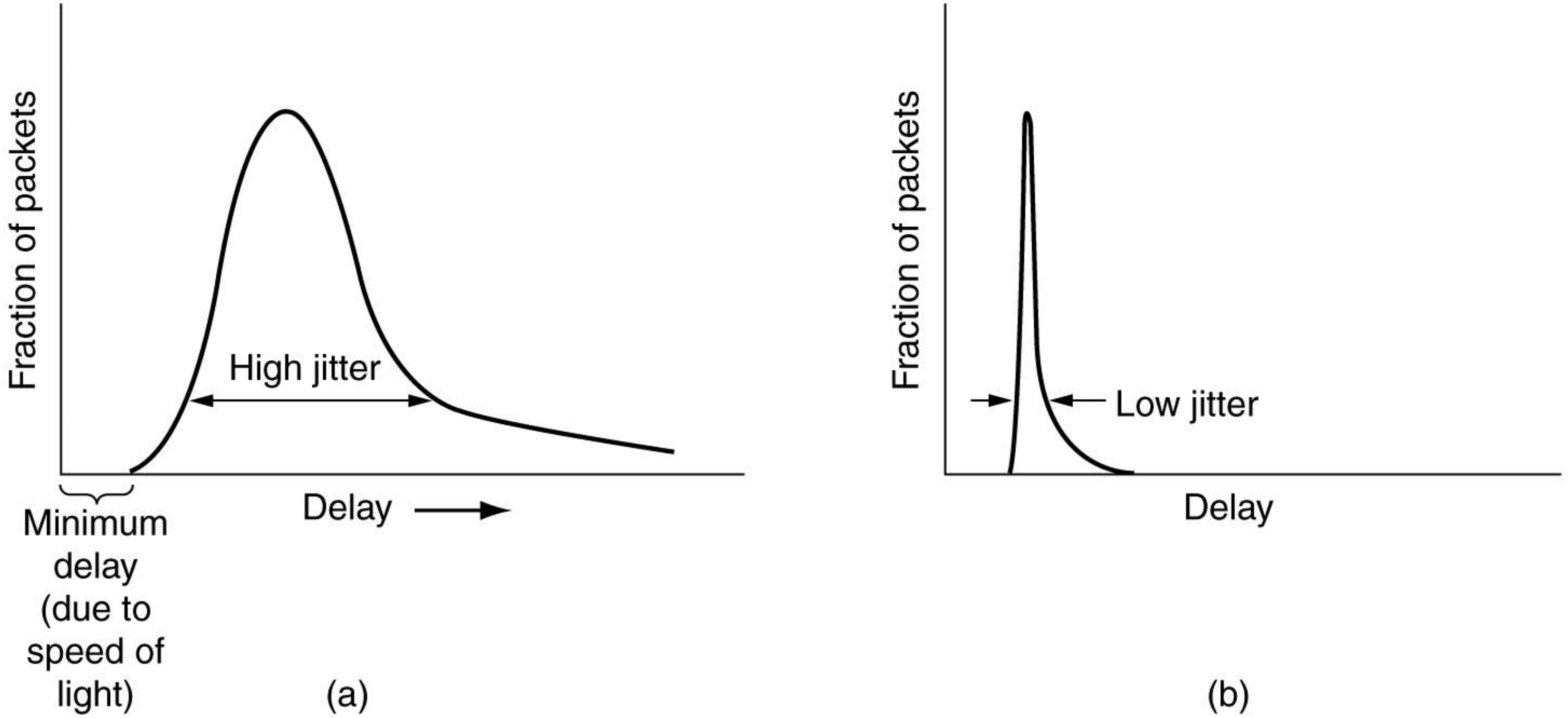
Quality of Service

- Application requirements
- Traffic shaping
- Packet scheduling
- Admission control
- Integrated services
- Differentiated services

Cont.

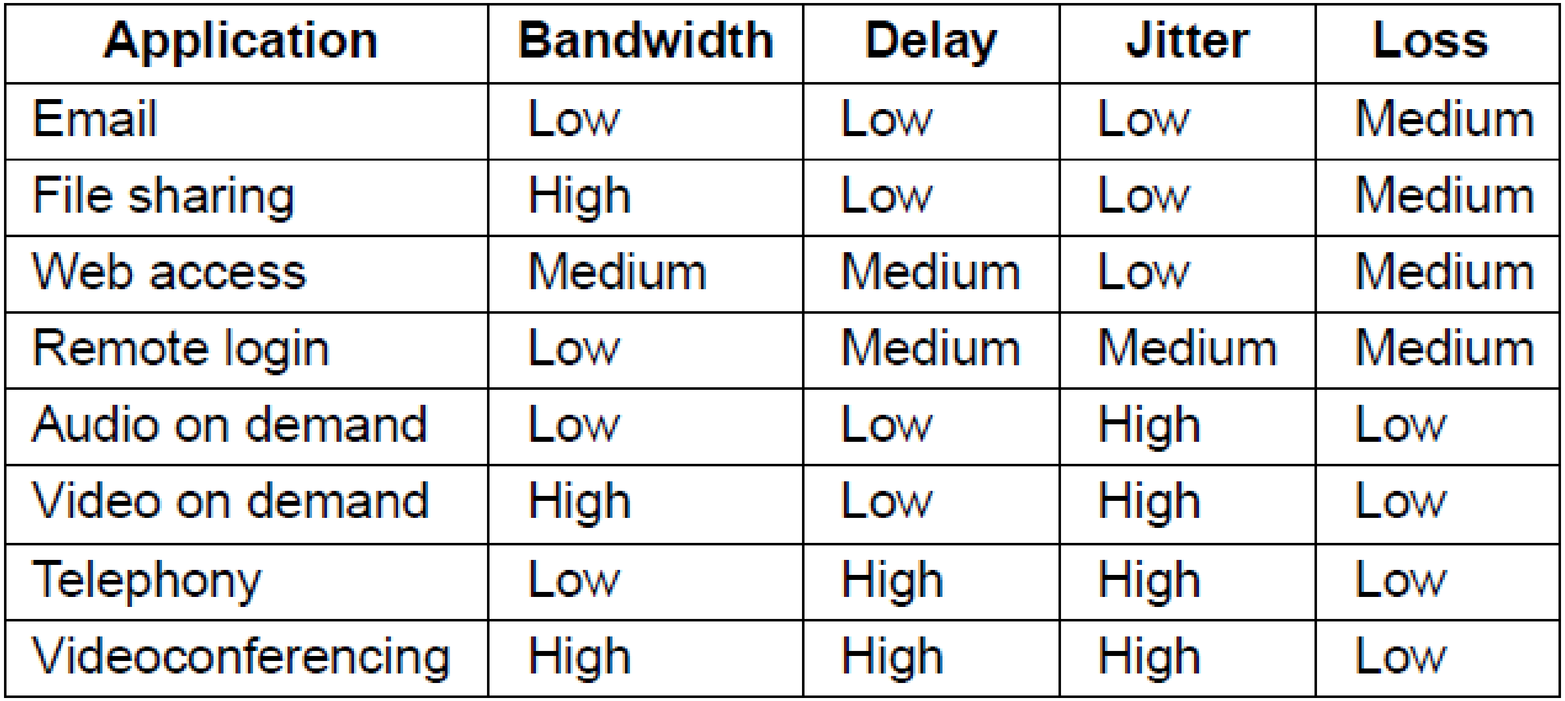
- A stream of packets from a source to a destination is called a **flow**.
  - In a connection-oriented network, all the packets belonging to a flow follow the same route.
  - In a connectionless network, all the packets sent from one process to another process may follow different route.
- QoS parameters
  - Bandwidth, delay, jitter, and loss(reliability).

Jitter



(a) High jitter. (b) Low jitter.

Application Requirements (1)



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

How stringent the quality-of-service requirements are.

Cont.

- ATM networks classify flows in four broad categories with respect to their QoS demands (These categories are also useful for other purposes and other networks) :
  - CBR: Constant bit rate (e.g., telephony)
  - rt-VBR: Real-time variable bit rate (e.g., compressed videoconferencing).
  - nrt-VBR: Non-real-time variable bit rate (e.g., watching a movie over the Internet).
  - ABR: Available bit rate (e.g., file transfer).

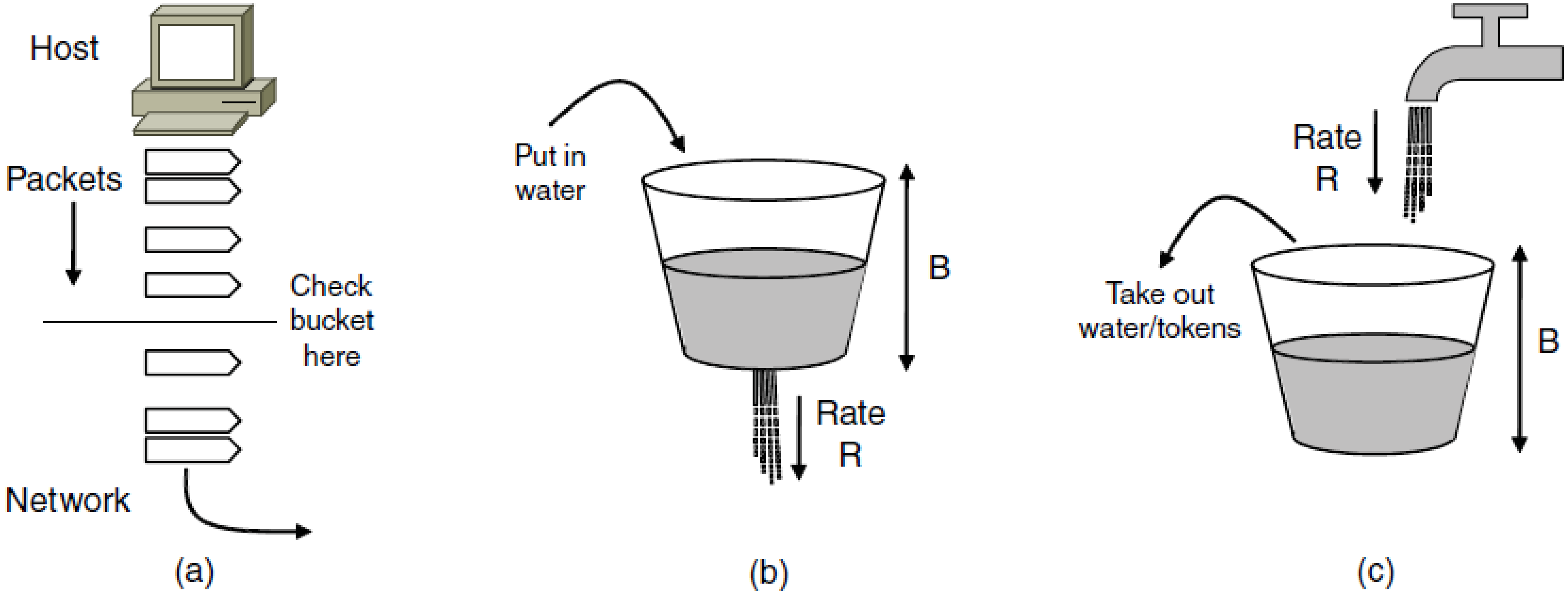
Categories of QoS and Examples

1. Constant bit rate
   - Telephony
2. Real-time variable bit rate
   - Compressed videoconferencing
3. Non-real-time variable bit rate
   - Watching a movie on demand
4. Available bit rate
   - File transfer

Traffic Shaping and traffic policing

- Traffic characteristics in telephone networks VS traffic characteristics in data networks.
  - Traffic in data networks is **bursty**.
  - Bursts of traffic are more difficult to handle than constant-rate traffic.
- **Traffic shaping** is about regulating the average *rate* and burstiness of a flow of data that enters the network.
  - In contrast, the sliding window protocols use one parameter to limit the amount of data in transit at once, indirectly limiting the rate.
- Traffic shaping reduces congestion and thus helps the carrier live up to its promise of **service level agreement**.
- Monitoring a traffic flow is called **traffic policing**.
  - Agreeing to a traffic shape and policing it afterward are easier with virtual circuit subnets than with datagram subnets. With datagram subnets, the same ideas can be applied to transport layer connections.

Traffic Shaping (1)



(a) Shaping packets. (b) A leaky bucket. (c) A token bucket

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

Leaky bucket algorithm

$$R_{out}(t) = \begin{cases} R & , B(t) > 0 \\ 0 & , B(t) = 0 \end{cases}$$

where $R_{out}(t)$ is the leaky bucket output rate and $B(t)$ the volume of water in the bucket at time t.

$$B(t+\Delta) = \left\{ \overline{B} \wedge \left(B(t) + \int_{t}^{t+\Delta} [R_{in}(x) - R_{out}(x)]dx \right) \right\}^{+}$$

where $R_{in}(t)$ is the input rate to the bucket at time t and $\overline{B}$ the bucket size.

Token bucket

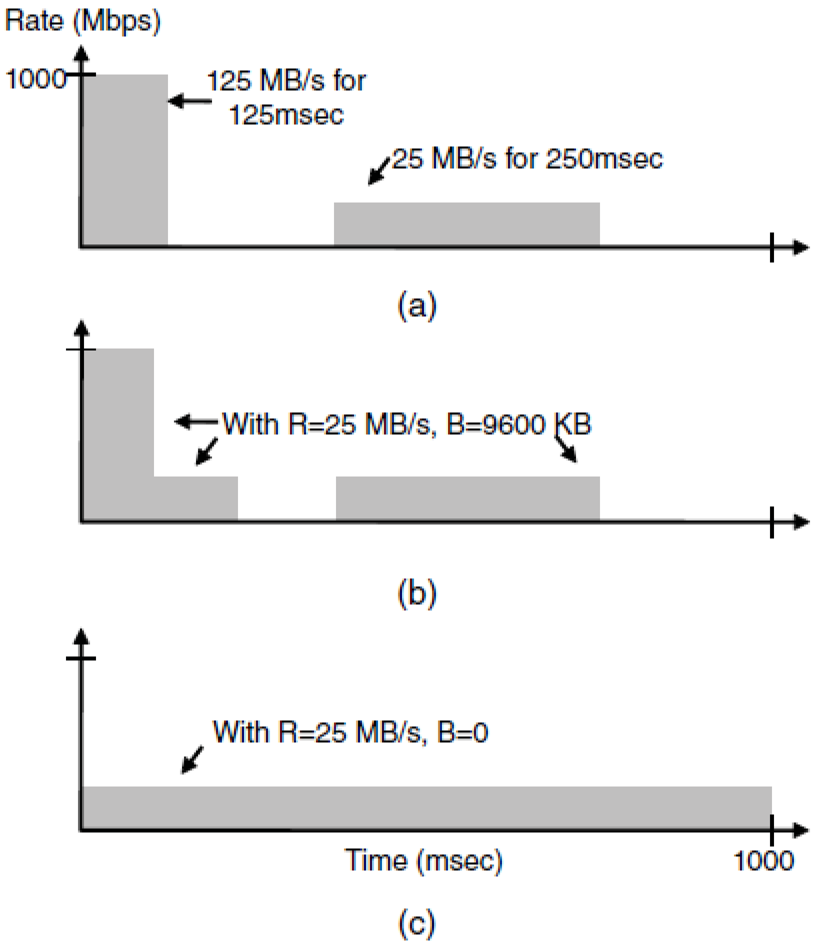
A (σ, ρ) traffic model :

σ : Burstiness

ρ : Token rate or long term average rate

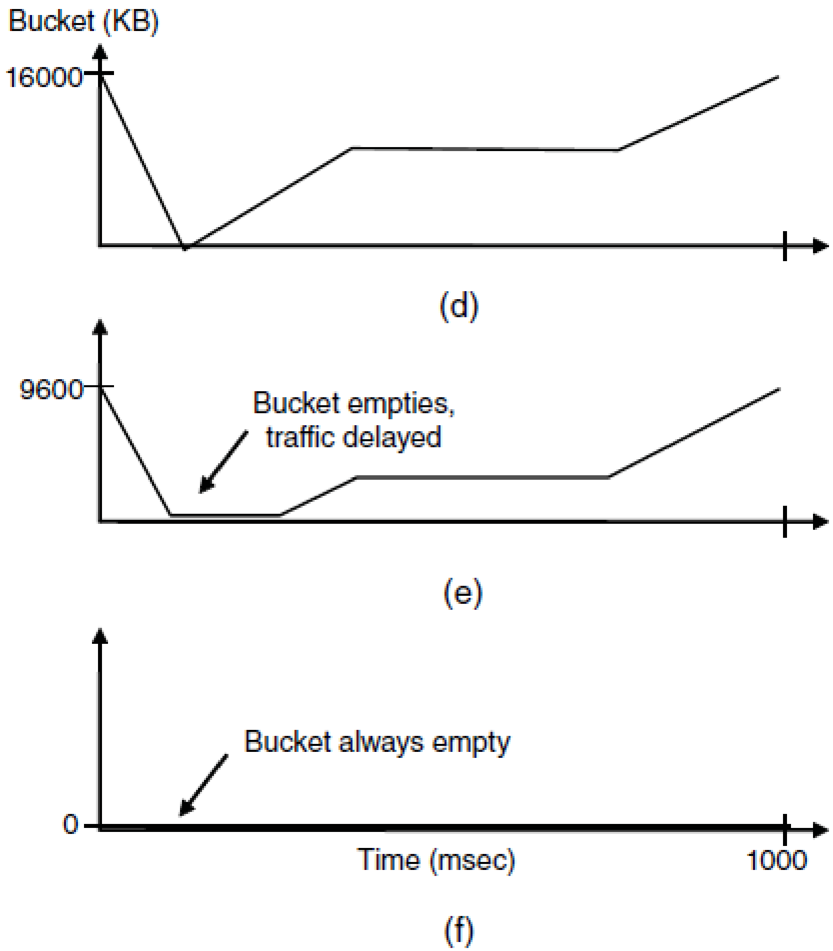
$$A(t, t+s) = \int_{t}^{t+s} R_{out}(\tau) d\tau \leq \sigma + \rho \times s, \text{ for all } t > 0 \text{ and } s > 0.$$

Traffic Shaping (2)



(a) Traffic from a host. Output shaped by a token bucket of rate 200 Mbps and capacity (b) 9600 KB, (c) 0 KB.

Traffic Shaping (3)



Token bucket level for shaping with rate 200 Mbps and capacity (d) 16000 KB, (e) 9600 KB, and (f) 0KB..

Cont.

- Calculate the maximum rate burst:

  S: the burst length, in second

  ρ: token arrival rate, in byte/sec

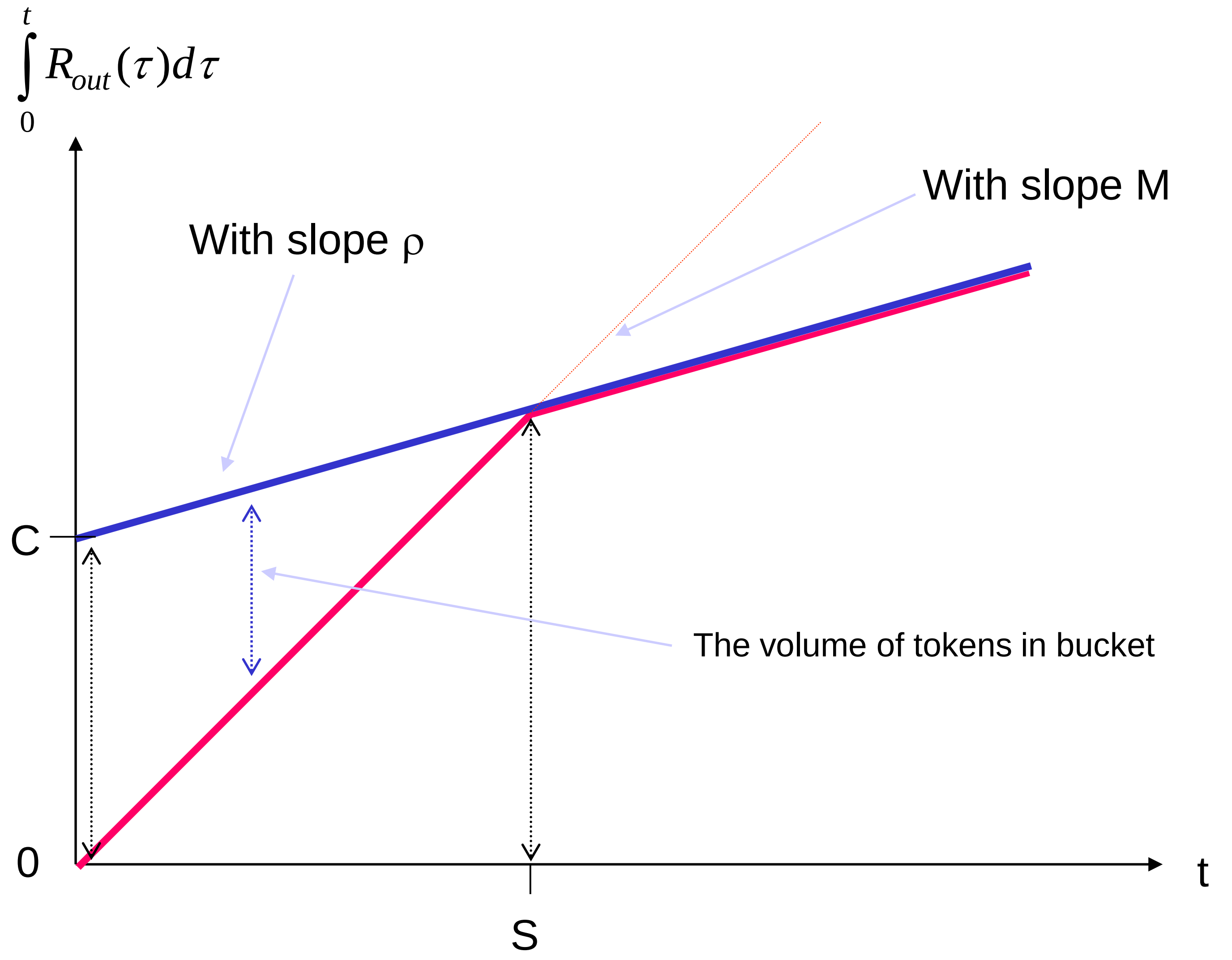
  M: the maximum output rate, in byte/sec

  C : the token bucket capacity, in byte.

  Hence,

  $$C+\rho S=MS,$$

  giving the result $S=C/(M-\rho)$.



Token bucket regulator: $$\int_0^t R_{out}(\tau)d\tau \le \rho t + C, \quad \forall t \in [0,\infty)$$

Packet Scheduling (1)

Kinds of resources can potentially be reserved for different flows:

1. Bandwidth.
2. Buffer space.
3. CPU cycles.

Packet scheduling

1. FIFO, or FCFS
   - **Tail drop**
   - No control for QoS: Aggressive sender can hog most of the capacity of a router and starve the other flows through the same router. → It requires a stronger ***isolation*** between flows.
2. Fair Queueing
3. Weighted Fair Queueing
   - Deficit round robin
4. Priority Queueing
5. Earliest deadline first (EDF)
6. ….

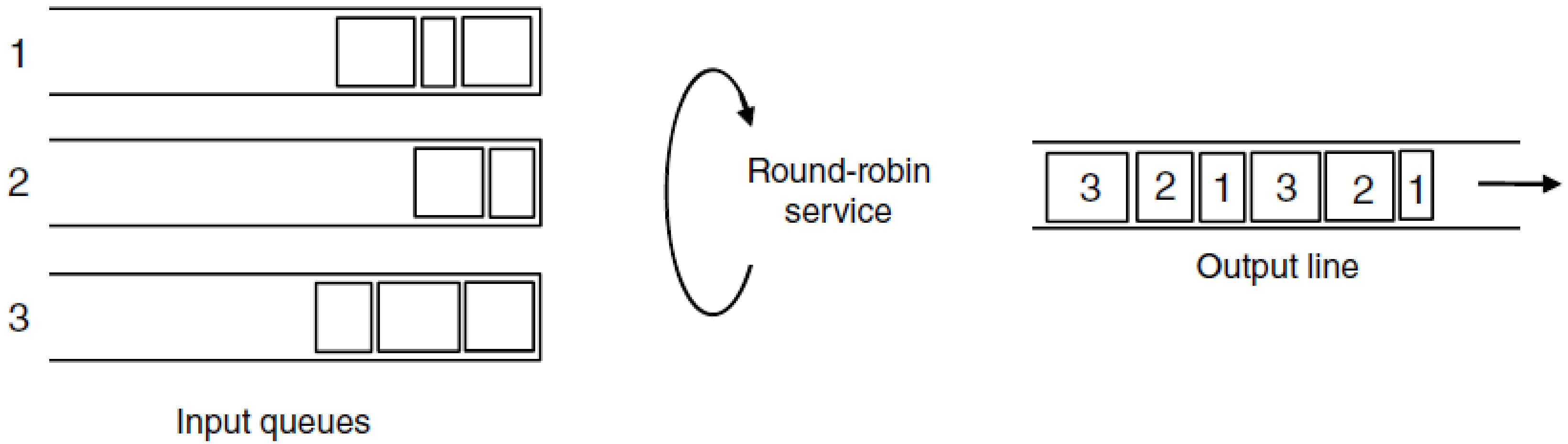
Fair Queueing

- **Fair Queueing** Algorithm (Nagle, 1987):
  - Routers have separate queues for each output line, one for each flow.
  - When a line becomes idle, the router scans the queues **round robin**, taking the first packet on the next queue.
- With *n* hosts competing for a given output line, each host gets to send one out of every *n* packets.
- A problem with the algorithm is that it gives more bandwidth to hosts that use large packets than to hosts that use small packets.

Cont.

- Improvement by Demers et al. (1990):
  - The round robin is done to simulate a byte-by-byte round robin, instead of a packet-by-packet round robin.
    - It compute a virtual time at which each packet would finish being sent.
    - The packets are then sorted in order of their *virtual finishing times* and sent in that order.
- One problem with FQ is that it gives all hosts the same priority.
  - Sometimes, it is desirable to give video servers more bandwidth than regular file servers.
    - The flow from video servers is given a larger weight.
    - This modified algorithm is called **weighted fair queueing** (WFQ, or packet generalized processor server (PGPS)).

Packet Scheduling (2)



Round-robin Fair Queuing

WFQ

W : the weight of a flow, i.e. the number of bytes per round

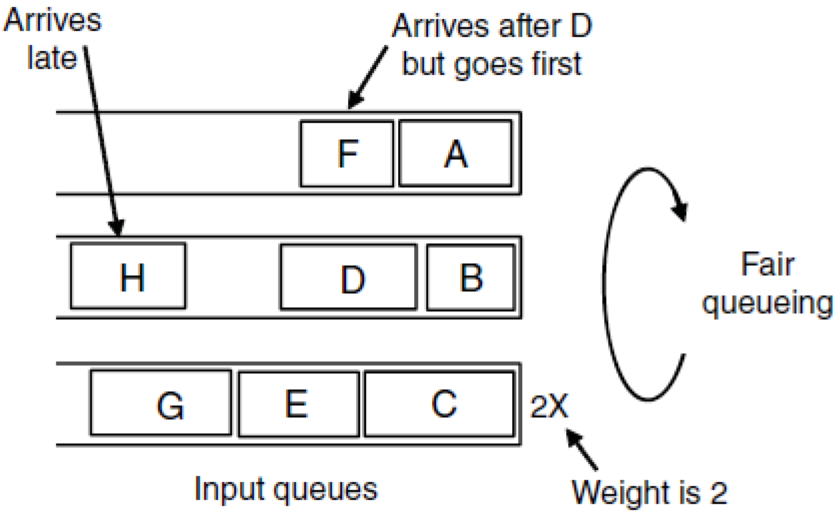
A_i : the arrival time of packet i

L_i : the length of packet i

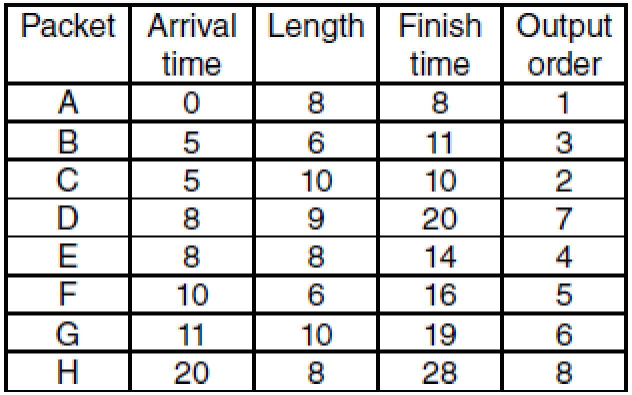
F_i : the virtual finishing time of packet i

$$F_i = \max(A_i, F_{i-1}) + L_i / W$$

Packet Scheduling (3)



(a)



Packet	Arrival time	Length	Finish time	Output order
A	0	8	8	1
B	5	6	11	3
C	5	10	10	2
D	8	9	20	7
E	8	8	14	4
F	10	6	16	5
G	11	10	19	6
H	20	8	28	8

(b)

(a) Weighted Fair Queueing.

(b) Finishing times for the packets.

Implementation complexity

- WFQ requires that packets be inserted by their finish time into a sorted queue.
  - With N flows, this is at least an O(logN) operation per packet.
- **Deficit round robin,** by Shreedhar and Varghese(1995), requires only O(1) operations per packet.

Packet scheduling

- Priority scheme
  - High priority packets are always sent before any low priority packets buffered.
  - Within a priority, packets are sent in FIFO order.
  - Disadvantage: A burst of high priority packets can starve the service of low-priority packets.
- EDF
  - Packets carry timestamps
  - The timestamp records how far the packet is behind or ahead of schedule as it is sent through a sequence of routers on the path.
  - Send packets in the order their timestamps.

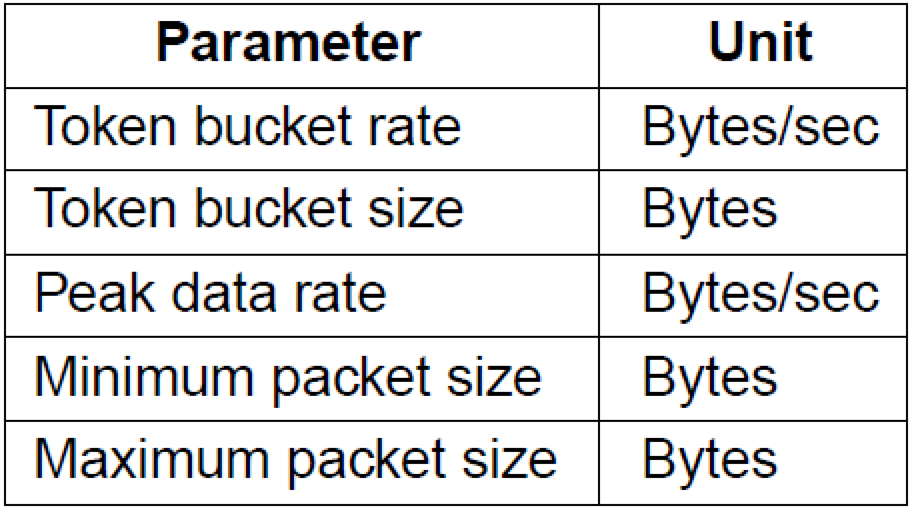
Admission control

- QoS guarantee for new flows over a network.
- QoS routing, rather than the single best path between each source and each destination.
- Translate the parameters of a **flow spec** to the requirement of router resources (bandwidth, buffers, cpu cycles)
- Hard guarantees (Deterministically) versus soft guarantee (Stochastically)
- Does there exist any negotiable flow parameter?

Cont.

- Flows are described in terms of **flow specification**.
  - The sender produces a flow specification proposing the parameters it would like to use.
  - As the specification propagates along the route, each route examines it and modifies the parameters as need be.
  - The modification can only reduce the flow, not increase it.
  - When it gets to the other end, the parameters can be established.
- Five parameters of the flow specification based on RFCs 2210 and 2211
  - *Token bucket rate*, *token bucket size*, *peak data rate*, *minimum packet size*, and *maximum packet size*.
  - The size in the last two includes the transport and network layer headers.

Admission Control (1)



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

An example **flow specification**

(RFCs 2210 and 2211 for IS)

Resource Reservation

- If a specific route for a flow is available, it becomes possible to reserve resources along that route. Three different kinds of resources can potentially be reserved:
  - Bandwidth
  - Buffer space
  - CPU cycles
    - M/M/1 queueing system:
      - Arrivals are Poisson distributed with mean rate λ packets/sec.
      - The service time of each packet is exponentially distributed with mean 1/μ sec.
      - The expected delay experienced by a packet is

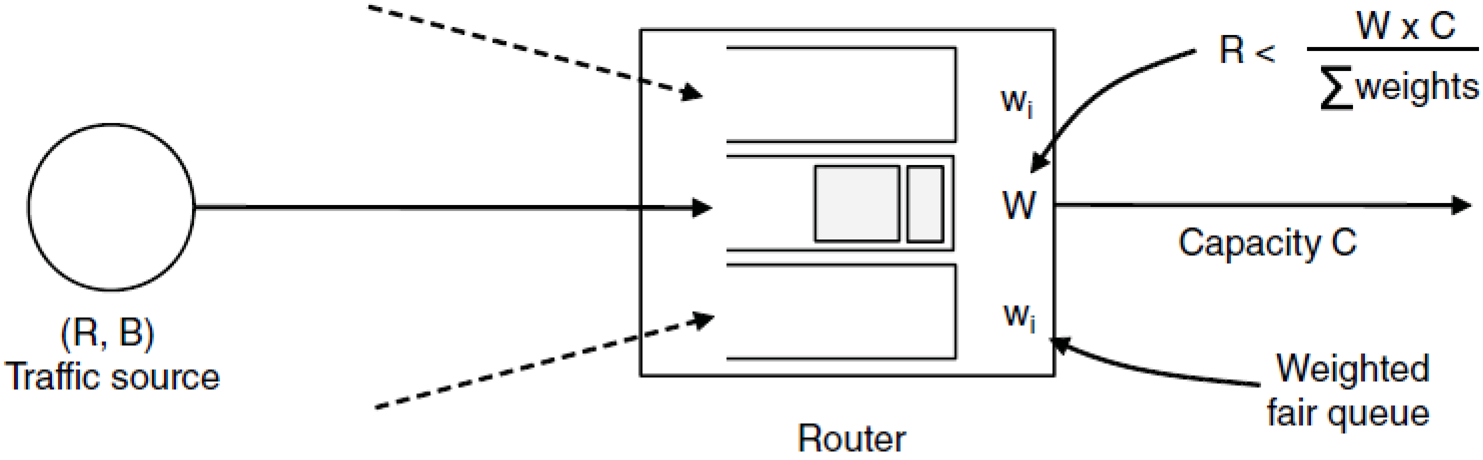
$$T = \frac{1}{\mu} \times \frac{1}{1-\lambda/\mu} = \frac{1}{\mu} \times \frac{1}{1-\rho}$$

      - Where ρ is the (CPU) utilization.

GPS&PGPS, by Parekh and Gallagher

- One method relating flow spec to router resources
  - Traffic sources are shaped by a (R,B), or (σ,ρ), token buckets
  - Packet scheduling methods are WFQ at each router
- Performance guarantee:
  - Bandwidth: minimum bandwidth guarantee ← due to the property of “isolation” given by FQ
  - Delay: worst case delay← due to a bounded burstiness and a minimum bandwidth which gives rise to a bounded delay $B/R < \propto$ (or $\sigma/ \rho< \propto$).
- Give hard guarantee
  - The token buckets bound the burstiness of the source.
  - Fair queueing isolates the bandwidth given to different flows
  - The result holds for a path through multiple routers in any network.

Admission Control (2)



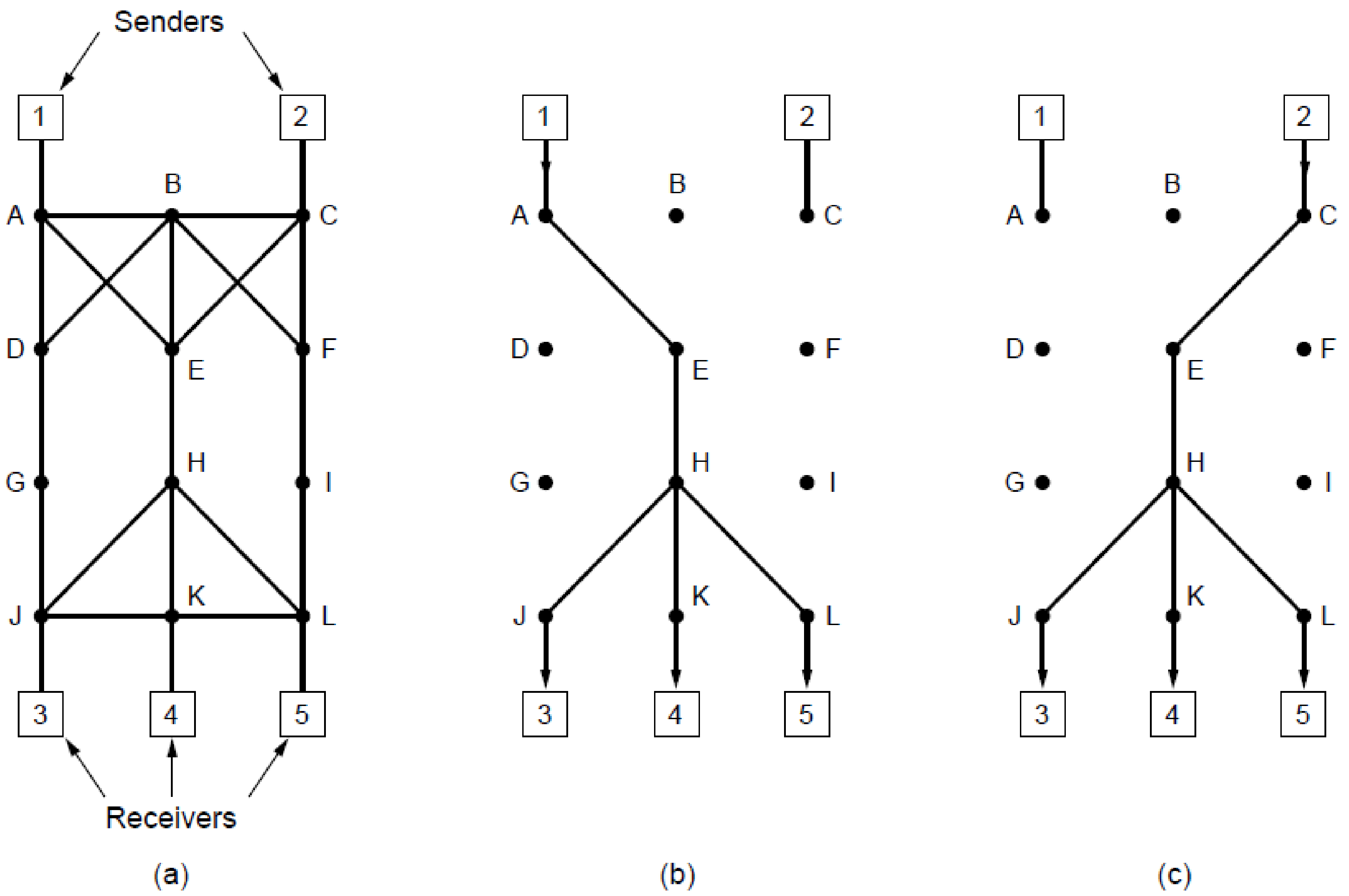
Bandwidth and delay guarantees with token buckets and WFQ.

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

Integrated Services

- An architecture for streaming multimedia:
  - The name for the result is called “flow-based algorithms” or “integrated services.”
  - It was aimed at both unicast and multicast applications.
- In multicast applications, groups can change membership dynamically.
  - The approach of having the senders reserve bandwidth in advance does not work well.

Integrated Services (1)



(a) A network. (b) The multicast spanning tree for host 1. (c) The multicast spanning tree for host 2.

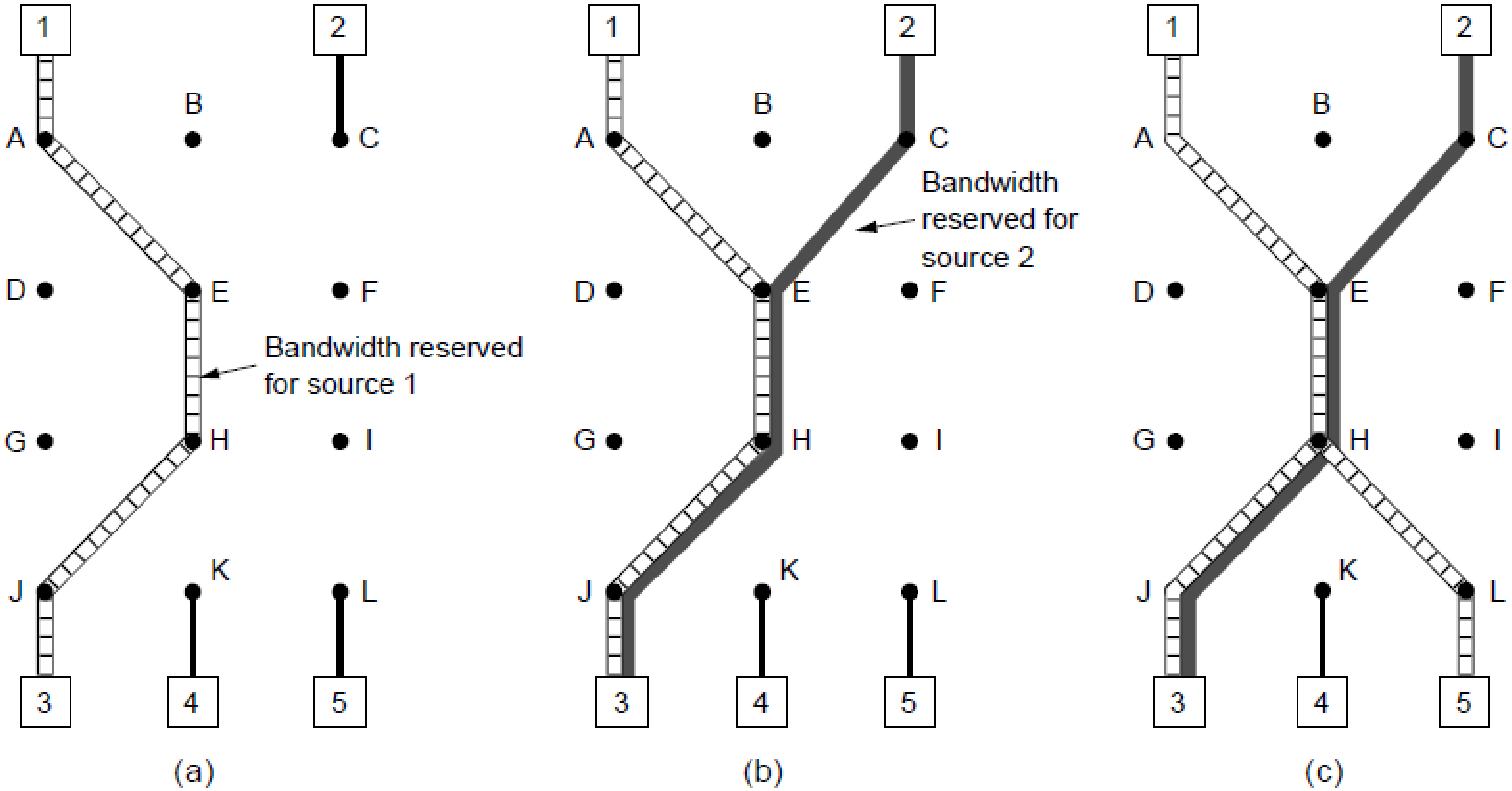
Cont.

- RSVP-The Resource reSerVation Protocol
  - The IS architecture, described in RFCs 2205--2210
  - It is used for making the reservations; other protocols are used for sending the data.
  - It allows multiple senders to transmit to multiple groups of receivers, permits individual receivers to switch channels freely, and optimizes bandwidth use while at the same time eliminating congestion.
- Its simplest form:
  - Multicast routing using spanning trees.
    - Each group is assigned a group address.
    - To send to a group, a sender puts the group's address in its packets.
    - The standard multicast routing algorithm then builds a spanning tree covering all group members. (Routing is not part of RSVP).
    - The only difference from normal multicasting is a little extra information that is multicast to the group periodically to tell the routers along the tree to maintain certain data structures in their memories.

Cont.

- To get better reception and eliminate congestion,
  - Any of the receivers in a group can send a reservation message up the tree to the sender.
  - The message is propagated using *reverse path forwarding algorithm.*
  - At each hop, the router notes the reservation and reserve the necessary bandwidth. If insufficient bandwidth is available, it reports back failure.
  - By the time the message gets back to the source, bandwidth has been reserved all the way from the sender to the receiver making the reservation request along the spanning tree.

Integrated Services (2)


1
2
B
A
C
D
E
F
Bandwidth reserved for source 1
G
H
I
J
K
L
3
4
5
(a)
1
2
B
A
C
Bandwidth reserved for source 2
D
E
F
G
H
I
J
K
L
3
4
5
(b)
1
2
B
A
C
D
E
F
G
H
I
J
K
L
3
4
5
(c)

(a) Host 3 requests a channel to host 1. (b) Host 3 then requests a second channel, to host 2.
(c) Host 5 requests a channel to host 1.

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

Cont.

- When making a reservation,
  - a receiver can (optionally) specify one or more sources that it wants to receive from.
  - It can also specify whether these choices are fixed for the duration of the reservation or whether the receiver wants to keep open the option of changing sources later.
- The routers use this information to optimize bandwidth planning.
  - Two receivers are only set up to share a path if they both agree not to change sources later on.
- Reserved bandwidth is decoupled from the choice of source.
  - Once a router has reserved bandwidth, it can switch to another source and keep that portion of the existing path that is valid for the new source.

Cont.

- Advantage:
  - Flow-based algorithms have the potential to offer good quality of service to one or more flows because they reserve whatever resources are needed along the route.
- Disadvantage
  - They require an advance setup to establish each flow, not scaling well when there are thousands or millions of flows.
  - They maintain internal per-flow state in the routers, making them vulnerable to router crashes.
  - The changes required to the router code are substantial and involve complex router-to-router exchanges for setting up the flows.
- Few implementations of RSVP or anything like it exist yet.

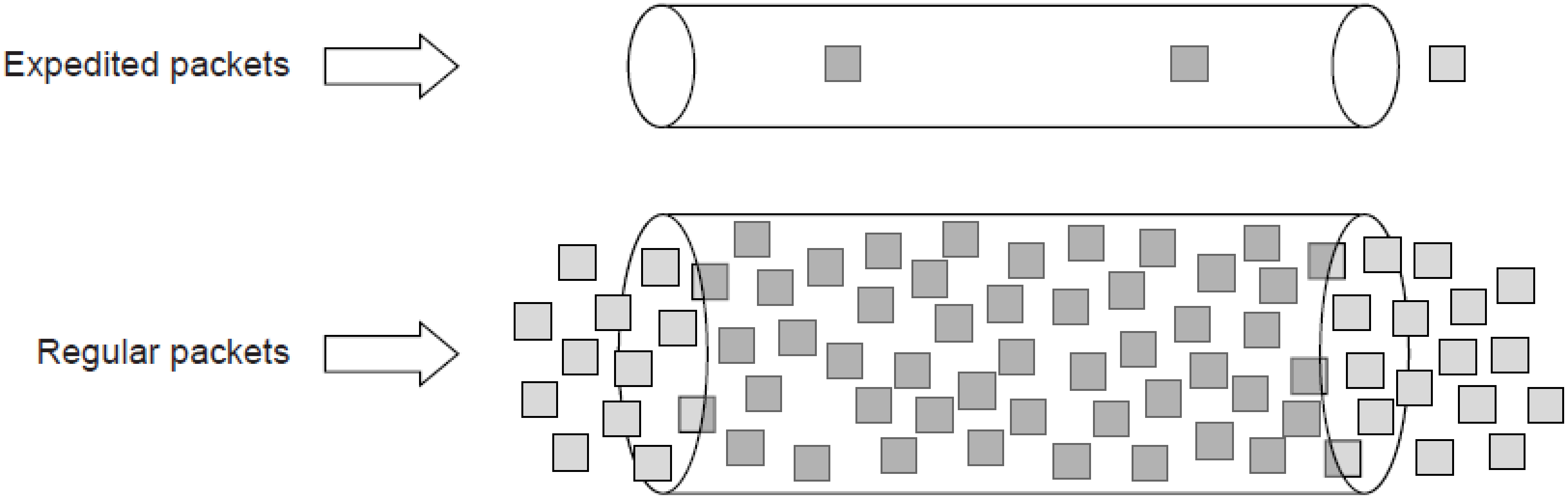
Differentiated Services

- **Differentiated services** (DS), in RFC 2474 and 2475
  - A simpler approach to QoS than RSVP.
  - It is known as **class-based** (as opposed to flow-based) quality of service.
    - It is offered by a set of routers forming an administrative domain.
    - The administration defines a set of service classes with corresponding forwarding rules.
    - If a customer signs up for DS, customer packets entering the domain are marked with the class to which they belong.
    - Traffic within a class may be required to conform to some specific shape, such as a leaky bucket with some specified drain rate.
- This scheme requires no advance setup, no resource reservation, and no time-consuming end-to-end negotiation for each flow.

Cont.

- The classes may differ in terms of delay, jitter, and probability of being discarded in the event of congestion.
- The difference between flow-based QoS and class-based QoS:
  - With a flow based scheme, each flow gets its own resources and guarantees.
  - With a class-based scheme, all streams together get the resources reserved for their class. The resources can not be taken away by packets from file transfer class or other classes, but no stream gets any private resources reserved for it alone.

Differentiated Services (1)

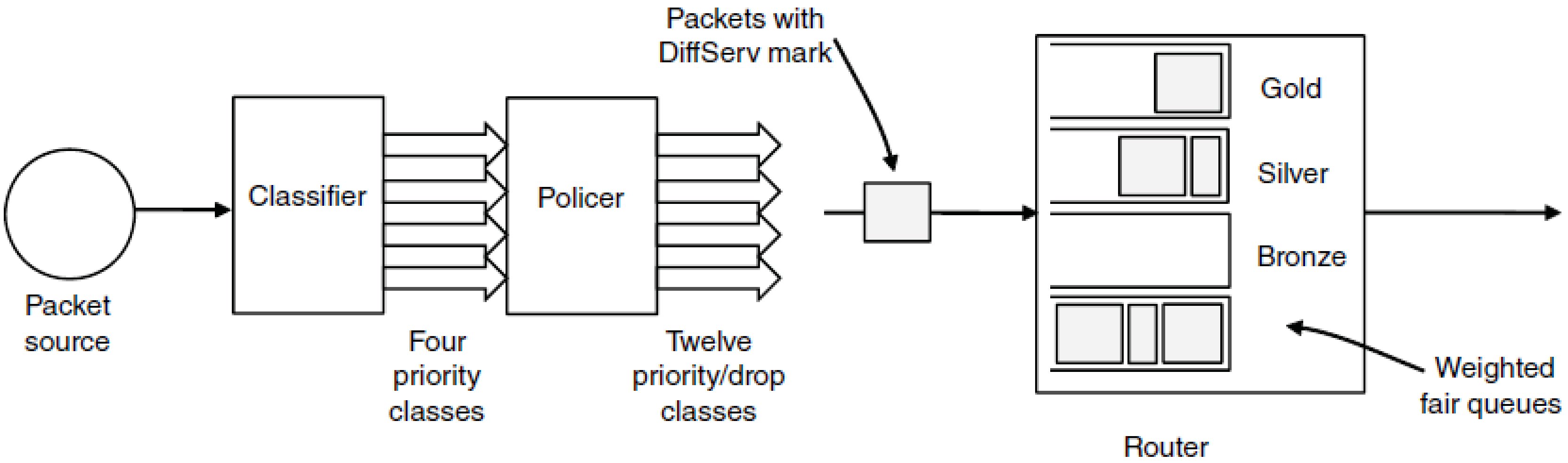


Expedited packets experience a traffic-free network

Cont.

- **Expedited forwarding** , in RFC 3246,
  - One of network-independent services class. (The choice of service classes is up to each operator.)
  - Two classes of service are available: regular and expedited.
    - The vast majority of the traffic is regular, but a small fraction of the packets are expedited.
    - They reserves bandwidth in a way that two logical pipes run through one physical line.
    - One way to implement this strategy is to program the routers to have two output queues for each outgoing line, one for expedited packets and one for regular packets.
    - Packet scheduling should use something like *priority queue* or *weighted fair queue*.

Differentiated Services (2)



A possible implementation of assured forwarding

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

Cont.

- **Assured forwarding**, in RFC 2597,
  - It specifies that there should be four priority classes, each class having its own resources.
  - It also defines three discard probabilities for packets that are experiencing congestion: low, medium, and high.
  - Taken together, it defines 12 service classes.
- Step 1: Classify packets into one of the four priority classes.
  - This step might be done on the sending host or in the ingress (first) router.

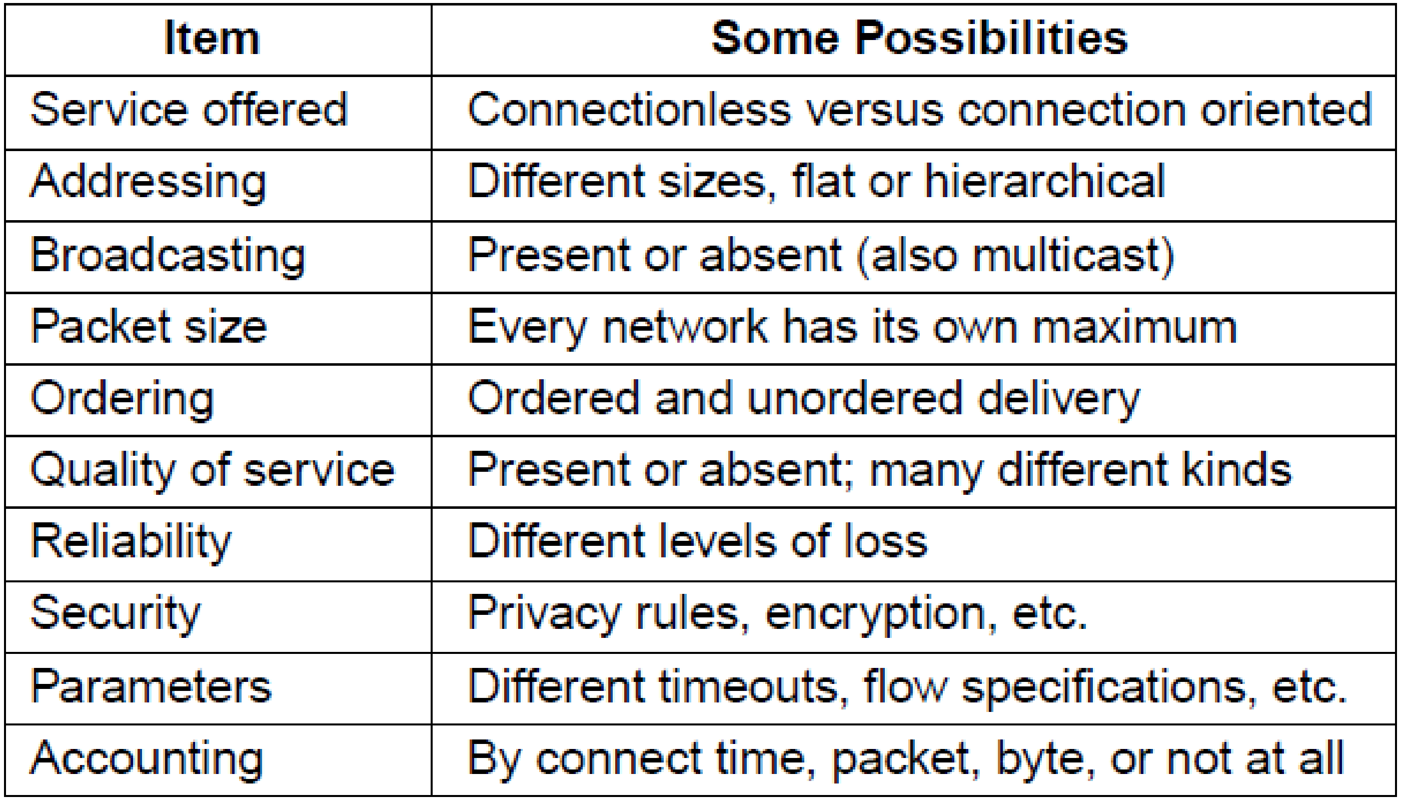
Cont.

- Step 2: Determine the discard class for each packet, by passing the packets of each priority class through a *traffic policer* such as a token bucket.
- Step 3: The packets are processed by routers in the network with a *packet scheduler* that distinguishes the different classes.

Internetworking

- How networks differ
- How networks can be connected
- Tunneling
- Internetwork routing
- Packet fragmentation

How Networks Differ



Item	Some Possibilities
Service offered	Connectionless versus connection oriented
Addressing	Different sizes, flat or hierarchical
Broadcasting	Present or absent (also multicast)
Packet size	Every network has its own maximum
Ordering	Ordered and unordered delivery
Quality of service	Present or absent; many different kinds
Reliability	Different levels of loss
Security	Privacy rules, encryption, etc.
Parameters	Different timeouts, flow specifications, etc.
Accounting	By connect time, packet, byte, or not at all

Some of the many ways networks can differ

Cont.

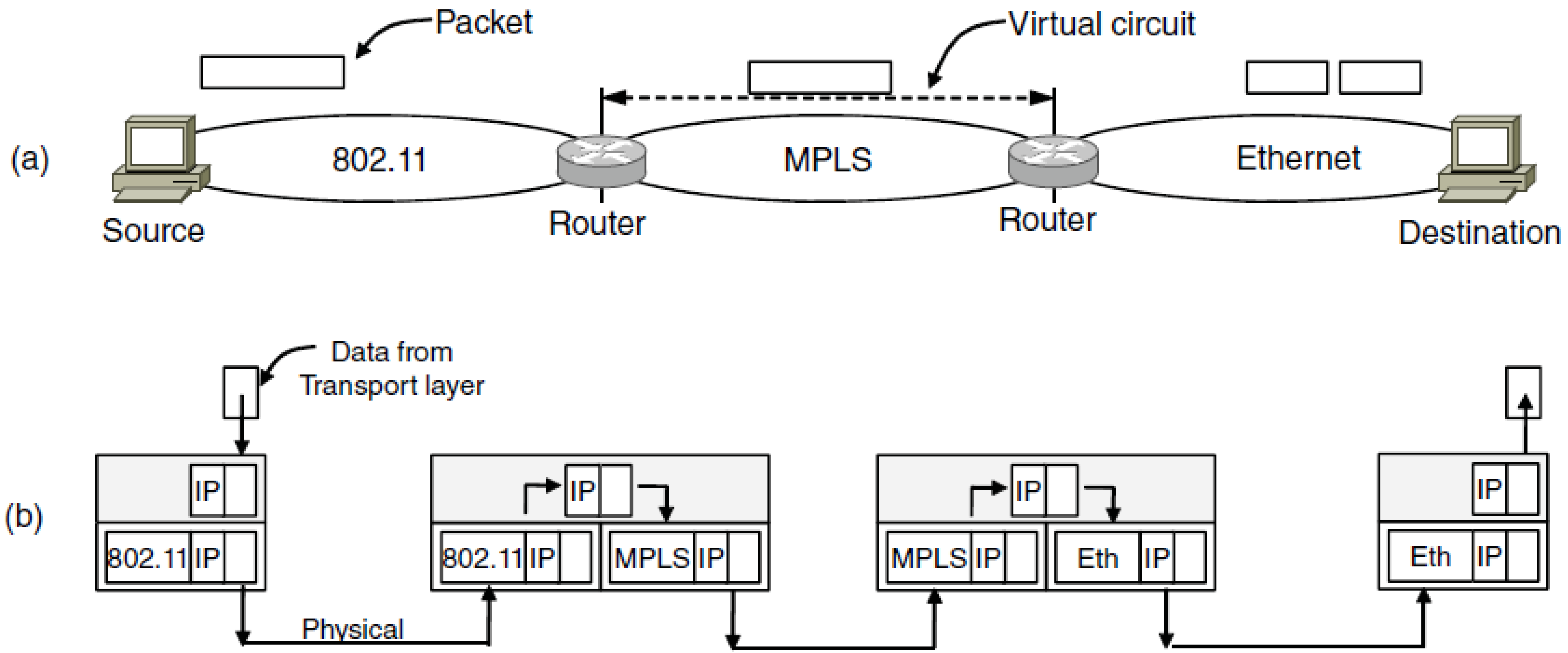
- Different devices for network interconnections:
- In the physical layer (Layer 1):
  - Repeaters, or hubs, which move *bits* from one network to an identical network.
- In the data link layer (Layer 2):
  - Bridges and switches, which accept *frames*, examine the MAC address, and forward the frames to a different network.
- In the network layer (Layer 3):
  - Routers, which connect two networks.
    - If two networks have dissimilar network layers, the router may be able to translate between the *packet* formats. Packet translation is increasingly rare.
    - A router that can handle multiple protocols is called a **multiprotocol router**.

Cont.

- In the transport layer (Layer 4):
  - Transport gateways, which can interface between two transport connections (by translating layer 4 *segments*).
- In the application layer (Layer 5):
  - Application gateways, which translate *message* semantics.

- An essential difference between the switched (bridged) case and the routed case:
  - With a switch (or bridge), the entire frame is transported on the basis of its MAC address.
  - With a router, the packet is extracted from the frame and the address in the packet is used for deciding where to send it.

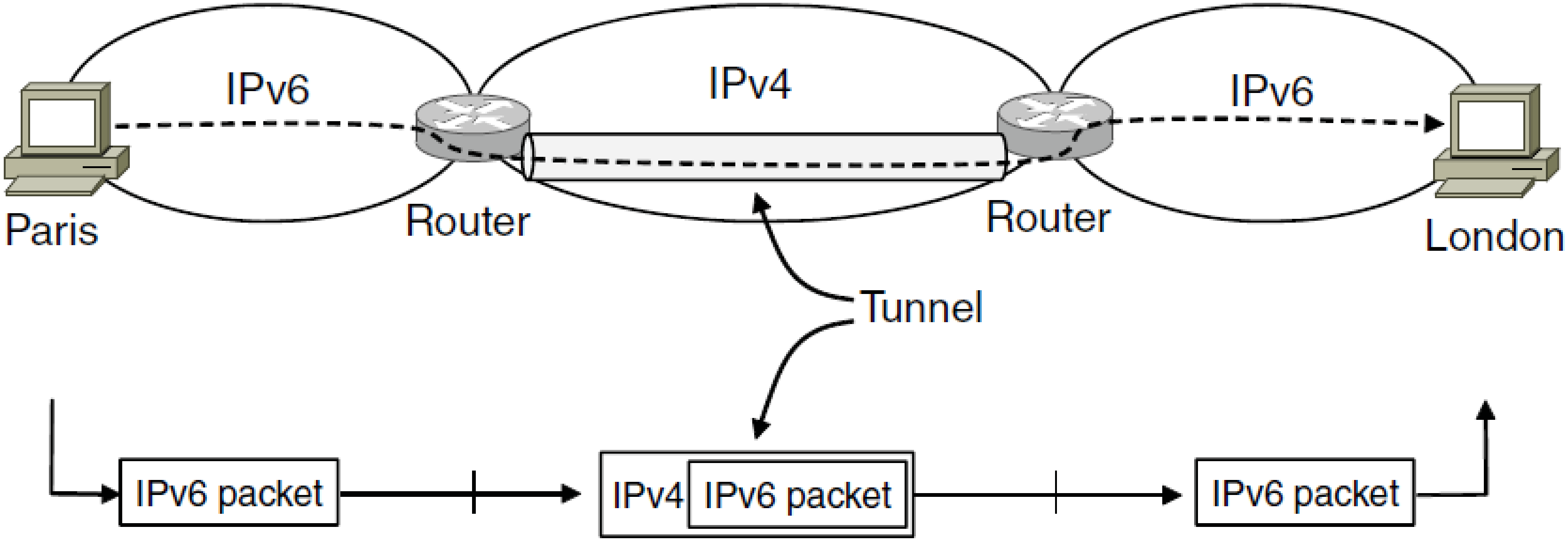
How Networks Can Be Connected



(a) A packet crossing different networks.

(b) Network and link layer protocol processing.

Tunneling (1)



Tunneling a packet from Paris to London.

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

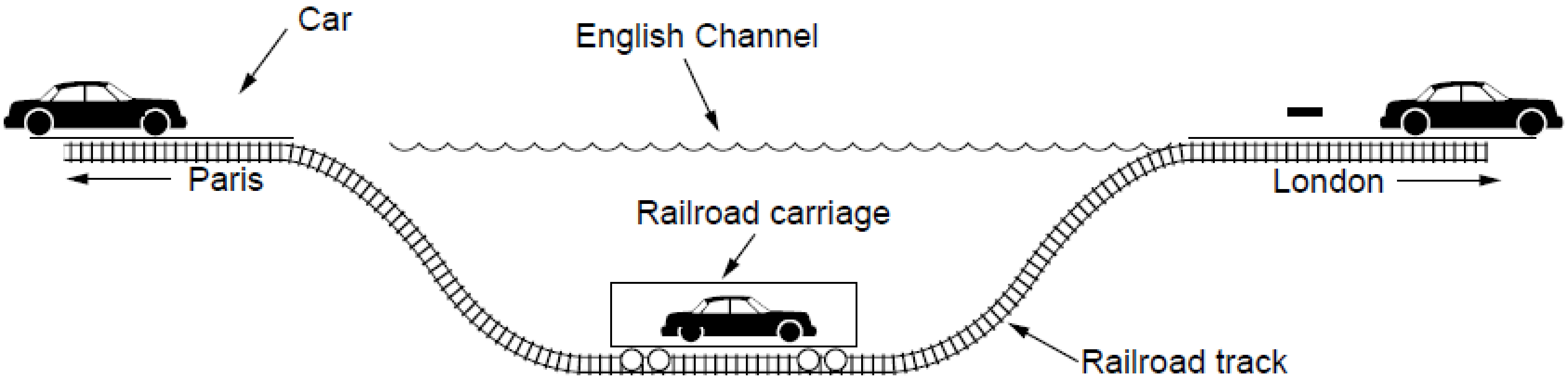
Tunneling

- It is very difficult to make two different networks interwork. However, there is a common special case:
  - *The source and destination hosts are on the same type of network, but there is a different network in between*.
  - An example is shown in the figures above, where the technique to the problem is called **tunneling**.
    - To send an IP packet to a host in London, a host in Paris office constructs an IPv6 packet containing an IPv6 address in London and sends it to the multiprotocol router that connects the Paris IPv6 network to the IPv4 Internet.

Cont.

  - When the multiprotocol router gets the IPv6 packet, *it encapsulates the packet with an IPv4 header addressed to the IPv4 side of the multiprotocol router that connects to the London IPv6 network.*
  - When the packet gets there, the London router removes the IPv6 packet and sends it onward to the destination host.
- **Overlay**:
  - The above network is an example of overlay “ IPv6 over IPv4”
- *Disadvantage* of tunneling:
  - None of the hosts on the network that is tunneled over can be reached.
- This is turned into an *advantage* with VPNs (Virtual Private Networks)

Tunneling (2)



Tunneling a car from France to England

Internetworking

- A two-level routing algorithm
  - Intradomain routing, interior gateway protocol (IGP)
    - Since each network is operated independently of all the others, it is often referred as an **Autonomous System** (AS).
      - E.g., an ISP network
  - Interdomain routing, exterior gateway protocol (EGP)
    - In the Internet, the protocol is called BGP (Border gateway protocol.
    - Constrained to some *routing policy* that governs the way autonomous networks select the routes.

Packet Fragmentation (1)

Packet size issues:

1. Hardware
2. Operating system
3. Protocols
4. Compliance with (inter)national standard.
5. Reduce error-induced retransmissions
6. Prevent packet occupying channel too long.

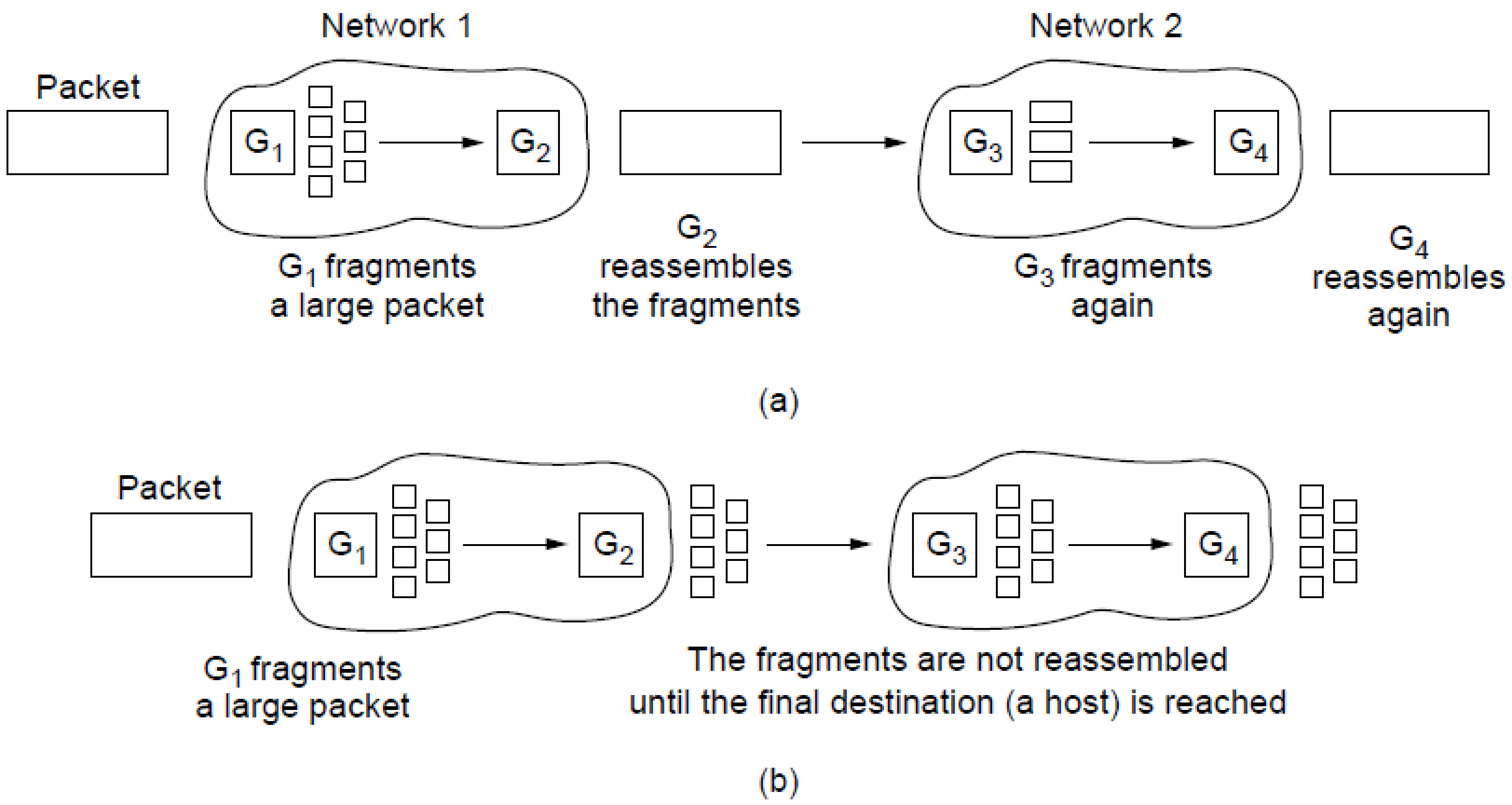
Cont.

- To reduce overheads, large packets are preferred for transmission.
- Problems: When a large packet wants to travel through network whose maximum packet size is to small.
- Solutions:

1. Find **Path MTU** (Path maximum Transmission Unit), to make sure the problem does not occur.
2. Allow routers to break a packet **into fragments** and send each fragment as a separate internet packet.
   - Two opposing strategies for recombining the fragments back into the original packet:
     1. Transparent fragmentation
     2. Nontransparent fragmentation.

- Transparent fragmentation:
  - Make fragmentation caused by a small-packet network *transparent* to any subsequent networks through which the packet must pass on its way to the destination.
- Nontransparent fragmentation:
  - Refrain from recombining fragments at any intermediate gateways. Once a packet has been fragmented, each fragment is treated as though it were an original packet.
  - Recombination occurs only at the destination host.

Packet Fragmentation (2)


Network 1
Network 2
Packet
G_1
G_2
G_3
G_4
G_1 fragments a large packet
G_2 reassembles the fragments
G_3 fragments again
G_4 reassembles again
(a)
Packet
G_1
G_2
G_3
G_4
G_1 fragments a large packet
The fragments are not reassembled until the final destination (a host) is reached
(b)

(a) Transparent fragmentation.

(b) Nontransparent fragmentation

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

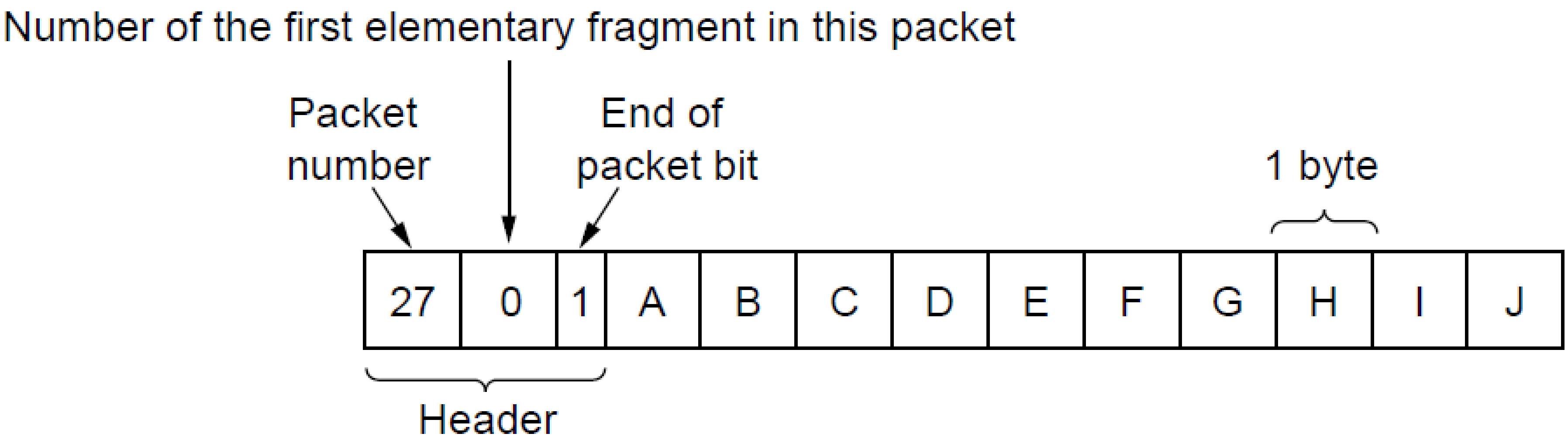
Cont.

- Problems with *transparent* fragmentation:
  - The exit gateway must know when it has received all the pieces.
  - All packets must exit via the same gateway.
  - The overhead required to repeatedly reassemble and then refragment a large packet passing through a series of small-packet networks.
- Problems with *nontransparent* fragmentation:
  - It requires *every* host to be able to do reassembly.
  - When a large packet is fragmented the total overhead increases.
    - Multiple exit gateways can be used and higher performance can be achieved.

Cont.

- When a packet is fragmented, the fragments must be numbered in such a way that the original data stream can be reconstructed.
  - Define an elementary fragment size small enough that the elementary fragment can pass through every network.
  - When a packet is fragmented, all the pieces are equal to the elementary fragment size except the last one, which may be shorter.
  - An internet packet may contain several fragments.
  - The internet header must provide the original packet number, and the number of the (first) elementary fragment contained in the packet.
  - There must also be a bit indicating that the last elementary fragment contained within the internet packet is the last one of the original packet.
  - This approach requires three fields in the Internet header:
    - The *original packet number*,
    - The *fragment number, as an* offset number
    - A *flag*, indicating whether it is the end of the packet.

Packet Fragmentation (3)



Number of the first elementary fragment in this packet

Packet number

End of packet bit

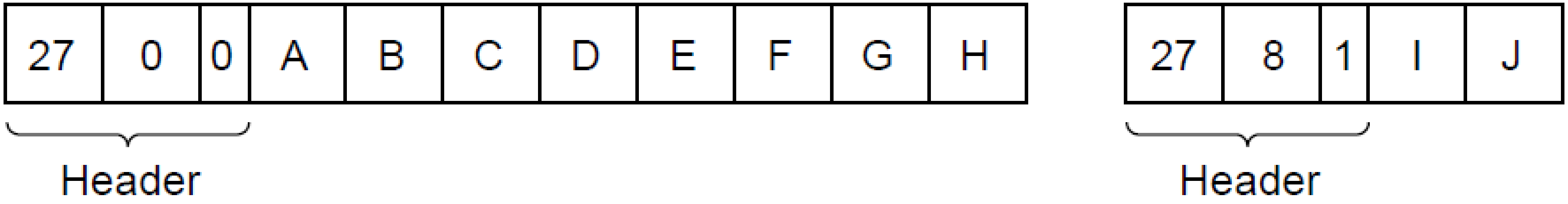
1 byte

27	0	1	A	B	C	D	E	F	G	H	I	J

Header

Fragmentation when the elementary data size is 1 byte.
(a) Original packet, containing 10 data bytes.

Packet Fragmentation (4)



27	0	0	A	B	C	D	E	F	G	H

Header

27	8	1	I	J

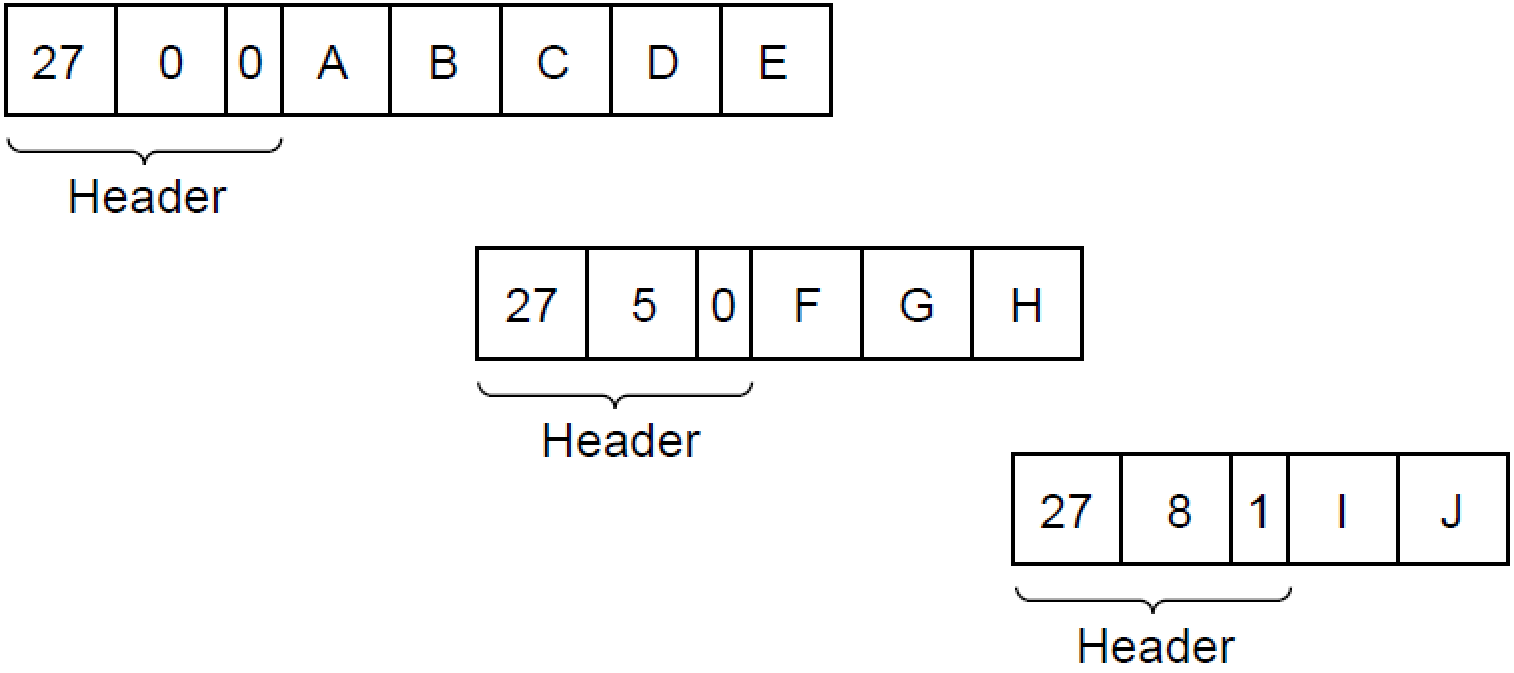
Header

Fragmentation when the elementary data size is 1 byte

(b) Fragments after passing through a network

with maximum packet size of 8 payload bytes plus header.

Packet Fragmentation (5)



27 | 0 | 0 | A | B | C | D | E

Header

27 | 5 | 0 | F | G | H

Header

27 | 8 | 1 | I | J

Header

Fragmentation when the elementary data size is 1 byte

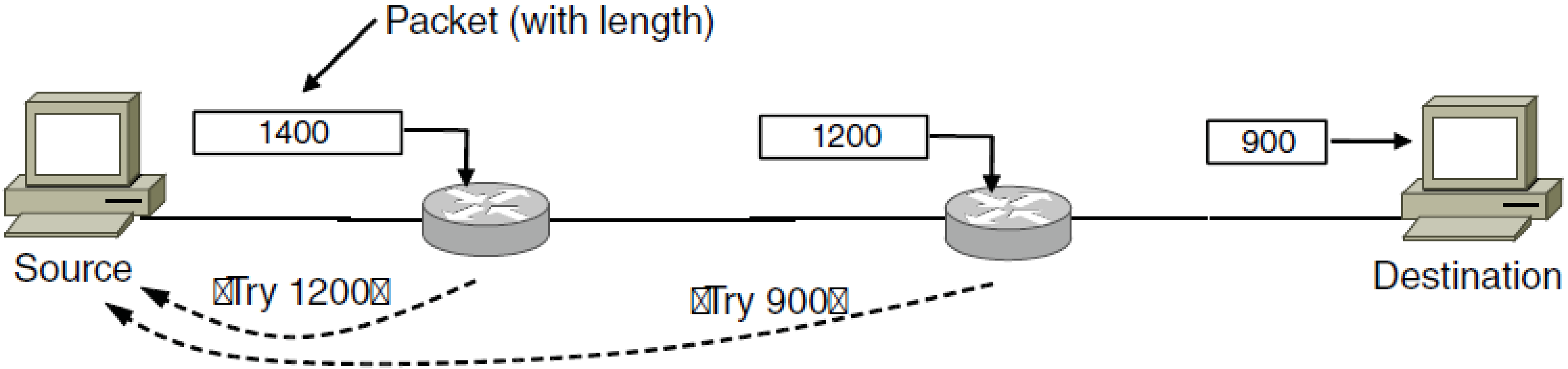
(c) Fragments after passing through a size 5 gateway.

Cont.

- Path MTU discovery:
  - Each IP packet is sent with its header bits set to indicate no fragmentation is allowed to be performed.
  - If a router receives a packet that is too large, it generates an error packet, returns it to the source, and drops the packet.
  - When the source receives the error packet, it uses the information inside to refragment the packet into pieces that are small enough for the router to handle.

Fragmentation is still needed between the source and destination.

Packet Fragmentation (6)



Path MTU Discovery

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

The Network Layer Principles (1)

1. Make sure it works
2. Keep it simple
3. Make clear choices
4. Exploit modularity
5. Expect heterogeneity

   . . .

The Network Layer Principles (2)

. . .

6. Avoid static options and parameters
7. Look for good design (not perfect)
8. Strict sending, tolerant receiving
9. Think about scalability
10. Consider performance and cost

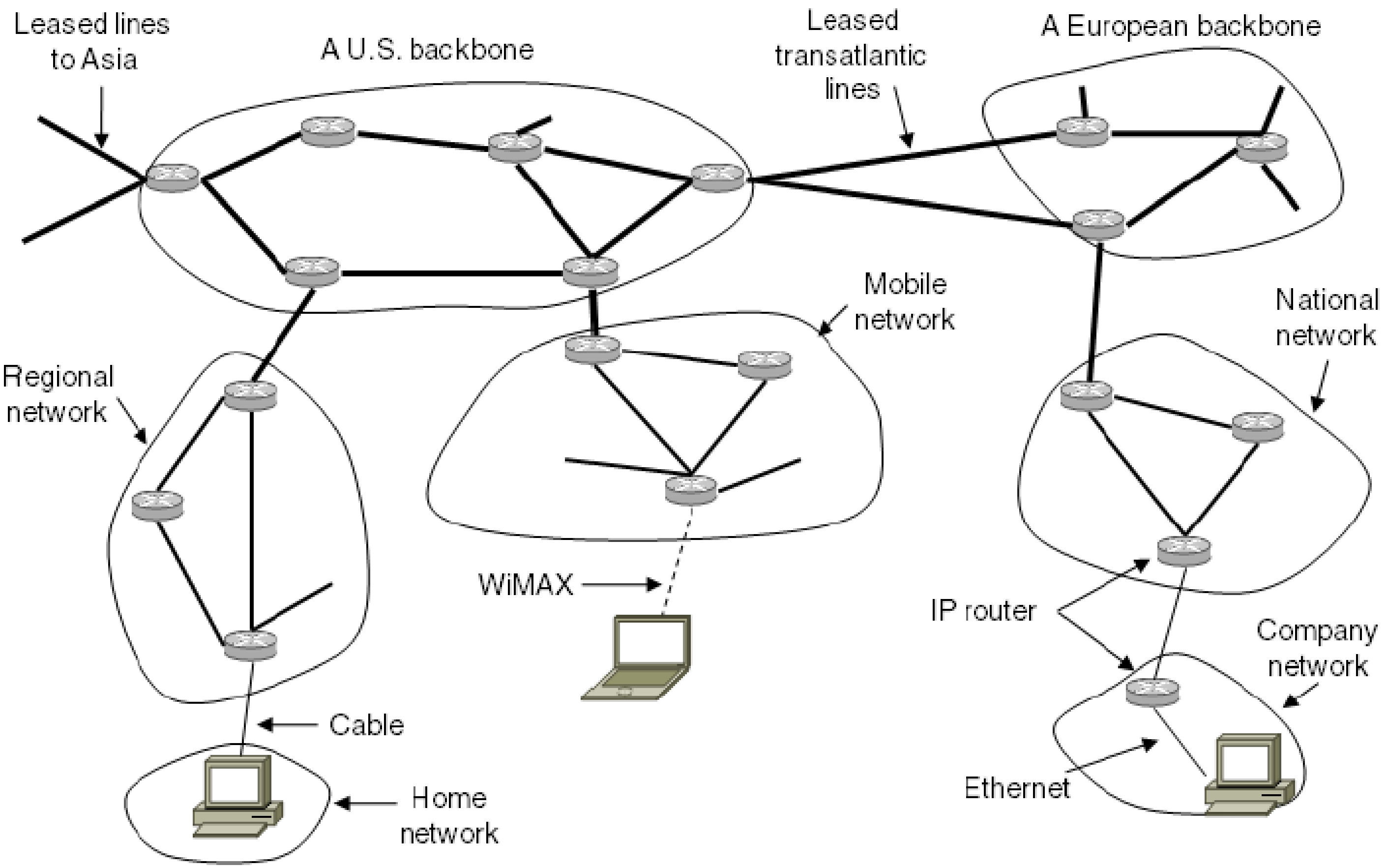
The Network Layer in the Internet (1)

- The IP Version 4 Protocol
- IP Addresses
- IP Version 6
- Internet Control Protocols
- Label Switching and MPLS
- OSPF—An Interior Gateway Routing Protocol
- BGP—The Exterior Gateway Routing Protocol
- Internet Multicasting
- Mobile IP

The Internet

- The Internet can be viewed as a collection of networks or ASes that are interconnected.
  - Tier 1 networks: the biggest of the backbones
- IP is the network layer protocol for internetworking
  - Best-effort service

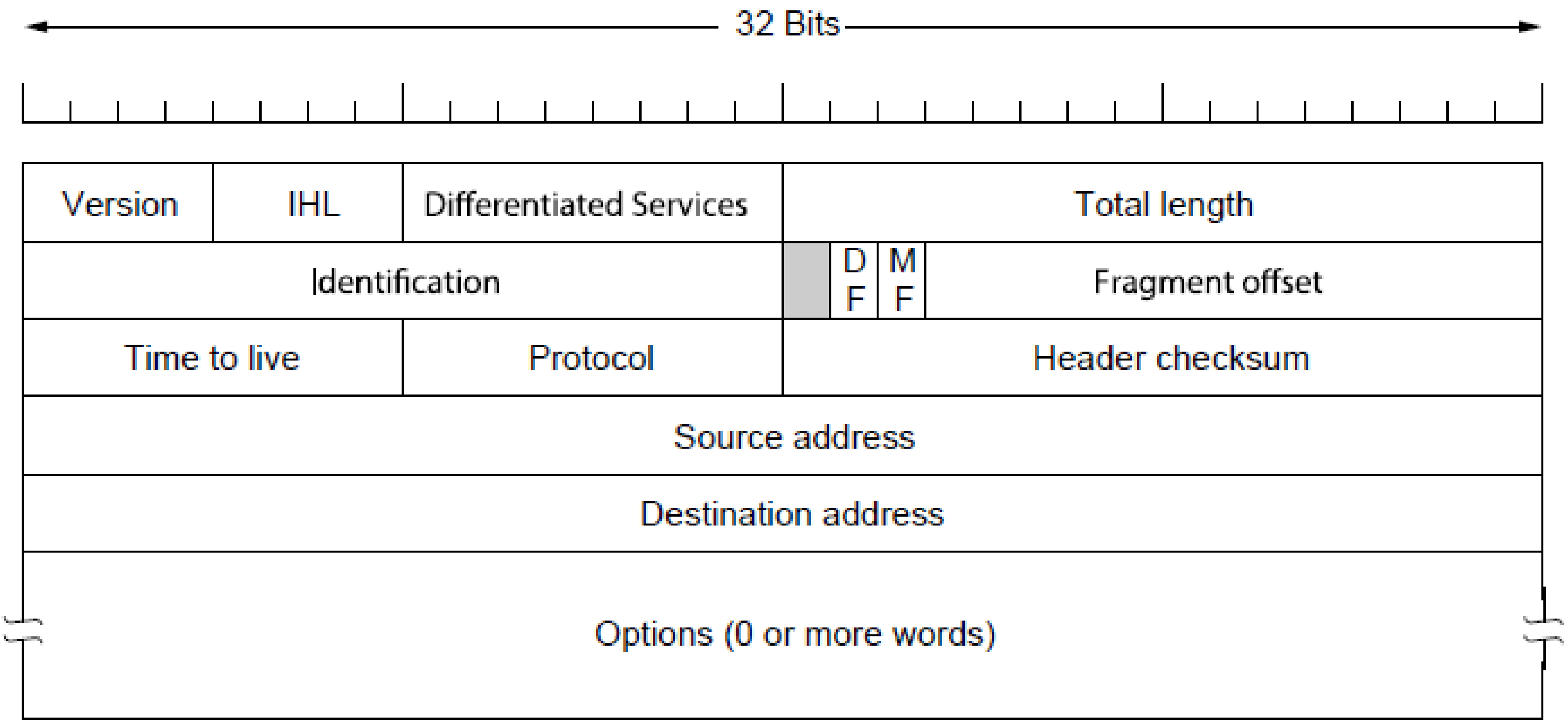
The Network Layer in the Internet (2)



The Internet is an interconnected collection of many networks.

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

The IP Version 4 Protocol (1)



The IPv4 (Internet Protocol) header.

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

Cont.

- An IP datagram consists of a header part and a body or *payload* part.
- The header has a 20-byte fixed part and a variable length optional part.
    - It is transmitted in big endian: from left to right, with the high-order bit of the *Version* field going first.
  - The *Version* field keeps track of which version of the protocol the datagram belongs to.
  - The IHL field tells how long the header is, in 32-bit words.
    - The minimum value is 5 when no options are present.
    - The maximum value is 15, which limits the header to 60 bytes, and thus the options field to 40 bytes.

Cont.

- The *Differential services*(called *Type of Service* originally) field was and is still intended to distinguish between different classes of services. The remaining are 2 unused bits.
  - Originally the 6-bit field contained, from left to right, a three-bit *Precedence* field and three flags, *D*,*T* and *R*.
    - The three flag bits allow the host to specify what it cares most about from the set {Delay, Throughput, Reliability}.
    - In practice, current routers ignore the *Type of Service* field altogether.
  - IETF changes the field slightly to accommodate differentiated services.
    - The top 6 bits are used to indicate which of service classes each packet belongs to. These classes include four queueing priorities, three discard probabilities, and the historical classes.
    - The bottom 2 bits are used to carry explicit congestion notification (ECN).
- The *Total length* includes everything in the datagram-both header and data, with a maximum length 65535 bytes.

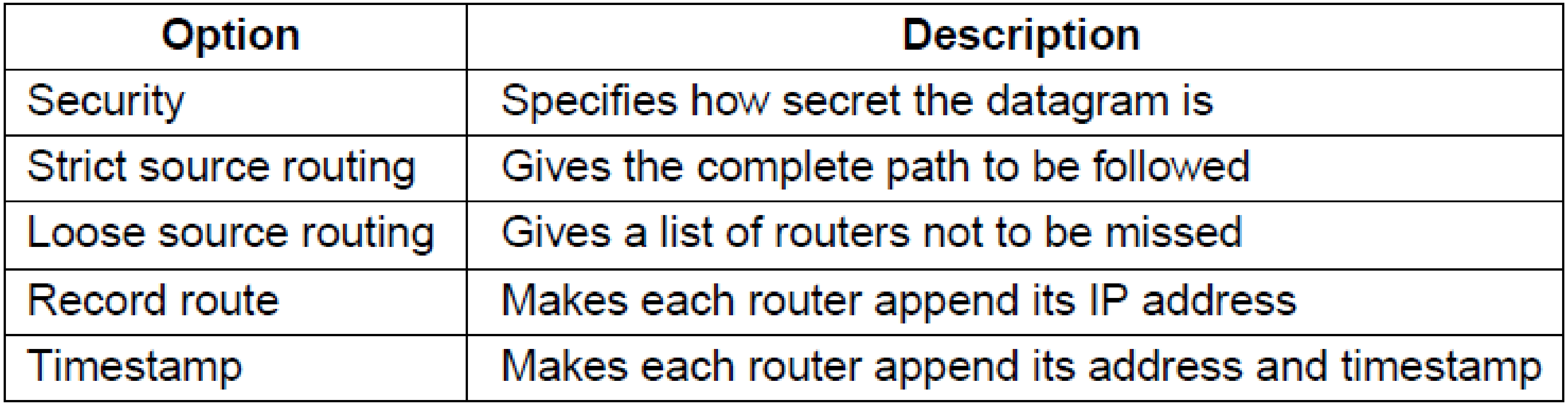
Cont.

- The *Identification* field is needed to allow the destination host to determine which datagram a newly arrived fragment belongs to. All the fragments of a datagram contain the same ID value.
- An unused bit and two 1-bit fields.
  - DF stands for ``Don’t fragment.”
  - MF stands for ``More fragments.” All fragments except the last one have this bit set.
- The *Fragment offset* tells where in the current datagram this fragment belongs.
  - All fragments except the last one in a datagram must be a multiple of 8 bytes, the elementary fragment unit.
  - Since 13 bits are provided, there is a maximum of 8192 fragments per datagram, giving a maximum datagram length 65536 bytes, one more than the *Total length* field.

Cont.

- The *Time to live(TtL)* field is a counter used to limit packet lifetimes.
  - It is supposed to count time in seconds.
  - In practice, it just counts hops. When it hits zero, the packet is discarded and a warning packet is sent back to the source host.
- The *Protocol* field tells the network layer which transport process to give it to.
  - TCP, UDP, and some others. (RFC 1700, www.iana.org)
- The *Header checksum* verifies the header only.
  - It must be recomputed at each hop, because at least one field always changes (the *Time to live* field).
- The *Source address* and *Destination address* indicate the network number and host number.

The IP Version 4 Protocol (2)



Option	Description
Security	Specifies how secret the datagram is
Strict source routing	Gives the complete path to be followed
Loose source routing	Gives a list of routers not to be missed
Record route	Makes each router append its IP address
Timestamp	Makes each router append its address and timestamp

Some of the IP options.

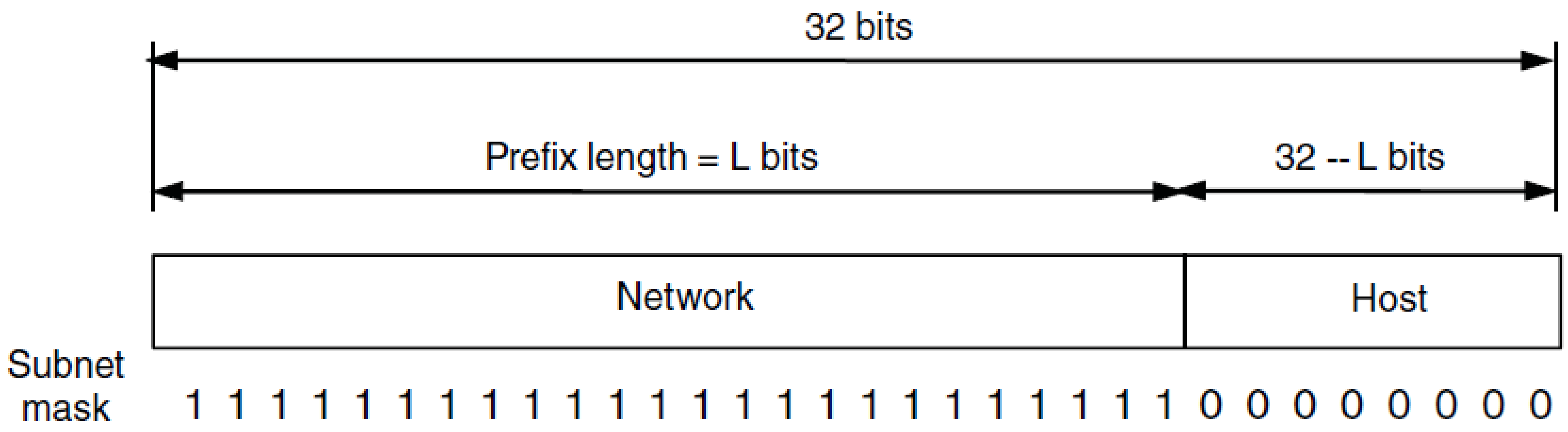
Cont.

- 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.
  - It has 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 four bytes.
  - Originally, five options were defined but subsequently new ones have been added.

Cont.

- The *Security* option tells how secret the information is.
  - In theory, a military router might use this field to specify not to route through certain countries.
  - In practice, all routers ignore it.
- The *Strict source routing* option gives the complete path from source to destination as a sequence of IP addresses.
- The *Loose source routing* option requires the packet to traverse the list of routers specified, and in the order specified, but it is allowed to pass through other routers on the way.
- The *Record route* option tells the routers along the path to append their IP address to the option field.
- The *Timestamp* option is like the *Record route* option, except that in addition to recording its 32-bit IP address, each router also records a 32-bit time-stamp.

IP Addresses (1)



An IP prefix.

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

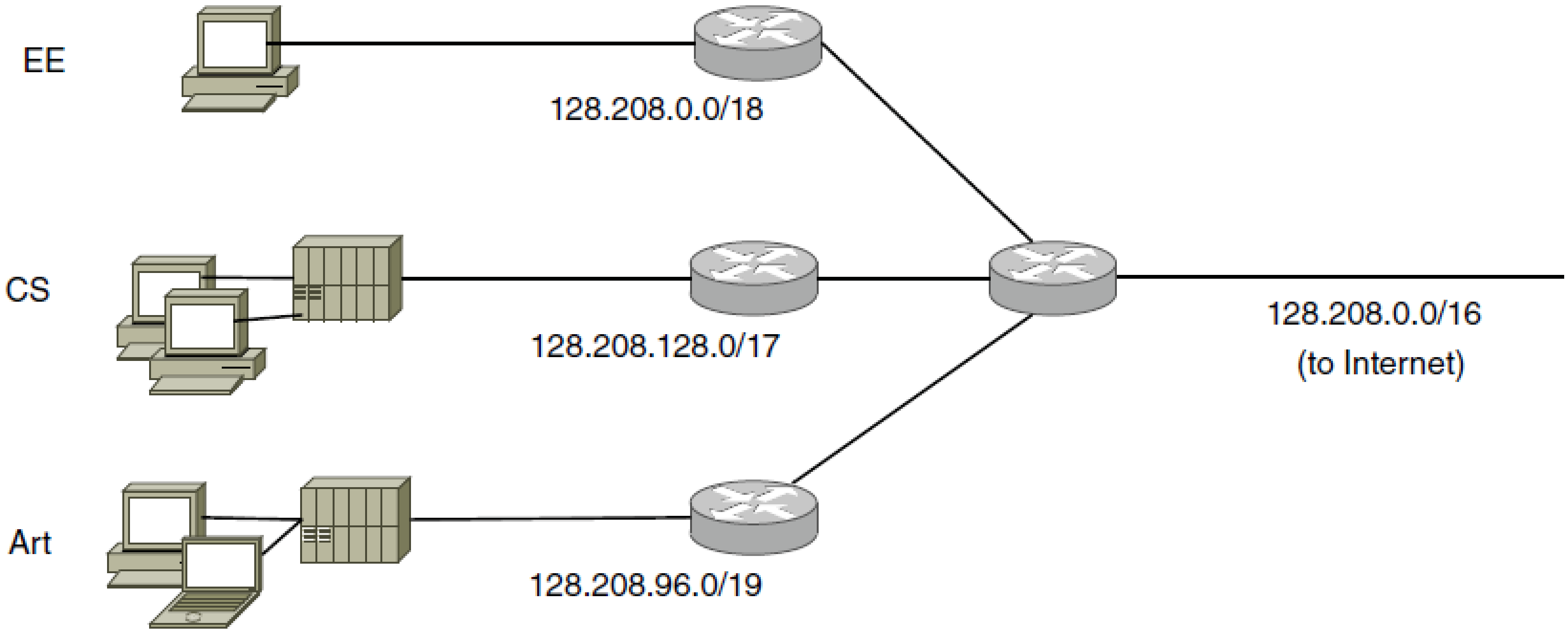
IP addresses

- Hierarchical addresses, unlike Ethernet address
- Written in dotted decimal notation, like 140.122.185.141
- Prefixes are written by given the lowest IP address in the block and the size of the block.
  - The size is determined by the number of bits in the network portion.
  - E.g., 128.208.0.0/24
- **Subnet mask**
  - The length of the prefix corresponds to a binary mask of 1s in the network portion.
- Advantage: Routers can forward packets based only the network portion of the addresses.
- Disadvantage:
  - The IP address of a host depends on *where* it is located in the network.
  - Wasteful of addresses.

Subnet

- Split a block of addresses into several parts for internal use as multiple networks.
  - E.g.
    - Computer Science (a/17): 10000000 11010000 1|xxxxxxx xxxxxxxx
    - Electrical Eng.(a/18) : 10000000 11010000 00|xxxxxx xxxxxxxx
    - Art: (a/19) : 10000000 11010000 011|xxxxx xxxxxxxx

IP Addresses (2)



Splitting an IP prefix into separate networks with subnetting.

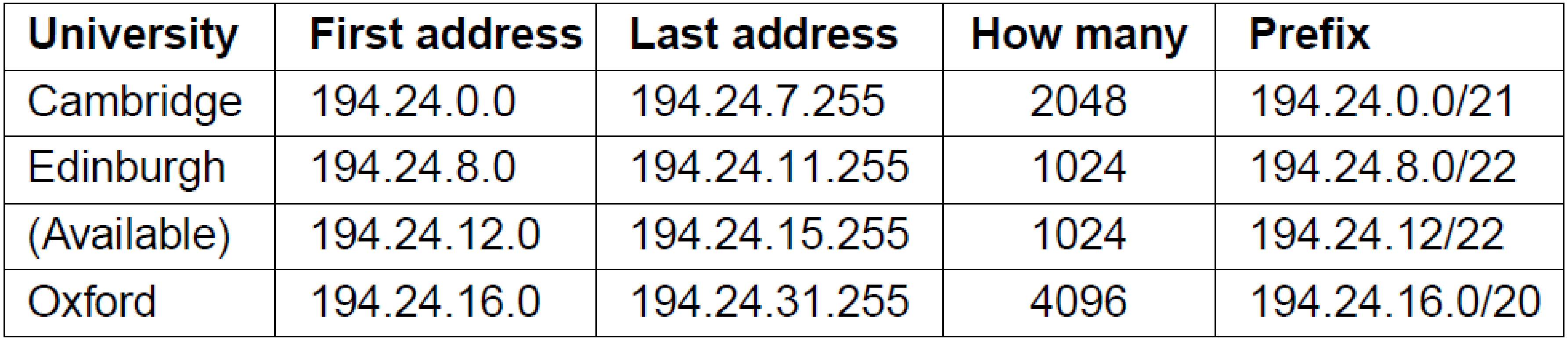
Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

CIDR – Classless InterDomain Routing

- A single routing table for all networks consisting of an array of *(IP address, subnet mask, outgoing line)* triples.
- When a packet comes in, its destination IP address is first extracted.
- The routing table is scanned entry by entry, masking the destination address and comparing it to the table entry looking for a match.
  - It is possible that multiple entries with different subnet mask lengths match, in which case the longest mask (the **longest matching prefix** or the most specific route) is used.

- Routing table explosion issues, particularly for routers in the ISPs and backbones (in the **default-free** zone)
  - Solution: **Route aggregation** by combining multiple small prefixes into a single larger prefix.
    - The resulting larger prefix is sometimes called a **supernet**.
    - **Aggregation** is heavily used throughout the Internet to reduce the size of routing tables.

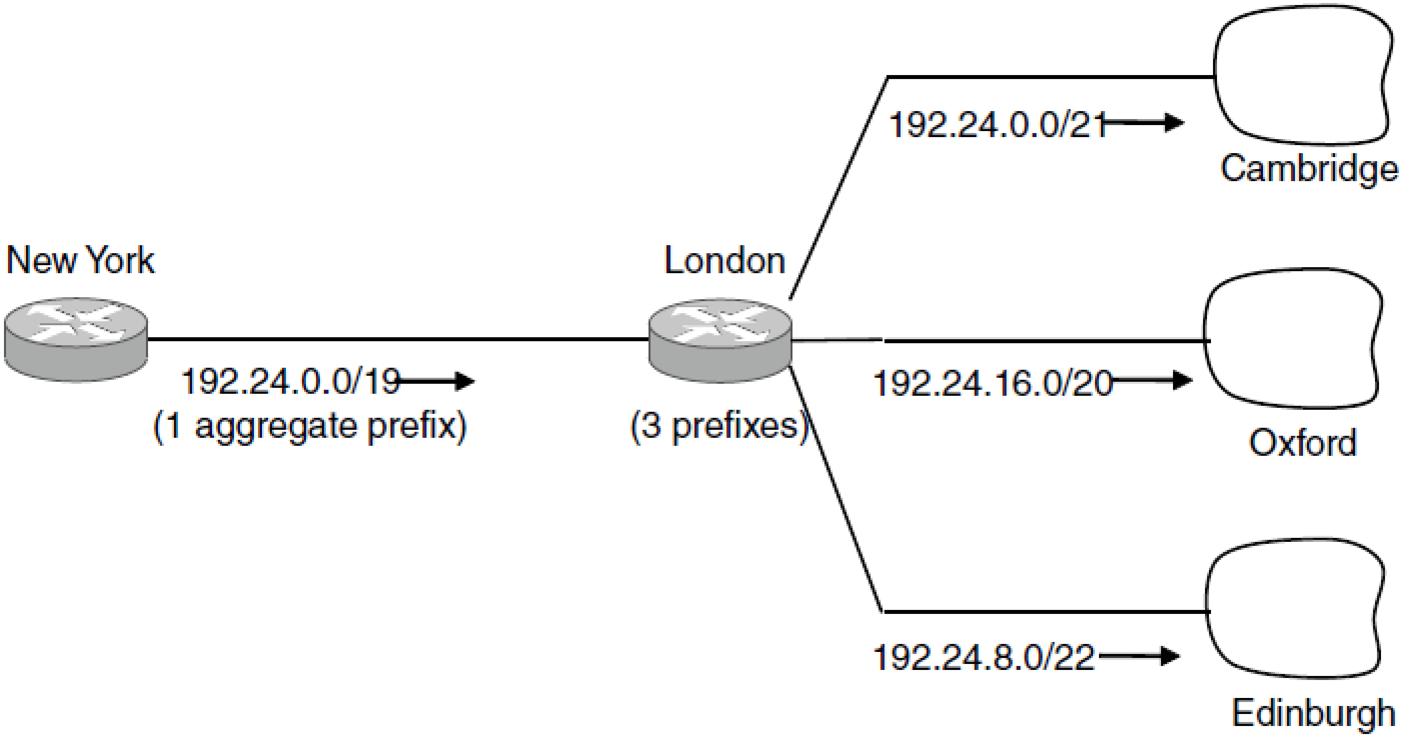
IP Addresses (3)



University	First address	Last address	How many	Prefix
Cambridge	194.24.0.0	194.24.7.255	2048	194.24.0.0/21
Edinburgh	194.24.8.0	194.24.11.255	1024	194.24.8.0/22
(Available)	194.24.12.0	194.24.15.255	1024	194.24.12/22
Oxford	194.24.16.0	194.24.31.255	4096	194.24.16.0/20

A set of IP address assignments

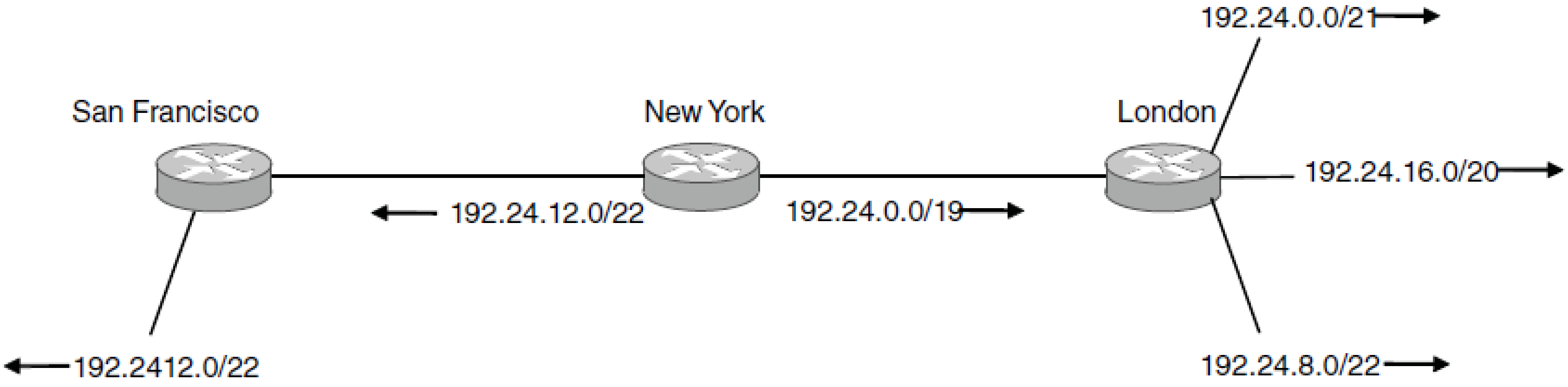
IP Addresses (4)



Aggregation of IP prefixes

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

IP Addresses (5)

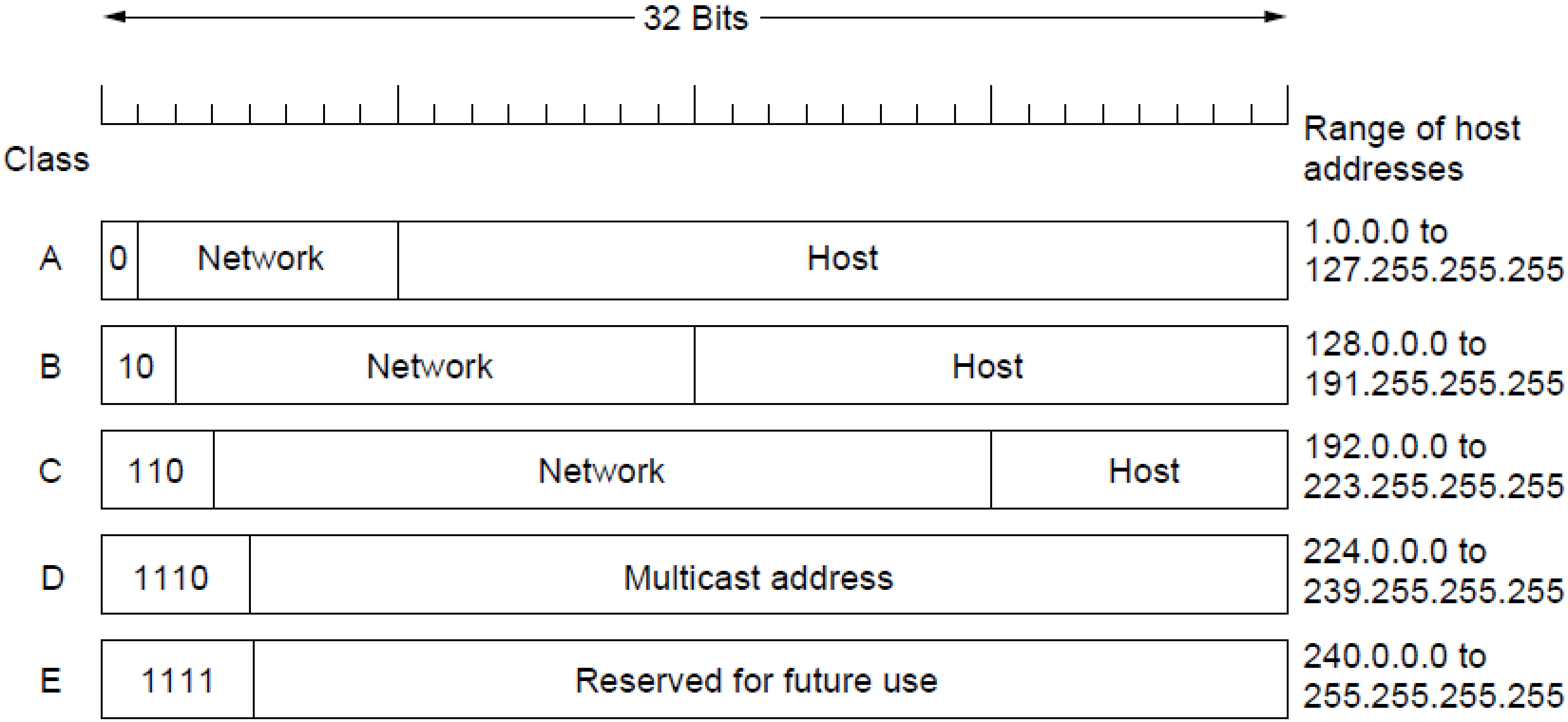


Longest matching prefix routing at the New York router.

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

Classful and Special addressing

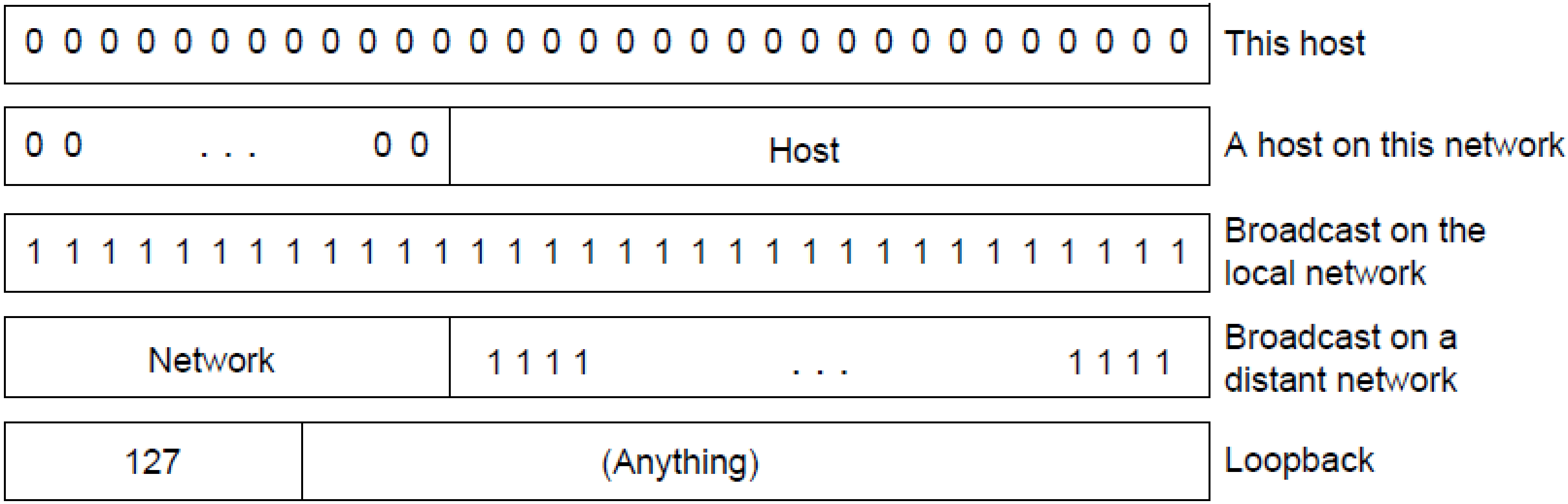
IP Addresses (6)



Class	32 Bits			Range of host addresses
A	0	Network	Host	1.0.0.0 to 127.255.255.255
B	10	Network	Host	128.0.0.0 to 191.255.255.255
C	110	Network	Host	192.0.0.0 to 223.255.255.255
D	1110	Multicast address		224.0.0.0 to 239.255.255.255
E	1111	Reserved for future use		240.0.0.0 to 255.255.255.255

IP address formats

IP Addresses (7)



0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		This host
0 0 . . . 0 0	Host	A host on this network
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		Broadcast on the local network
Network	1 1 1 1 . . . 1 1 1 1	Broadcast on a distant network
127	(Anything)	Loopback

Special IP addresses

IP addresses

- To resolve issues on IPv4 address shortage:
  - Dynamic assignment
    - Feasible for dialup networking and mobile terminals
    - Impractical for business customers
  - NAT (Network Address Translation, described in RFC 3022) and private IP address
  - Replace by IPv6

NAT

- Assign each home or company a single IP address (or at most, a small number of them) for Internet traffic.
- Within the company, every computer gets a unique IP address, which is used for routing intramural traffic.
- When a packet exits the company and goes to the ISP, an address translation takes place.
- Three ranges of IP addresses have been declared as private.
    - Companies may use them internally as they wish. The only rule is that no packets containing these addresses may appear on the Internet itself.
    - The three ranges are:

10.0.0.0	–	10.255.255.255/8	(16,777,216 hosts)
172.16.0.0	–	172.31.255.255/12	(1,048,576 hosts)
192.168.0.0	–	192.168.255.255/16	(65,536 hosts)

Cont.

- Within the company premises, every machine has a unique address of the form 10.*x*.*y*.*z*. When a packet leaves the company premises, it passes through a NAT box that converts the internal IP source address to the company's true IP address.
  - The NAT is often combined in a single device with a firewall.
  - NAT makes use of the TCP or UDP *source port* and *destination port* (16 bits + 16 bits).
    - Whenever an outgoing packet enters the NAT box, the 10.*x*.*y*.*z* source address is replaced by the company's true IP address.
    - In addition, the TCP (or UDP) *source port* field is replaced by an index into the NAT box's 65536-entry translation table.
      - This table contains the original IP address and the original source port.
    - Both the IP and TCP (or UDP) header checksums are recomputed and inserted into the packet.

Cont.

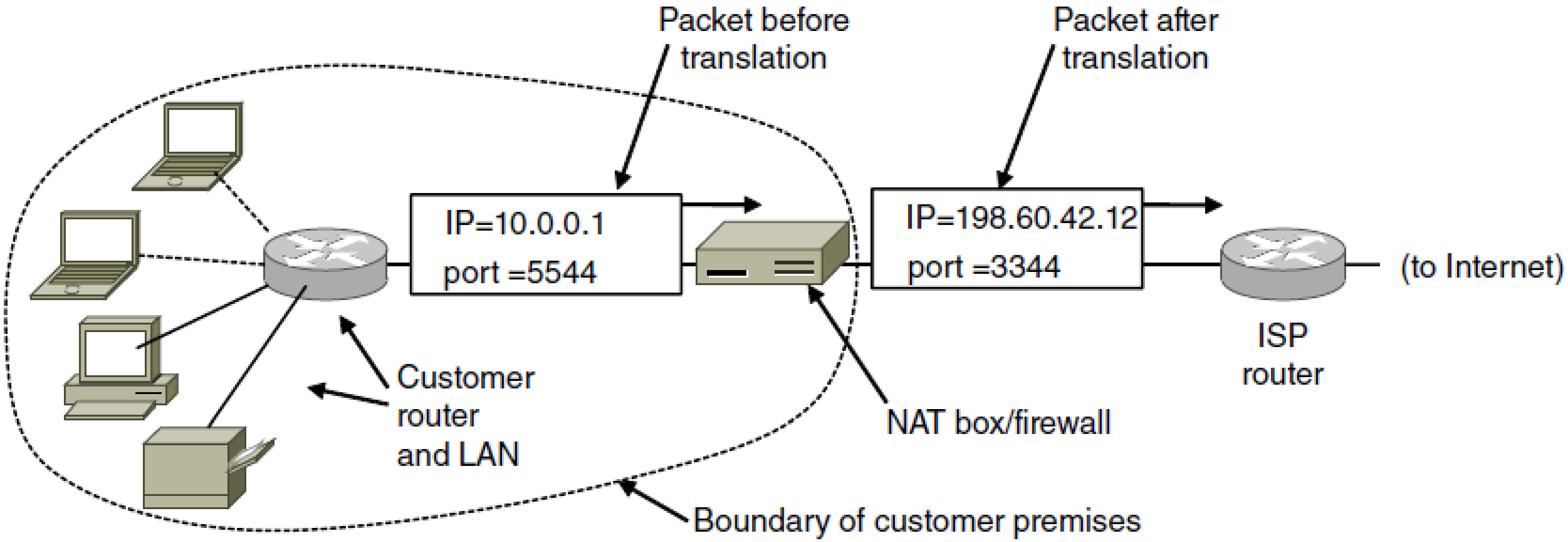
- When a packet arrives at the NAT box from the ISP, the *destination port* in the TCP header is extracted and used as an index into the NAT box's mapping table.
  - From the entry located, the internal IP address and original TCP source port are extracted and inserted into the packet.
  - Both the IP and TCP checksums are recomputed and inserted into the packet.
  - The packet is then passed to the company router for normal delivery using the 10.*x*.*y*.*z* address.

Cont.

- Some of objections to NAT:

1. NAT violates the architectural model of IP, which states that every IP address uniquely identifies a single machine worldwide.
2. NAT breaks the end-to-end connectivity model of the Internet.
3. NAT changes the Internet from a connectionless network to a kind of connection-oriented network.
4. NAT violates the most fundamental rule of protocol layering: layer *k* may not make any assumptions about what layer *k*+1 has put into the payload field.
5. Processes on the Internet are not required to use TCP or UDP.
6. Some applications insert IP addresses in the body of the text. The receiver then extracts these addresses and uses them. Since NAT knows nothing about these addresses, it cannot replace them, so any attempt to use them on the remote side will fail.
7. Since the TCP *source port* field is 16 bits, at most 65536–4096 machines can be mapped onto an IP address.

IP Addresses (8)



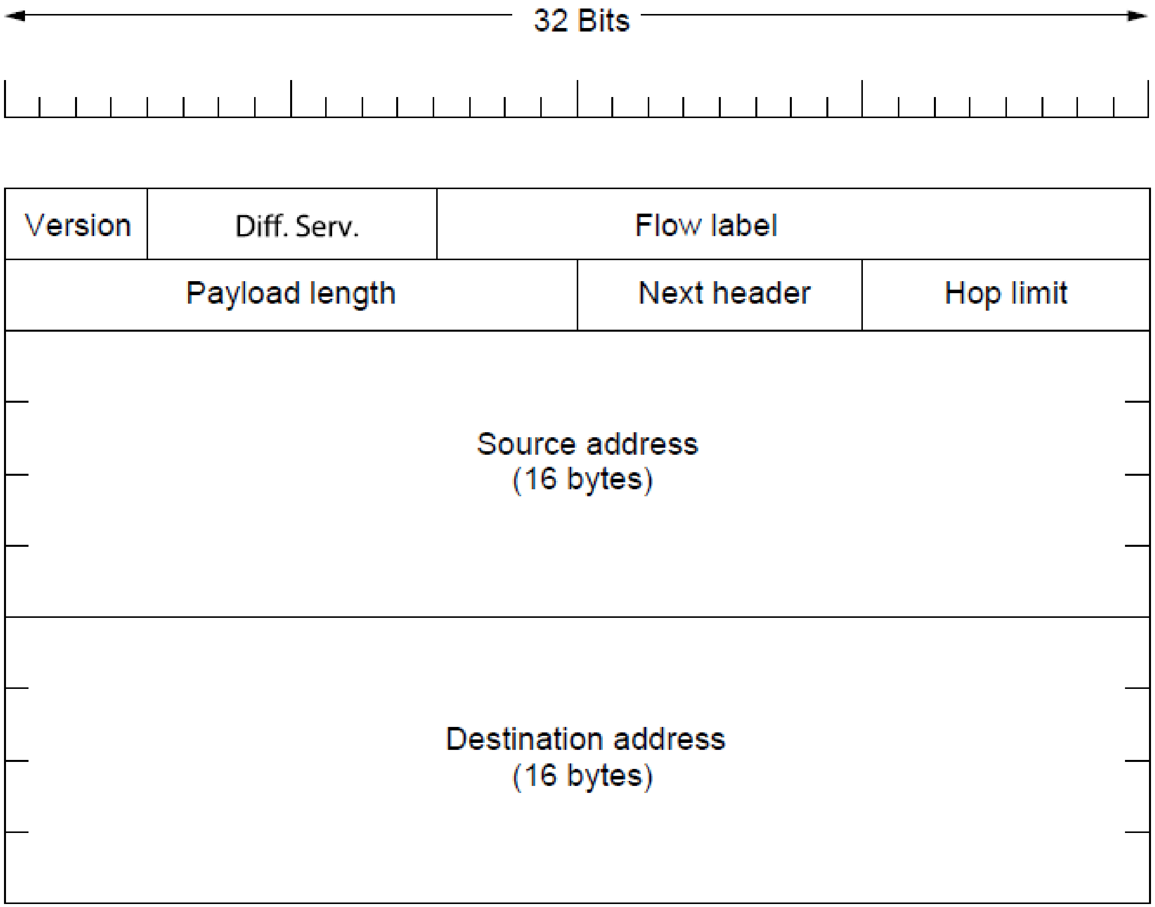
Placement and operation of a NAT box.

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

IP Version 6 Goals

- Support billions of hosts
- Reduce routing table size
- Simplify protocol
- Better security
- Attention to type of service
- Aid multicasting
- Roaming host without changing address
- Allow future protocol evolution
- Permit coexistence of old, new protocols. . .

IP Version 6 (1)



32 Bits

Version

Diff. Serv.

Flow label

Payload length

Next header

Hop limit

Source address
(16 bytes)

Destination address
(16 bytes)

The IPv6 fixed header (required).

Cont.

- In general, IPv6 is not compatible with IPv4, but it is compatible with all the other Internet protocols, including TCP, UDP, ICMP, IGMP, OSPF, BGP, and DNS, with small modifications being required.
- RFC 2460--2466.
- Major improvements of IPv6:
  - An IPv6 address has 16 bytes long.
  - An IPv6 header contains only 7 fields (versus 13 in IPv4).
  - It has better support for options. The way options are represented is different, making it simple for routers to skip over options not intended for them.
  - Authentication and privacy are key features of the new IP.
  - More attention has been paid to quality of service.

Cont.

- The *Version* field is 6 for IPv6.
- The *Differentiated services* (originally called *traffic class*) field is used to distinguish the class of service for packets with different real-time delivery requirements. ☹
- The *Flow label* will be used to allow a source and destination to set up a pseudoconnection with particular properties and requirements.☹
  - An attempt to have both the flexibility of a datagram subnet and the guarantees of a virtual circuit subnet.
  - Each flow is designated by the source address, destination address, and flow number, so many flows may be active at the same time between a given pair of IP addresses.

Cont.

- The *Payload length* field tells how many bytes follow the 40-byte header.
  - The 40 header bytes are no longer counted as part of the length as they used to be in IPv4 *Total length* field.
- The *Next header* field tells which of the (currently) six extension headers, if any, follows this one.
  - If this header is the last IP header, the *Next header* field tells which transport protocol handler (e.g. TCP, UDP) to pass the packet to.
- The *Hop limit* field is used to keep packets from living forever.
  - It is, in practice, the same as the *Time to live* field in IPv4, a field that is decremented on each hop.

Cont.

- The *Source address* and the *Destination address* fields have fixed length of 16 bytes each.
  - The 16-byte IPv6 address is written as *eight groups* of four hexadecimal digits with colons between the groups, such as

    8000:0000:0000:0000:0123:4567:89AB:CDEF
  - Three authorized optimizations on IPv6 addresses:
    1. Leading zeros within a group can be omitted.
    2. One or more groups of 16 zeros can be replaced by a pair of colons.

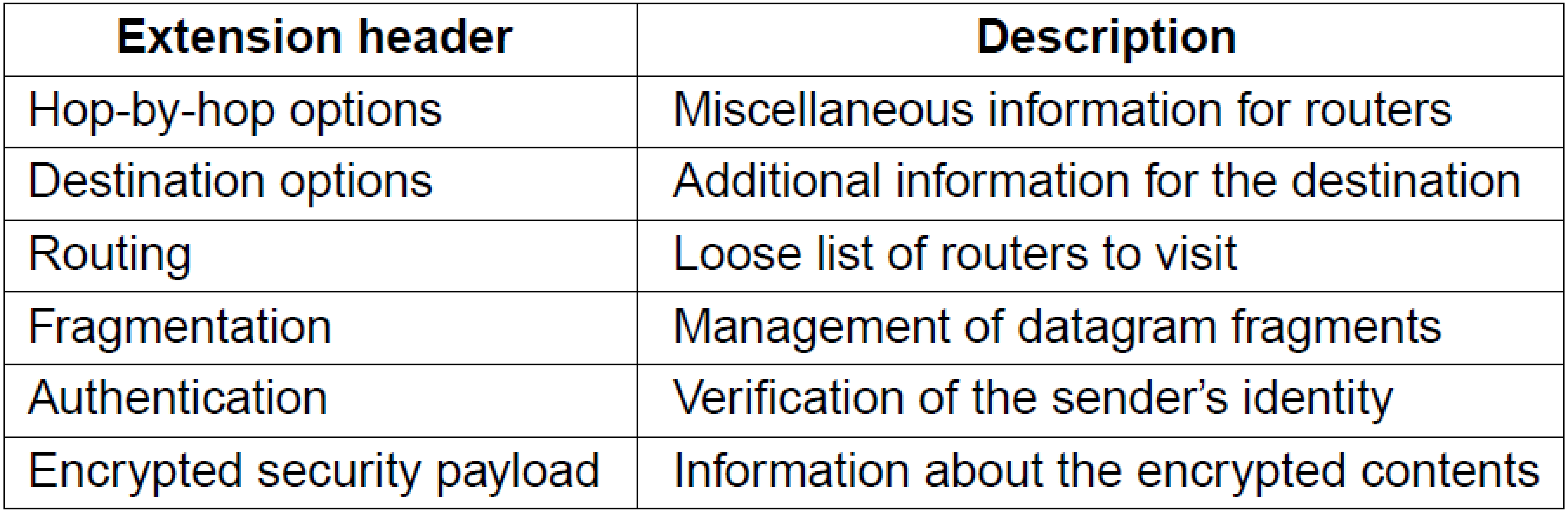
       8000::123:4567:89AB:CDEF
    3. IPv4 addresses can be written as a pair of colons and an old dotted decimal number, for example

       ::140.122.77.131

Cont.

- Compare the IPv4 header with the IPv6 header:
  - The *IHL* field is gone because the IPv6 has a fixed length.
  - The *Protocol* field was taken out because the Next header field tells what follows the last IP header.
  - All the fields relating to fragmentation were removed because IPv6 takes a different approach to fragmentation.
    - All IPv6 conformant hosts and routers must support packets of 1280 bytes (a minimum). ☹
      - When a host sends an IPv6 packet that is too large, the router that is unable to forward it drops the packet and sends back an error message.
  - The *Checksum* field is gone.

IP Version 6 (2)



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

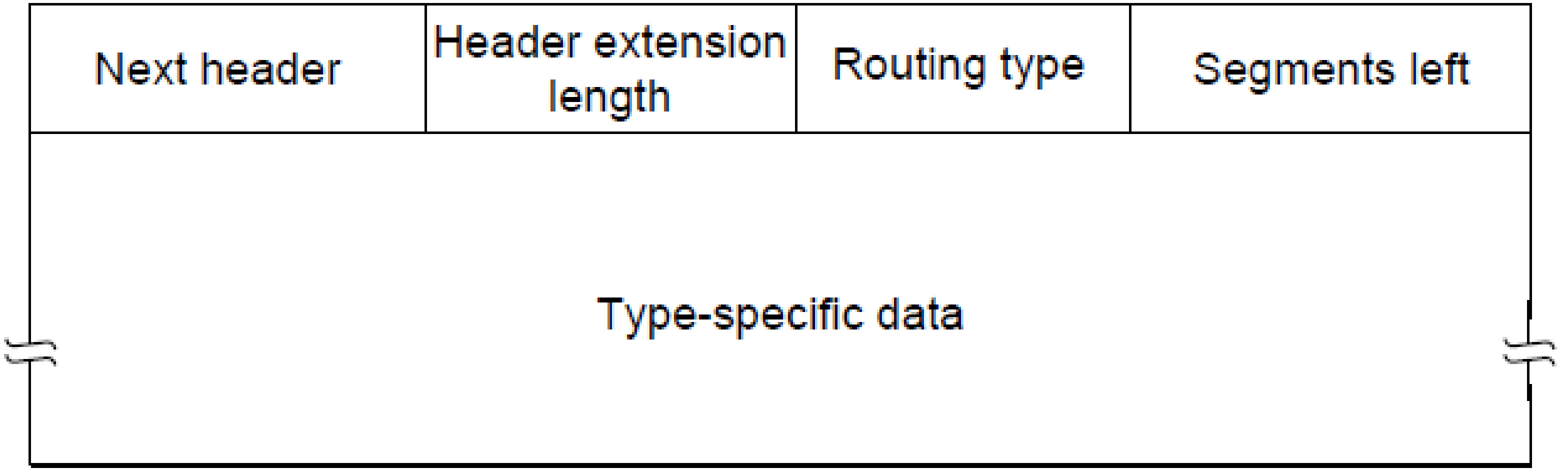
IPv6 extension headers

IP Version 6 (3)

<table>
<tr><td>Next header</td><td>0</td><td>194</td><td>4</td></tr>
<tr><td colspan="4">Jumbo payload length</td></tr>
</table>

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

IP Version 6 (4)



Next header

Header extension length

Routing type

Segments left

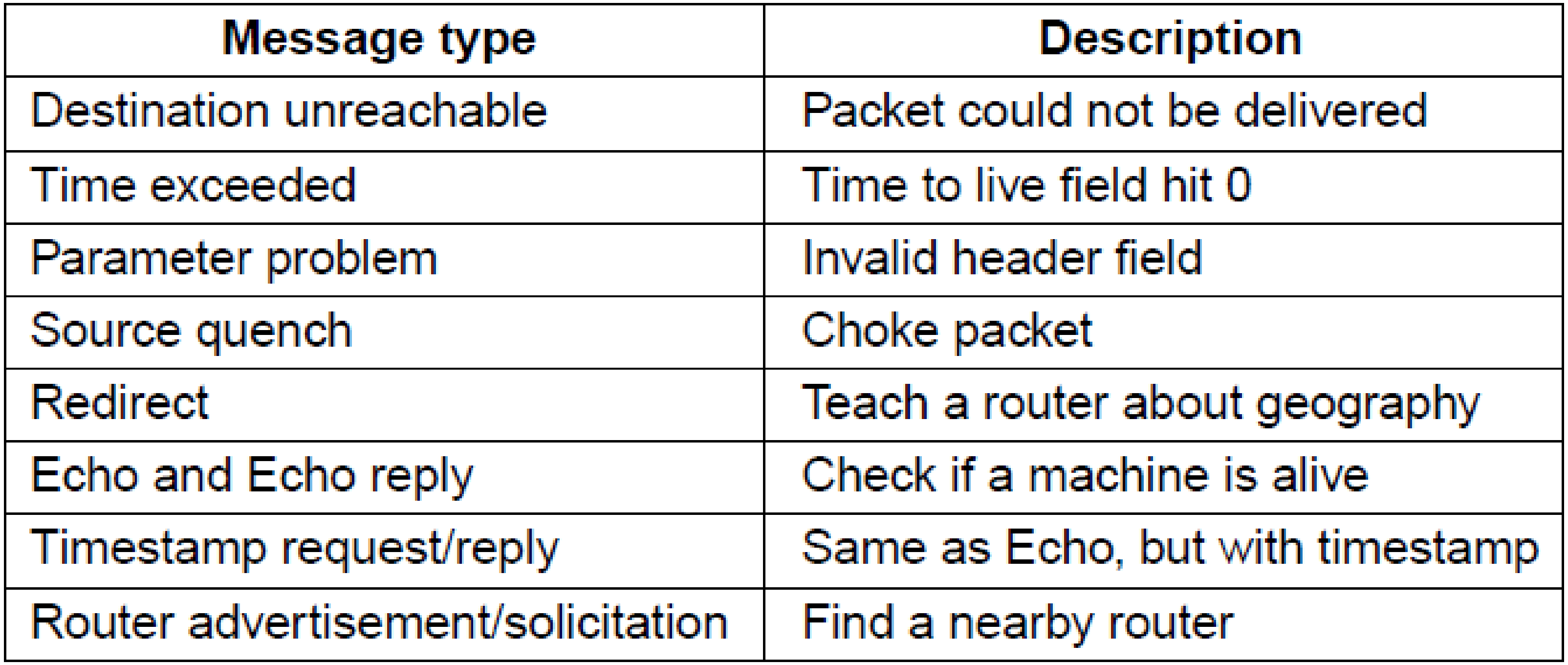
Type-specific data

The extension header for routing.

Internet Control Protocols

- ICMP
  - The Internet Control Message Protocol
- ARP (→NDP: Neighbor Discovery Protocol, for IPv6)
  - The Address Resolution Protocol
- DHCP
  - The Dynamin Host Configuration Protocol

Internet Control Protocols (1)



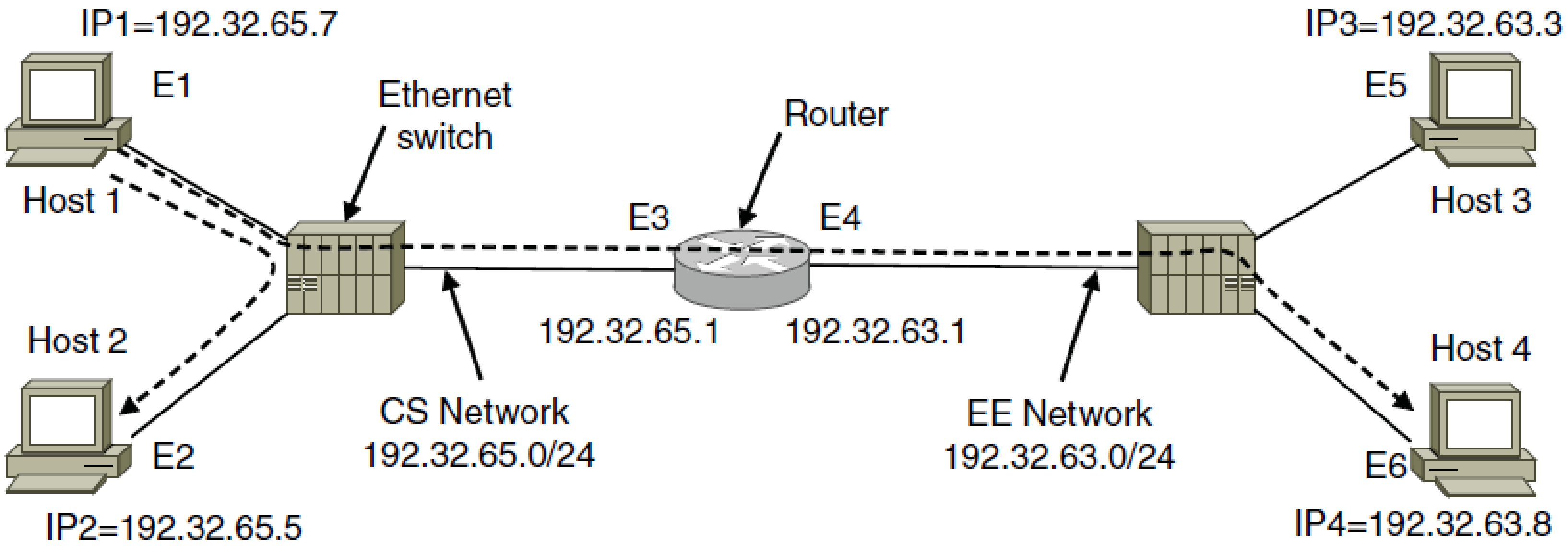
Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo and Echo reply	Check if a machine is alive
Timestamp request/reply	Same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

The principal ICMP message types.

ARP

- The data link layer hardware does not understand Internet address.
  - LAN interface boards only understand LAN addresses.
    - The Ethernet address has 48 bits.
  - How do IP addresses get mapped onto data link layer address?
    - One solution is to have a configuration file somewhere in the system that maps IP addresses onto Ethernet address.
    - A better solution is for a host to output a broadcast packet onto the Ethernet and to get a reply. The protocol for asking and getting a reply is called ARP.

Internet Control Protocols (2)



Frame	Source IP	Source Eth.	Destination IP	Destination Eth.
Host 1 to 2, on CS net	IP1	E1	IP2	E2
Host 1 to 4, on CS net	IP1	E1	IP4	E3
Host 1 to 4, on EE net	IP1	E4	IP4	E6

Two switched Ethernet LANs joined by a router

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

Cont.

- ARP is defined in RFC 826.
- The advantage of using ARP over configuration files is the simplicity.
- Some optimizations to make ARP more efficient:
  - It caches ARP results.
  - Have the host which issues ARP requests include its IP to Ethernet mapping in the ARP packet.
  - Have every machine broadcast its mapping when it boots.
    - This broadcast is generally done in the form of an ARP looking for its own IP address. ← known as a **gratuitous ARP**.
      - There should not be a response, but it makes an entry in everyone’s ARP cache.
      - If a response does arrive, two machines have been assigned the same IP address.

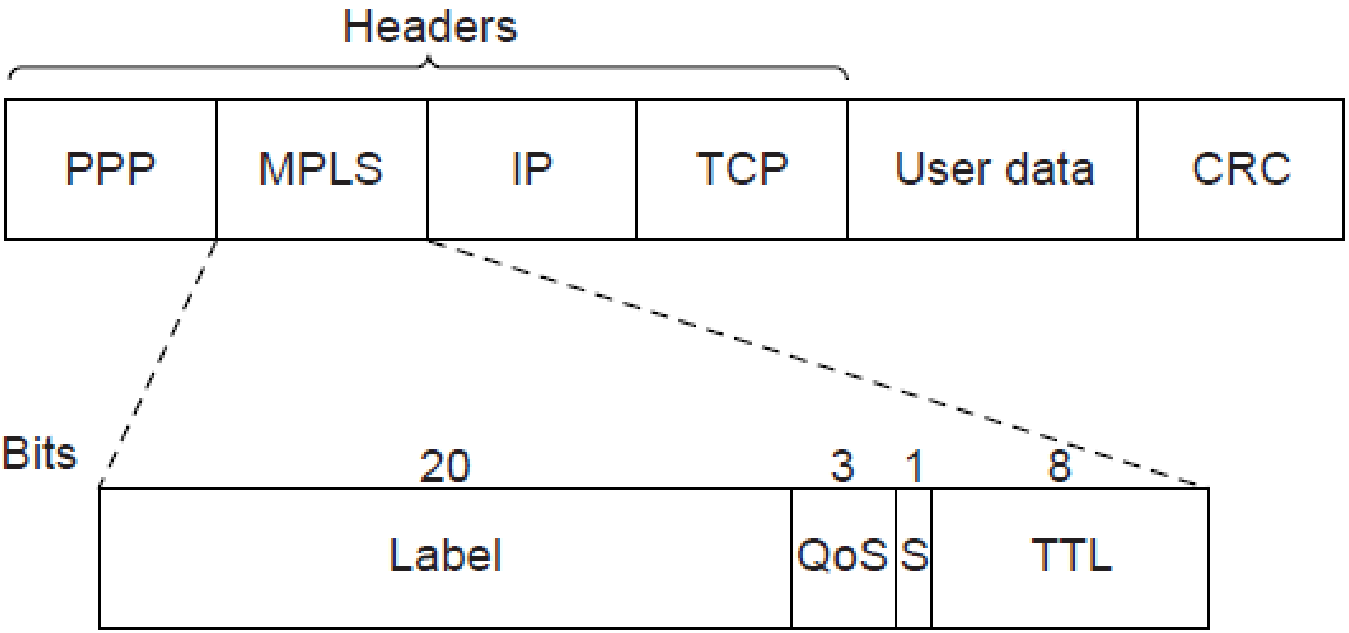
Cont.

- To allow mappings to change, entries in the ARP cache should time out after a few minutes. (Allow to replace a LAN card, or to be a mobile host)
  - When the IP addresses are in different networks (subnets), two solutions:
    - **Proxy ARP**: A router is configured *to respond to ARP request* for outside/local networks.
    - Have the source host *immediately* see that the destination is on a remote network and just send all such traffic to the router, the **default gateway**, that handles all remote traffic.

DHCP

- Given an Ethernet address, what is the corresponding IP address?
  - DHCP (**Dynamic Host Configuration Protocol**)
    - With DHCP, each network must have a DHCP server.
    - Allow both *manual* IP address assignment and *automatic* assignment.
    - DHCP, in RFCs 2131 and 2132
    - Relay DHCP discover packets.
      - To find its IP address, a newly-booted machine broadcasts a DHCP DISCOVER packet.
      - If the DHCP server is not directly attached to the network, the router will be configured to receive DHCP broadcasts and relay them to the DHCP server.
      - How long should an IP address be allocated?
      - A fixed period of time, by **IP leasing** techniques with renewal.

Label Switching and MPLS (1)



Transmitting a TCP segment using IP, MPLS, and PPP.

Cont.

- Add a label in front of each packet and do the routing based on the label rather than on the destination address.
- Using table lookup techniques, routing can be done very quickly and any necessary resources can be reserved along the path.--- close to virtual circuits.
- Called by the name label switching, tag switching or multiprotocol label switching (MPLS).
- In some sense, MPLS is layer 2.5
- The generic MPLS header has four fields:
  - The *label* field holds the index.
  - The *QoS* field indicates the class of service.
  - The *S* field relates to stacking multiple labels in hierarchical networks.
  - The *TTL* field

Cont.

- It is possible to build MPLS switches that can forward both IP packets and non-IP packets.
- When an MPLS-enhanced packet arrives at a LSR(Label Switching Routers), the label is used as an index into a table to determine the outgoing line to use and also the new label to use.
- One difference from traditional virtual circuits is the level of aggregation.
  - The flows that are grouped together under a single label are said to belong to the same FEC (Forwarding Equivalence Classes).
    - This class covers not only where the packets are going, but also their service class, because of all their packets are treated the same way for forwarding purposes.

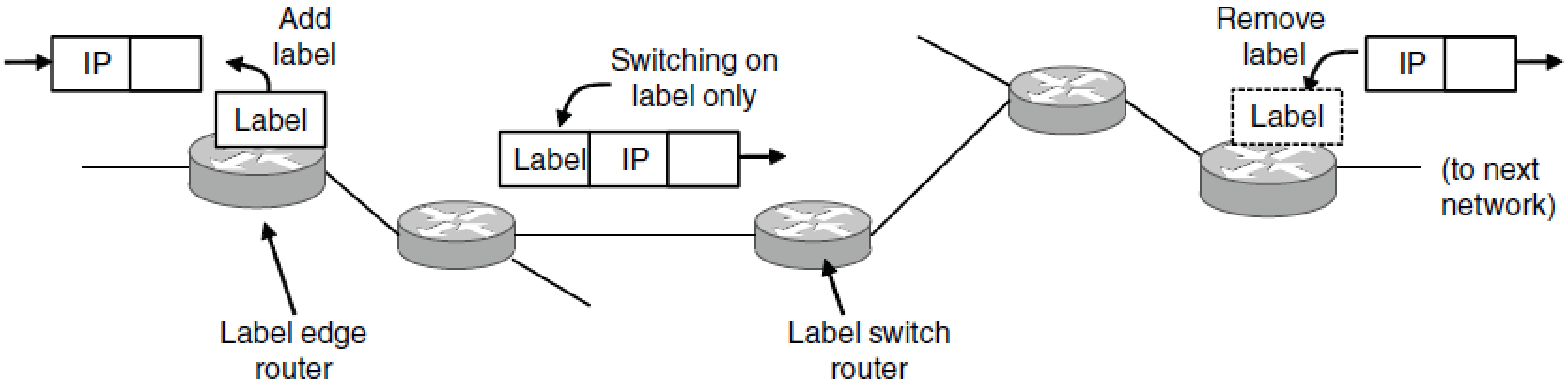
Cont.

- One major difference between MPLS and conventional VC designs is how the forwarding table is constructed.
  - In traditional VC networks, when a user wants to establish a connection, a setup packet is launched into the network to create the path and make the forwarding table entries.
  - MPLS does not work that way because there is no setup phase for each connection.

Cont.

- Two approaches for the forwarding table entries to be created:
  - The **data-driven** approach: When a packet arrives, the first router it hits contacts the router downstream where the packet has to go and asks it generate a label for the flow. This method is applied recursively.
    - It is primarily used on networks in which the underlying transport is ATM.
  - The **control-driven** approach: It is used on networks not based on ATM.
    - When a router is booted, it checks to see for which routes it is the final destination.
    - It then creates one or more FEC for them, allocates a label for each one, and passes the labels to its neighbors.
    - They, in turn, enter the labels in their forwarding tables and send new labels to their neighbors, until all the routers have acquired the path.

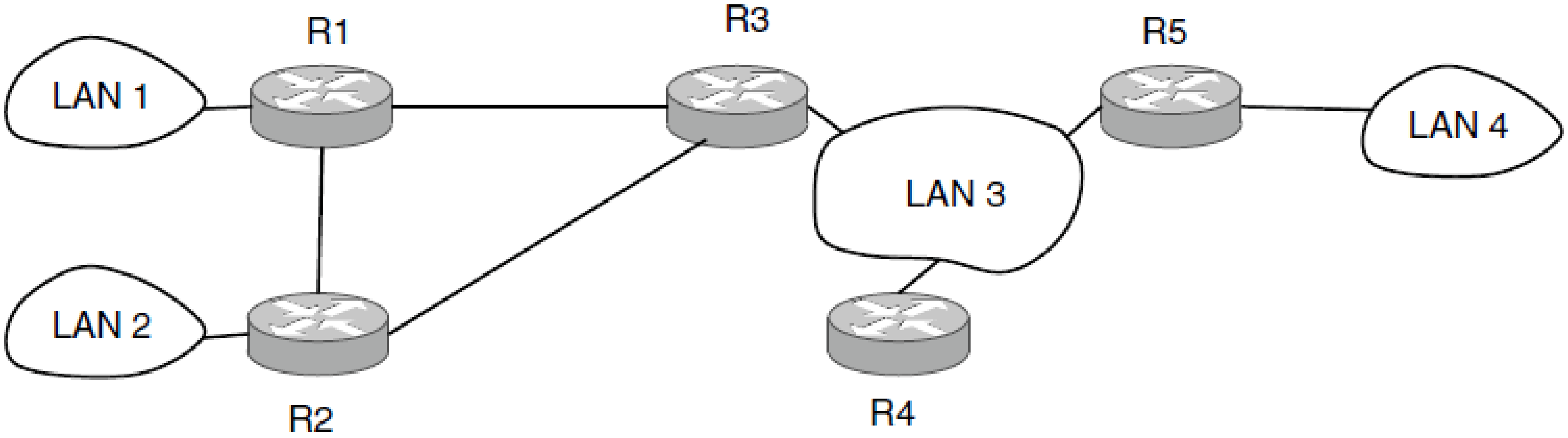
Label Switching and MPLS (2)



Forwarding an IP packet through an MPLS network

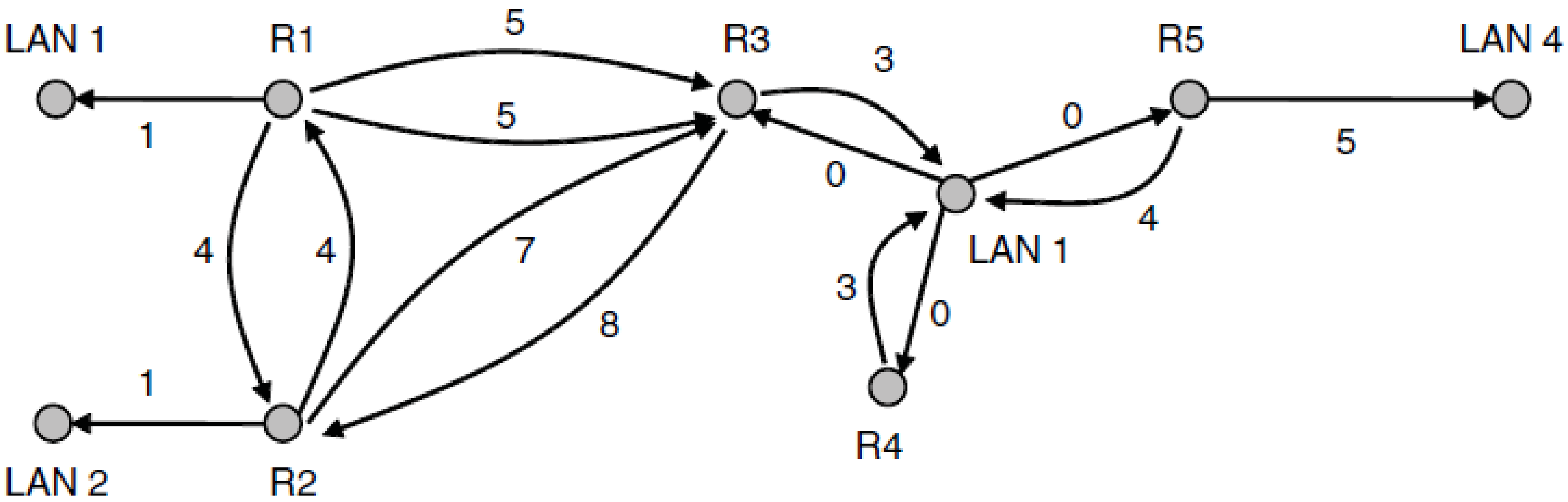
Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

OSPF—An Interior Gateway Routing Protocol (1)



An autonomous system

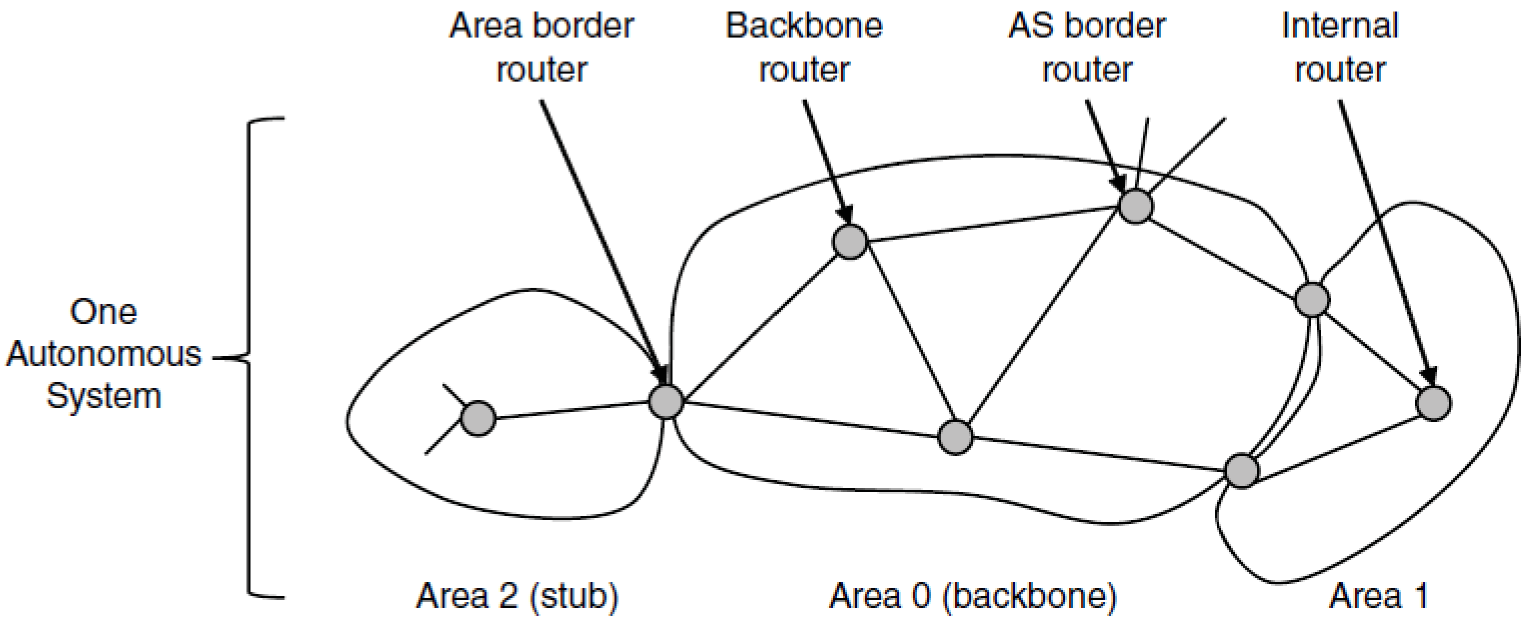
OSPF—An Interior Gateway Routing Protocol (2)



A graph representation of the previous slide.

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

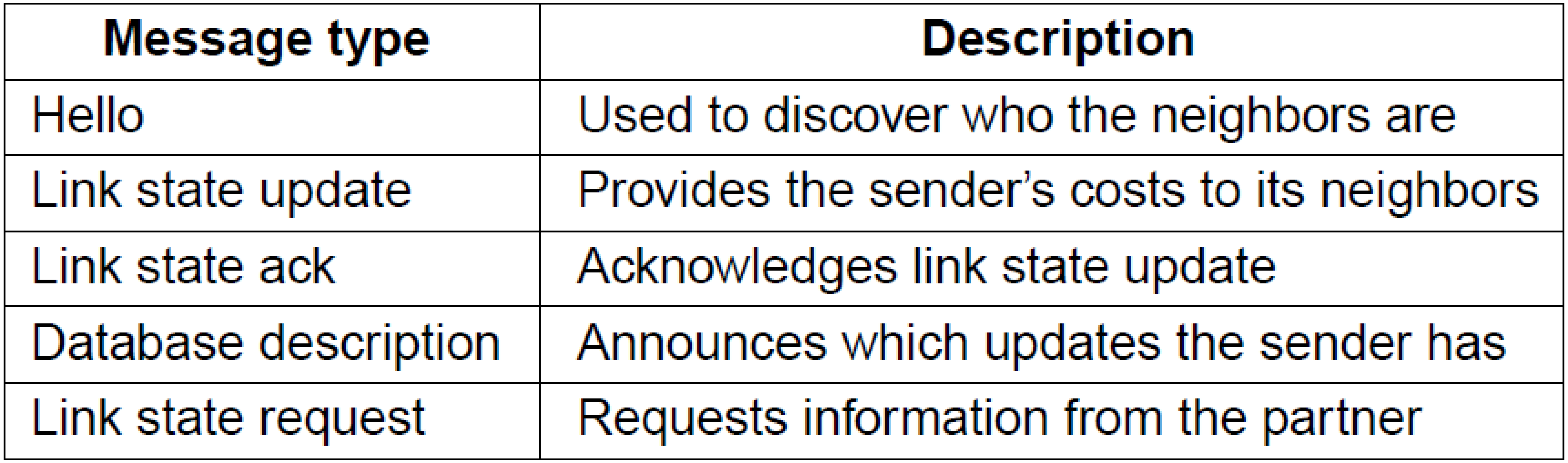
OSPF—An Interior Gateway Routing Protocol (3)



The relation between ASes, backbones, and areas in OSPF.

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

OSPF—An Interior Gateway Routing Protocol (4)



Message type	Description
Hello	Used to discover who the neighbors are
Link state update	Provides the sender’s costs to its neighbors
Link state ack	Acknowledges link state update
Database description	Announces which updates the sender has
Link state request	Requests information from the partner

The five types of OSPF messages

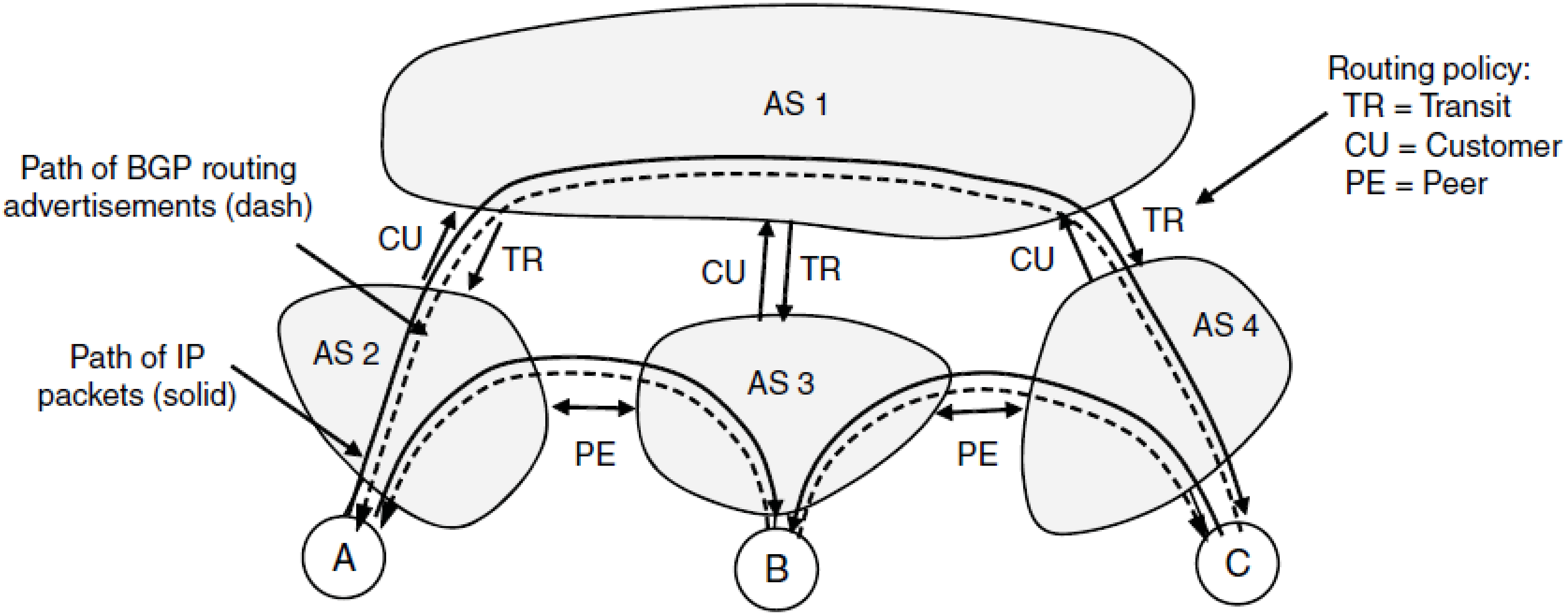
Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

BGP—The Exterior Gateway Routing Protocol (1)

Examples of routing constraints:

1. No commercial traffic for educat. network
2. Never put Iraq on route starting at Pentagon
3. Choose cheaper network
4. Choose better performing network
5. Don’t go from Apple to Google to Apple

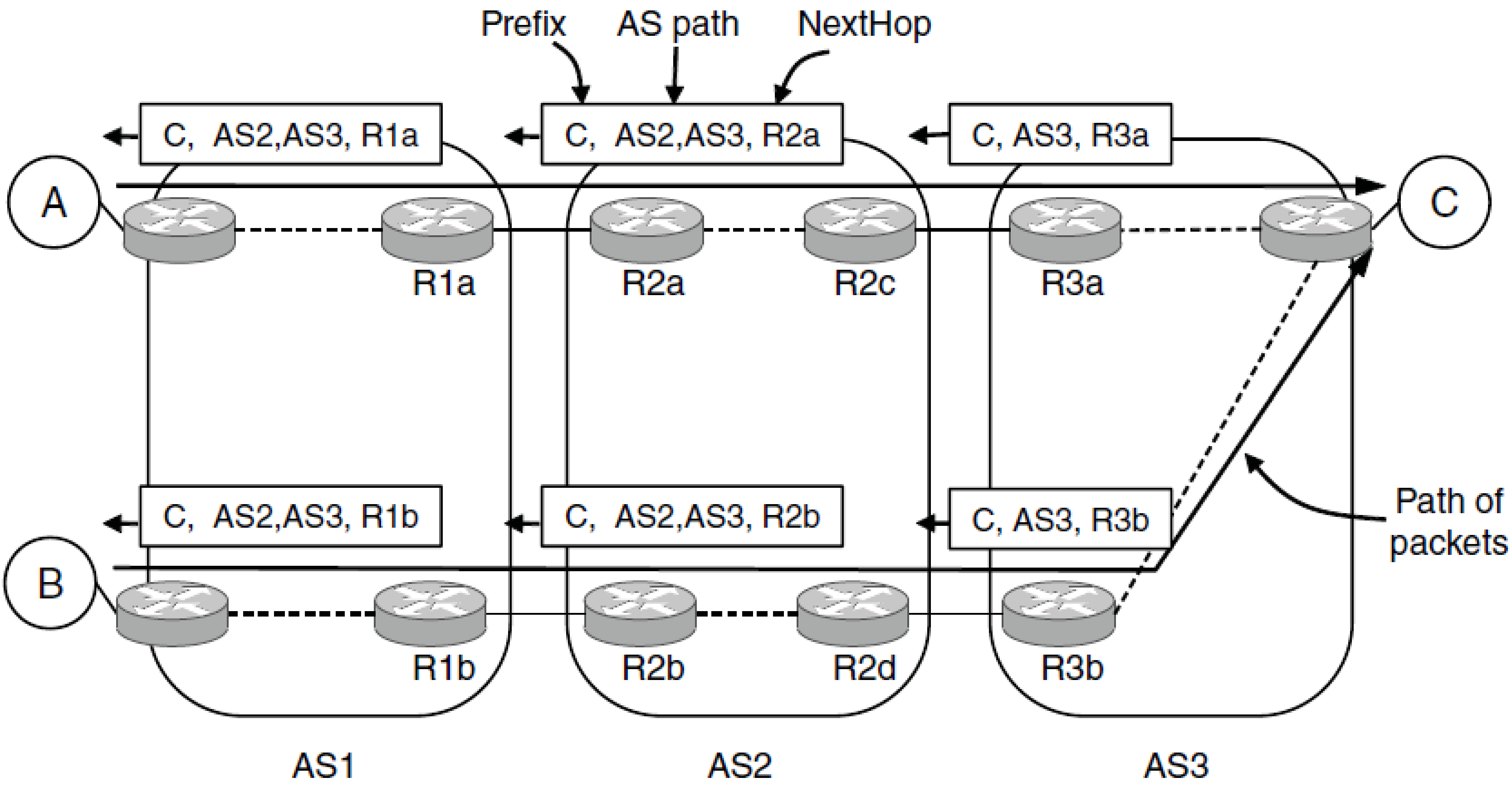
BGP—The Exterior Gateway Routing Protocol (2)



Routing policies between four Autonomous Systems

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

BGP—The Exterior Gateway Routing Protocol (3)



Propagation of BGP route advertisements

Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

Mobile IP

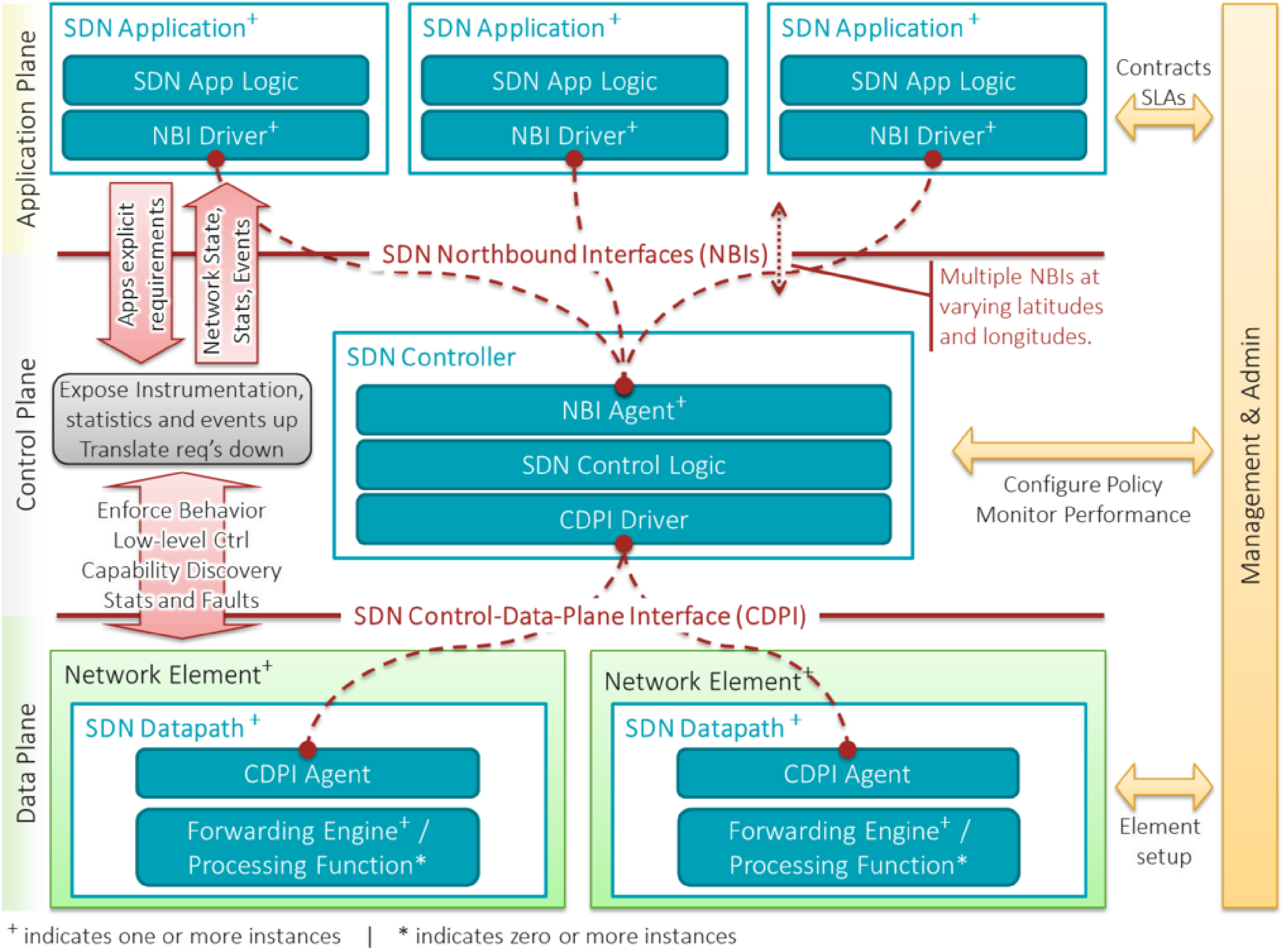
Goals

1. Mobile host use home IP address anywhere.
2. No software changes to fixed hosts
3. No changes to router software, tables
4. Packets for mobile hosts – restrict detours
5. No overhead for mobile host at home.

End

Chapter 5

SDN (?)



Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011