COMPUTER NETWORKS- (20MCA23C)
UNIT-IV
'THE NETWORK LAYER'

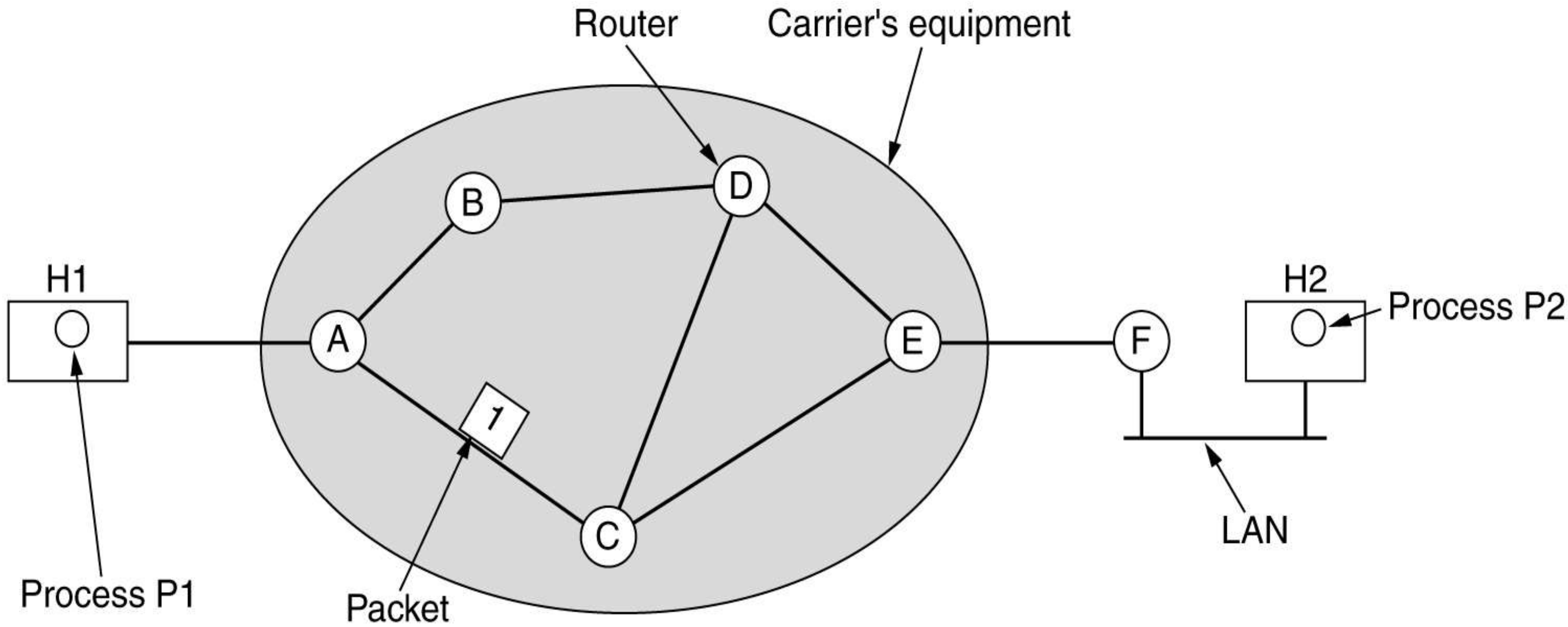
FACULTY:

DR. R.A. ROSELINE, M.SC., M.PHIL., PH.D.,
ASSOCIATE PROFESSOR AND HEAD,
POST GRADUATE AND RESEARCH DEPARTMENT OF COMPUTER APPLICATIONS,
GOVERNMENT ARTS COLLEGE (AUTONOMOUS), COIMBATORE – 641 018.

NETWORK LAYER DESIGN ISSUES

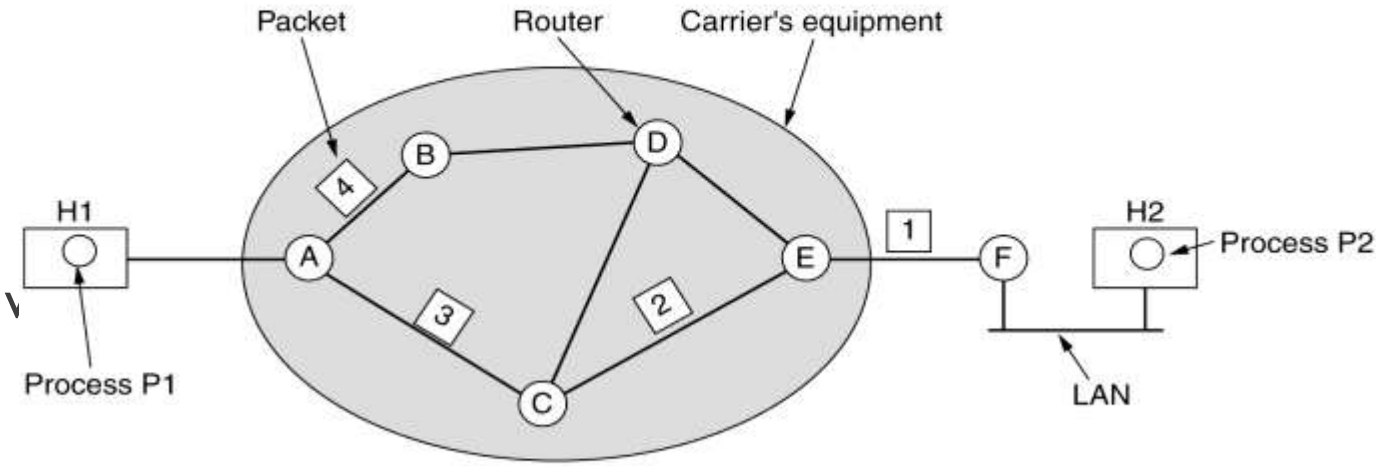
- Store-and-Forward Packet Switching
- Services Provided to the Transport Layer
- Implementation of Connectionless Service
- Implementation of Connection-Oriented Service
- Comparison of Virtual-Circuit and Datagram Subnets

STORE-AND-FORWARD PACKET SWITCHING



IMPLEMENTATION OF CONNECTIONLESS SERVICE

- Routing v



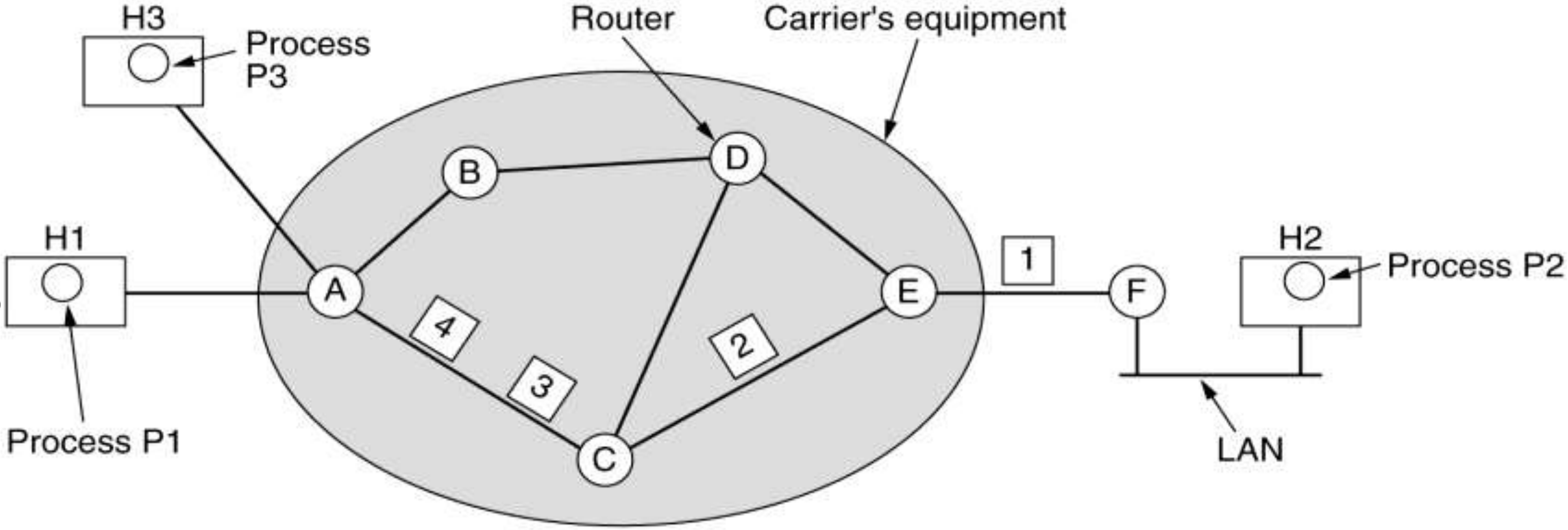
A's table

Dest.	A's table: initially (Line)	A's table: later (Line)	C's table	E's table
A	–	–	A	C
B	B	B	A	D
C	C	C	–	C
D	B	B	D	D
E	C	B	E	–
F	C	B	E	F

Dest. Line

IMPLEMENTATION OF CONNECTION-ORIENTED SERVICE

- R



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

COMPARISON OF VIRTUAL-CIRCUIT AND DATAGRAM SUBNETS

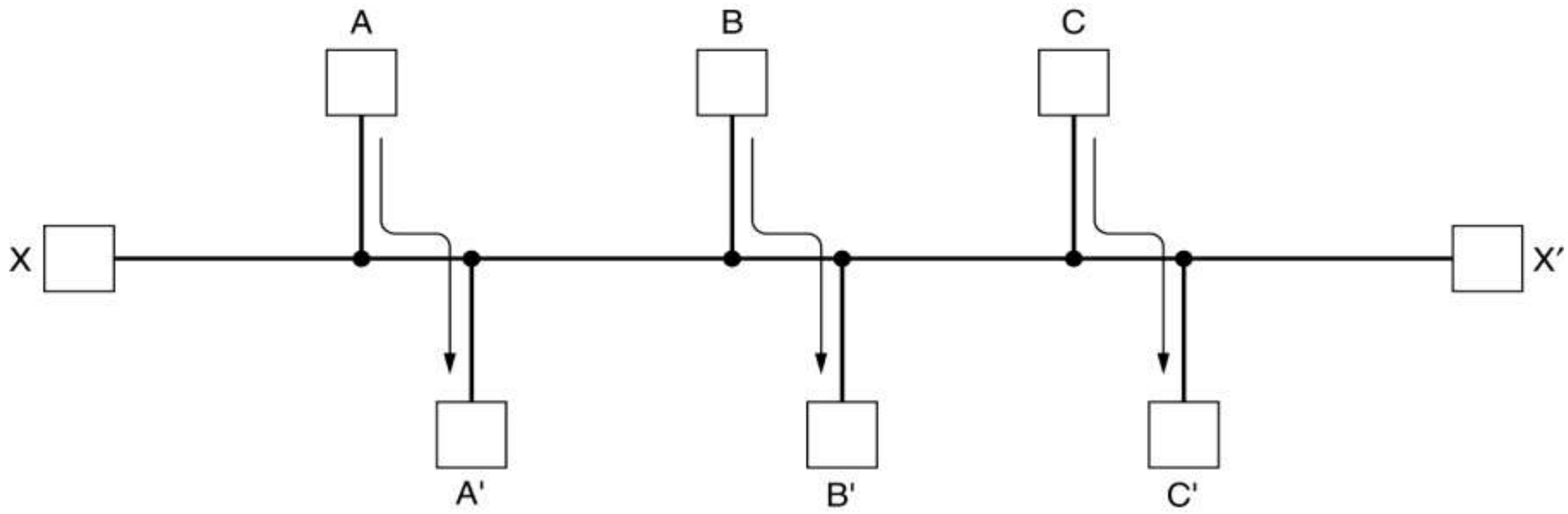
Issue	Datagram subnet	Virtual-circuit subnet
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per 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

ROUTING ALGORITHMS (1)

- 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

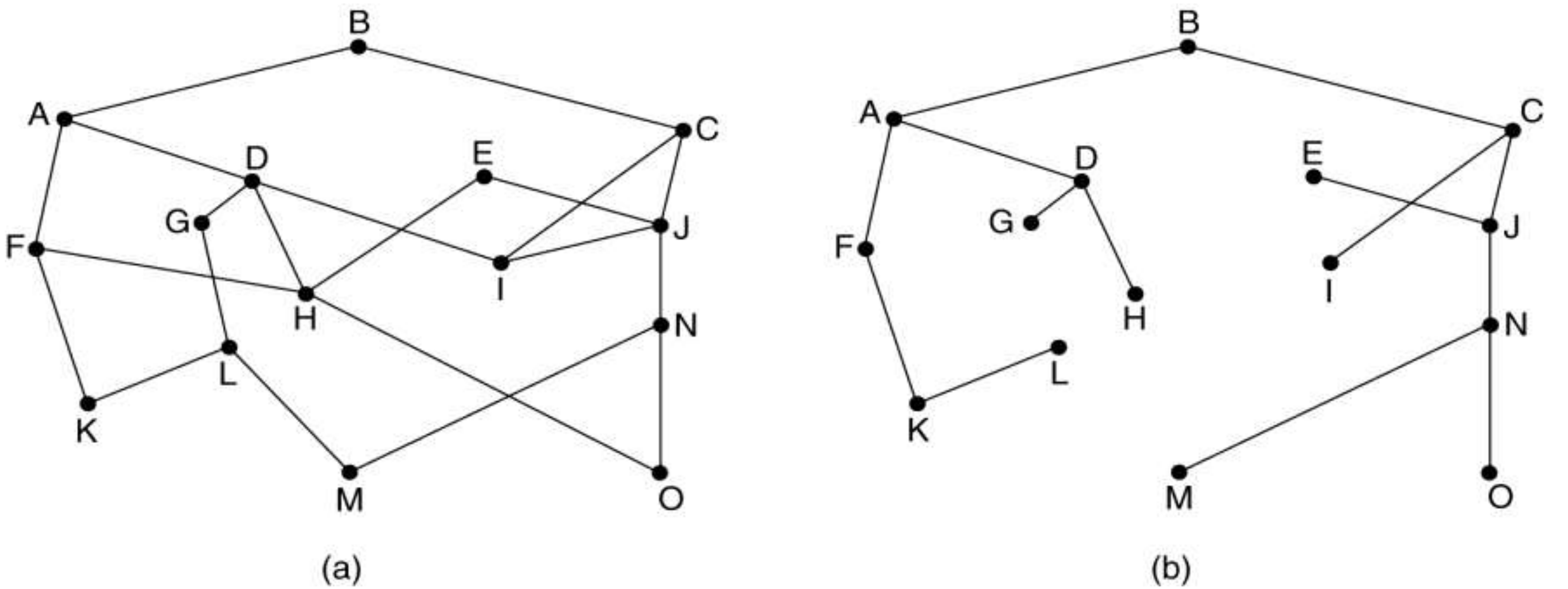
ROUTING ALGORITHMS (2)

- Conflict between fairness and optimality.



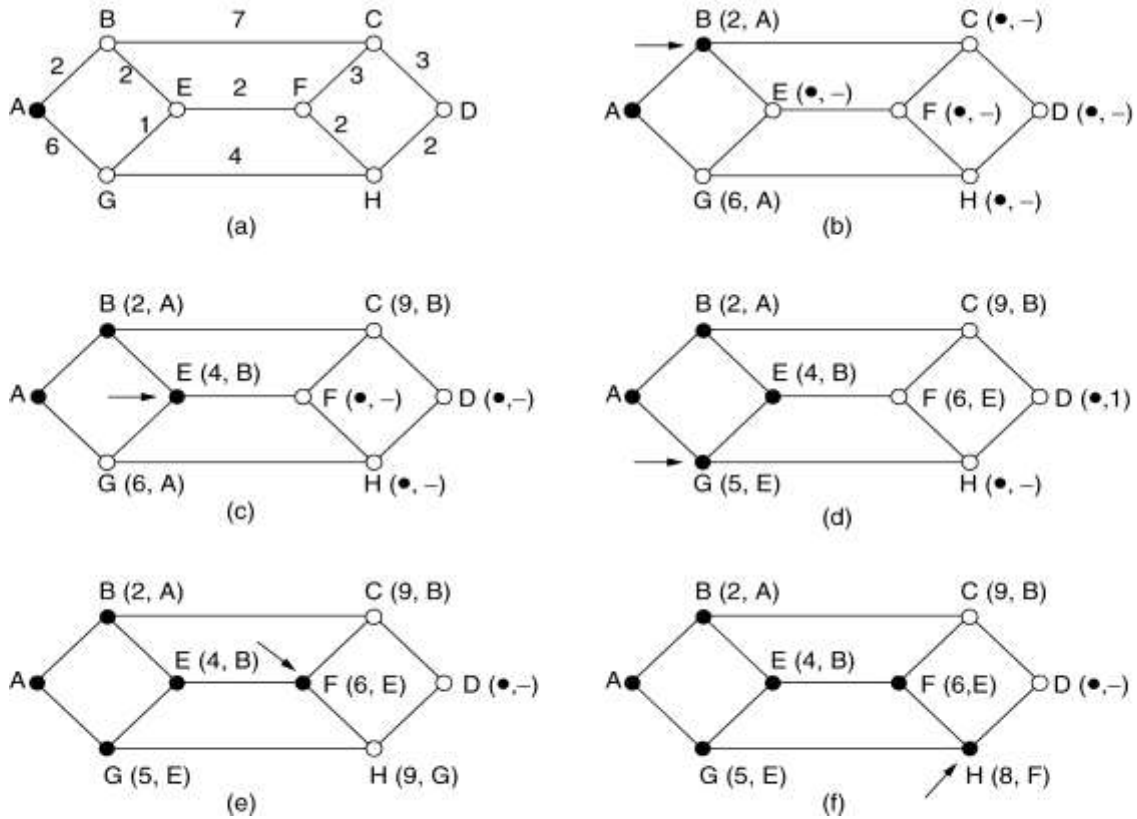
THE OPTIMALITY PRINCIPLE

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



SHORTEST PATH ROUTING

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



FLOODING

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

5-8 top

```
#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 */
```

FLOODING (2)

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

5-8 bottom

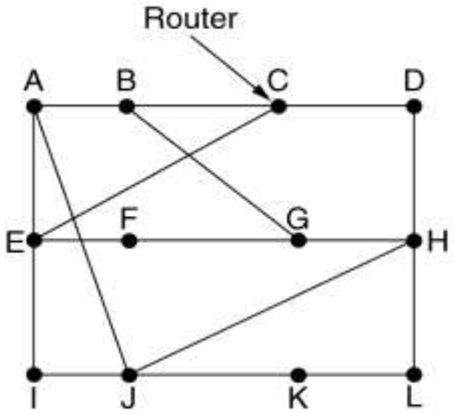
```
do {                                        /* Is there a better path from k? */
     for (i = 0; i < n; i++)                /* this graph has n nodes */
          if (dist[k][i] != 0 && state[i].label == tentative) {
               if (state[k].length + dist[k][i] < state[i].length) {
                    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);
}
```

DISTANCE VECTOR ROUTING

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



(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)

DISTANCE VECTOR ROUTING (2)

- 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
		⋮			
	•	•	•	•	

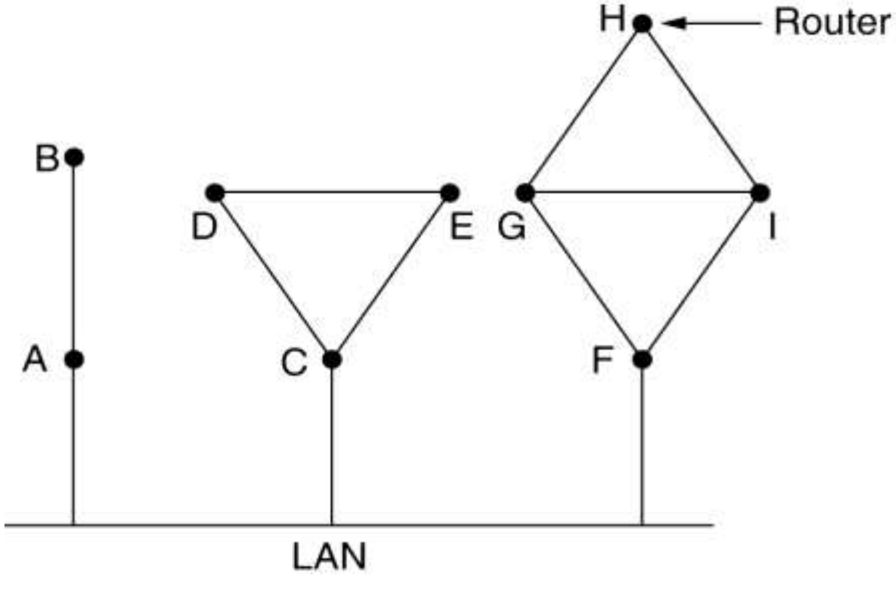
(b)

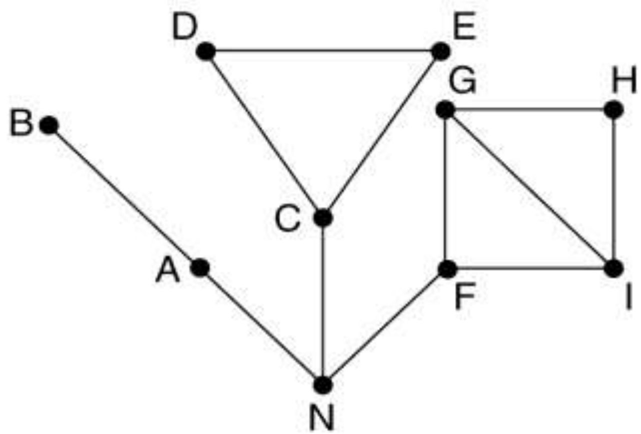
LINK STATE ROUTING

- Each router must do the following:
- Discover its neighbors, learn their network address.
- Measure the delay or cost to each of its neighbors.
- Construct a packet telling all it has just learned.
- Send this packet to all other routers.
- Compute the shortest path to every other router.

LEARNING ABOUT THE NEIGHBORS

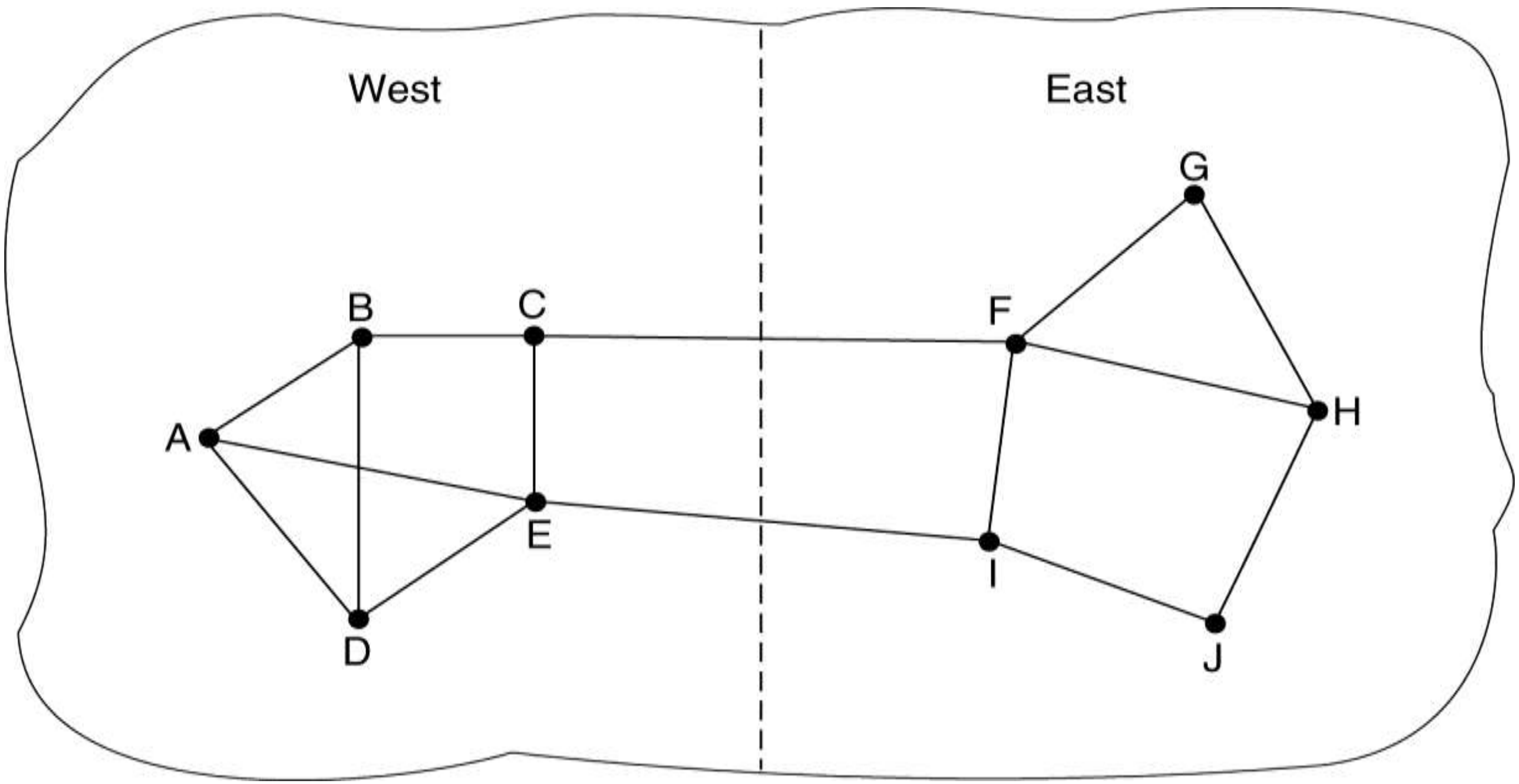
- (a) Nine routers and a LAN. (b) A graph model of (a).



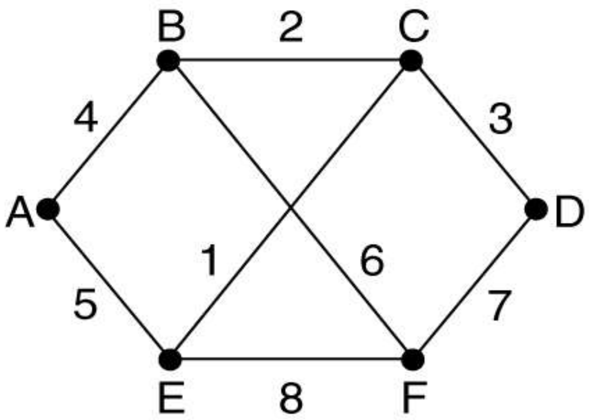


MEASURING LINE COST

- A subnet in which the East and West parts are connected by two lines.



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)

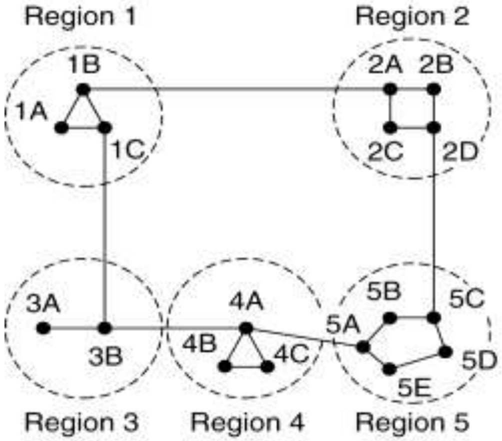
DISTRIBUTING THE LINK STATE PACKETS

- The packet buffer for router B in the previous slide (Fig. 5-13).

Source	Seq.	Age	Send flags			ACK flags			Data
			A	C	F	A	C	F	
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	

HIERARCHICAL ROUTING

- 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

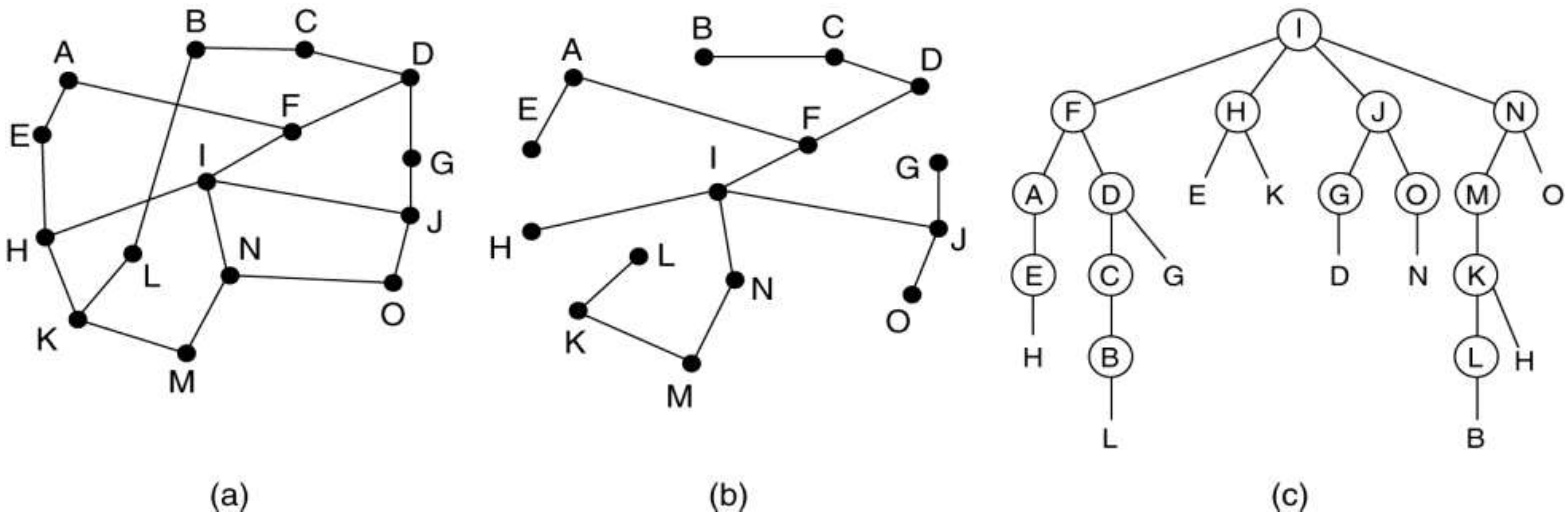
(b)

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

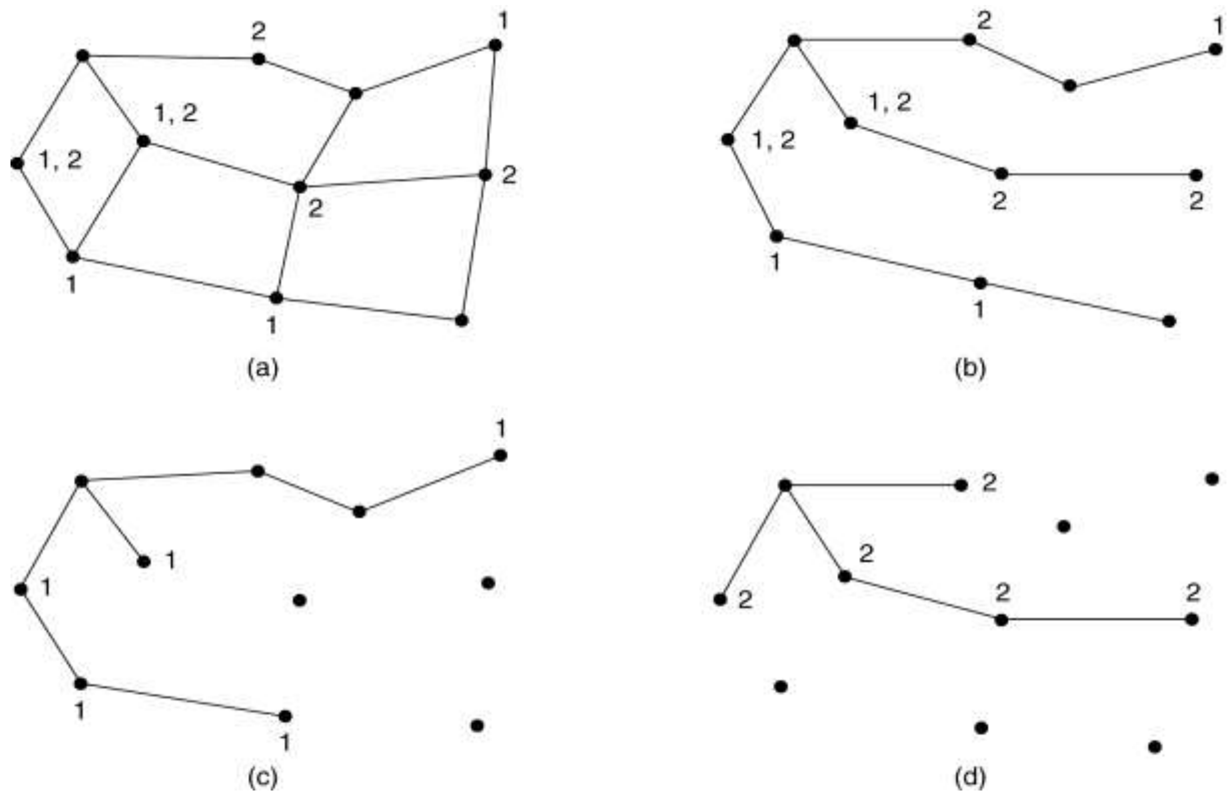
(c)

BROADCAST ROUTING



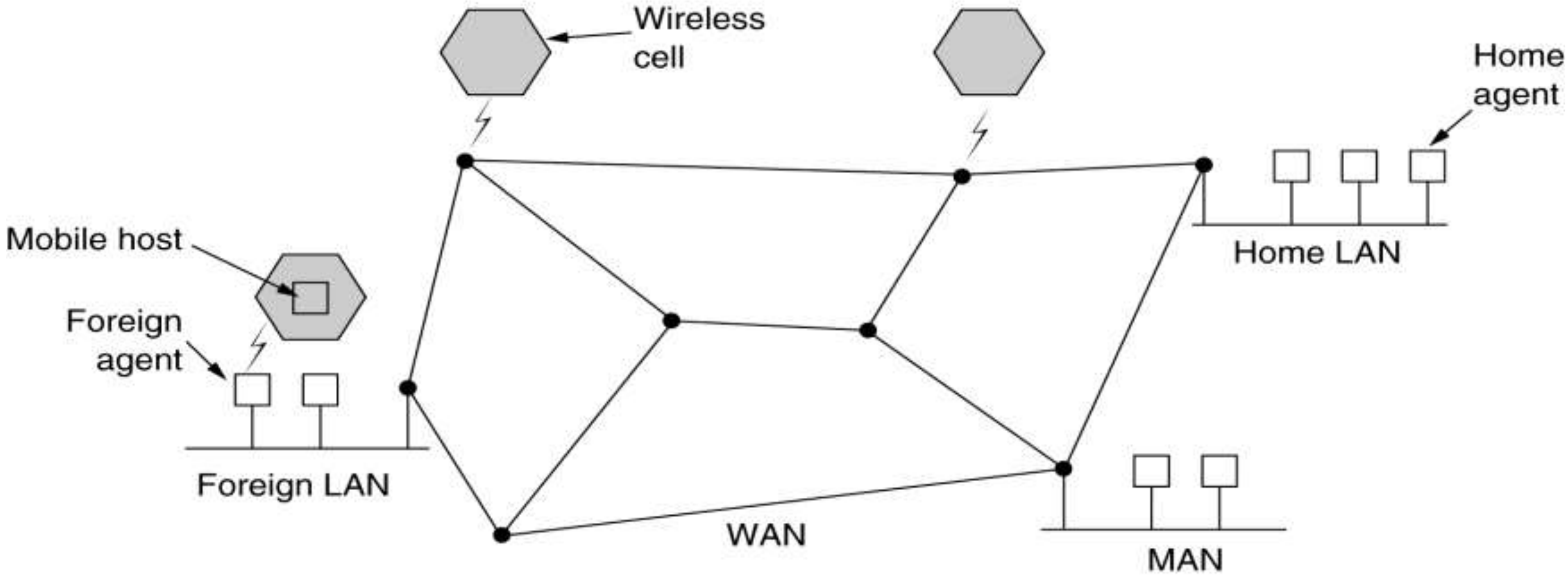
MULTICAST ROUTING

- (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.



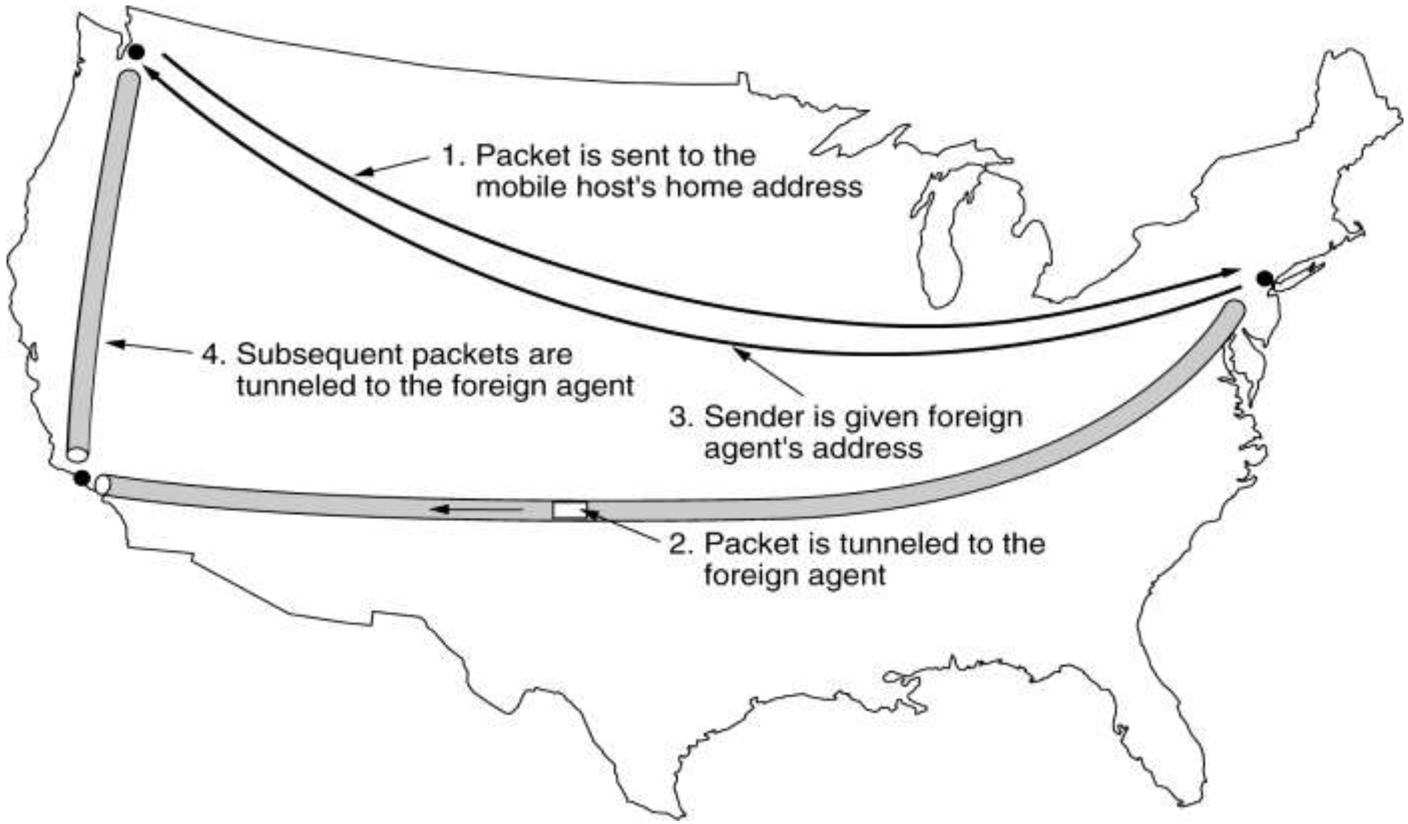
ROUTING FOR MOBILE HOSTS

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



ROUTING FOR MOBILE HOSTS (2)

- Packet routing for mobile users.

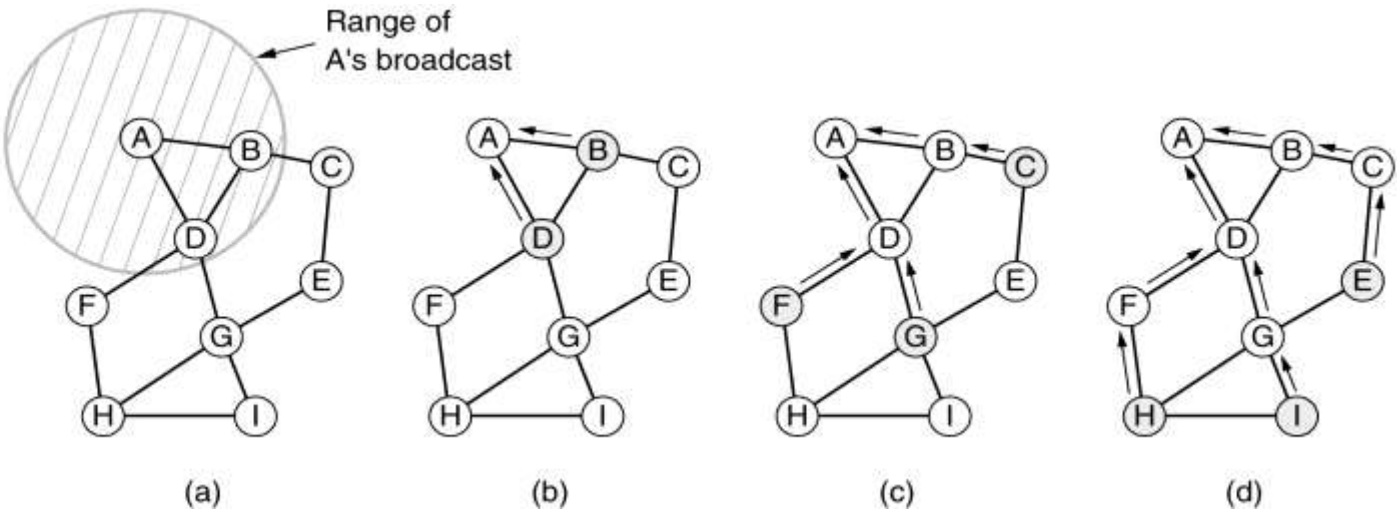


ROUTING IN AD HOC NETWORKS

- Possibilities when the routers are mobile:
- Military vehicles on battlefield.
  - No infrastructure.
- A fleet of ships at sea.
  - All moving all the time
- Emergency works at earthquake .
  - The infrastructure destroyed.
- A gathering of people with notebook computers.
  - In an area lacking 802.11.

ROUTE DISCOVERY

- (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.



ROUTE DISCOVERY (2)

- Format of a ROUTE REQUEST packet.

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

ROUTE DISCOVERY (3)

- Format of a ROUTE REPLY packet.

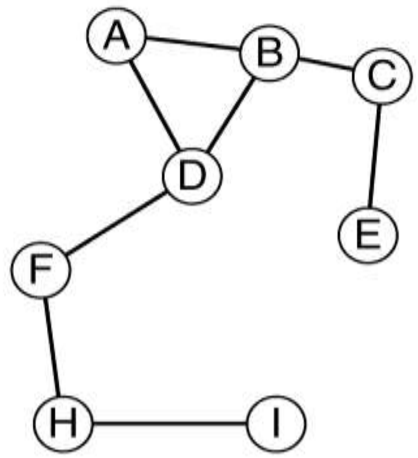
Source address	Destination address	Destination sequence #	Hop count	Lifetime

ROUTE MAINTENANCE

- (a) D's routing table before G goes down.
- (b) The graph after G has gone down.

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	

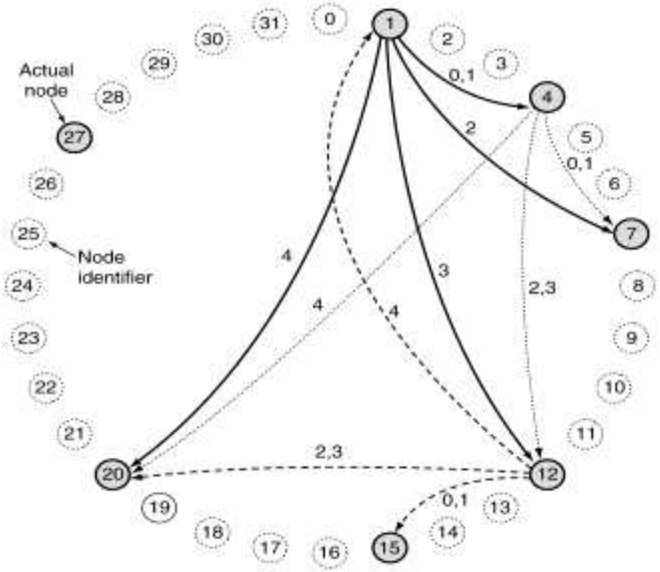
(a)



(b)

NODE LOOKUP IN PEER-TO-PEER NETWORKS

- (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.



(a)

Node 1's finger table

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

Node 4's finger table

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

Node 12's finger table

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

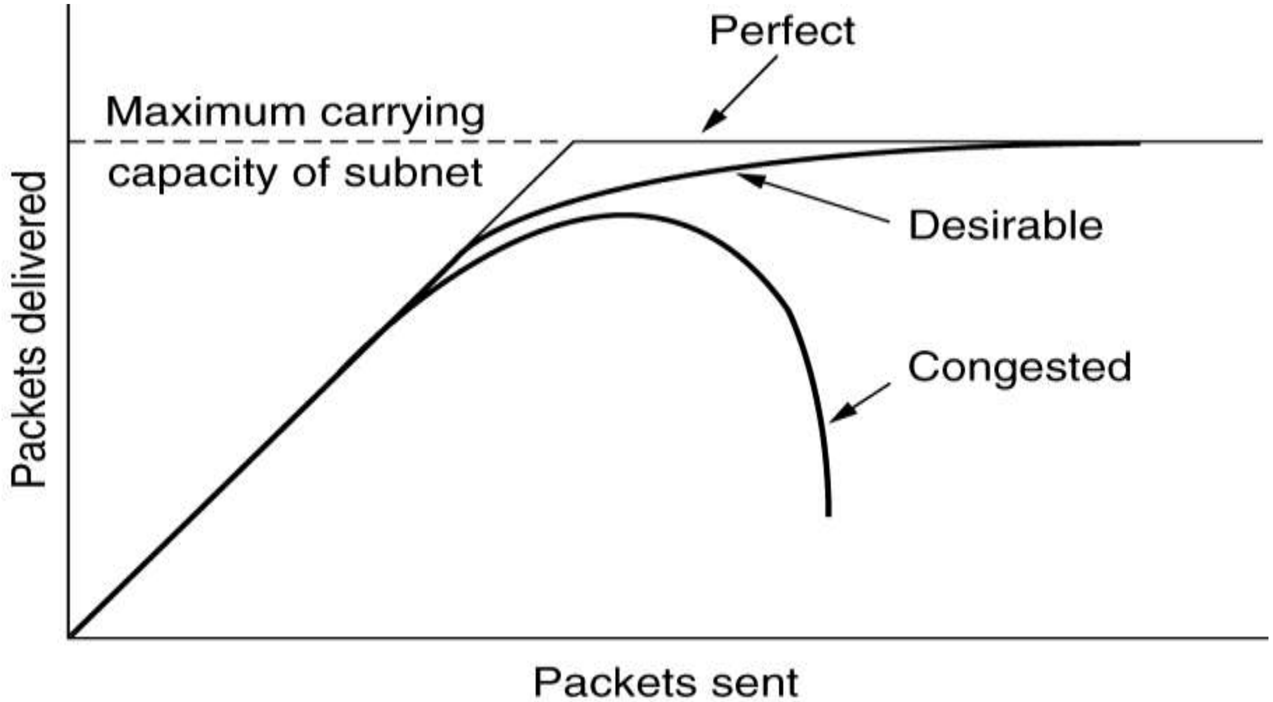
(b)

CONGESTION CONTROL ALGORITHMS

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

CONGESTION

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



GENERAL PRINCIPLES OF CONGESTION CONTROL

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

CONGESTION PREVENTION POLICIES

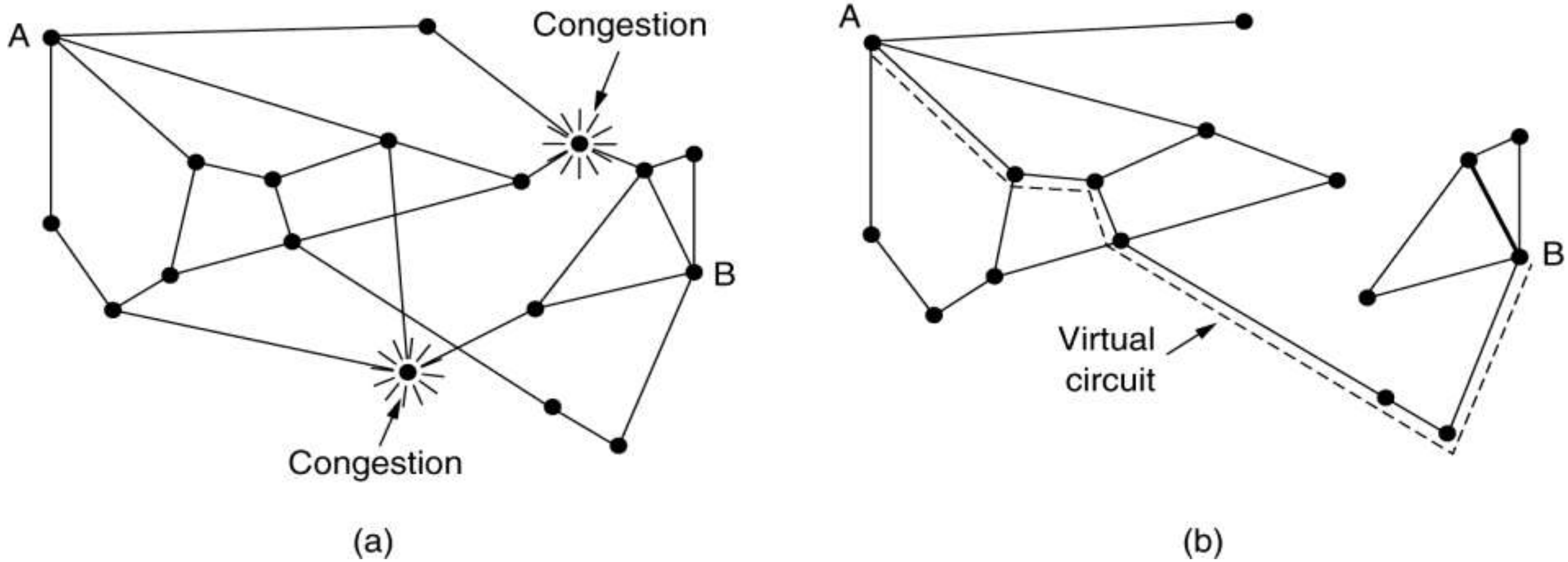
- Policies that affect congestion.

5-26

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 VIRTUAL-CIRCUIT SUBNETS

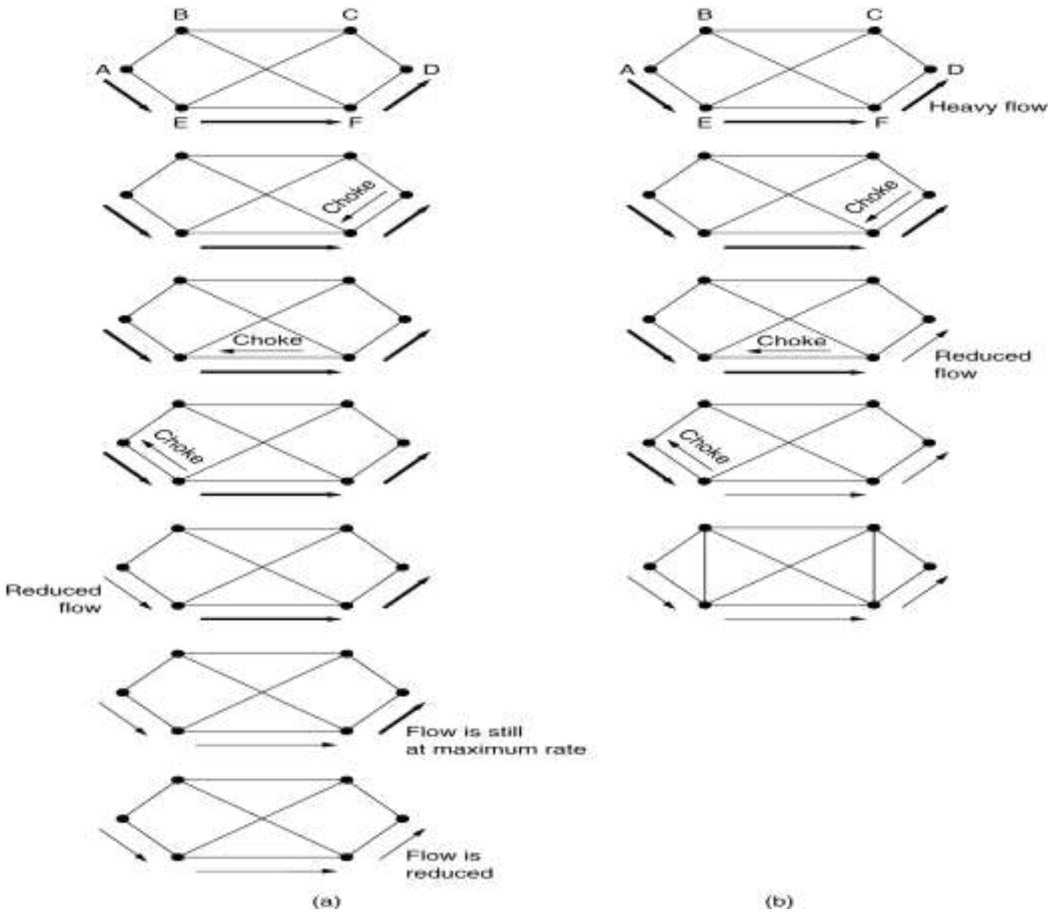
- (a) A congested subnet. (b) A redrawn subnet, eliminates congestion and a virtual circuit from A to B.



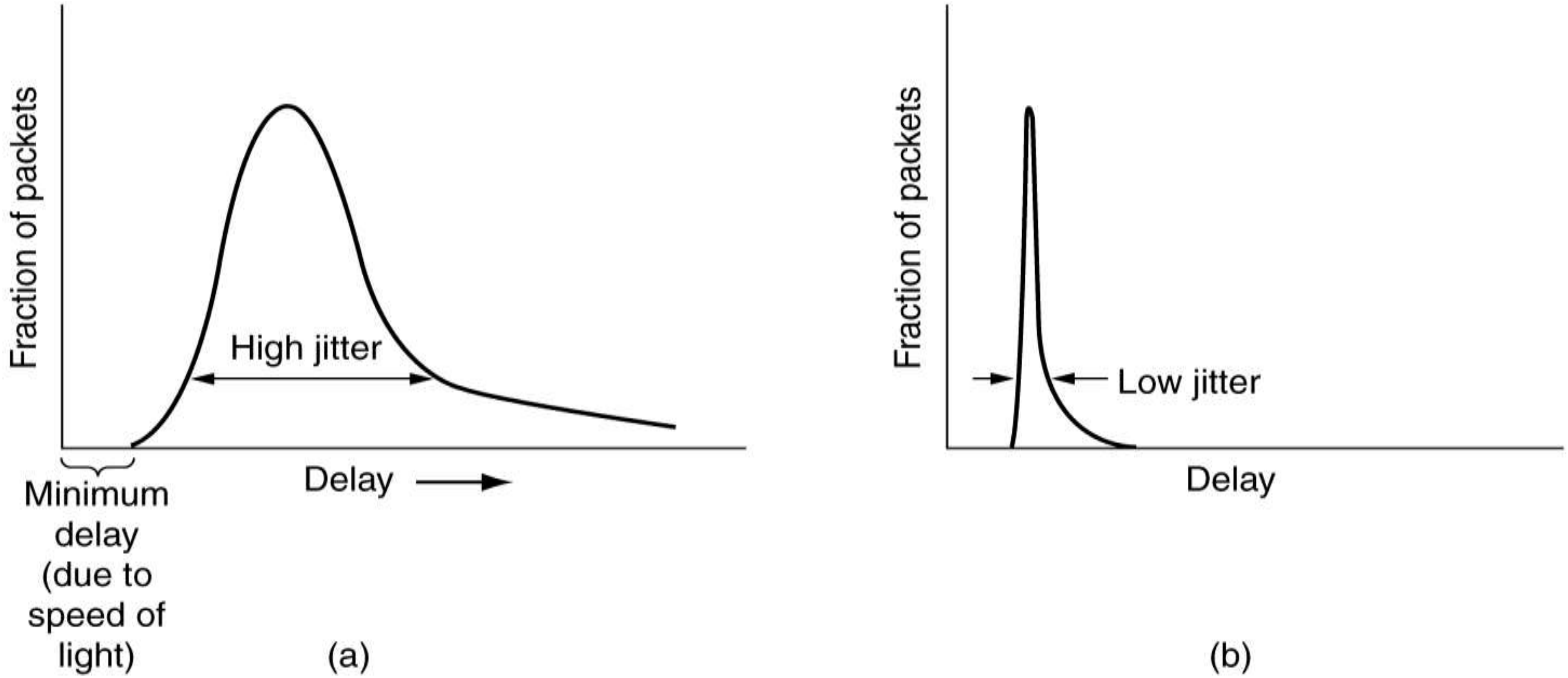
HOP-BY-HOP CHOKE PACKETS

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

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



JITTER CONTROL



QUALITY OF SERVICE

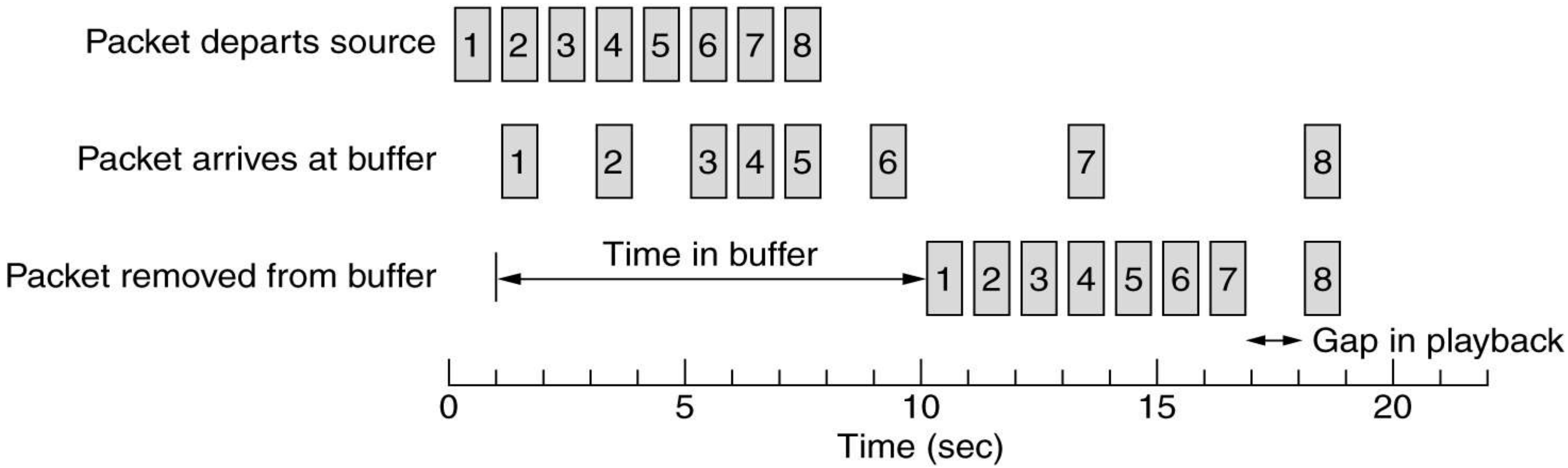
- Requirements
- Techniques for Achieving Good Quality of Service
- Integrated Services
- Differentiated Services
- Label Switching and MPLS

REQUIREMENTS

- How stringent the quality-of-service requirements are.

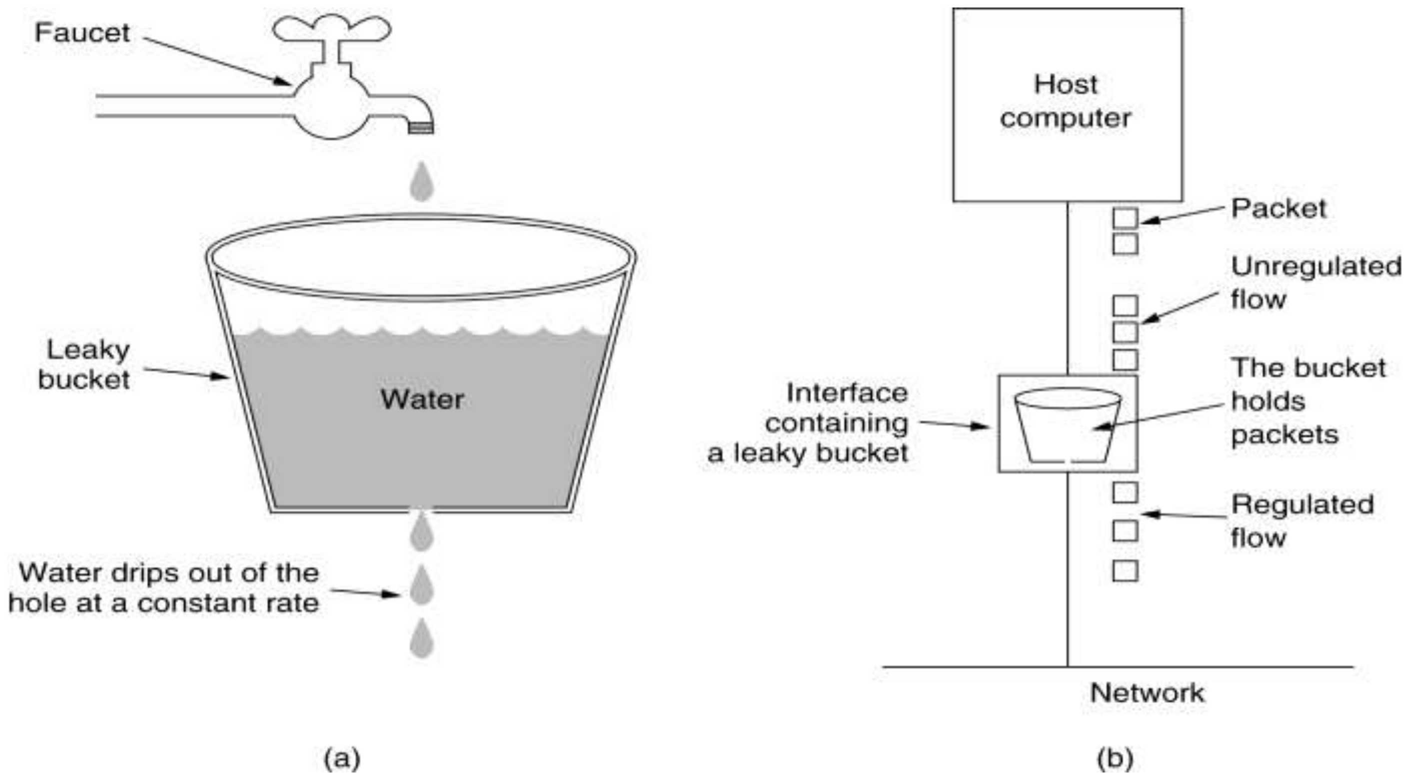
Application	Reliability	Delay	Jitter	Bandwidth
E-mail	High	Low	Low	Low
File transfer	High	Low	Low	Medium
Web access	High	Medium	Low	Medium
Remote login	High	Medium	Medium	Low
Audio on demand	Low	Low	High	Medium
Video on demand	Low	Low	High	High
Telephony	Low	High	High	Low
Videoconferencing	Low	High	High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

BUFFERING


Packet departs source
1 2 3 4 5 6 7 8
Packet arrives at buffer
1 2 3 4 5 6 7 8
Packet removed from buffer
Time in buffer
1 2 3 4 5 6 7 8
Gap in playback
0 5 10 15 20
Time (sec)

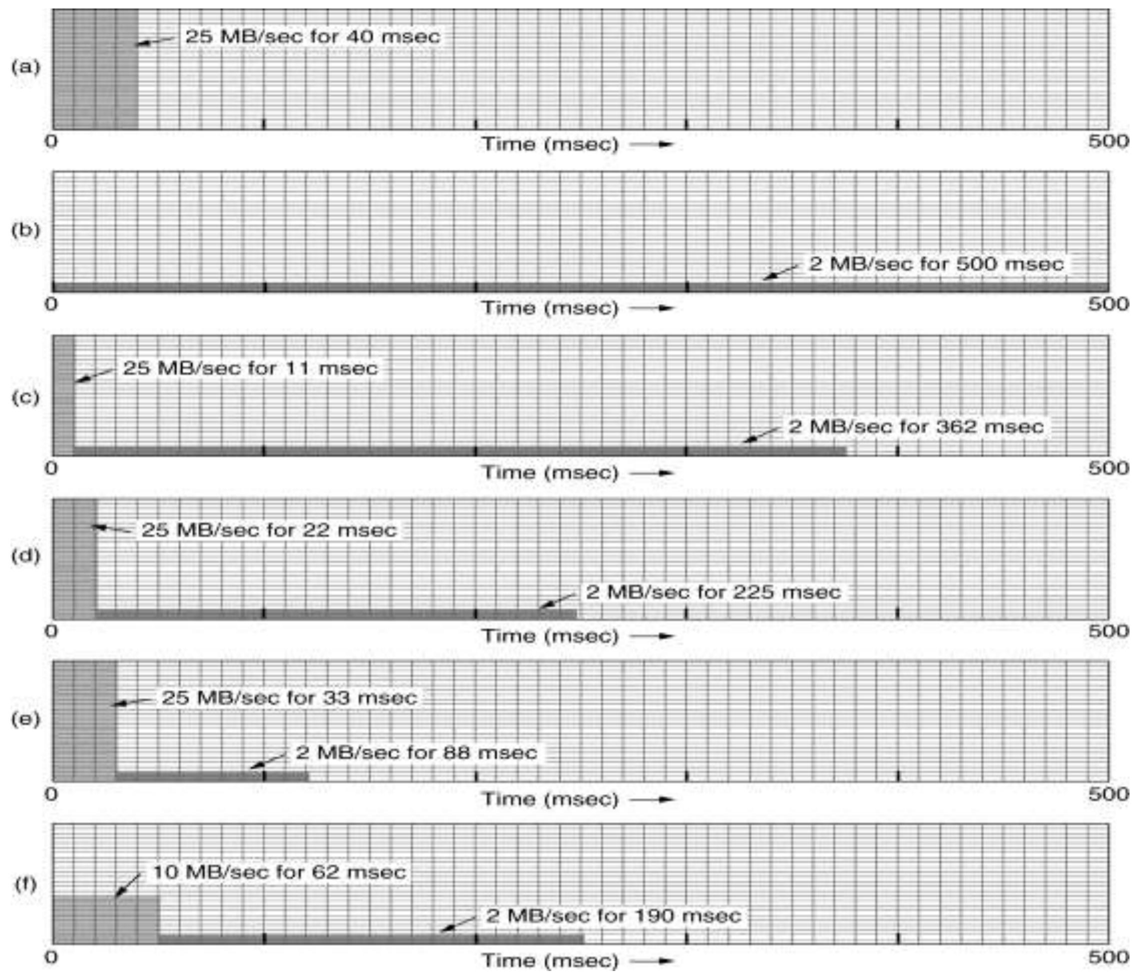
THE LEAKY BUCKET ALGORITHM

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



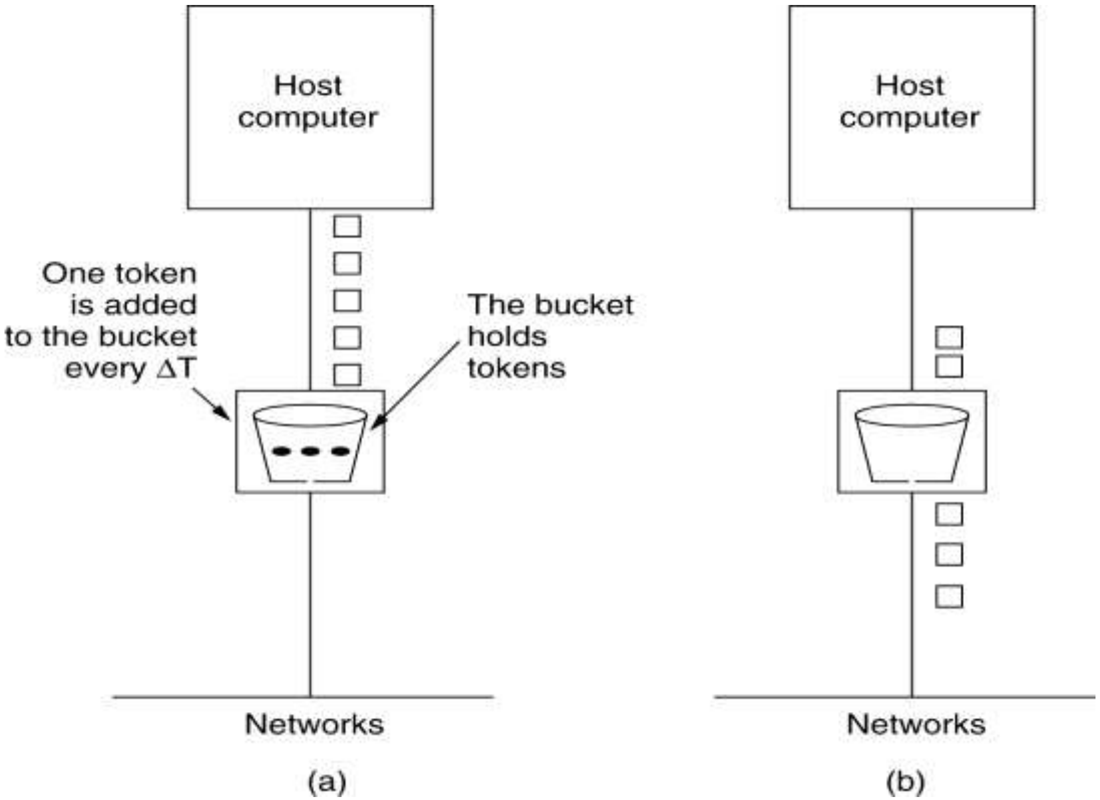
THE LEAKY BUCKET ALGORITHM

(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.



THE TOKEN BUCKET ALGORITHM

- (a) Before. (b) After.

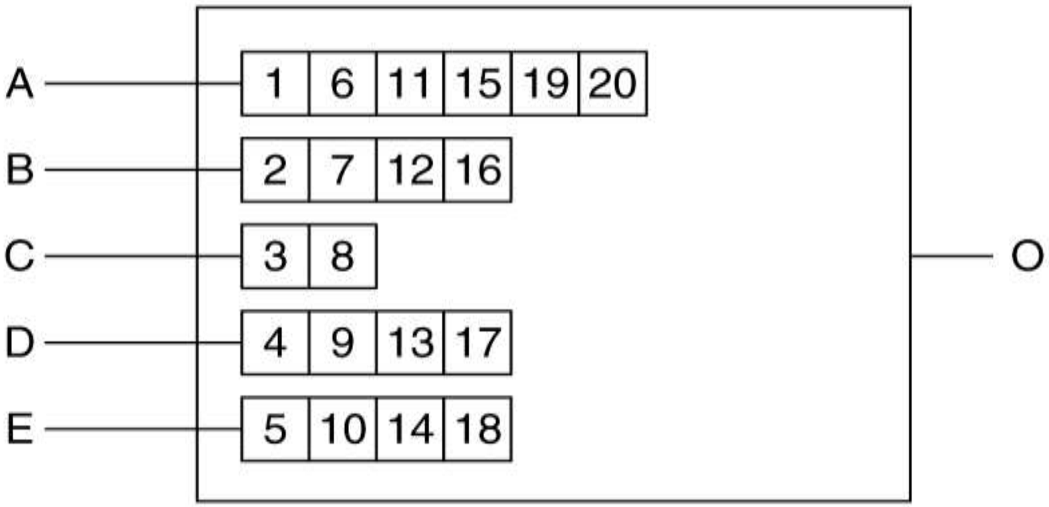


ADMISSION CONTROL

- An example of flow specification.

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

PACKET SCHEDULING



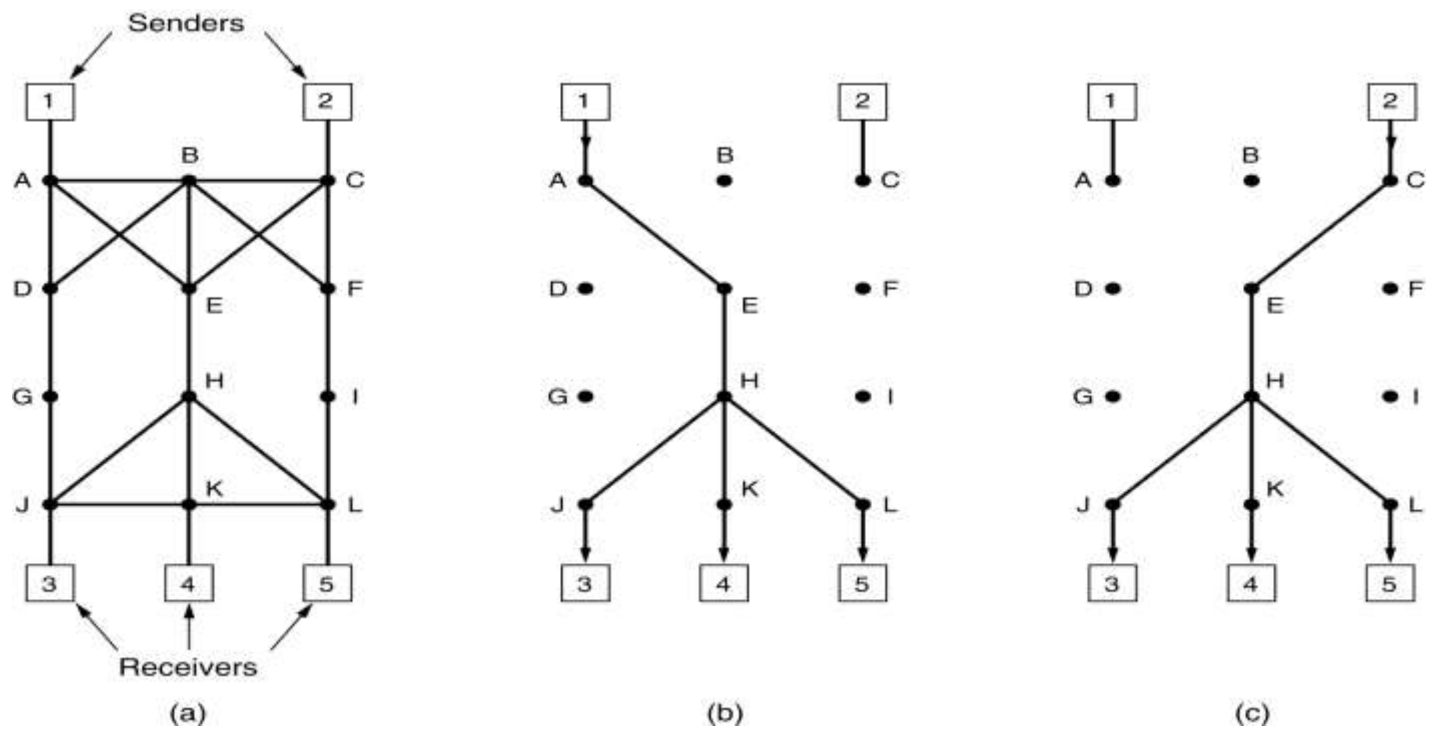
(a)

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

(b)

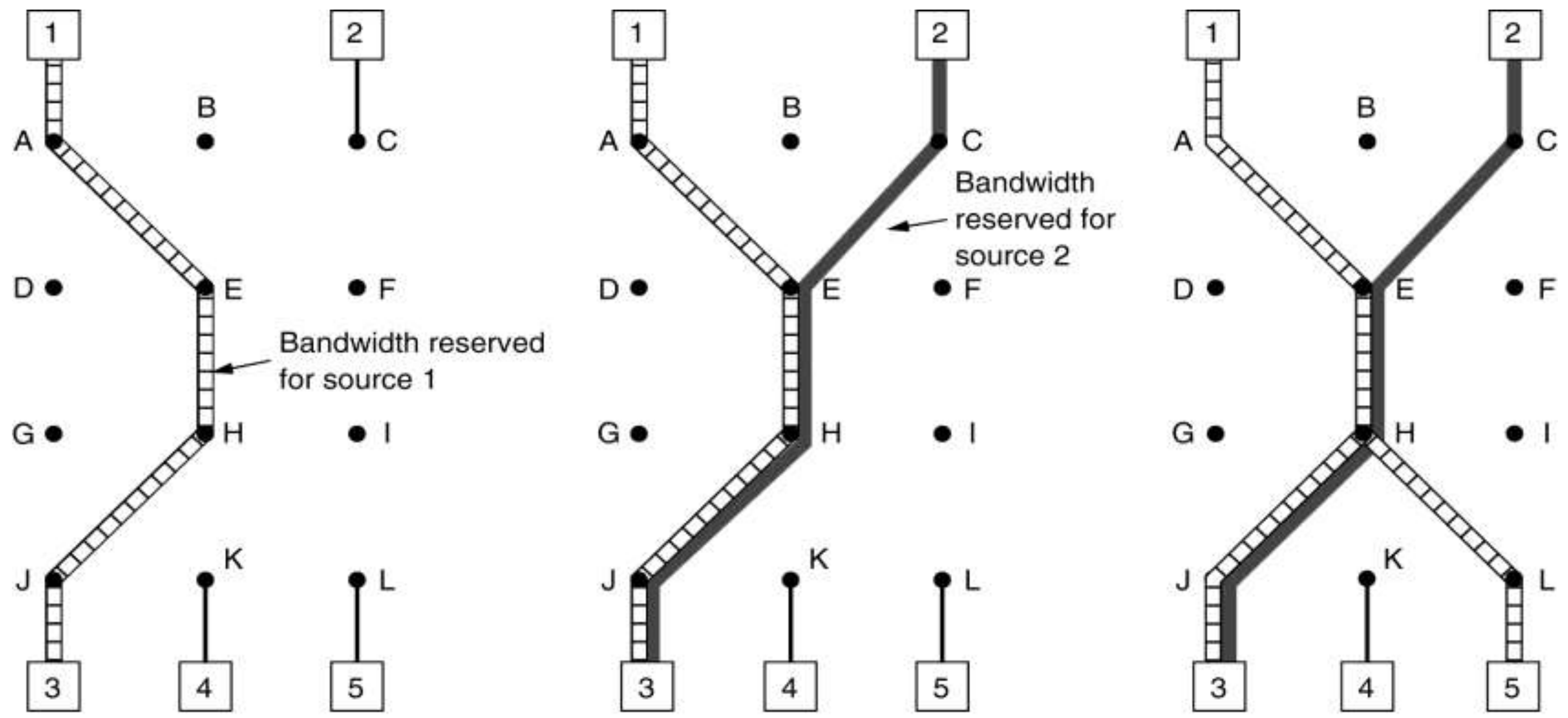
RSVP-THE RESERVATION PROTOCOL

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



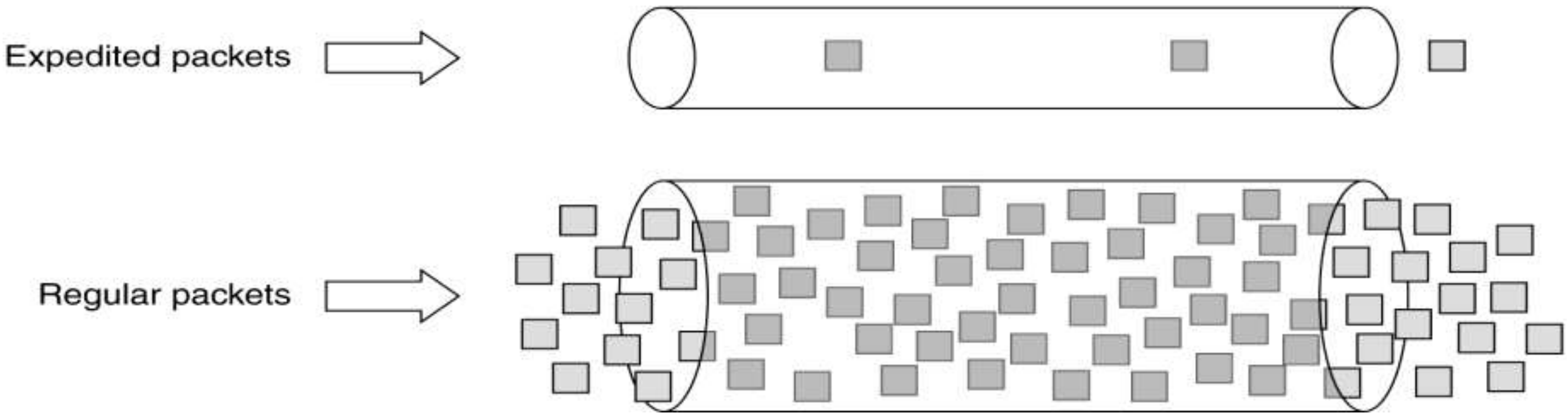
RSVP-THE RESERVATION PROTOCOL (2)

- (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.

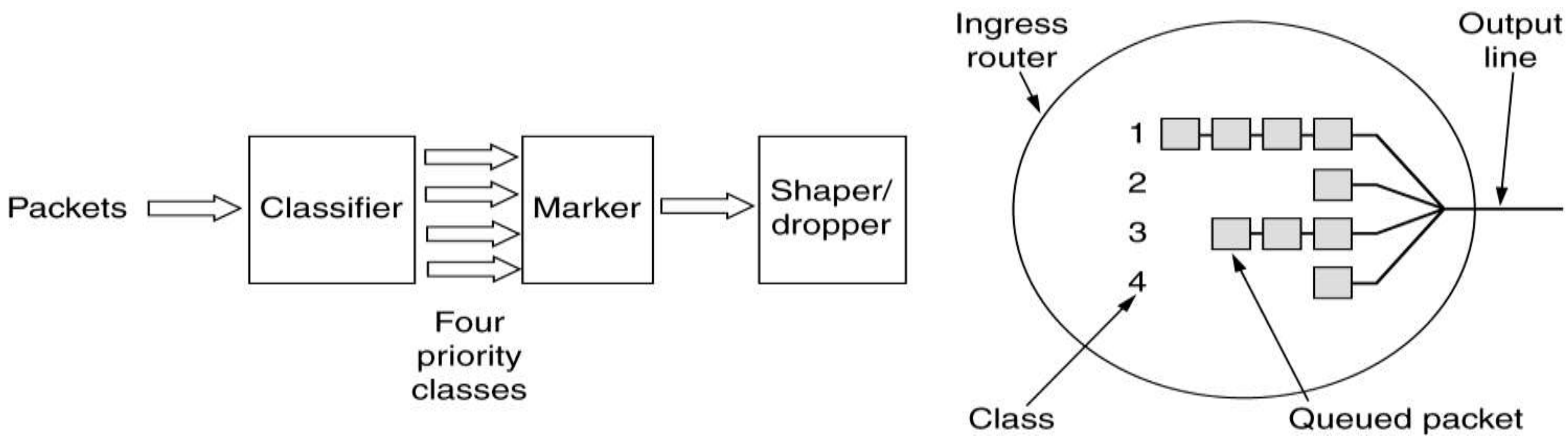


(a) (b) (c)

EXPEDITED FORWARDING

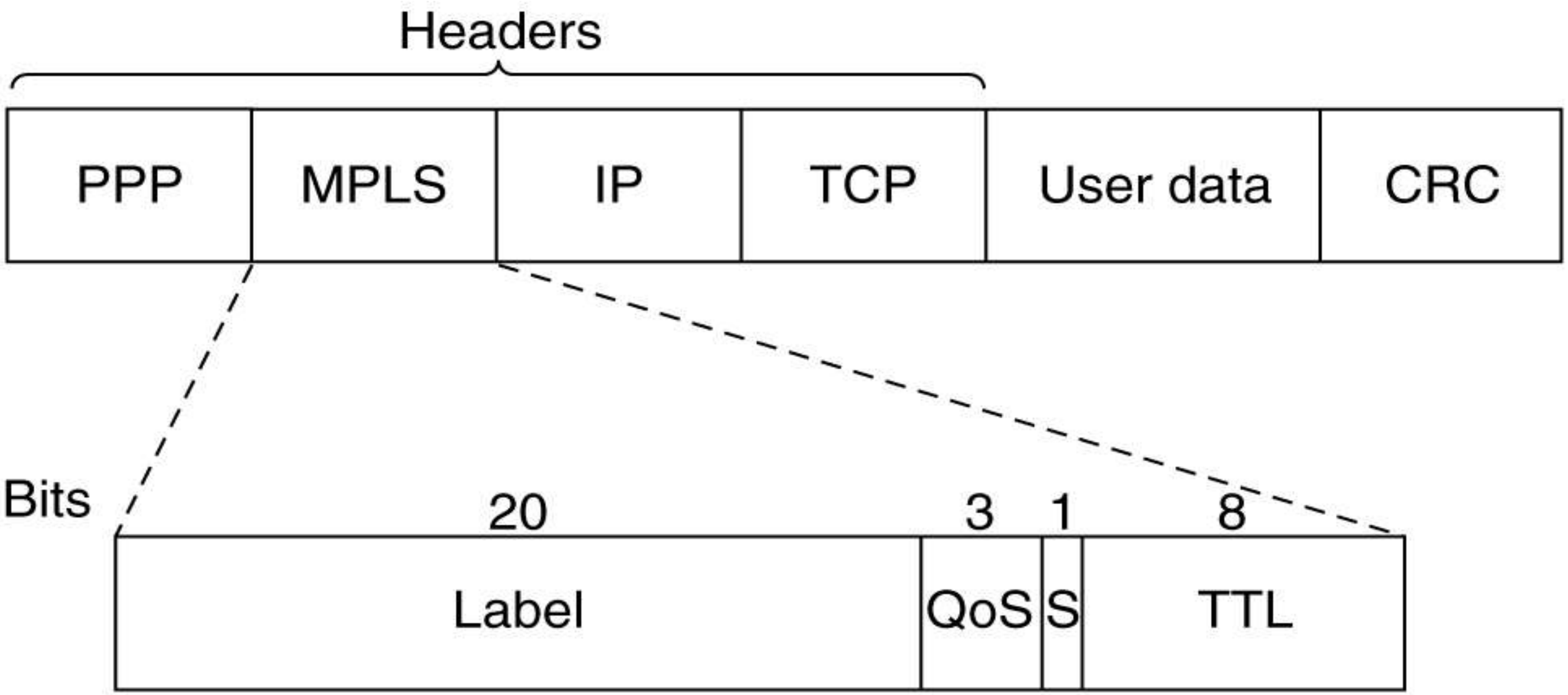


ASSURED FORWARDING



LABEL SWITCHING AND MPLS

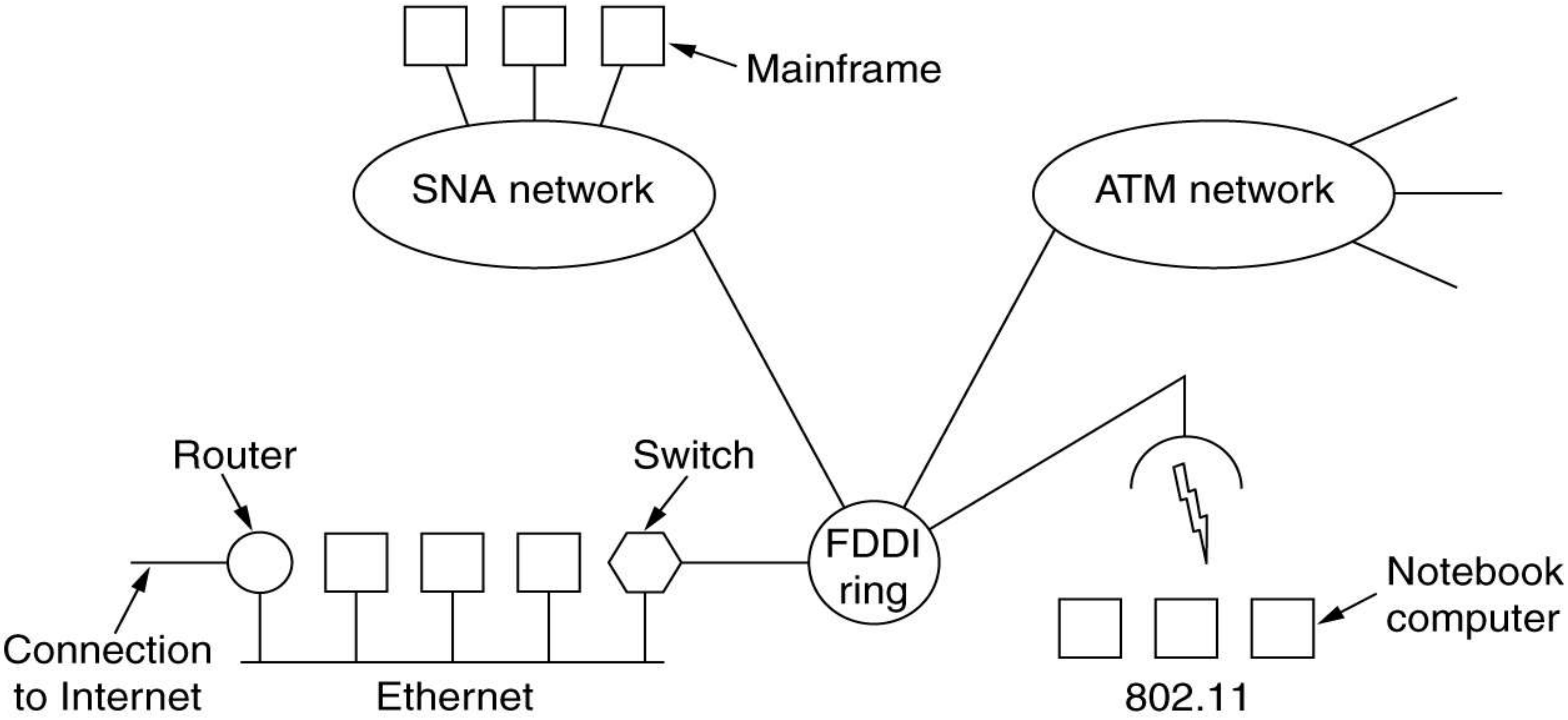
- Transmitting a TCP segment using IP, MPLS, and PPP.



INTERNETWORKING

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

CONNECTING NETWORKS



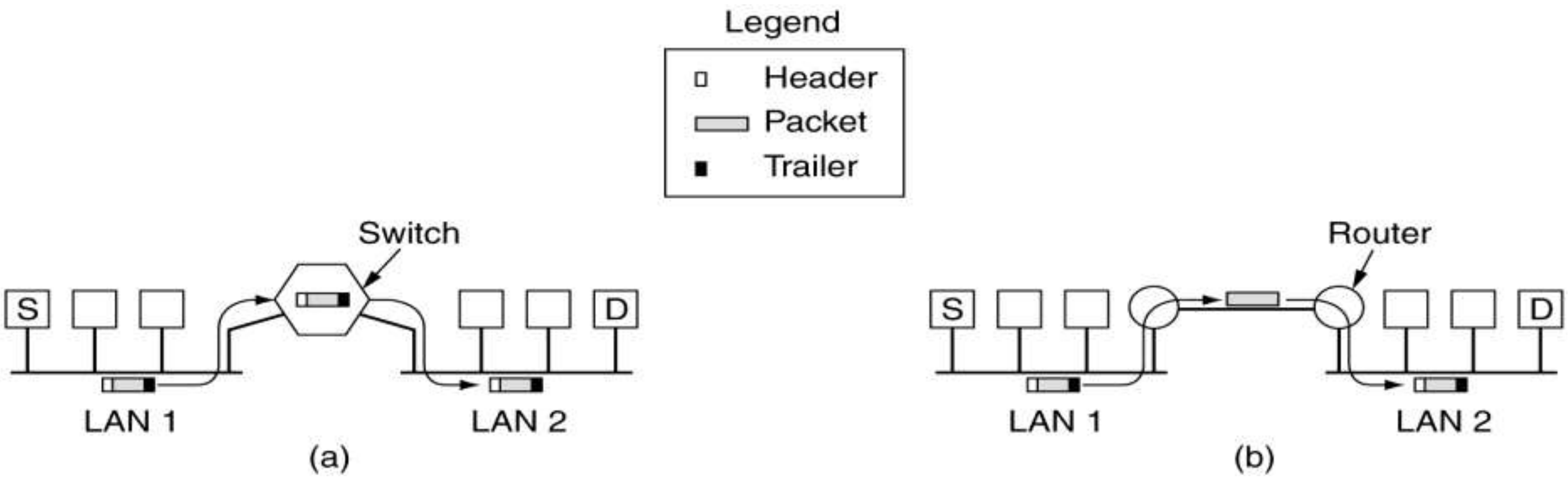
HOW NETWORKS DIFFER

- Some of the many ways networks can 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

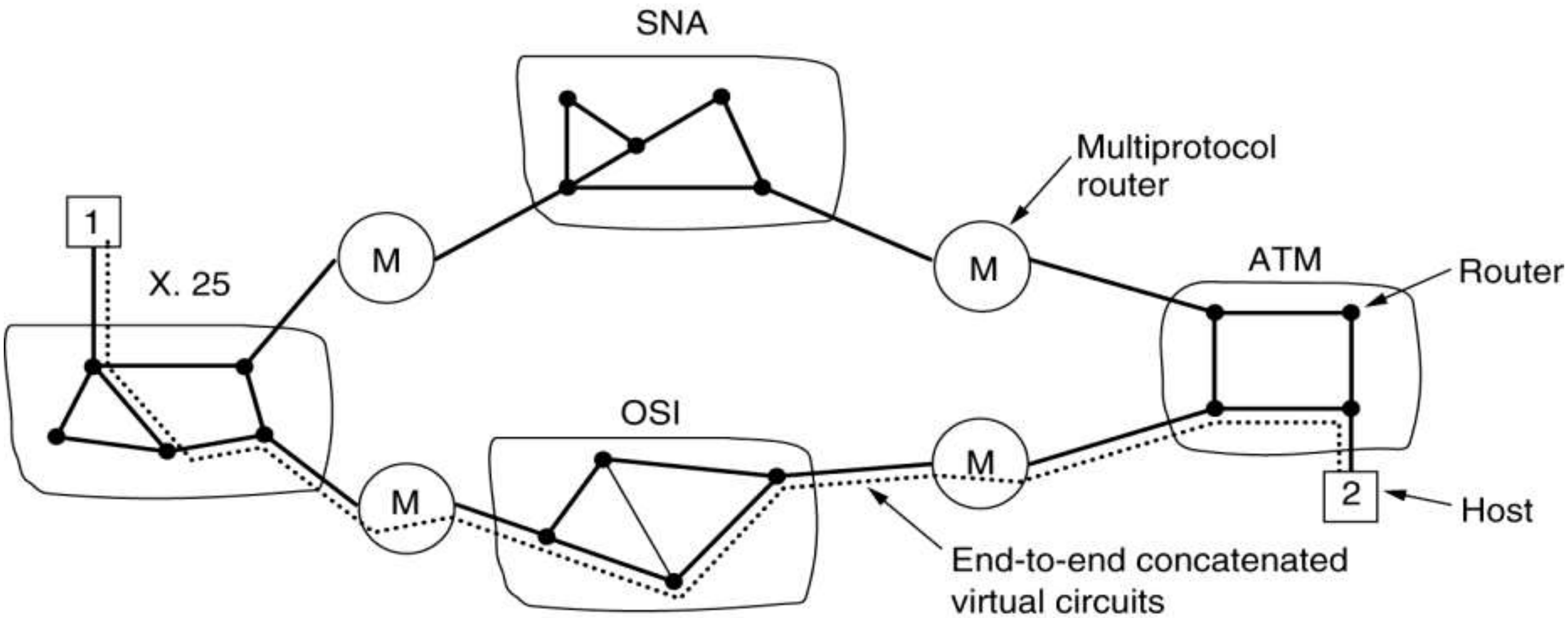
HOW NETWORKS CAN BE CONNECTED

- (a) Two Ethernets connected by a switch.
- (b) Two Ethernets connected by routers.

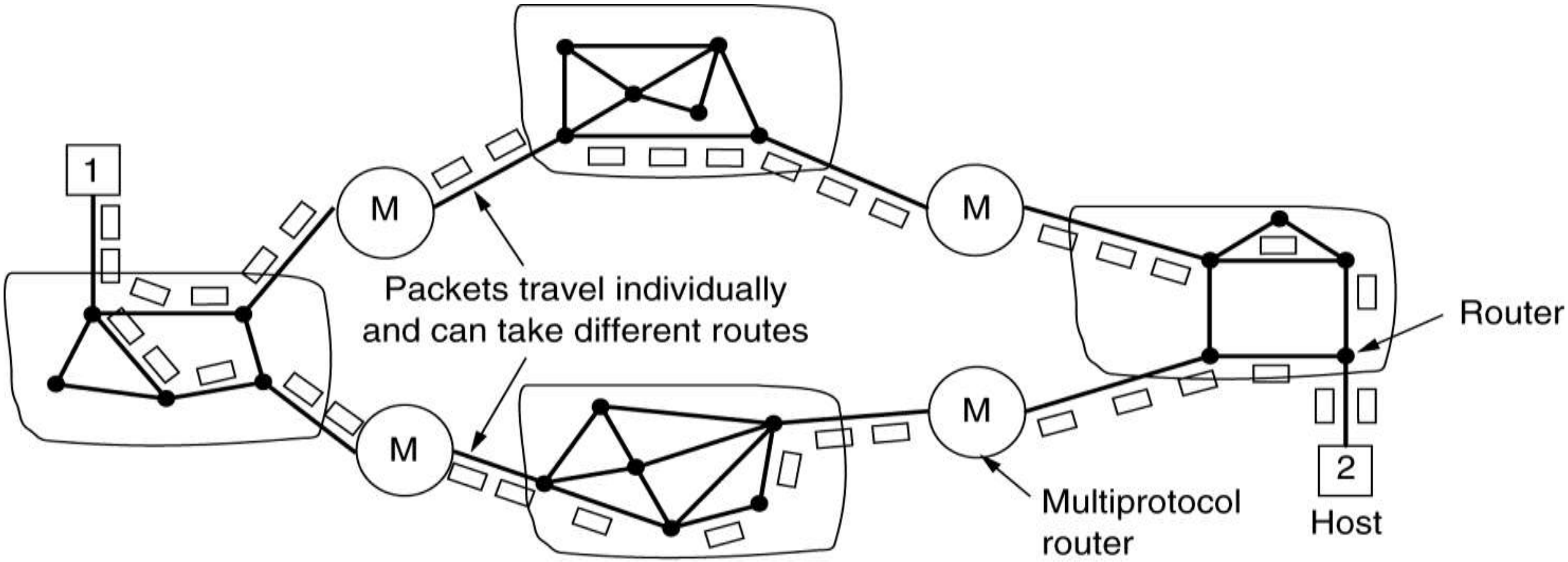


CONCATENATED VIRTUAL CIRCUITS

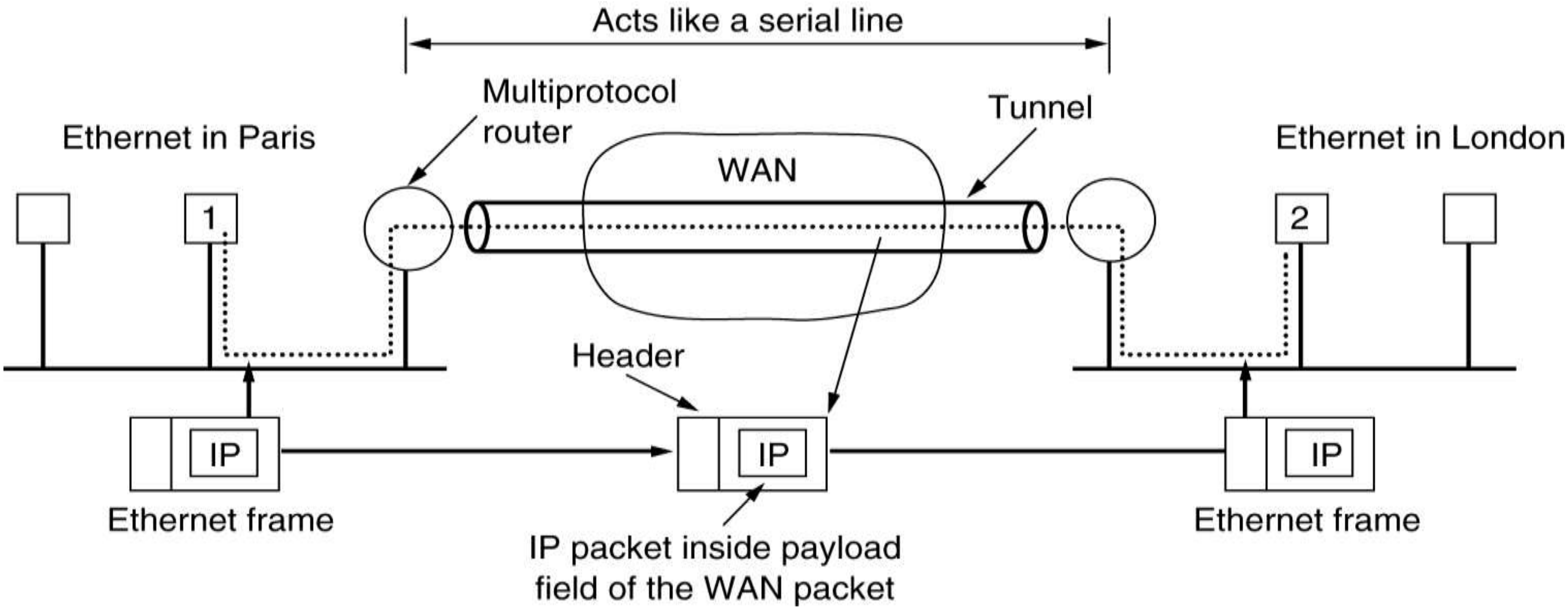
- Internetworking using concatenated virtual circuits.



CONNECTIONLESS INTERNETWORKING

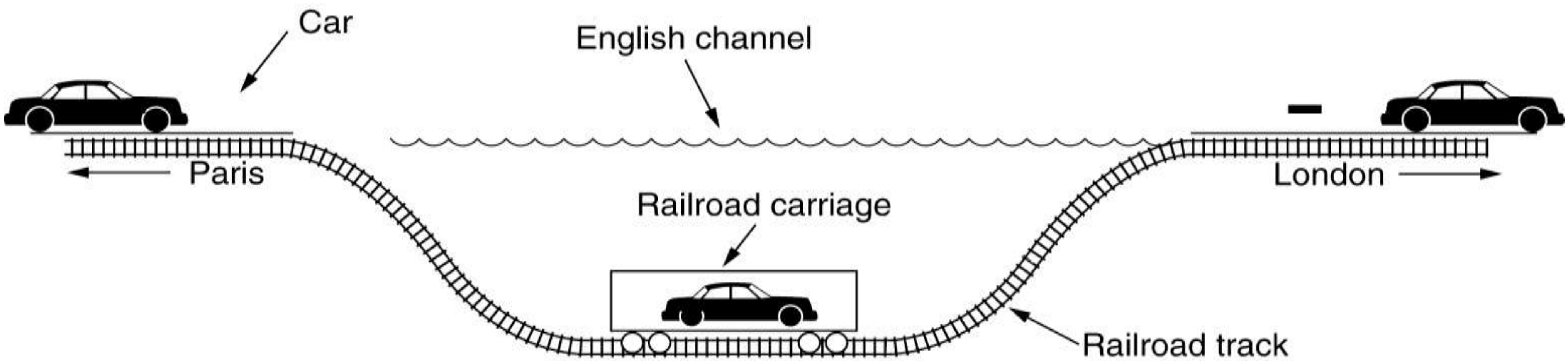


TUNNELING



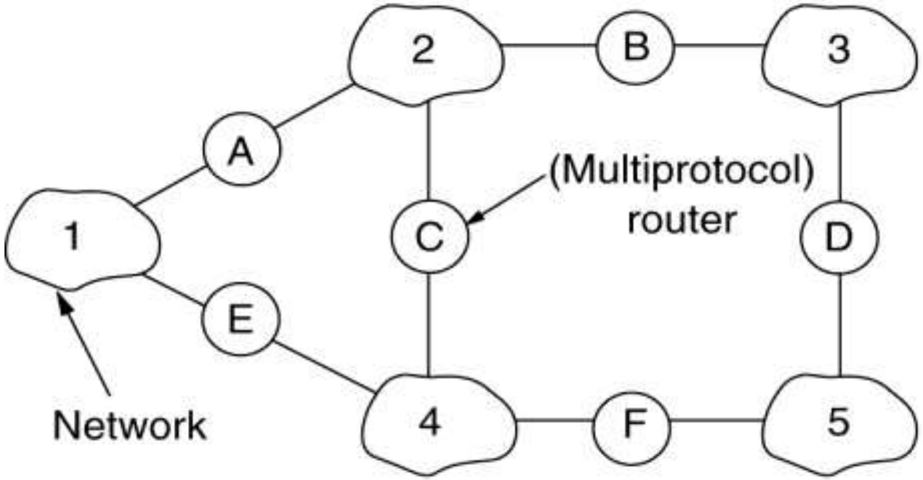
Acts like a serial line
Multiprotocol router
Tunnel
Ethernet in Paris
WAN
Ethernet in London
1
2
Header
IP
IP
IP
Ethernet frame
IP packet inside payload field of the WAN packet
Ethernet frame

TUNNELING (2)

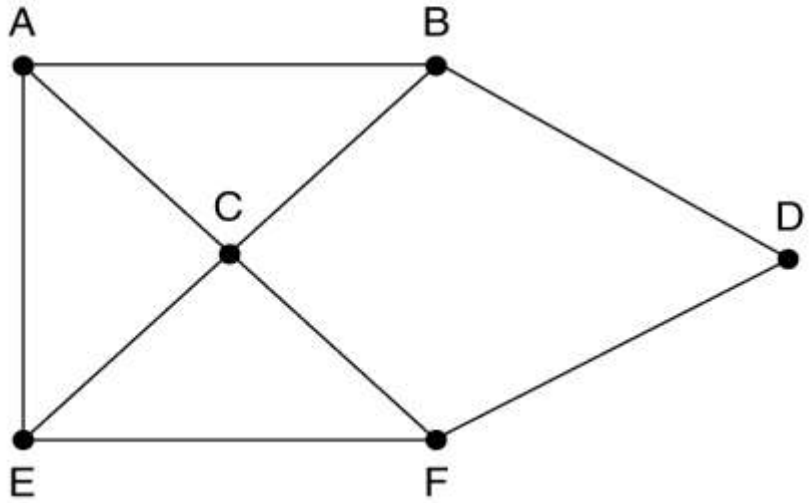


Car
English channel
Paris
London
Railroad carriage
Railroad track

INTERNETWORK ROUTING



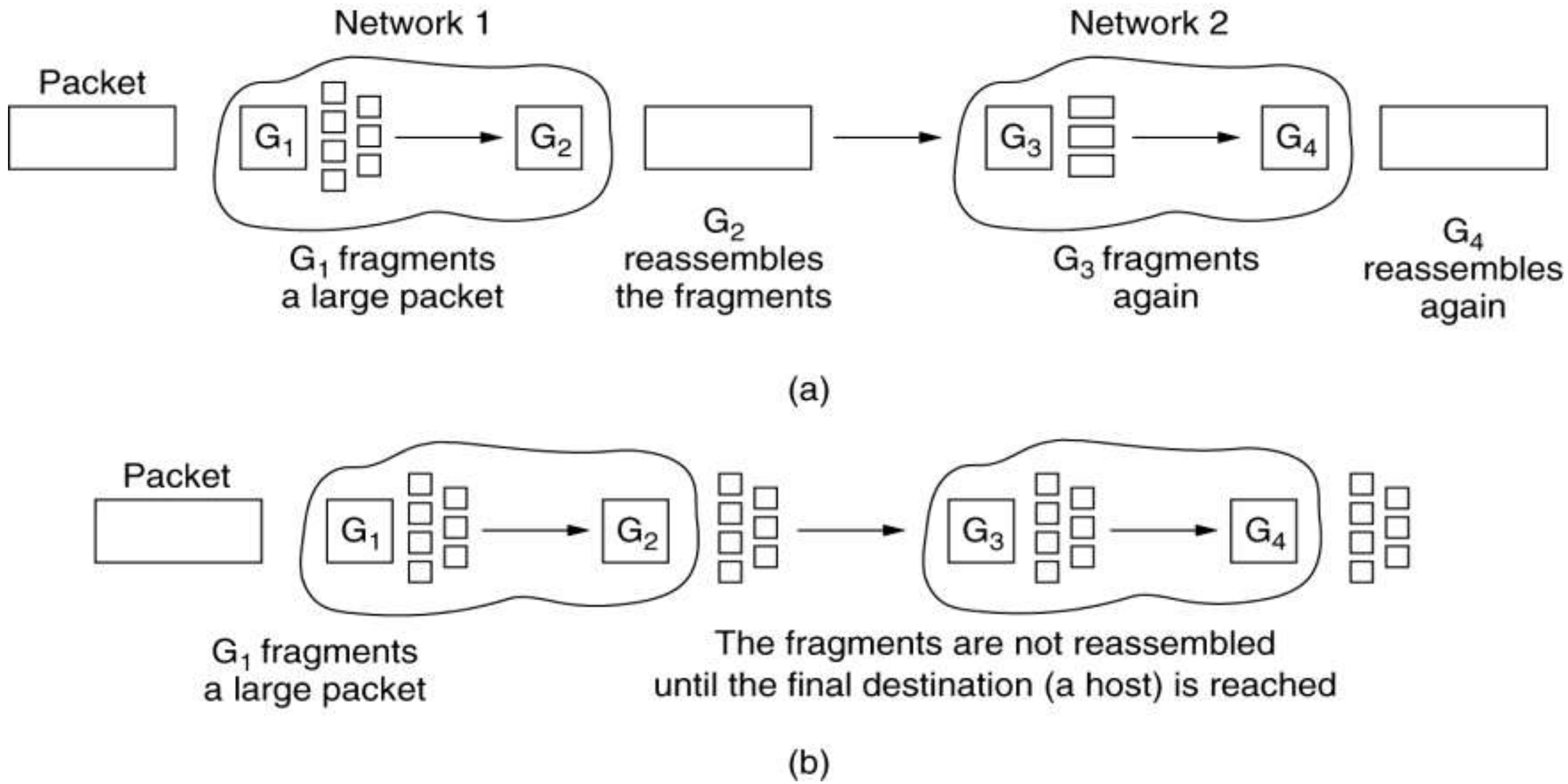
(a)



(b)

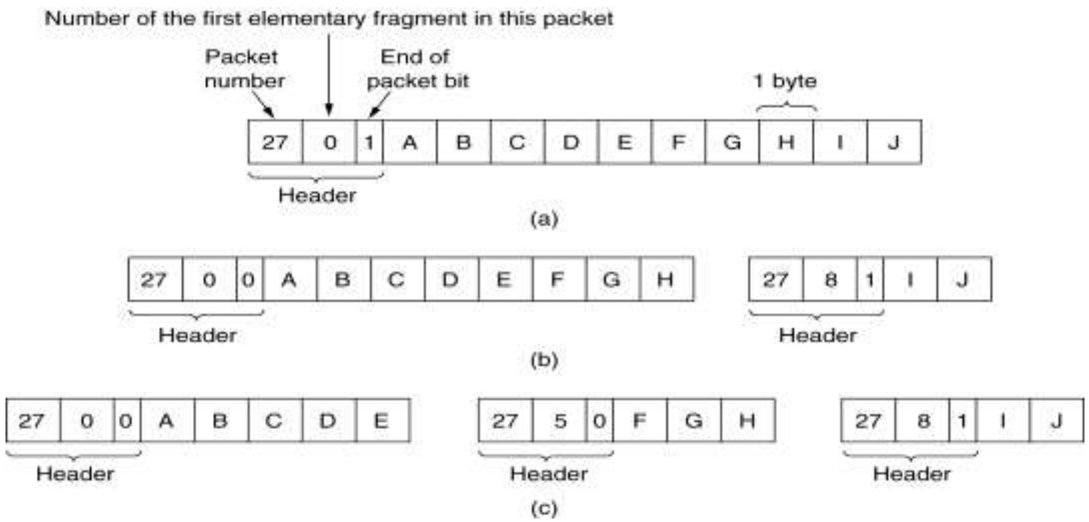
FRAGMENTATION

- (a) Transparent fragmentation. (b) Nontransparent fragmentation.



FRAGMENTATION (2)

- 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.



THE NETWORK LAYER IN THE INTERNET

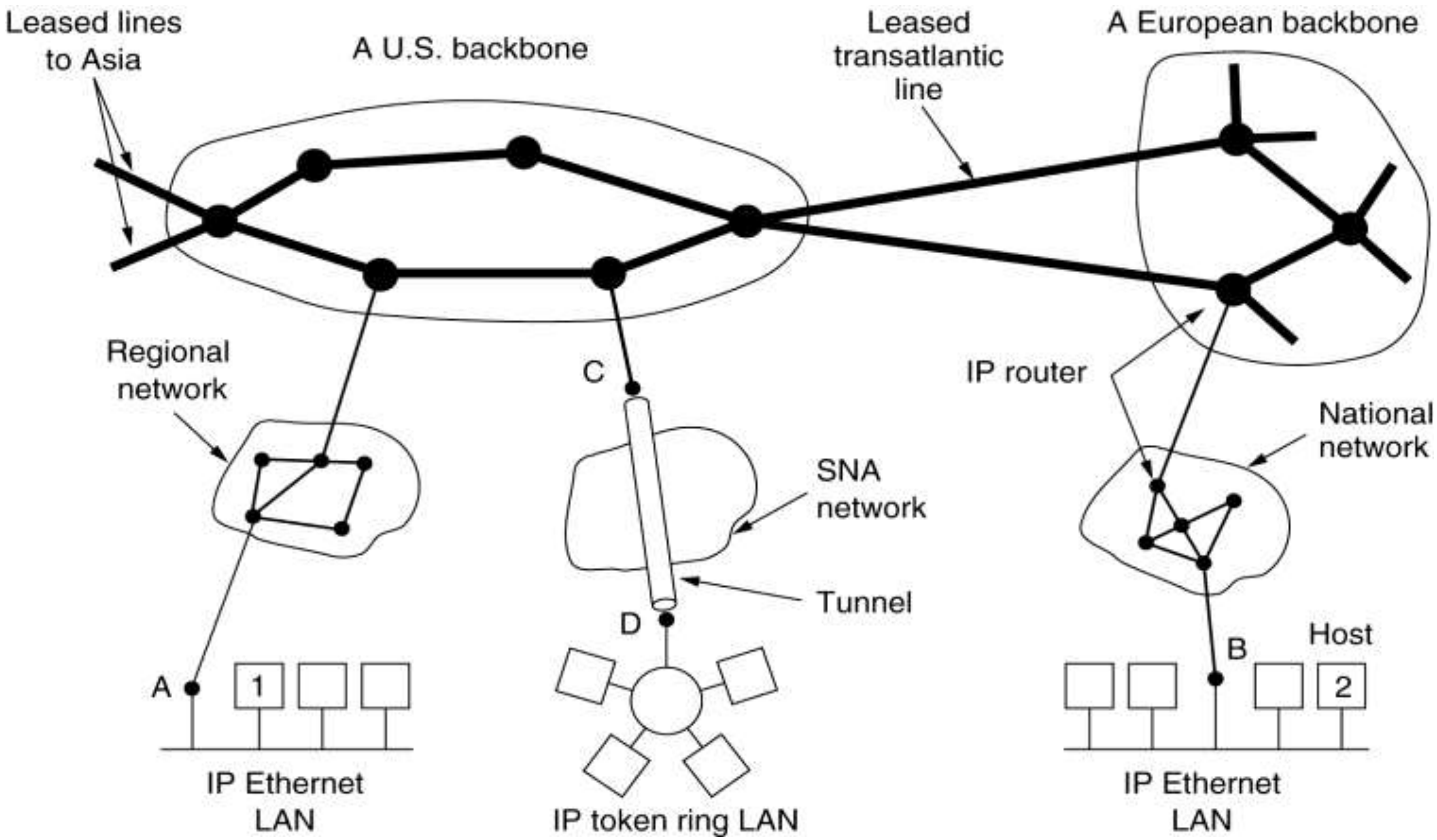
- The IP Protocol
- IP Addresses
- Internet Control Protocols
- OSPF – The Interior Gateway Routing Protocol
- BGP – The Exterior Gateway Routing Protocol
- Internet Multicasting
- Mobile IP
- IPv6

DESIGN PRINCIPLES FOR INTERNET

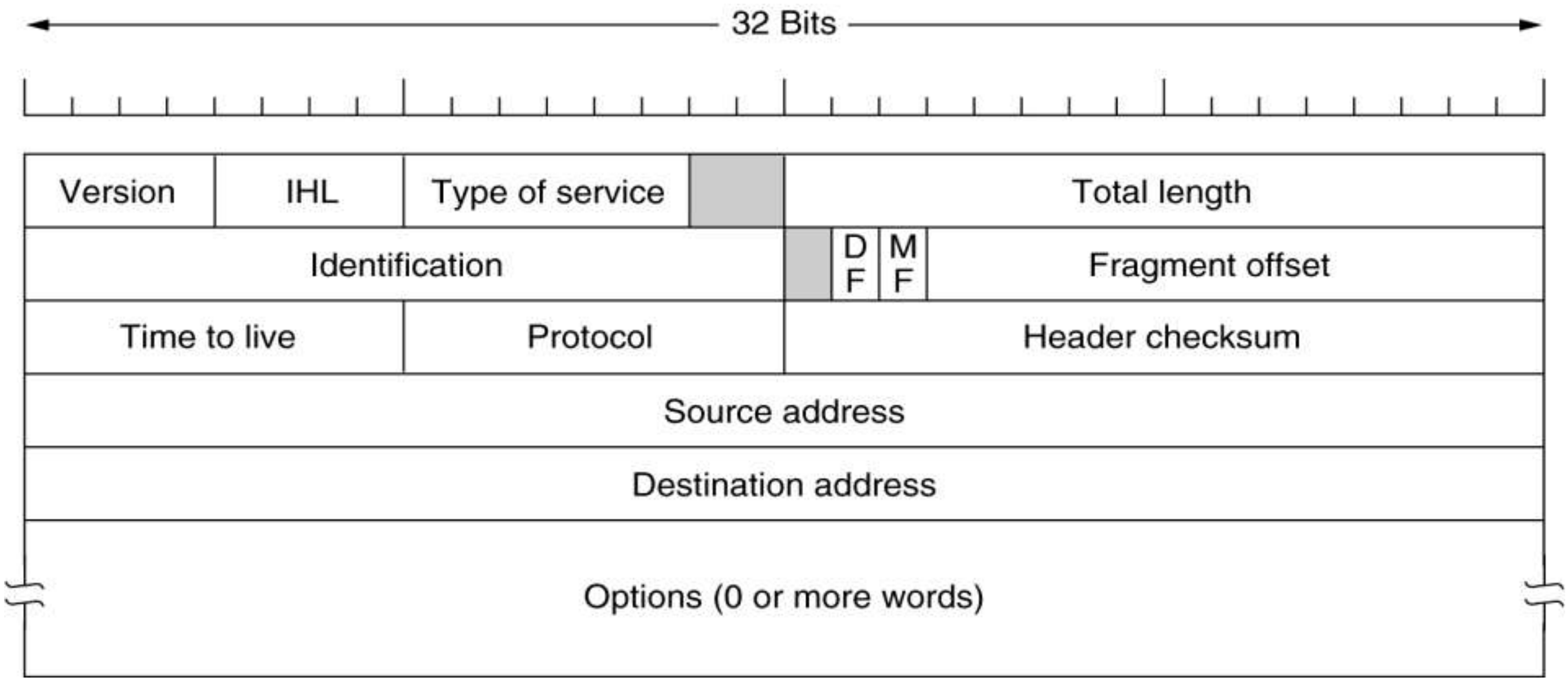
- Make sure it works.
- Keep it simple.
- Make clear choices.
- Exploit modularity.
- Expect heterogeneity.
- Avoid static options and parameters.
- Look for a good design; it need not be perfect.
- Be strict when sending and tolerant when receiving.
- Think about scalability.
- Consider performance and cost.

COLLECTION OF SUBNETWORKS

- The Internet is an interconnected collection of many networks.



THE IP PROTOCOL



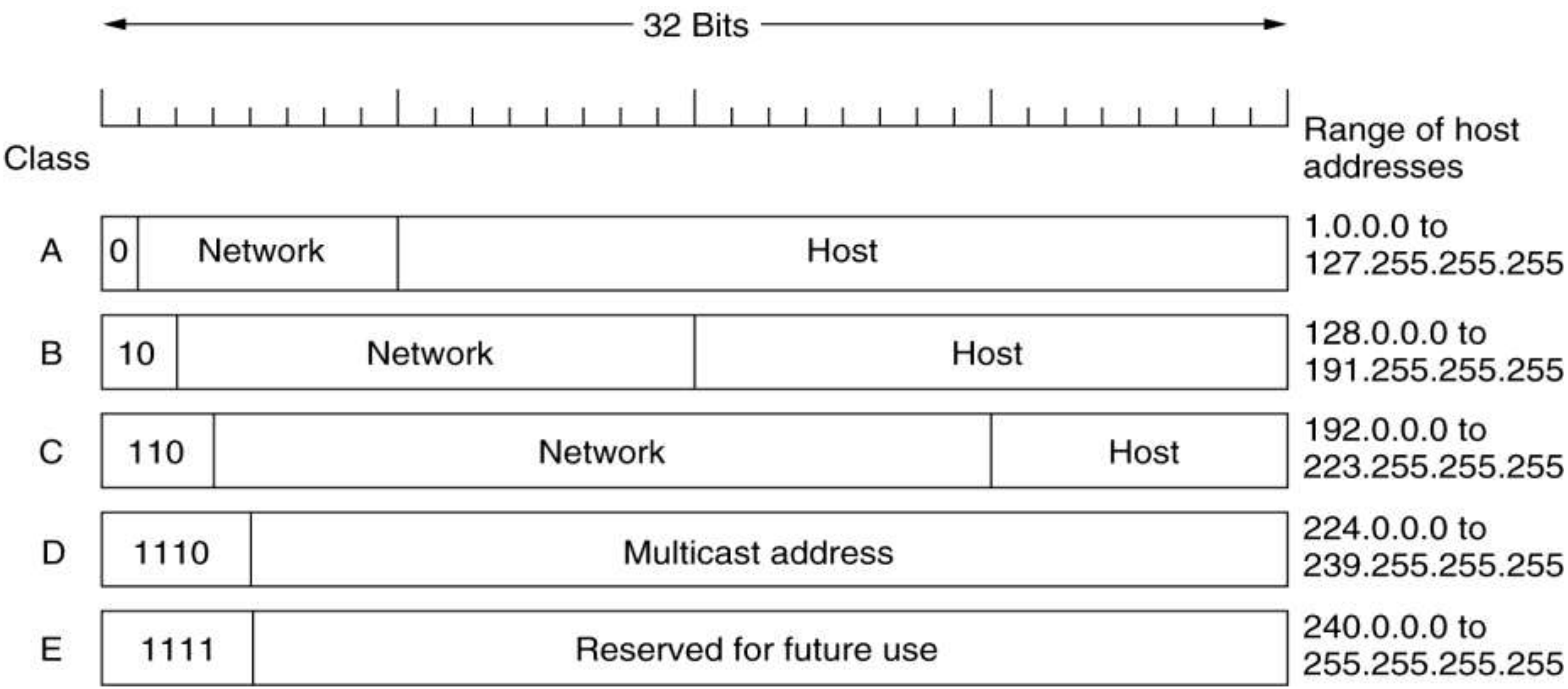
THE IP PROTOCOL (2)

- Some of the IP options.

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

IP ADDRESSES

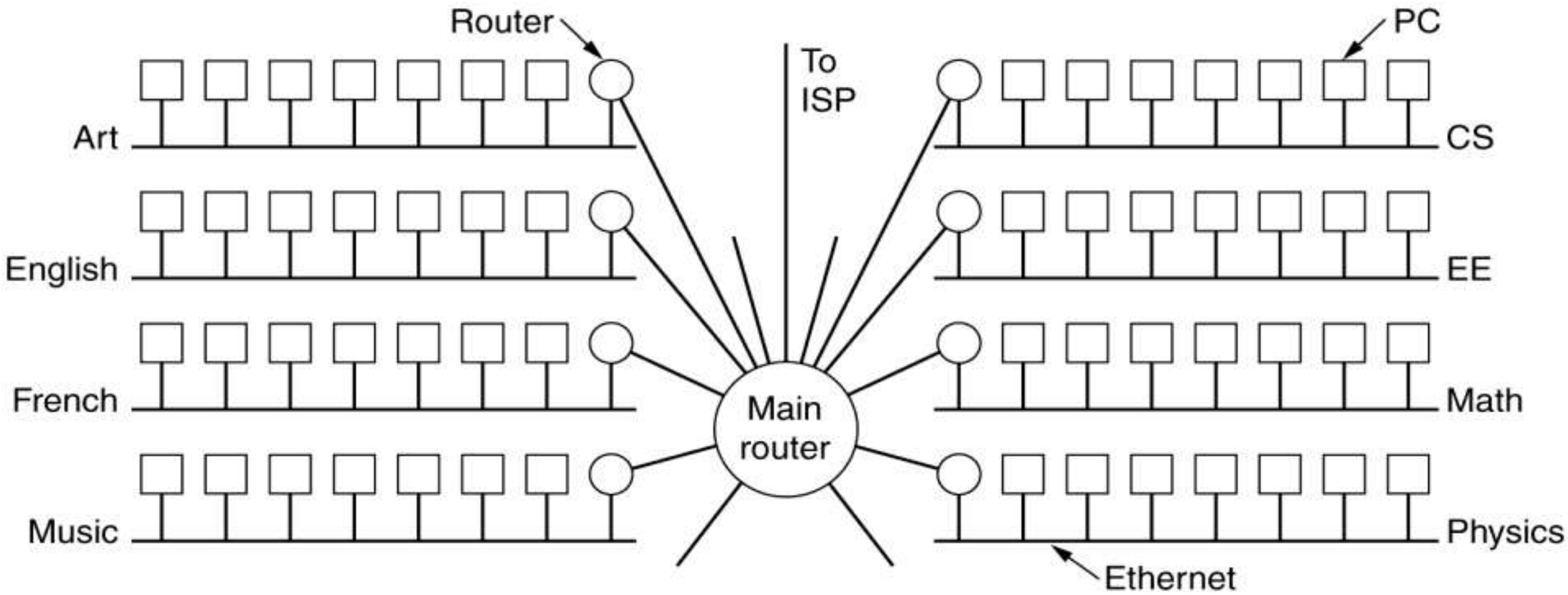
- IP address formats.



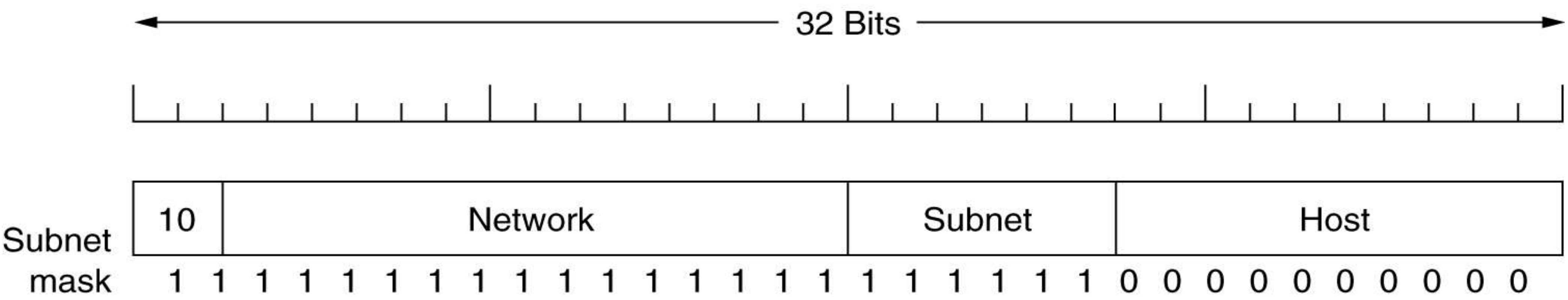
IP ADDRESSES (2)

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

SUBNETS


Router
PC
To
ISP
Art
CS
English
EE
French
Math
Main
router
Music
Physics
Ethernet

SUBNETS (2)


32 Bits
10
Network
Subnet
Host
Subnet
mask
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0

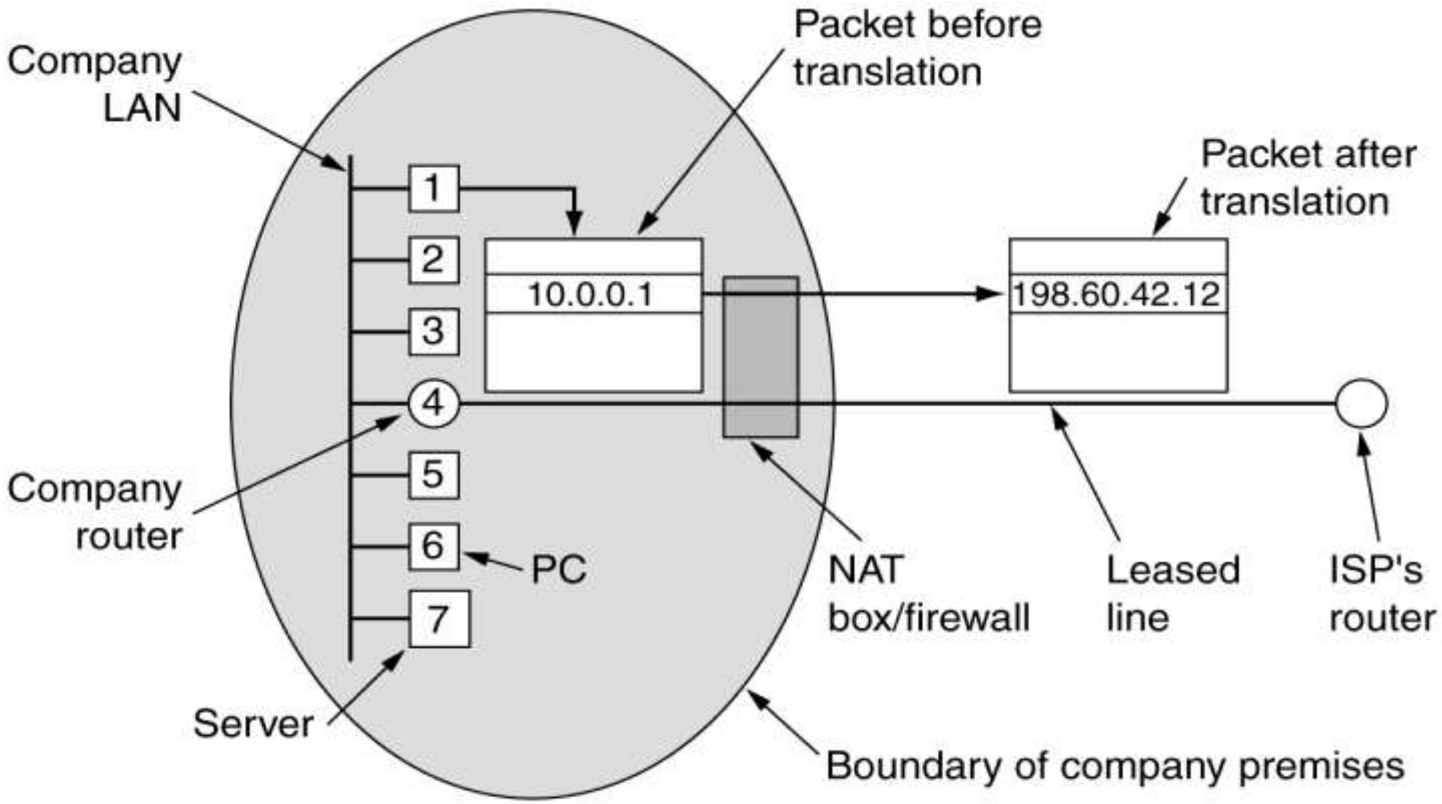
CDR – CLASSLESS INTERDOMAIN ROUTING

- A set of IP address assignments.

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

NAT – NETWORK ADDRESS TRANSLATION

- Placement and operation of a NAT box.



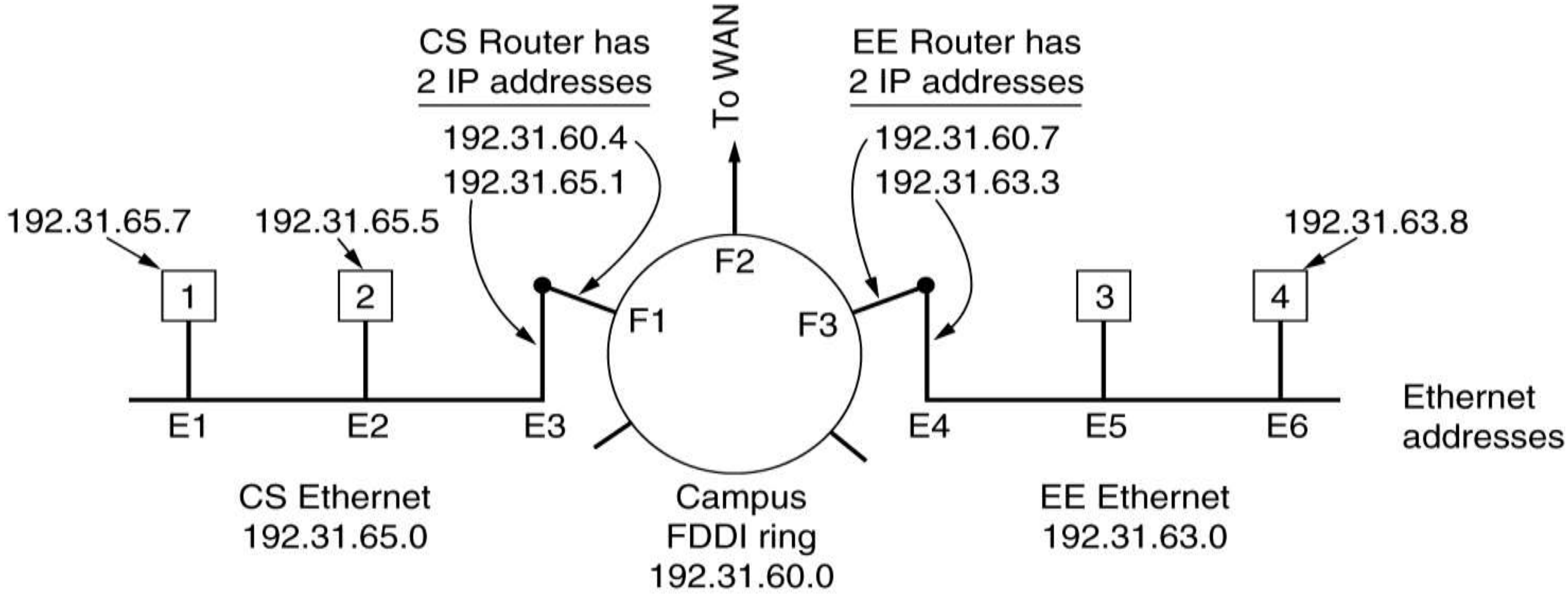
INTERNET CONTROL MESSAGE PROTOCOL

- The principal ICMP message types.

Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo 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

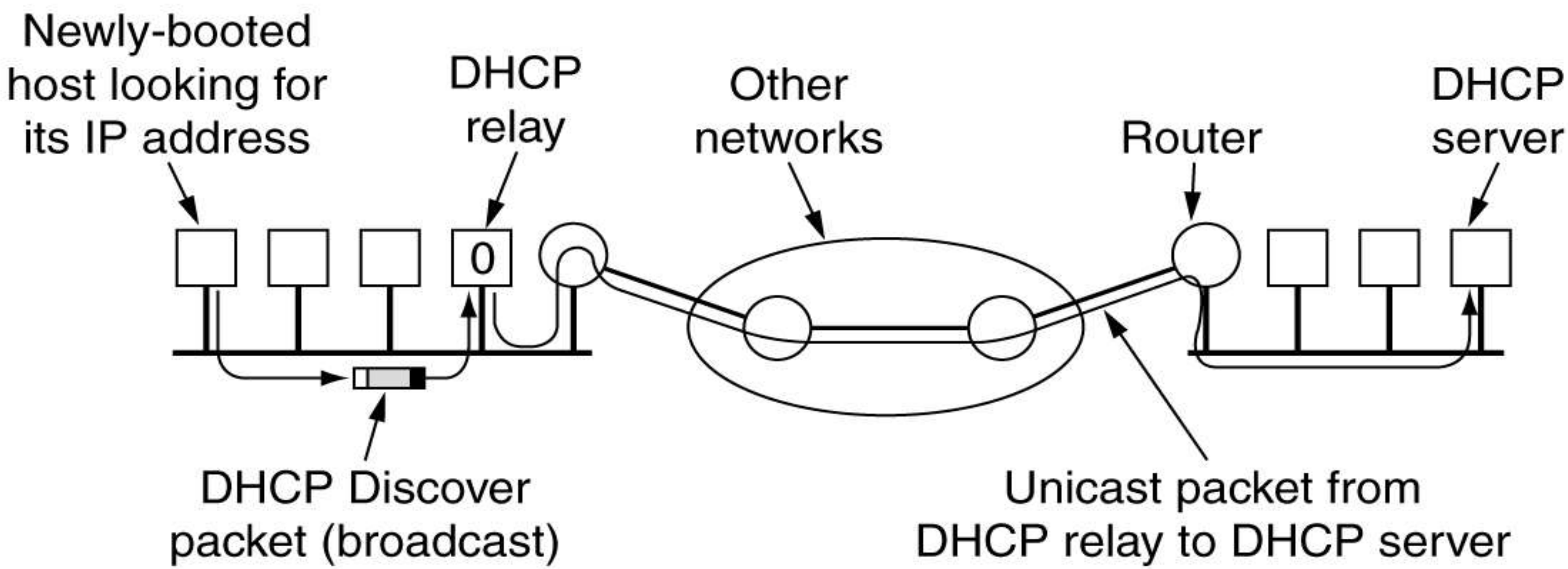
ARP– THE ADDRESS RESOLUTION PROTOCOL

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



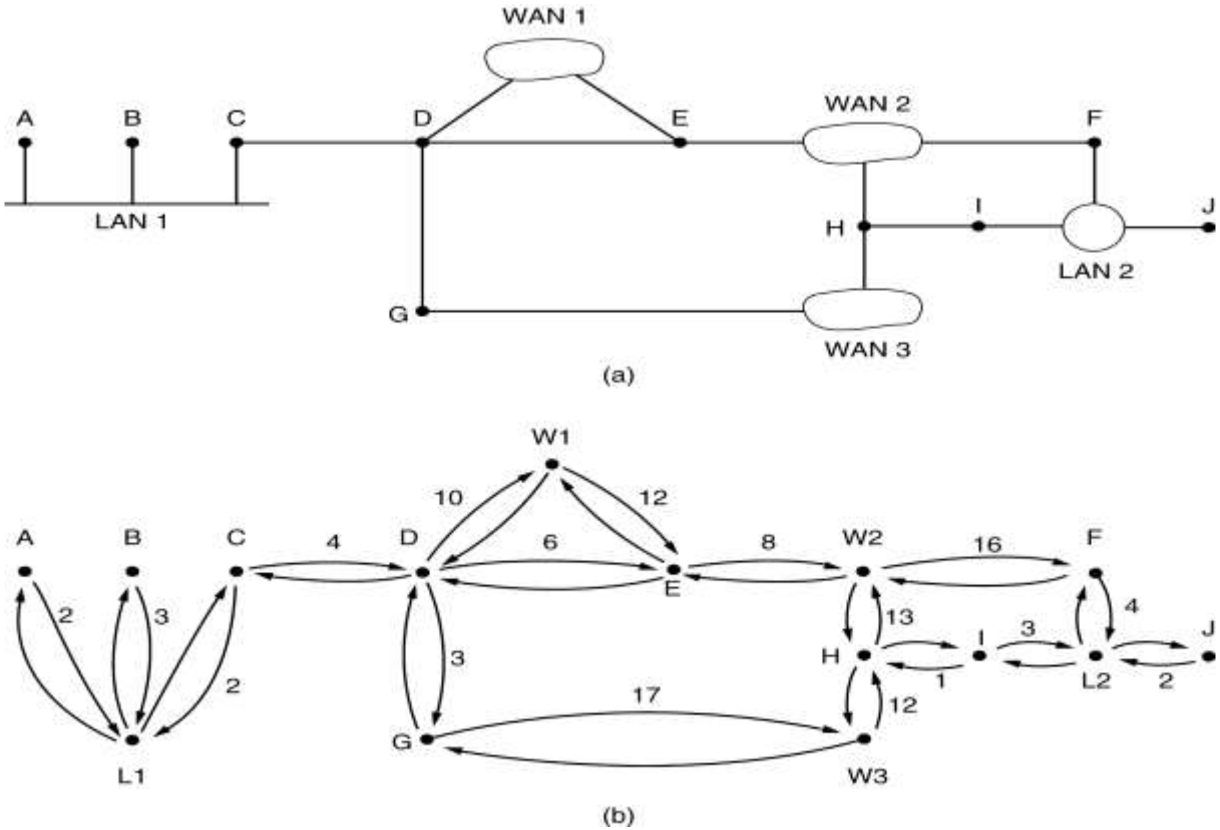
DYNAMIC HOST CONFIGURATION PROTOCOL

- Operation of DHCP.



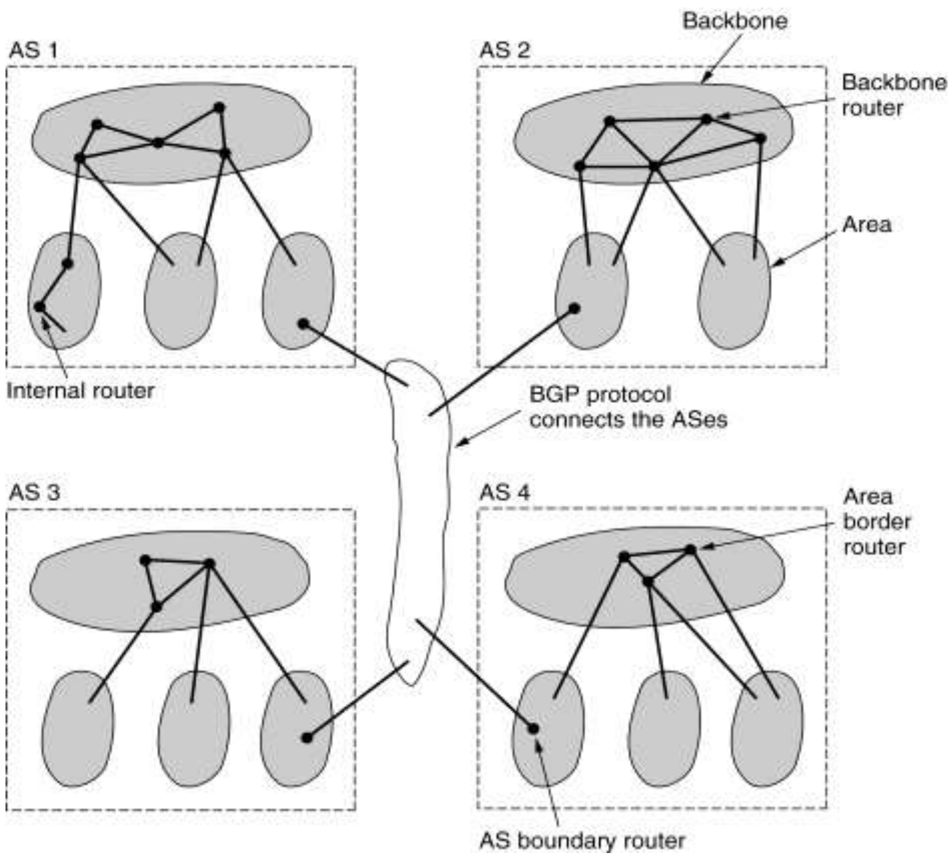
OSPF – THE INTERIOR GATEWAY ROUTING PROTOCOL

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



OSPF (2)

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



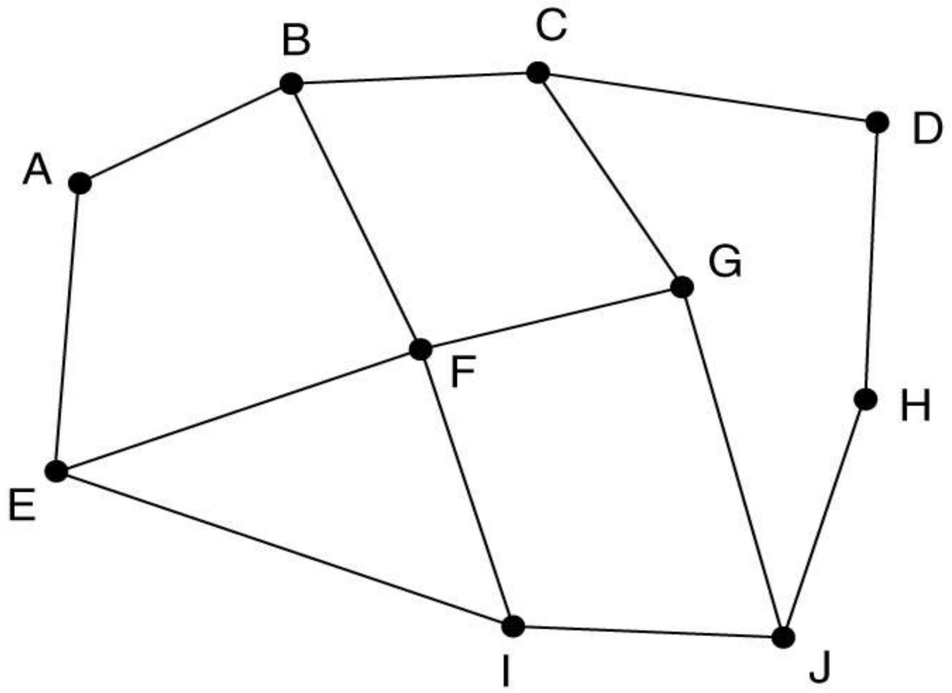
OSPF (3)

- The five types of OSPF messeges.

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

BGP – THE EXTERIOR GATEWAY ROUTING PROTOCOL

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



(a)

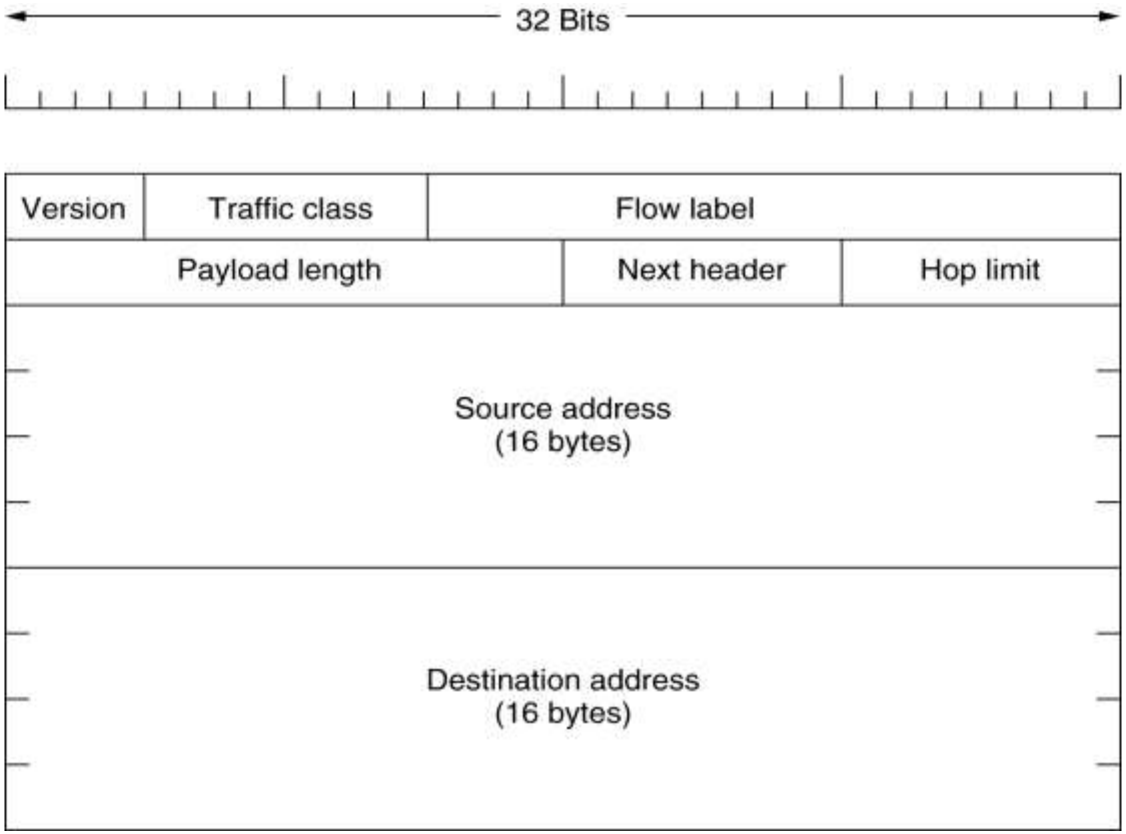
Information F receives
from its neighbors about D

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

(b)

THE MAIN IPV6 HEADER

- The IPv6 fixed header (required).



EXTENSION HEADERS

- IPv6 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

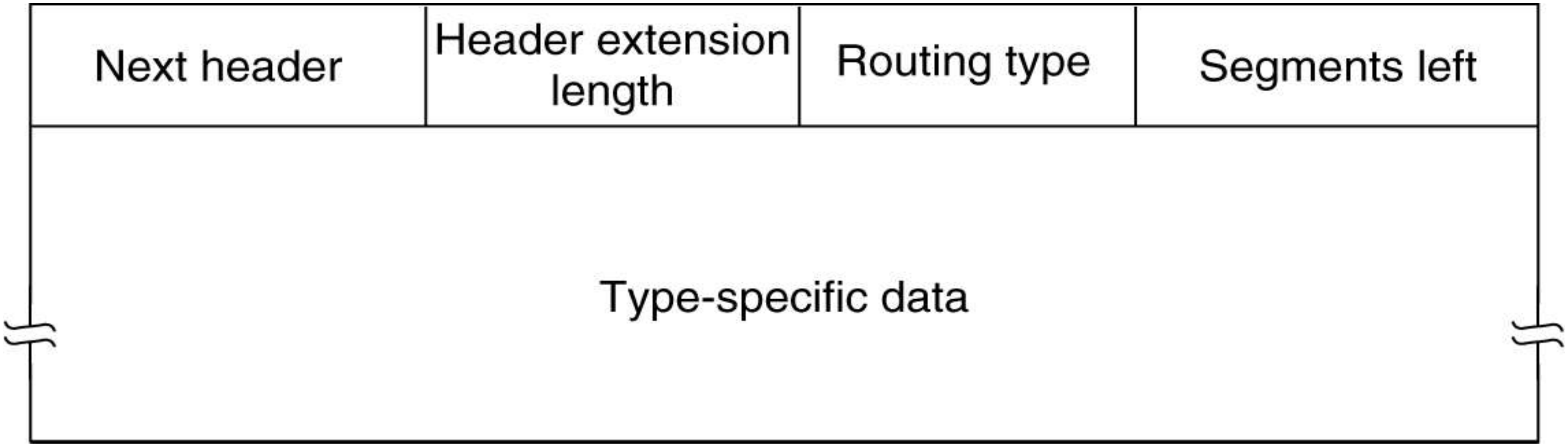
EXTENSION HEADERS (2)

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

Next header	0	194	4
Jumbo payload length			

EXTENSION HEADERS (3)

- The extension header for routing.



Thank you

The content in this material are from the textbooks and reference books given in the syllabus.