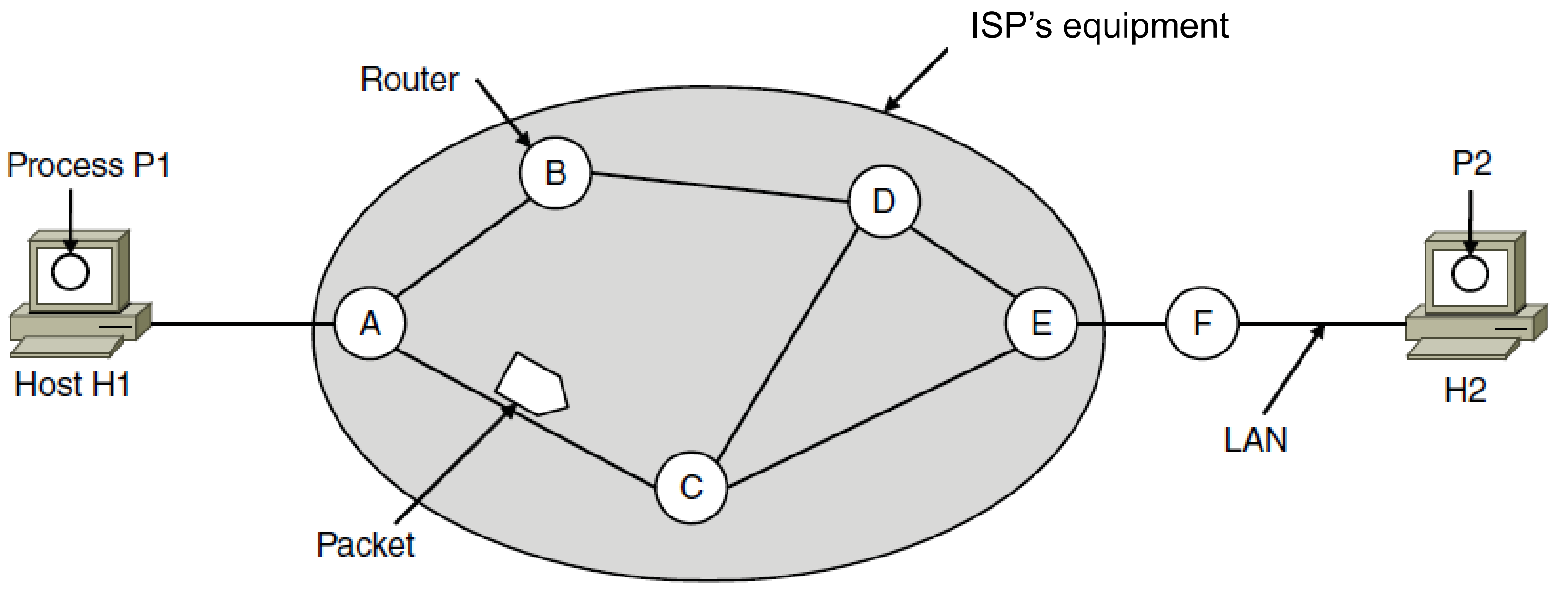
The Network Layer

Chapter 5

Network Layer Design Issues

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

Store-and-Forward Packet Switching

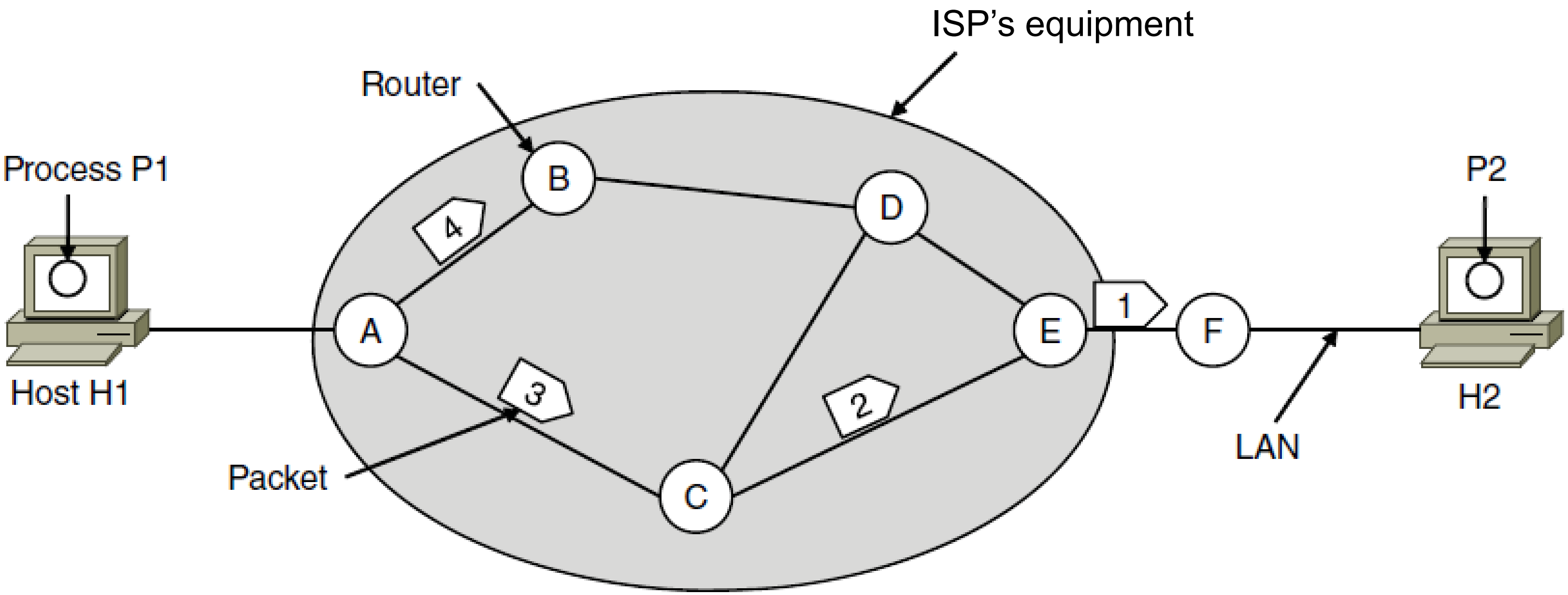


The environment of the network layer protocols.

Services Provided to the Transport Layer

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

Implementation of Connectionless Service



A's table (initially)

Dest.	Line
A	
B	B
C	C
D	B
E	C
F	C

A's table (later)

A	
B	B
C	C
D	B
E	D
F	D

C's Table

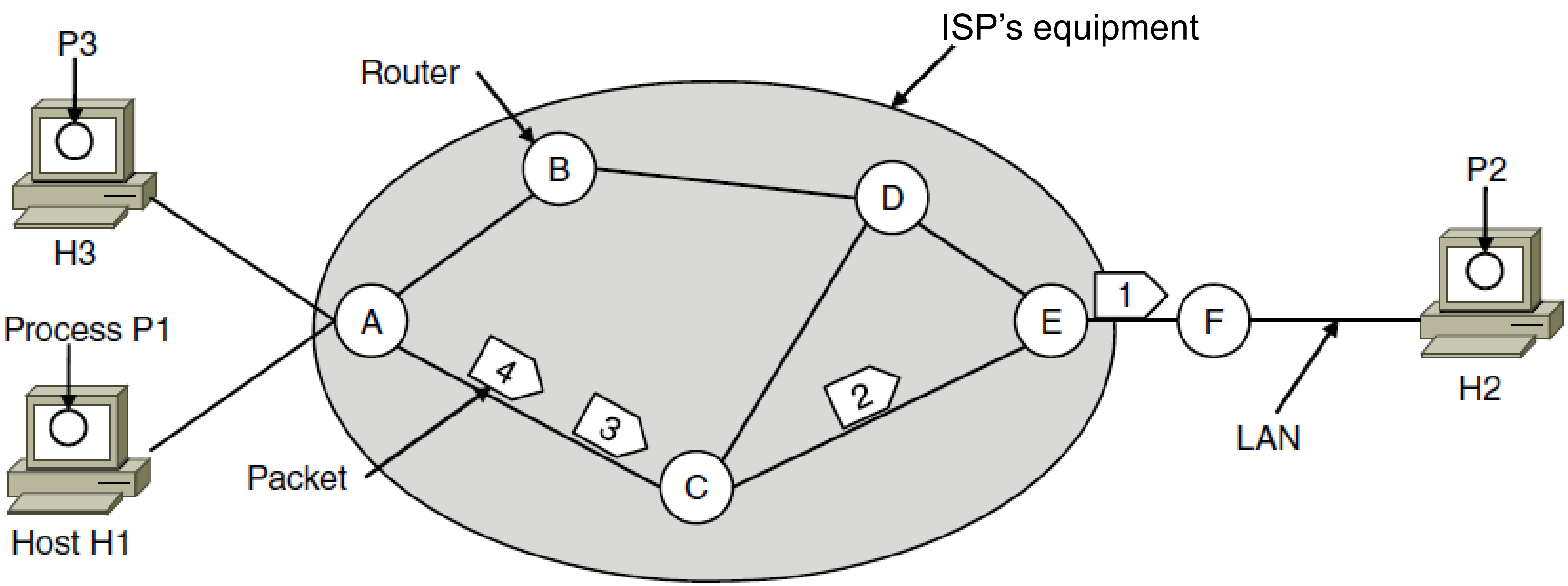
A	A
B	A
C	
D	E
E	E
F	E

E's Table

A	C
B	D
C	C
D	D
E	
F	F

Routing within a datagram network

Implementation of Connection-Oriented Service



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 network

Comparison of Virtual-Circuit and Datagram Networks

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

Comparison of datagram and virtual-circuit networks

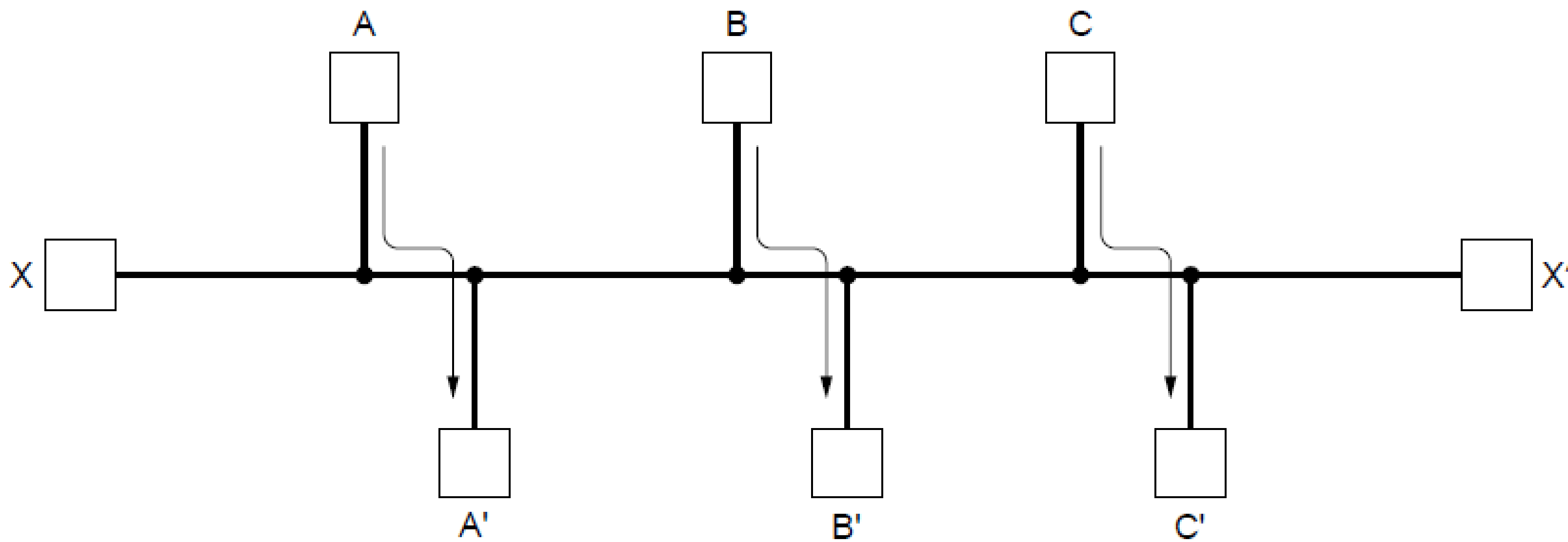
Routing Algorithms (1)

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

Routing Algorithms (2)

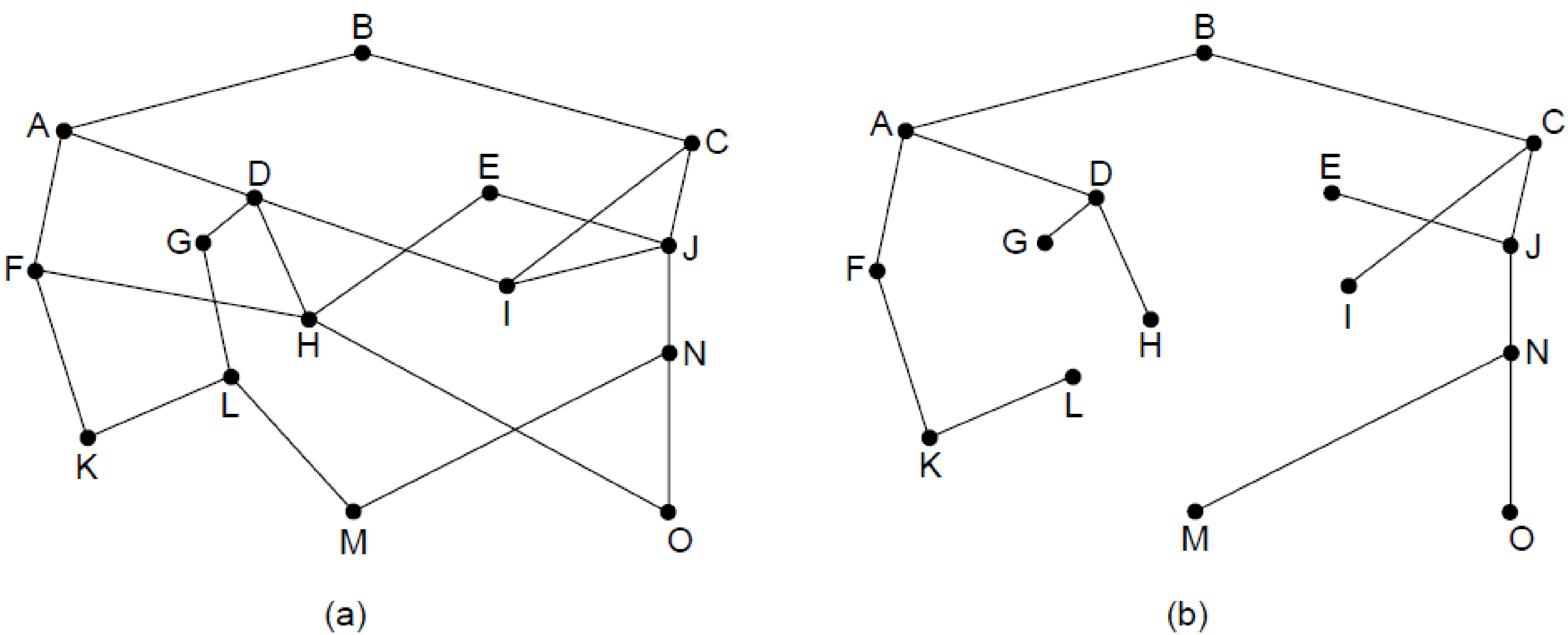
- Broadcast routing
- Multicast routing
- Anycast routing
- Routing for mobile hosts
- Routing in ad hoc networks

Fairness vs. Efficiency



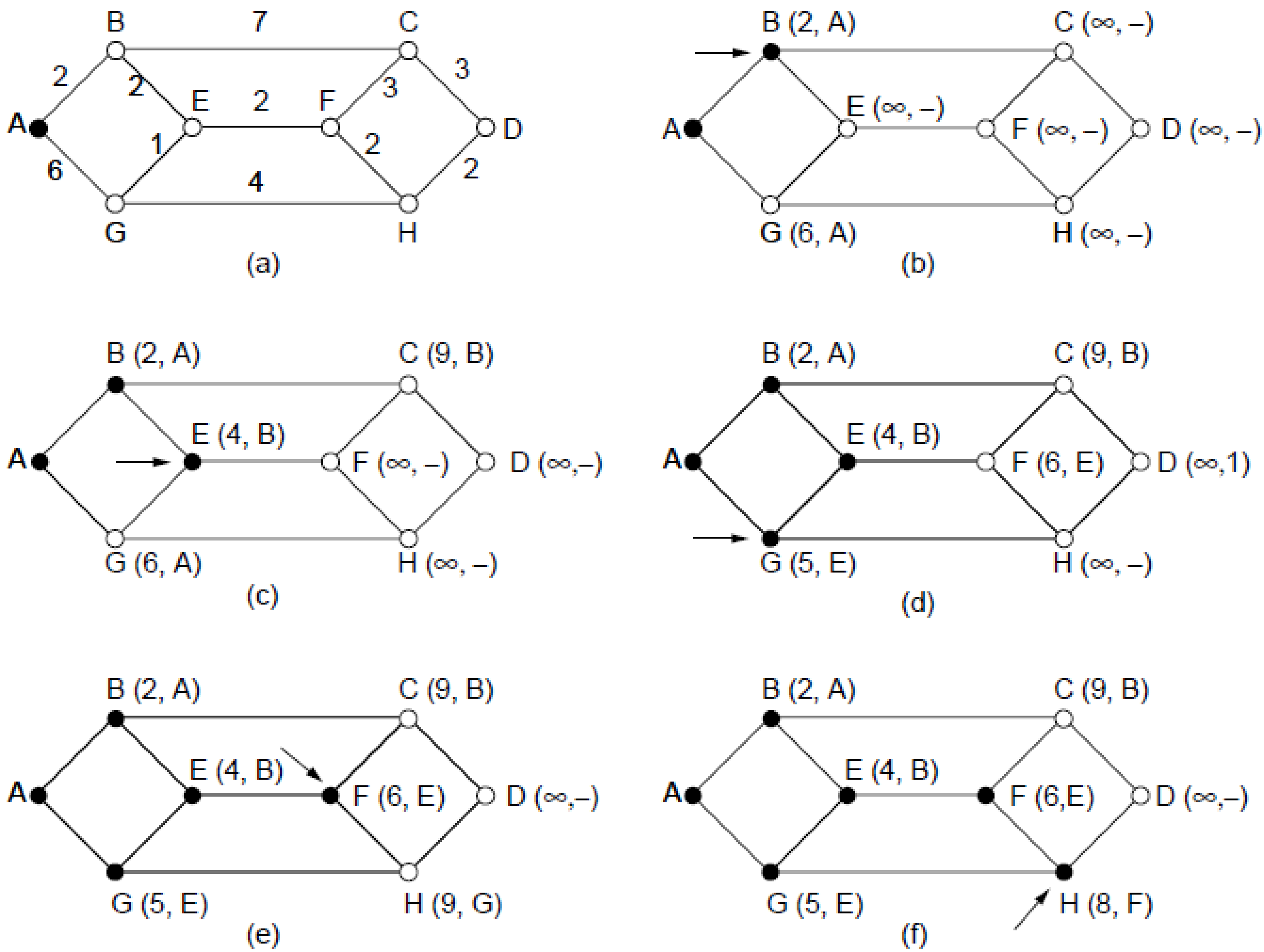
Network with a conflict between fairness and efficiency.

The Optimality Principle



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

Shortest Path Algorithm (1)



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

Shortest Path Algorithm (2)

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

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

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

    . . .
```

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

Shortest Path Algorithm (3)

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

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

Shortest Path Algorithm (4)

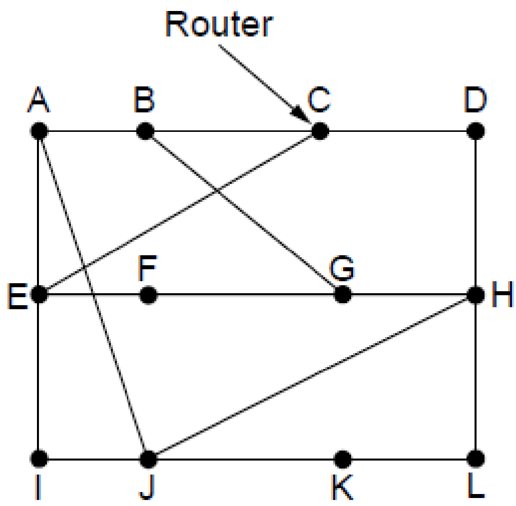
. . .

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

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

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

Distance Vector Routing



(a)

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

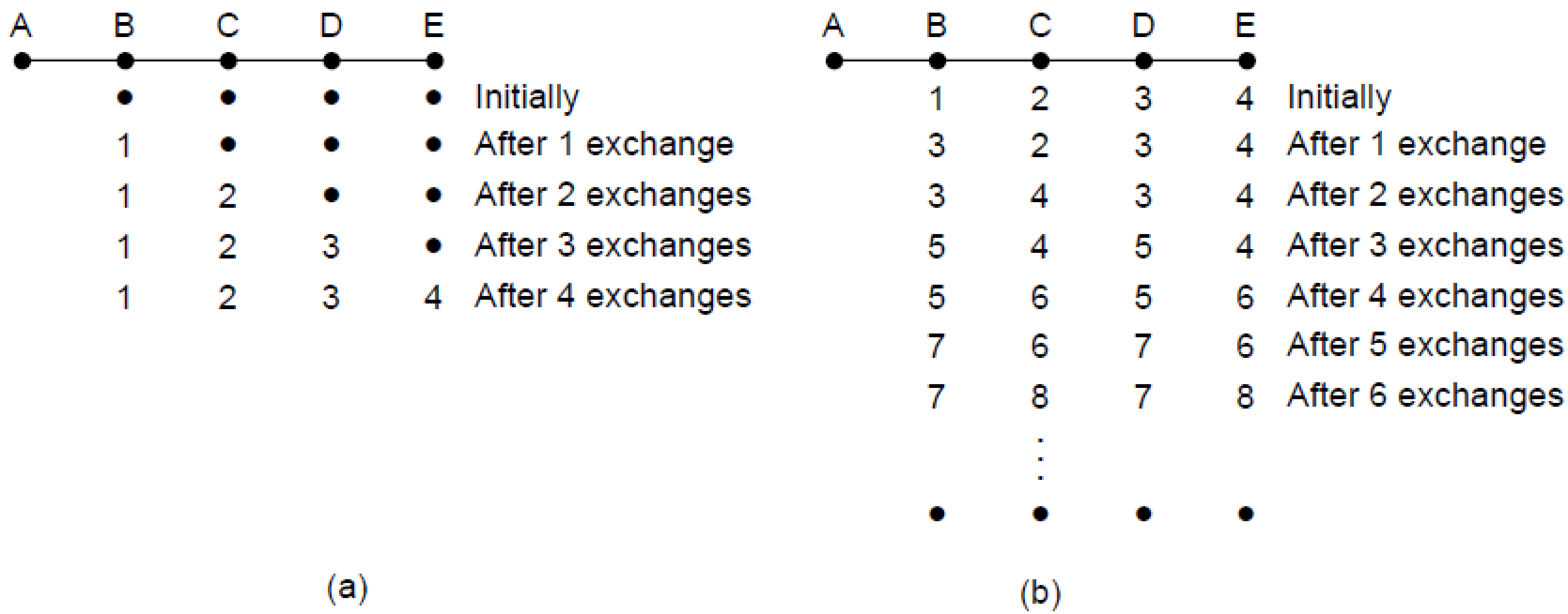
Vectors received from J's four neighbors

(b)

(a) A network.

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

The Count-to-Infinity Problem

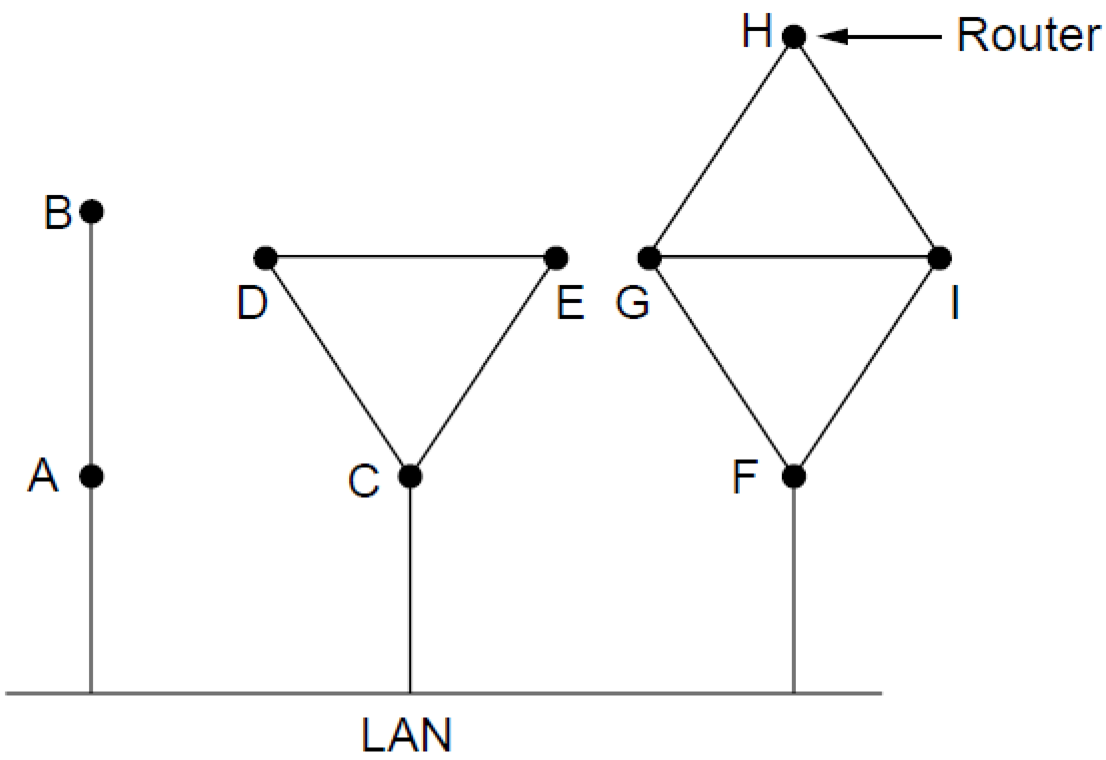


The count-to-infinity problem

Link State Routing

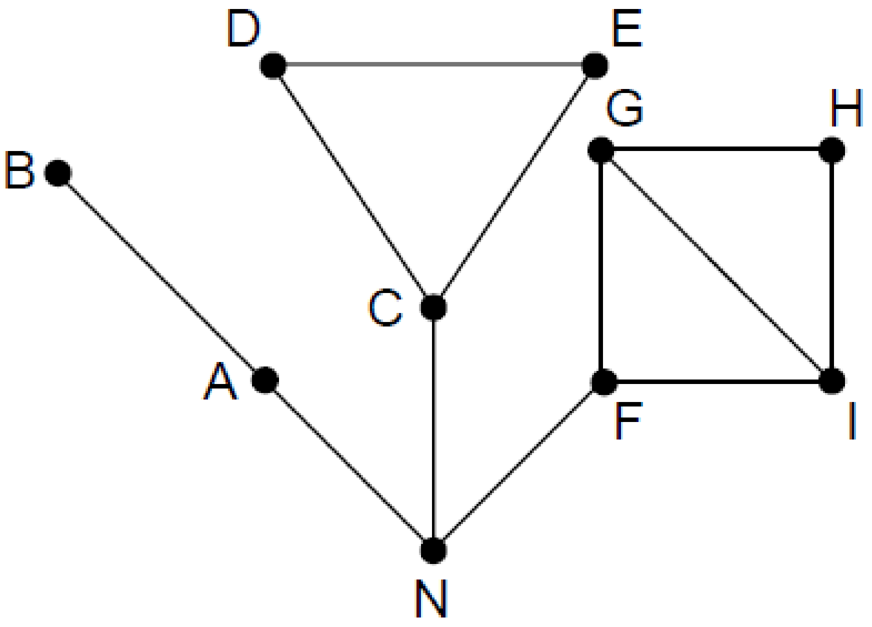
1. Discover neighbors, learn network addresses.
2. Set distance/cost metric to each neighbor.
3. Construct packet telling all learned.
4. Send packet to, receive packets from other routers.
5. Compute shortest path to every other router.

Learning about the Neighbors (1)



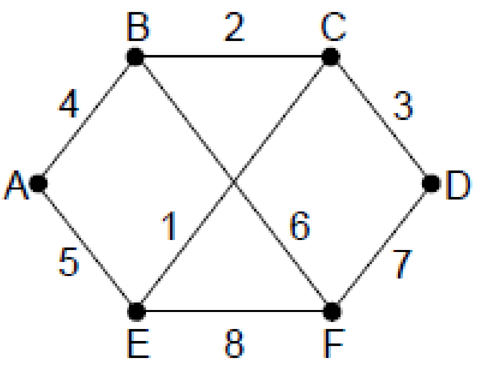
Nine routers and a broadcast LAN.

Learning about the Neighbors (2)



A graph model of previous slide.

Building Link State Packets



(a)

Link State Packets

A		B		C		D		E		F	
Seq.		Seq.		Seq.		Seq.		Seq.		Seq.	
Age		Age		Age		Age		Age		Age	
B	4	A	4	B	2	C	3	A	5	B	6
E	5	C	2	D	3	F	7	C	1	D	7
		F	6	E	1			F	8	E	8

(b)

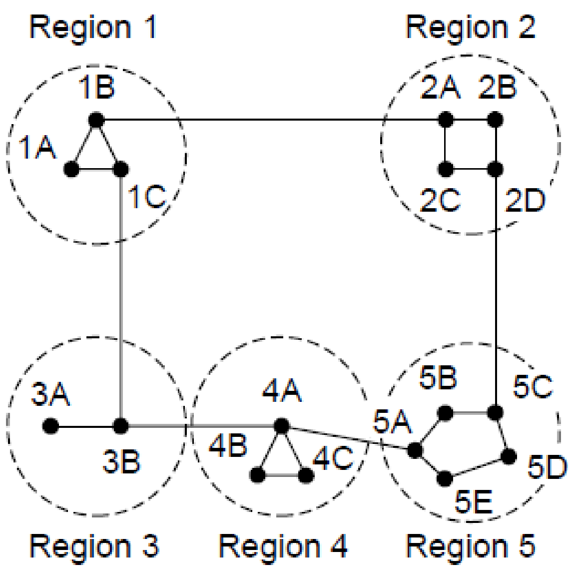
(a) A network. (b) The link state packets for this network.

Distributing the Link State Packets

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

The packet buffer for router *B* in previous slide

Hierarchical Routing



(a)

Full table for 1A

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

(b)

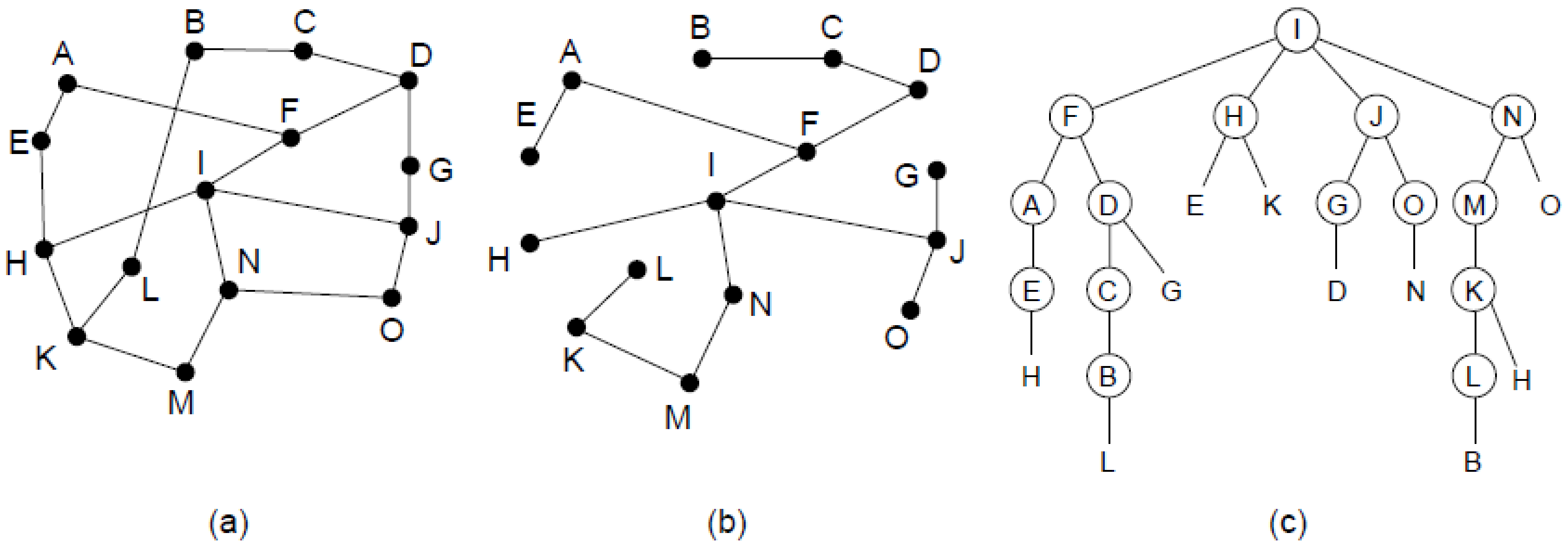
Hierarchical table for 1A

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

(c)

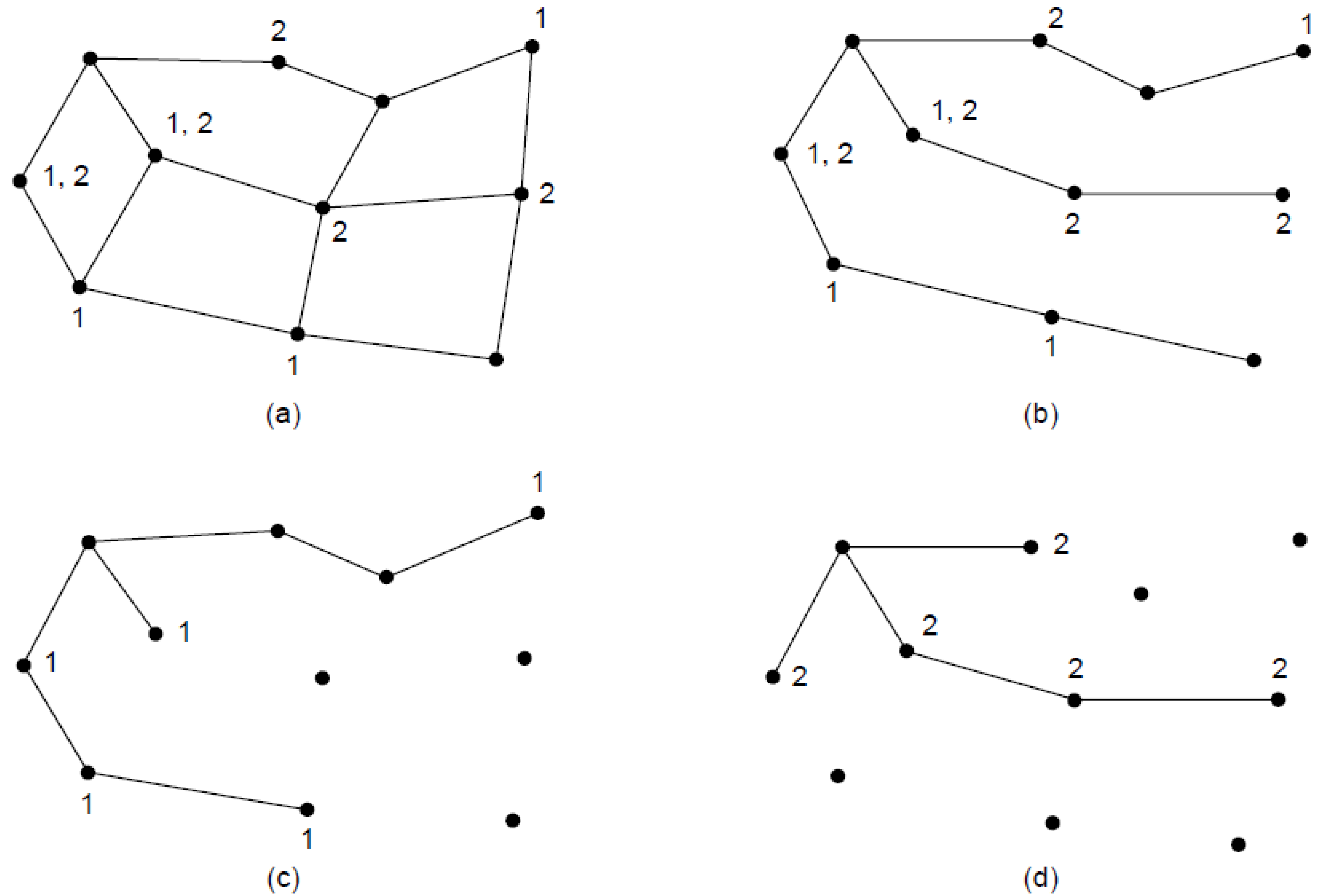
Hierarchical routing.

Broadcast Routing



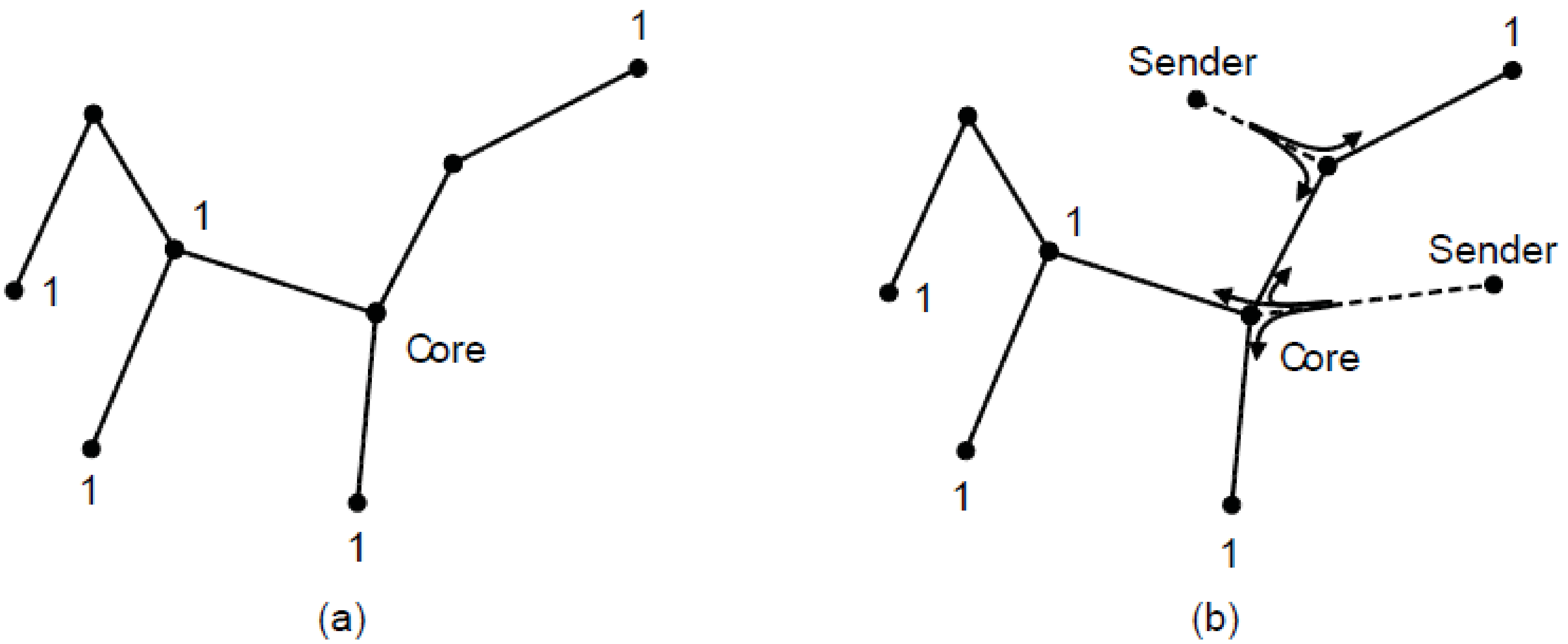
Reverse path forwarding. (a) A network. (b) A sink tree.
(c) The tree built by reverse path forwarding.

Multicast Routing (1)



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

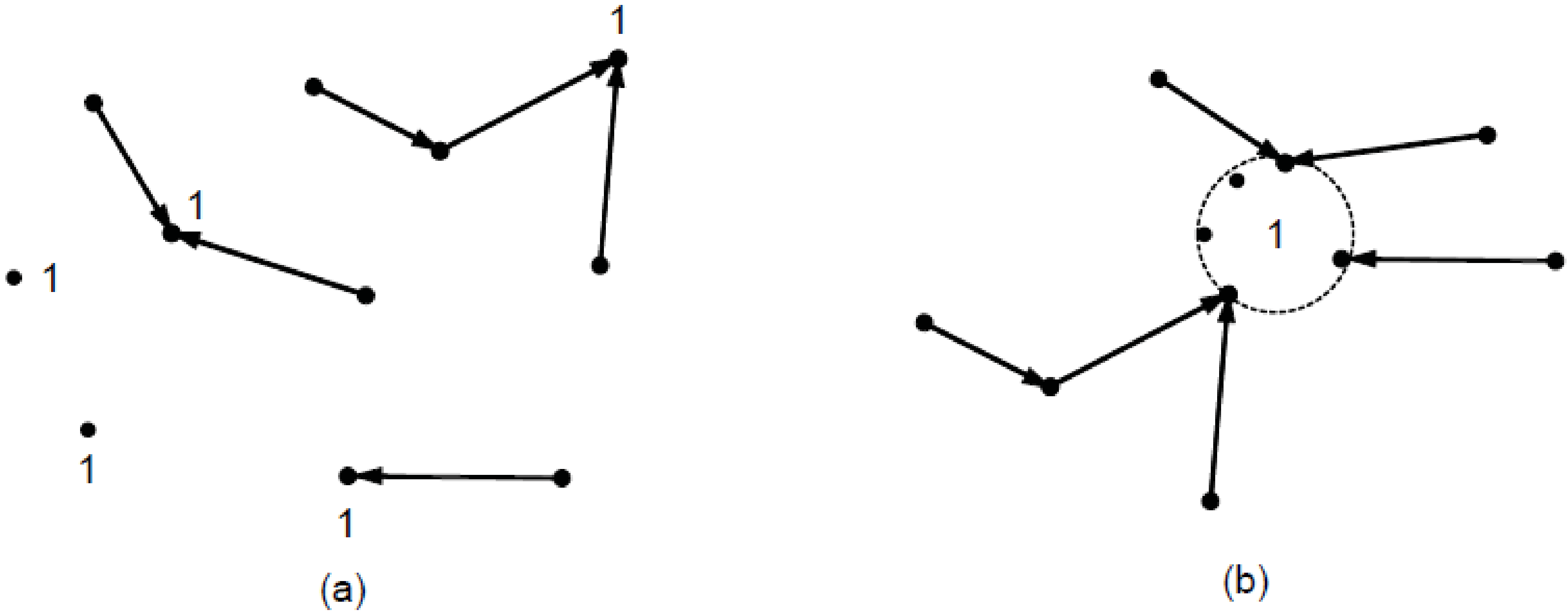
Multicast Routing (2)



(a) Core-based tree for group 1.

(b) Sending to group 1.

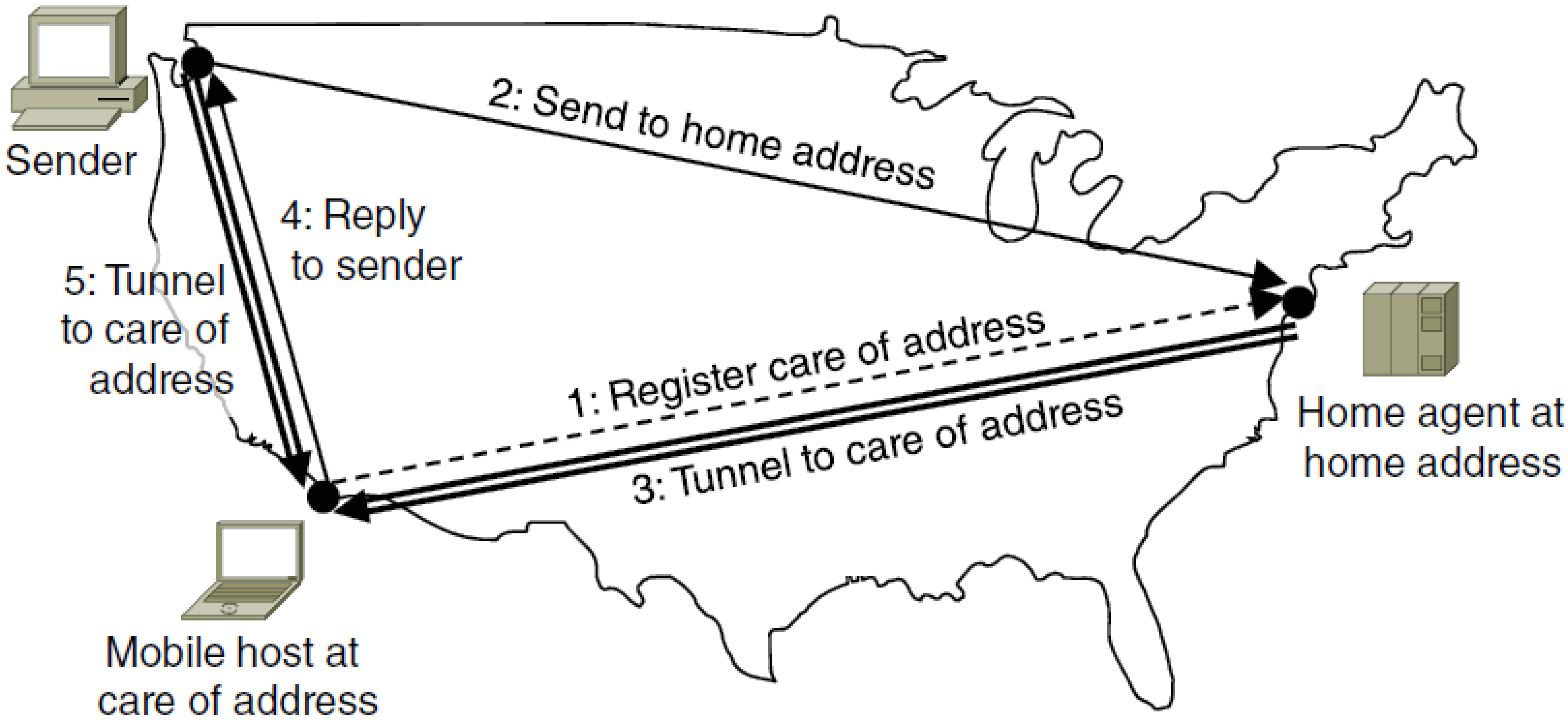
Anycast Routing



(a) Anycast routes to group 1.

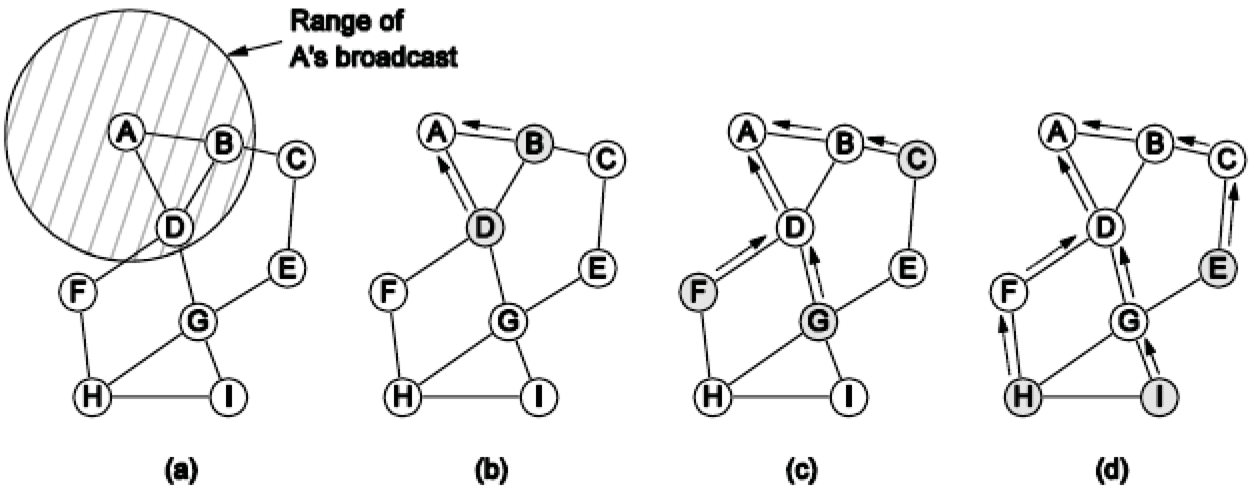
(b) Topology seen by the routing protocol.

Routing for Mobile Hosts



Packet routing for mobile hosts

Routing in Ad Hoc Networks



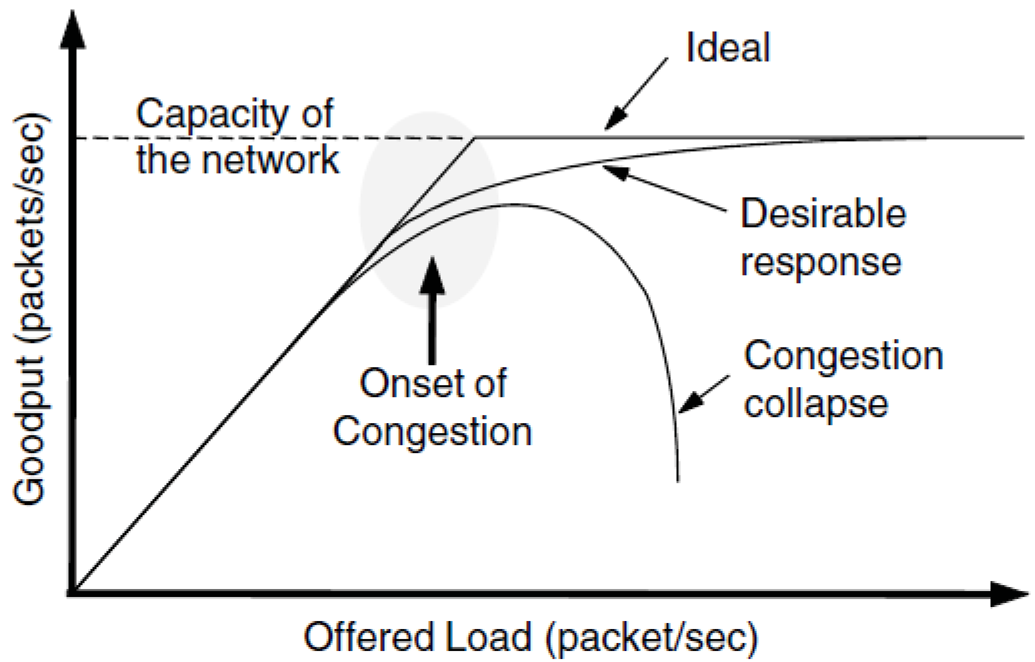
(a) Range of A's broadcast.
(b) After B and D receive it.
(c) After C, F, and G receive it.
(d) After E, H, and I receive it.

The shaded nodes are new recipients. The dashed lines show possible reverse routes. The solid lines show the discovered route.

Congestion Control Algorithms (1)

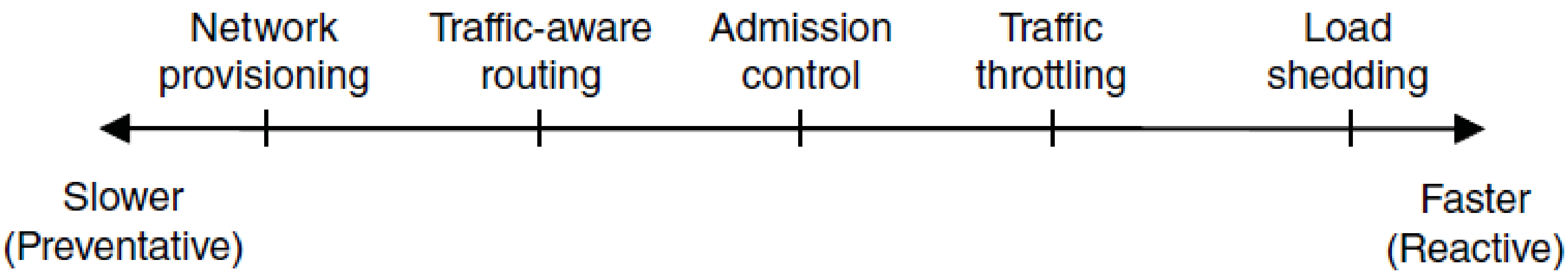
- Approaches to congestion control
- Traffic-aware routing
- Admission control
- Traffic throttling
- Load shedding

Congestion Control Algorithms (2)



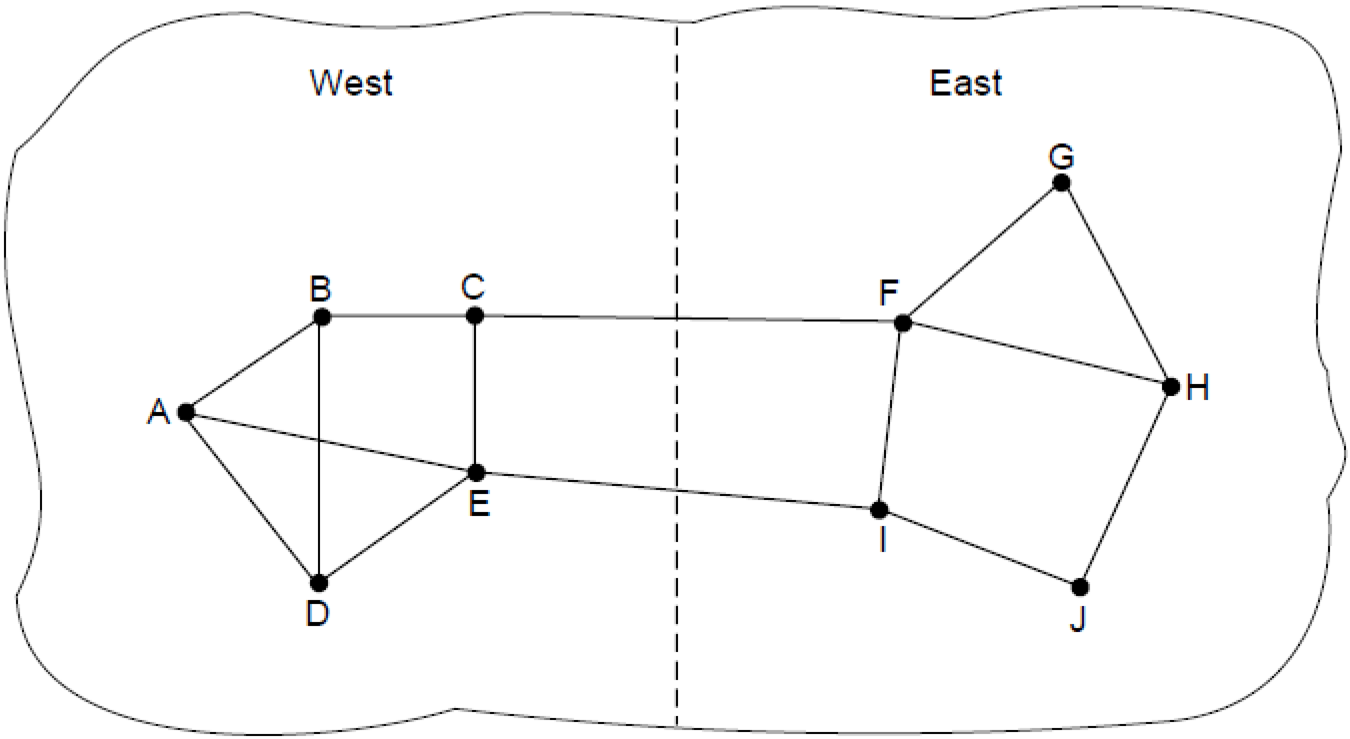
When too much traffic is offered, congestion sets in and performance degrades sharply.

Approaches to Congestion Control



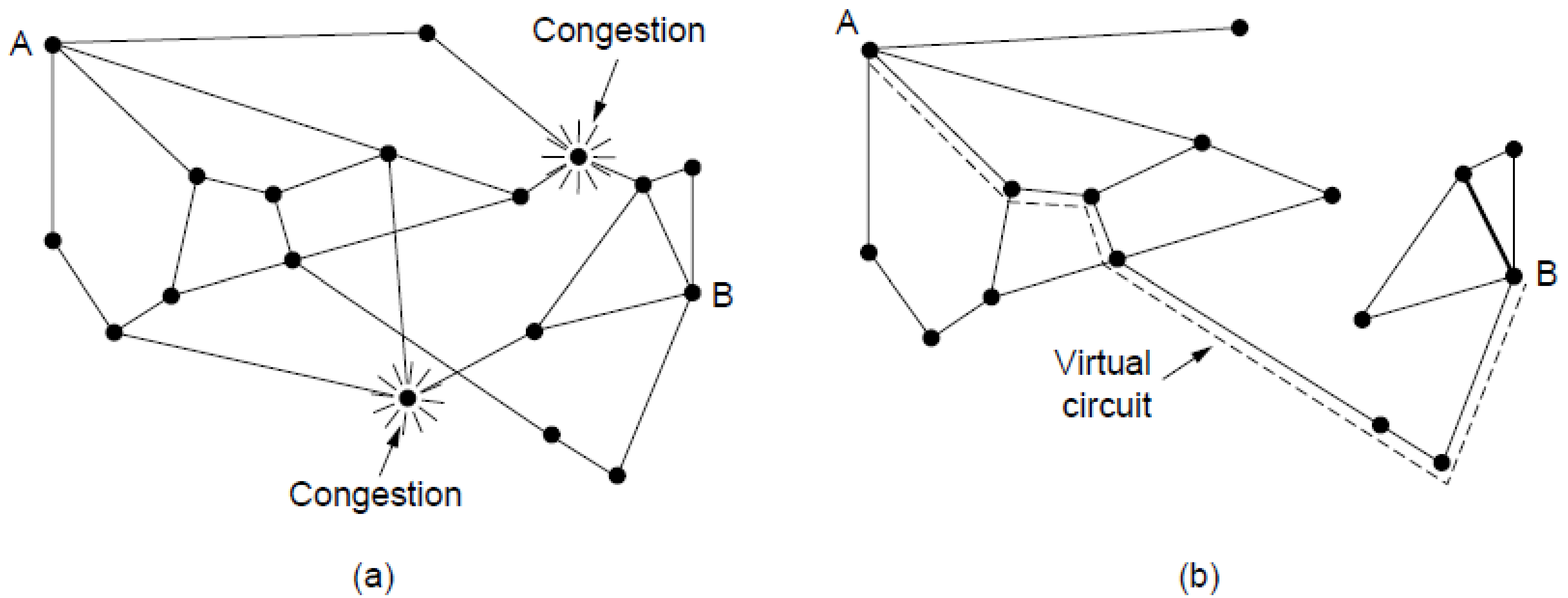
Timescales of approaches to congestion control

Traffic-Aware Routing



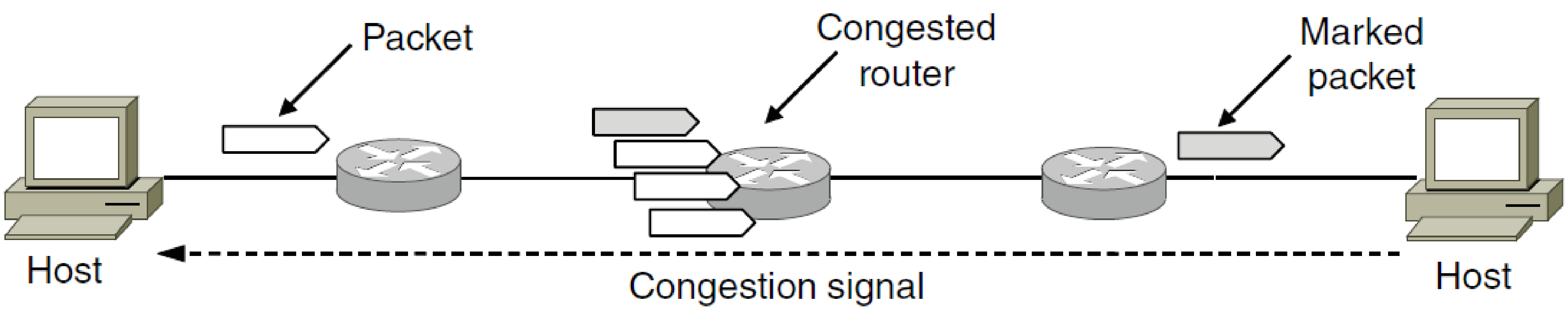
A network in which the East and West parts are connected by two links.

Traffic Throttling (1)



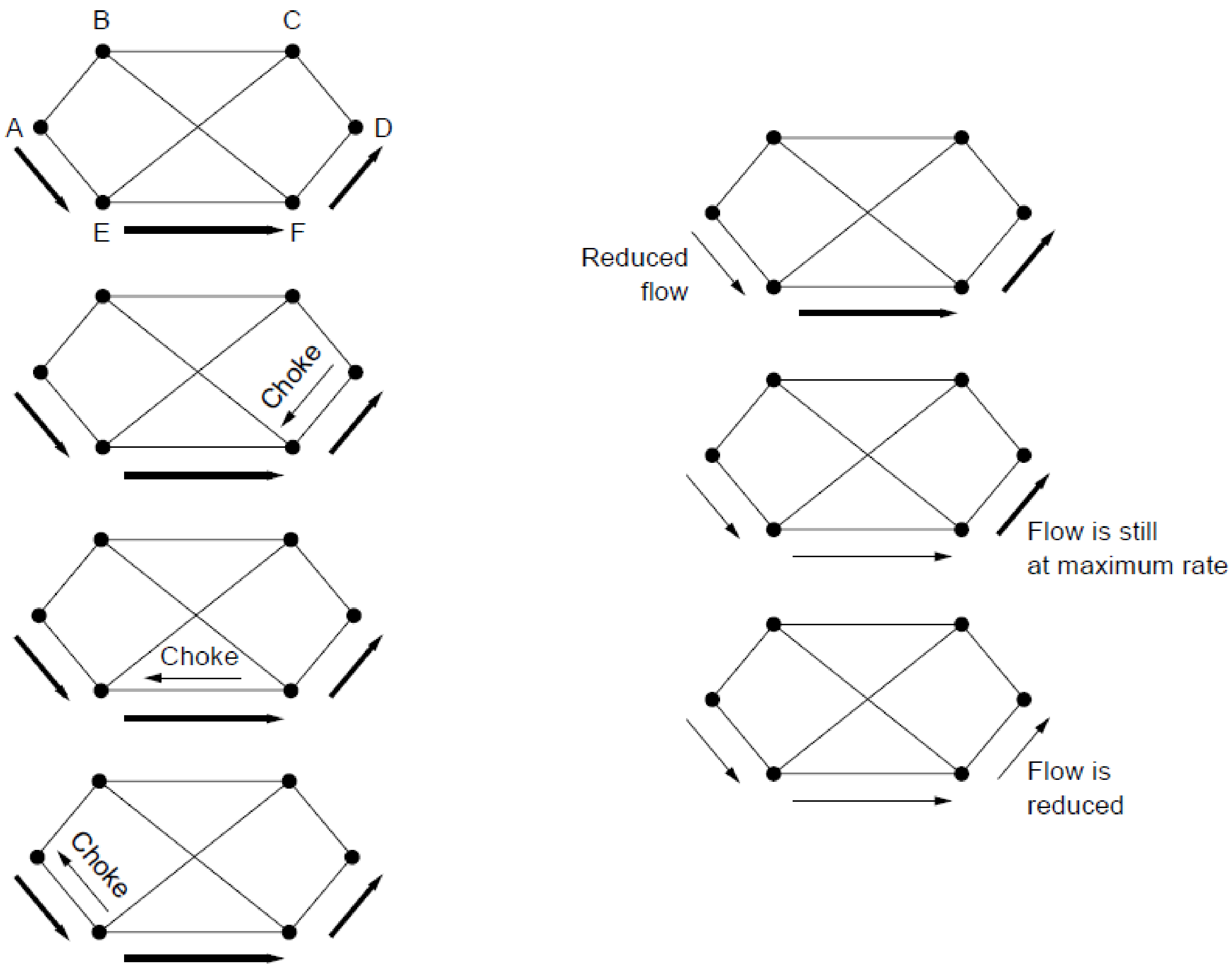
(a) A congested network. (b) The portion of the network that is not congested. A virtual circuit from A to B is also shown.

Traffic Throttling (2)



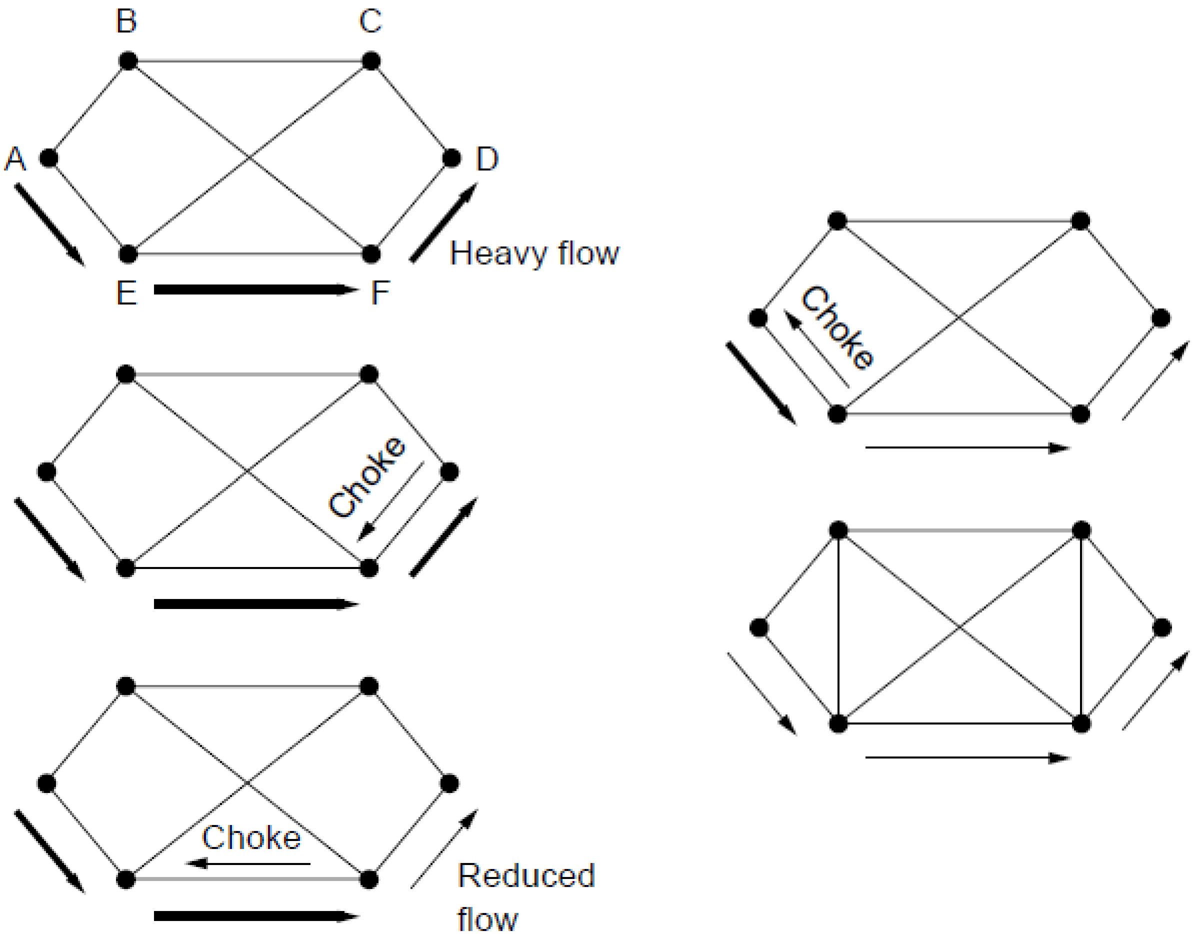
Explicit congestion notification

Load Shedding (1)



A choke packet that affects only the source..

Load Shedding (2)



A choke packet that affects each hop it passes through.

Quality of Service

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

Application Requirements (1)

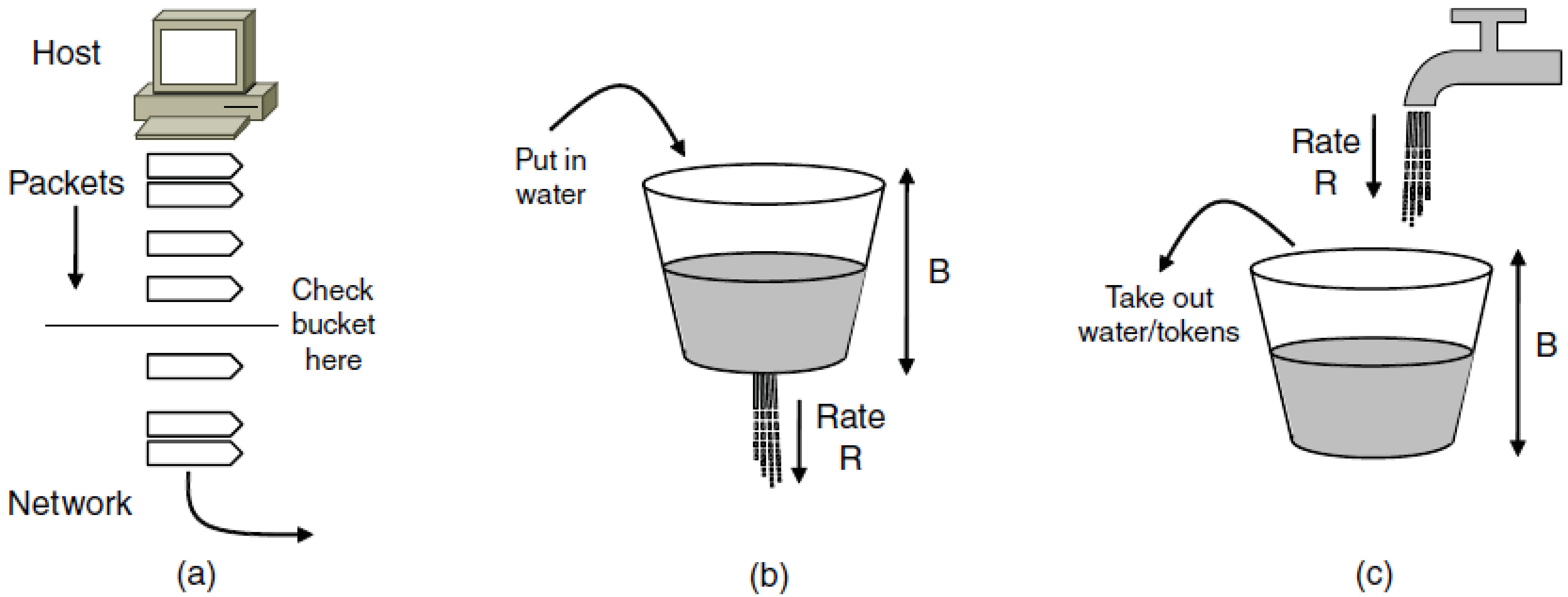
Application	Bandwidth	Delay	Jitter	Loss
Email	Low	Low	Low	Medium
File sharing	High	Low	Low	Medium
Web access	Medium	Medium	Low	Medium
Remote login	Low	Medium	Medium	Medium
Audio on demand	Low	Low	High	Low
Video on demand	High	Low	High	Low
Telephony	Low	High	High	Low
Videoconferencing	High	High	High	Low

How stringent the quality-of-service requirements are.

Categories of QoS and Examples

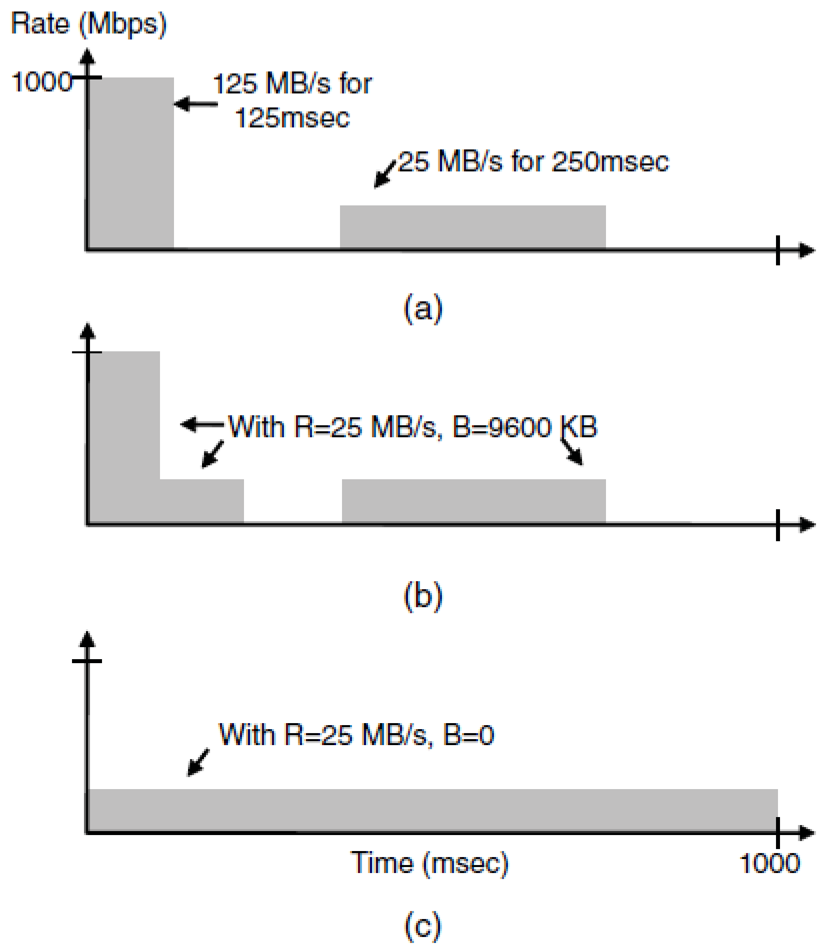
1. Constant bit rate
   - Telephony
2. Real-time variable bit rate
   - Compressed videoconferencing
3. Non-real-time variable bit rate
   - Watching a movie on demand
4. Available bit rate
   - File transfer

Traffic Shaping (1)



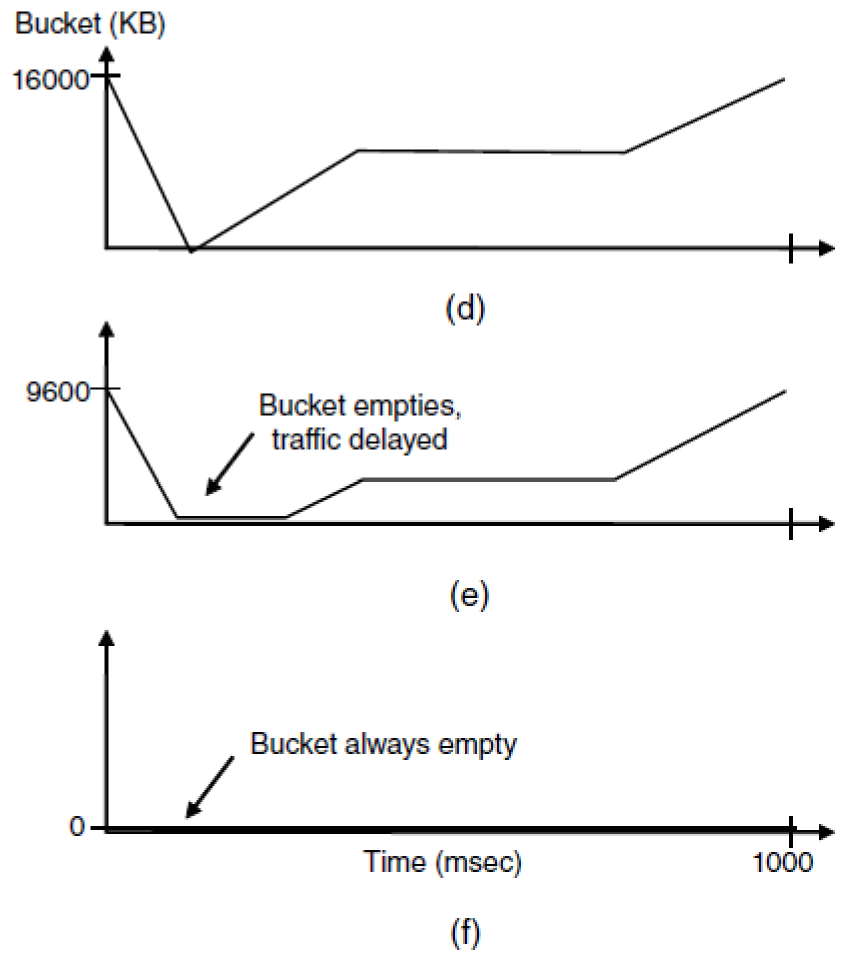
(a) Shaping packets. (b) A leaky bucket. (c) A token bucket

Traffic Shaping (2)



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

Traffic Shaping (3)



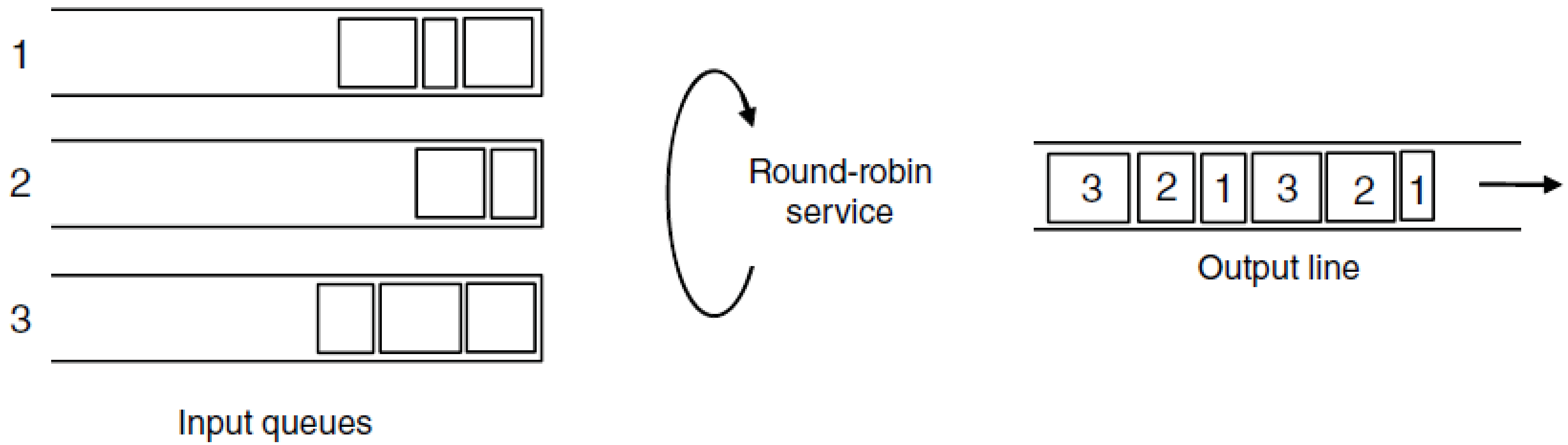
Token bucket level for shaping with rate 200 Mbps and capacity (d) 16000 KB, (e) 9600 KB, and (f) 0KB..

Packet Scheduling (1)

Kinds of resources can potentially be reserved for different flows:

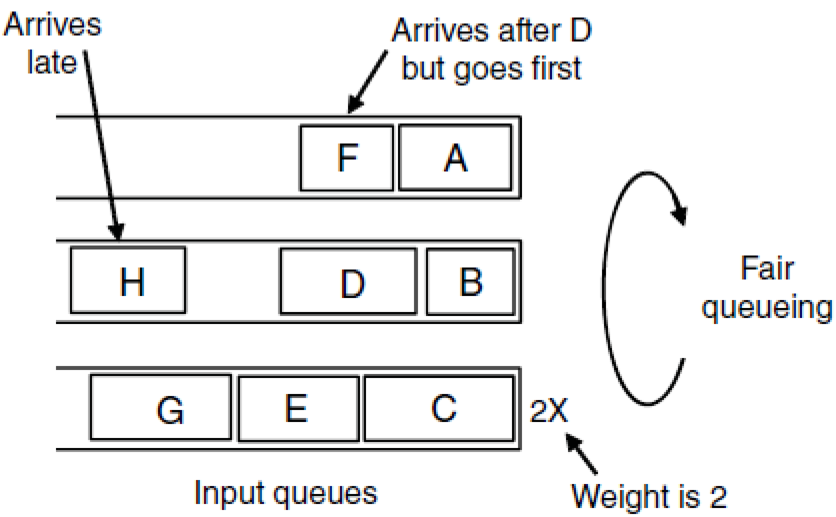
1. Bandwidth.
2. Buffer space.
3. CPU cycles.

Packet Scheduling (2)



Round-robin Fair Queuing

Packet Scheduling (3)



(a)

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

(b)

(a) Weighted Fair Queueing.

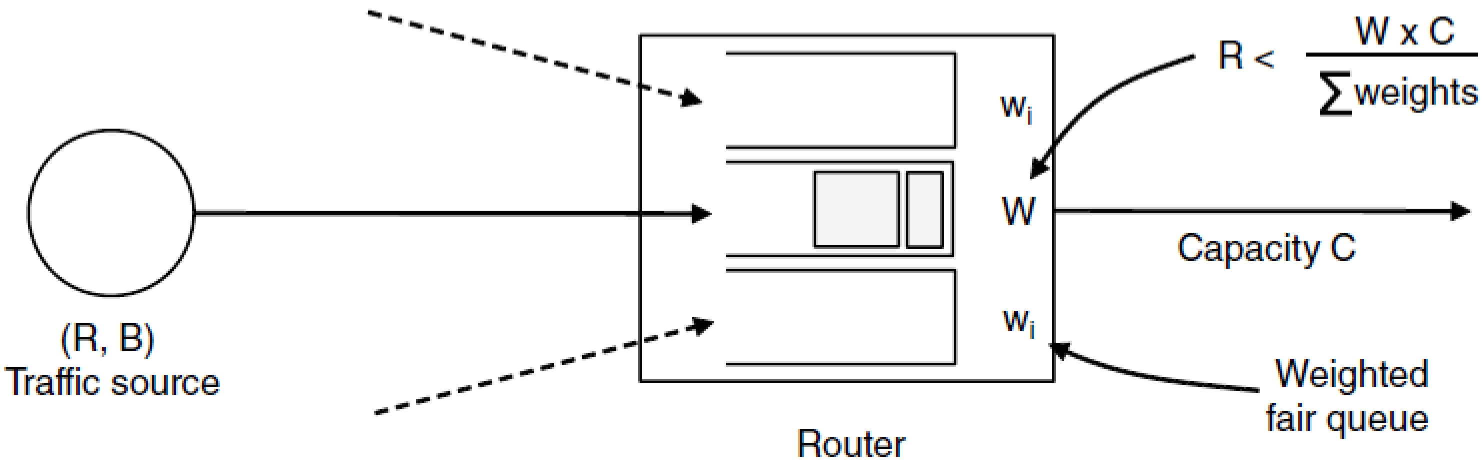
(b) Finishing times for the packets.

Admission Control (1)

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

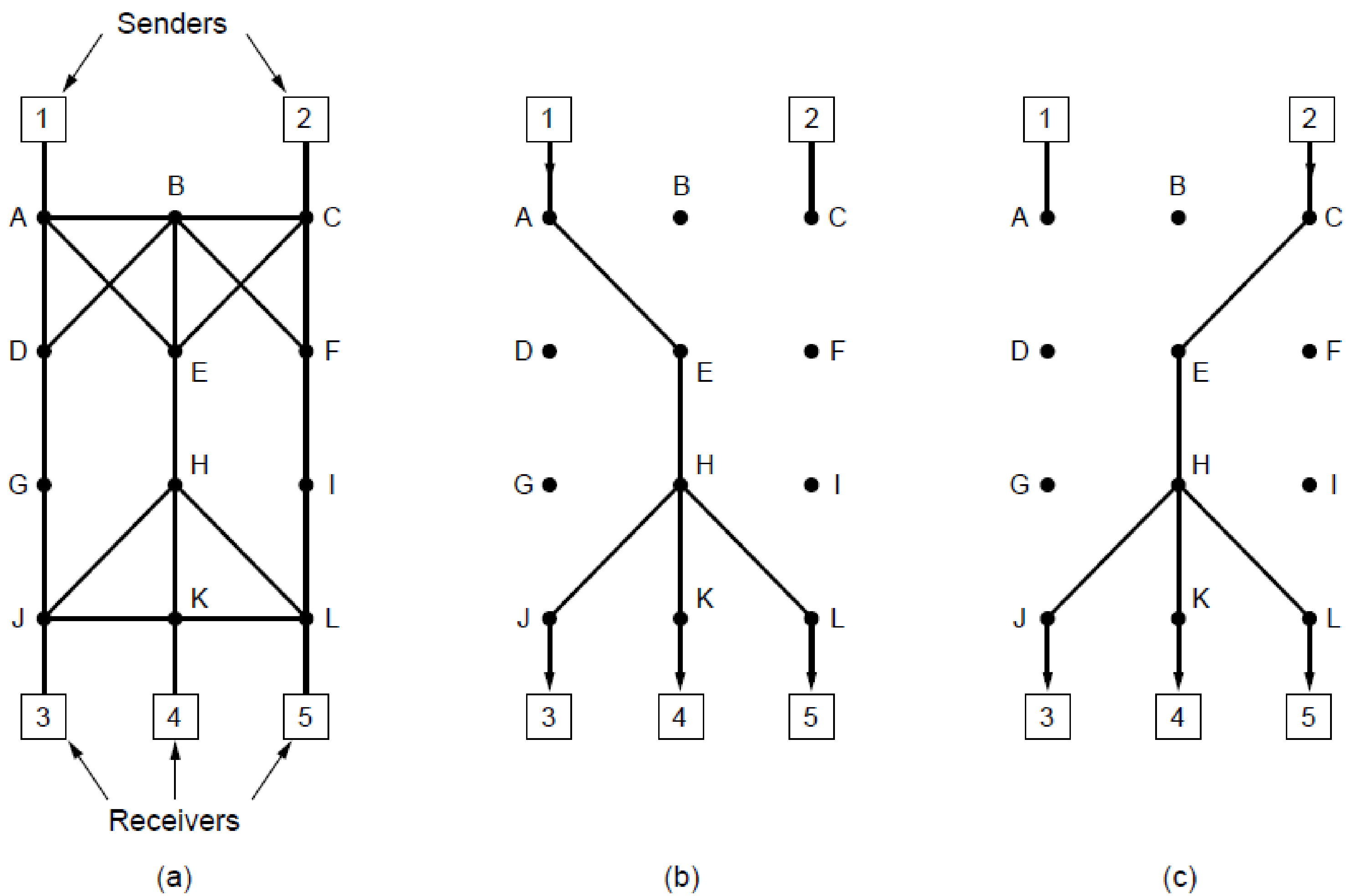
An example flow specification

Admission Control (2)



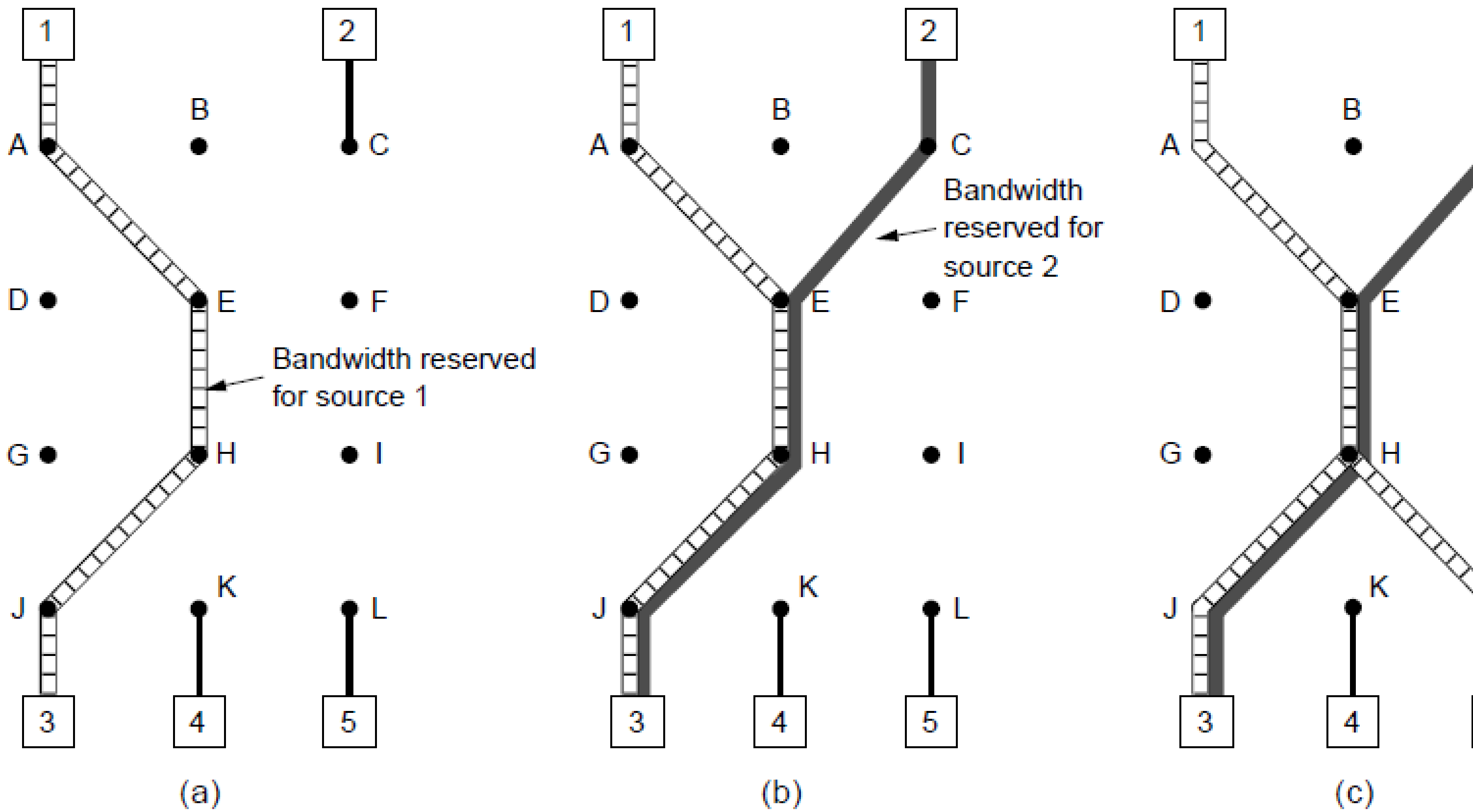
Bandwidth and delay guarantees with token buckets and WFQ.

Integrated Services (1)



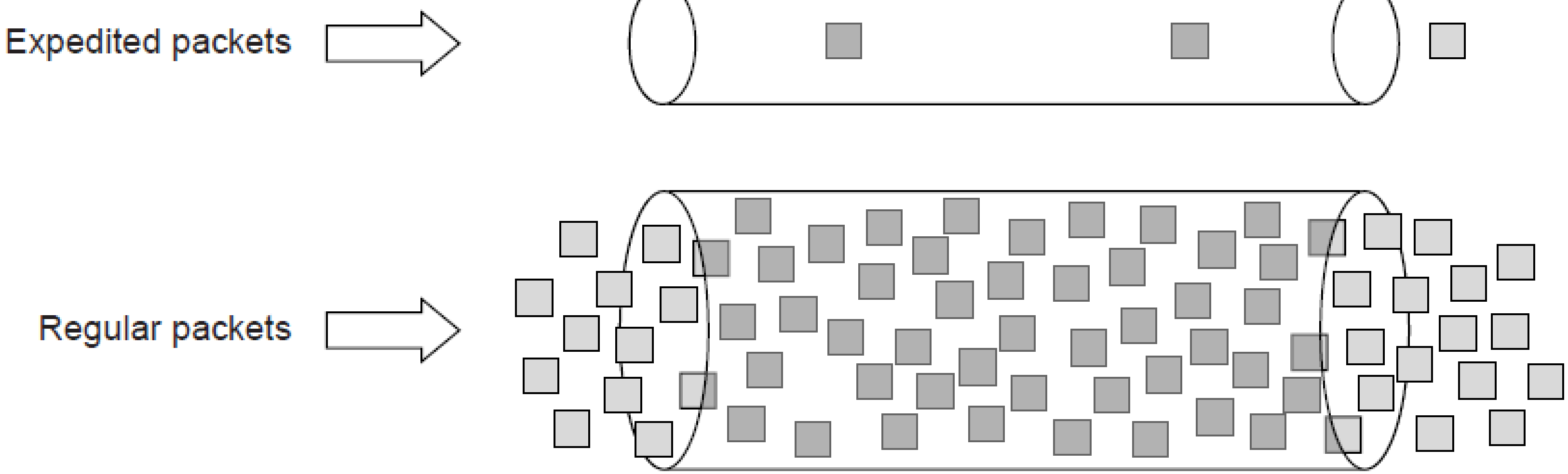
(a) A network. (b) The multicast spanning tree for host 1. (c) The multicast spanning tree for host 2.

Integrated Services (2)



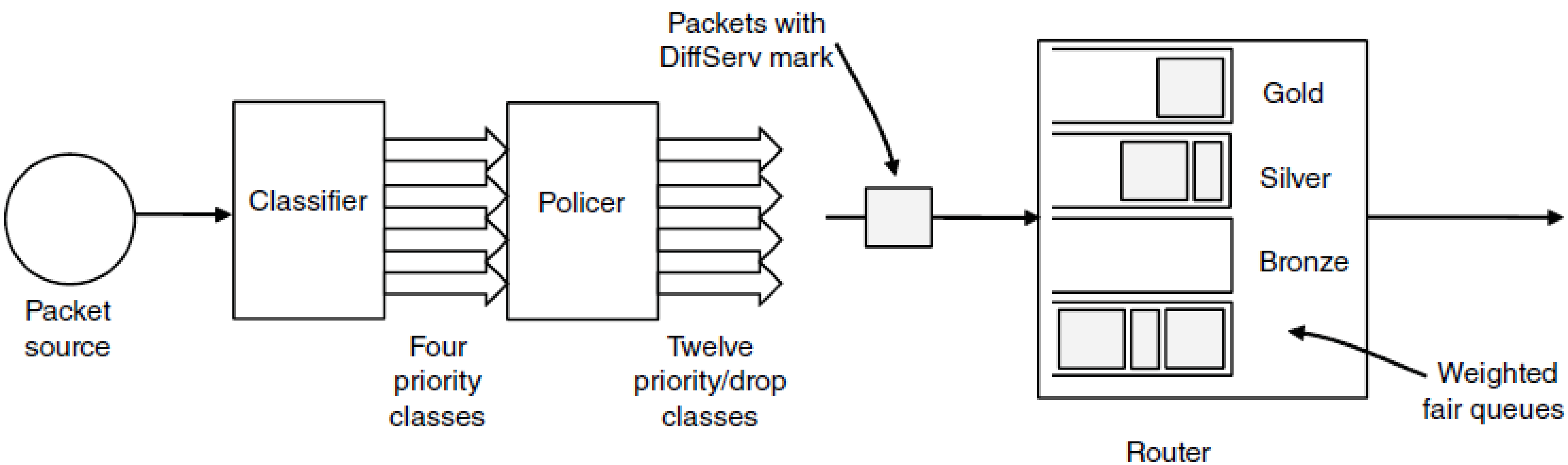
(a) Host 3 requests a channel to host 1. (b) Host 3 then requests a second channel, to host 2.
(c) Host 5 requests a channel to host 1.

Differentiated Services (1)



Expedited packets experience a traffic-free network

Differentiated Services (2)



A possible implementation of assured forwarding

Internetworking

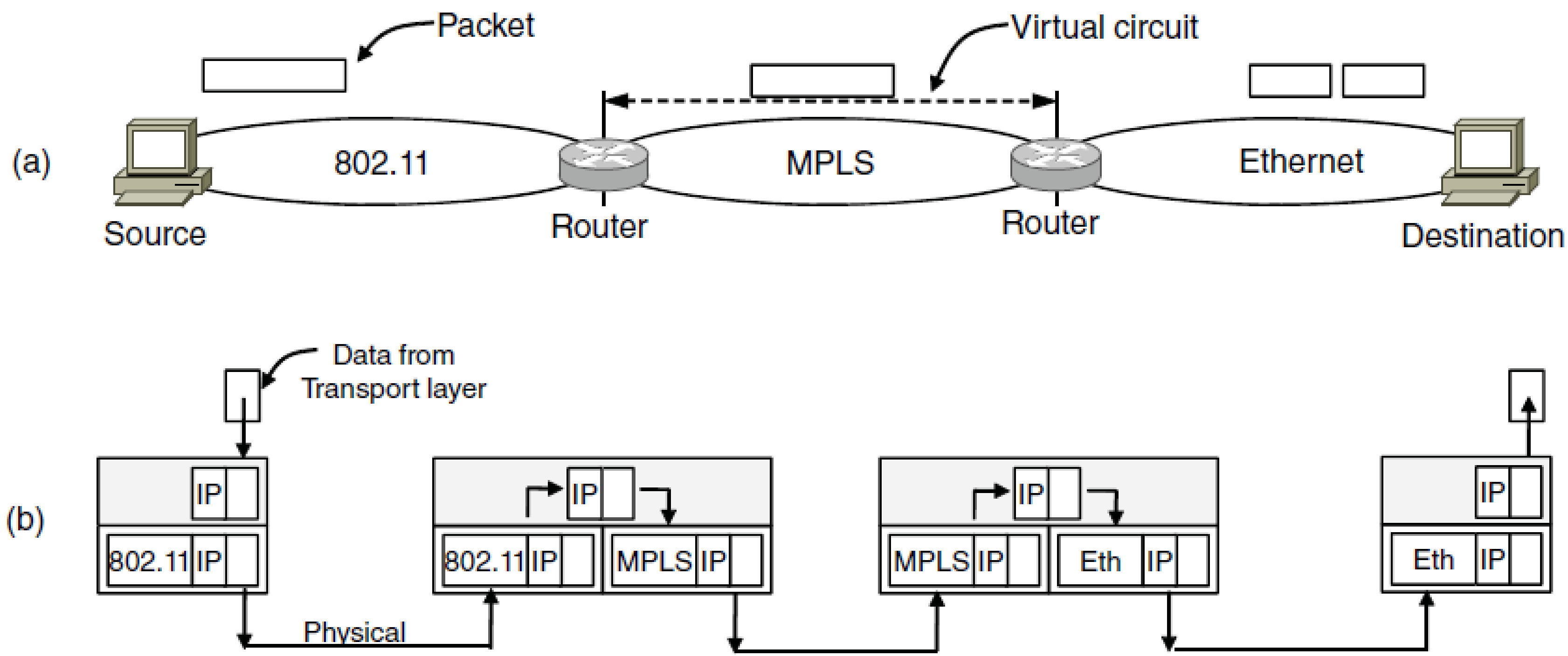
- How networks differ
- How networks can be connected
- Tunneling
- Internetwork routing
- Packet fragmentation

How Networks Differ

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

Some of the many ways networks can differ

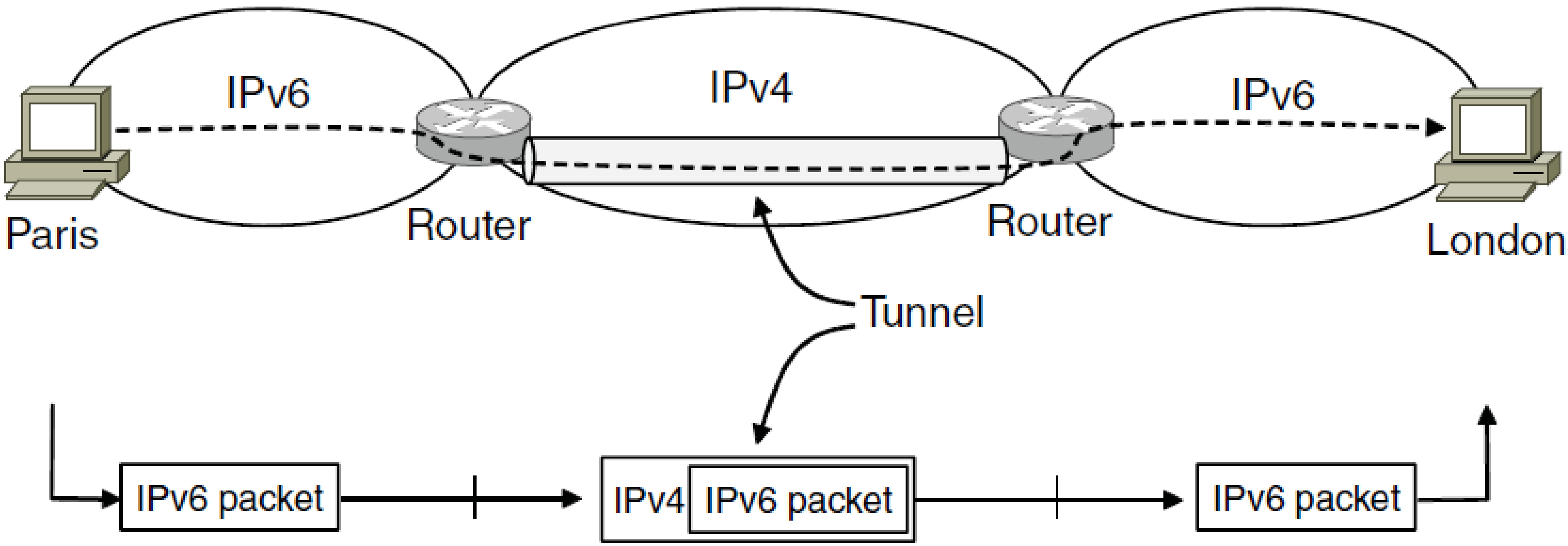
How Networks Can Be Connected



(a) A packet crossing different networks.

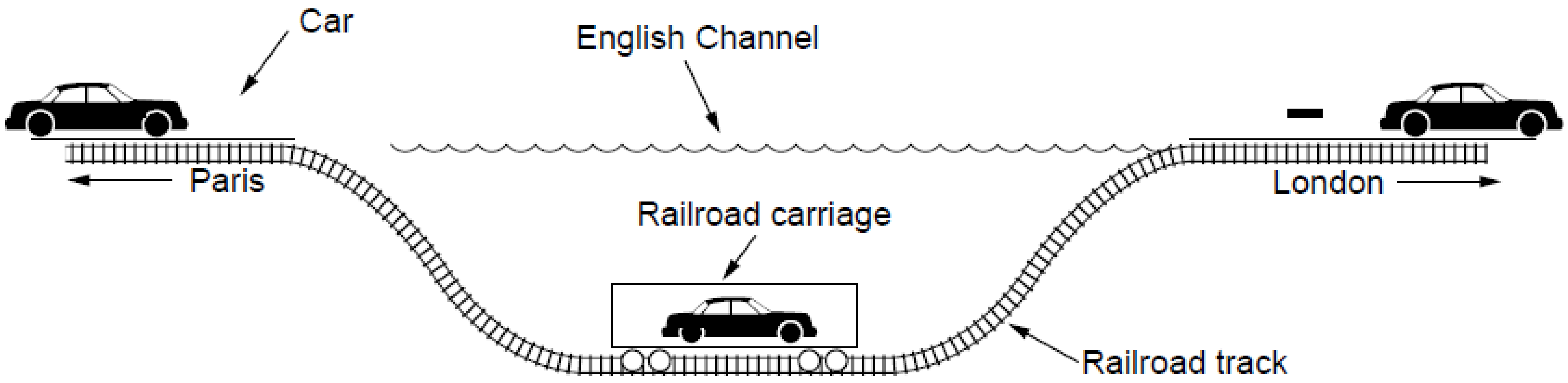
(b) Network and link layer protocol processing.

Tunneling (1)



Tunneling a packet from Paris to London.

Tunneling (2)



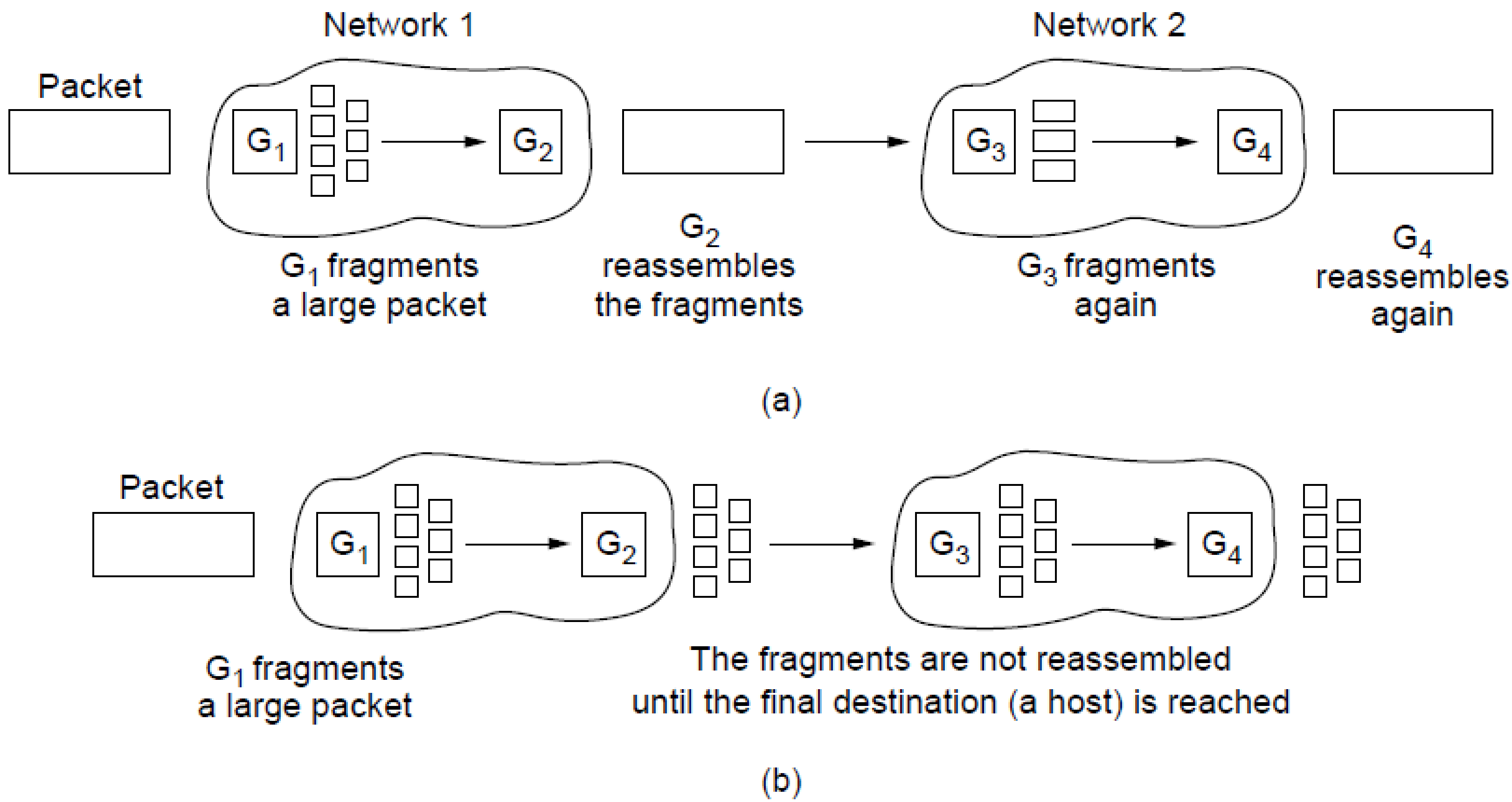
Tunneling a car from France to England

Packet Fragmentation (1)

Packet size issues:

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

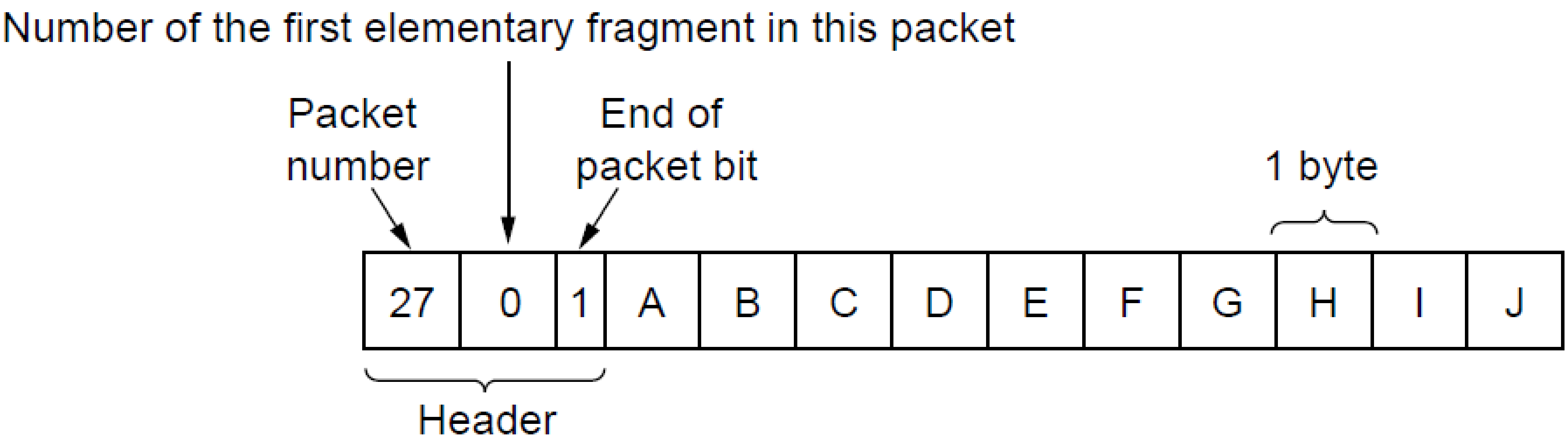
Packet Fragmentation (2)



(a) Transparent fragmentation.

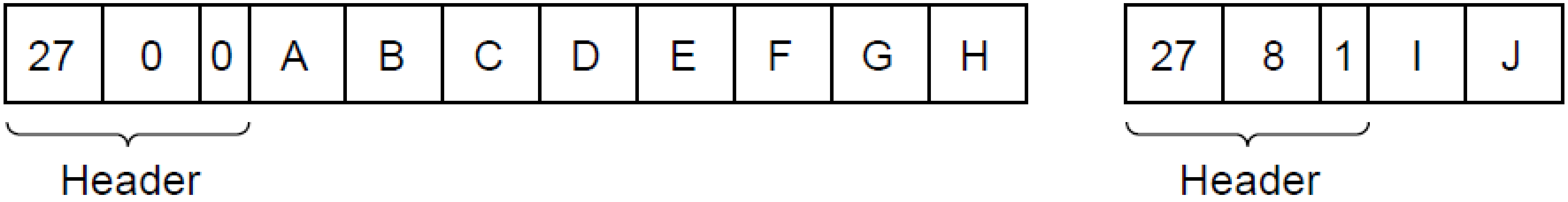
(b) Nontransparent fragmentation

Packet Fragmentation (3)



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

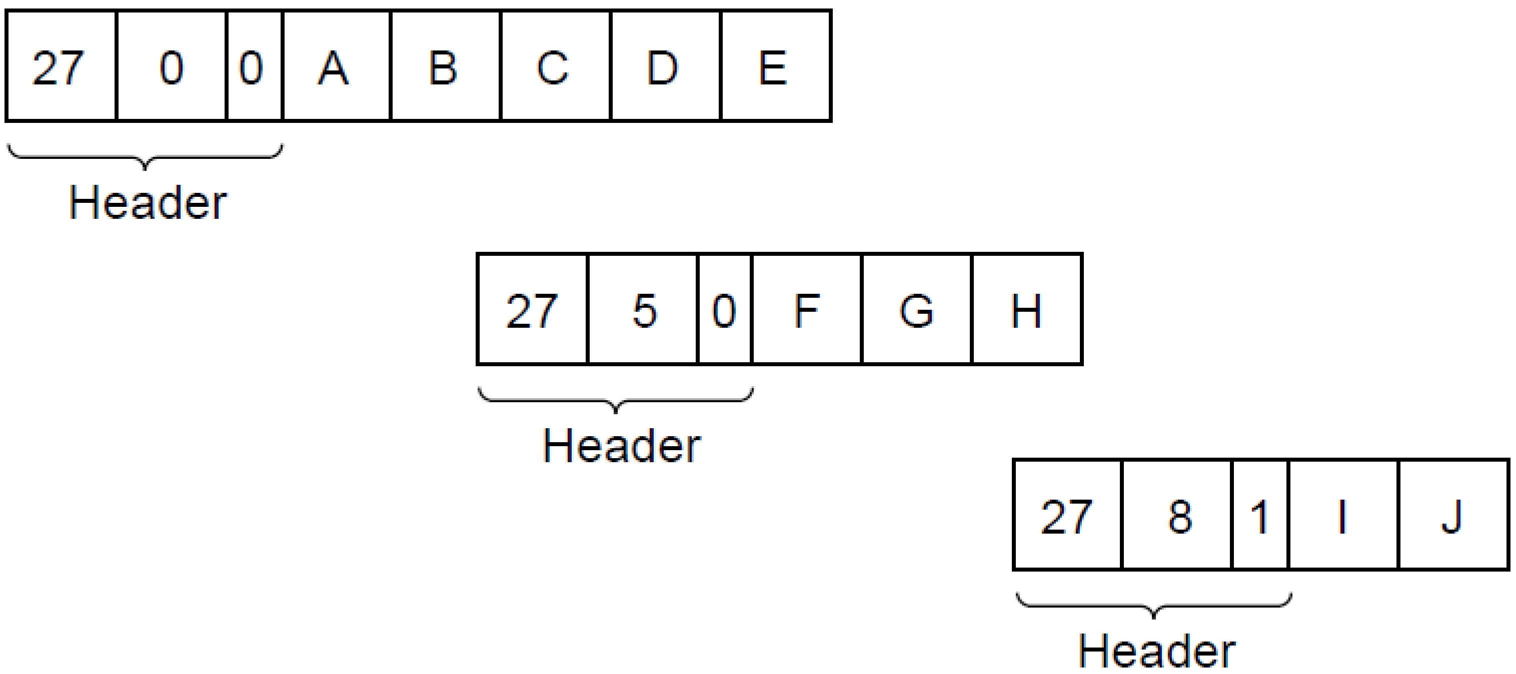
Packet Fragmentation (4)



Fragmentation when the elementary data size is 1 byte

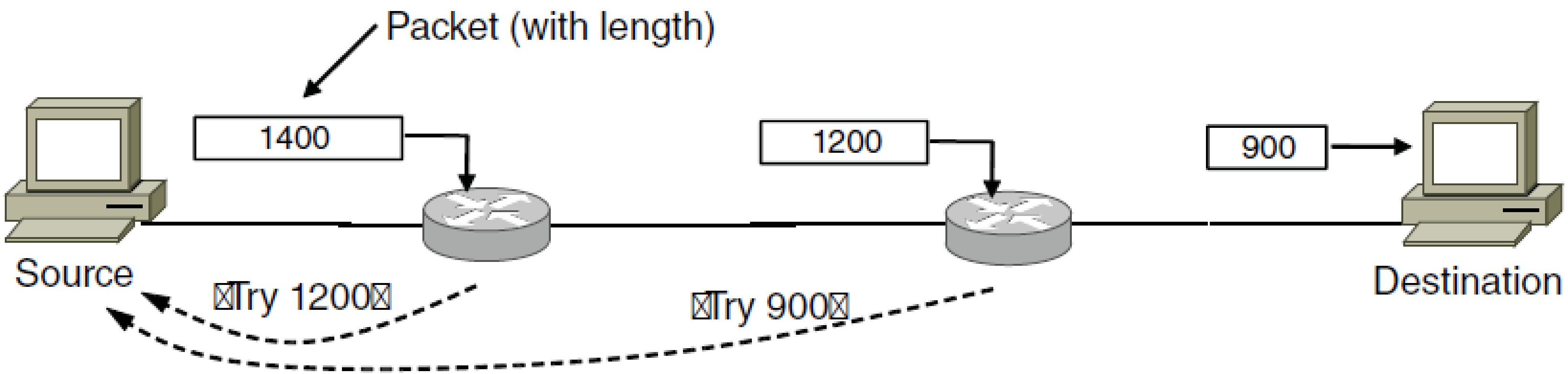
(b) Fragments after passing through a network with maximum packet size of 8 payload bytes plus header.

Packet Fragmentation (5)



Fragmentation when the elementary data size is 1 byte
(c) Fragments after passing through a size 5 gateway.

Packet Fragmentation (6)



Path MTU Discovery

The Network Layer Principles (1)

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

   . . .

The Network Layer Principles (2)

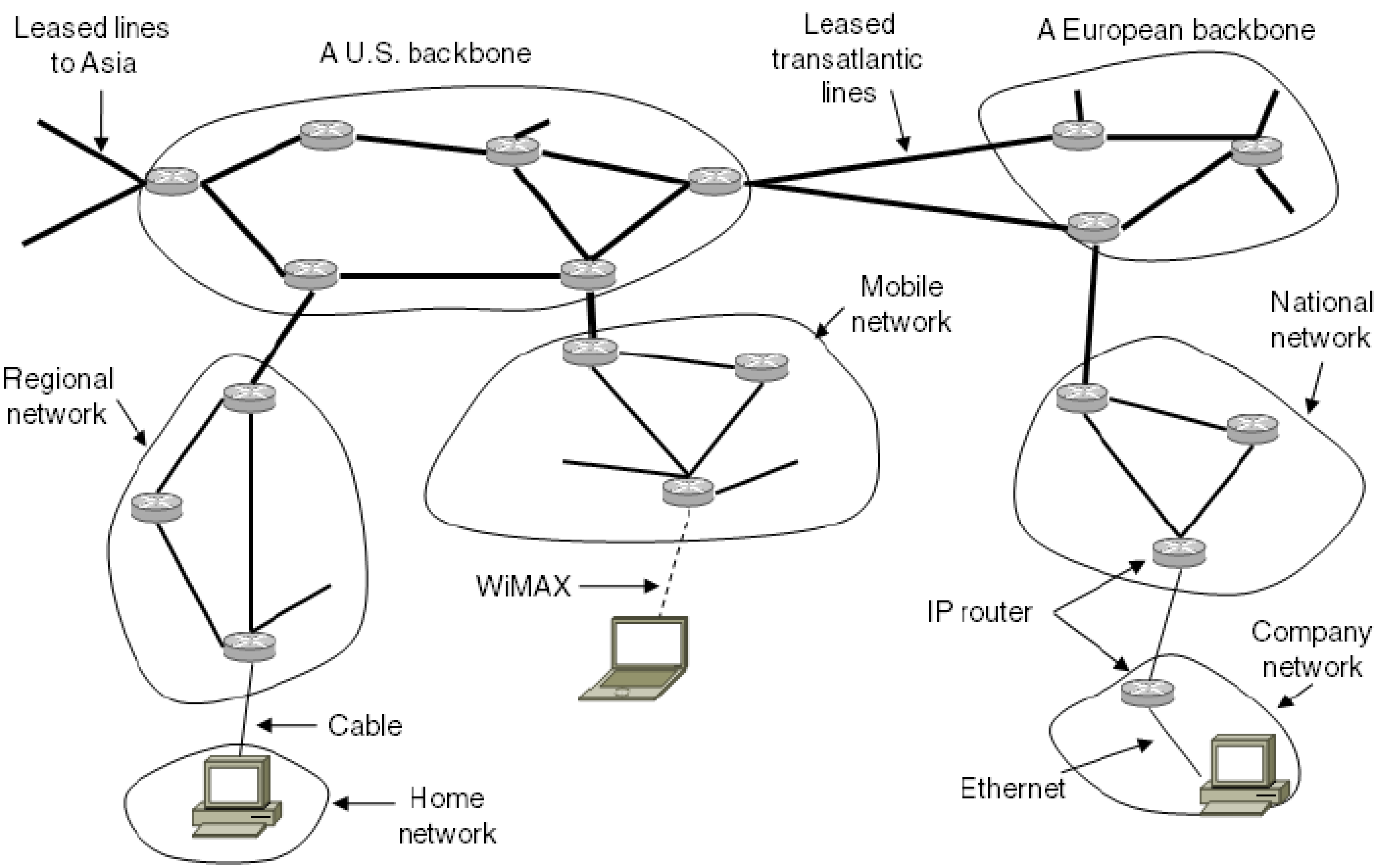
. . .

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

The Network Layer in the Internet (1)

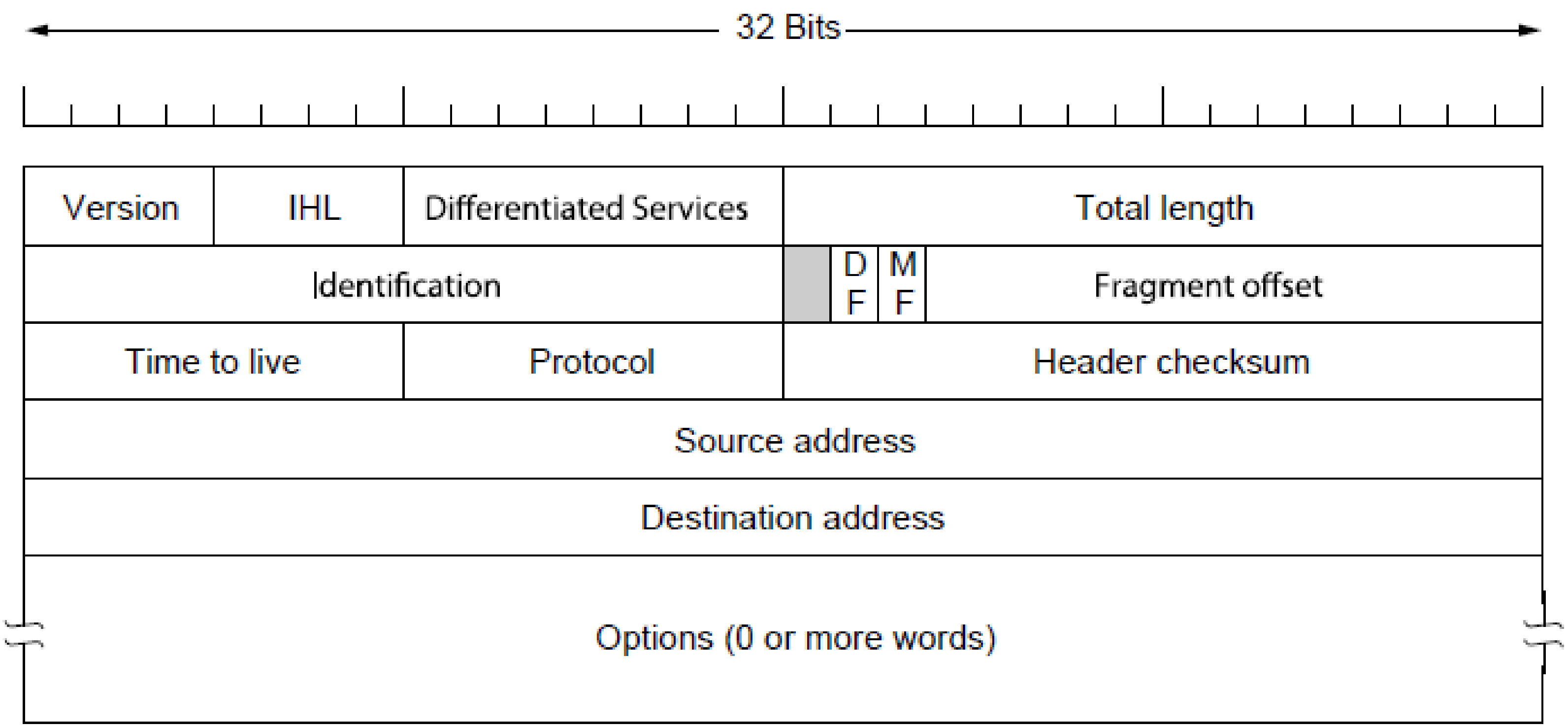
- The IP Version 4 Protocol
- IP Addresses
- IP Version 6
- Internet Control Protocols
- Label Switching and MPLS
- OSPF—An Interior Gateway Routing Protocol
- BGP—The Exterior Gateway Routing Protocol
- Internet Multicasting
- Mobile IP

The Network Layer in the Internet (2)



The Internet is an interconnected collection of many networks.

The IP Version 4 Protocol (1)



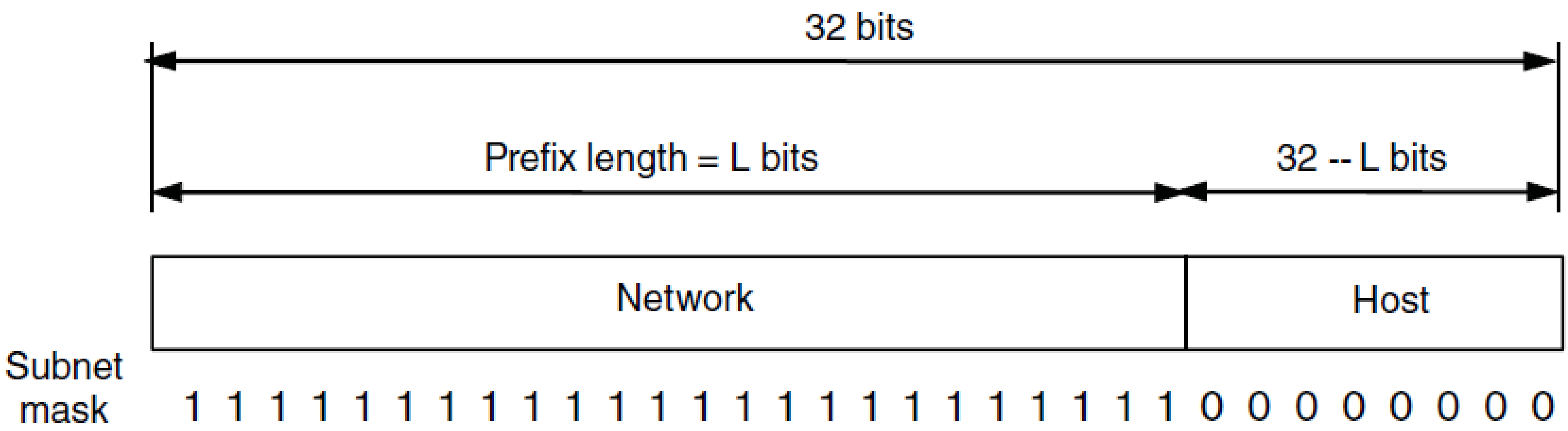
The IPv4 (Internet Protocol) header.

The IP Version 4 Protocol (2)

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

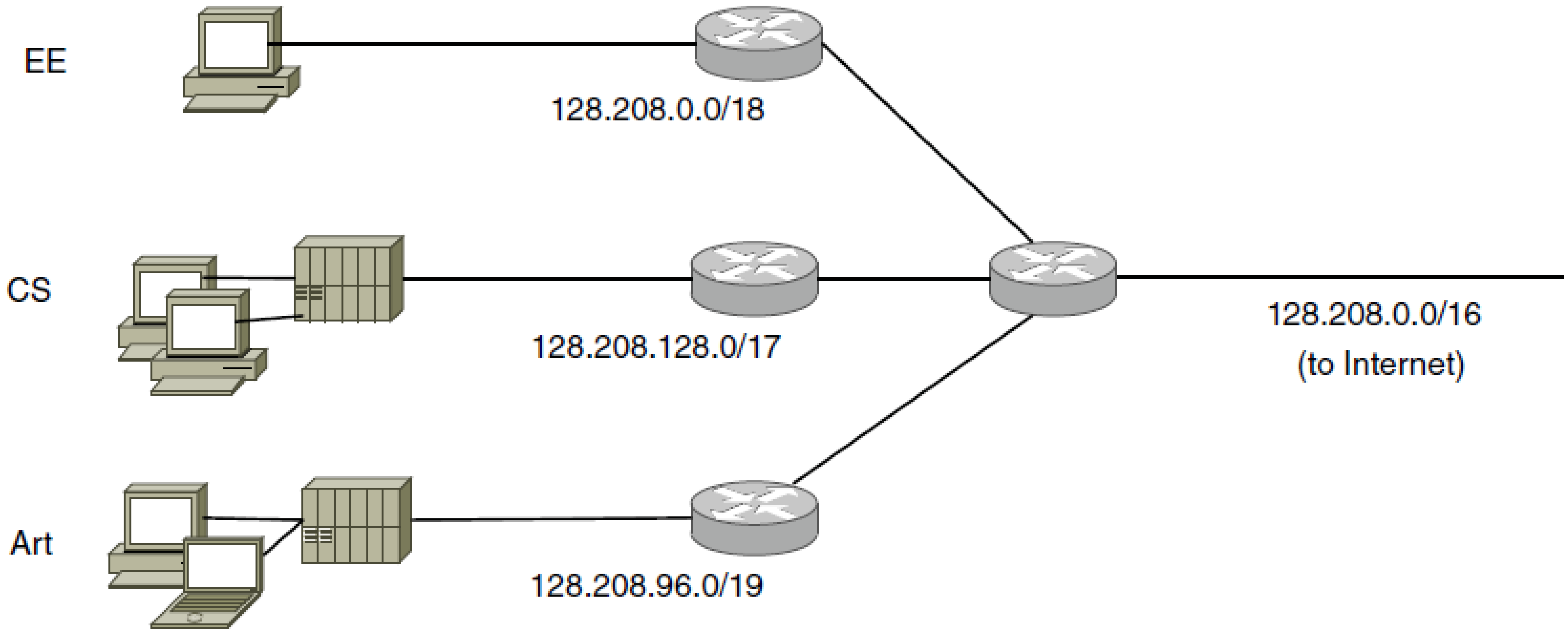
Some of the IP options.

IP Addresses (1)



An IP prefix.

IP Addresses (2)



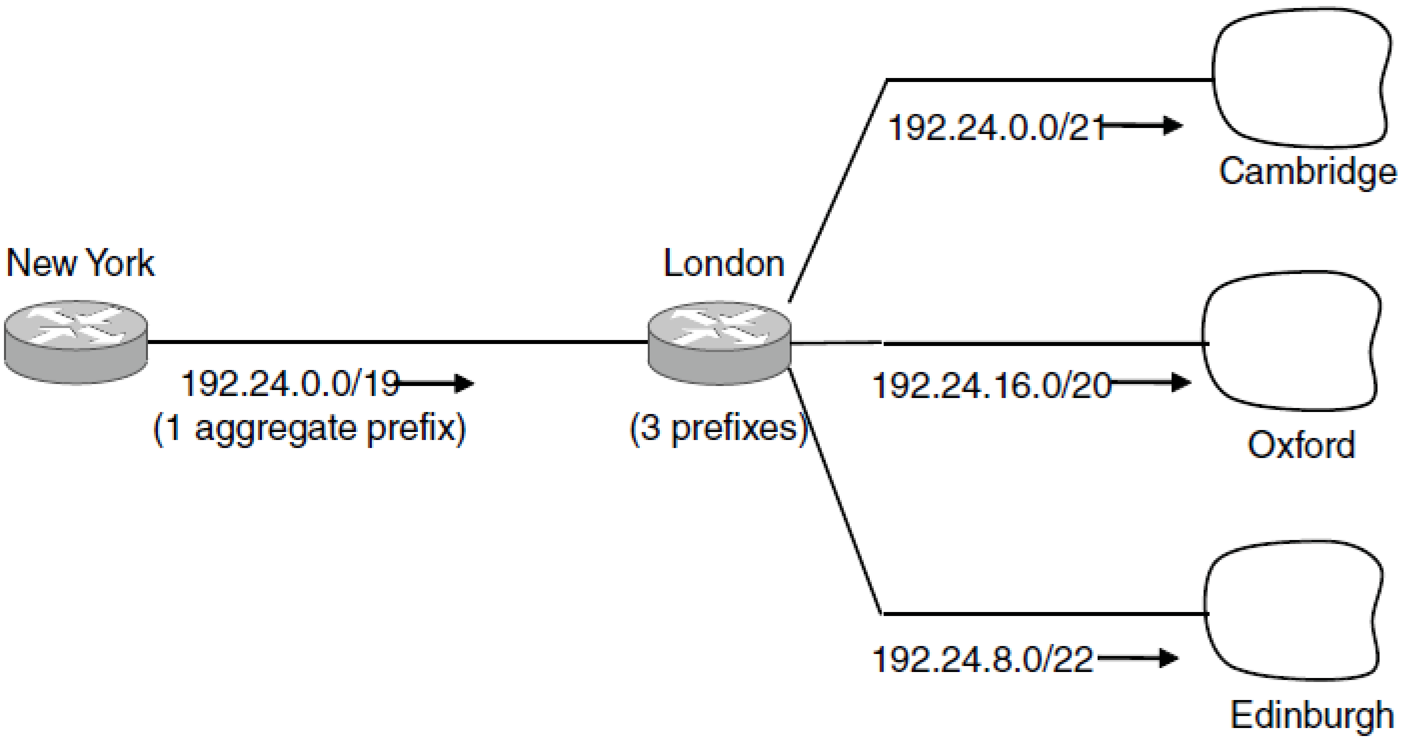
Splitting an IP prefix into separate networks with subnetting.

IP Addresses (3)

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

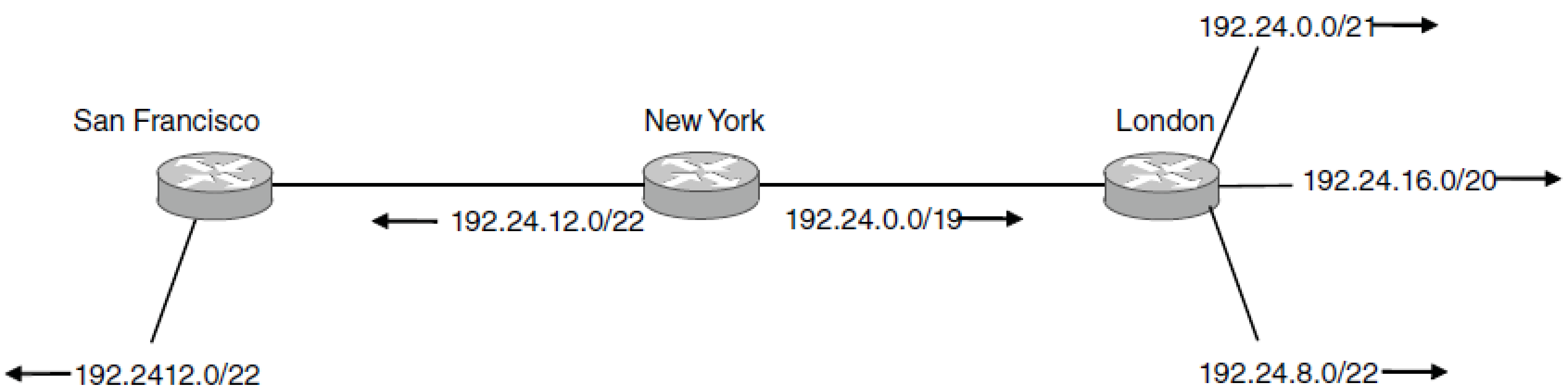
A set of IP address assignments

IP Addresses (4)



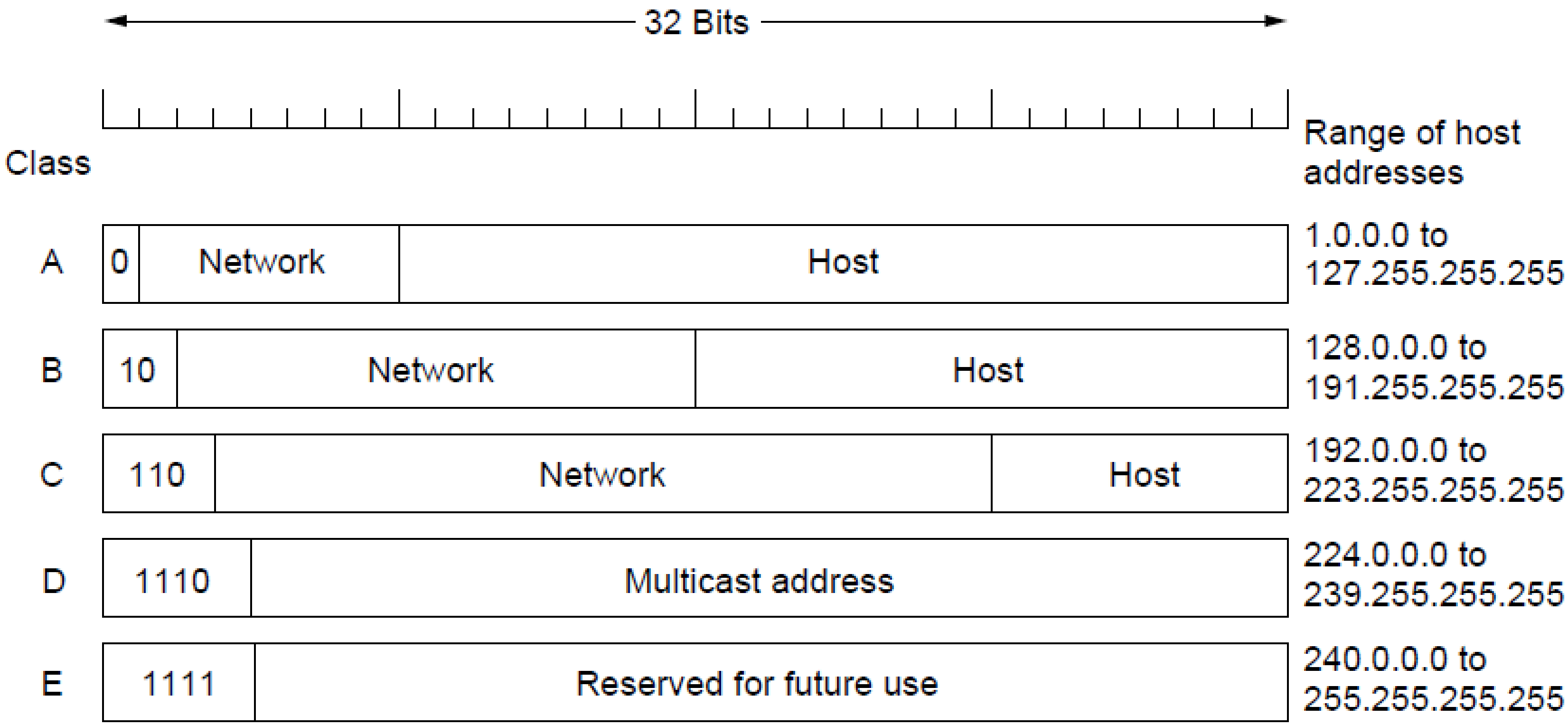
Aggregation of IP prefixes

IP Addresses (5)



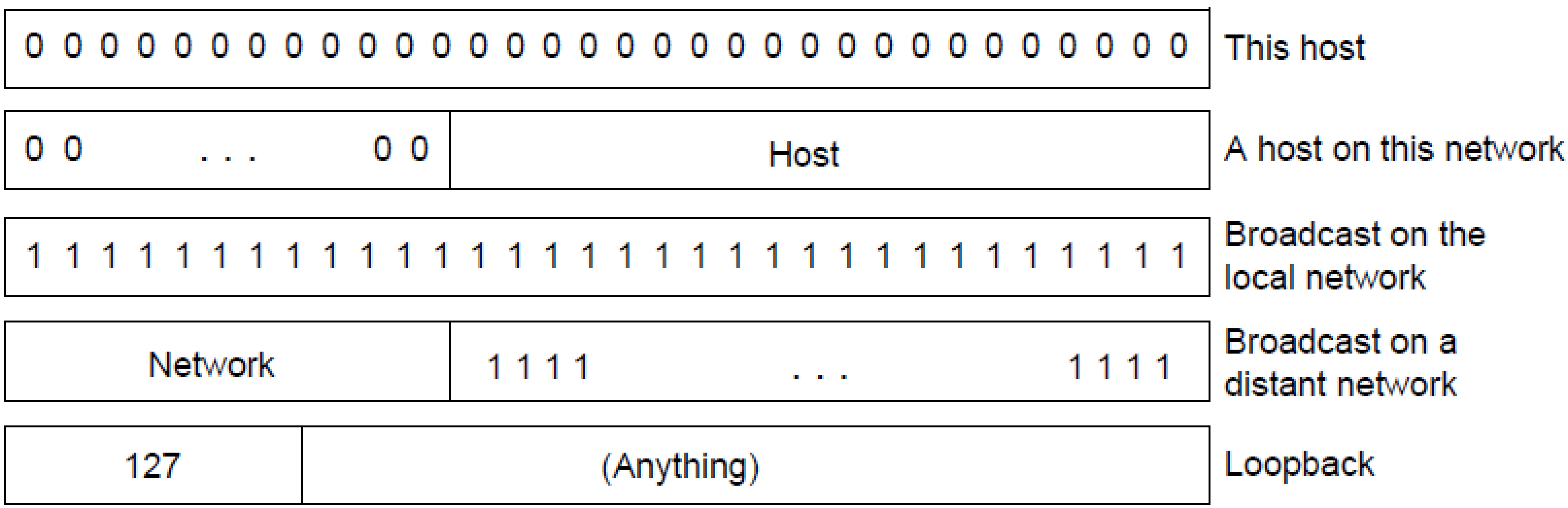
Longest matching prefix routing at the New York router.

IP Addresses (6)



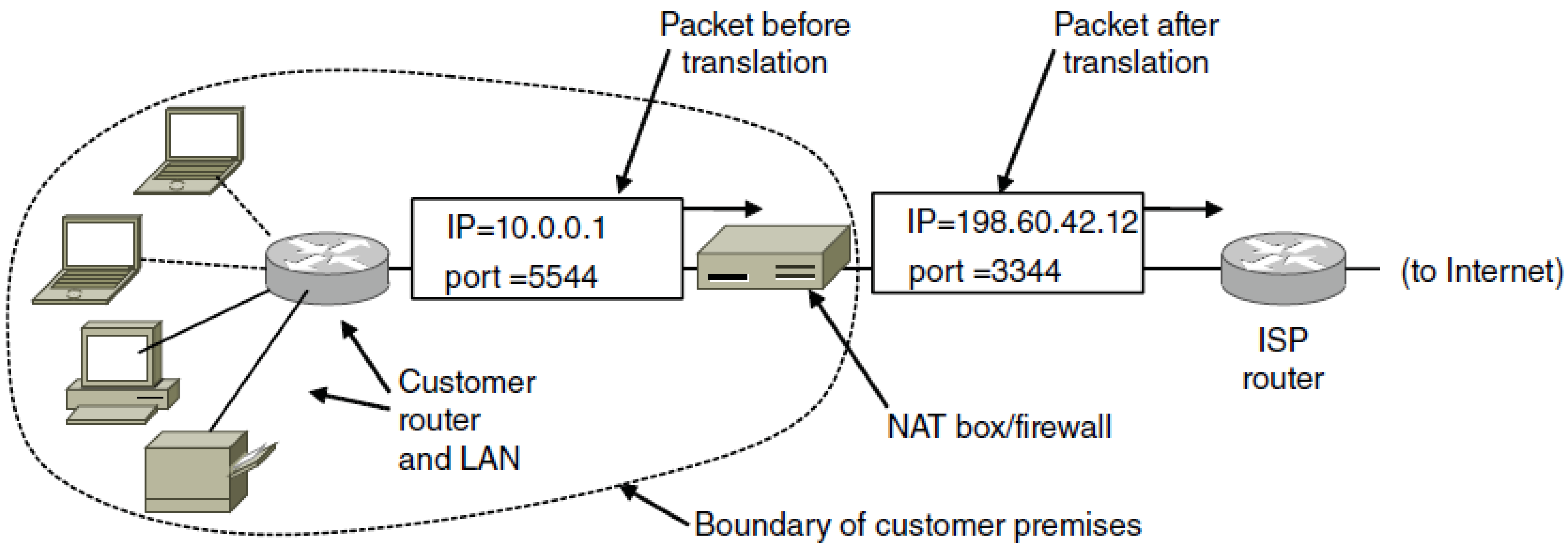
IP address formats

IP Addresses (7)



Special IP addresses

IP Addresses (8)

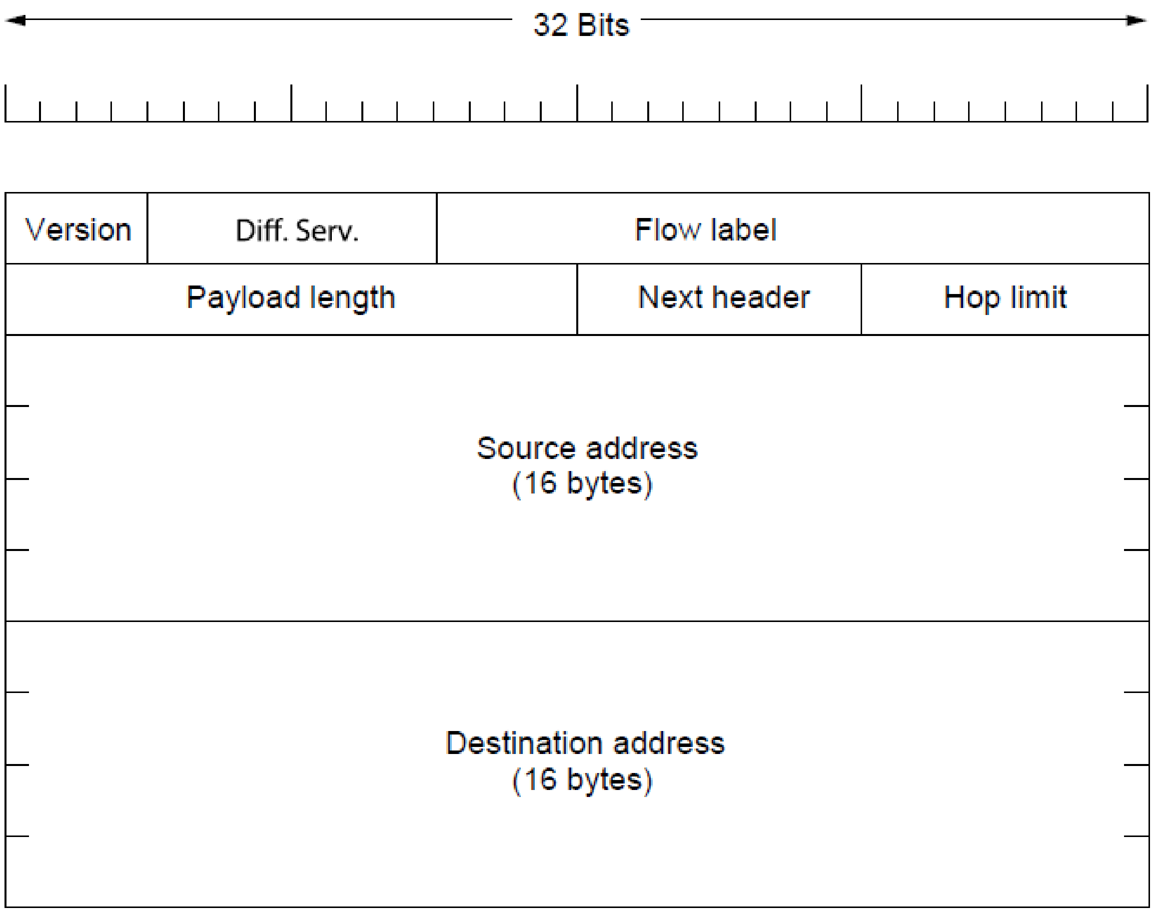


Placement and operation of a NAT box.

IP Version 6 Goals

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

IP Version 6 (1)



The IPv6 fixed header (required).

IP Version 6 (2)

Extension header	Description
Hop-by-hop options	Miscellaneous information for routers
Destination options	Additional information for the destination
Routing	Loose list of routers to visit
Fragmentation	Management of datagram fragments
Authentication	Verification of the sender's identity
Encrypted security payload	Information about the encrypted contents

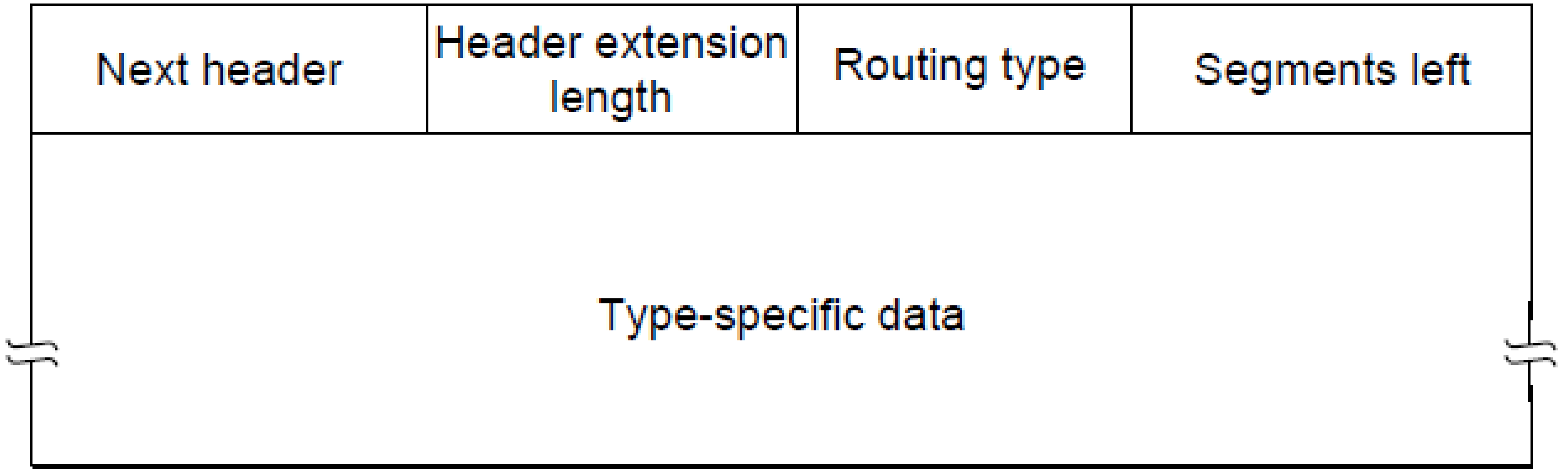
IPv6 extension headers

IP Version 6 (3)

Next header	0	194	4
Jumbo payload length			

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

IP Version 6 (4)



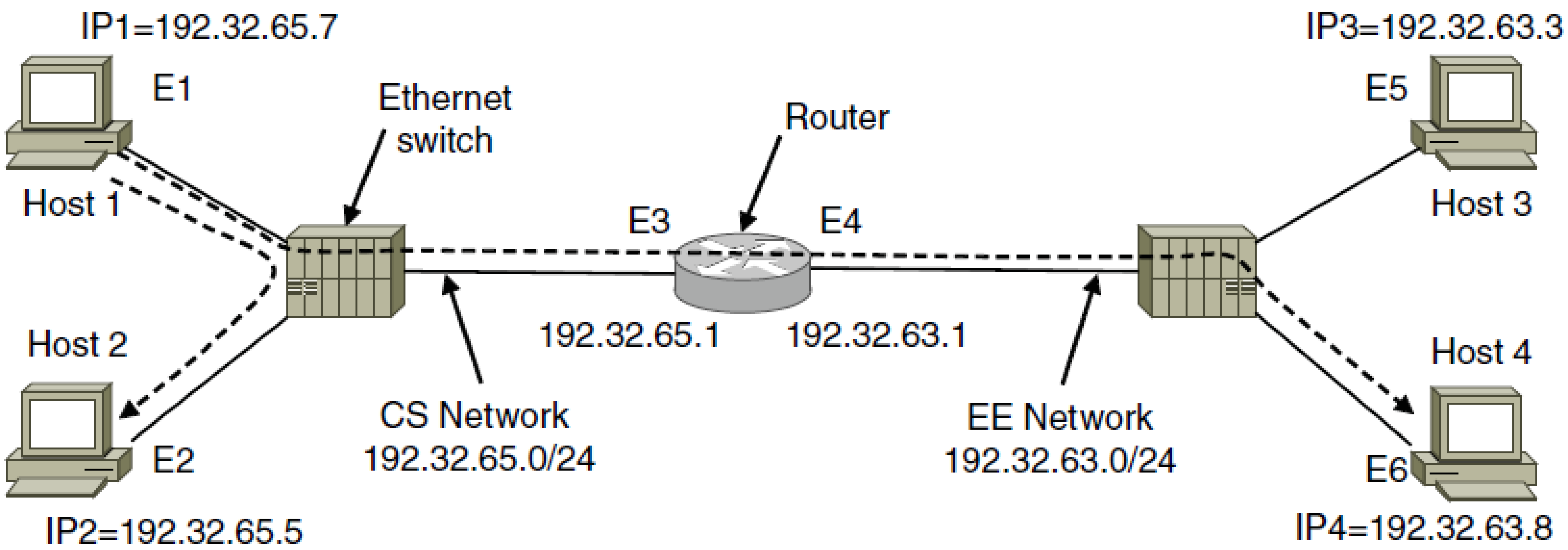
The extension header for routing.

Internet Control Protocols (1)

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

The principal ICMP message types.

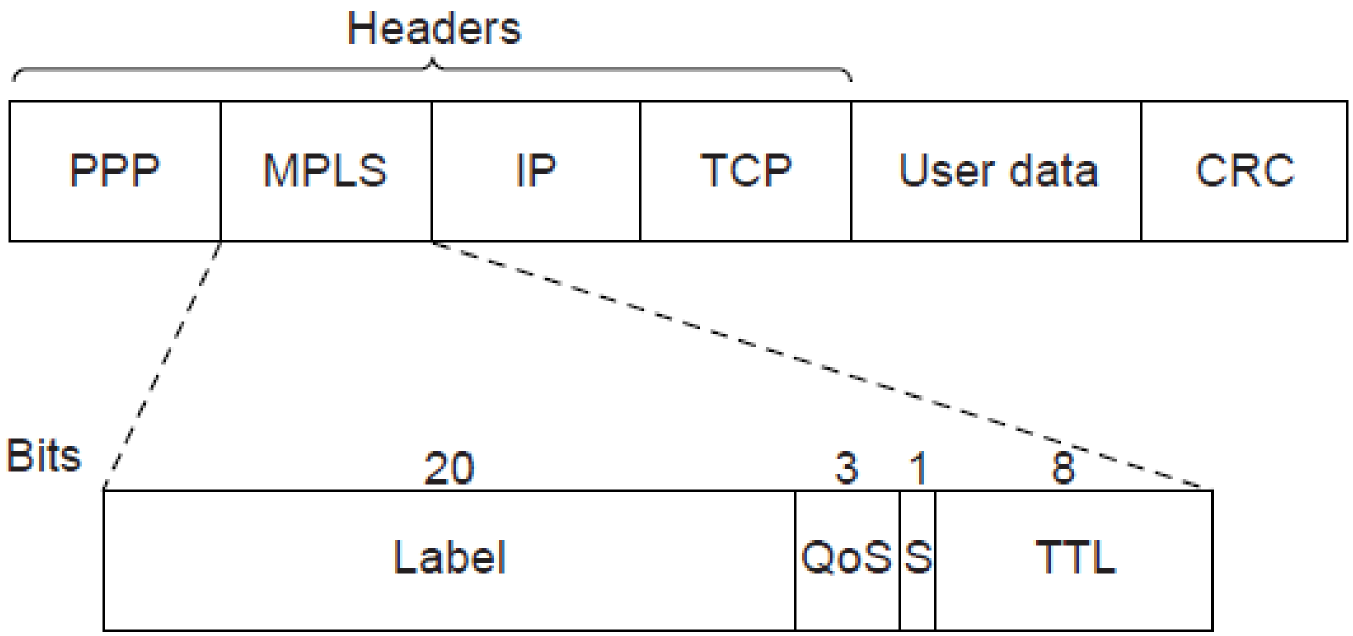
Internet Control Protocols (2)



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

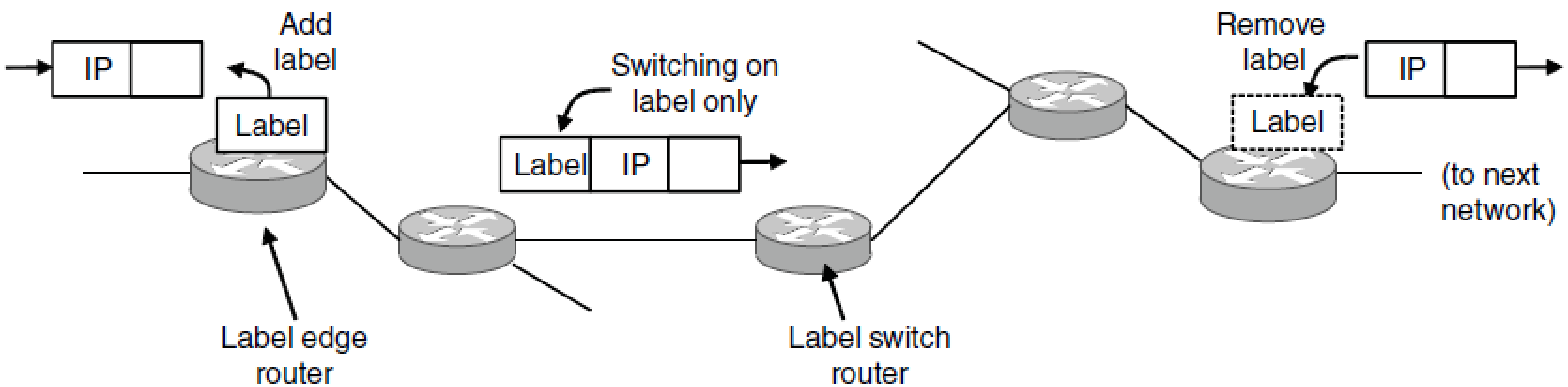
Two switched Ethernet LANs joined by a router

Label Switching and MPLS (1)



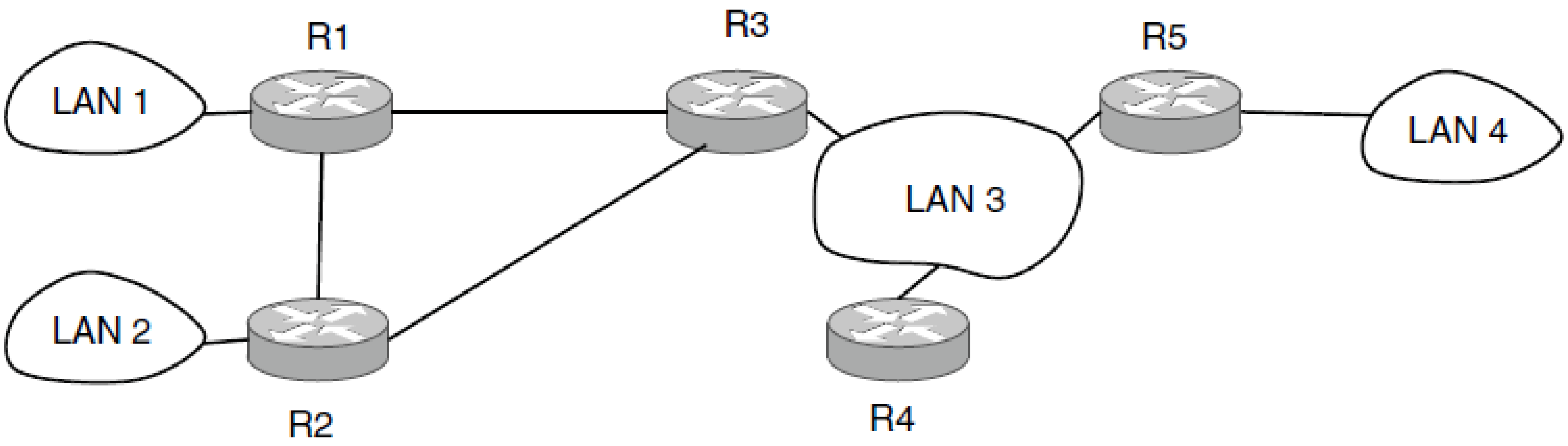
Transmitting a TCP segment using IP, MPLS, and PPP.

Label Switching and MPLS (2)



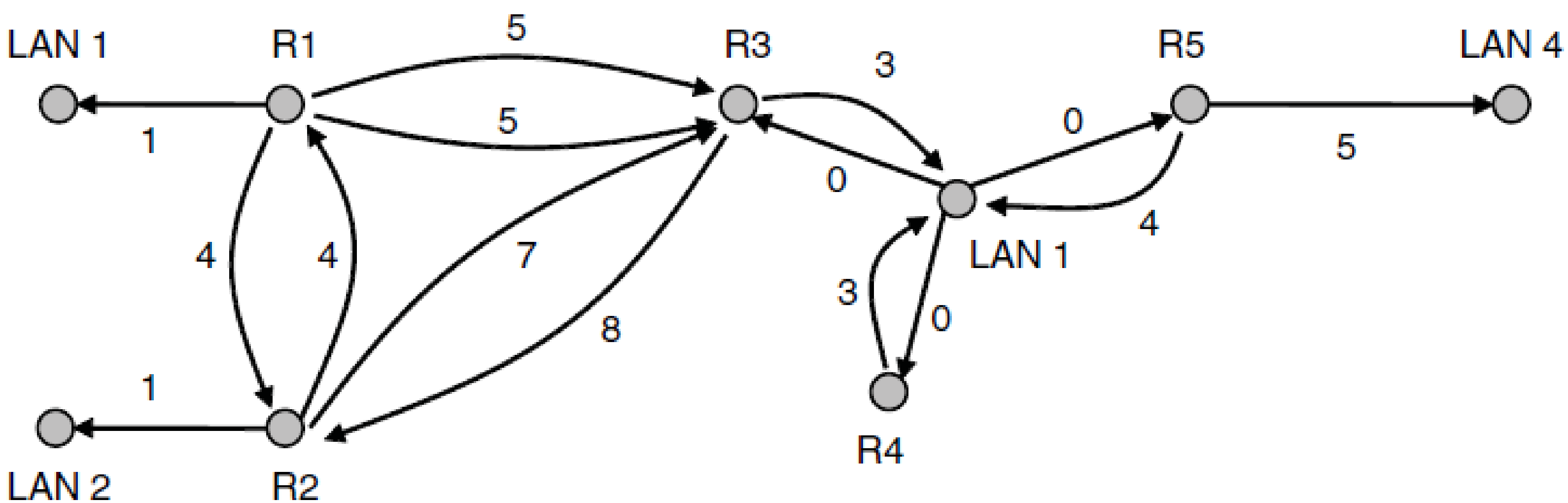
Forwarding an IP packet through an MPLS network

OSPF—An Interior Gateway Routing Protocol (1)



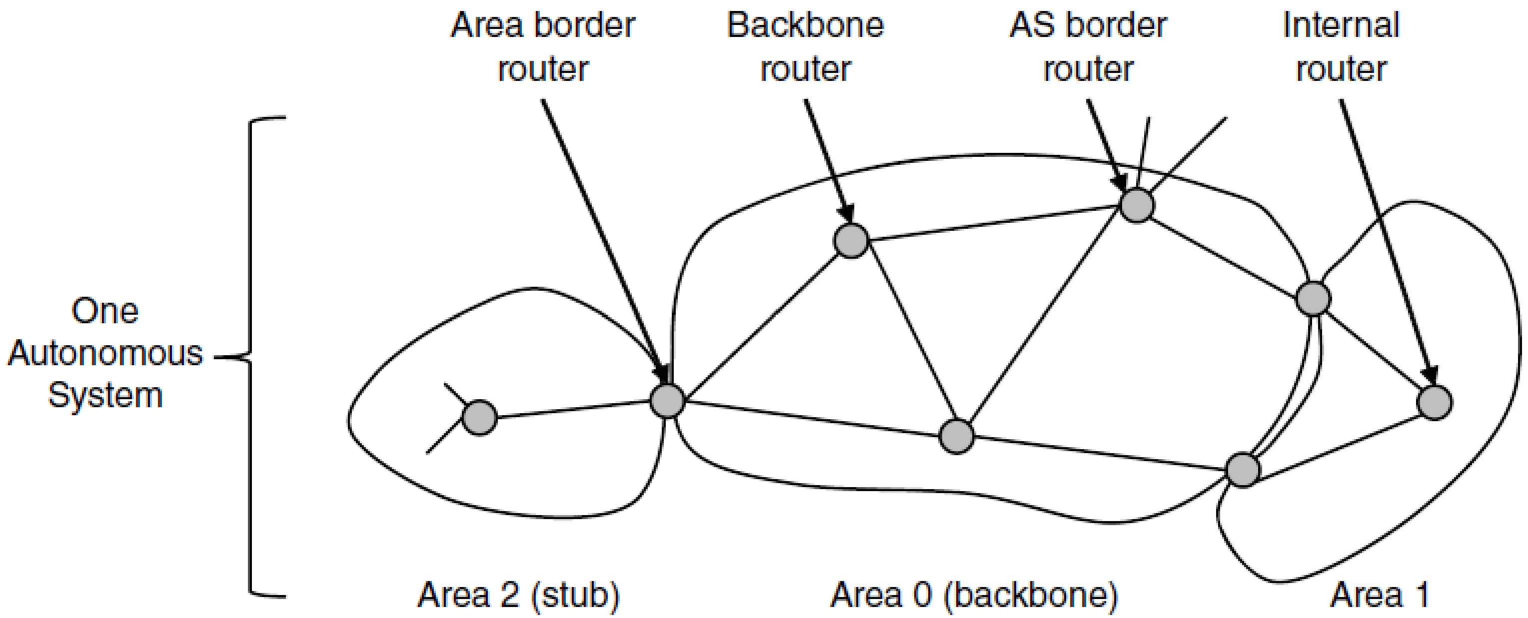
An autonomous system

OSPF—An Interior Gateway Routing Protocol (2)



A graph representation of the previous slide.

OSPF—An Interior Gateway Routing Protocol (3)



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

OSPF—An Interior Gateway Routing Protocol (4)

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

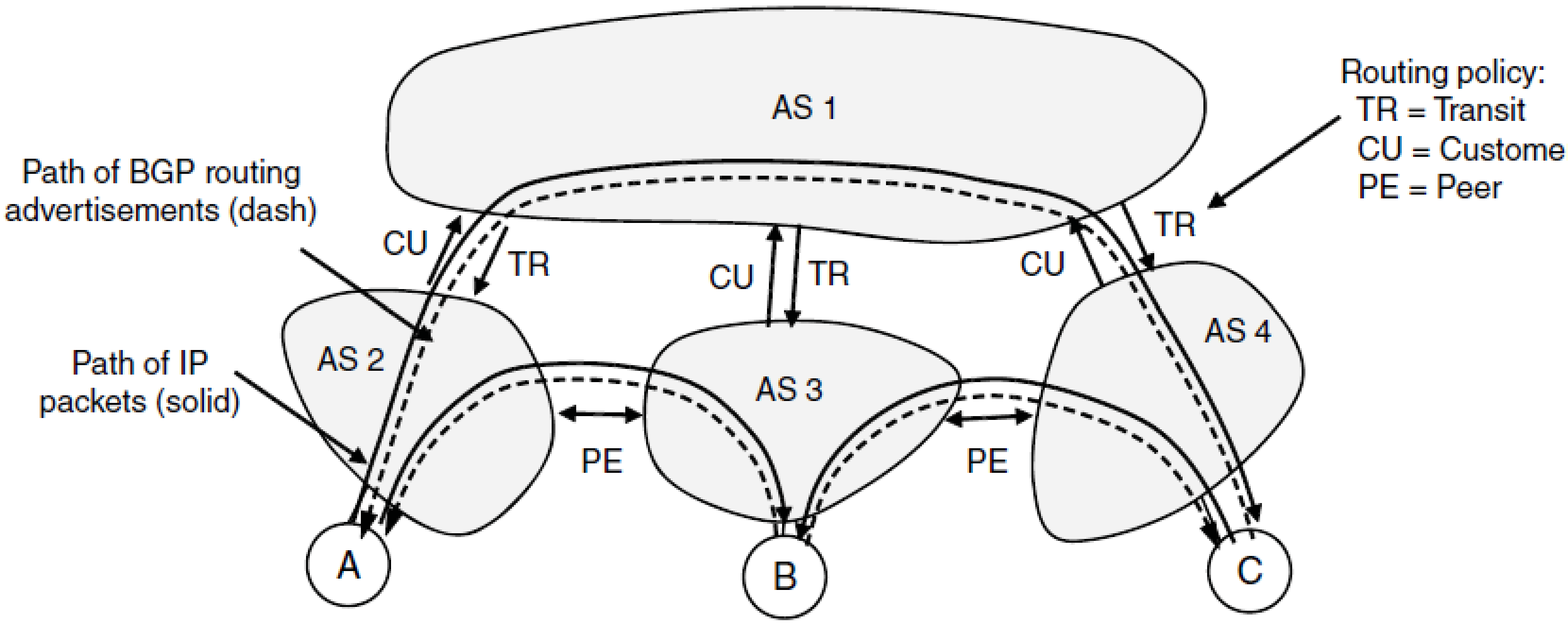
The five types of OSPF messages

BGP—The Exterior Gateway Routing Protocol (1)

Examples of routing constraints:

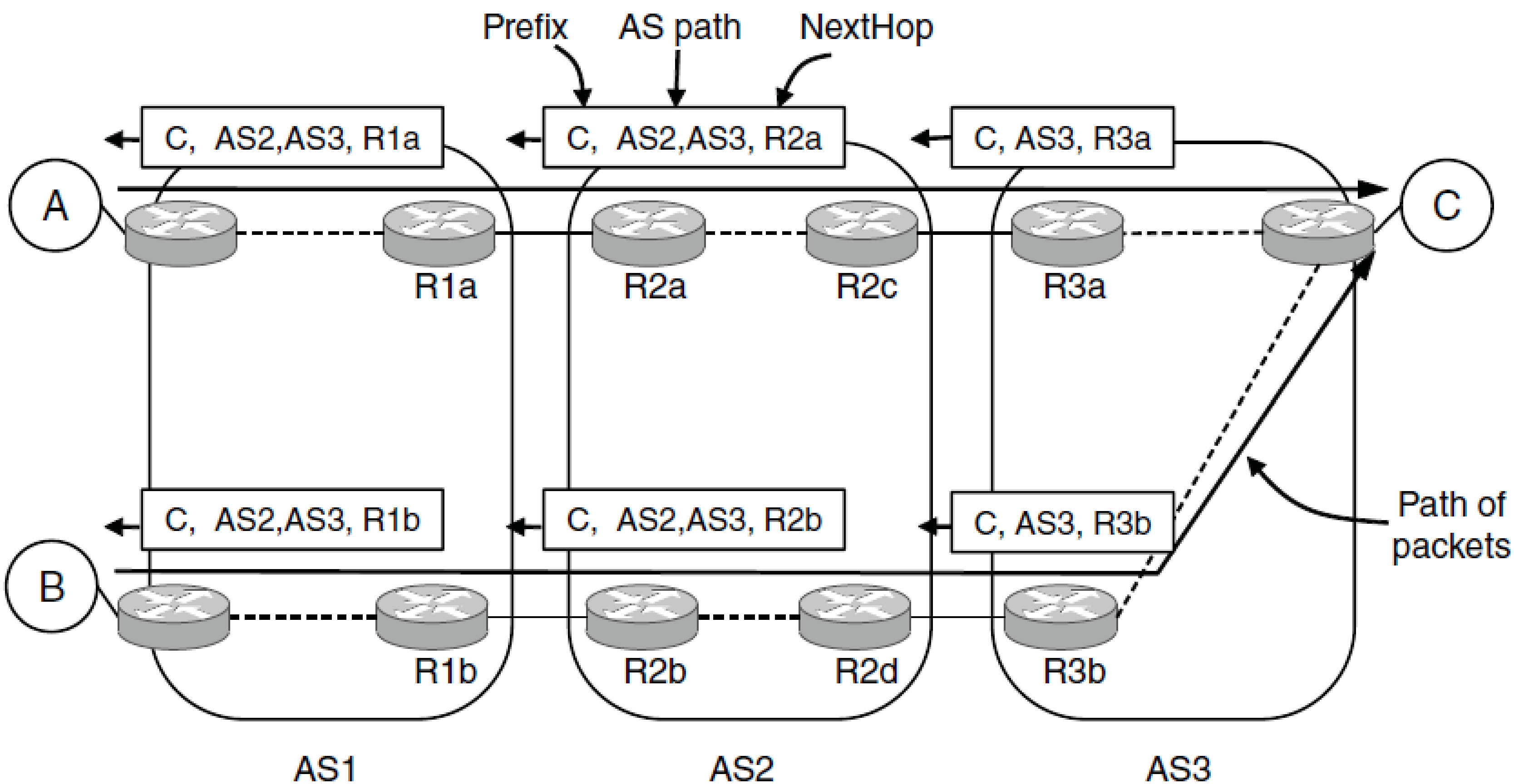
1. No commercial traffic for educat. network
2. Never put Iraq on route starting at Pentagon
3. Choose cheaper network
4. Choose better performing network
5. Don’t go from Apple to Google to Apple

BGP—The Exterior Gateway Routing Protocol (2)



Routing policies between four Autonomous Systems

BGP—The Exterior Gateway Routing Protocol (3)



Propagation of BGP route advertisements

Mobile IP

Goals

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

End

Chapter 5