UNIT III

The Network Layer

5.1. NETWORK LAYER DESIGN ISSUES

5.2. ROUTING ALGORITHMS

5.3. CONGESTION CONTROL ALGORITHMS

5.4. QUALITY OF SERVICE

5.5. INTERNETWORKING

5.6. THE NETWORK LAYER IN THE INTERNET

5.7. SUMMARY

PREVIOUS LAYERS

- The purpose of the physical layer is to transport a raw bit stream from one machine to another.
- The main task of the data link layer is to transform a raw transmission faculty into a line that appears free of undetected transmission errors to the network layer.

The Network Layer (4)

- This layer is concerned with getting packets from the source all the way to the destination.
- Getting to the destination may require making many hops at intermediate routers along the way.
- Thus, the network layer is the lowest layer that deals with end-to-end transmission.

The Network Layer (5)

- To achieve its goals, the network layer must know about the topology of the communication subset (i.e., the set of all routers) and choose appropriate paths through it.
- It must also take care to choose routes to avoid overloading some of the communication lines and routers while leaving others idle.

The Network Layer (6)

- Finally, when the source and destination are in different networks, new problems occur. It is up to the network layer to deal with them.
- In this chapter we will study all these issues and illustrate them, primarily using the Internet and its network layer protocol, IP, although wireless networks will also be addressed.

5.1. Network Layer Design Issues (1)

- In the following sections we will provide anintroduction to some of the issues that the designers of the network layer must grapple with.
- These issues include the service provided tothe transport layer and the internal design of the subnet.

5.1. Network Layer Design Issues (2)

- Store-and-Forward Packet Switching
- Services Provided to the Transport Layer
- Implementation of Connectionless Service
- Implementation of Connection-Oriented Service
- Comparison of Virtual-Circuit and Datagram Subnets

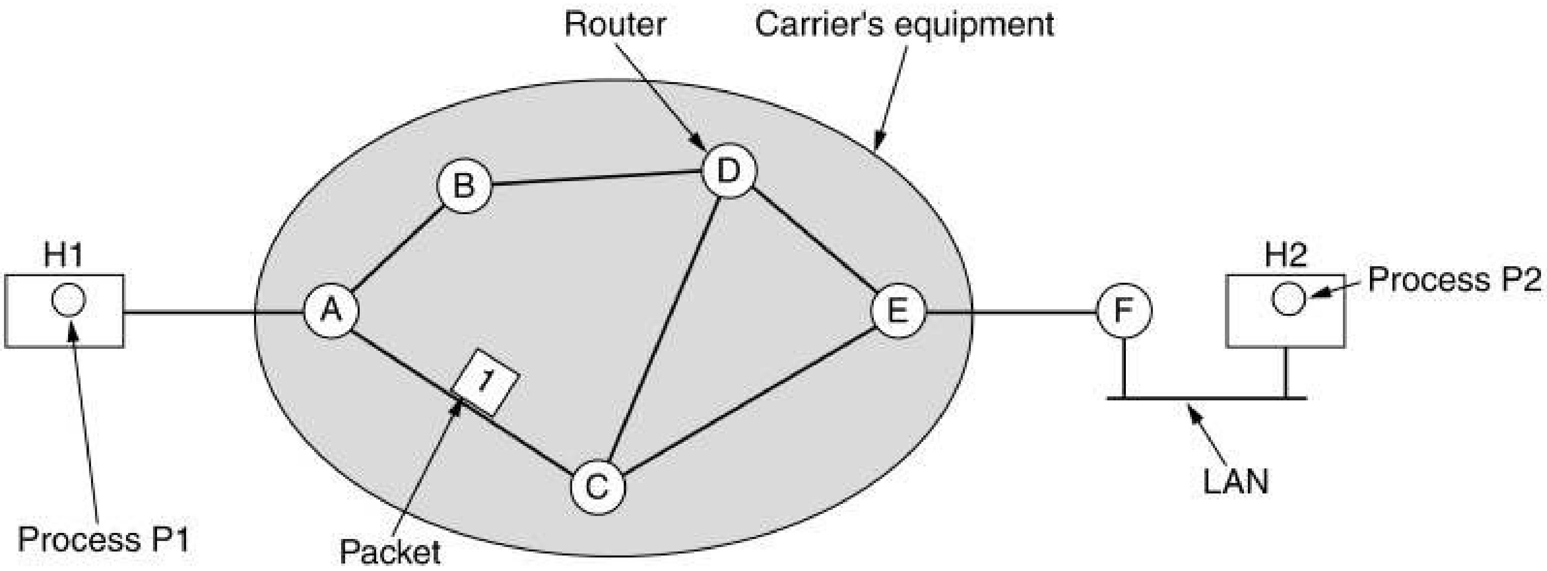
5.1. Network Layer Design Issues (9)

1. Store-and-Forward Packet Switching (1)

- Let us restate the context in which the network layer protocols operate.

5.1. Network Layer Design Issues (10)

1.Store-and-Forward Packet Switching (2)



The environment of the network layer protocols.

5.1. Network Layer Design Issues (11)

1.Store-and-Forward Packet Switching (3)

- A host with a packet to send transmits it to the nearest router, either on its own LAN or over a point-to-point link to the carrier.
- The packet is stored there until it has fully arrived so the checksum can be verified.

5.1. Network Layer Design Issues (12)

1.Store-and-Forward Packet Switching (4)

- Then it is forwarded to the next router along the path until it reaches the destination host, where it is delivered.
- This mechanism is store-and-forward packet switching, as we have seen in previous chapters.

5.1. Network Layer Design Issues (13)

2.Services Provided to the Transport Layer (1)

- The network layer provides services to the transport layer at the network layer/transport layer interface.
- An important question is what kind of services the network layer provides to the transport layer.
- The network layer services have been designed with the following goals in mind.

5.1. Network Layer Design Issues (8)

2.Services Provided to the Transport Layer (2)

a) The services should be independent of the router technology.

b) The transport layer should be shielded from the number, type, and topology of the routers present.

c) The network addresses made available to the transport layer should use a uniform numbering plan, even across LANs and WANs.

5.1. Network Layer Design Issues (9)

2.Services Provided to the Transport Layer (3)

- The discussion centers on whether the network layer should provide connection-oriented service or connectionless service.
- Internet community's opinion: the routers' job is moving packets around and nothing else. The subnet is inherently unreliable. Therefore, the hosts should accept the fact that the network is unreliable and do error control and flow control themselves.

5.1. Network Layer Design Issues (10)

2.Services Provided to the Transport Layer (4)

- This viewpoint leads quickly to the conclusion that the network service should be connectionless, with primitives SEND PACKET and RECEIVE PACKET.
- Furthermore, each packet must carry the full destination address, because each packet sent is carried independently of its predecessors, if any.

5.1. Network Layer Design Issues (11)

2.Services Provided to the Transport Layer (5)

- Telephone companies' opinion: the subnet should provide a reliable, connection-oriented service.
- In this view, quality of service is the dominant factor, and without connections in the subnet, quality of service is very difficult to achieve, especially for real-time traffic such as voice and video.

5.1. Network Layer Design Issues (18)

2. Services Provided to the Transport Layer (6)

- The Internet offers connectionless network-layer service.
- ATM networks offer connection-oriented network-layer service.
- However, it is interesting to note that as quality-of-service guarantees are becoming more and more important, the Internet is evolving.

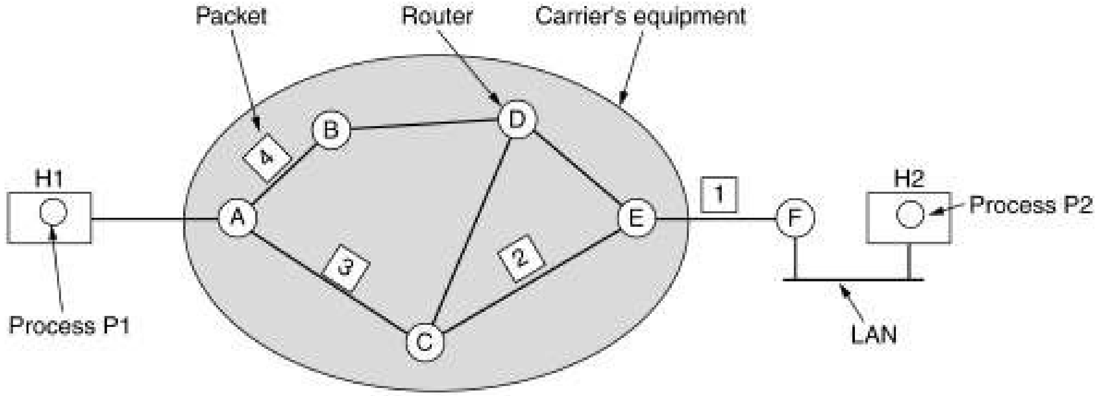
5.1. Network Layer Design Issues (19)

3. Implementation of Connectionless Service (1)

- If connectionless service is offered, packets are injected into the subnet individually and routed independently of each other.
- In this context, the packets are frequently called datagrams and the subnet is called a datagram subnet.

5.1. Network Layer Design Issues (20)

3.Implementation of Connectionless Service (2)



A's table				C's table		E's table	
initially		later					
A	–	A	–	A	A	A	C
B	B	B	B	B	A	B	D
C	C	C	C	C	–	C	C
D	B	D	B	D	D	D	D
E	C	E	B	E	E	E	–
F	C	F	B	F	E	F	F

Dest. Line

Routing within a diagram subnet.

5.1. Network Layer Design Issues (15)

3.Implementation of Connectionless Service (3)

Routers

- When a packet comes into a router, the frame header and trailer are stripped off and the packet located in the frame's payload field is passed to the routing software. This software uses the packet header to choose an output line.

5.1 Network Layer Design Issues (16)

3. Implementation of Connectionless Service(3)

- Each router has an internal table telling it where to send packets for each possible destination.
- The algorithm that manages the tables and makes the routing decisions is called the routing algorithm. Routing algorithms are one of the main things we will study in this chapter.

5.1 Network Layer Design Issues (17)

4. Implementation of Connection-Oriented Service (1)

- If connection-oriented service is used, a path from the source router to the destination router must be established before any data packets can be sent.
- This connection is called a VC (virtual circuit), in analogy with the physical circuits set up by the telephone system, and the subnet is called a virtual-circuit subnet.

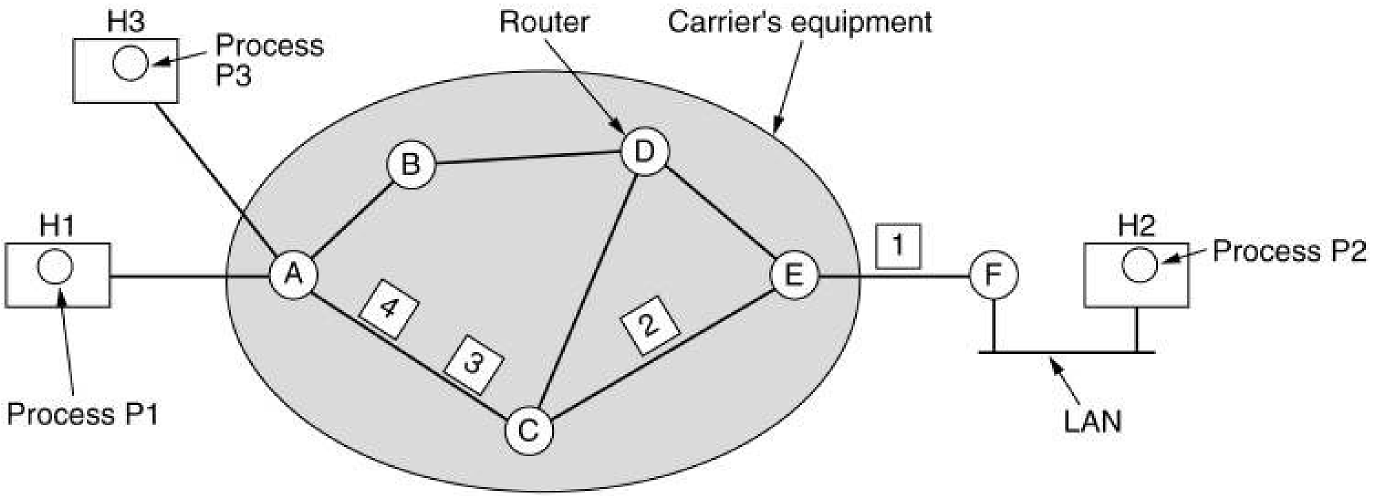
5.1 Network Layer Design Issues(18)

4.Implementation of Connection-Oriented Service (2)

- When a connection is established, a route from the source machine to the destination machine is chosen as part of the connection setup and stored in tables inside the routers.
- When the connection is released, the virtual circuit is also terminated.
- With connection-oriented service, each packet carries an identifier telling which virtual circuit it belongs to.

5.1.Network Layer Design Issues(19)

4.Implementation of Connection-Oriented Service (3)



A's table

In		Out	
H1	1	C	1
H3	1	C	2

C's table

A	1	E	1
A	2	E	2

E's table

C	1	F	1
C	2	F	2

Routing within a virtual-circuit subnet.

5.1 Network Layer Design Issues (20)

5. Comparison of Virtual-Circuit and Datagram Subnets(1)

- Both virtual circuits and datagrams have their supporters and their detractors.
- One trade-off is between router memory space and bandwidth. VC allow packets to contain circuit numbers instead of full destination addresses. If the packets tend to be fairly short, a full destination address in every packet may represent a significant amount of overhead and hence, wasted bandwidth.

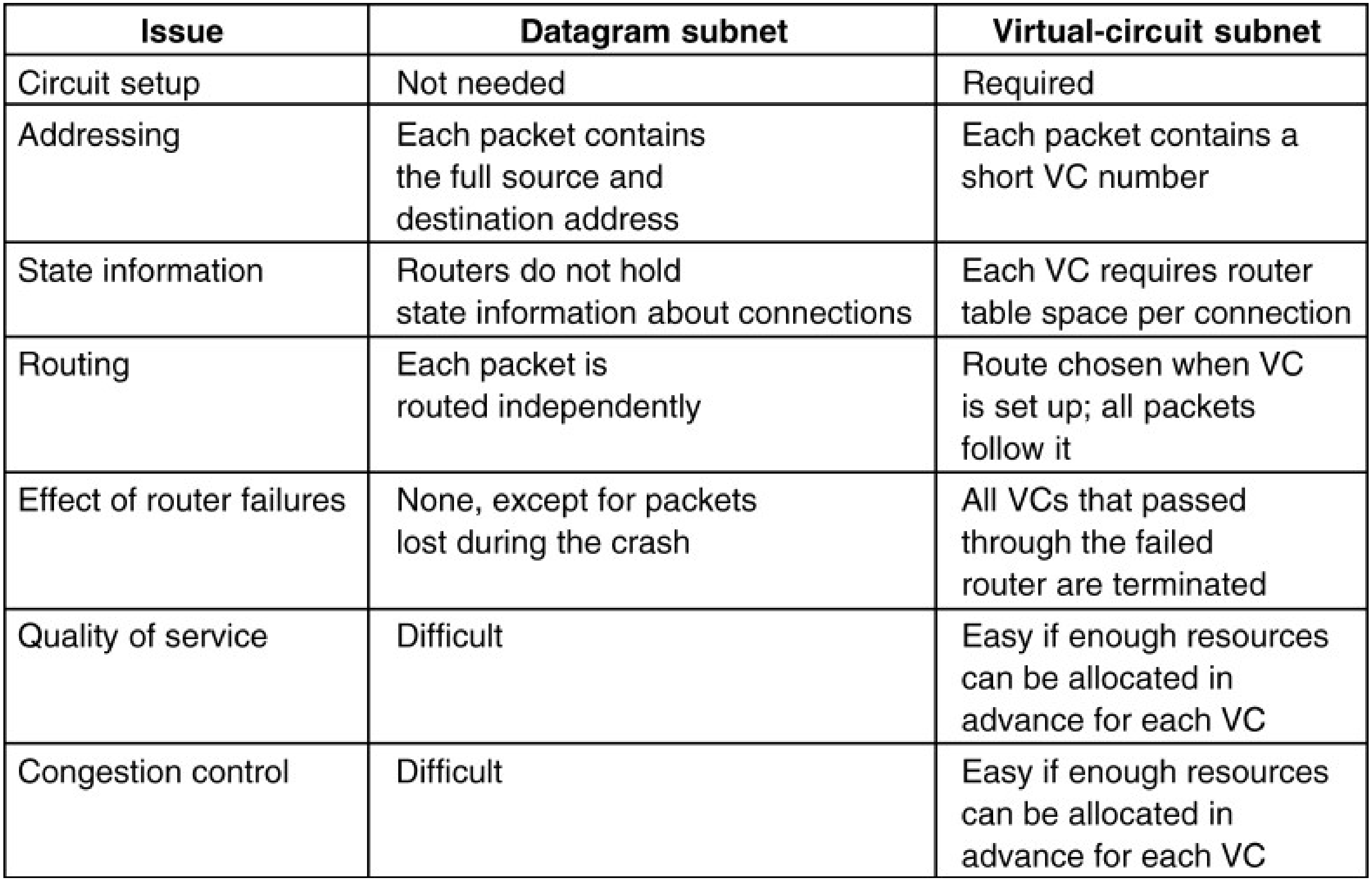
5.1 Network Layer Design Issues (21)

5. Comparison of Virtual-Circuit and Datagram Subnets(2)

- Another trade-off is setup time versus address parsing time.
- Using VC requires a setup phase, which takes time and consumes resources.
- VC have some advantages in guaranteeing quality of service and avoiding congestion within the subnet because resources can be reserved in advance, when the connection is established.

5.1 Network Layer Design Issues (22)

5. Comparison of Virtual-Circuit and Datagram Subnets (3)



Issue	Datagram subnet	Virtual-circuit subnet
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	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

5.2 Routing Algorithms (29)

- The main function of NL (Network Layer) is routing packets from the source machine to the destination machine.
- The algorithms that choose the routes and the data structures that they use are a major area of network layer design.
- The routing algorithm is that part of the NL software responsible for deciding which output line an incoming packet should be transmitted on.

5.2 Routing Algorithms (30)

- There are two processes inside router:

a) One of them handles each packet as it arrives, looking up the outgoing line to use for it in the routing table. This process is forwarding.

b) The other process is responsible for filling in and updating the routing tables. That is where the routing algorithm comes into play. This process is routing.

5.2 Routing Algorithms (31)

- Regardless of whether routes are chosen independently for each packet or only when new connections are established, certain properties are desirable in a routing algorithm:
- correctness, simplicity,
- robustness, stability,
- fairness, and optimality.

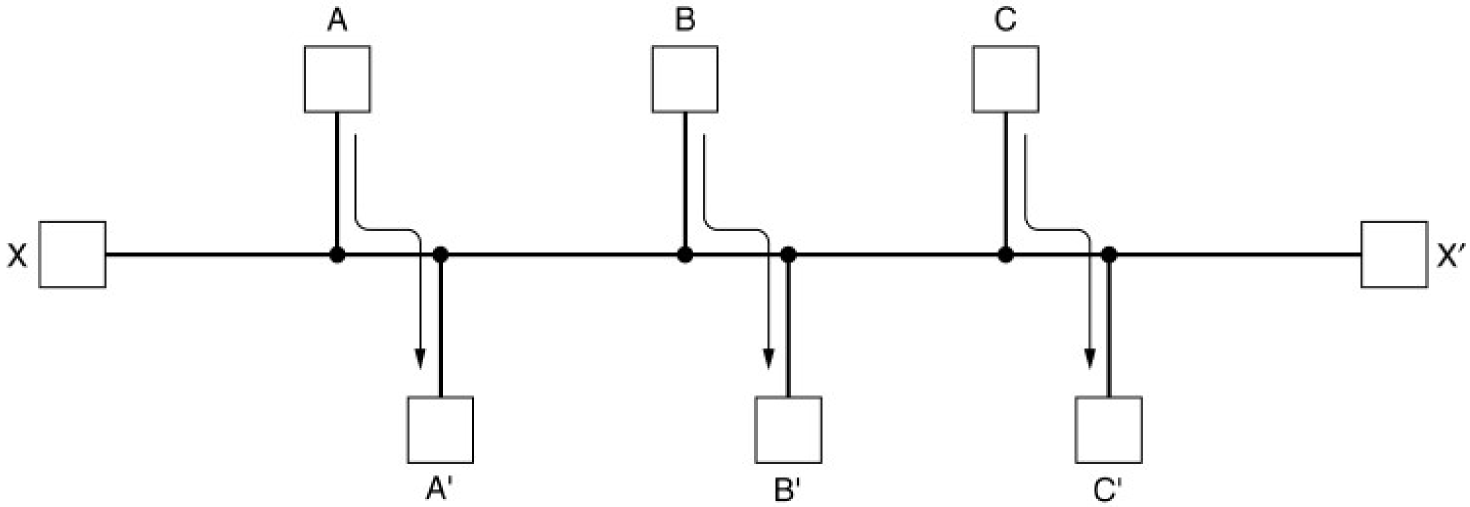
5.2 Routing Algorithms (32)

- Correctness and simplicity hardly require coment.
- Robustness: the routing algorith should be able to cope with changes in topology and traffic without requiring all jobs in all hosts to be aborted and the network to be rebooted every time some router crashes.

5.2 Routing Algorithms (33)

- Stability is also an important goal for the routing algorithm. A stable algorithm reaches equilibrium and stays there.
- Fairness and optimality may sound obvious – surely no reasonable person would oppose them – but as it turn out, they are often contradictory goals.

5.2 Routing Algorithms (34)



Conflict between fairness and optimality.

5.2 Routing Algorithms (35)

- Routing algorithms can be grouped into two major classes: nonadaptive and adaptive.
- Nonadaptive algorithm do not base their routing decisions on measurements or estimates of the current traffic and topology. Instead, the choice of the route to use to get from I to J is computed in advance, off line, and downloaded to the routers when the network is booted. This procedure is sometimes called static routing.

5.2 Routing Algorithms (36)

- Adaptive algorithm, in contrast, change their routingdecisions to reflect changes in the topology, and usually the traffic as well.
- Adaptive algorithms differ in where they get their information (e.g., locally, from adjacent routers, or from all routers), when they change the routes (e.g., every ΔT sec, when the load changes or when the topology changes), and what metric is used for optimization (e.g., distance, number of hops, or estimated transit time). This procedure is called dynamic routing.

5.2 Routing Algorithms (37)

- The Optimality Principle
- Shortest Path Routing
- Flooding
- Distance Vector Routing
- Link State Routing
- Hierarchical Routing
- Broadcast Routing
- Multicast Routing
- Routing for Mobile Hosts
- Routing in Ad Hoc Networks

5.2 Routing Algorithms (38)

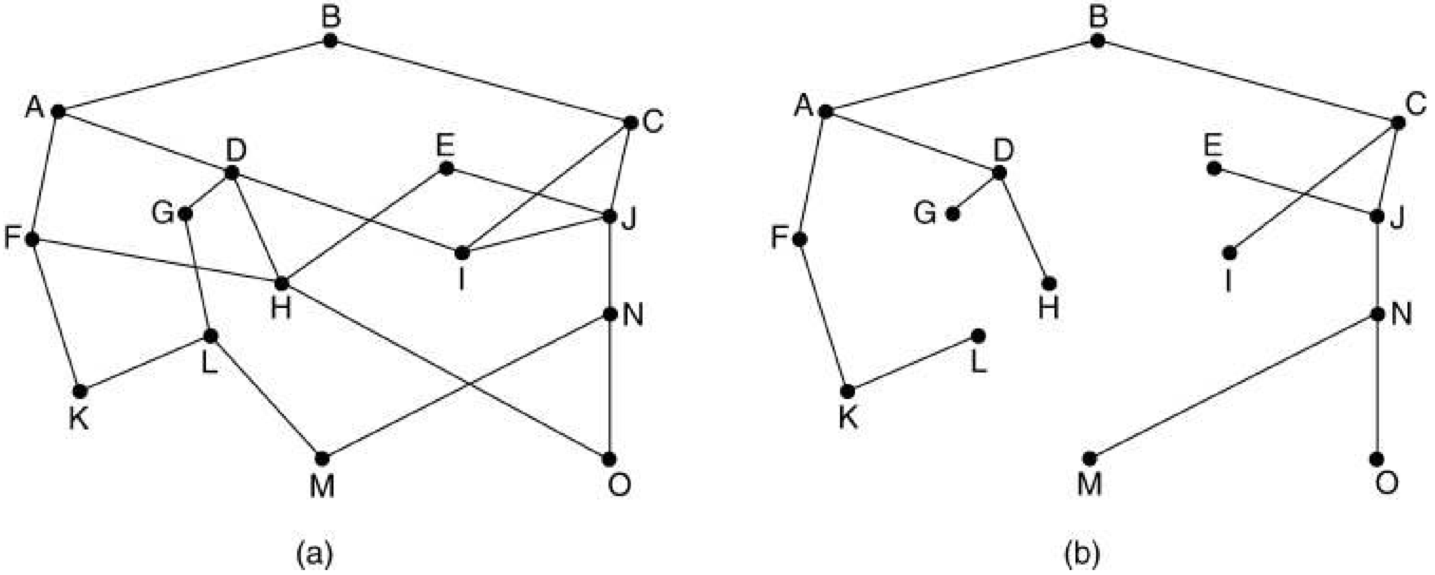
The Optimality Principle (1)

- One can make a general statement about optimal routes without regard to network topology or traffic.
- This statement is known as the optimality principle.
- It states that if router J is on the optimal path from router I to router K, then the optimal path from J to K also falls along the same route.

5.2 Routing Algorithms (39)
The Optimality Principle (2)

- As a direct consequence of the optimality principle, we can see that the set of optimal routes from all sources to a given destination form a tree rooted at the destination.
- Such a tree is called a sink tree.
- The goal of all routing algorithms is to discover and use the sink trees for all routers

5.2 Routing Algorithms (12)
The Optimality Principle (3)



(a) A subnet. (b) A sink tree for router B.

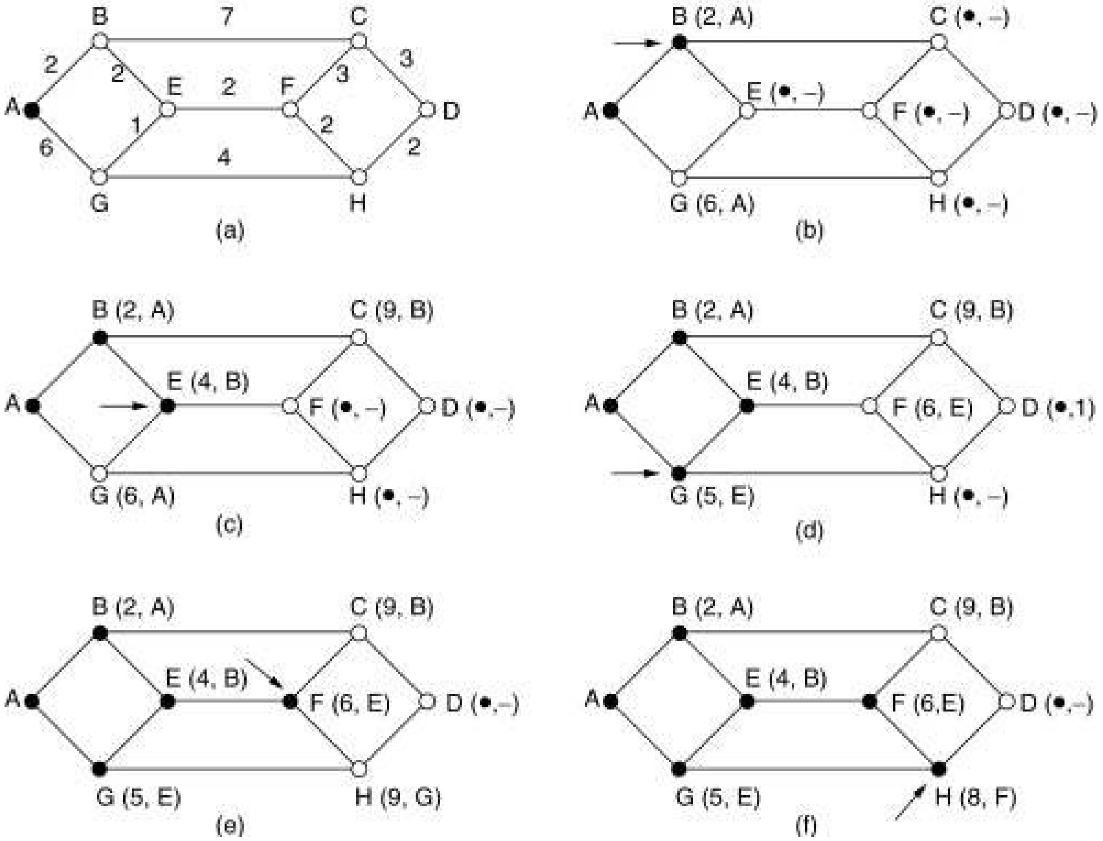
5.2 Routing Algorithms (41)

Shortest Path Routing (1)

- The idea is to build a graph of the subnet, with each node of the graph representing a router and each arc of the graph representing a communication line or link.
- To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph.

5.2 Routing Algorithms (42)

Shortest Path Routing (2)



The first 5 steps used in computing the shortest path from A to D. The arrows indicate the working node.

5.2 Routing Algorithms (15)
Shortest Path Routing (3)

- Many other metrics besides hops and physical distance are also possible.
- For example, each arc could be labeled with the mean queuing and transmission delay for some standard test packet as determined by hourly test runs.
- With this graph labeling, the shortest path is the fastest path rather than the path with the fewest arcs or kilometers.

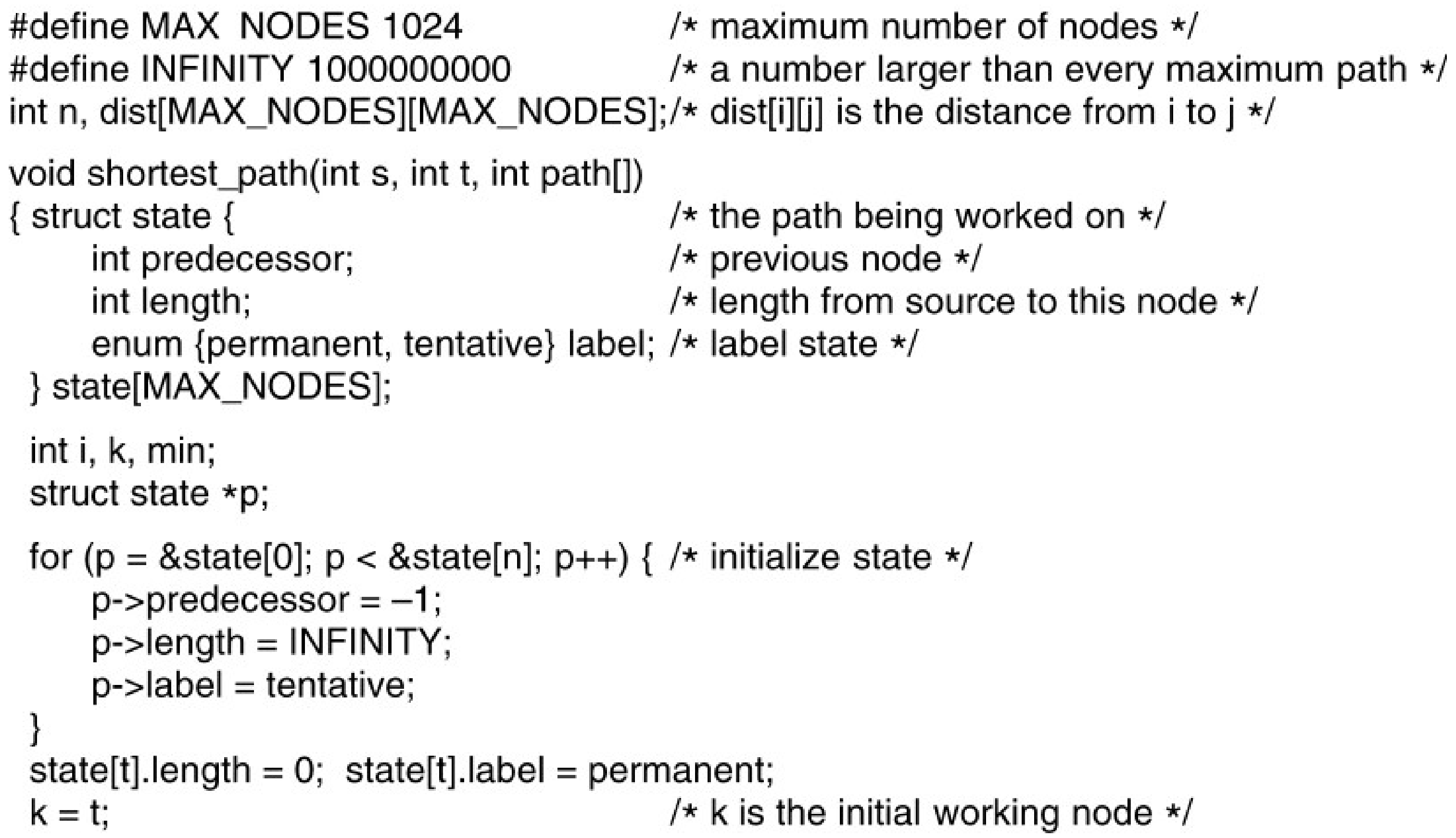
5.2 Routing Algorithms (16)

Shortest Path Routing (4)

- In the general case, the labels on the arcs could be computed as a function of the distance, bandwidth, average traffic, communication cost, mean queue length, measured delay, and other factors.
- By changing the weighting function, the algorithm would then compute the “shortest” path measured according to any one of a number of criteria or to a combination of criteria.

5.2 Routing Algorithms (17)

Shortest Path Routing (5)

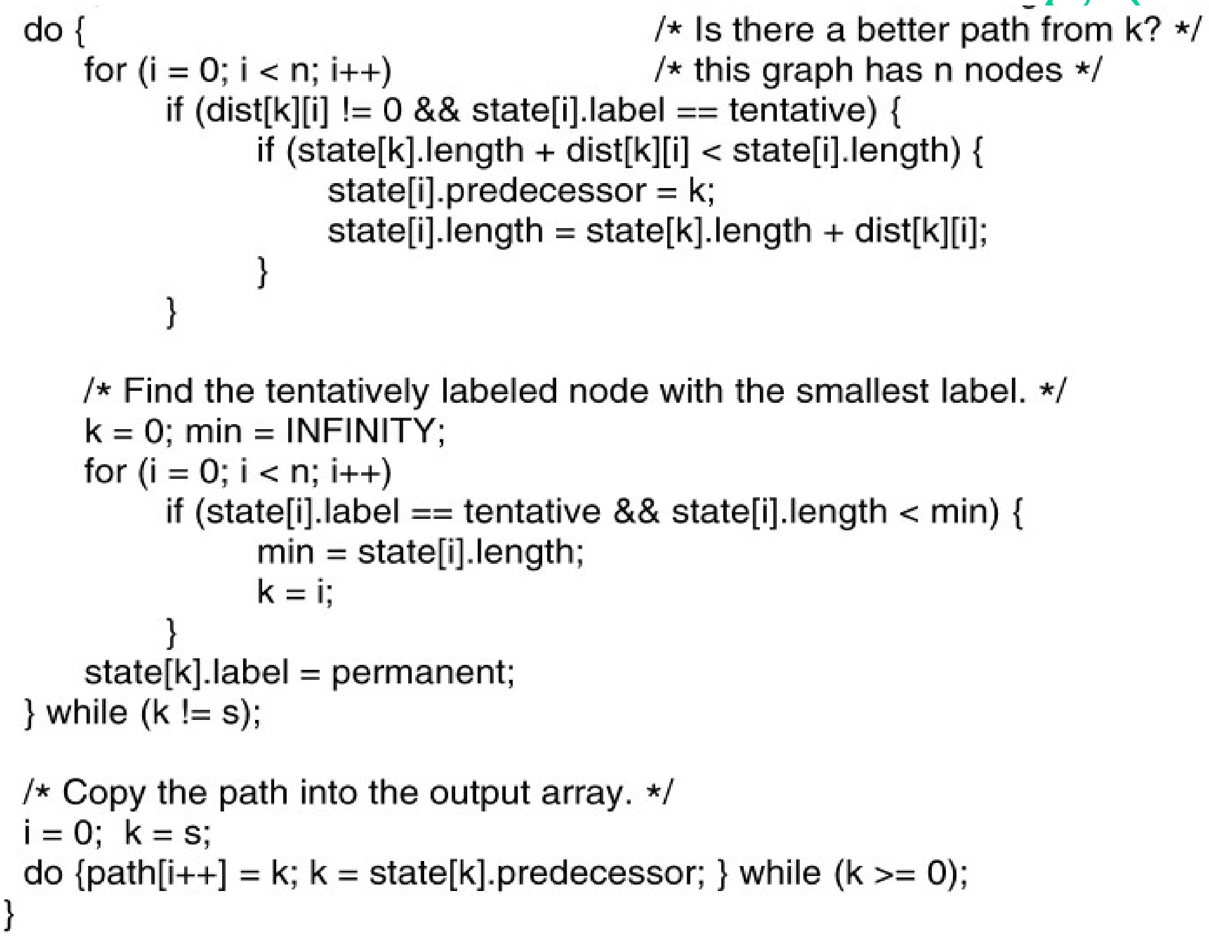


```
#define MAX  NODES 1024                   /* maximum number of nodes */
#define INFINITY 1000000000               /* a number larger than every maximum path */
int n, dist[MAX_NODES][MAX_NODES];/* dist[i][j] is the distance from i to j */

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;

  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 */
```

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

5.2 Routing Algorithms (18)

Shortest Path Routing (6)



```
    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];
                   }
             }

        /* 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.

5.2 Routing Algorithms (19)
Flooding (1)

- Another static algorithm is flooding, in which every incoming packet is sent out on every outgoing line except the one it arrived on.
- Flooding obviously generates vast numbers of duplicate packets, in fact, an infinite number unless some measures are taken to damp the process.

5.2 Routing Algorithms (20)
Flooding (2)

- One such measure is to have a hop counter contained in the header of each packet, which is decremented at each hop, with the packet being discarded when the counter reaches zero.
- Ideally, the hop counter should be initialized to the length of the path from source to destination.

5.2 Routing Algorithms (21)
Flooding (3)

- A variation of flooding that is slightly more practical is selective flooding.
- In this algorithm the routers do not send every incoming packet out on every line, only on those lines that are going approximately in the right direction.
- Flooding is not practical in most applications.

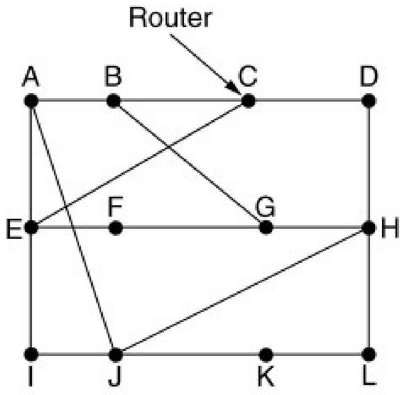
5.2 Routing Algorithms (22)
Distance Vector Routing (1)

- Distance Vector Routing is dynamic routing algorithm.
- Distance Vector Routing algorithms operate by having each router maintain a table (i.e., a vector) giving the best known distance to each destination and which line to use to get there.
- These tables are updated by exchanging information with the neighbors.

5.2 Routing Algorithms (23)
Distance Vector Routing (2)

- As an example, assume that delay is used as a metric and that the router knows the delay to each of its neighbors.
- Once every T msec each router sends to each neighbor a list of its estimated delays to each destination.
- It also receives a similar list from each neighbor.

5.2 Routing Algorithms (24)
Distance Vector Routing (3)



(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

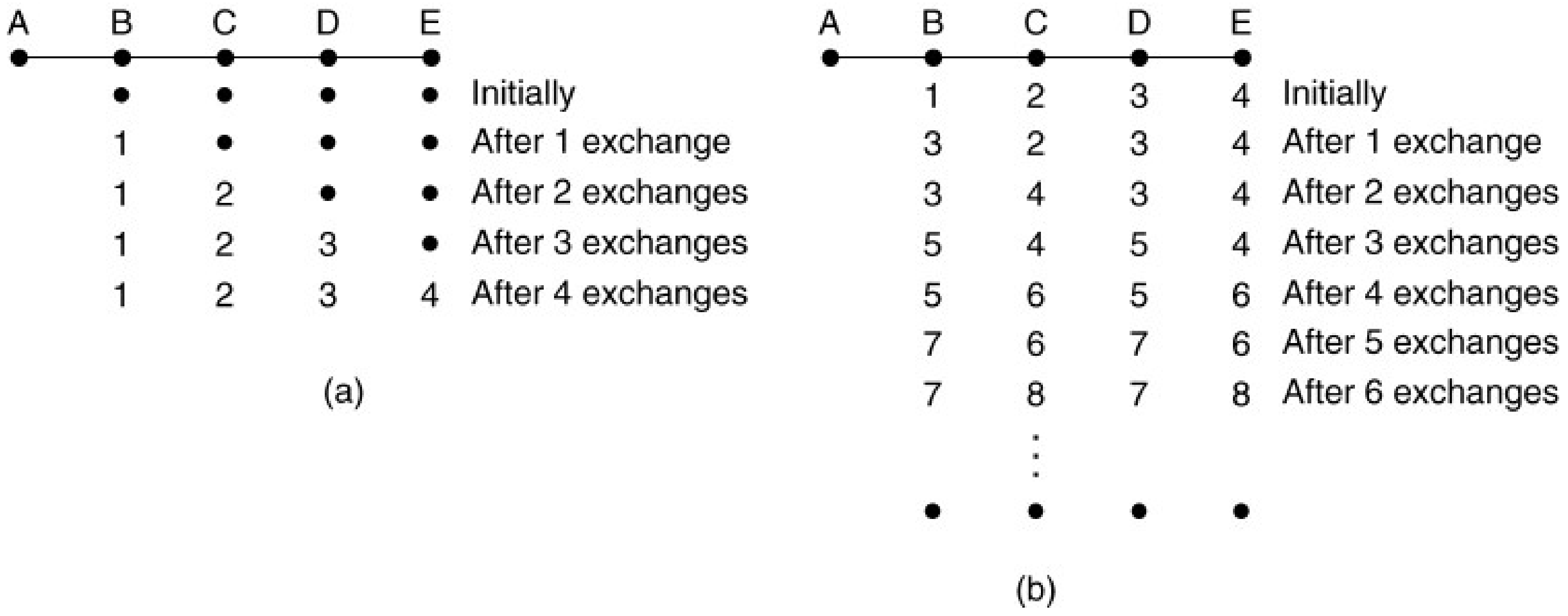
(b)

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

5.2 Routing Algorithms (25)
Distance Vector Routing (4)
The Count-to-Infinity Problem (1)

- Distance Vector Routing works in theory but has a serious drawback in practice.
- In particular, it reacts rapidly to good news, but leisurely to bad news.
- The core of the problem is that when X tells Y that it has a path somewhere, Y has no way of knowing whether it itself is on the path.

5.2 Routing Algorithms (26)
Distance Vector Routing (5)
The Count-to-Infinity Problem (2)



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.

5.2 Routing Algorithms (55)
Link State Routing (1)

- Two primary problems caused distance vector routing's demise.
- First, since the delay metric was queue length, it did not take line bandwidth into account when choosing routes.
- Second, the algorithm often took too long to converge (the count-to-infinity problem)

5.2 Routing Algorithms (56)

Link State Routing (2)

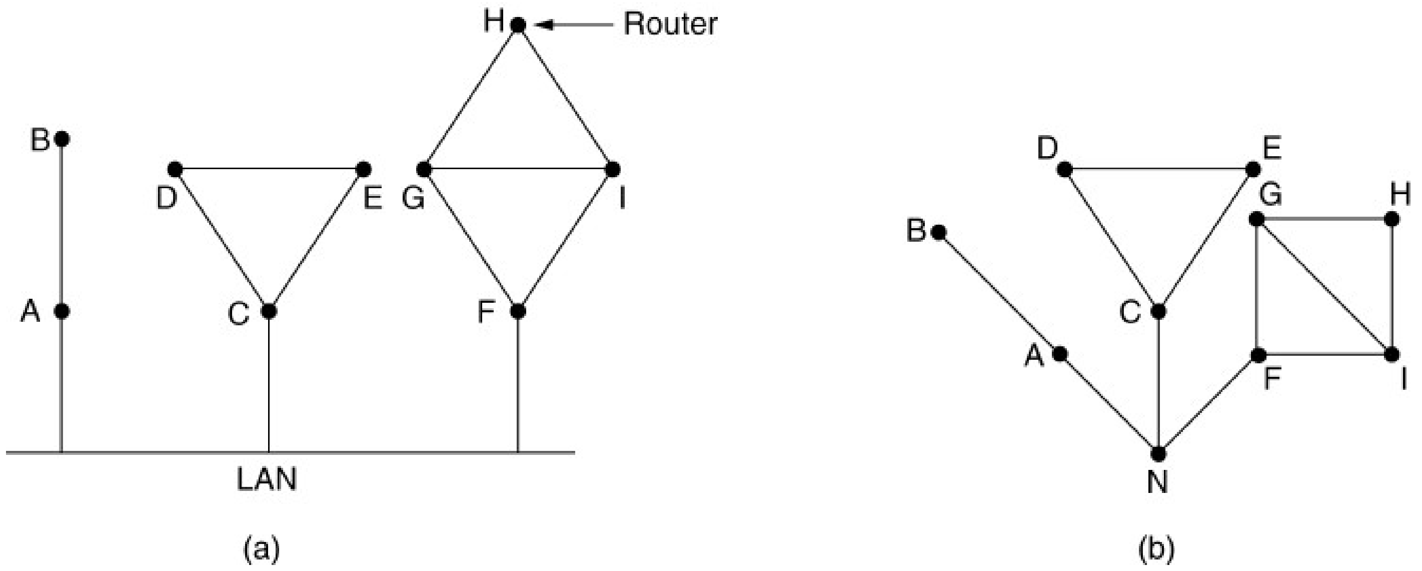
- Distance vector routing was replaced by Link State Routing
- Link State Routing is also dynamic routing algorithm.

5.2 Routing Algorithms (57)
Link State Routing (3)

Each router must do the following:

1. Discover its neighbors, learn their network address.
2. Measure the delay or cost to each of its neighbors.
3. Construct a packet telling all it has just learned.
4. Send this packet to all other routers.
5. Compute the shortest path to every other router.

5.2 Routing Algorithms (30)
Link State Routing (4)
Learning about the Neighbors

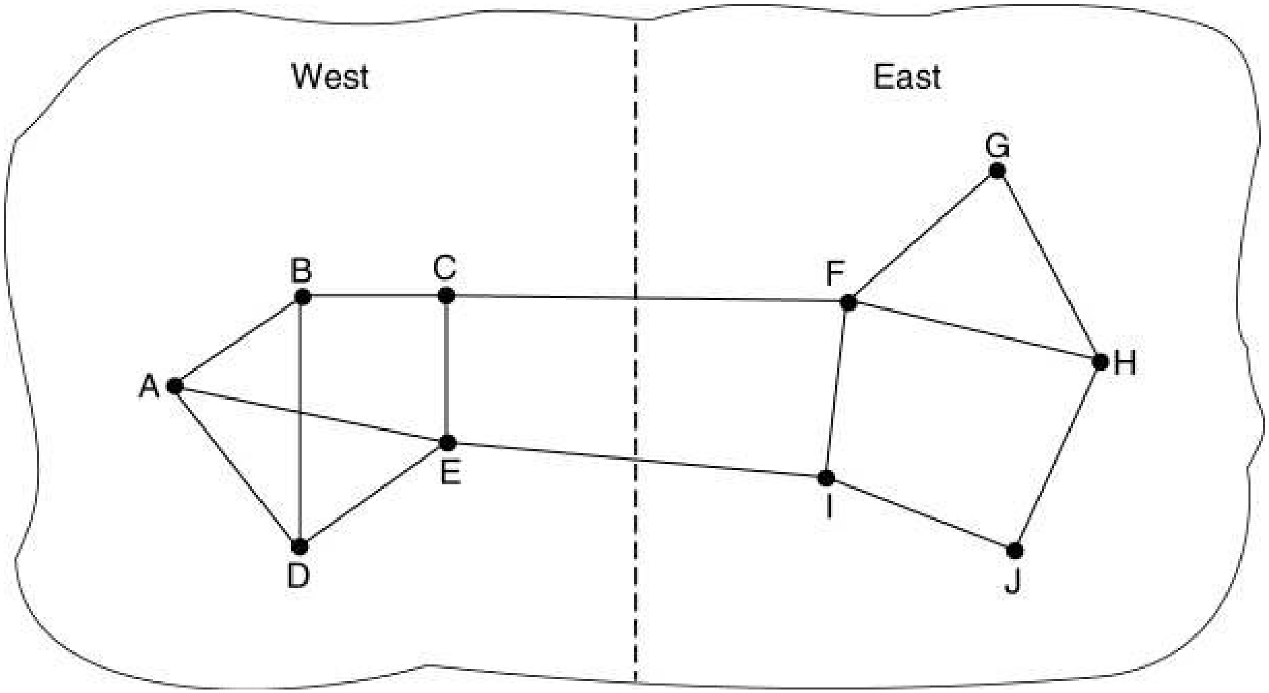


(a) Nine routers and a LAN. (b) A graph model of (a).

5.2 Routing Algorithms (31)

Link State Routing (5)

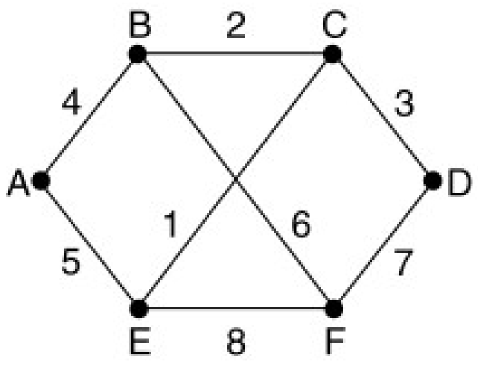
Measuring Line Cost



5.2 Routing Algorithms (32)

Link State Routing (6)

Building Link State Packets



(a)

Link State Packets

A	
Seq.	
Age	
B	4
E	5

B	
Seq.	
Age	
A	4
C	2
F	6

C	
Seq.	
Age	
B	2
D	3
E	1

D	
Seq.	
Age	
C	3
F	7

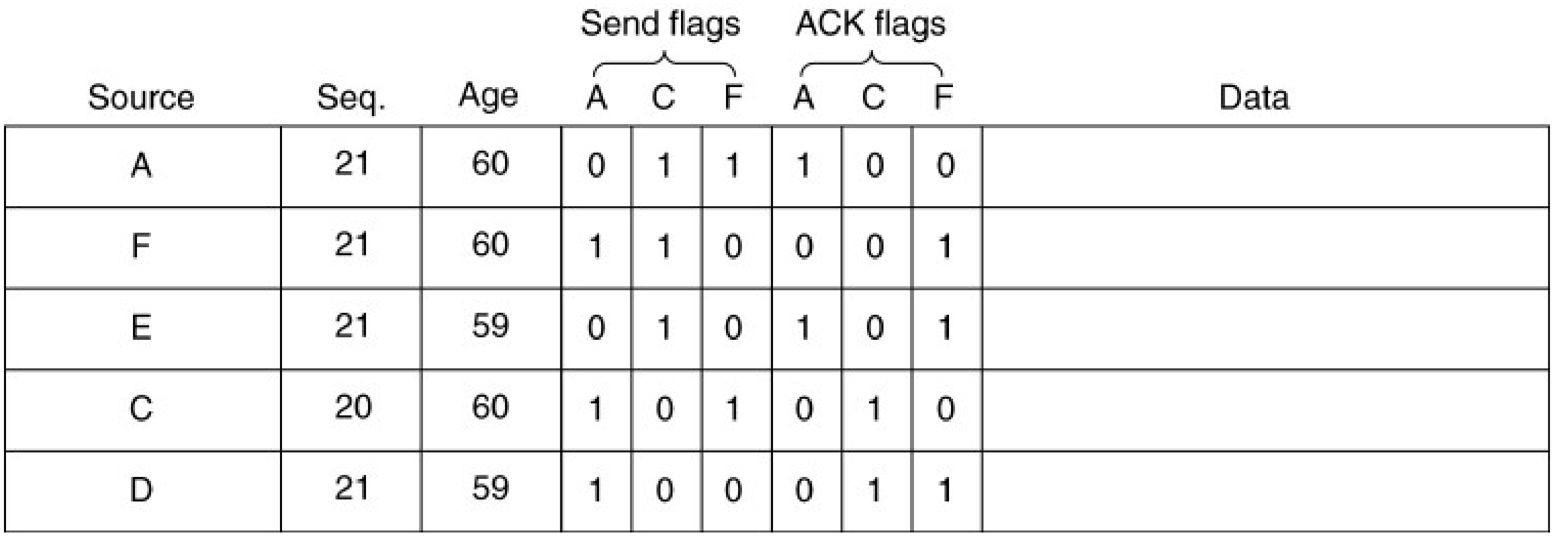
E	
Seq.	
Age	
A	5
C	1
F	8

F	
Seq.	
Age	
B	6
D	7
E	8

(b)

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

5.2 Routing Algorithms (61)
Link State Routing (7)
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	

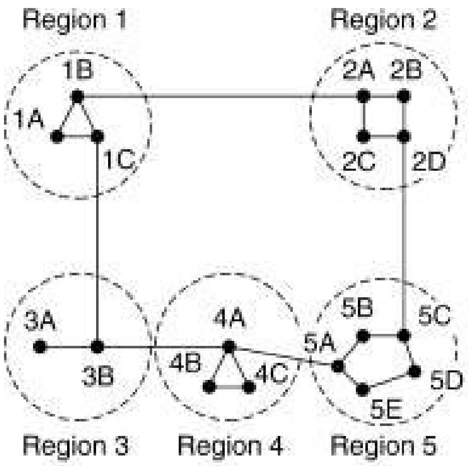
The packet buffer for router B in the previous slide

5.2 Routing Algorithms (62)
Link State Routing (8)
Computing the New Routes

- Once a router has accumulated a full set of link state packets, it can construct the entire subnet graph because every link is represented.
- Every link is, in fact, represented twice, once for each direction.

5.2 Routing Algorithms (63)

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

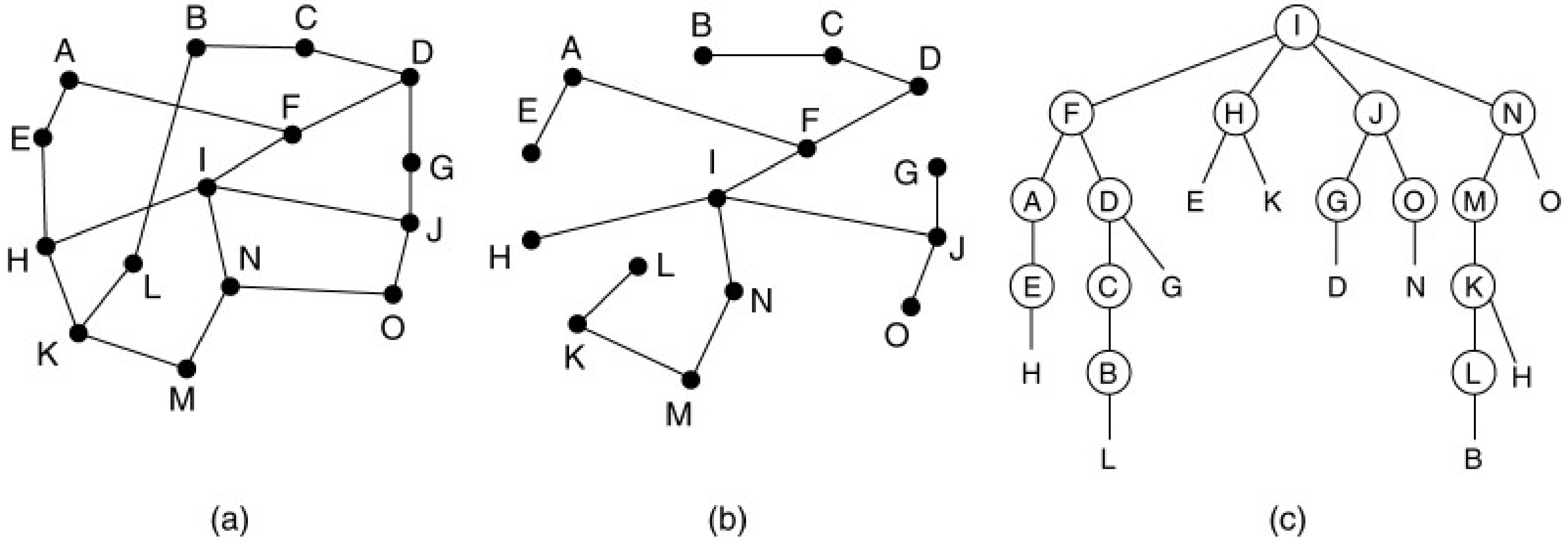
(b)

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

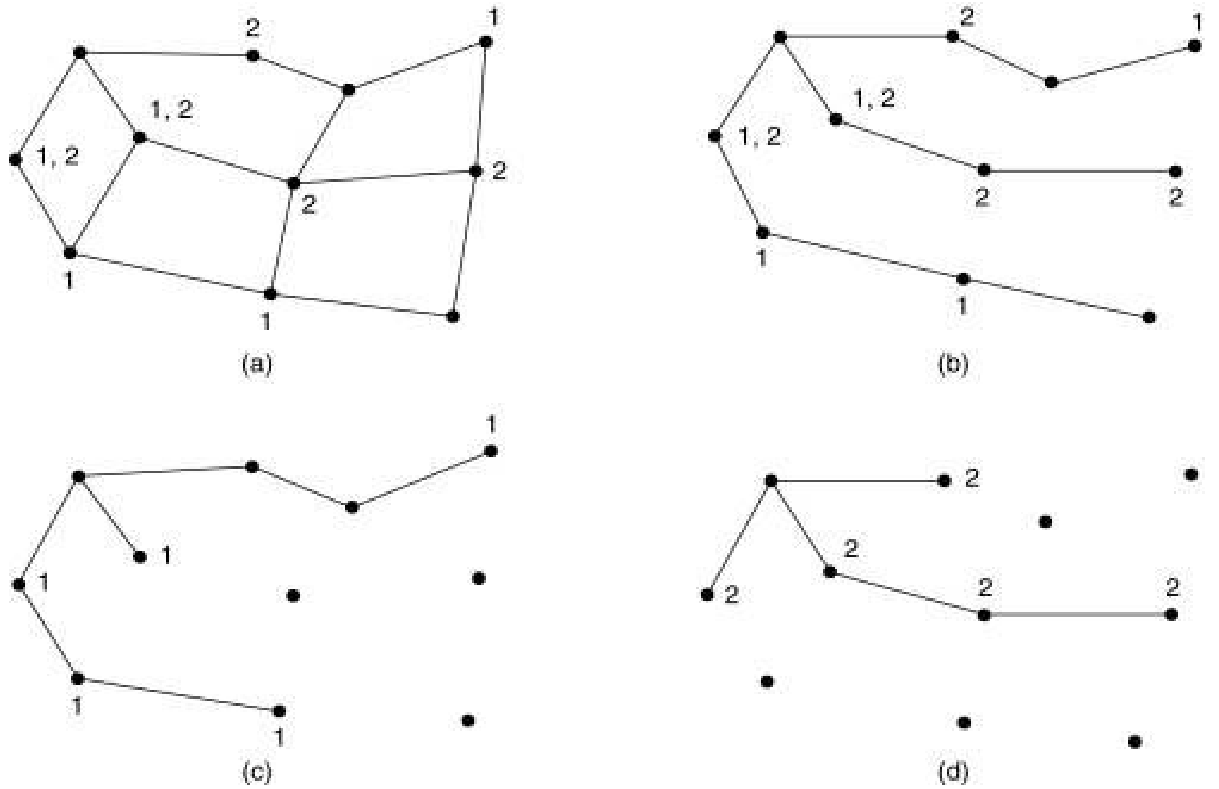
(c)

5.2 Routing Algorithms (64)
Broadcast Routing



Reverse path forwarding. (a) A subnet. (b) a Sink tree. (c) The tree built by reverse path forwarding.

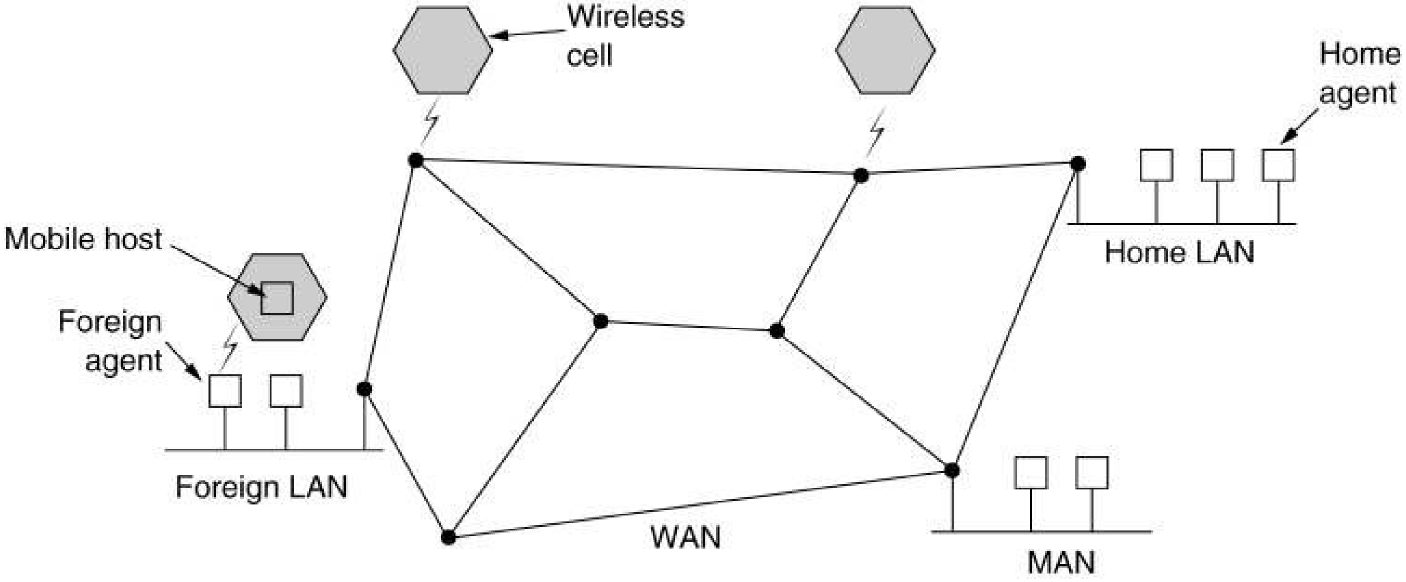
5.2 Routing Algorithms (65)
Multicast Routing



(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.

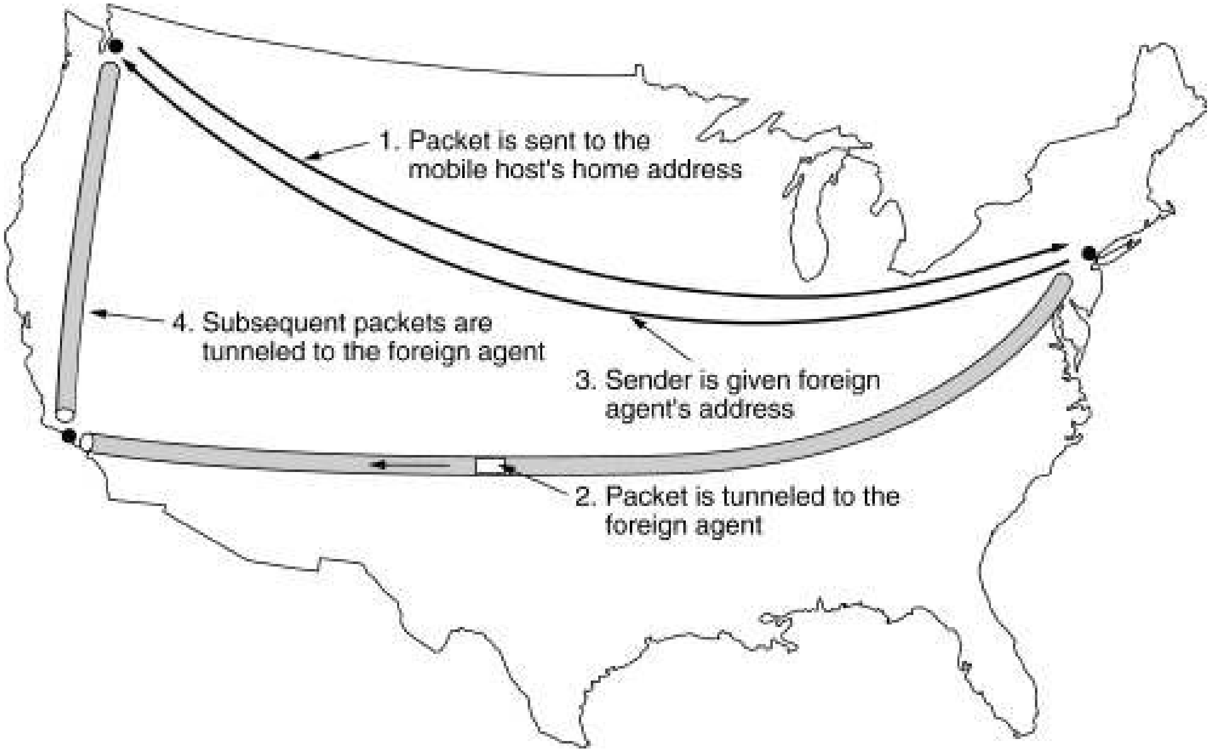
5.2 Routing Algorithms (66)

Routing for Mobile Hosts (1)



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

5.2 Routing Algorithms (67)
Routing for Mobile Hosts (2)


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.

5.2 Routing Algorithms (68)

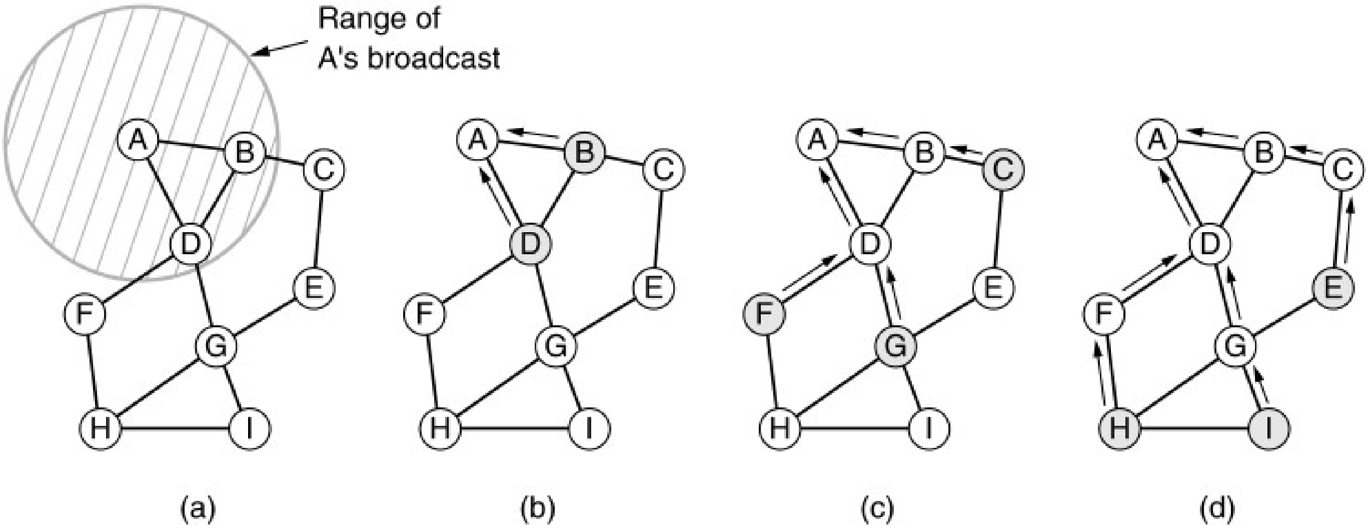
Routing in Ad Hoc Networks

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 works at earthquake .
   - The infrastructure destroyed.
4. A gathering of people with notebook computers.
   - In an area lacking 802.11.

5.2 Routing Algorithms (41)

Route Discovery (1)



a) Range of A's broadcast.

b) After B and D have received A's broadcast.

c) After C, F, and G have received A's broadcast.

d) After E, H, and I have received A's broadcast.

Shaded nodes are new recipients. Arrows show possible reverse routes.

5.2 Routing Algorithms (70)

Route Discovery (2)

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

Format of a ROUTE REQUEST packet.

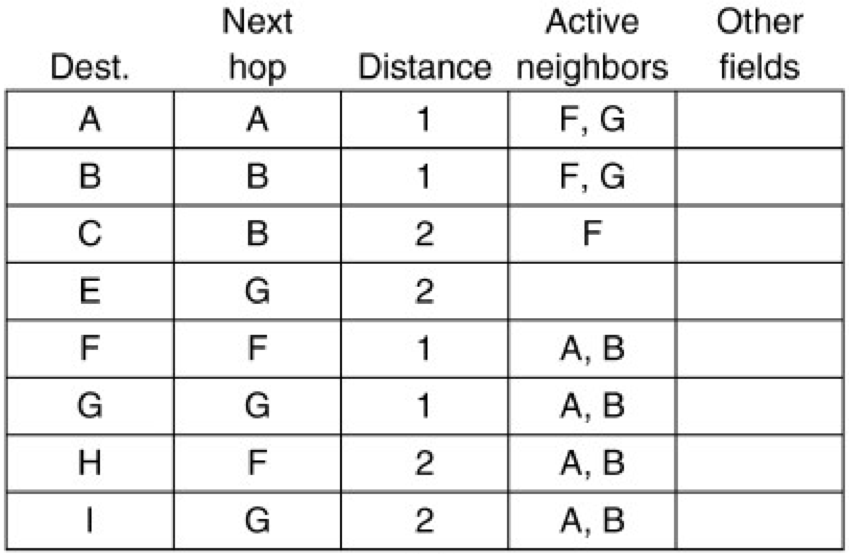
5.2 Routing Algorithms (43)

Route Discovery (3)

Source address	Destination address	Destination sequence #	Hop count	Lifetime

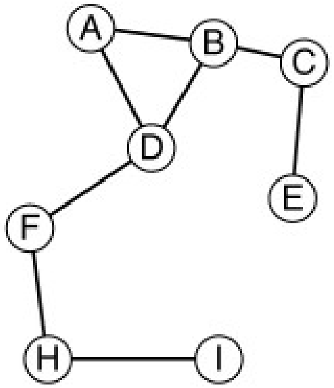
Format of a ROUTE REPLY packet.

5.2 Routing Algorithms (44)
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)



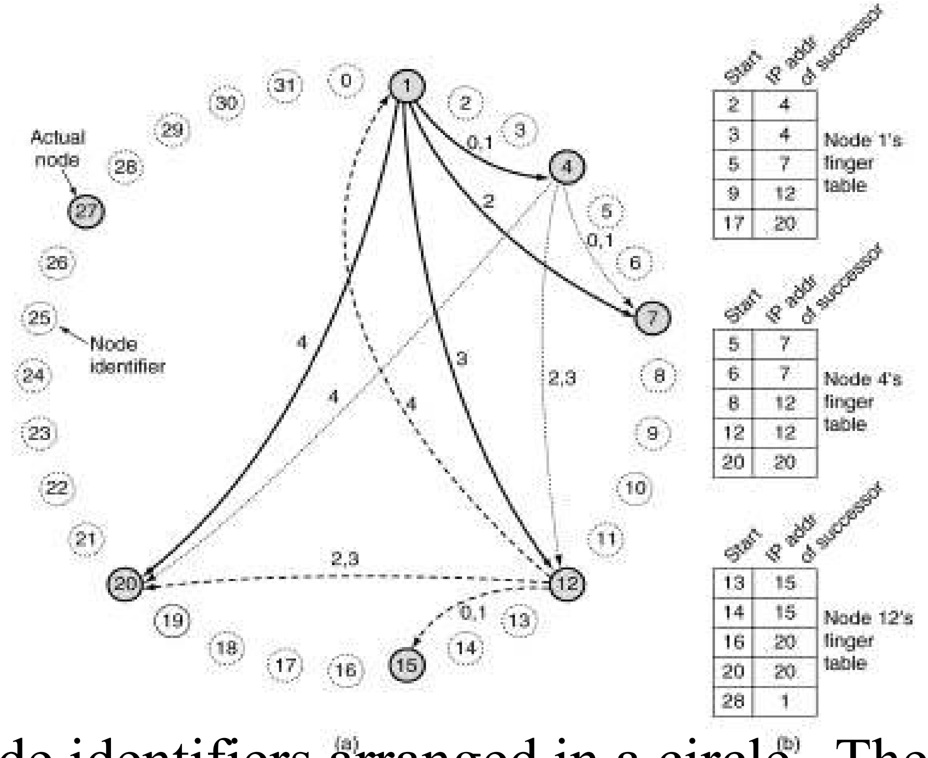
(b)

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

(b) The graph after G has gone down.

5.2 Routing Algorithms (45)

Node Lookup in Peer-to-Peer Networks



Start	IP addr of successor
2	4
3	4
5	7
9	12
17	20

Node 1's finger table

Start	IP addr of successor
5	7
6	7
8	12
12	12
20	20

Node 4's finger table

Start	IP addr of successor
13	15
14	15
16	20
20	20
28	1

Node 12's finger table

(a) A set of 32 node identifiers arranged in a circle. The shaded ones correspond to actual machines. The arcs show the fingers from nodes 1, 4, and 12. The labels on the arcs are the table indices.

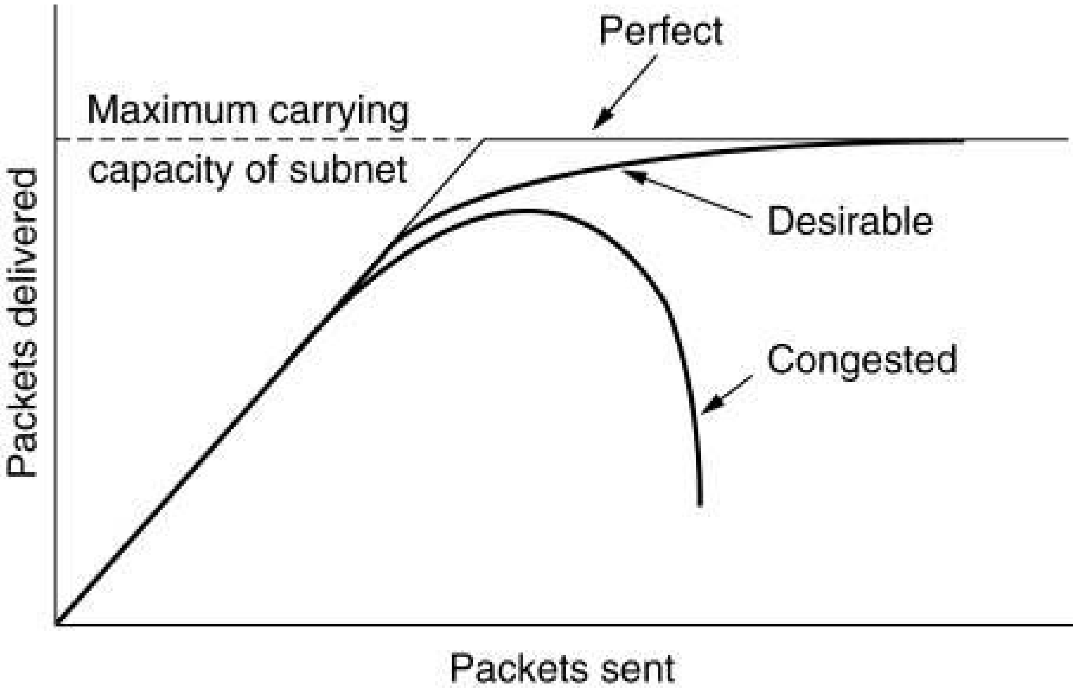
(b) Examples of the finger tables.

5.3. Congestion Control Algorithms (1)

- When too many packets are present in (a part of) the subnet, performance degrades.
- This situation is called congestion.

5.3. Congestion Control Algorithms (2)

Congestion



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

5.3. Congestion Control Algorithms (76)

- Congestion can be brought on by several factors.
- If all of a sudden, streams of packets begin arriving on three or four input lines and all need the same output line, queue will build up.
- If there is insufficient memory to hold all of them, packets will be lost.
- Slow processors can also cause congestion

5.3. Congestion Control Algorithms (77)

- General Principles of Congestion Control
- Congestion Prevention Policies
- Congestion Control in Virtual-Circuit Subnets
- Congestion Control in Datagram Subnets
- Load Shedding
- Jitter Control

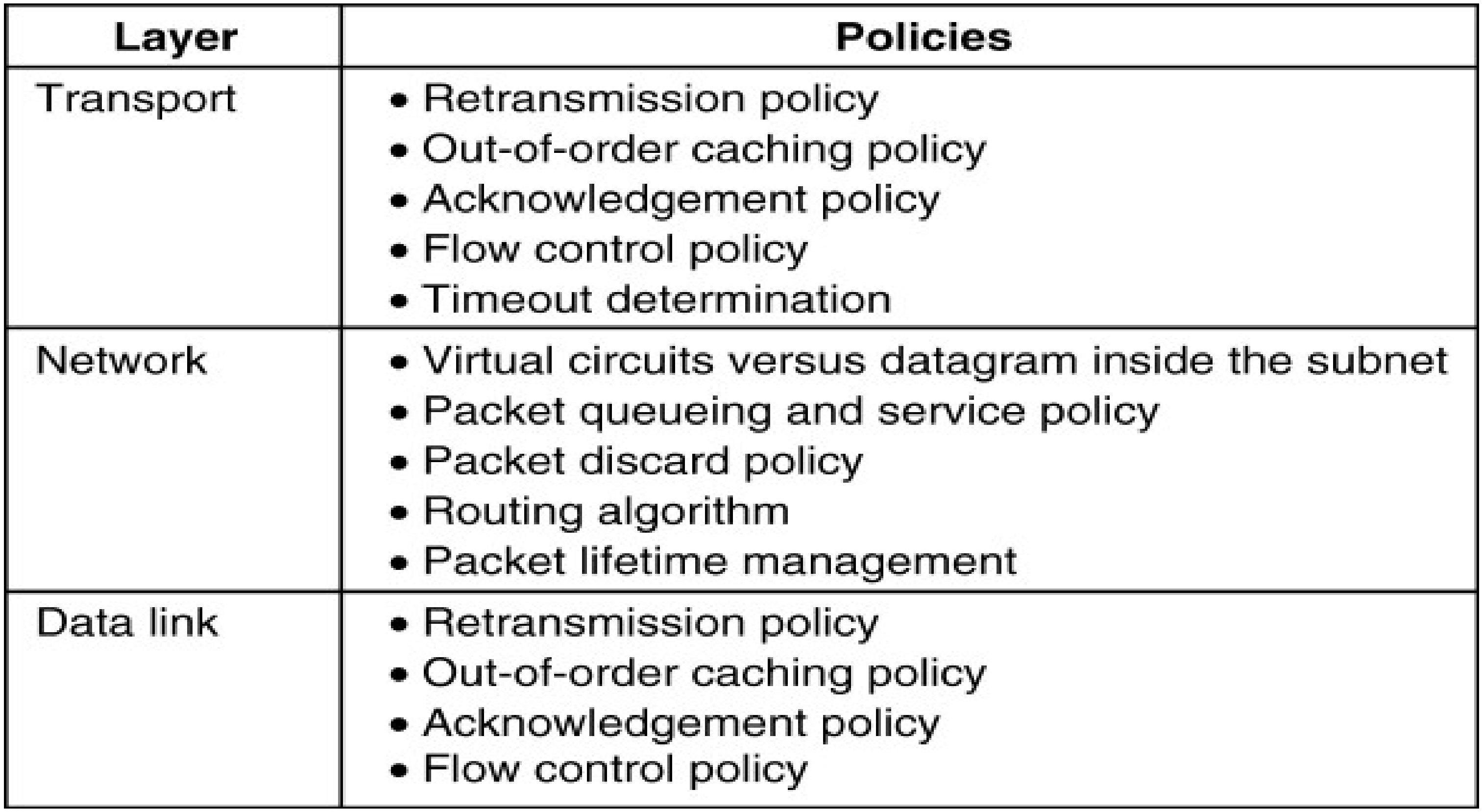
5.3. Congestion Control Algorithms ()

General Principles of Congestion Control

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

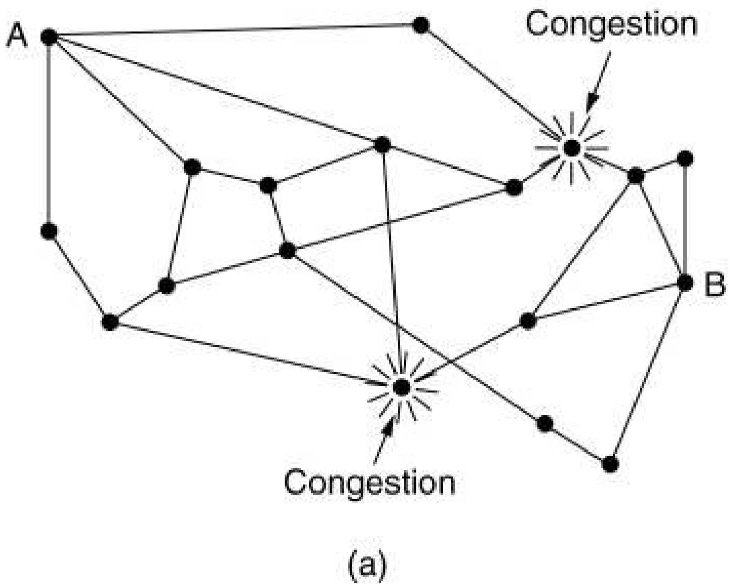
5.3. Congestion Control Algorithms (79)

Congestion Prevention Policies



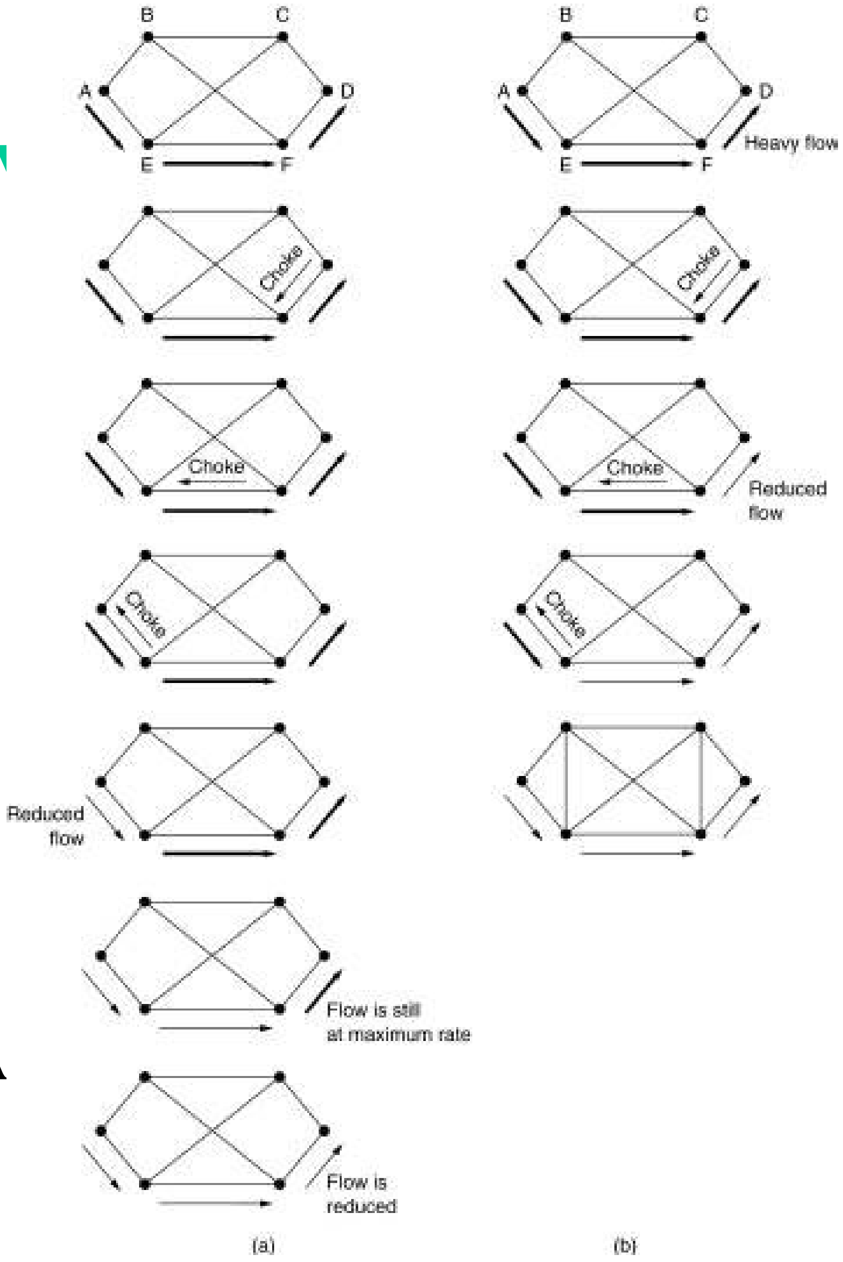
Layer	Policies
Transport	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy • Timeout determination
Network	• Virtual circuits versus datagram inside the subnet • Packet queueing and service policy • Packet discard policy • Routing algorithm • Packet lifetime management
Data link	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy

Congestion Control in '



(a) A congested subnet. (b) A congestion and a virtual

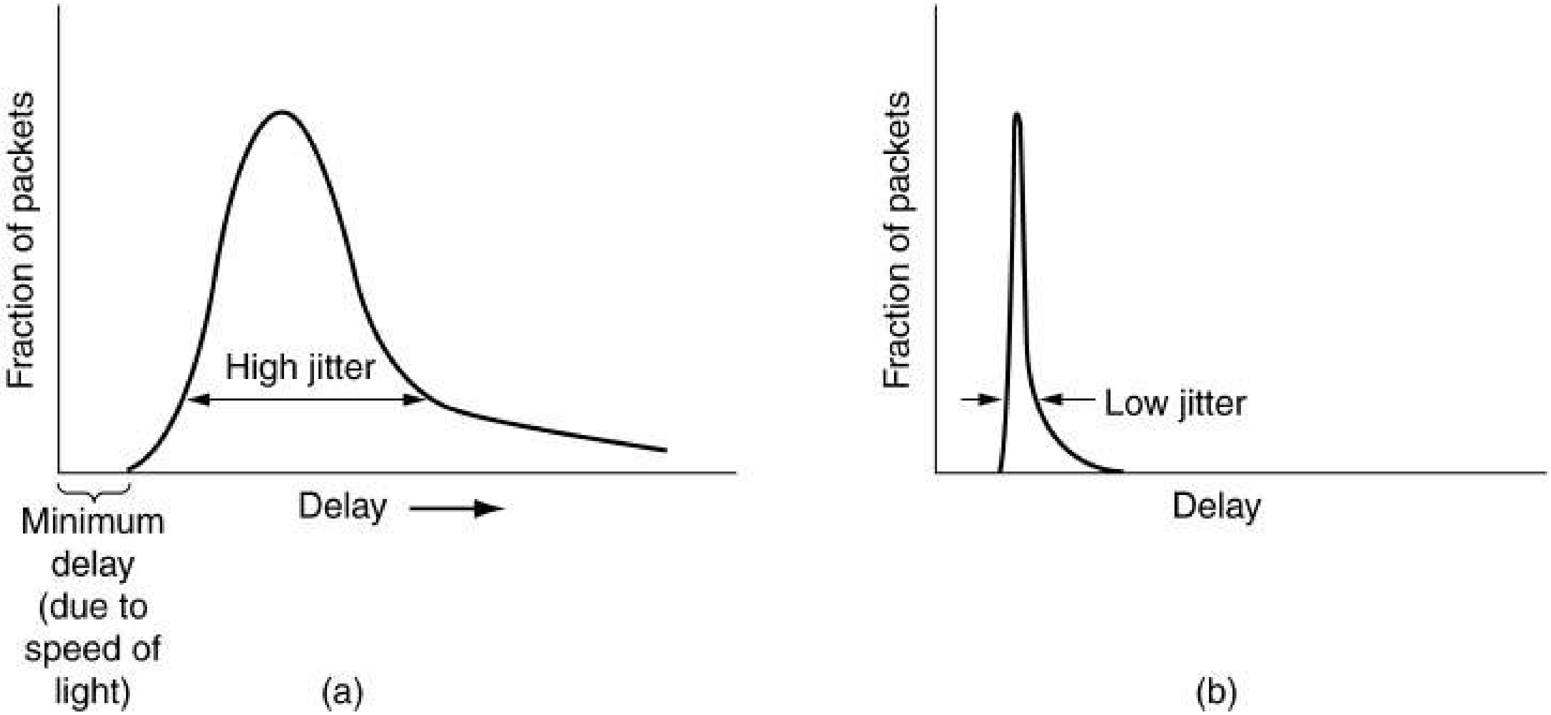
Hop-by-Hop Choke Packets



(a) A choke packet that affects
only the source.

(b) A choke packet that affects
each hop it passes through.

Jitter Control



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

The Network Layer

5.1. NETWORK LAYER DESIGN ISSUES

5.2. ROUTING ALGORITHMS

5.3. CONGESTION CONTROL ALGORITHMS

5.4. QUALITY OF SERVICE

5.5. INTERNETWORKING

5.6. THE NETWORK LAYER IN THE INTERNET

5.7. SUMMARY

5.4 Quality of Service

- The next step beyond just dealing with congestion is to actually try to achieve a promised quality of service.
- The methods that can be used for this include buffering at the client, traffic shaping, resource reservation, and admission control.

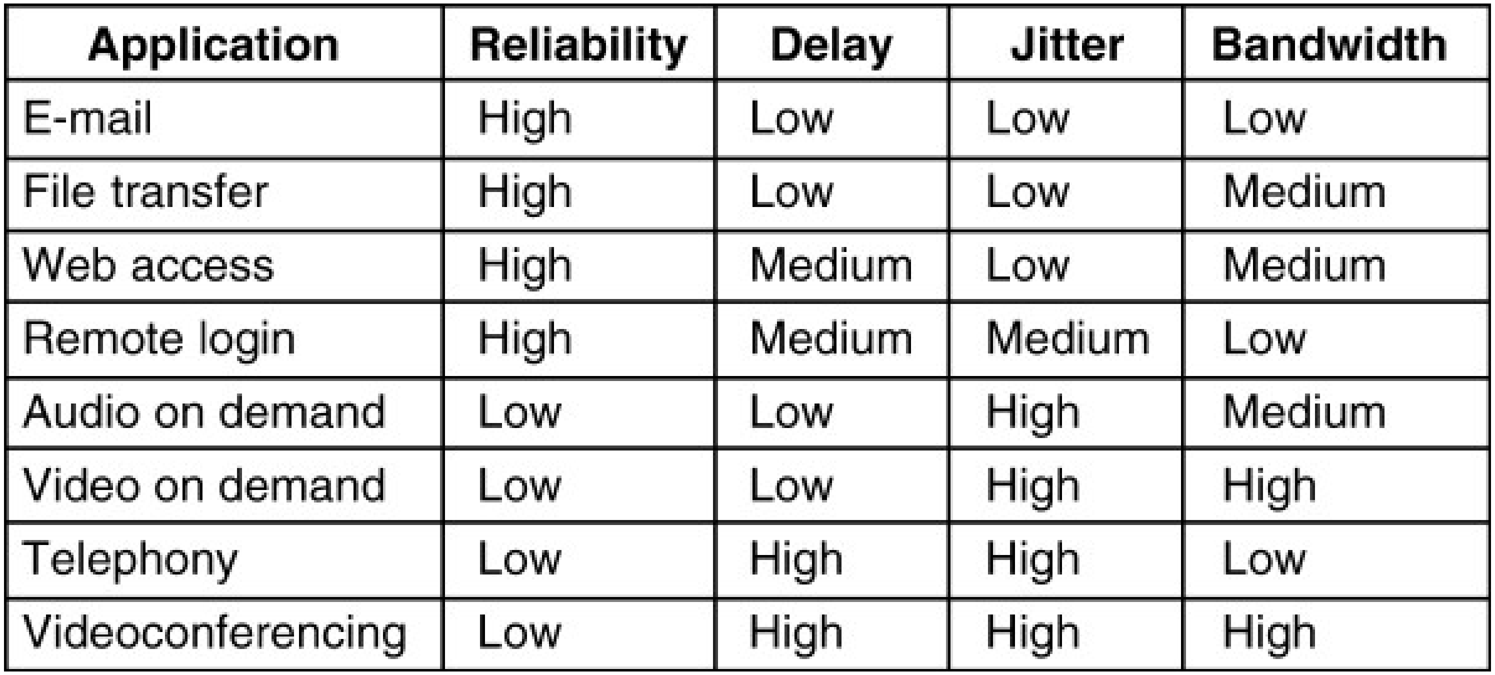
5.4 Quality of Service

- Requirements
- Techniques for Achieving Good Quality of Service
- Integrated Services
- Differentiated Services
- Label Switching and MPLS

5.4 Quality of Service

- A stream of packets from source to a destination is called flow.
- In connection- oriented network, all the packets belonging to a flow follow the same route.
- In connectionless network, all the packets belonging to flow may follow different route.
- Reliability, delay, jitter, and bandwidth are main characteristics of the flows.
- Together these characteristics determine the QoS (Quality of Service) the flow requires.

5.4 Quality of Service Requirements



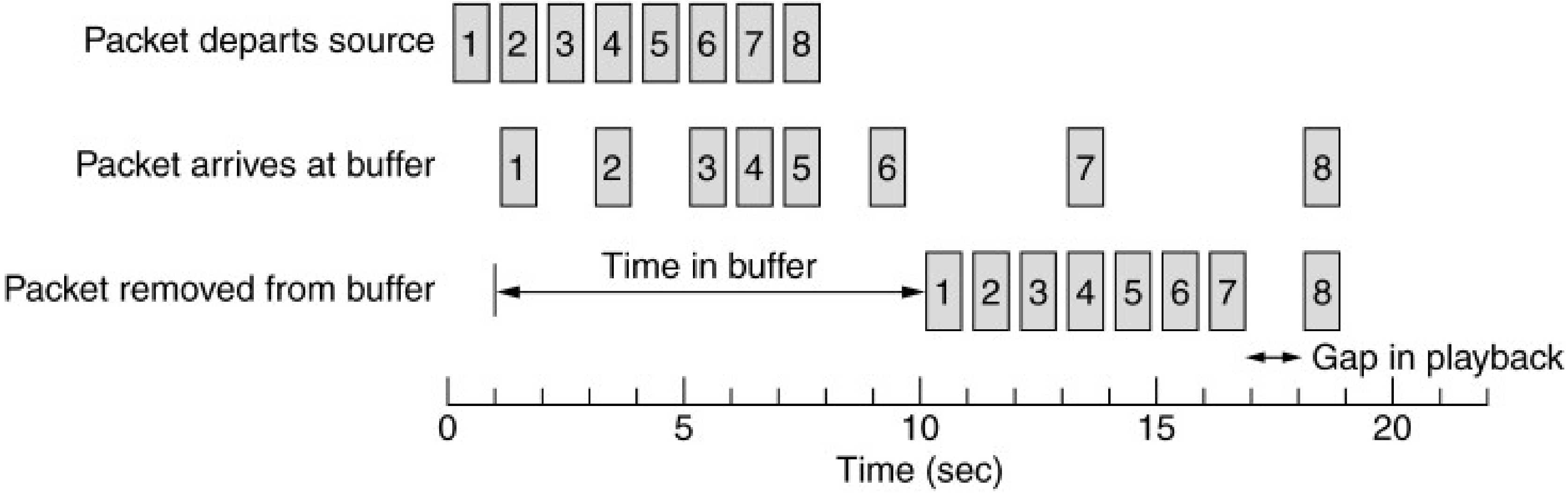
Application	Reliability	Delay	Jitter	Bandwidth
E-mail	High	Low	Low	Low
File transfer	High	Low	Low	Medium
Web access	High	Medium	Low	Medium
Remote login	High	Medium	Medium	Low
Audio on demand	Low	Low	High	Medium
Video on demand	Low	Low	High	High
Telephony	Low	High	High	Low
Videoconferencing	Low	High	High	High

How stringent the quality-of-service requirements are.

5.4 Quality of Service Overprovisioning

- An easy solution is to provide so much router capacity, buffer space, and bandwidth that the packets just fly through easily.
- For example, the telephone system.
- This solution is very expensive.

5.4 Quality of Service
Buffering



- Flows can be buffered on the receiving side before being delivered.
- Buffering does not affect the reliability or bandwidth, and increases the delay, but it smoothes out the jitter.
- Smoothing the output stream by buffering packets.

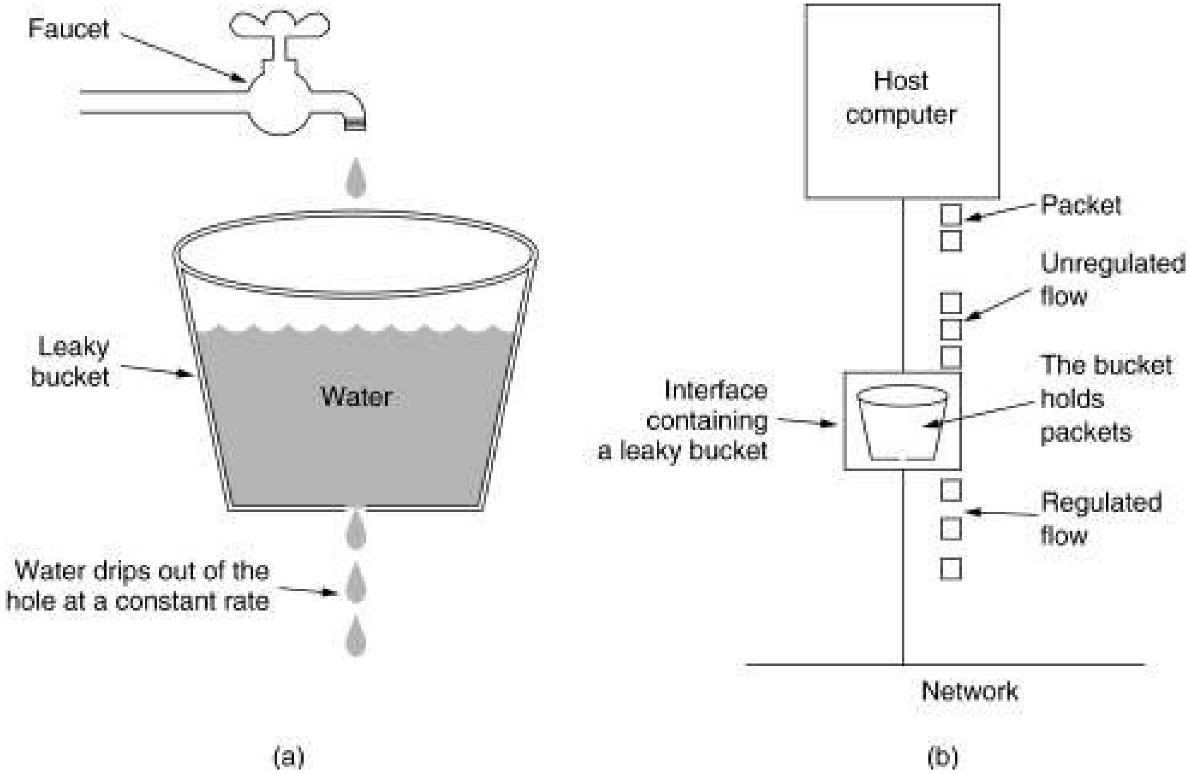
Traffic Shading

- Nonuniform output is common if the server is handling many streams at once, and it also allows other actions, such as fast forward and rewind, user authentication, etc.
- Traffic Shading smoothes out the traffic on the server side, rather than on the client side.
- This method is about regulating the average (and burstiness) of data transmission.

5.4 Quality of Service The Leaky Bucket Algorithm

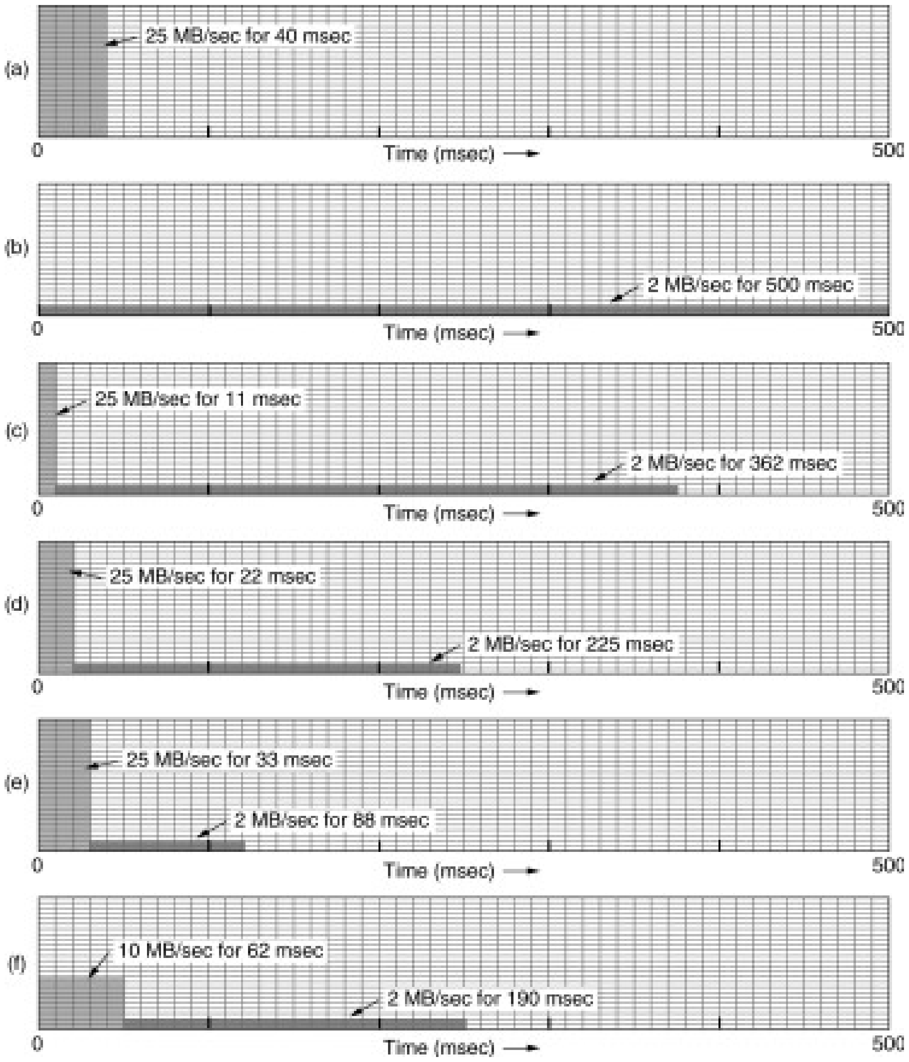
- Each host is connected to the network by an interface containing a leaky bucket, that is, a final internal queue.
- If a packet arrives at the queue when it is full, the packet is discarded.
- This is called the leaky bucket algorithm.
- In fact it is nothing other than a single server queuing system with constant service time.

5.4 Quality of Service
The Leaky Bucket Algorithm



(a) A leaky bucket with water. (b) A leaky bucket with packets.

5.4 Quality of Service The Leaky Bucket Algorithm

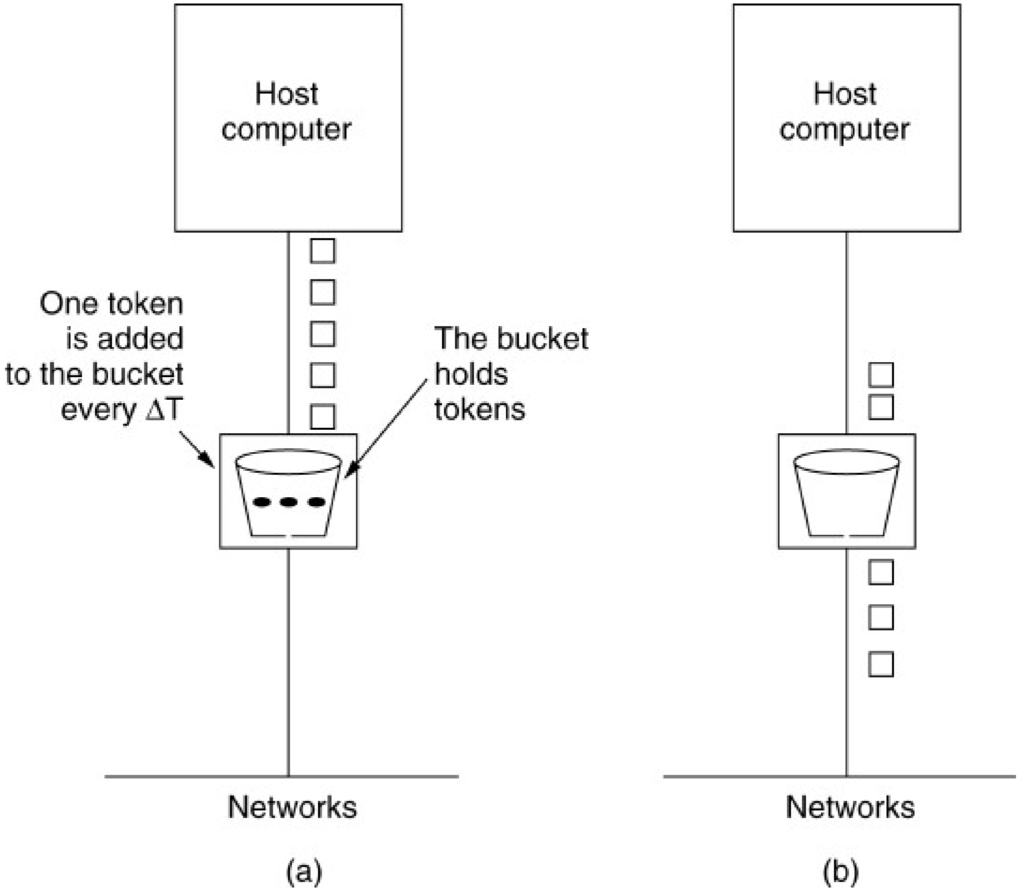


(a) Input to a leaky bucket. (b) Output from a leaky bucket. Output from a token bucket with capacities of (c) 250 KB, (d) 500 KB, (e) 750 KB, (f) Output from a 500KB token bucket feeding a 10-MB/sec leaky bucket.

5.4 Quality of Service The Token Bucket Algorithm

- The leaky bucket algorithm enforces a rigid output pattern at the average rate, no matter how bursty the traffic is.
- For many applications, it is better to allow the output to speed up somewhat when large bursts arrive, so a more flexible algorithm is needed, preferably one that never loses data.
- One is the token bucket algorithm.

5.4 Quality of Service
The Token Bucket Algorithm



(a) Before. (b) After.

5.4 Quality of Service
The Token Bucket Algorithm

- The token bucket algorithm does allow idle hosts to save up permission to send large bursts, up to the maximum size of the bucket, *n*, later.
- The token bucket algorithm also throws away tokens (i.e. transmission capacity) when the bucket fills up but never discards packets.

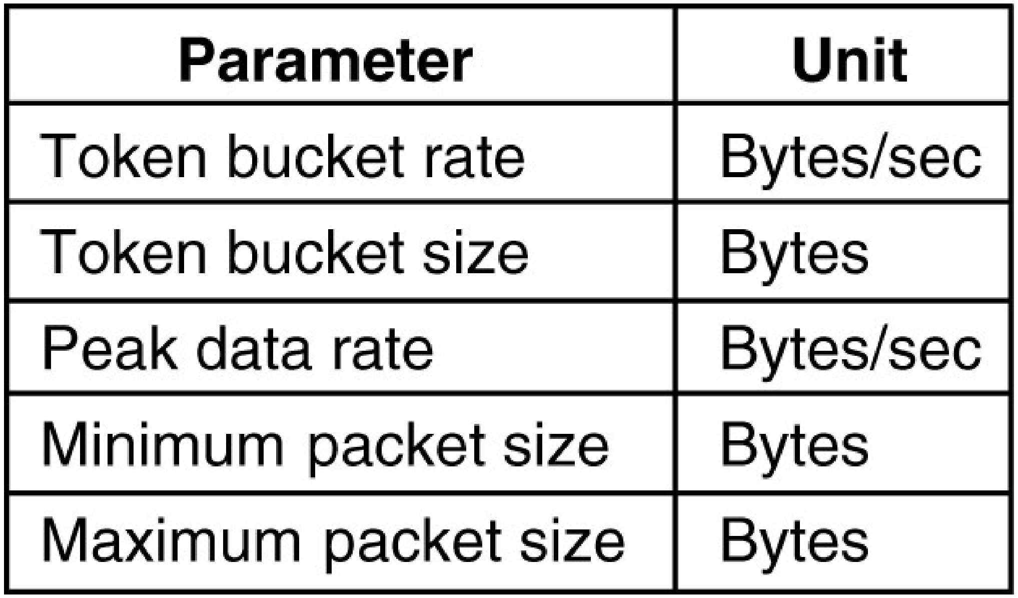
5.4 Quality of Service Resource Reservation

- In order to guaranty the quality of service all the packets of a flow must follow the same route.
- For this purpose it is necessary to reserve resources along that route.
- 1) Bandwidth
- 2) Buffer space
- 3) CPU cycles.

5.4 Quality of Service Admission Control

- The decision to accept or reject a flow is not a simple matter of comparing the (bandwidth, buffers, cycles) requested by the flow with the routers' excess capacity in those three dimensions.
- It is a little more complicated than that.
- Flows must be described accurately in terms of flow specification.

5.4 Quality of Service
Admission Control



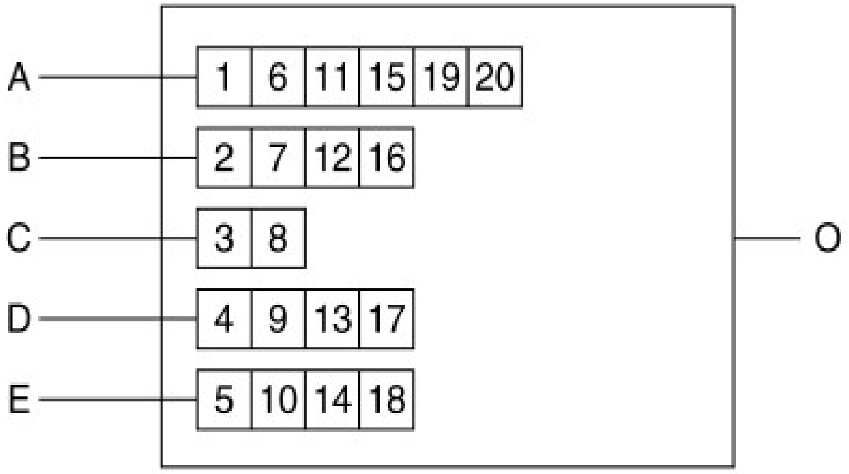
Parameter	Unit
Token bucket rate	Bytes/sec
Token bucket size	Bytes
Peak data rate	Bytes/sec
Minimum packet size	Bytes
Maximum packet size	Bytes

An example of flow specification.

5.4 Quality of Service
Packet Scheduling

- If a router is handling multiple flows, there is a danger that one flow will hog too much of its capacity and starve all the other flows.
- Processing packets in the order of their arrival means that an aggressive sender can capture most of the capacity of the routers its packets traverse, reducing the quality of service for others.

5.4 Quality of Service
Packet Scheduling



(a)

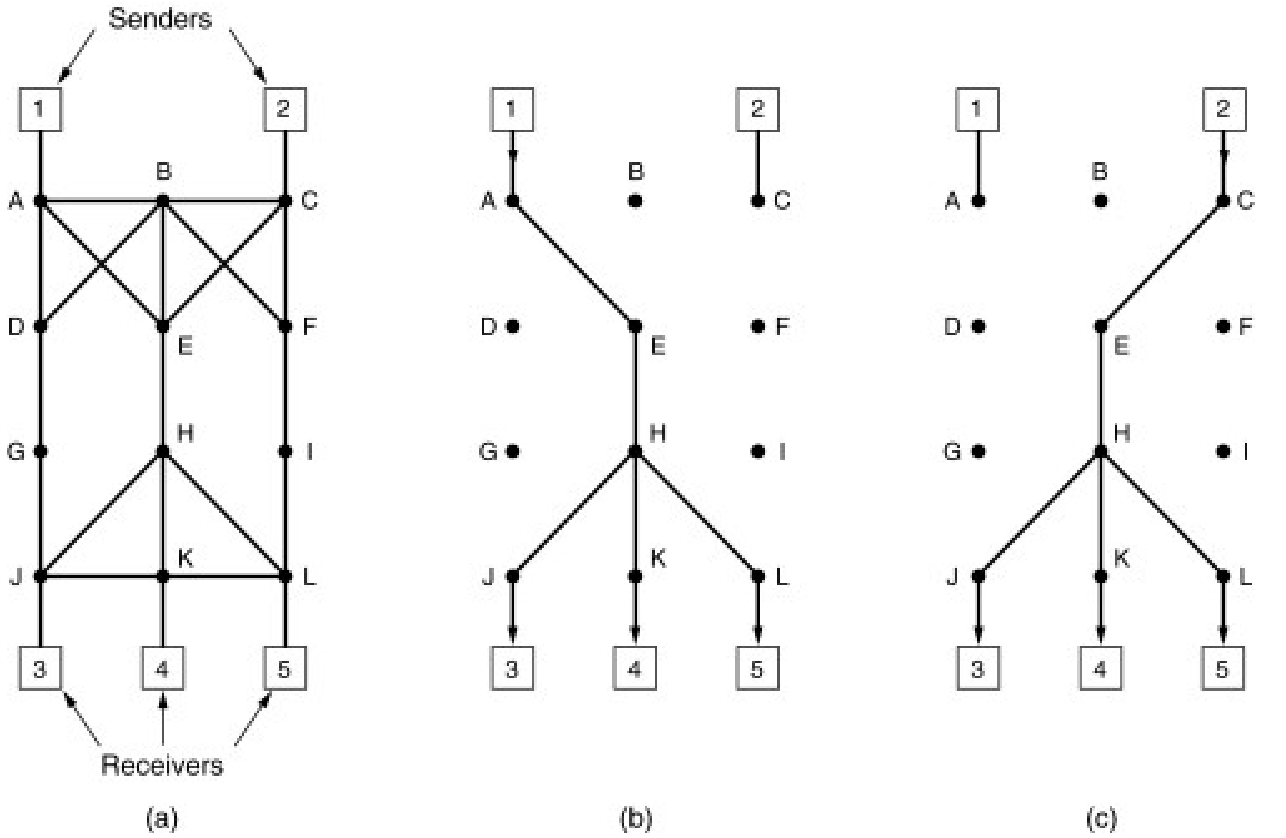
Packet	Finishing time
C	8
B	16
D	17
E	18
A	20

(b)

(a) A router with five packets queued for line O.

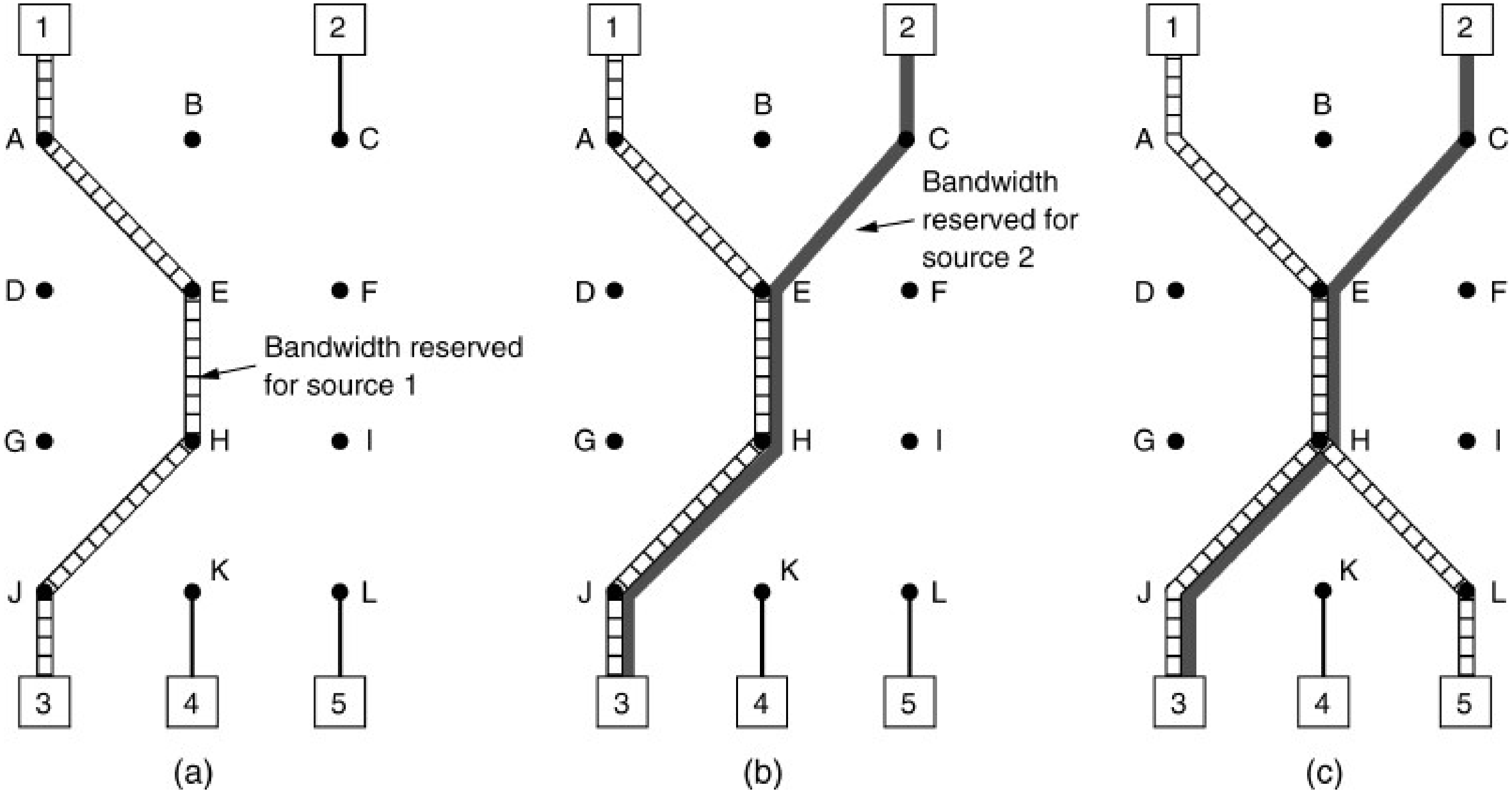
(b) Finishing times for the five packets.

5.4 Quality of Service
RSVP-The ReSerVation Protocol



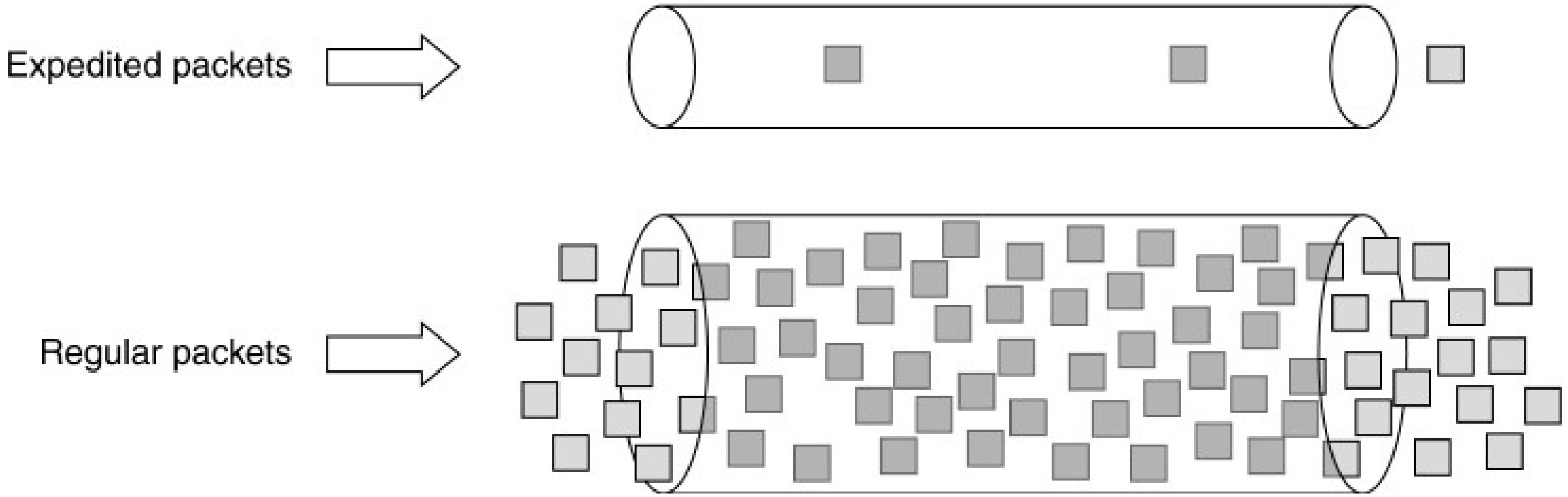
(a) A network, (b) The multicast spanning tree for host 1.
(c) The multicast spanning tree for host 2.

5.4 Quality of Service
RSVP-The ReSerVation Protocol (2)


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

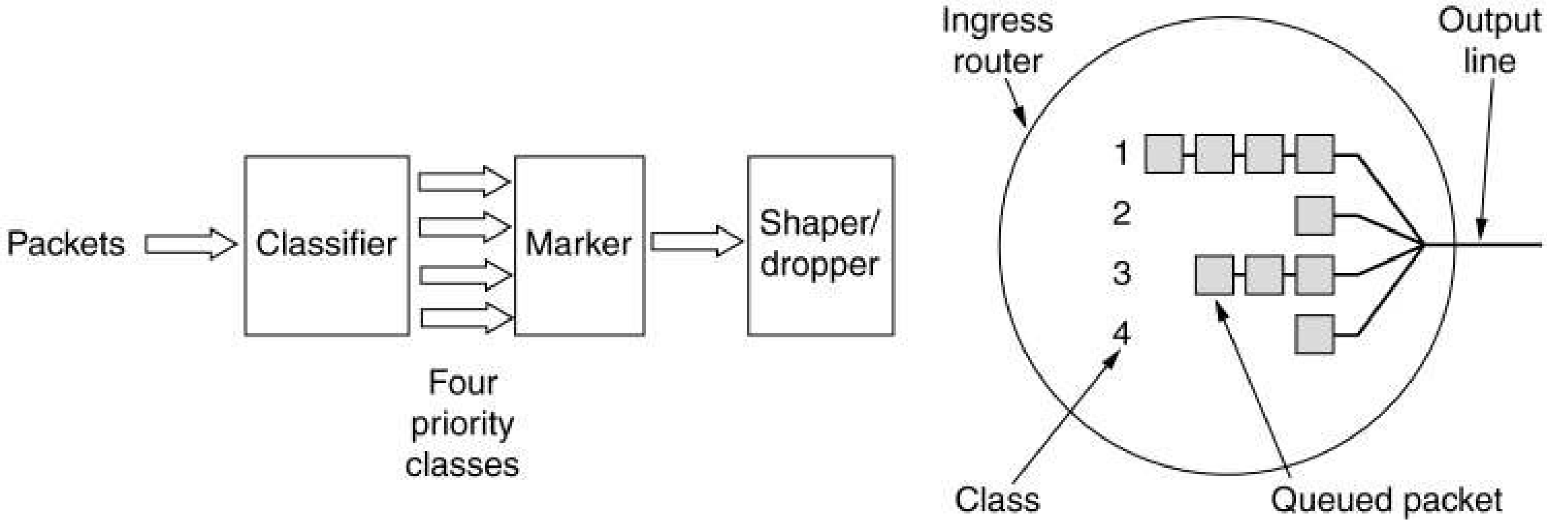
(a) Host 3 requests a channel to host 1. (b) Host 3 then requests a second channel, to host 2. (c) Host 5 requests a channel to host 1.

5.4 Quality of Service
Expedited Forwarding



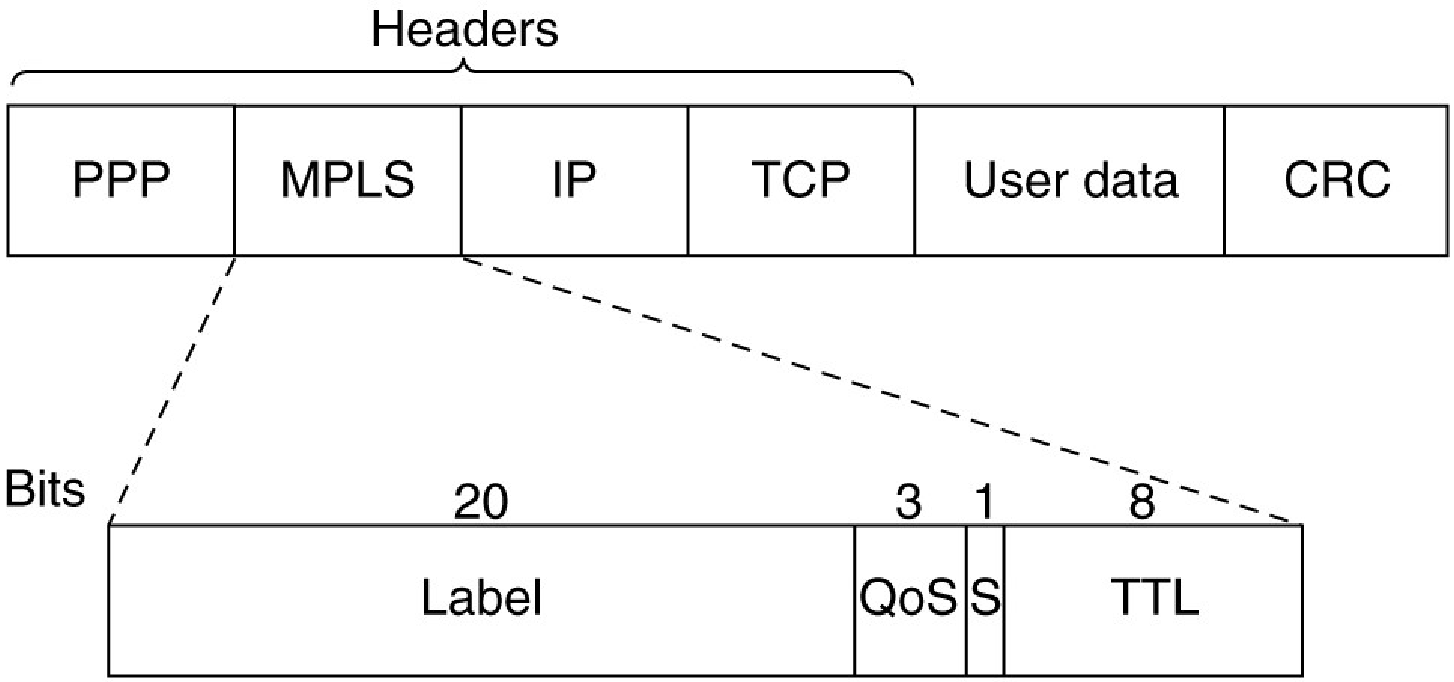
Expedited packets experience a traffic-free network.

5.4 Quality of Service
Assured Forwarding



A possible implementation of the data flow for assured forwarding.

5.4 Quality of Service
Label Switching and MPLS



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

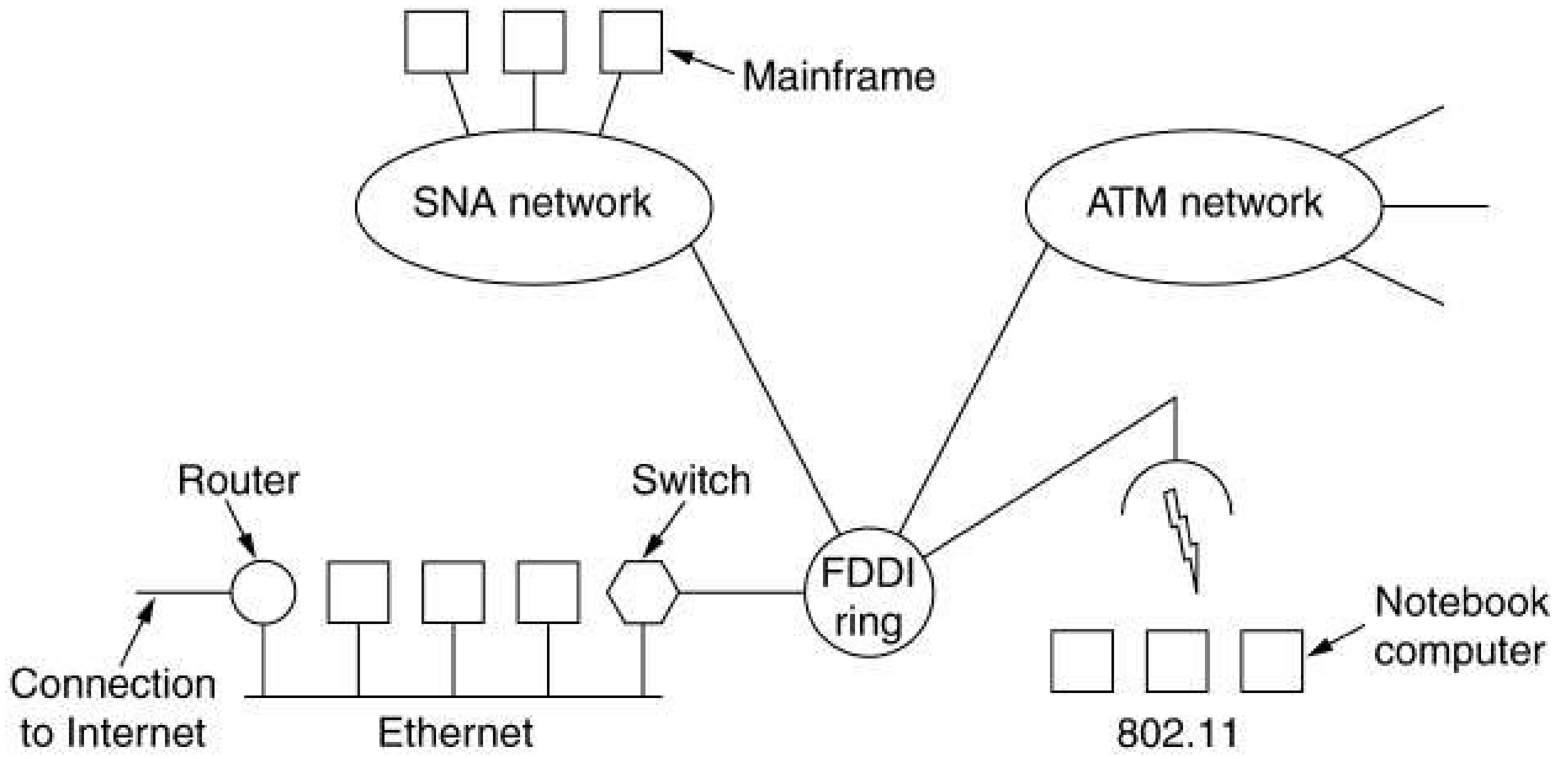
5.5. Internetworking

- Networks differ in various ways, so when multiple networks are interconnected problems can occur.
- Sometimes the problems can be finessed by tunneling a packet through a hostile network, but if the source and destination networks are different, this approach fails.
- When different networks have different maximum packet sizes, fragmentation may be called for.

5.5. Internetworking

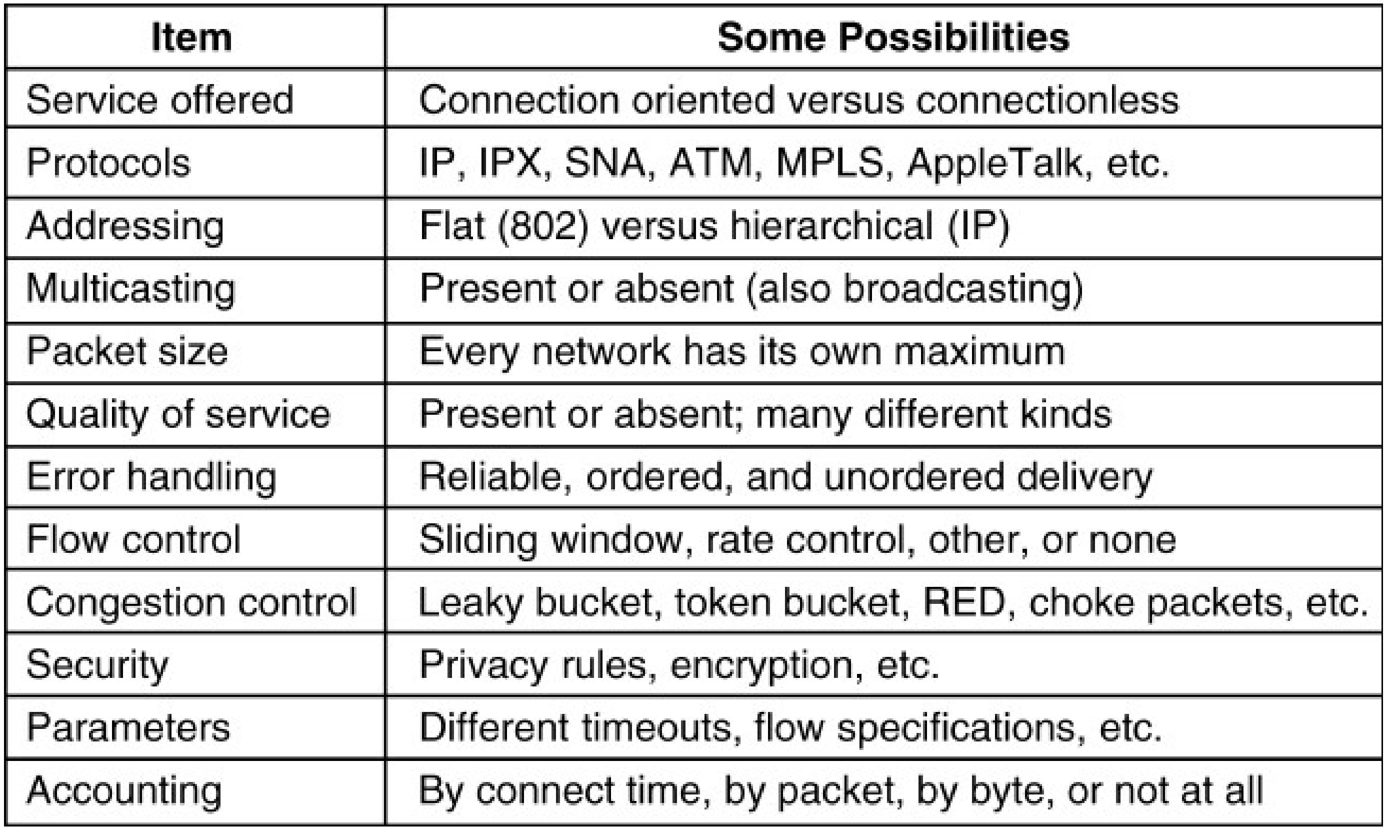
- How Networks Differ
- How N1etworks Can Be Connected
- Concatenated Virtual Circuits
- Connectionless Internetworking
- Tunneling
- Internetwork Routing
- Fragmentation

5.5. Internetworking
Connecting Networks



A collection of interconnected networks.

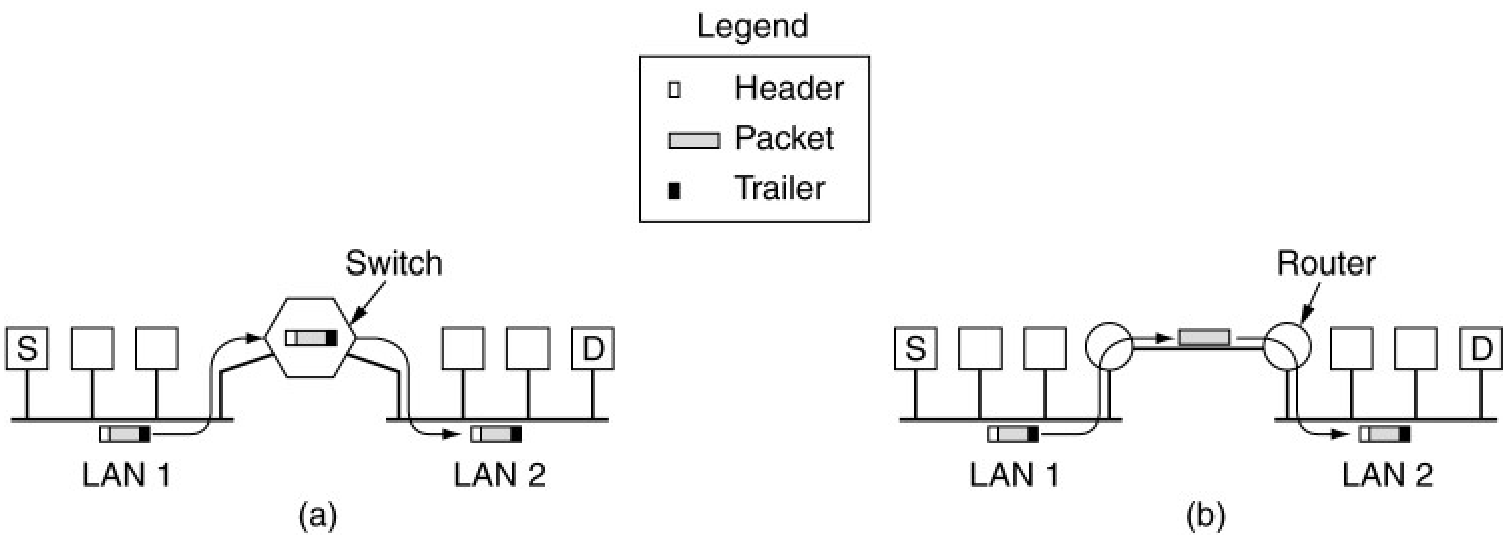
5.5. Internetworking
How Networks Differ



Item	Some Possibilities
Service offered	Connection oriented versus connectionless
Protocols	IP, IPX, SNA, ATM, MPLS, AppleTalk, etc.
Addressing	Flat (802) versus hierarchical (IP)
Multicasting	Present or absent (also broadcasting)
Packet size	Every network has its own maximum
Quality of service	Present or absent; many different kinds
Error handling	Reliable, ordered, and unordered delivery
Flow control	Sliding window, rate control, other, or none
Congestion control	Leaky bucket, token bucket, RED, choke packets, etc.
Security	Privacy rules, encryption, etc.
Parameters	Different timeouts, flow specifications, etc.
Accounting	By connect time, by packet, by byte, or not at all

Some of the many ways networks can differ.

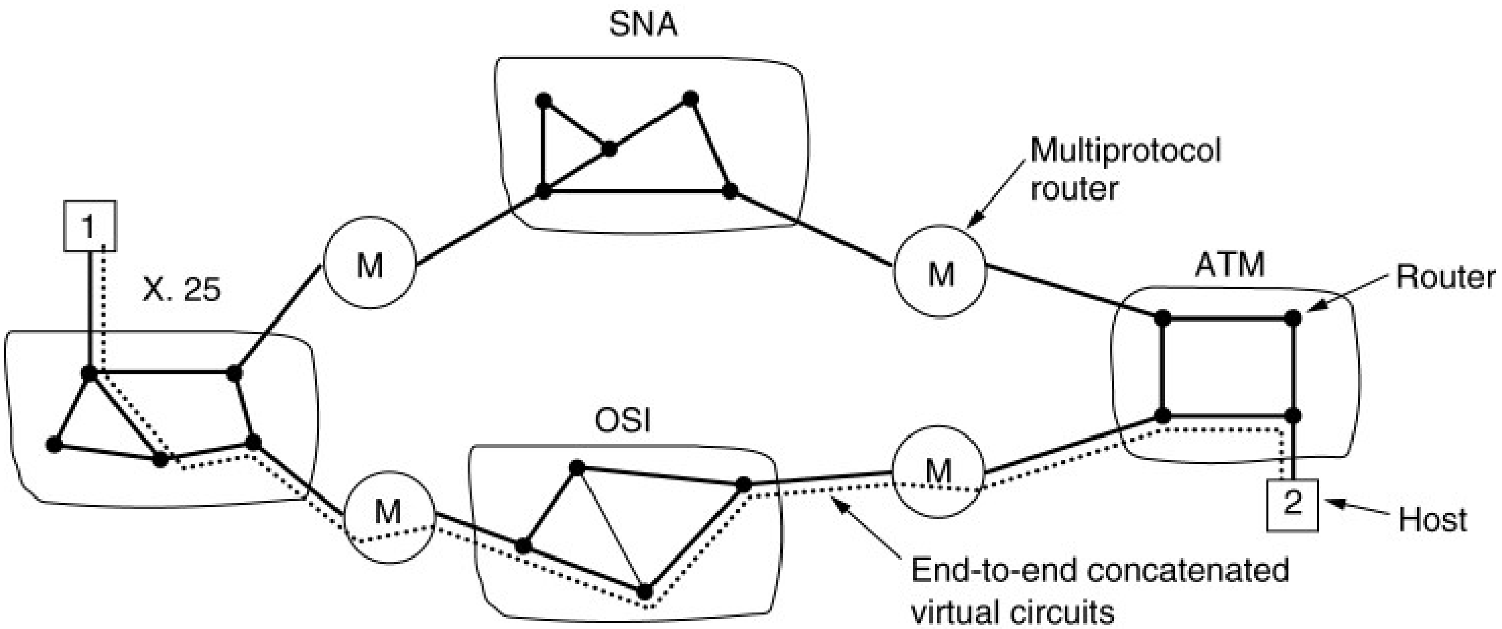
5.5. Internetworking
How Networks Can Be Connected



(a) Two Ethernets connected by a switch.

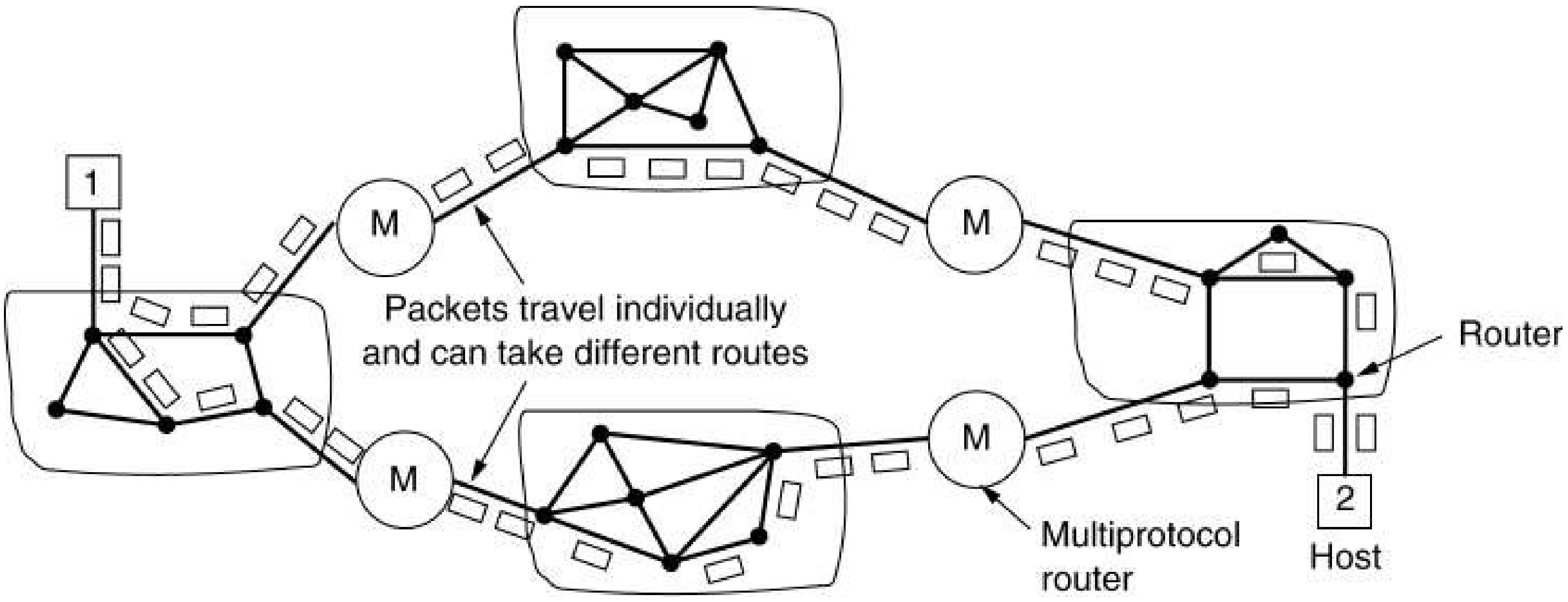
(b) Two Ethernets connected by routers.

5.5. Internetworking Concatenated Virtual Circuits



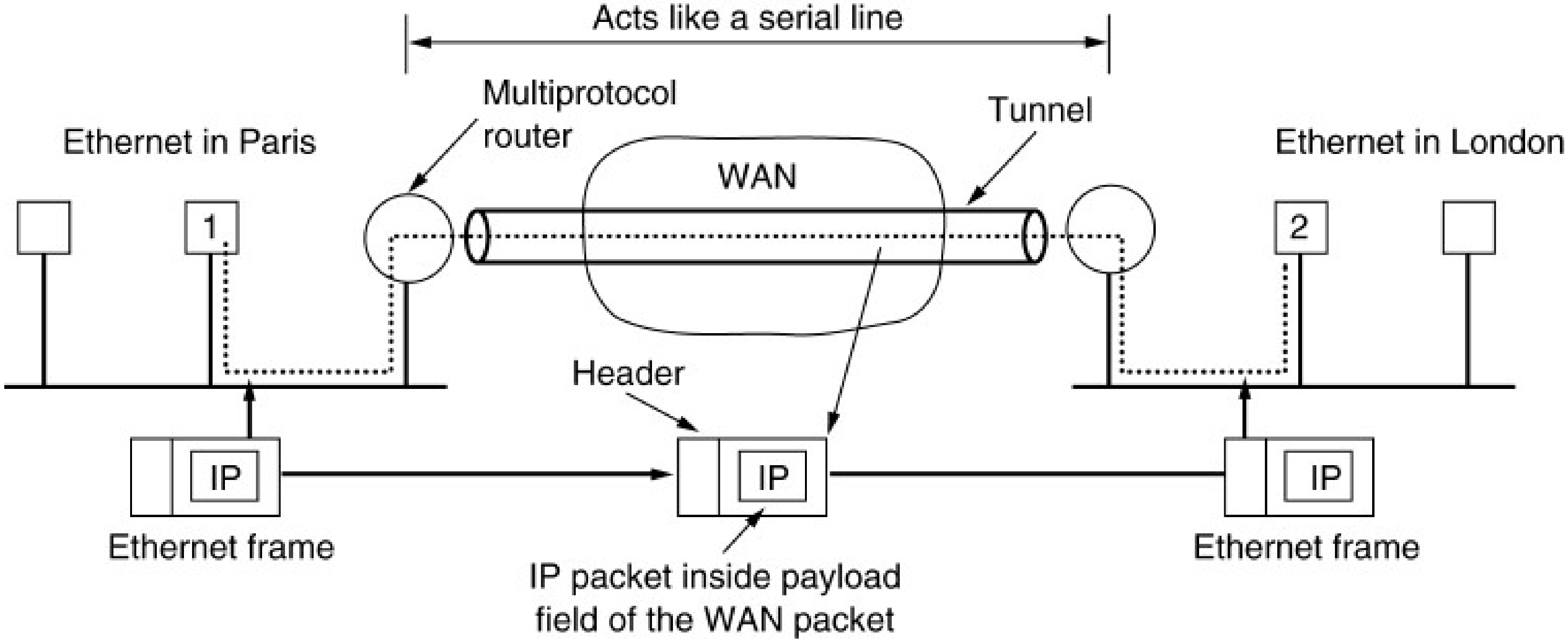
Internetworking using concatenated virtual circuits.

5.5. Internetworking
Connectionless Internetworking



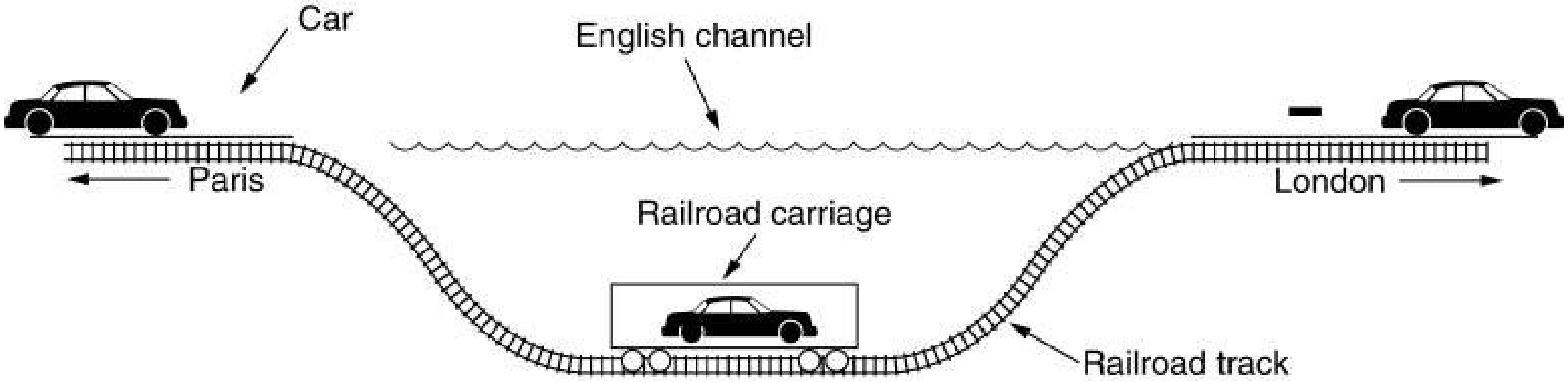
A connectionless internet.

5.5. Internetworking Tunneling



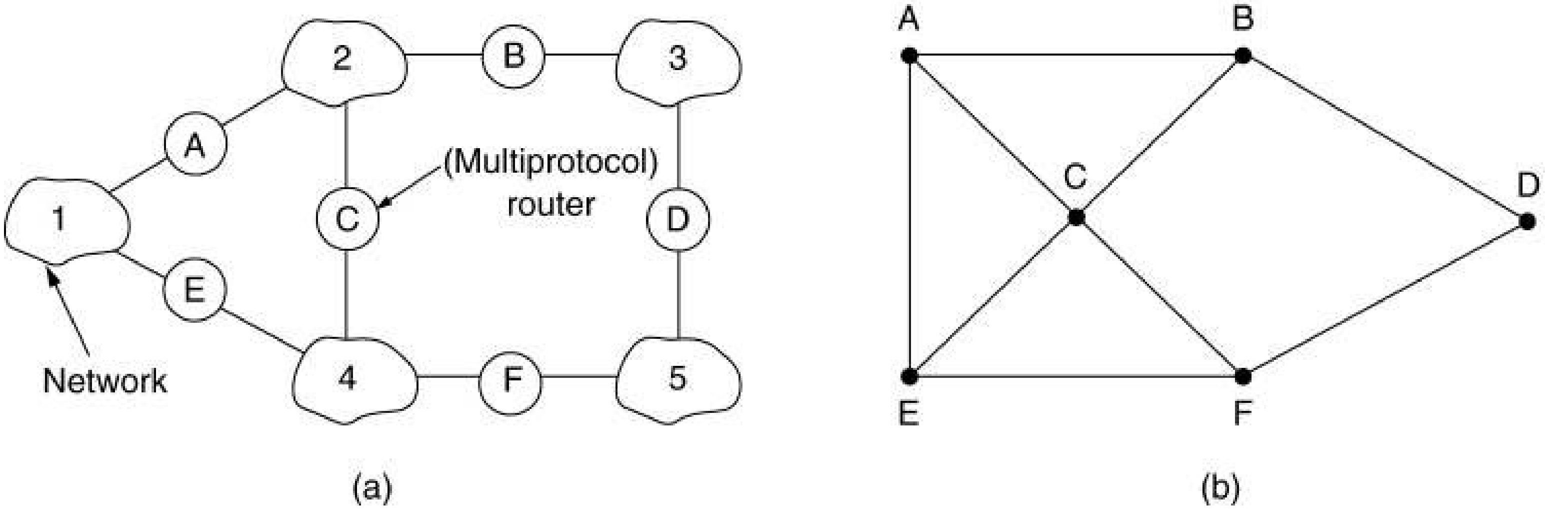
Tunneling a packet from Paris to London.

5.5. Internetworking
Tunneling (2)



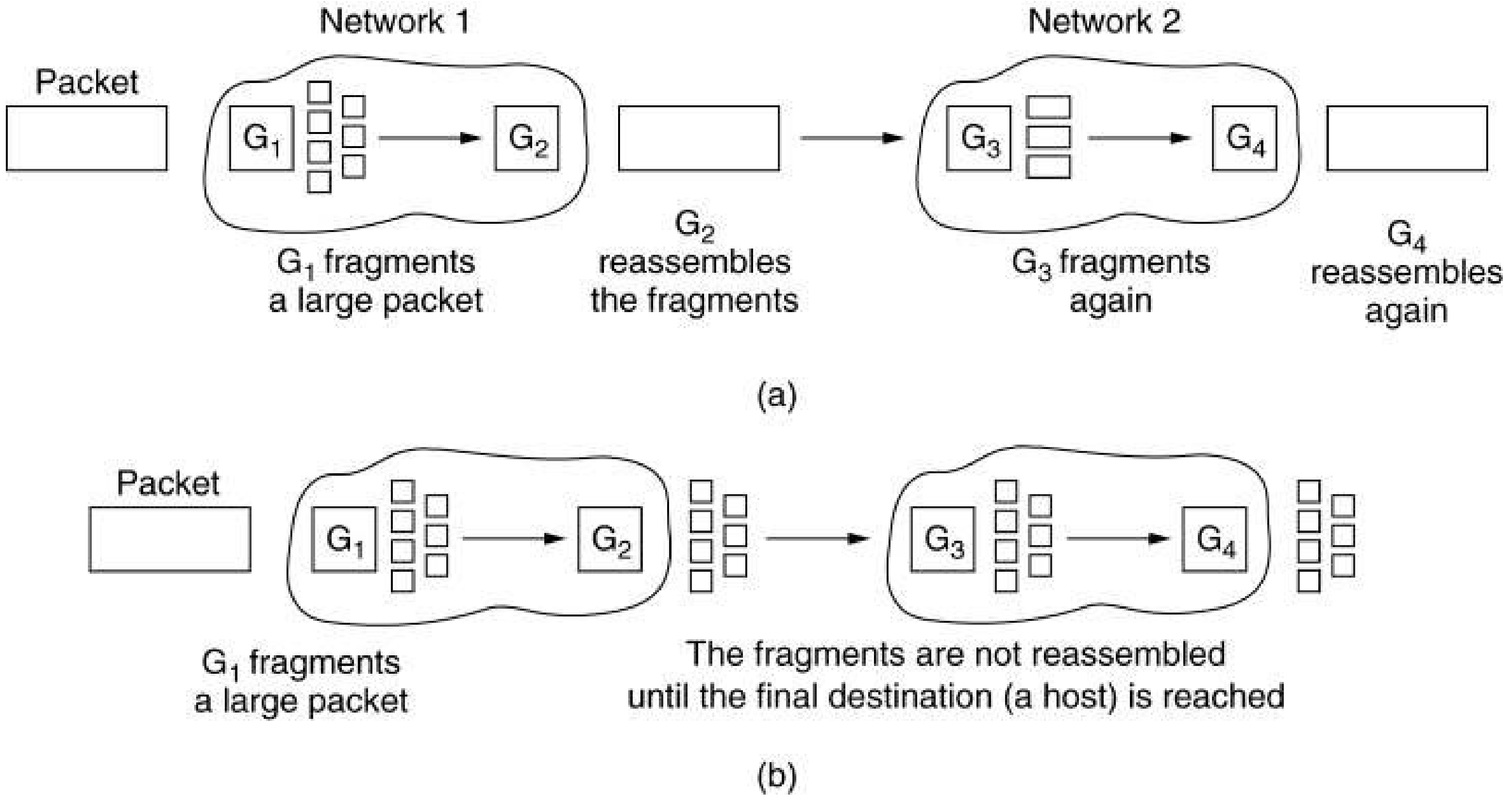
Tunneling a car from France to England.

5.5. Internetworking
Internetwork Routing



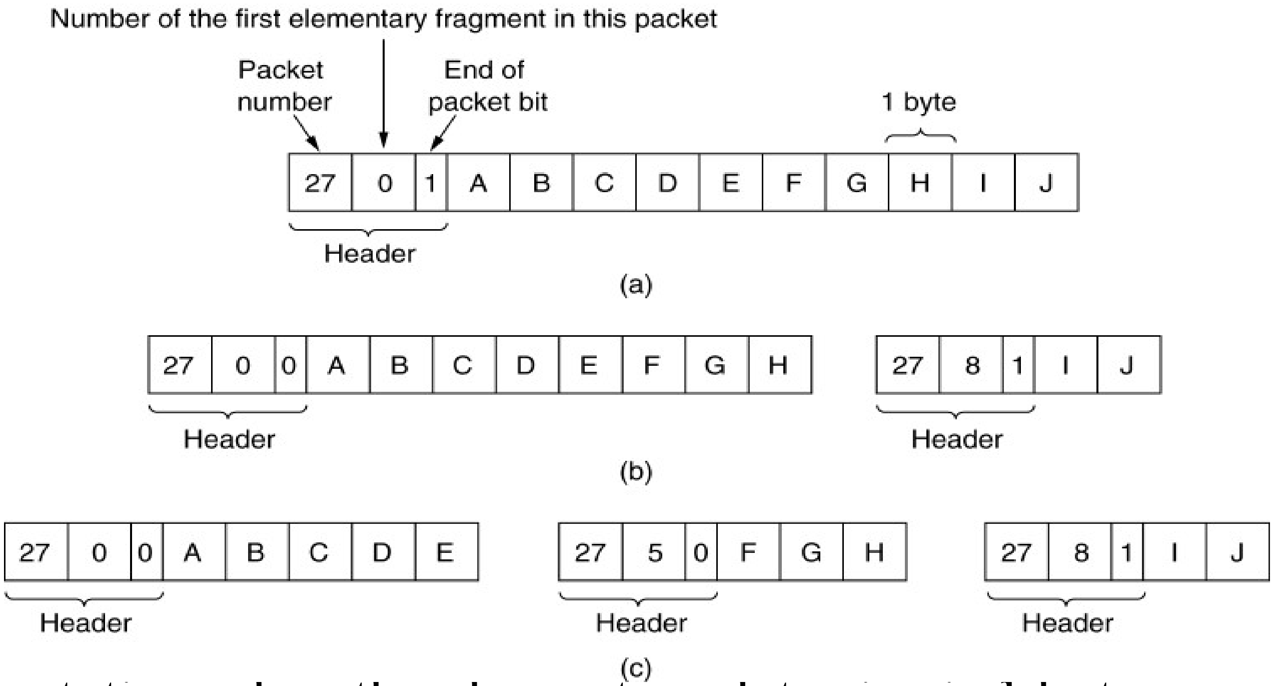
(a) An internetwork. (b) A graph of the internetwork.

5.5. Internetworking Fragmentation


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.

5.5. Internetworking Fragmentation (2)


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
(a)
27 | 0 | 0 | A | B | C | D | E | F | G | H
Header
27 | 8 | 1 | I | J
Header
(b)
27 | 0 | 0 | A | B | C | D | E
Header
27 | 5 | 0 | F | G | H
Header
27 | 8 | 1 | I | J
Header
(c)

Fragmentation when the elementary data size is 1 byte.

(a) Original packet, containing 10 data bytes.

(b) Fragments after passing through a network with maximum packet size of 8 payload bytes plus header.

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

5.6. The Network Layer in the Internet

- The internet has a rich variety of protocols related to the network layer.
- These include the data transport protocol, IP, but also the control protocols ICMP, ARP, and RARP, and the routing protocols OSPF and BGP

5.6 The Network Layer in the Internet

- The top 10 design principles (from most important to least important) for THE INTERNET:

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

5.6. The Network Layer in the Internet
Design Principles for Internet

5. Expect heterogeneity.
6. Avoid static options and parameters.
7. Look for a good design; it need not be perfect.
8. Be strict when sending and tolerant when receiving.
9. Think about scalability.
10. Consider performance and cost.

5.6. The Network Layer in the Internet
The Details of the Internet's Network Layer.

a) At the network layer, THE INTERNET can be viewed as a collection of subnetworks or AUTONOMOUS SYSTEMS (ASes) that are interconnected.

b) There is no real structure, but SEVERAL MAJOR BACKBONES exit.

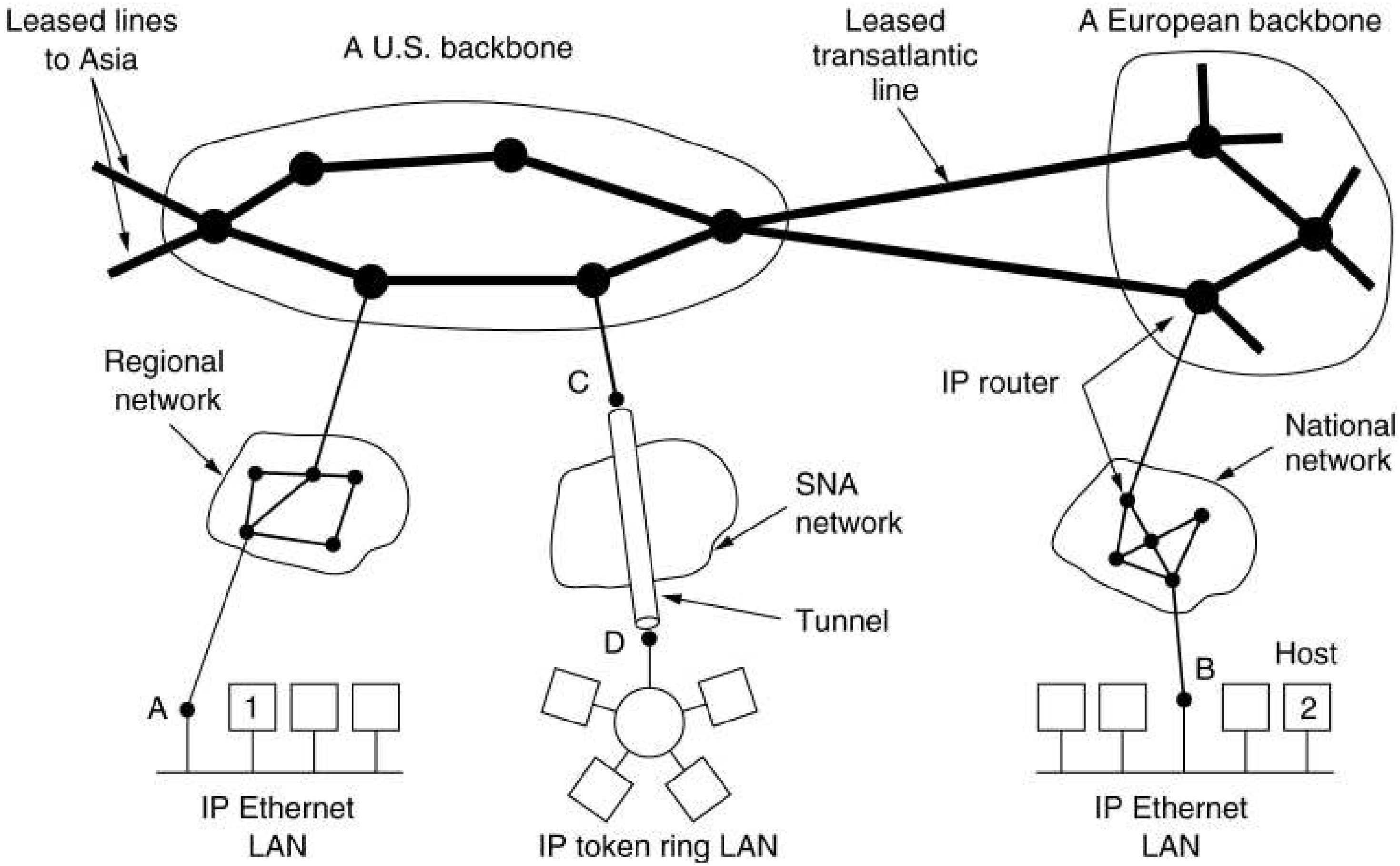
5.6. The Network Layer in the Internet

The Details of the Internet's Network Layer.

c) These are constructed from HIGH-BANDWIDTH LINES and FAST ROUTERS.

d) Attached to the backbones are regional (midlevel) networks, and attached to these regional networks are the LANs at many universities, companies, and internet service providers.

5.6. The Network Layer in the Internet
Collection of Subnetworks



The Internet is an interconnected collection of many networks.

5.6. The Network Layer in the Internet
The Details of the Internet’s Network Layer.

- The glue that holds the whole internet together is THE NETWORK LAYER PROTOCOL, IP (INTERNET PROTOCOL)
- Its job is TO PROVIDE A BEST-EFFORTS (i.e., NOT GUARANTEED) WAY TO TRANSPORT DATAGRAMS FROM SOURCE TO DESTINATION, without regard to whether these machines are on the same network or whether there are other networks in between them.

5.6. The Network Layer in the Internet
Communication in The Internet

- THE TRANSPORT LAYER takes data streams and breaks them up into DATAGRAMS.
- In theory, DATAGRAMS can be up to 64 KB each, but in practice they are usually not more than 1500 BYTES (so they fit in ONE ETHERNET FRAME).

5.6. The Network Layer in the Internet
Communication in The Internet

- Each DATAGRAM is transmitted through THE INTERNET, possibly being fragmented into smaller units as it goes.
- When all the pieces finally get to the destination machine, they are reassembled by THE NETWORK LAYER, into THE ORIGINAL DATAGRAM.

5.6. The Network Layer in the Internet
Communication in The Internet

- This datagram is then handed to The Transport Layer, which inserts it into the receiving process' input stream.
- As can be seen from figure, a packet originating at host 1 has to traverse six networks to get to host 2. In practice, it is often much more than six.

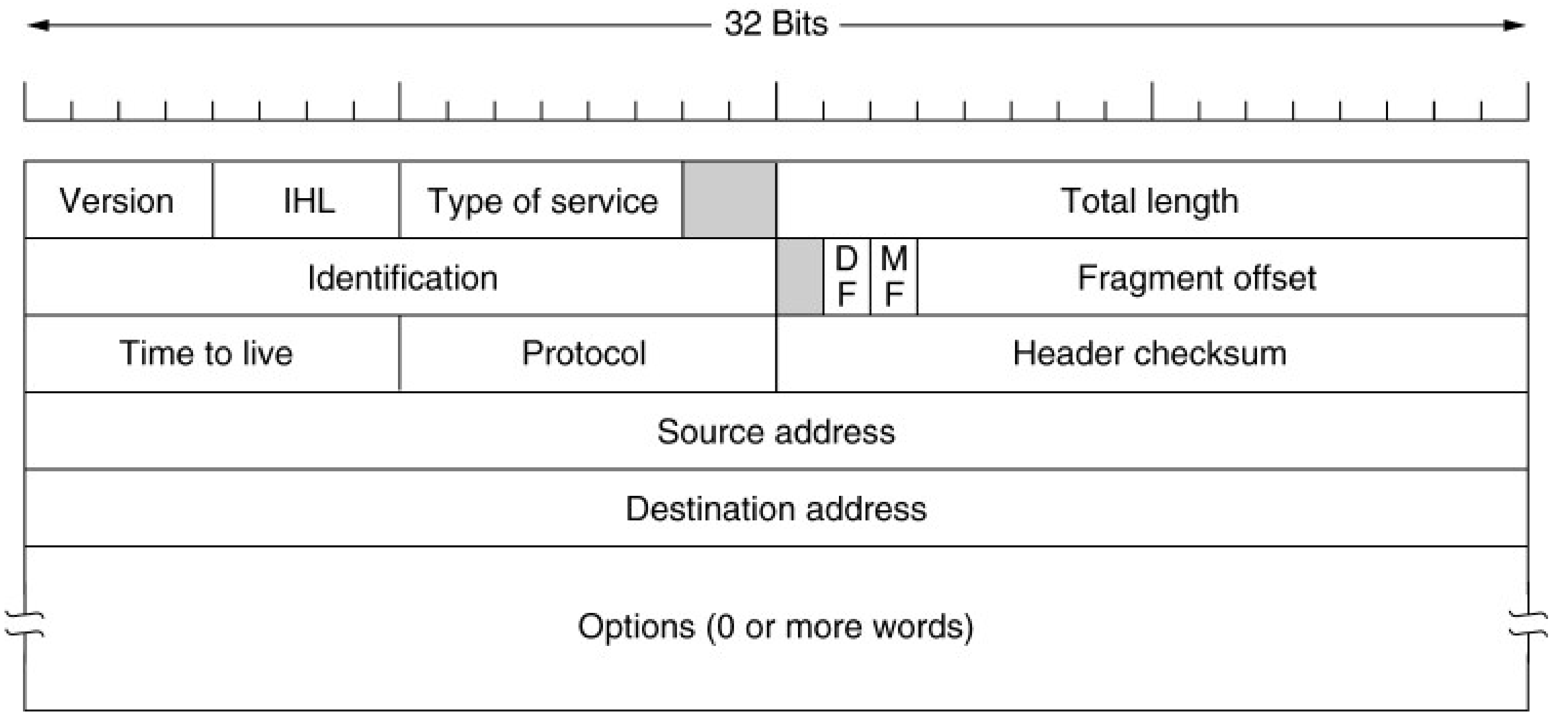
5.6. The Network Layer in the Internet

- The IP Protocol
- IP Addresses
- Internet Control Protocols
- OSPF –The Interior Gateway Routing Protocol
- BGP – The Exterior Gateway Routing Protocol
- Internet Multicasting
- Mobile IP
- IPv6

5.6. The Network Layer in the Internet
5.61. The IP Protocol

- The format of the IP DATAGRAMS
- An IP DATAGRAM consists of a header part and a text part.
- The header has a 20-byte fixed part and a variable length optional part.

5.6. The Network Layer in the Internet
The IP Protocol



32 Bits

Version IHL Type of service Total length

Identification DF MF Fragment offset

Time to live Protocol Header checksum

Source address

Destination address

Options (0 or more words)

The IPv4 (Internet Protocol) header.

The IP Protocol

- VERSION – the version field keeps track of which version of the protocol the datagram belongs to (currently a transmission between IPv4 and IPv6 is going on).
- IHL – since the header length is not constant, this field in header is provided to tell how long the header is, in 32-bıt words.

The IP Protocol

- The TYPE OF SERVICE field was and is still intended to distinguish between different classes of service (reliability, speed)
- The TOTAL LENGTH includes everything in the datagram – both header and data (the maximums length is 65,535 bytes)

The IP Protocol

- IDENTIFICATION field is needed to allow the destination host to determine which datagram a newly arrived fragment belongs to.
- DF (DON’T FRAGMENT) – it is an order to the routers not to fragment datagram because destination is incapable of putting the pieces back together again.

The IP Protocol

- MF (MORE FRAGMENTS) – all fragments except the last one have this bit set. It is needed to know when all fragments of a datagram have arrived.
- FRAGMENT OFFSET tells where in the current datagram this fragment belongs (8192 fragments per diagram).

The IP Protocol

- TIME TO LIVE field is a counter used to limit packet lifetimes (maximum lifetime of 255 sec).
- PROTOCOL. When network layer has assembled a complete datagram, it needs to know what to do with it. The PROTOCOL field tells it which transport process to give it to (TCP, UDP, others).

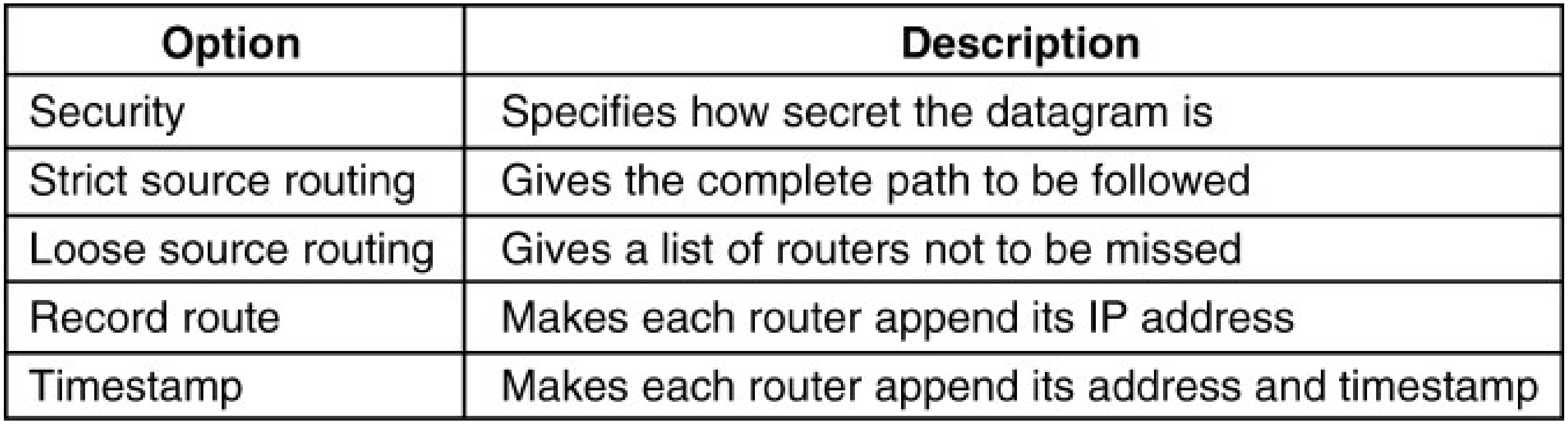
The IP Protocol

- HEADER CHECKSUM verifies the header only. Such a checksum is useful for detecting errors generated by bad memory words inside a router.
- SOURCE ADDRESS and DESTINATION ADDRESS indicate the network number and host number.

The IP Protocol

- OPTIONS field was designed to provide an escape to allow subsequent versions of the protocol to include information not present in the original design, to permit experimenters to try out new ideas, and to avoid allocating header bits to information that is rarely needed

The IP Protocol



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.

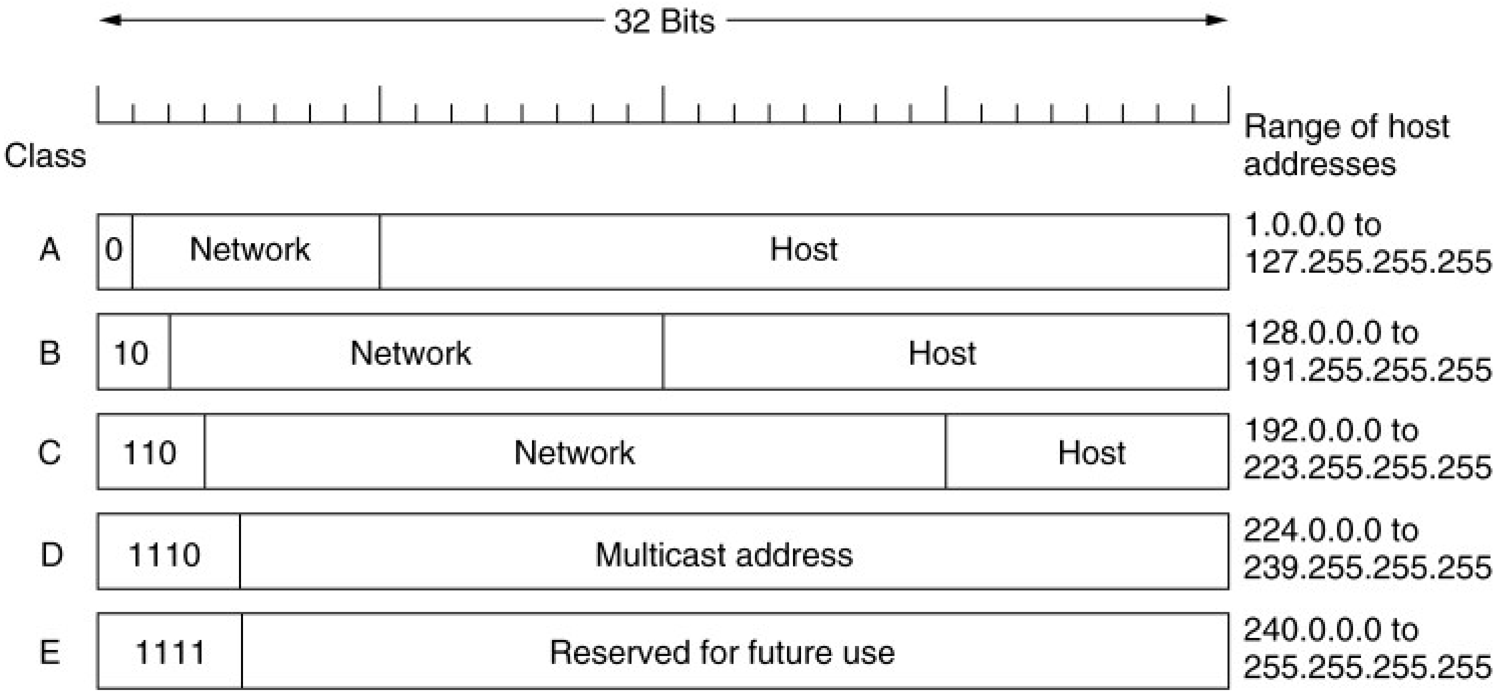
5.6. The Network Layer in the Internet
5.6.2. IP Addresses

- Every host and router on the internet has an IP ADDRESS, which encodes its NETWORK NUMBER and HOST NUMBER.
- The combination is unique: in principle, no two machines on the internet have the same IP address.

IP Addresses

- All IP addresses are 32 bits long and are used in the SOURCE ADDRESS and DESTINATION ADDRESS fields of IP packets.
- For several decades, IP addresses were divided into the FIVE CATEGORIES.
- This allocation has come to be called CLASSFUL ADDRESING.

IP Addresses



32 Bits

Class			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

- CLASS A format allows for up to 128 networks with 16 million hosts each, B - 16,384 networks with up to 64K hosts, C - 2 million networks (e.g., LANs) with up to 256 hosts each (although a few of these are special), and D is multicast, in which a datagram is directed to multiple hosts.

- Addresses beginning with 1111 are reserved for future use.
- Over 500,000 networks are now connected to the Internet, and the number grows every year.
-
-
-

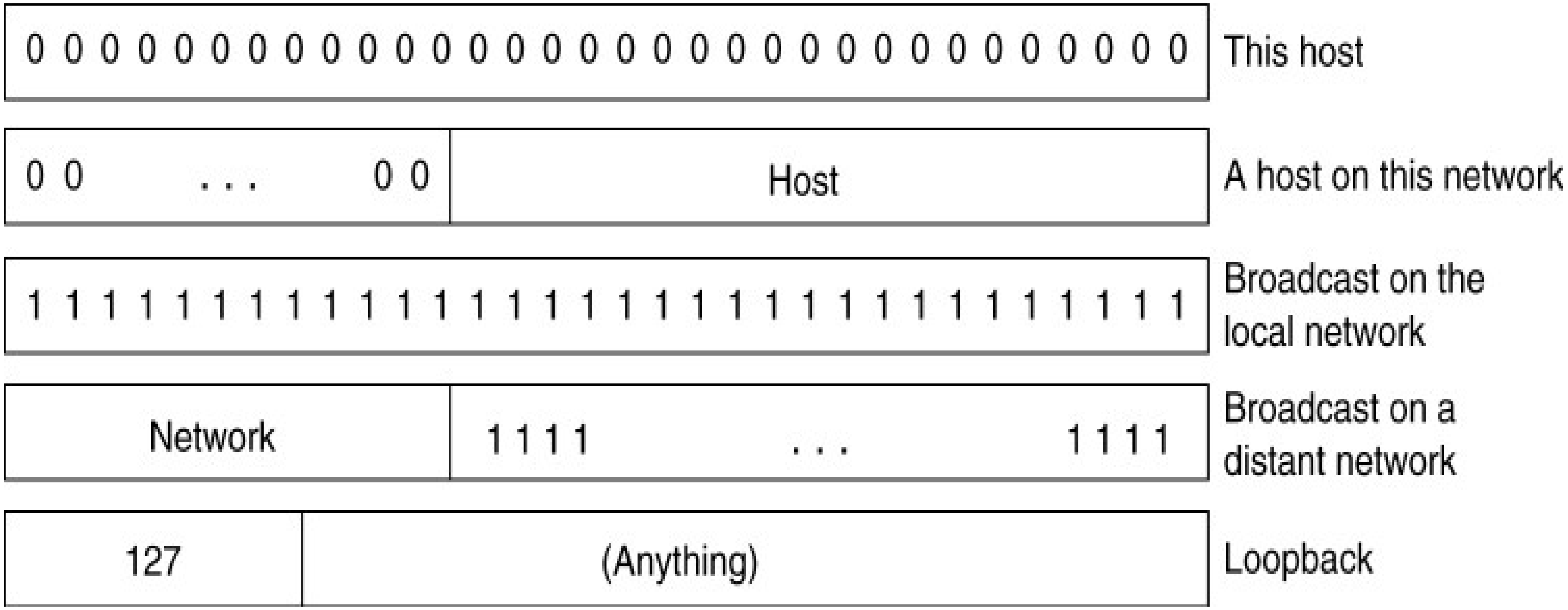
- Network numbers are managed by a nonprofit corporation called ICANN (Internet Corporation for Assigned Names and Numbers) to avoid conflicts.

IP Addresses

Network addresses, which are 32-bit numbers, are usually written in DOTTED DECIMAL NOTATION .

- In this format, each of the 4 bytes is written in decimal, from 0 to 255.
- The lowest IP address is 0.0.0.0 and the highest is 255.255.255.255

IP Addresses



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

- The value 0 and -1 (all 1s) have special meanings.
- The value 0 means this network or this host.
- The value of -1 is used as a broadcast address to mean all hosts on the indicated network.

5.6. The Network Layer in the Internet Subnets

- All the hosts in a network must have same network number.
- This property of IP addressing can cause problem as networks grow.
- For example, consider a university that started out with one class B network used by the Computer Science Dept. for computers on its ETHERNET

Subnets

- A year later, the Electrical Engineering Dept. wanted to get on the Internet, so they bought a repeater to extend the CS ETHERNET to their building.
- As time went on, many other departments acquired computers and the limit of four repeaters per ETHERNET was quickly reached.

Subnets

- A different organization was required.
- Getting a second network address would be hard to do since network addresses are scarce and the university already had enough addresses for over 60,000 hosts.

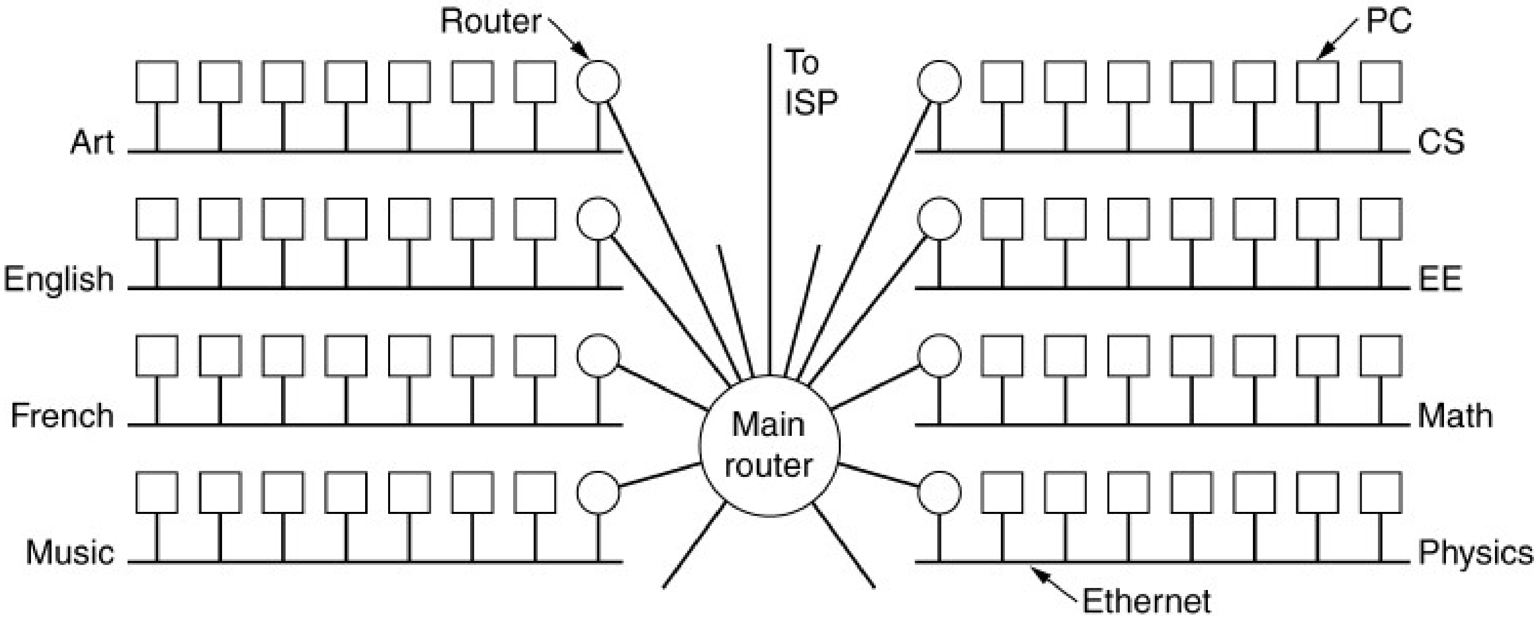
Subnets

- The problem is the rule that a single class A, B, or C address refers to one network, not to a collection of LANs.
- As more and more organizations run into this situation, a small change was made to the addressing system to deal with it.

Subnets

- The solution is to allow a network to be split into several parts for internal use but still act like a single network to the outside world.
- A typical campus network nowadays might look like that of next slide, which a main router connected to an ISP or regional network and numerous Ethernets spread around campus in different departments.

Subnets



A campus network consisting of LANs for various departments.

Subnets

- Each of the Ethernets has its own router connected to main router.
- When a packet comes into the main router, how does it know which subnet (Ethernet) to give it to?
- One way would be to have a table with 65,536 entries in the main router telling which router to use for each host on campus.

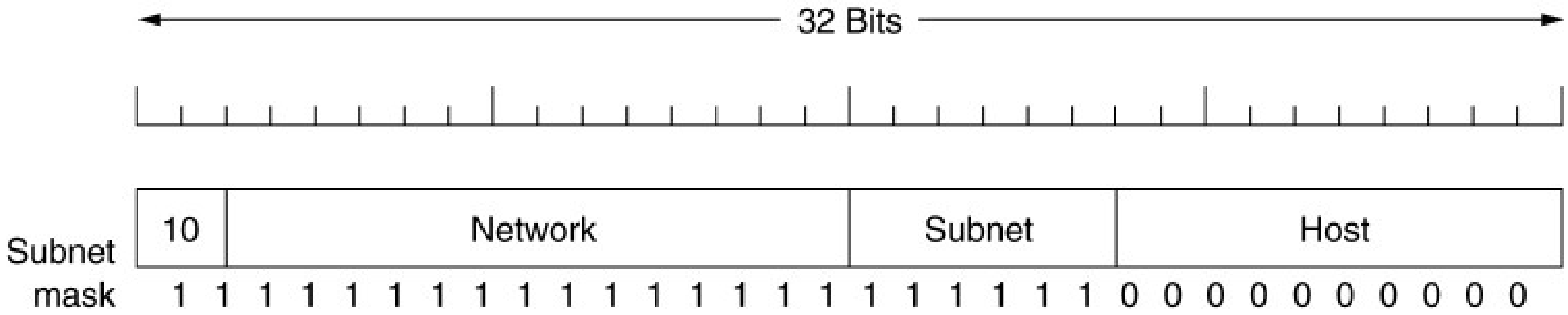
Subnets

- This idea would work, but it would require a very large table in the main router and a lot of manual maintenance as hosts were added, moved, or taken out of service.
- Instead, a different scheme was invented.

Subnets

- For example, if the university has 35 departments, it could use a 6-bit subnet number and a 10-bit host number, allowing for up to 64 Ethernets, each with a maximum of 1022 hosts
- To implement subnetting, the main router needs a SUBNET MASK that indicates the split between network + subnet number and host.

Subnets



A class B network subnetted into 64 subnets.

Subnets

- SUBNET MASKS are also written in dotted decimal notation, with the addition of a slash followed by the number of bits in THE NETWORK + SUBNET part.
- For previous example, the subnet mask can be written as 255.255.252.0
- An alternative notation is /22 to indicate that the subnet mask is 22 bits long

Subnets

- Outside the network, the subnetting is not visible, so allocating a new subnet does not require contacting ICANN or changing any external databases.
- In this example, the first subnet might use IP addresses starting at 130.50.4.1; the second subnet might start at 130.50.8.1; and so on

Subnets

- To see why the subnets are counting by fours, note that the corresponding binary addresses are as follows:
- SUBNET 1:

10000010 00110010 000001|00 00000001

- SUBNET 2:

10000010 00110010 000010|00 00000001

Subnets

- Here the vertical bar | shows the boundary between the subnet number and the host number.
- To its left is the 6-bit subnet number; to its right is the 10-bit host number

Subnets

- To see how subnets work, it is necessary to explain how IP packets are processed at a router.
- Each router has a table listing some number of (network, 0) IP addresses and some number of (this-network, host) IP addresses.

Subnets

- The first kind tells how to get to distant networks.
- The second kind tells how to get to local hosts.
- Associated with each table is the network interface to use to reach the destination, and certain other information.

Subnets

- When an IP packet arrives, its destination address is looked up in the routing table.
- If the packet is for a distant network, it is forwarded to the next router on the interface given in the table.
- If it is a local host (e.g., on the router's LAN), it is sent directly to the destination.

Subnets

- If the network is not present, the packet is forwarded to a default router with more extensive tables.
- This algorithm means that each router only has to keep track of other networks and local hosts, not (network, host) pairs, greatly reducing the size of the routing table.

Subnets

- When subnetting is introduced, the routing tables are changed, adding entries of the form (this-network, subnet, 0) and (this-network, this-subnet, host).
- Thus, a router on subnet k knows how to get to all the other subnets and also how to get to all the hosts on subnet k.

Subnets

- It does not have to know the details about hosts on other subnets.
- In fact, all that needs to be changed is to have each router do a Boolean AND with the network's subnet mask to get rid of the host number and look up the resulting address in its tables (after determining which network class it is)

Subnets

- For example, a packet addressed to 130.50.15.6 and arriving at the main router is ANDed with the subnet mask 255.255.252.0/22 to give the address 130.50.12.0
- This address is looked up in the routing tables to find out which output line to use to get to the router for subnet 3.
- Subnetting thus reduces router table space by creating a three – level hierarchy consisting of network, subnet, and host.

5.6. The Network Layer in the Internet CIDR – Classless InterDomain Routing

- The basic idea of CIDR is to allocate the remaining IP addresses in variable – sized blocks, without regard to the classes.
- Dropping the classes makes forwarding more complicated.

CIDR – Classless InterDomain Routing

- In the old classful system, forwarding worked like this. When a packet arrived at a router, a copy of the IP address was shifted right 28 bits to yield a 4-bit class number.
- Once the entry was found, the outgoing line could be looked up and the packet forwarded.

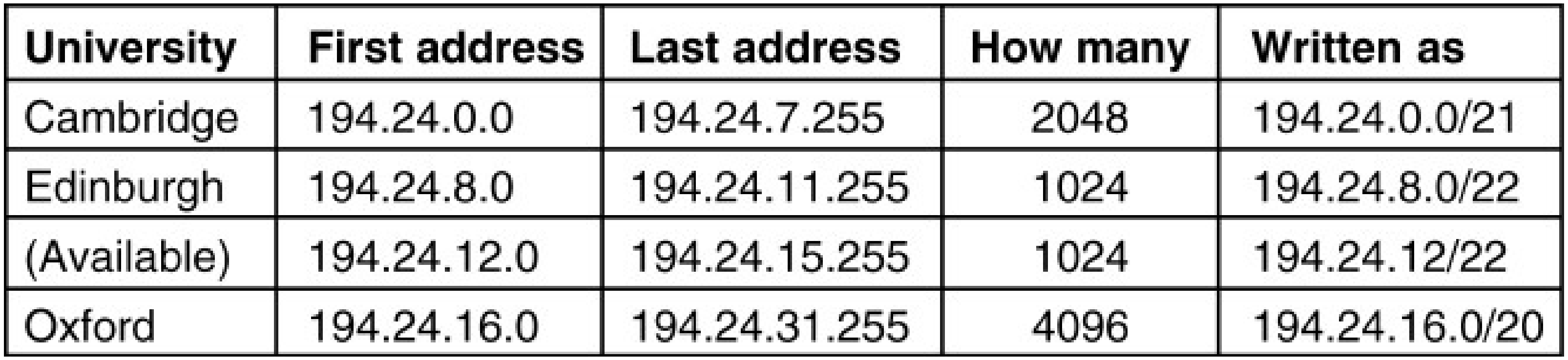
CIDR – Classless InterDomain Routing

- With CIDR, this simple algorithm no longer works.
- Instead, each routing table entry is extended by giving it a 32-bit mask.
- Thus, there is now a single routing table for all networks consisting of an array of (IP address, subnet mask, outgoing line) triples.

CIDR – Classless InterDomain Routing

- When a packet comes in, its destination IP address is first extracted.
- Then the routing table is scanned entry by entry, masking the destination address and comparing it to the table entry looking for a match.

CIDR – Classless InterDomain Routing



University	First address	Last address	How many	Written as
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. The routing tables all over the world are now updated with the three assigned entries.

ADRESS	MASK
C:11000010 00011000 00000000 00000000	1-1 1-1 11111000 0-0
E:11000010 00011000 00001000 00000000	1-1 1-1 11111100 0-0
O:11000010 00011000 00010000 00000000	1-1 1-1 11110000 0-0

5.6. The Network Layer in the Internet NAT – Network Address Translation

- The problem of running out of IP addresses is not a theoretical problem that might occur at some point in the distant future.
- It is happening right here and right now.
- The long-term solution is for the whole internet to migrate to IPv6, which has 128 – bit addresses.

NAT – Network Address Translation

- This transition is slowly occurring, but it will be years before the process is completed.
- As a consequence, some people felt that a quick fix was needed for the short term.
- This quick fix came in the form of NAT (Network Address Translation)

NAT – Network Address Translation

- The basic idea behind NAT is to assign each 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.

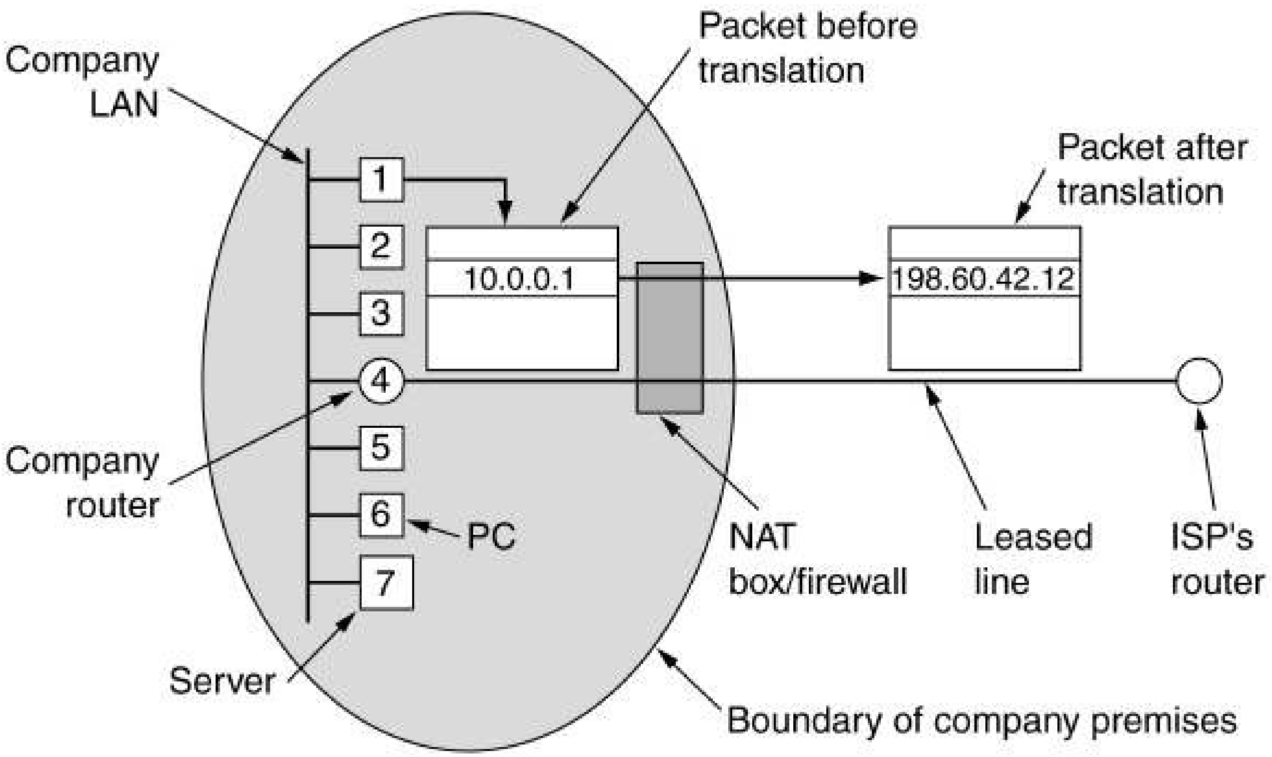
NAT – Network Address Translation

- However, when a packet exits the company and goes to the ISP, an address translation takes place.
- To make this scheme possible, three ranges of IP addresses have been declared as private.
- Companies may use them internally as they wish.

NAT – Network Address Translation

- The three reserved 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 h.)
- 192.168.0.0 – 192.168.255.255/16 (65,536 h.)

NAT – Network Address Translation



Placement and operation of a NAT box.

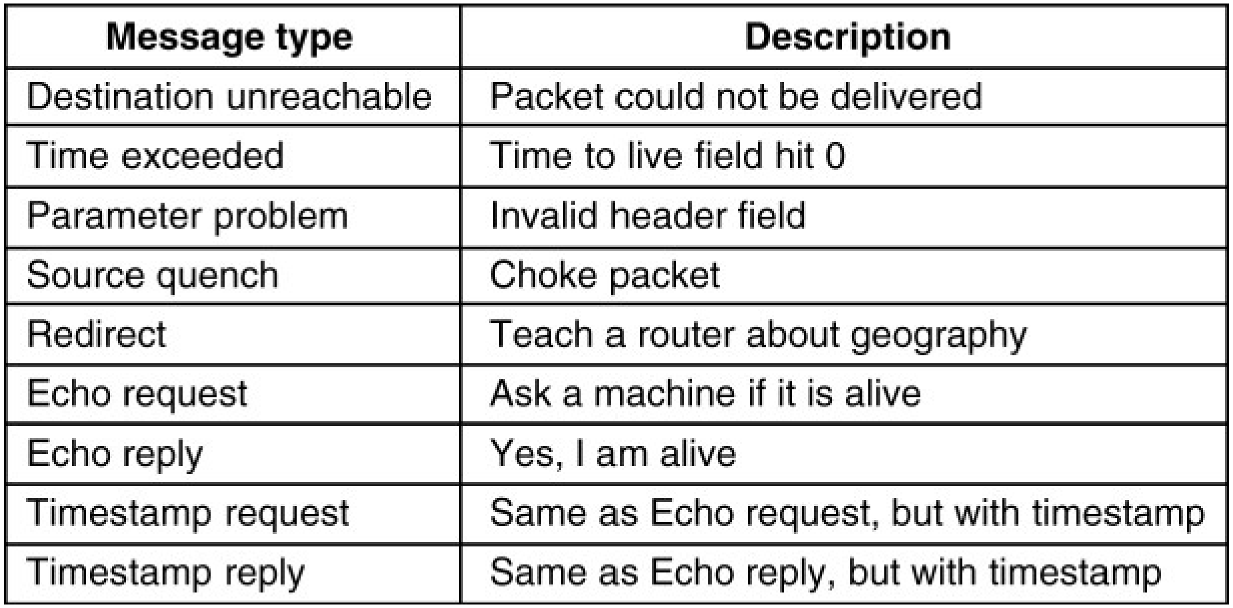
5.6. The Network Layer in the Internet
INTERNET CONTROL PROTOCOLS

- In addition to IP, which is used for data transfer, the internet has several control protocols used in the network layer, including ICMP, ARP, RARP, BOOTP, and DHCP.

ICMP - Internet Control Message Protocol

- The operation of the Internet is monitored closely by the routers.
- When something unexpected occurs, the event is reported by the ICMP, which is also used to test the internet.
- About a dozen types of ICMP message are defined.

Internet Control Message Protocol



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 request	Ask a machine if it is alive
Echo reply	Yes, I am alive
Timestamp request	Same as Echo request, but with timestamp
Timestamp reply	Same as Echo reply, but with timestamp

The principal ICMP message types.

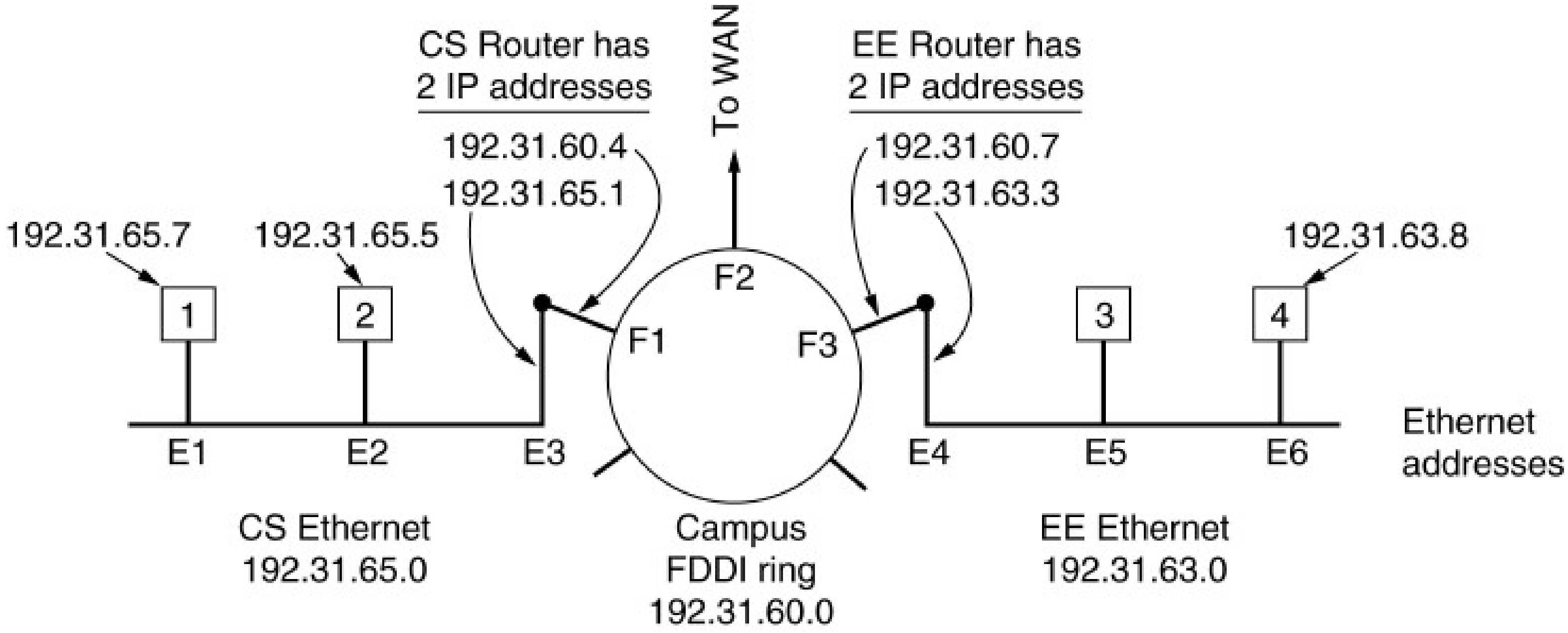
ICMP - Internet Control Message Protocol

- Each ICMP message type is encapsulated in an IP packet.
- In addition to this messages, others have been defined.

ARP– The Address Resolution Protocol

- How do IP addresses get mapped onto data link layer addresses, such as Ethernet?
- To explain how this works, let us use the example, in which a small university with several class C (now called /24) networks is illustrated.

ARP– The Address Resolution Protocol



Three interconnected /24 networks: two Ethernets and an FDDI ring.

ARP– The Address Resolution Protocol

- Let us start out by seeing how a user on host 1 sends a packet to a user on host 2.

ARP– The Address Resolution Protocol

- A better solution is for host 1 to output a broadcast packet onto the Ethernet asking: who owns IP address 192.31.65.5?
- The broadcast will arrive at every machine on Ethernet 192.31.65.0, and each one will check its IP address.
- Host 2 alone will respond with its Ethernet addresses (E2).

ARP– The Address Resolution Protocol

- In this way host 1 learns that IP address 192.31.65.5 is on the host with Ethernet address E2.
- The protocol used for asking this question and getting the reply is called ARP (Address Resolution Protocol)
- Almost every machine on the internet runs it.

ARP– The Address Resolution Protocol

- Now let us see how host 1 sends a packet to host 4 (192.31.63.8).
- Using ARP will fail because host 4 will not see the broadcast (routers do not forward Ethernet-level broadcasts).
- There are two solutions.

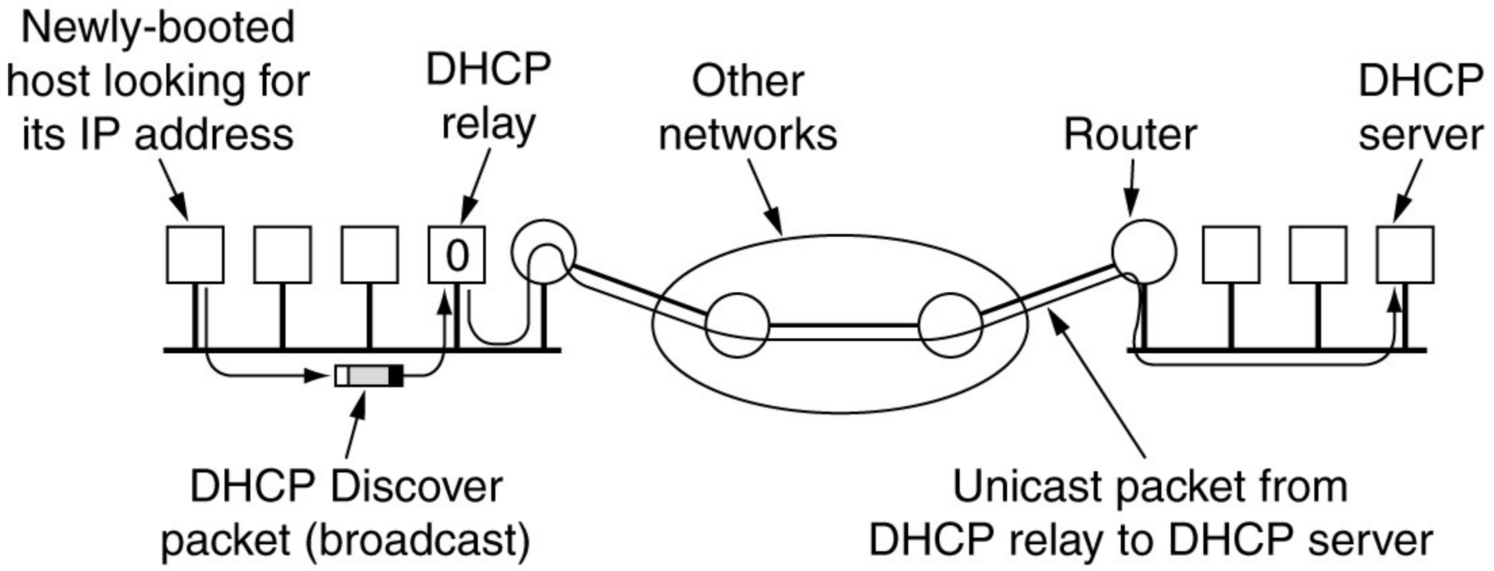
ARP– The Address Resolution Protocol

- First, the CS router could be configured to respond to ARP requests for network 192.31.63.0 (and possibly other local networks).
- In this case, host 1 will make an ARP cache entry of (192.31.63.8, E3) and happily send all traffic for host 4 to the local router. This solution is called PROXY ARP.

ARP– The Address Resolution Protocol

- The second solution is to have host 1 immediately see that the destination is on a remote network and just send all such traffic to a default Ethernet address that handles all remote traffic, in this case E3.
- This solution does not require having the CS router know which remote networks it is serving.

Dynamic Host Configuration Protocol


Newly-booted host looking for its IP address
DHCP relay
0
Other networks
Router
DHCP server
DHCP Discover packet (broadcast)
Unicast packet from DHCP relay to DHCP server

Operation of DHCP.

5.6. The Network Layer in the Internet
5.6.4. OSPF – The Interior Gateway Routing Protocol

- Internet is made up of a large number of Autonomous Systems (AS).
- Each AS is operated by a different organization and can use its own routing algorithm inside.

OSPF – The Interior Gateway Routing Protocol

- For example, the internal networks of companies X, Y, and Z are usually seen as three ASes if all three are on the internet.
- All three may use different routing algorithms internally.

OSPF–The Interior Gateway Routing Protocol

- Nevertheless, having standards, even for internal routing, simplifies the implementations at the boundaries between ASes and allows reuse of code.
- In this section we will study Routing within an AS.
- In the next one, we will look at Routing between ASes.

OSPF–The Interior Gateway Routing Protocol

- A routing algorithm within an AS is called an Interior Gateway Protocol.
- An algorithm for routing between ASes is called an Exterior Gateway Protocol.

OSPF–The Interior Gateway Routing Protocol

- Successor, called OSPF (Open Shortest Path First), became a standard in 1990.
- Most router vendors now support it, and it has become the Main Interior Gateway Protocol.

OSPF–The Interior Gateway Routing Protocol

- OSPF supports three kinds of connections and networks:

a) Point-to-point lines between exactly two routers.

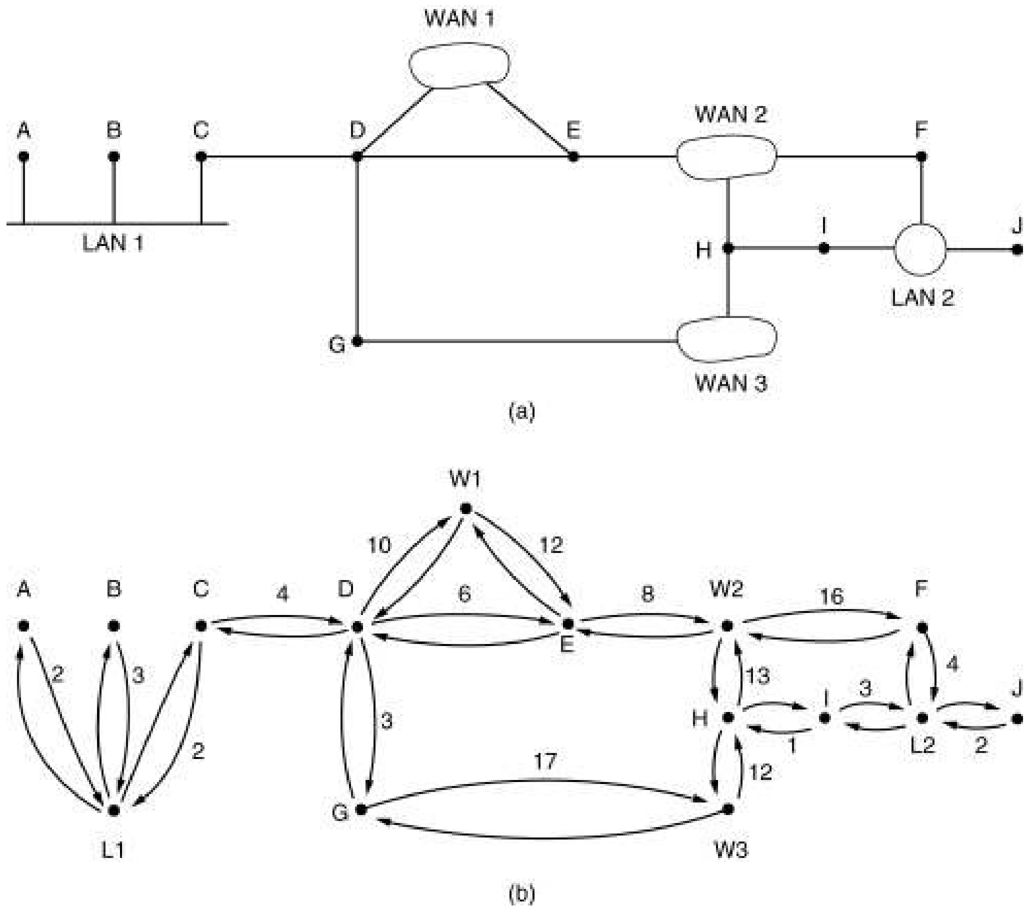
b) Multiaccess networks with broadcasting (e.g., most LANs).

c) Multiaccess networks without broadcasting (e.g., most packet-switched WANs).

OSPF–The Interior Gateway Routing Protocol

- A multiaccess network is one that can have multiples routers on it, each of which can directly communicate with all the others.
- All LANs and WANs have this property.
- Following figure shows an AS containing all three kinds of networks.

OSPF–The Interior Gateway Routing Protocol



(a) An autonomous system. (b) A graph representation of (a).

OSPF–The Interior Gateway Routing Protocol

- OSPF operates by abstracting the collection of actual networks, routers, and lines into a directed graph in which each arc is assigned a cost (distance, delayed.).
- It then computes the shortest path based on the weight on the arcs.

OSPF–The Interior Gateway Routing Protocol

- Many of ASes in the internet are themselves large and nontrivial to manage.
- OSPF allows them to be divided into numbered AREA, where an area is a network or a set of contiguous networks.

OSPF–The Interior Gateway Routing Protocol

- Every AS has a Backbone area, called Area 0.
- All areas are connected to the Backbone, possible by Tunnels, so it is possible to go from any area in AS to any other area in AS via the backbone.

OSPF–The Interior Gateway Routing Protocol

- A tunnel is represented in the graph as an arc and has a cost.
- Each router that is connected to two or more areas is part of the backbone.
- As with other areas, the topology of the backbone is not visible outside the backbone.

OSPF–The Interior Gateway Routing Protocol

- Within an area, each router has the same link state database and runs the same shortest path algorithm.
- A router that connects to two areas needs the databases for both areas and must run the shortest path algorithm for each one separately.

OSPF–The Interior Gateway Routing Protocol

- During normal operation, three kinds of routes may be needed: Intra-area, Interarea, and Inter-AS.
- Intra-area routes are the easiest, since the source router already knows the shortest path to the destination router.

OSPF–The Interior Gateway Routing Protocol

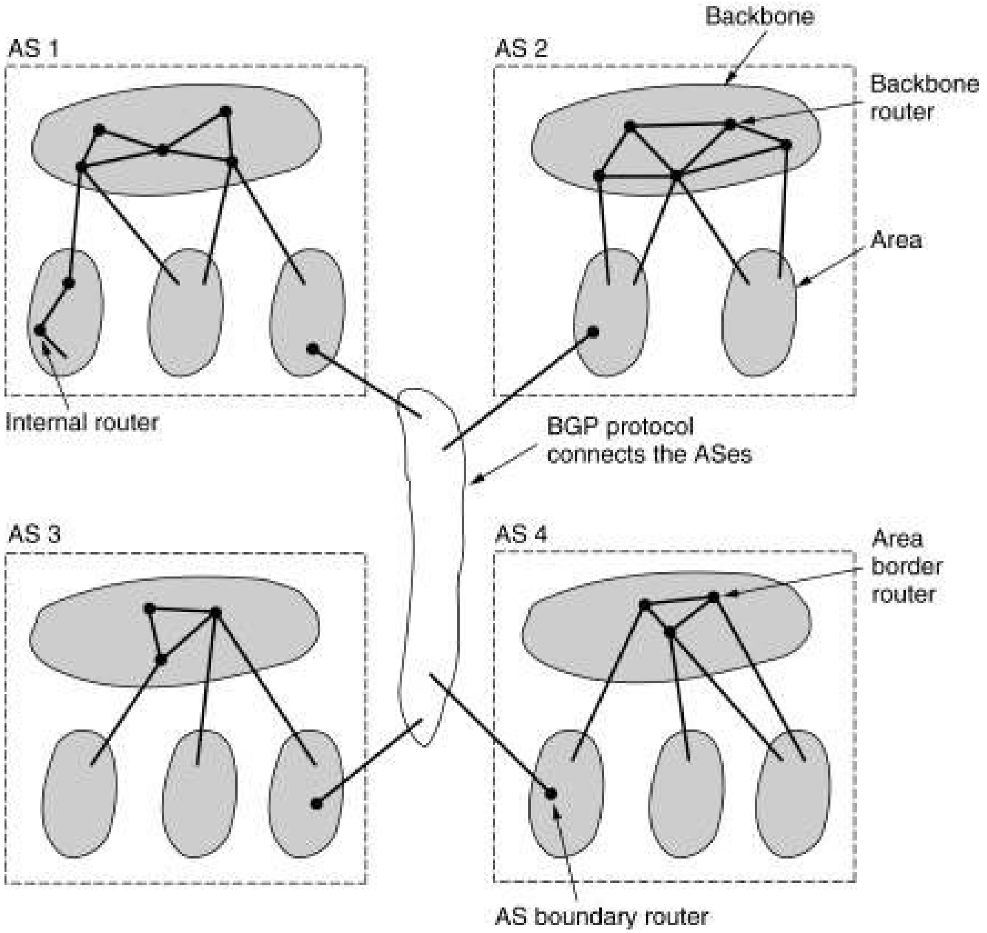
- Interarea routing always proceeds in three steps:

a) Go from the source to the backbone;

b) Go across the backbone to the destination area;

c) Go to the destination

OSPF (2)



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

OSPF–The Interior Gateway Routing Protocol

- OSPF distinguishes four classes of routers:

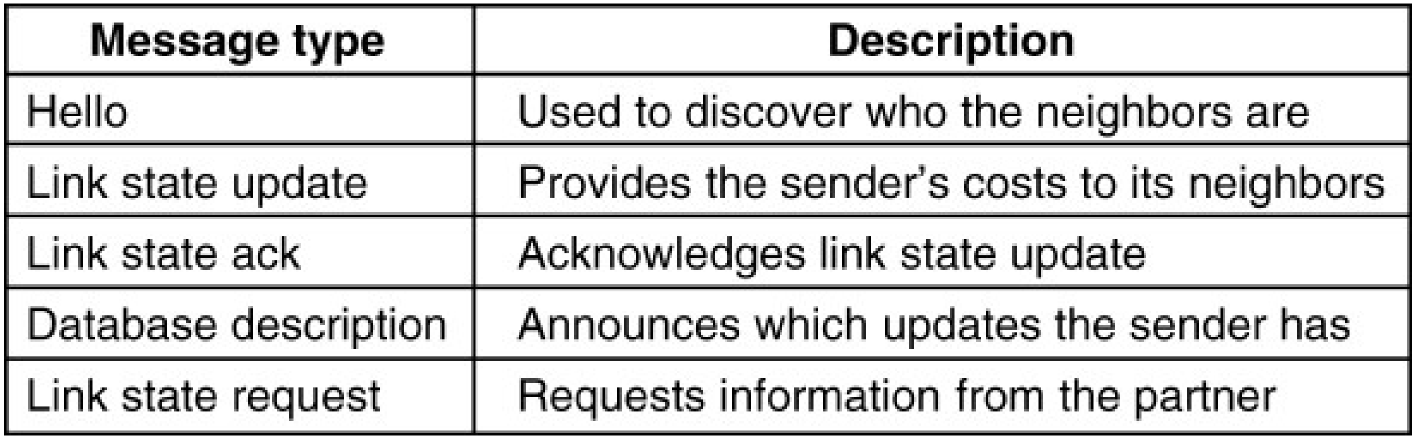
a) Internal routers are wholly within one area.

b) Area border routers connect two or more areas.

c) Backbone routers are on the backbone

d) As boundary routers talk to routers in other ASes.

OSPF



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.

OSPF–The Interior Gateway Routing Protocol

- Using flooding, each router informs all the other routers in its area of its neighbors and costs.
- This information allows each router to construct the graph for its area (s) and compute the shortest path.
- The backbone area does this too.

5.6. The Network Layer in the Internet

5.6.5. BGP – The Exterior Gateway Routing Protocol

- Between ASes, a different protocol, BGP (Border Gateway Protocol), is used.
- A different protocol is needed between ASes because the goals of an interior gateway protocol and an exterior gateway protocol are not the same.

BGP – The Exterior Gateway Routing Protocol

- All an interior gateway protocol has to do is move packets as efficiently as possible from the source to the destination.
- Exterior gateway protocols in general, and BGP in particular, have been designed to allow many kinds of routing polices to be enforced in the inter AS traffic.

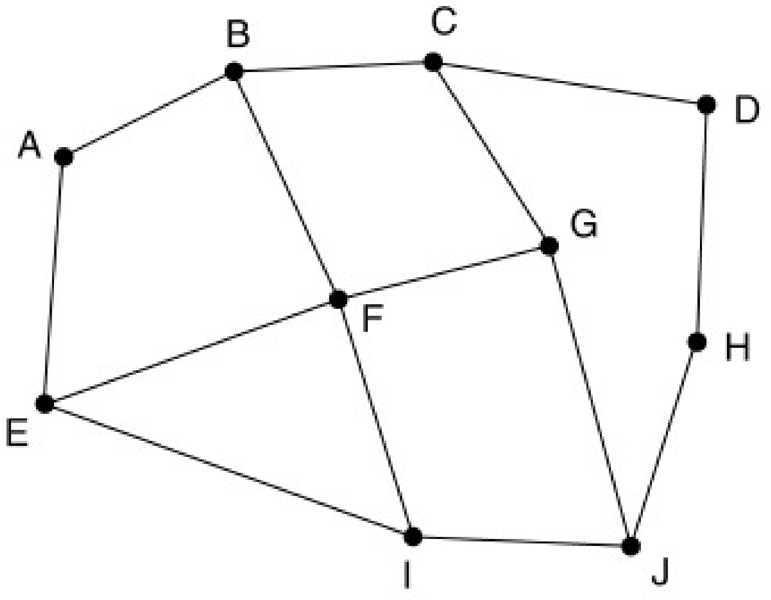
BGP – The Exterior Gateway Routing Protocol

- A few examples of routing constraints are:
- No transit traffic through certain ASes.
- Never put Iraq on a route starting at the Pentagon.
- Do not use the united states to get from British Columbia to Ontario.

BGP – The Exterior Gateway Routing Protocol

- Pairs of BGP routers communicate with each other by establishing TCP connections.
- Operating this way provides reliable communication and hides all the details of the network being passed through.
- BGP is fundamentally a distance vector protocol.

BGP – The Exterior Gateway Routing Protocol



(a)

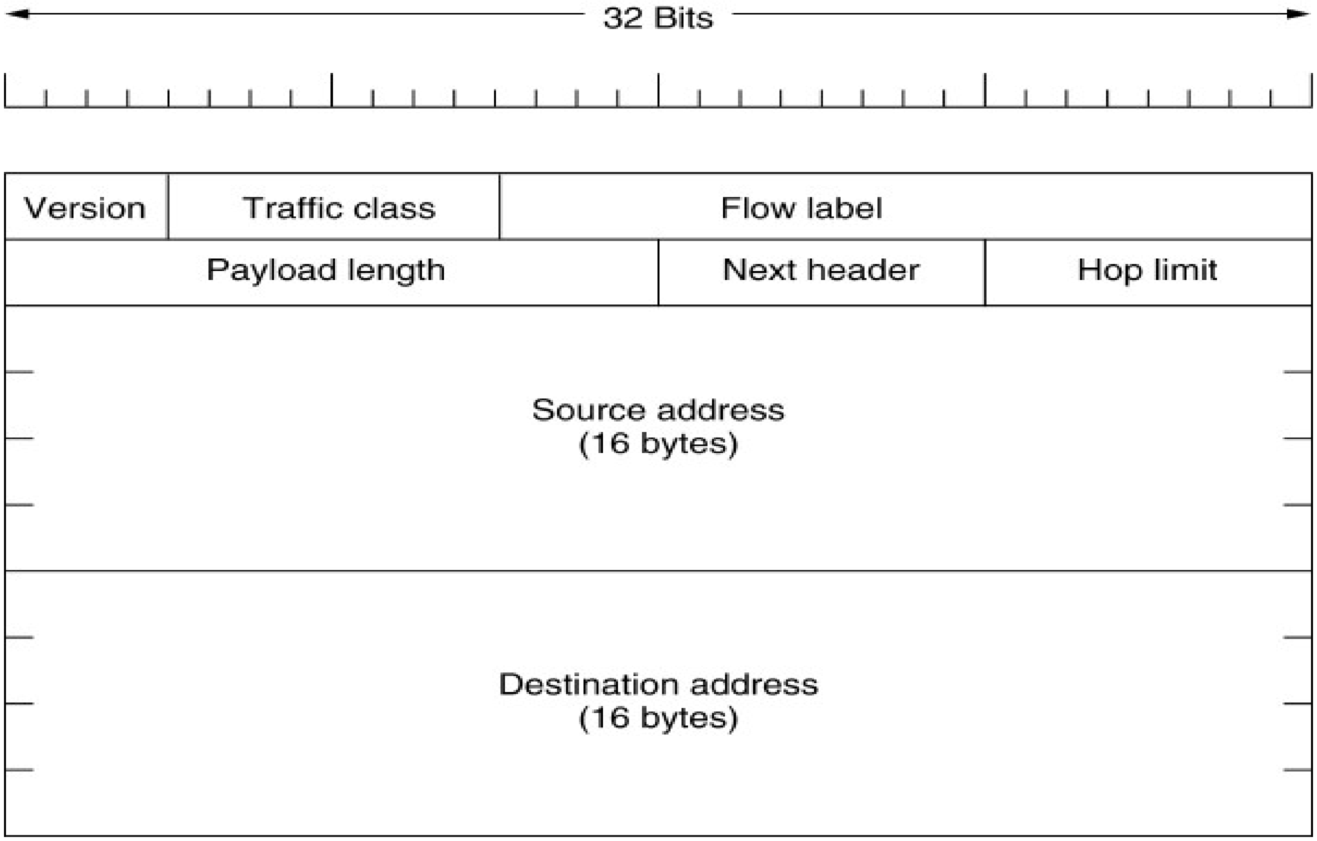
Information F receives
from its neighbors about D

From B: "I use BCD"
From G: "I use GCD"
From I: "I use IFGCD"
From E: "I use EFGCD"

(b)

(a) A set of BGP routers. (b) Information sent to F.

The Main IPv6 Header



32 Bits

Version | Traffic class | Flow label

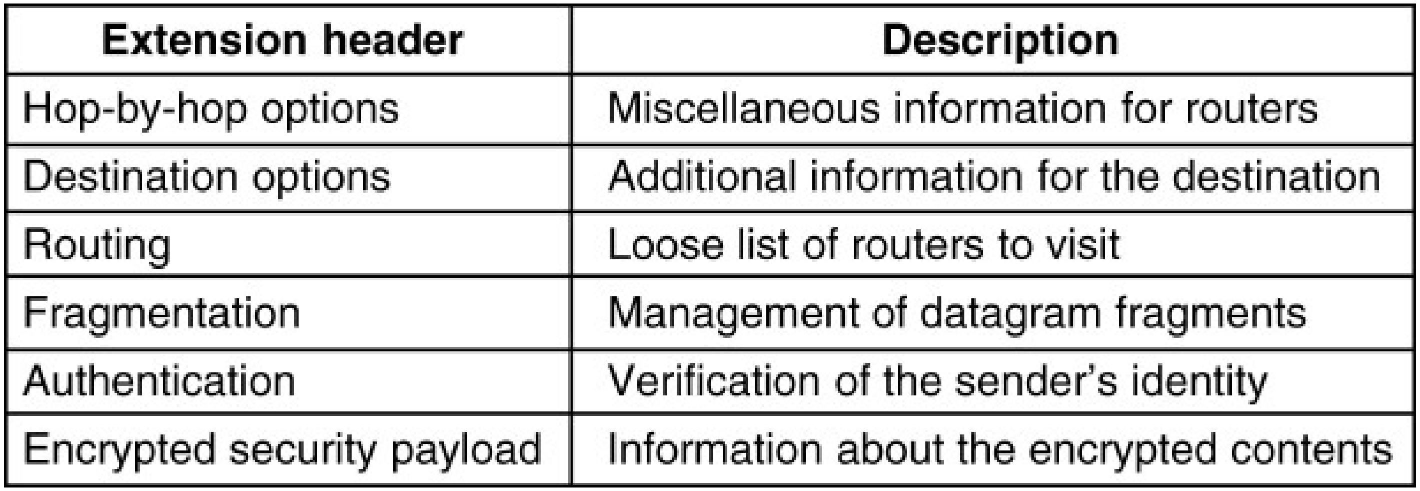
Payload length | Next header | Hop limit

Source address
(16 bytes)

Destination address
(16 bytes)

The IPv6 fixed header (required).

Extension Headers



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

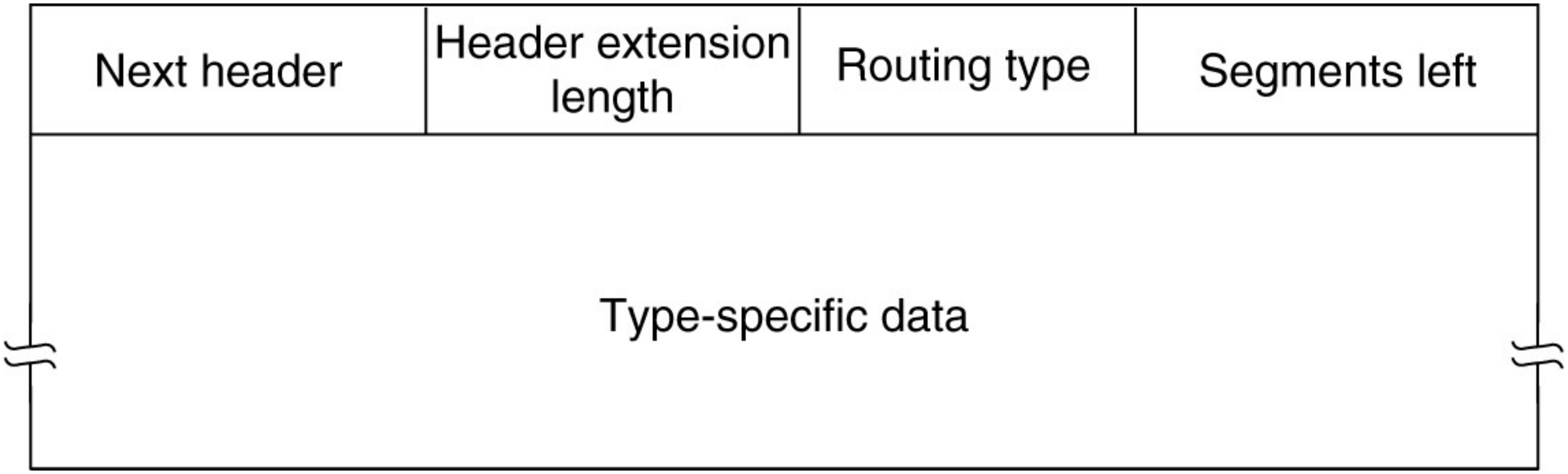
IPv6 extension headers.

Extension Headers (2)

Next header	0	194	4
Jumbo payload length			

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

Extension Headers (3)



Next header

Header extension length

Routing type

Segments left

Type-specific data

The extension header for routing.