5

THE NETWORK LAYER

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

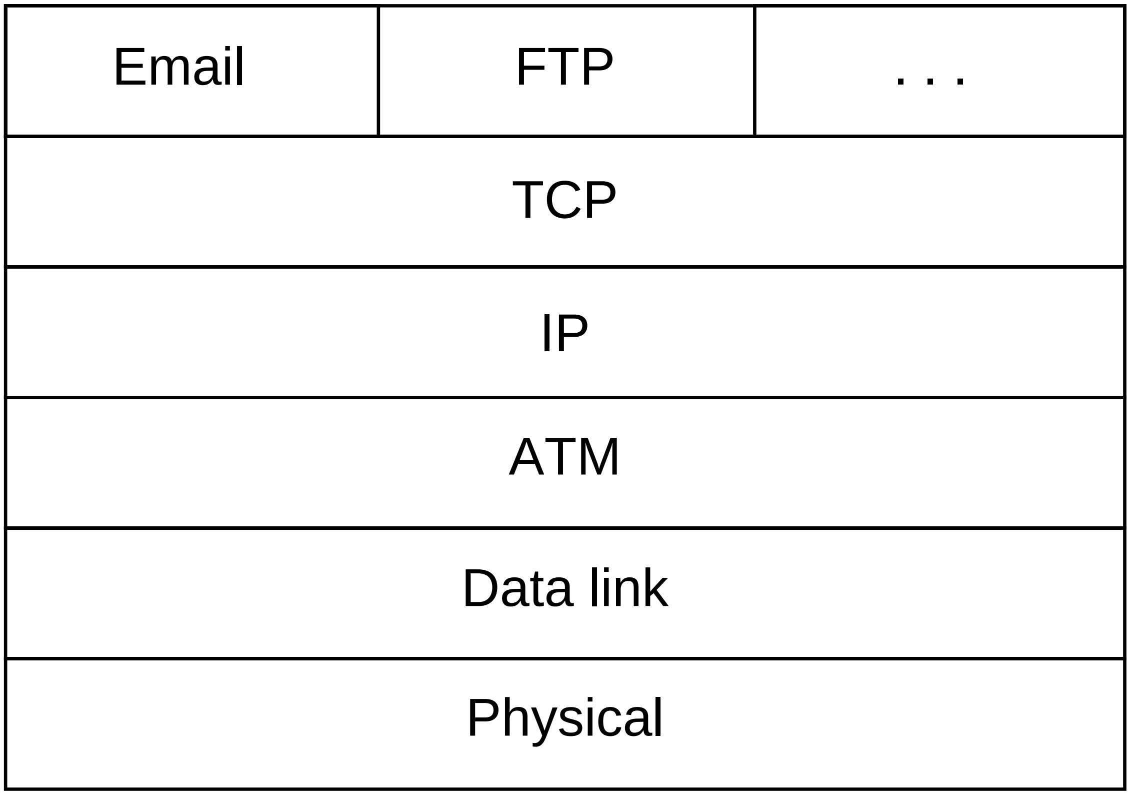


Fig. 5-1. Running TCP/IP over an ATM subnet.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

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

Fig. 5-2. Comparison of datagram and virtual circuit subnets.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

Upper layer	Type of subnet	
	Datagram	Virtual circuit
Connectionless	UDP over IP	UDP over IP over ATM
Connection-oriented	TCP over IP	ATM AAL1 over ATM

Fig. 5-3. Examples of different combinations of service and subnet structure.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

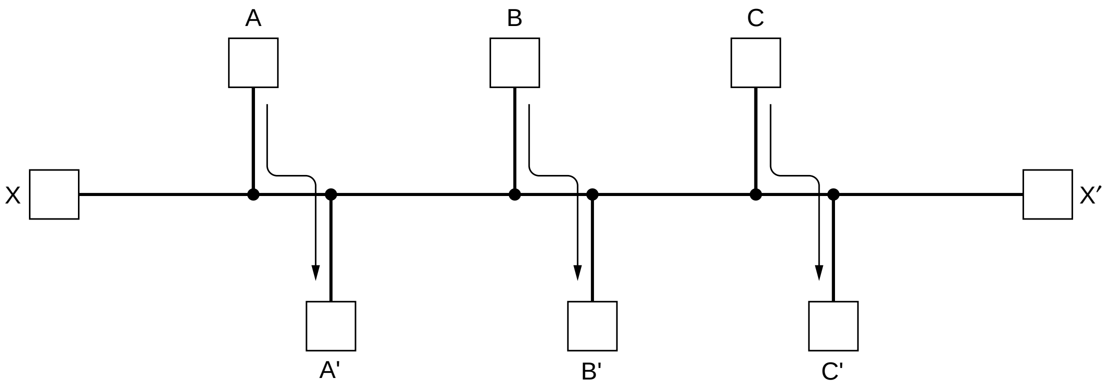


Fig. 5-4. Conflict between fairness and optimality.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

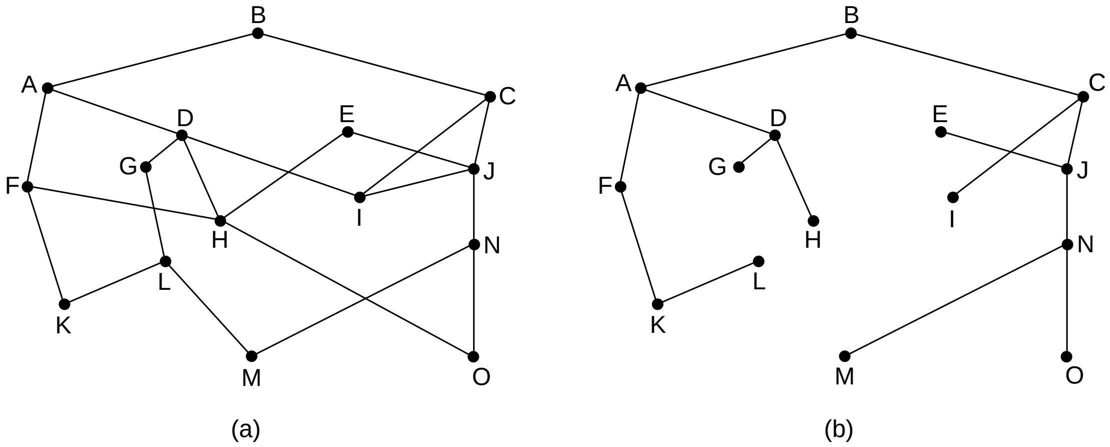


Fig. 5-5. (a) A subnet. (b) A sink tree for router *B*.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

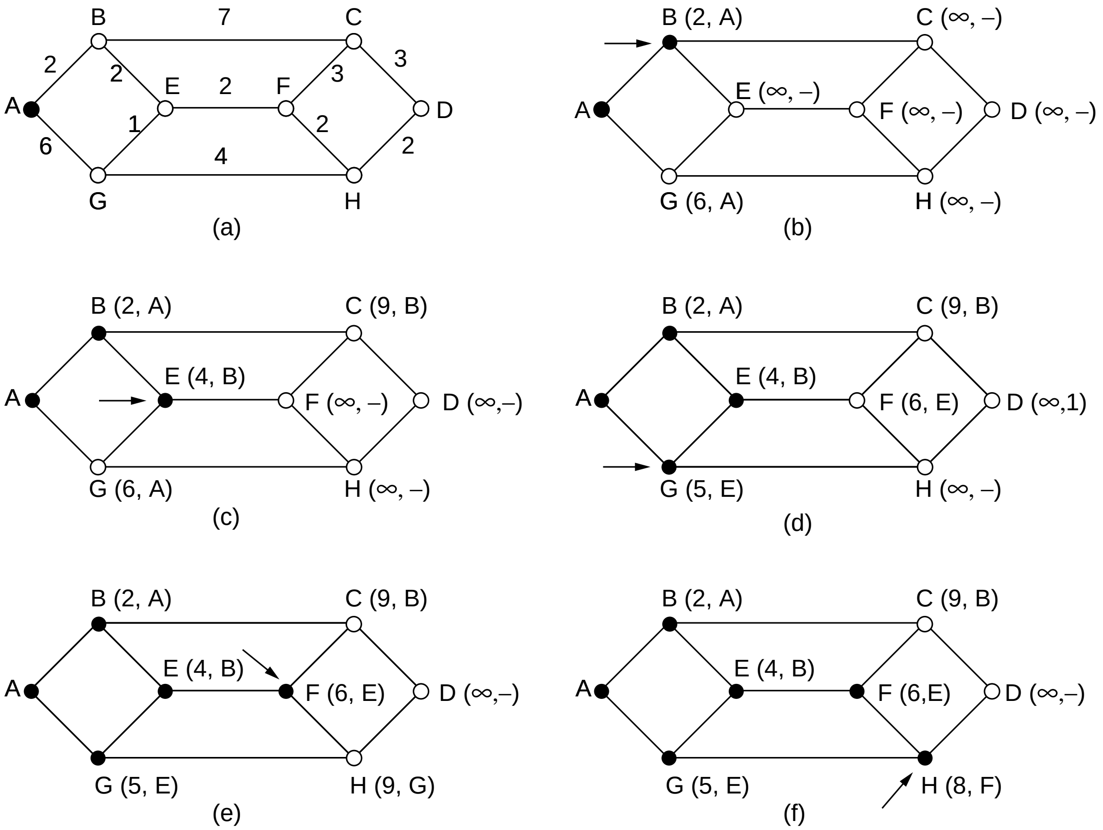


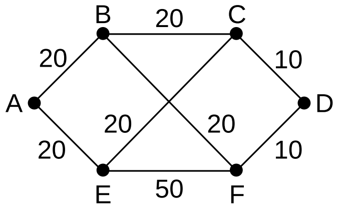
Fig. 5-6. The first five steps used in computing the shortest path from *A* to *D*. The arrows indicate the working node.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

```
#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 */
  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);
}
```

Fig. 5-7. Dijkstra’s algorithm to compute the shortest path.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall



(a)

		Destination					
		A	B	C	D	E	F
Source	A		9 AB	4 ABC	1 ABFD	7 AE	4 AEF
	B	9 BA		8 BC	3 BFD	2 BFE	4 BF
	C	4 CBA	8 CB		3 CD	3 CE	2 CEF
	D	1 DFBA	3 DFB	3 DC		3 DCE	4 DF
	E	7 EA	2 EFB	3 EC	3 ECD		5 EF
	F	4 FEA	4 FB	2 FEC	4 FD	5 FE	

(b)

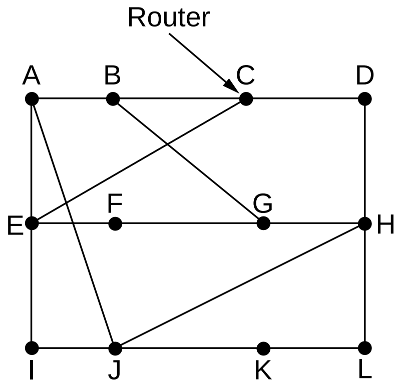
Fig. 5-8. (a) A subnet with line capacities shown in kbps. (b) The traffic in packets/sec and the routing matrix.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

i	Line	λ_i (pkts/sec)	C_i (kbps)	μC_i (pkts/sec)	T_i (msec)	Weight
1	AB	14	20	25	91	0.171
2	BC	12	20	25	77	0.146
3	CD	6	10	12.5	154	0.073
4	AE	11	20	25	71	0.134
5	EF	13	50	62.5	20	0.159
6	FD	8	10	12.5	222	0.098
7	BF	10	20	25	67	0.122
8	EC	8	20	25	59	0.098

Fig. 5-9. Analysis of the subnet of Fig. 5-0 using a mean packet size of 800 bits. The reverse traffic (*BA*, *CB*, etc.) is the same as the forward traffic.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall



(a)

New estimated delay from J

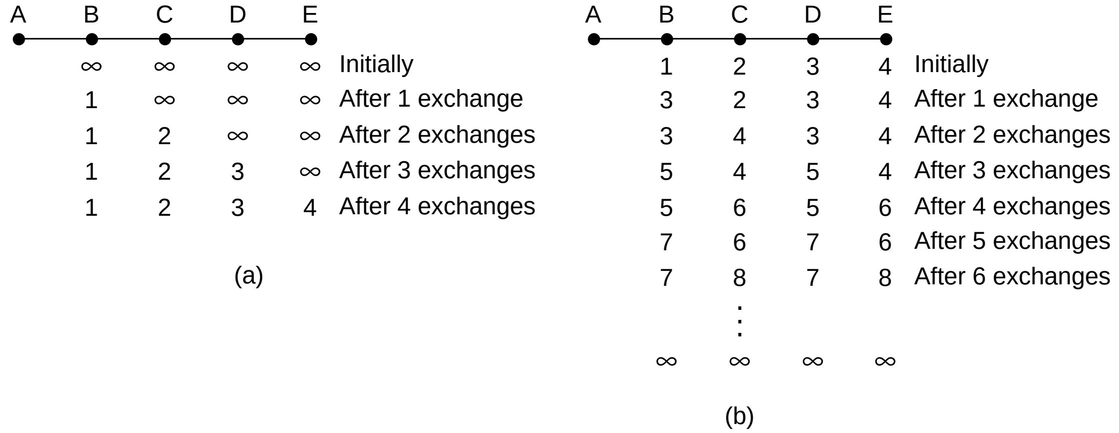
To	A	I	H	K	New estimated delay from J	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	–
K	24	22	22	0	6	K
L	29	33	9	9	15	K
	JA delay is 8	JI delay is 10	JH delay is 12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

Vectors received from J's four neighbors

(b)

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

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall



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

(a)

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

(b)

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

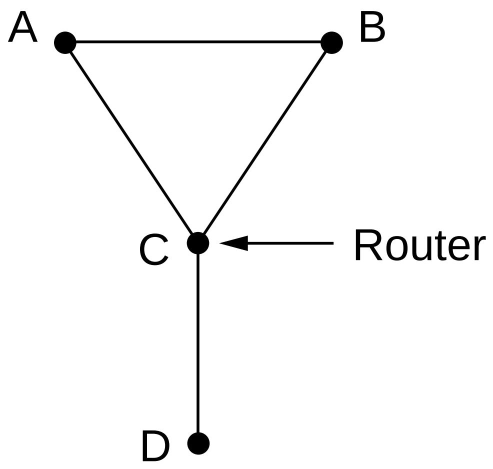


Fig. 5-12. An example where split horizon fails.

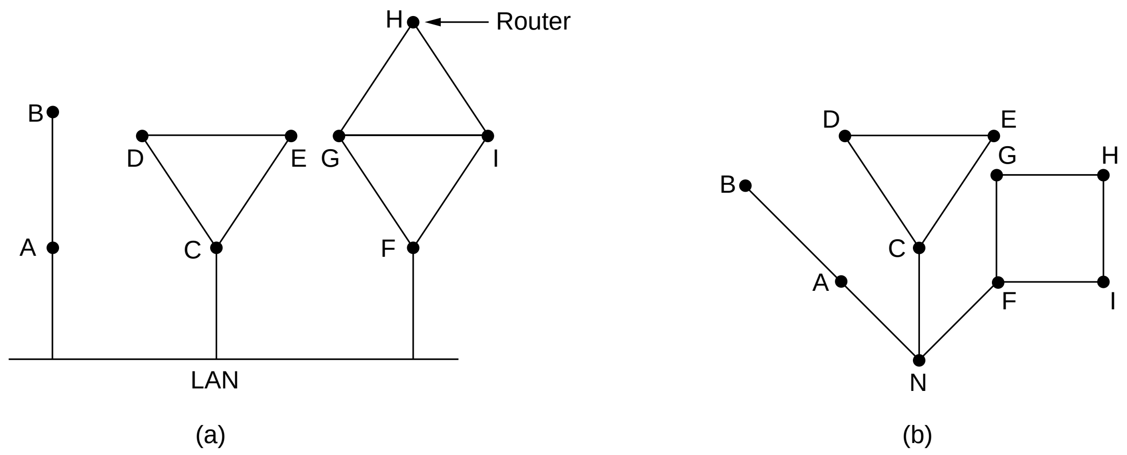


Fig. 5-13. (a) Nine routers and a LAN. (b) A graph model of (a).

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

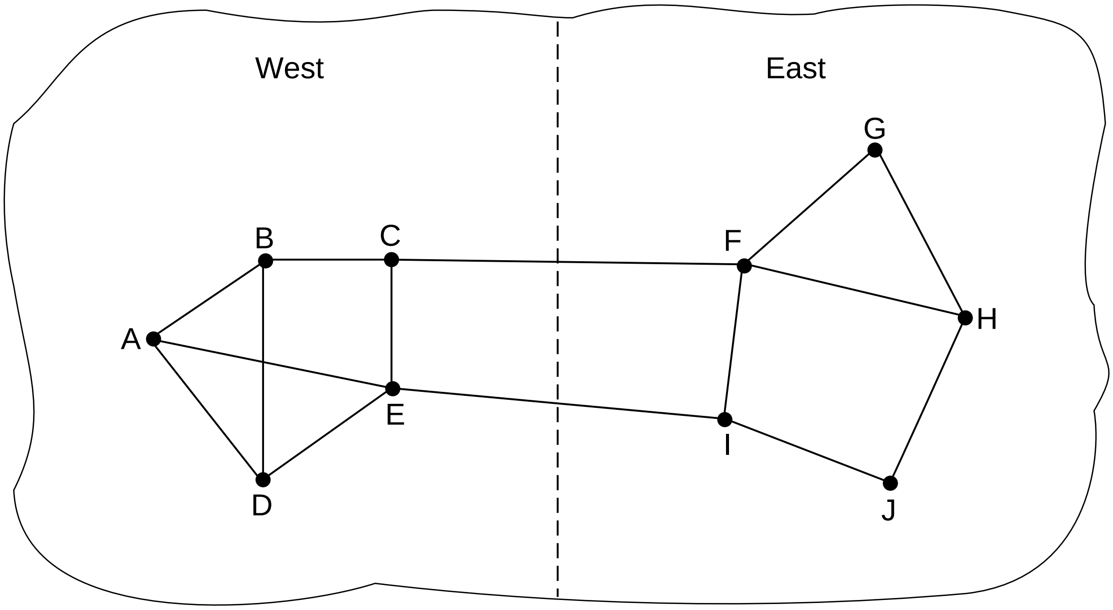
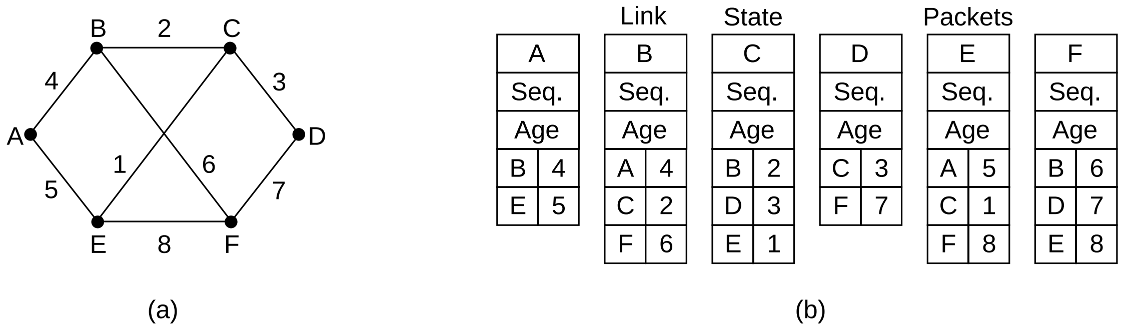


Fig. 5-14. A subnet in which the East and West parts are connected by two lines.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall



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

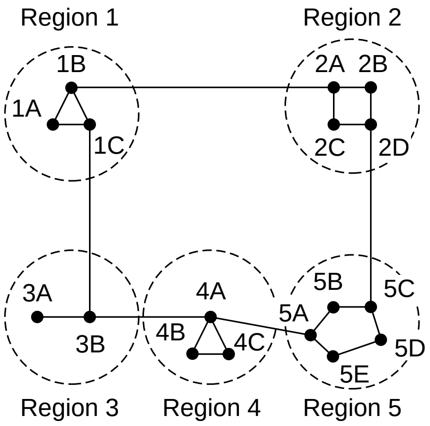
Fig. 5-15. (a) A subnet. (b) The link state packets for this subnet.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

			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	

Fig. 5-16. The packet buffer for router *B* in Fig. 5-15.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall



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)

Fig. 5-17. Hierarchical routing.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

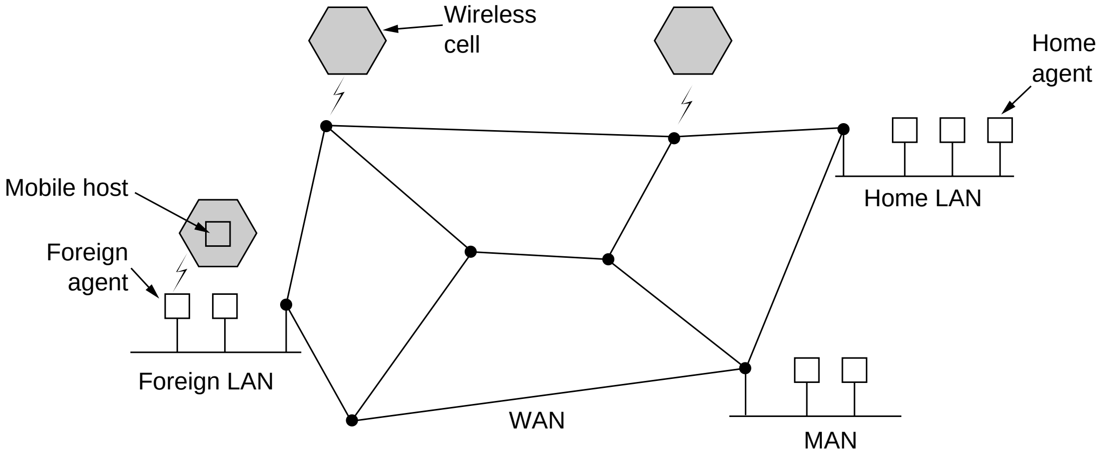


Fig. 5-18. A WAN to which LANs, MANs, and wireless cells are attached.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

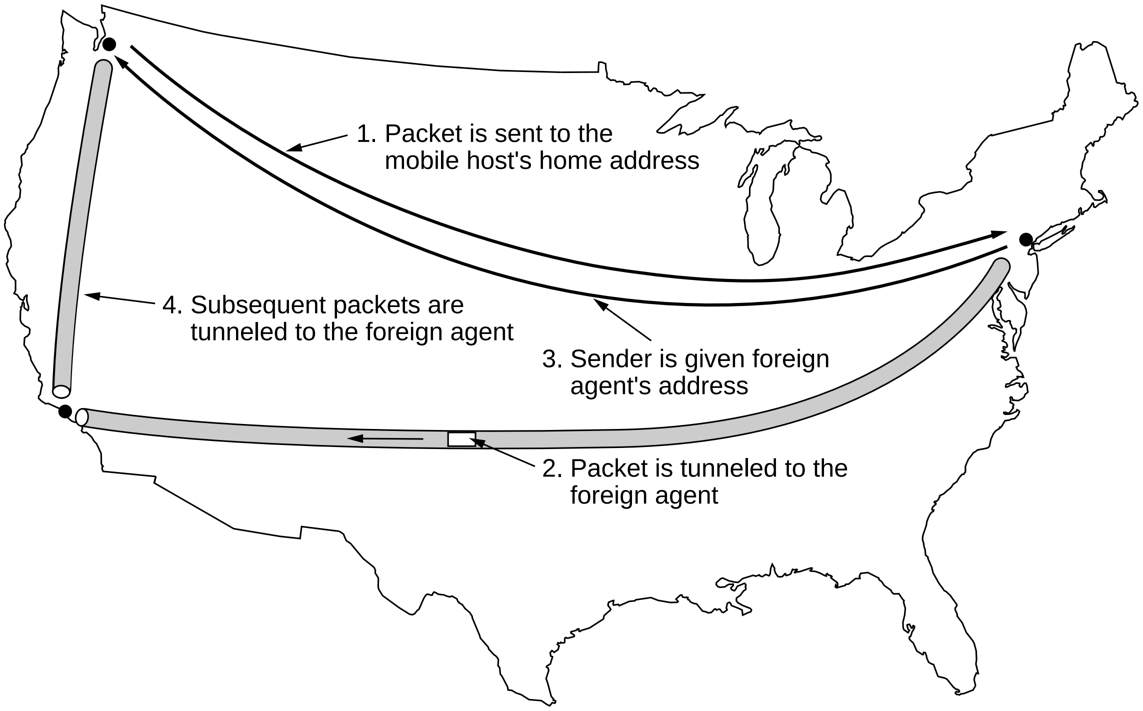

1. Packet is sent to the mobile host's home address
4. Subsequent packets are tunneled to the foreign agent
3. Sender is given foreign agent's address
2. Packet is tunneled to the foreign agent

Fig. 5-19. Packet routing for mobile users.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

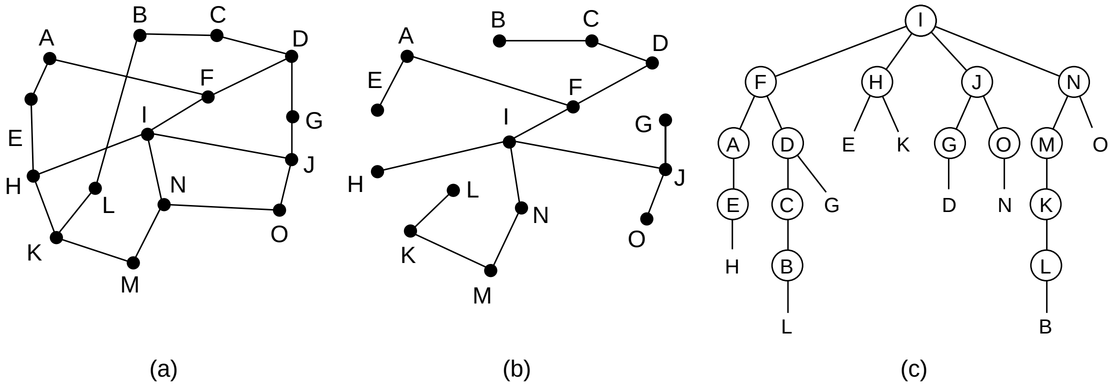


Fig. 5-20. Reverse path forwarding. (a) A subnet. (b) A spanning tree. (c) The tree built by reverse path forwarding.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

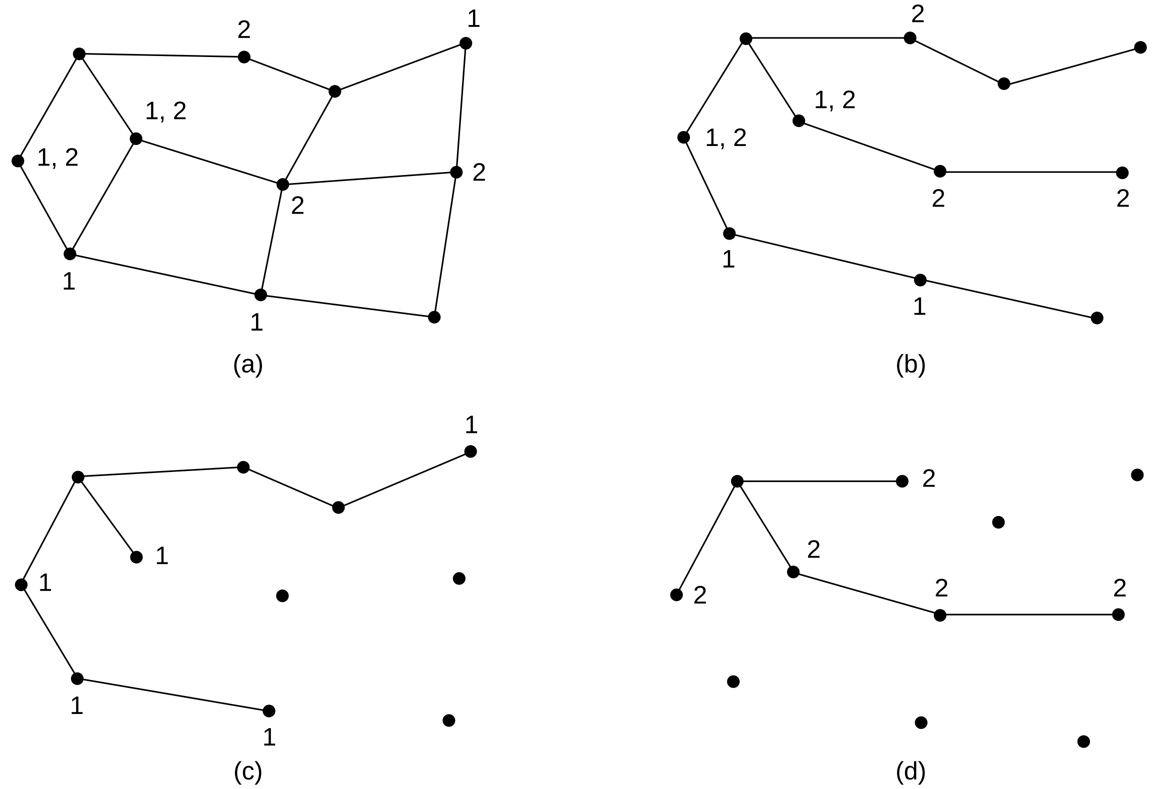


Fig. 5-21. (a) A subnet. (b) A spanning tree for the leftmost router. (c) A multicast tree for group 1. (d) A multicast tree for group 2.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

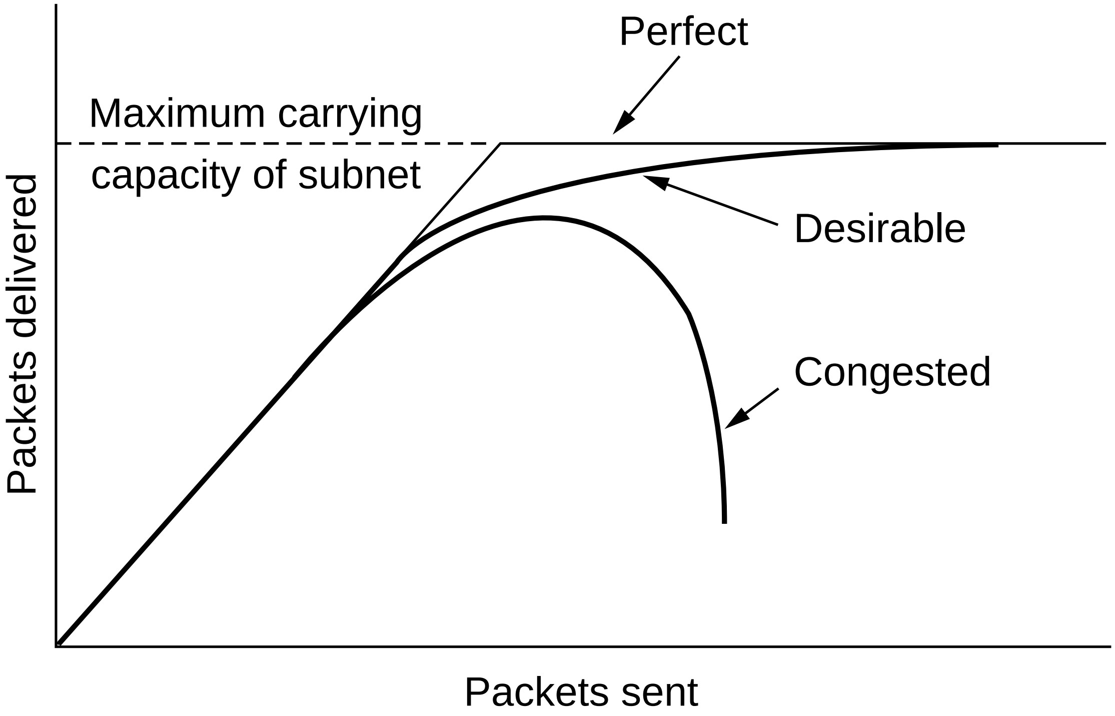


Fig. 5-22. When too much traffic is offered, congestion sets in and performance degrades sharply.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

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

Fig. 5-23. Policies that affect congestion.

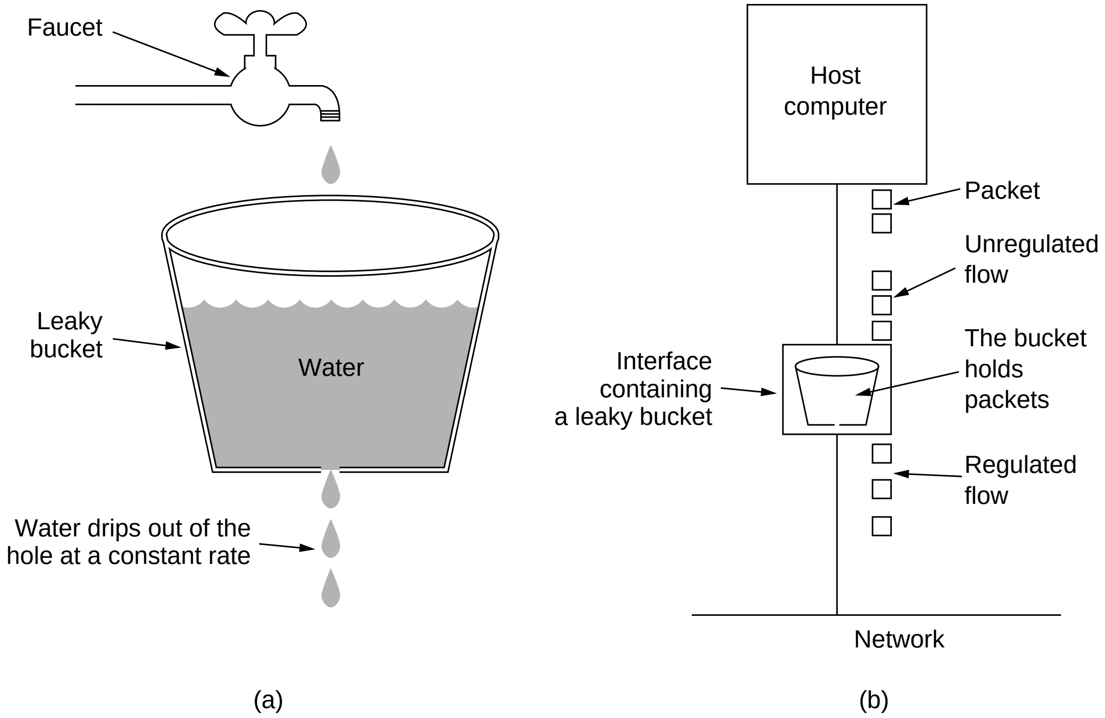


Fig. 5-24. (a) A leaky bucket with water. (b) A leaky bucket with packets.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

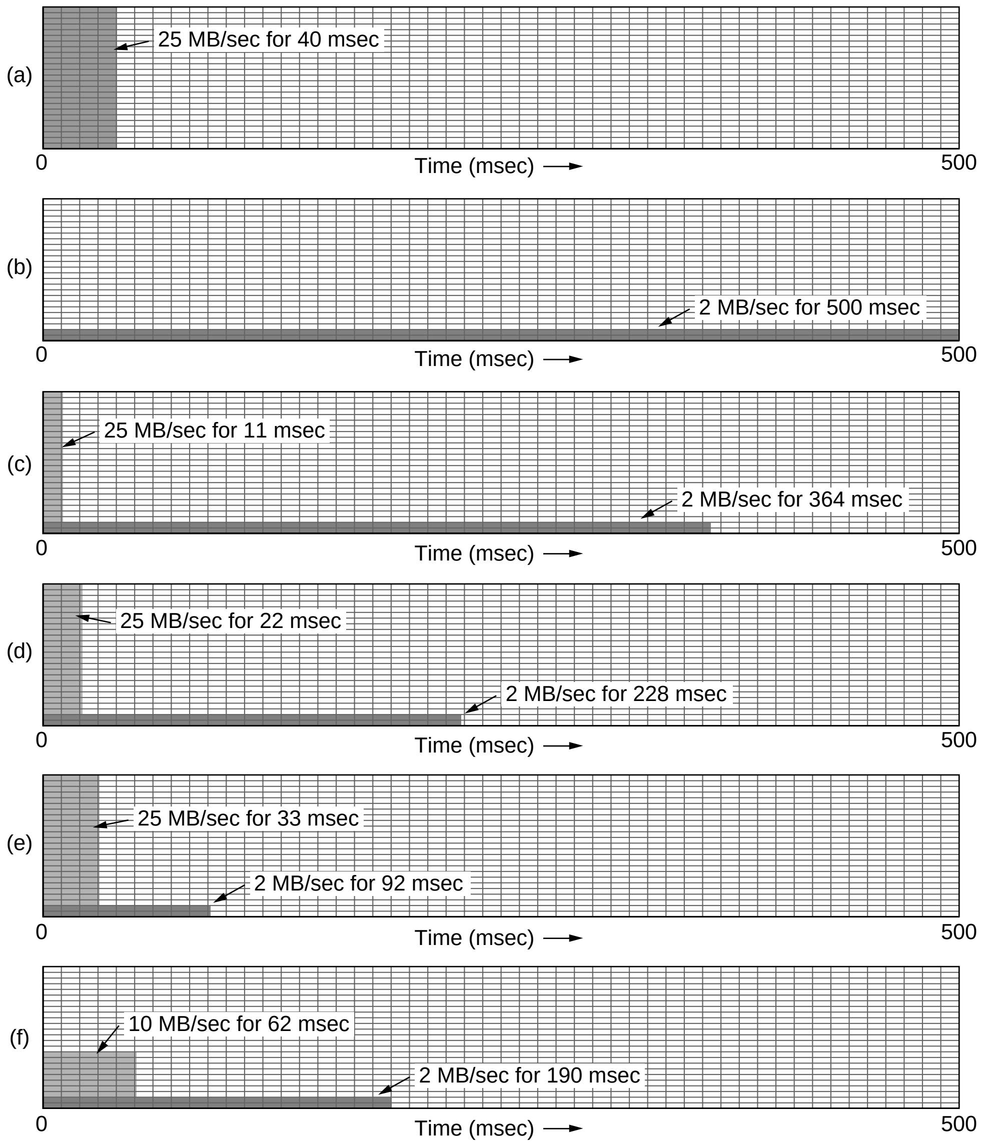


Fig. 5-25. (a) Input to a leaky bucket. (b) Output from a leaky bucket. (c) - (e) Output from a token bucket with capacities of 250KB, 500KB, and 750KB. (f) Output from a 500KB token bucket feeding a 10 MB/sec leaky bucket.

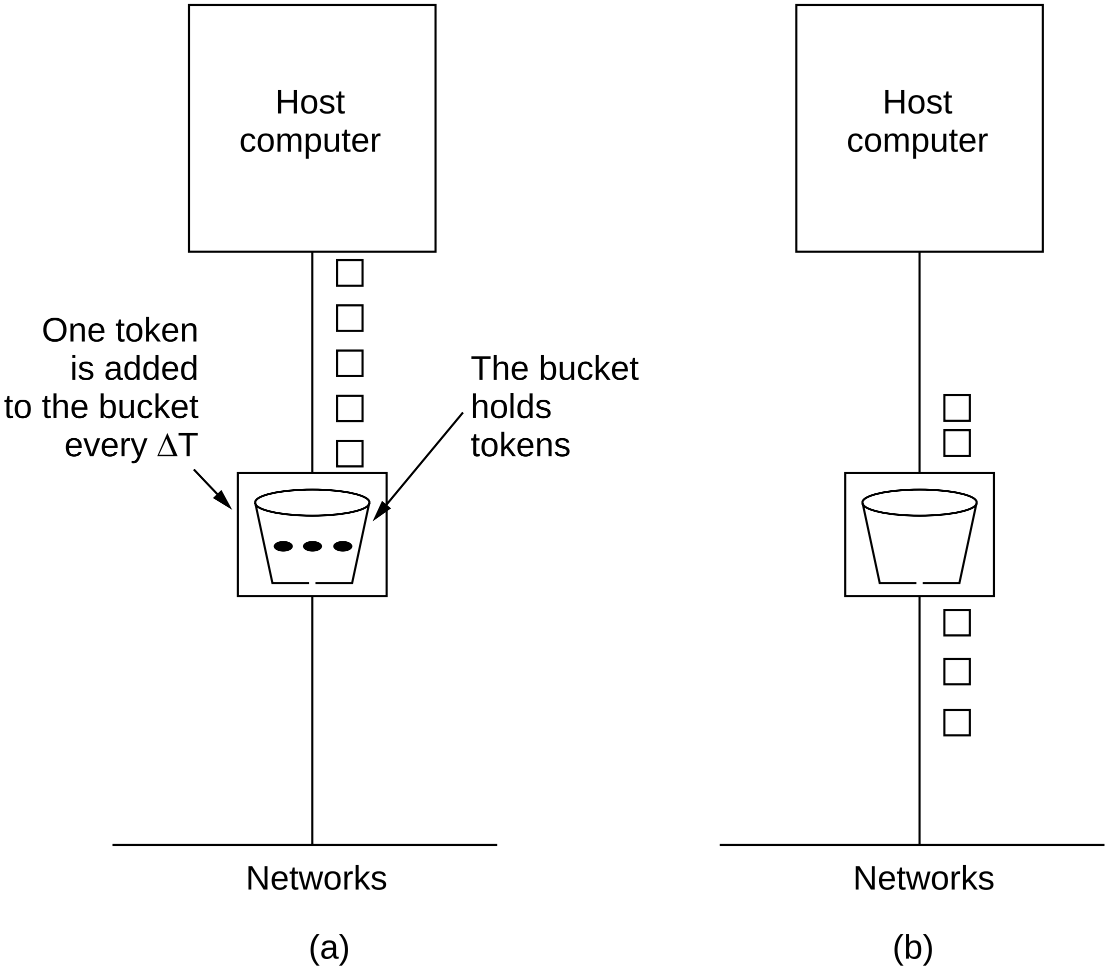


Fig. 5-26. The token bucket algorithm. (a) Before. (b) After.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

Characteristics of the Input	**Service Desired**
Maximum packet size (bytes)	Loss sensitivity (bytes)
Token bucket rate (bytes/sec)	Loss interval (μsec)
Token bucket size (bytes)	Burst loss sensitivity (packets)
Maximum transmission rate (bytes/sec)	Minimum delay noticed (μsec)
	Maximum delay variation (μsec)
	Quality of guarantee

Fig. 5-27. An example flow specification.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

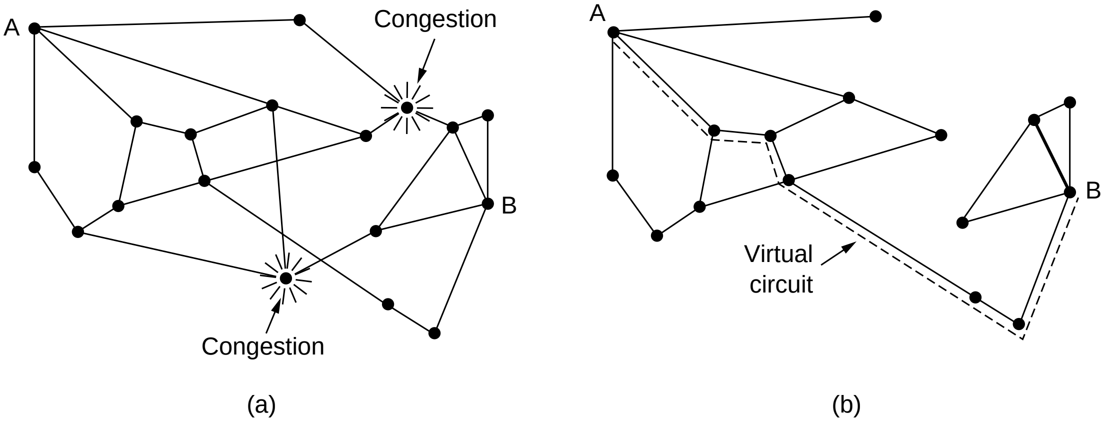
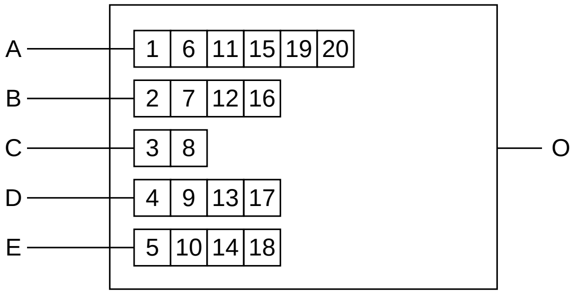


Fig. 5-28. (a) A congested subnet. (b) A redrawn subnet that eliminates the congestion and a virtual circuit from *A* to *B*.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall



(a)

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

(b)

Fig. 5-29. (a) A router with five packets queued for line *O*. (b) Finishing times for the five packets.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

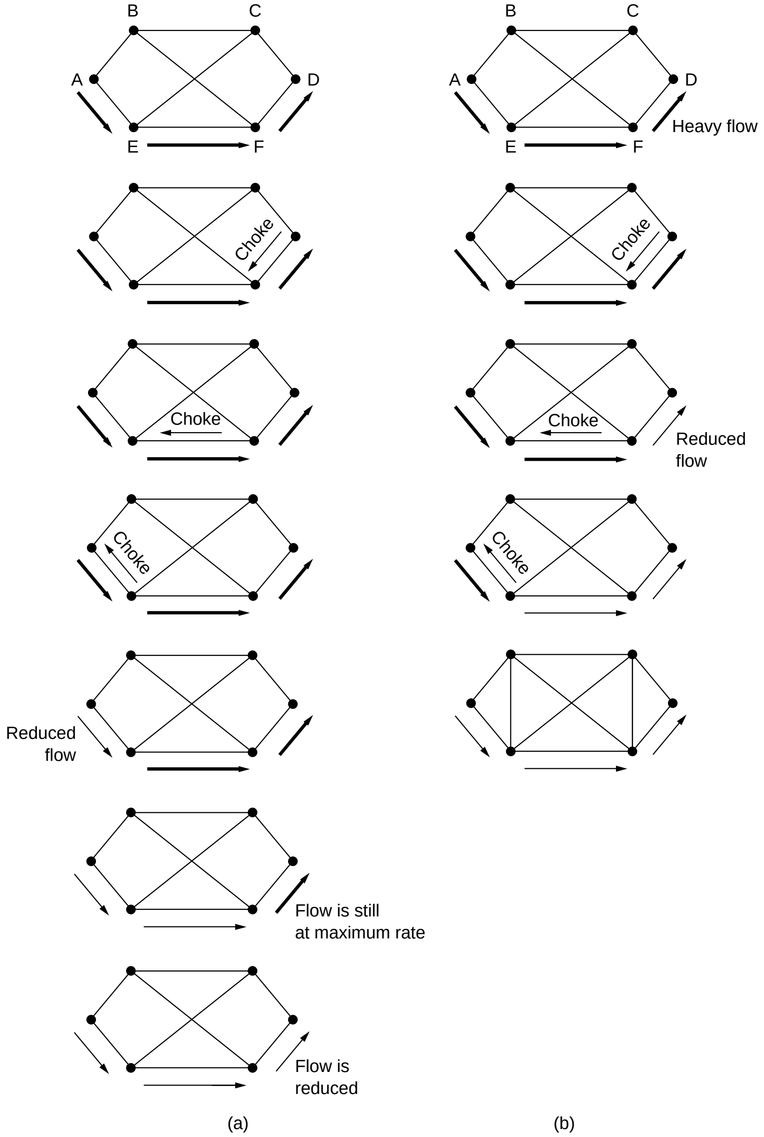


Fig. 5-30. (a) A choke packet that affects only the source. (b) A choke packet that affects each hop it passes through.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

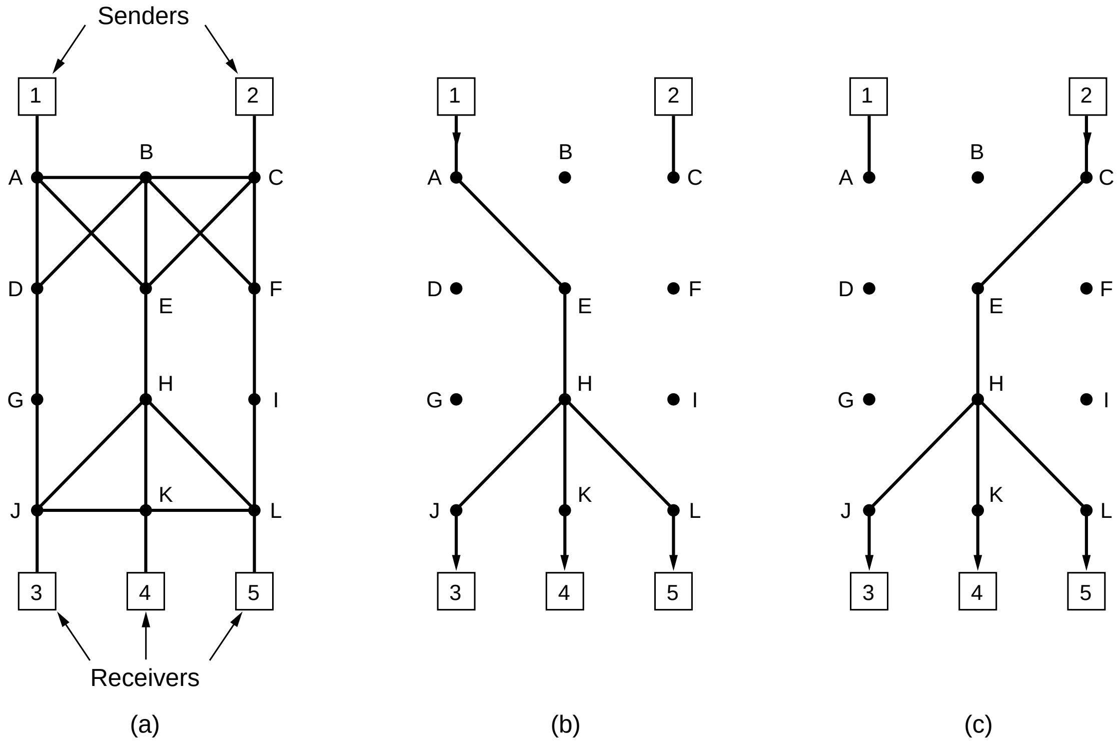


Fig. 5-31. (a) A network. (b) The multicast spanning tree for host 1. (c) The multicast spanning tree for host 2.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

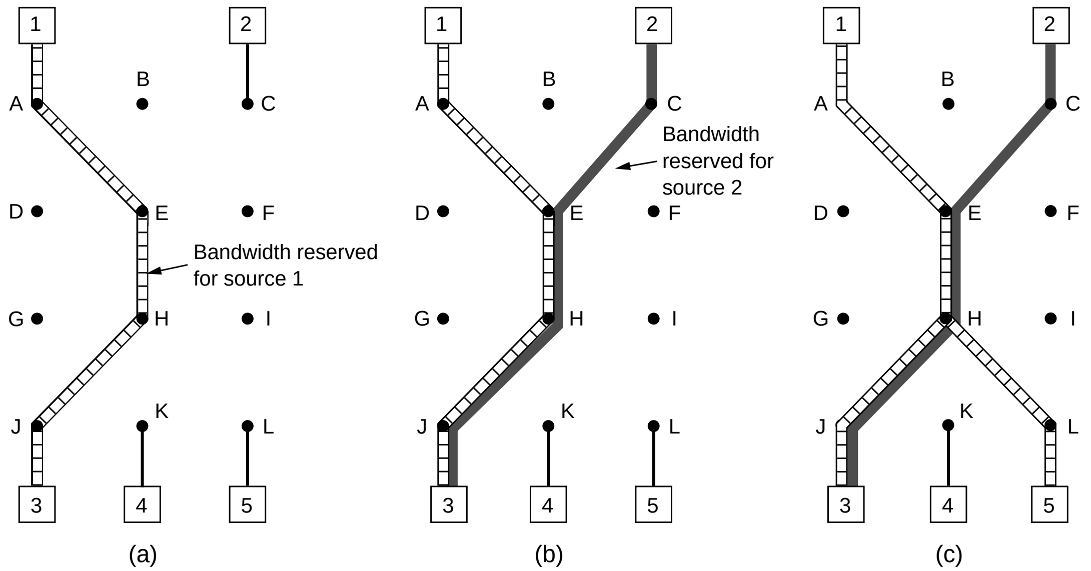

1
2
B
A
C
D
E
F
Bandwidth reserved for source 1
G
H
I
J
K
L
3
4
5
(a)
1
2
B
A
C
Bandwidth reserved for source 2
D
E
F
G
H
I
J
K
L
3
4
5
(b)
1
2
B
A
C
D
E
F
G
H
I
J
K
L
3
4
5
(c)

Fig. 5-32. (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.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

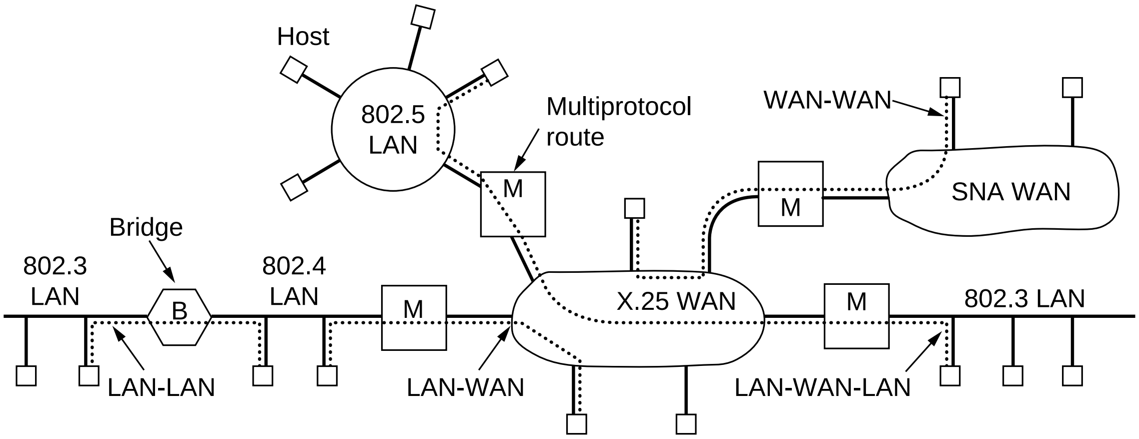


Fig. 5-33. Network interconnection.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

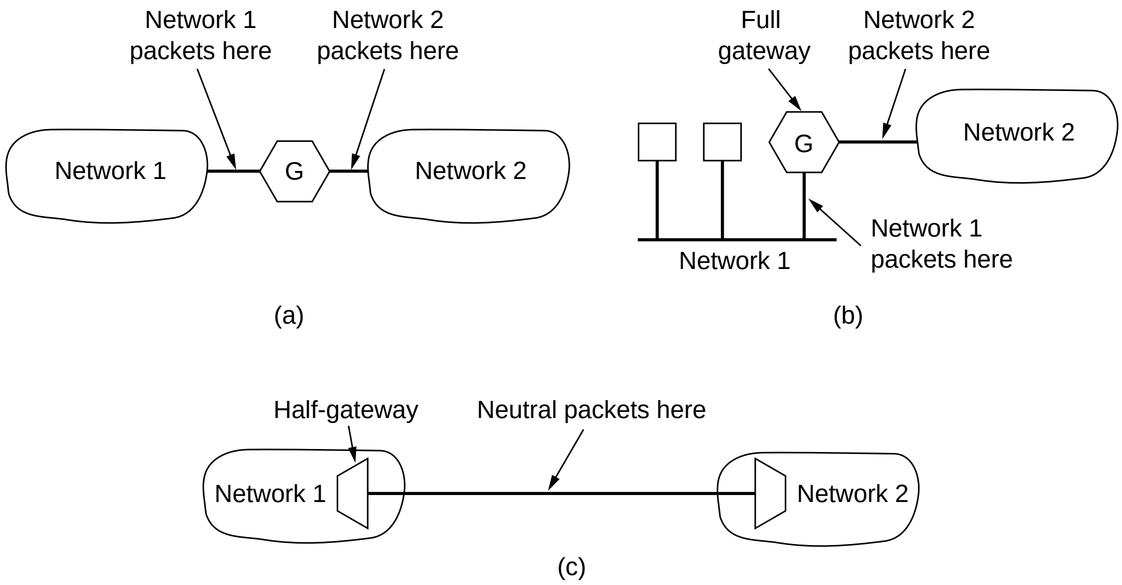


Fig. 5-34. (a) A full gateway between two WANs. (b) A full gateway between a LAN and a WAN. (c) Two half-gateways.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

Item	Some Possibilities
Service offered	Connection-oriented versus connectionless
Protocols	IP, IPX, CLNP, AppleTalk, DECnet, etc.
Addressing	Flat (802) versus hierarchical (IP)
Multicasting	Present or absent (also broadcasting)
Packet size	Every network has its own maximum
Quality of service	May be 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, 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

Fig. 5-35. Some of the many ways networks can differ.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

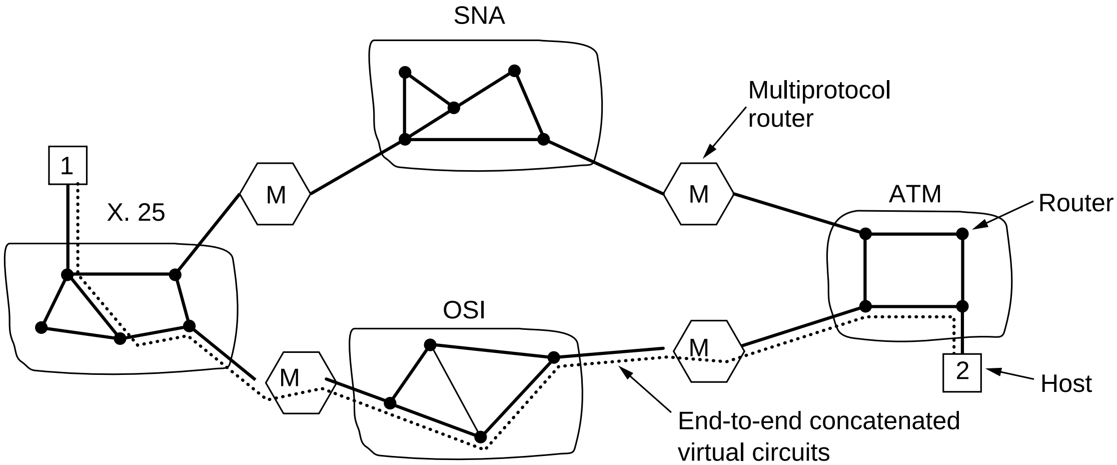


Fig. 5-36. Internetworking using concatenated virtual circuits.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

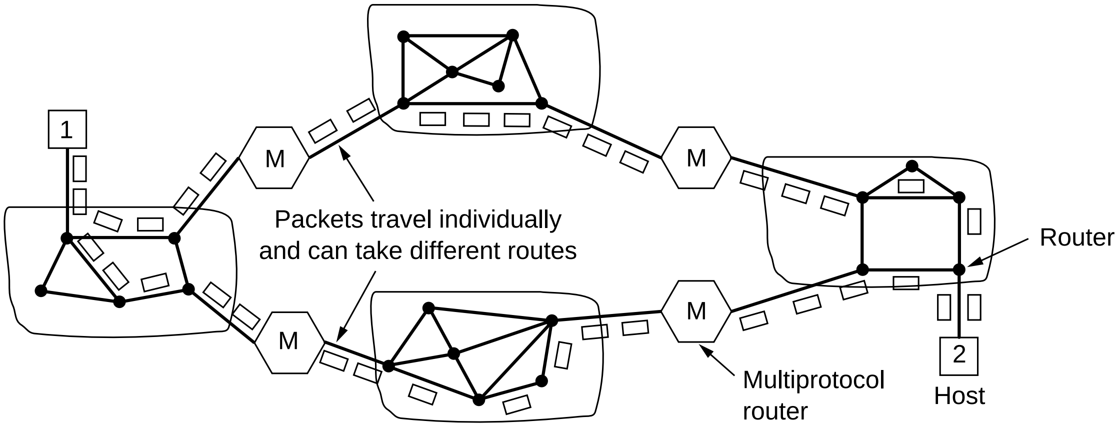


Fig. 5-37. A connectionless internet.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

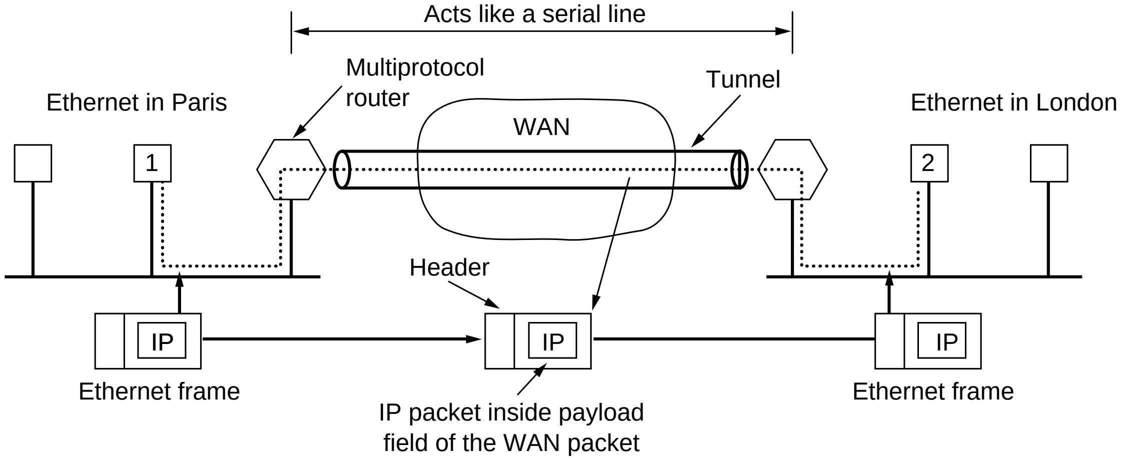


Fig. 5-38. Tunneling a packet from Paris to London.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

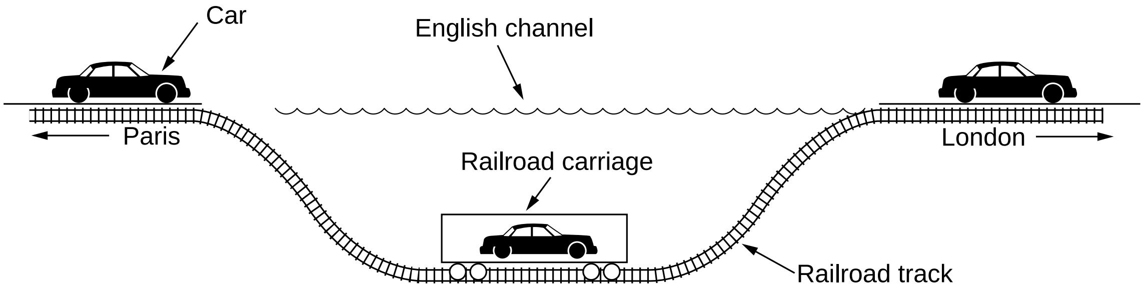


Fig. 5-39. Tunneling a car from France to England.

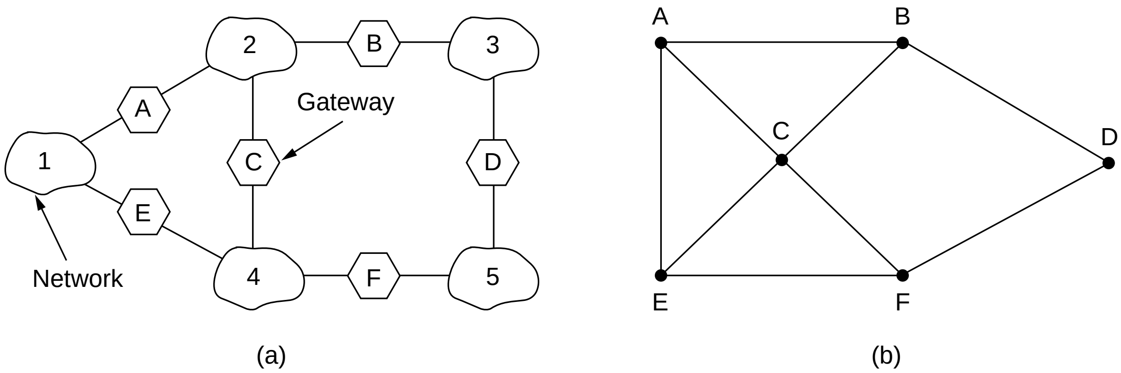


Fig. 5-40. (a) An internetwork. (b) A graph of the internetwork.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

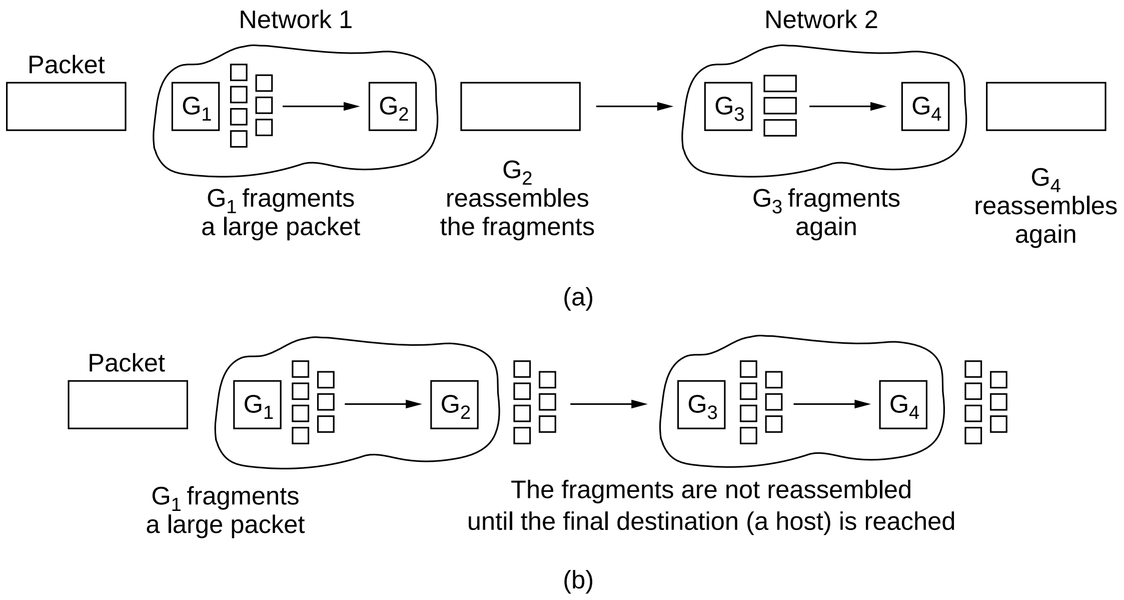

Network 1
Network 2
Packet
G_1
G_2
G_3
G_4
G_1 fragments a large packet
G_2 reassembles the fragments
G_3 fragments again
G_4 reassembles again
(a)
Packet
G_1
G_2
G_3
G_4
G_1 fragments a large packet
The fragments are not reassembled until the final destination (a host) is reached
(b)

Fig. 5-41. (a) Transparent fragmentation. (b) Nontransparent fragmentation.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

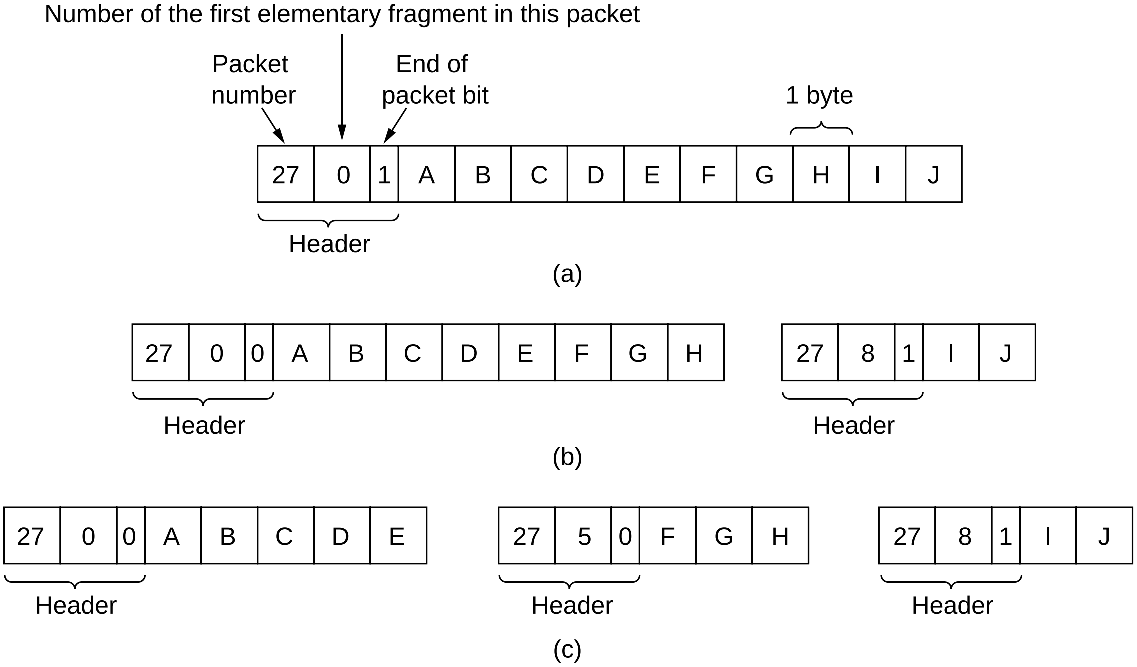


Fig. 5-42. Fragmentation when the elementary data size is 1 byte. (a) Original packet, containing 10 data bytes. (b) Fragments after passing through a network with maximum packet size of 8 bytes. (c) Fragments after passing through a size 5 gateway.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

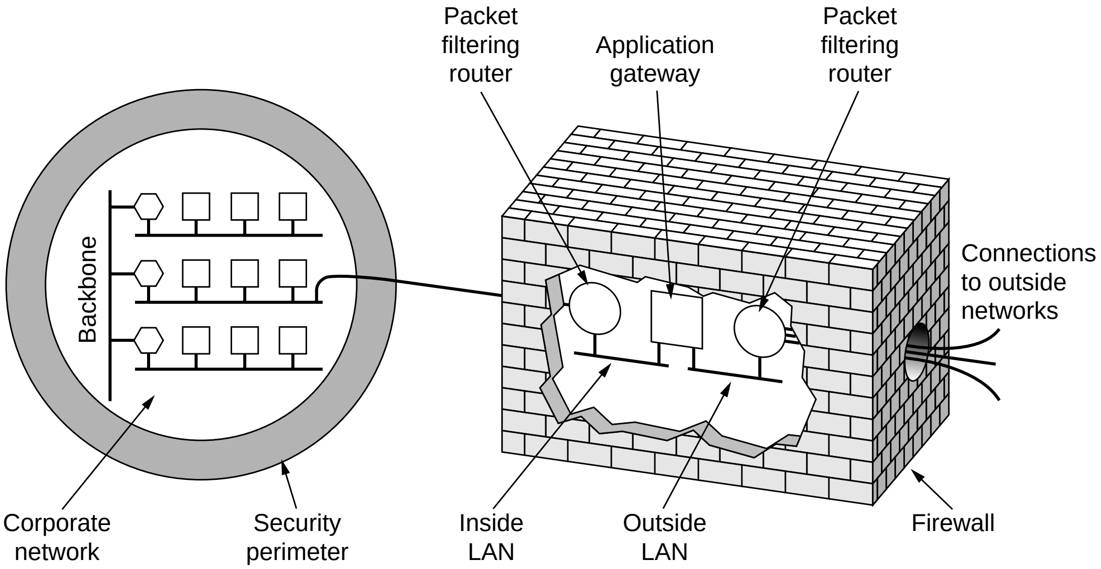


Fig. 5-43. A firewall consisting of two packet filters and an application gateway.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

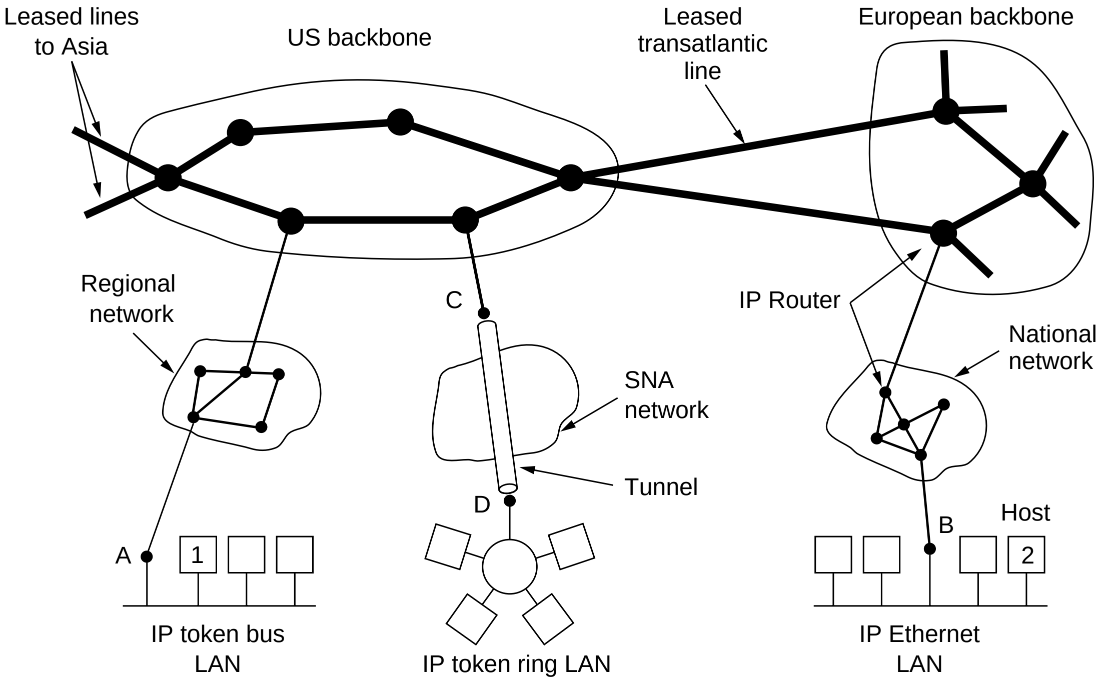


Fig. 5-44. The Internet is an interconnected collection of many networks.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

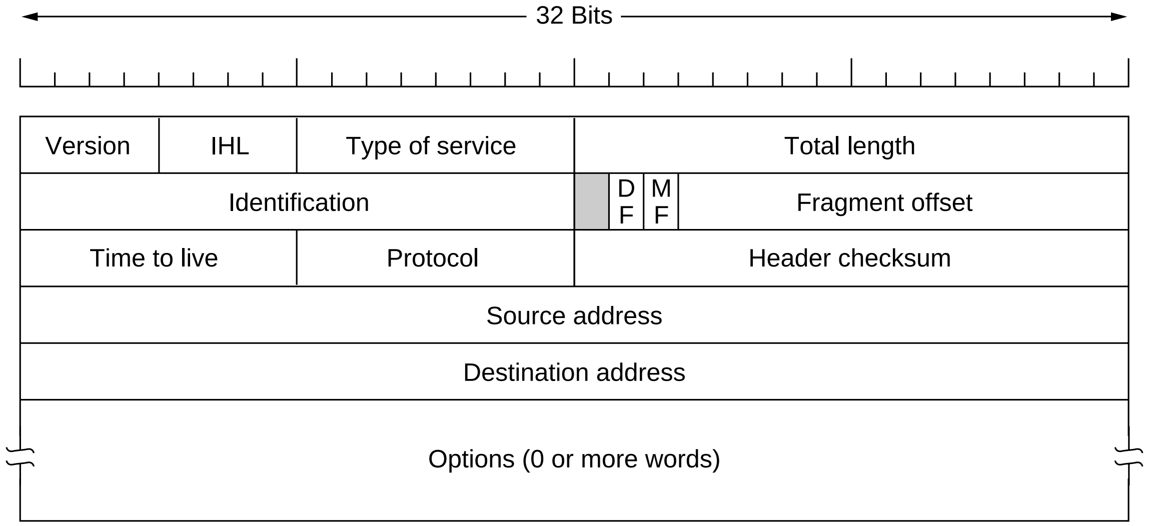


Fig. 5-45. The IP (Internet Protocol) header.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

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

Fig. 5-46. IP options.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

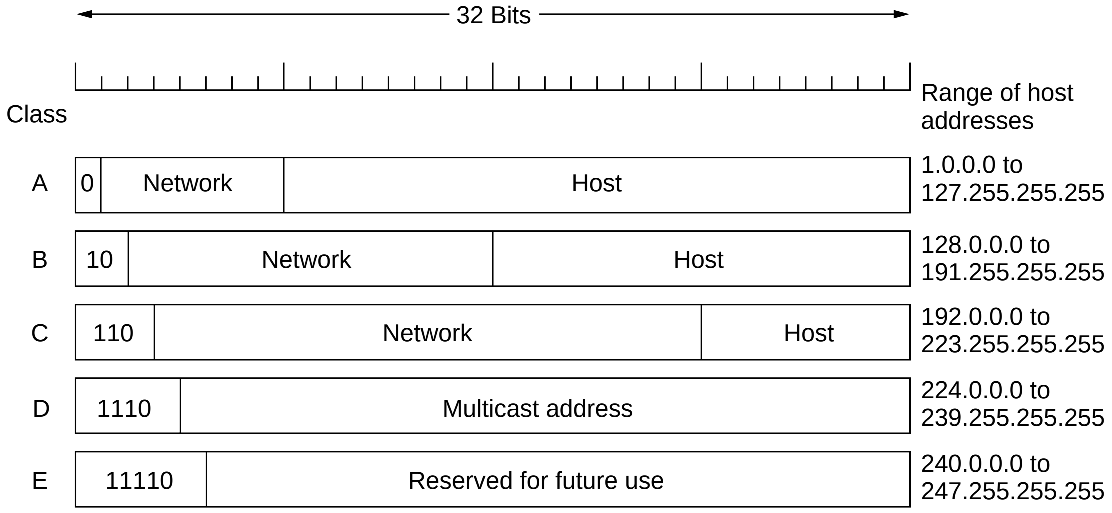
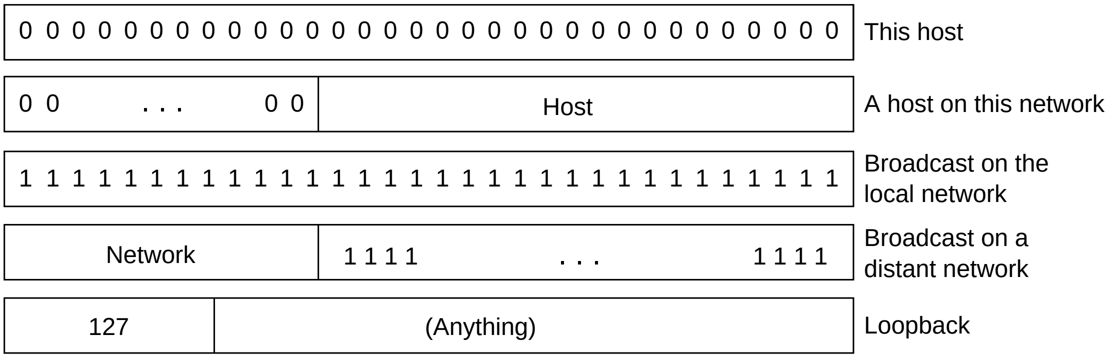

32 Bits
Class
Range of host addresses
A
0
Network
Host
1.0.0.0 to 127.255.255.255
B
10
Network
Host
128.0.0.0 to 191.255.255.255
C
110
Network
Host
192.0.0.0 to 223.255.255.255
D
1110
Multicast address
224.0.0.0 to 239.255.255.255
E
11110
Reserved for future use
240.0.0.0 to 247.255.255.255

Fig. 5-47. IP address formats.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall



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

Fig. 5-48. Special IP addresses.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

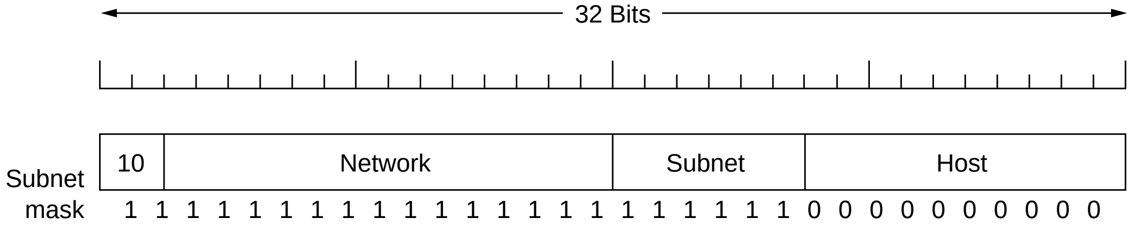


Fig. 5-49. One of the ways to subnet a class B network.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

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

Fig. 5-50. The principal ICMP message types.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

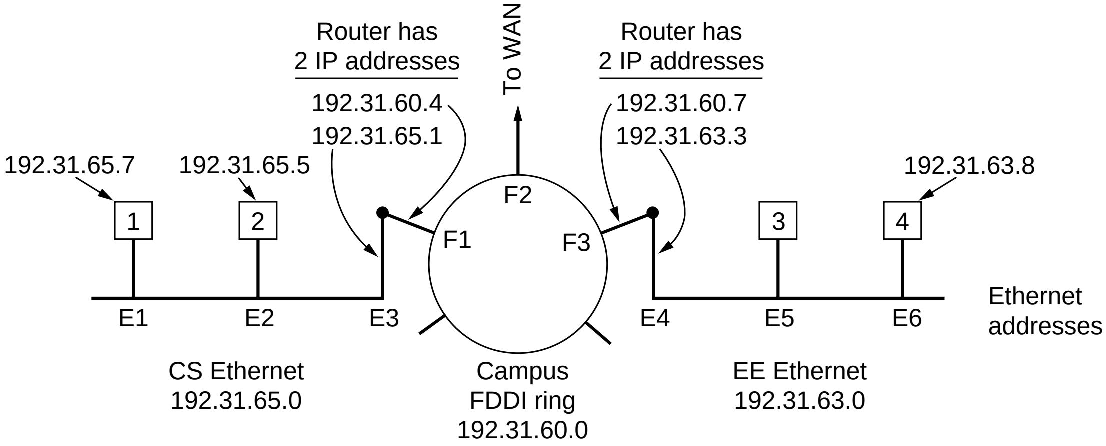
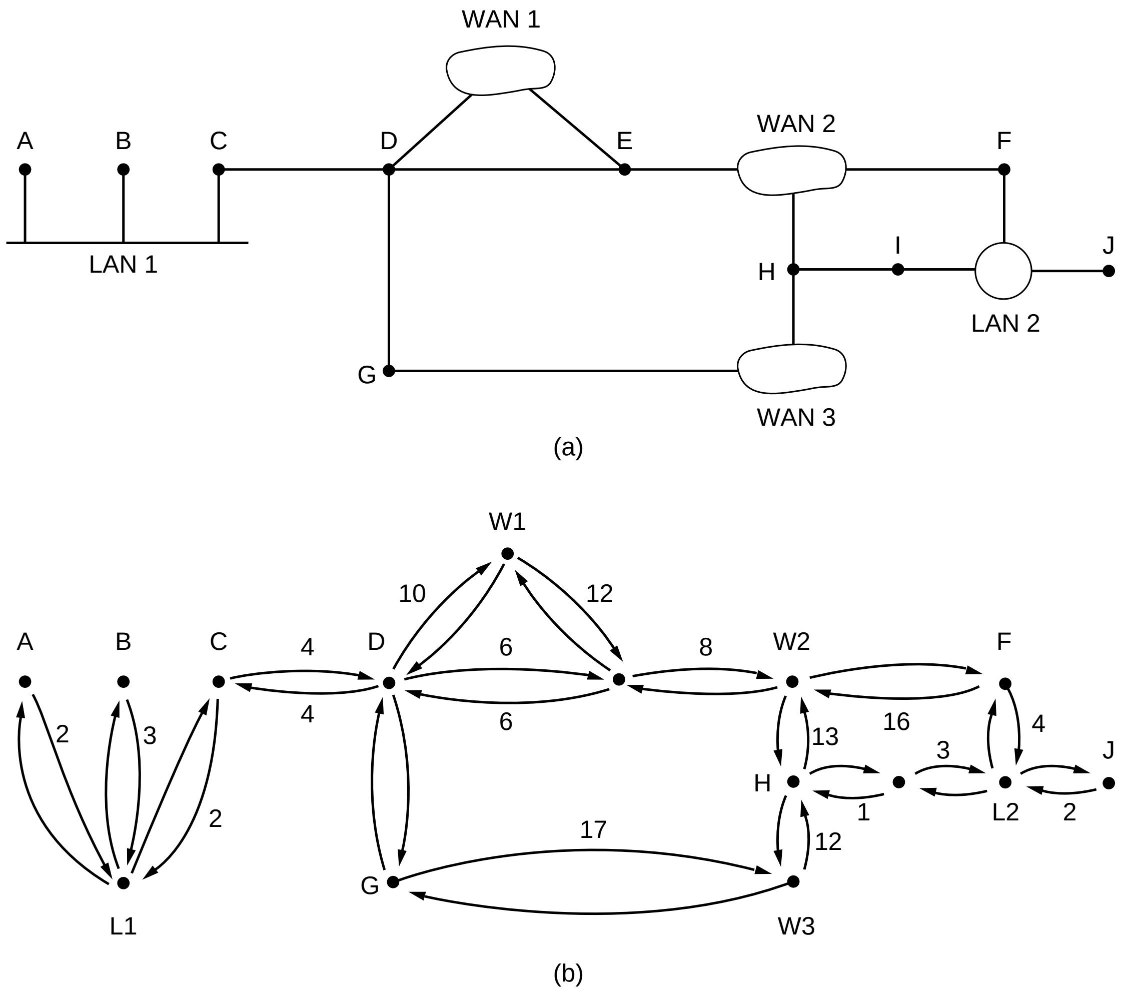


Fig. 5-51. Three interconnected class C networks: two Ethernets and an FDDI ring.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall



(a)

(b)

Fig. 5-52. (a) An autonomous system. (b) A graph representation of (a).

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

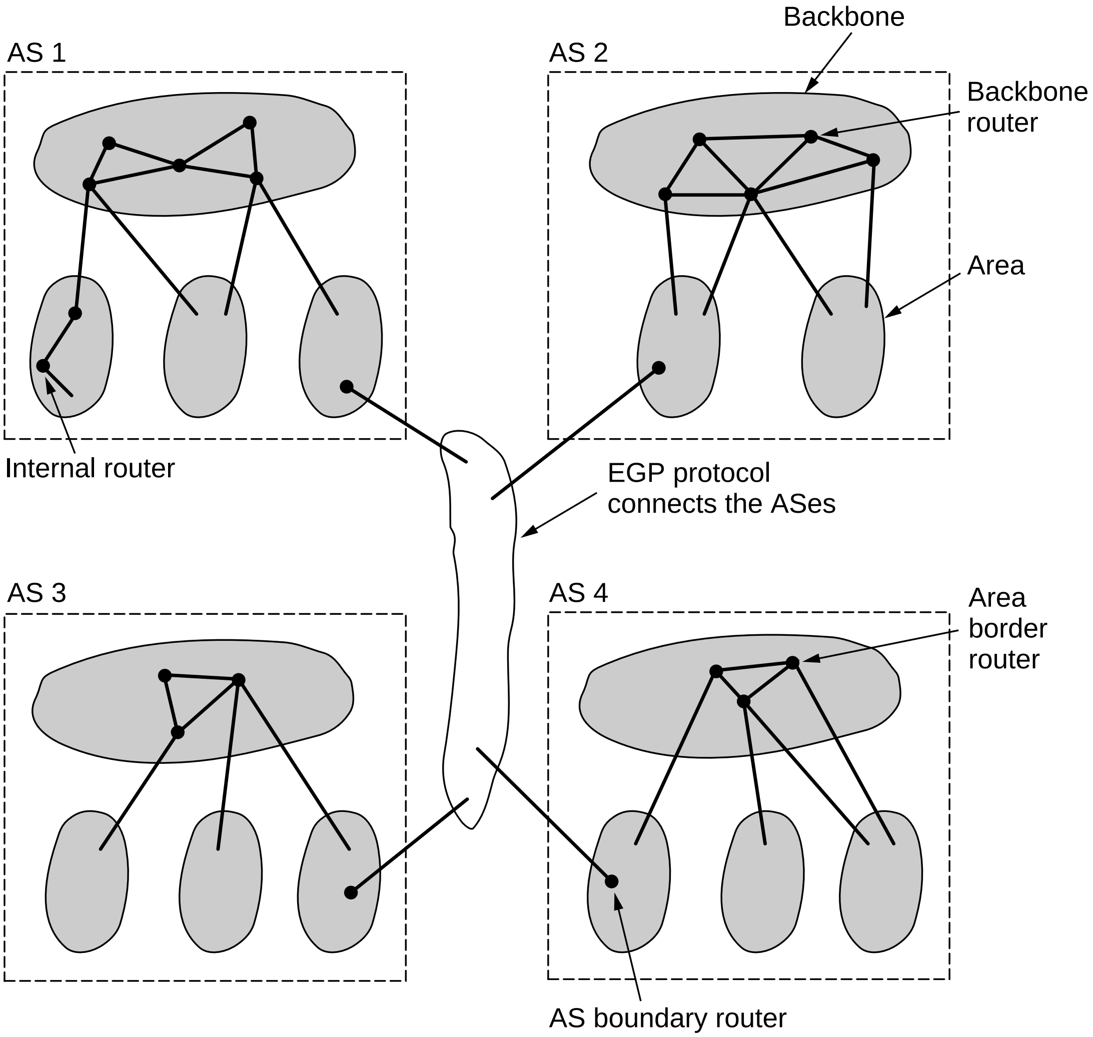


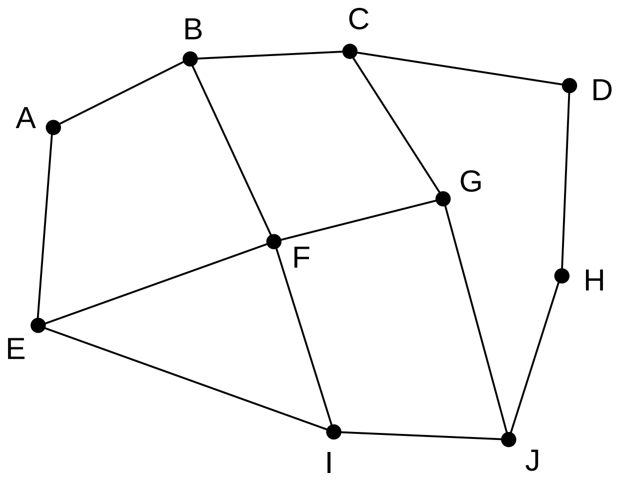
Fig. 5-53. The relation between ASes, backbones, and areas in OSPF.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

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

Fig. 5-54. The five types of OSPF messages.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall



(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)

Fig. 5-55. (a) A set of BGP routers. (b) Information sent to *F*.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

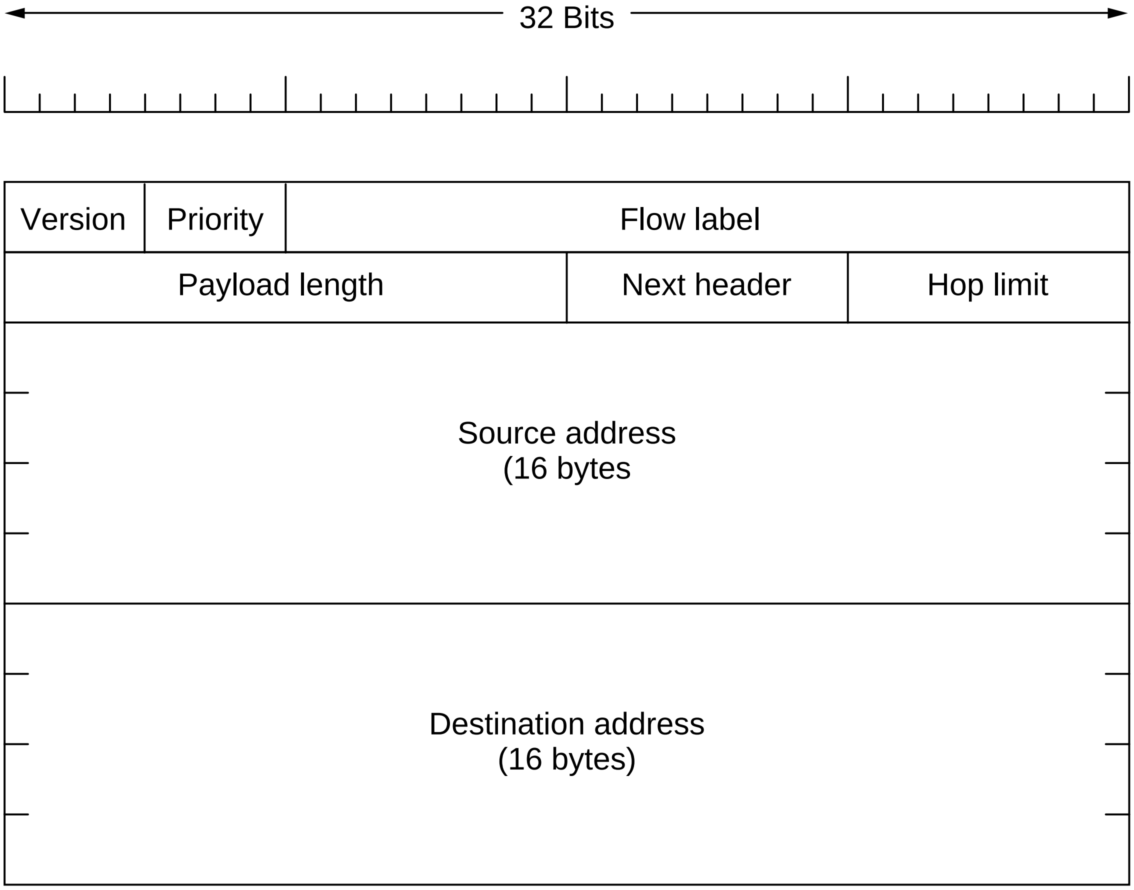


Fig. 5-56. The IPv6 fixed header (required).

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

Prefix (binary)	Usage	Fraction
0000 0000	Reserved (including IPv4)	1/256
0000 0001	Unassigned	1/256
0000 001	OSI NSAP addresses	1/128
0000 010	Novell NetWare IPX addresses	1/128
0000 011	Unassigned	1/128
0000 1	Unassigned	1/32
0001	Unassigned	1/16
001	Unassigned	1/8
010	Provider-based addresses	1/8
011	Unassigned	1/8
100	Geographic-based addresses	1/8
101	Unassigned	1/8
110	Unassigned	1/8
1110	Unassigned	1/16
1111 0	Unassigned	1/32
1111 10	Unassigned	1/64
1111 110	Unassigned	1/128
1111 1110 0	Unassigned	1/512
1111 1110 10	Link local use addresses	1/1024
1111 1110 11	Site local use addresses	1/1024
1111 1111	Multicast	1/256

Fig. 5-57. IPv6 addresses

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

Extension header	Description
Hop-by-hop options	Miscellaneous information for routers
Routing	Full or partial route to follow
Fragmentation	Management of datagram fragments
Authentication	Verification of the sender’s identity
Encrypted security payload	Information about the encrypted contents
Destination options	Additional information for the destination

Fig. 5-58. IPv6 extension headers.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

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

Fig. 5-59. The hop-by-hop extension header for large datagrams (jumbograms).

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

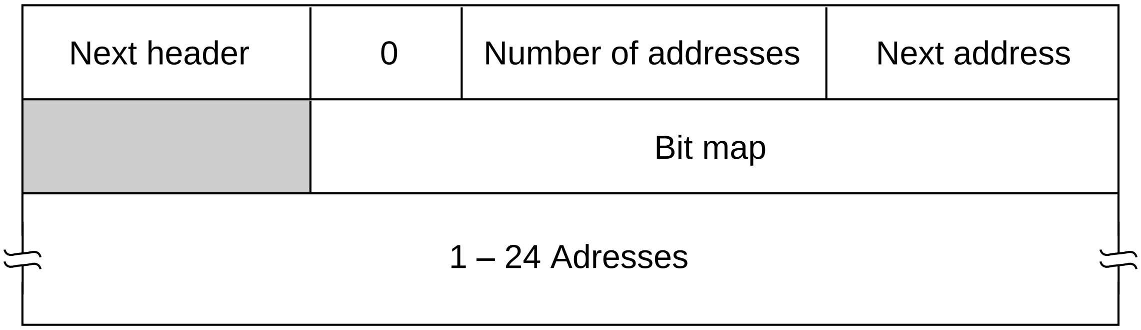


Fig. 5-60. The extension header for routing.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

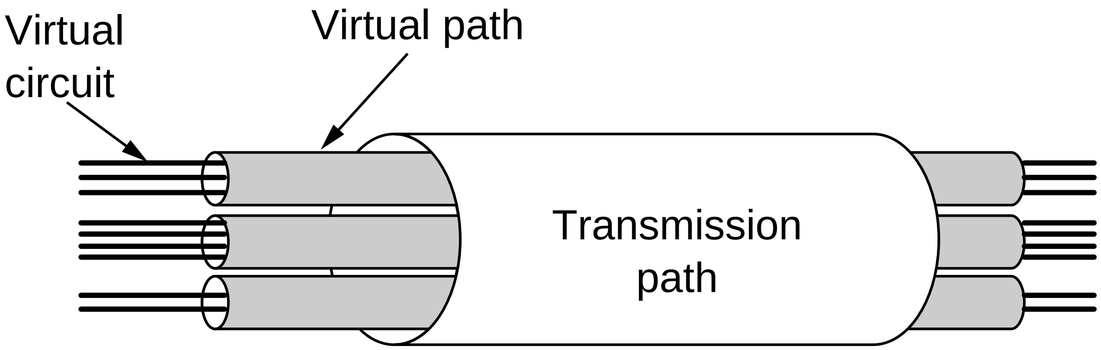


Fig. 5-61. A transmission path can hold multiple virtual paths, each of which can hold multiple virtual circuits.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

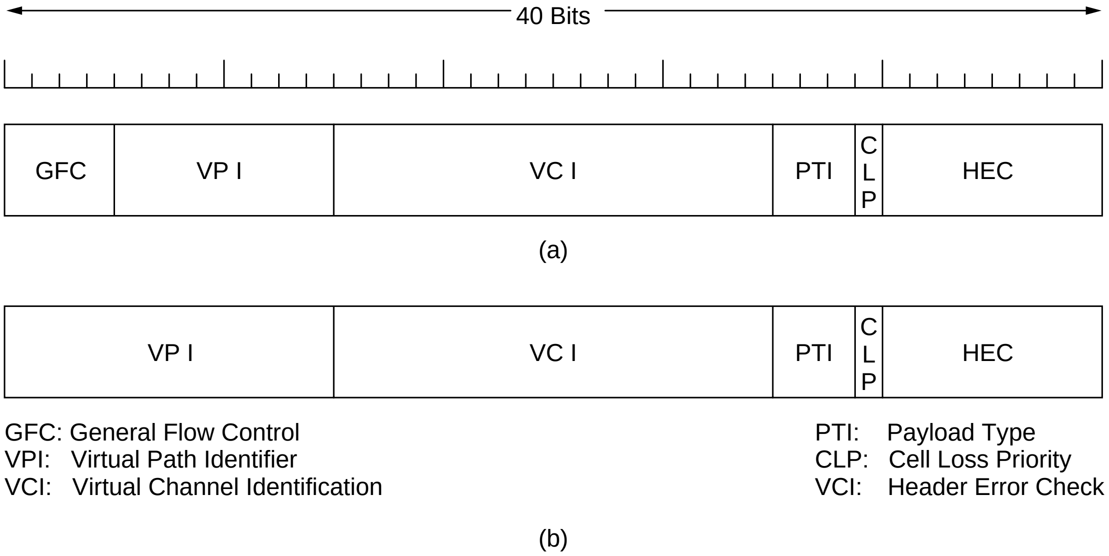


Fig. 5-62. (a) The ATM layer header at the UNI. (b) The ATM layer header at the NNI.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

Payload type	Meaning
000	User data cell, no congestion, cell type 0
001	User data cell, no congestion, cell type 1
010	User data cell, congestion experienced, cell type 0
011	User data cell, congestion experienced, cell type 1
100	Maintenance information between adjacent switches
101	Maintenance information between source and destination switches
110	Resource Management cell (used for ABR congestion control)
111	Reserved for future function

Fig. 5-63. Values of the *PTI* field.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

Message	Meaning when sent by host	Meaning when sent by network
SETUP	Please establish a circuit	Incoming call
CALL PROCEEDING	I saw the incoming call	Your call request will be attempted
CONNECT	I accept the incoming call	Your call request was accepted
CONNECT ACK	Thanks for accepting	Thanks for making the call
RELEASE	Please terminate the call	The other side has had enough
RELEASE COMPLETE	Ack for RELEASE	Ack for RELEASE

Fig. 5-64. Messages used for connection establishment and release.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

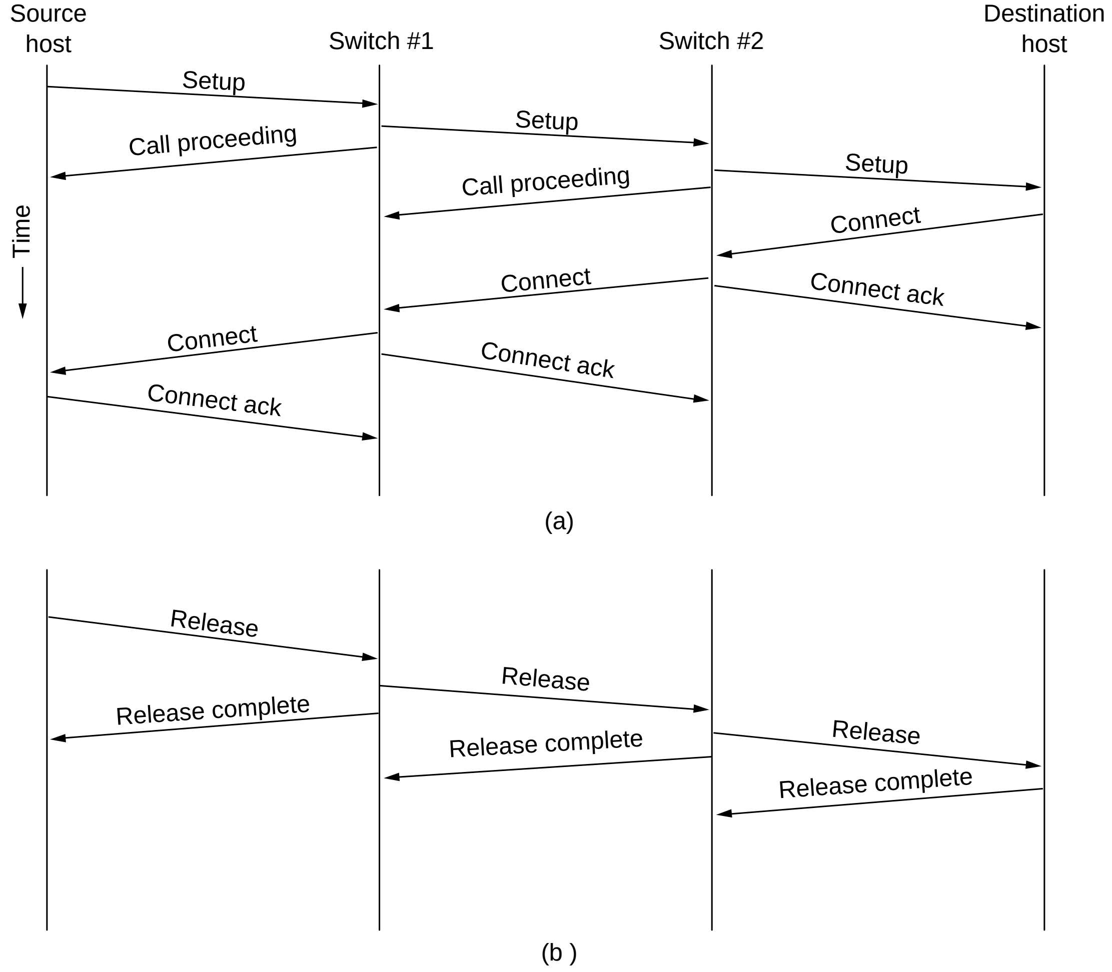


Fig. 5-65. (a) Connection setup in an ATM network. (b) Connection release.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

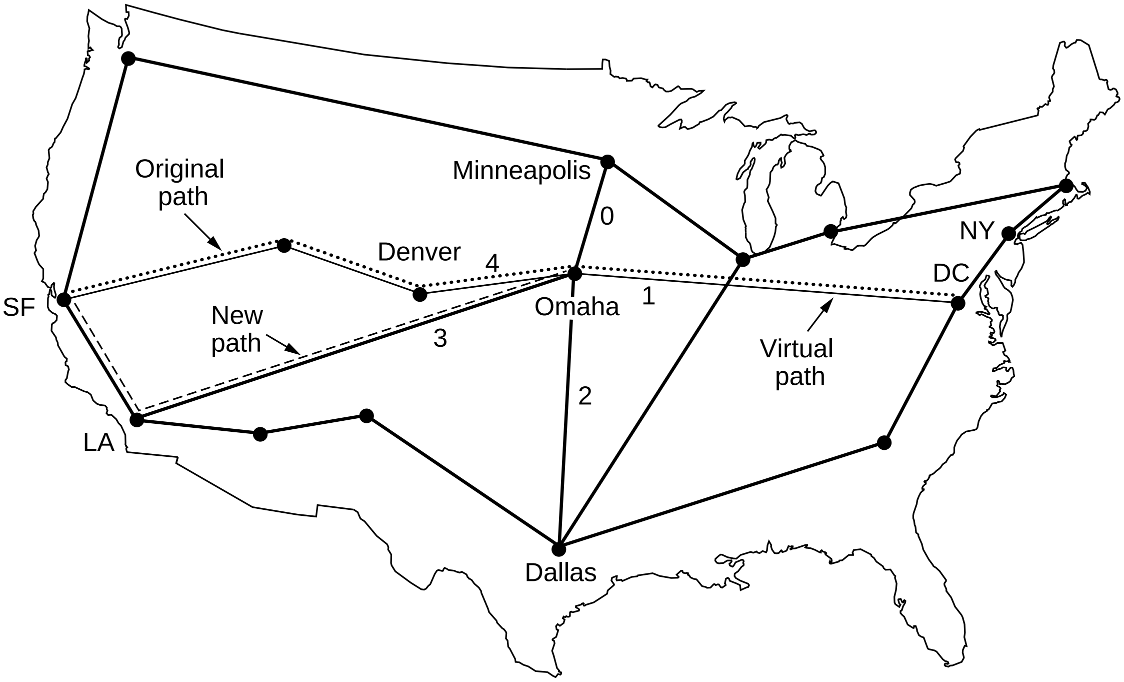


Fig. 5-66. Rerouting a virtual path reroutes all of its virtual circuits.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

Source	Incoming line	Incoming VPI	Destination	Outgoing line	Outgoing VPI	Path:
NY	1	1	SF	4	1	New
NY	1	2	Denver	4	2	New
LA	3	1	Minneapolis	0	1	New
DC	1	3	LA	3	2	New
NY	1	1	SF	4	1	Old
SF	4	3	DC	1	4	New
DC	1	5	SF	4	4	New
NY	1	2	Denver	4	2	Old
SF	4	5	Minneapolis	0	2	New
NY	1	1	SF	4	1	Old

Fig. 5-67. Some routes through the Omaha switch of Fig. 5-0.

	VPI_table for Minn.		VPI_table for DC		VPI_table for Dallas		VPI_table for LA		VPI_table for Denver	
Incoming	Outgoing		Outgoing		Outgoing		Outgoing		Outgoing	
VPI	Line	VPI	Line	VPI	Line	VPI	Line	VPI	Line	VPI
0										
1	3	1	4	1			0	1	1	1
2	4	5	4	2			1	3	1	2
3			3	2					1	4
4			4	3					1	5
5			4	4					0	2
6										
7										
8										
4095										
	Line 0		Line 1		Line 2		Line 3		Line 4	

Fig. 5-68. The table entries for the routes of Fig. 5-0.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

Class	Description	Example
CBR	Constant bit rate	T1 circuit
RT-VBR	Variable bit rate: real time	Real-time videoconferencing
NRT-VBR	Variable bit rate: non-real time	Multimedia email
ABR	Available bit rate	Browsing the Web
UBR	Unspecified bit rate	Background file transfer

Fig. 5-69. The ATM service categories.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

Service characteristic	**CBR**	**RT-VBR**	**NRT-VBR**	**ABR**	**UBR**
Bandwidth guarantee	Yes	Yes	Yes	Optional	No
Suitable for real-time traffic	Yes	Yes	No	No	No
Suitable for bursty traffic	No	No	Yes	Yes	Yes
Feedback about congestion	No	No	No	Yes	No

Fig. 5-70. Characteristics of the ATM service categories.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

Parameter	Acronym	Meaning
Peak cell rate	PCR	Maximum rate at which cells will be sent
Sustained cell rate	SCR	The long-term average cell rate
Minimum cell rate	MCR	The minimum acceptable cell rate
Cell delay variation tolerance	CDVT	The maximum acceptable cell jitter
Cell loss ratio	CLR	Fraction of cells lost or delivered too late
Cell transfer delay	CTD	How long delivery takes (mean and maximum)
Cell delay variation	CDV	The variance in cell delivery times
Cell error rate	CER	Fraction of cells delivered without error
Severely-errored cell block ratio	SECBR	Fraction of blocks garbled
Cell misinsertion rate	CMR	Fraction of cells delivered to wrong destination

Fig. 5-71. Some of the quality of service parameters.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

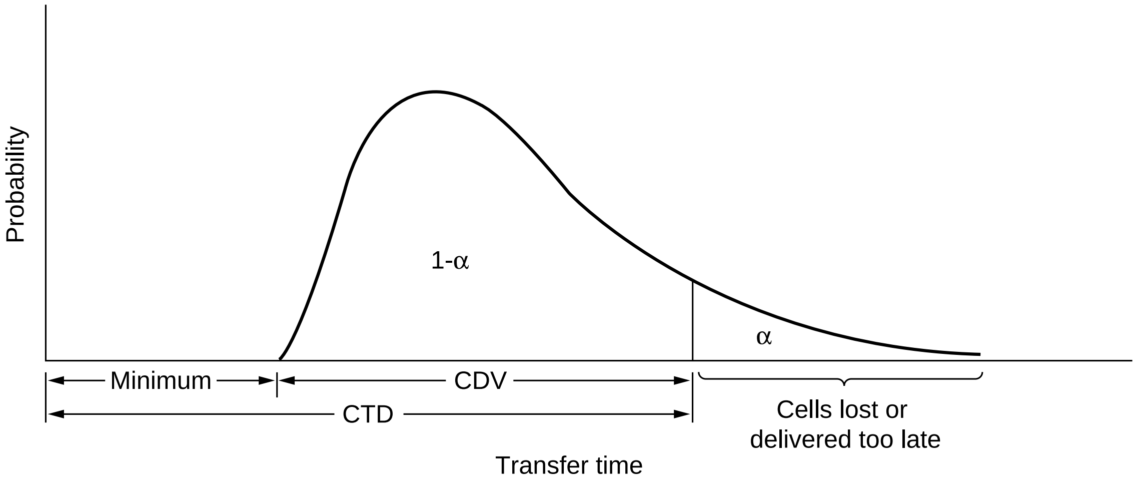


Fig. 5-72. The probability density function for cell arrival times.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

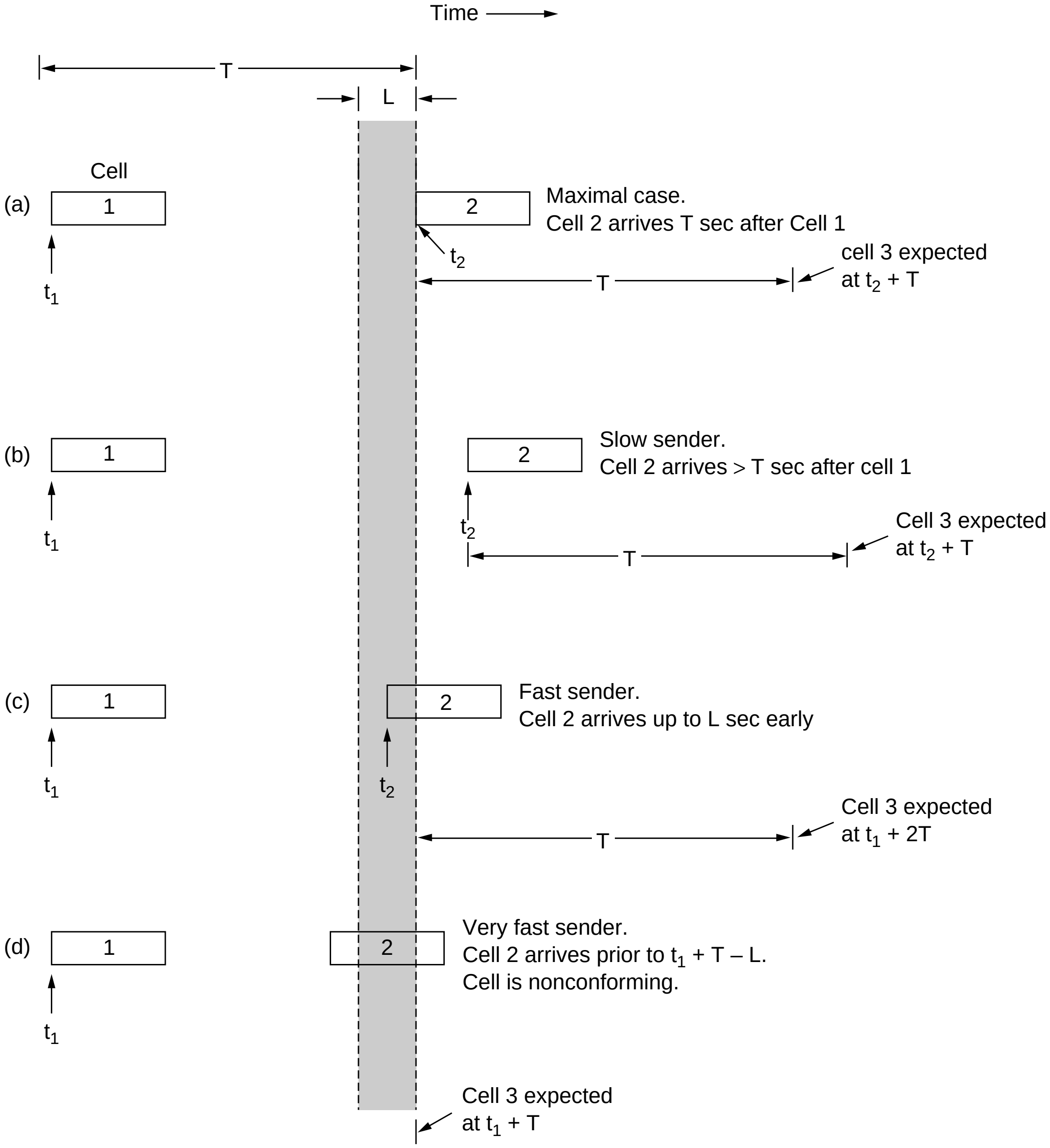


Fig. 5-73. The generic cell rate algorithm.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

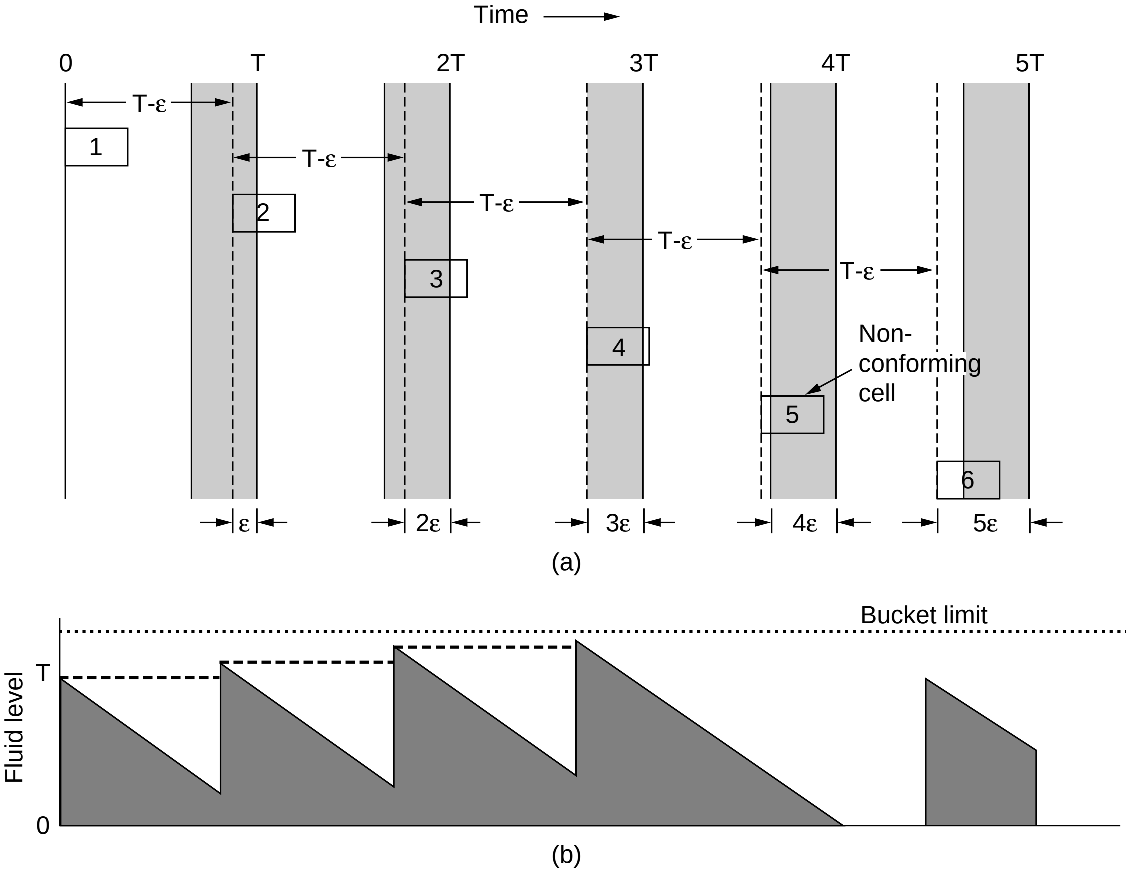


Fig. 5-74. (a) A sender trying to cheat. (b) The same cell arrival pattern, but now viewed in terms of a leaky bucket.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

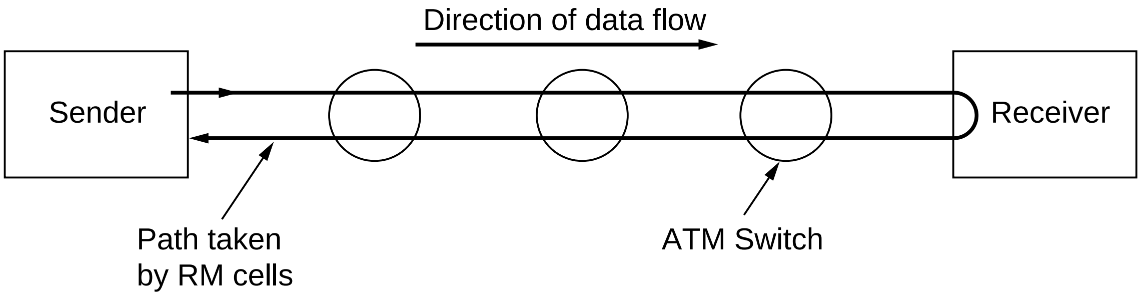


Fig. 5-75. The path taken by RM cells.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall

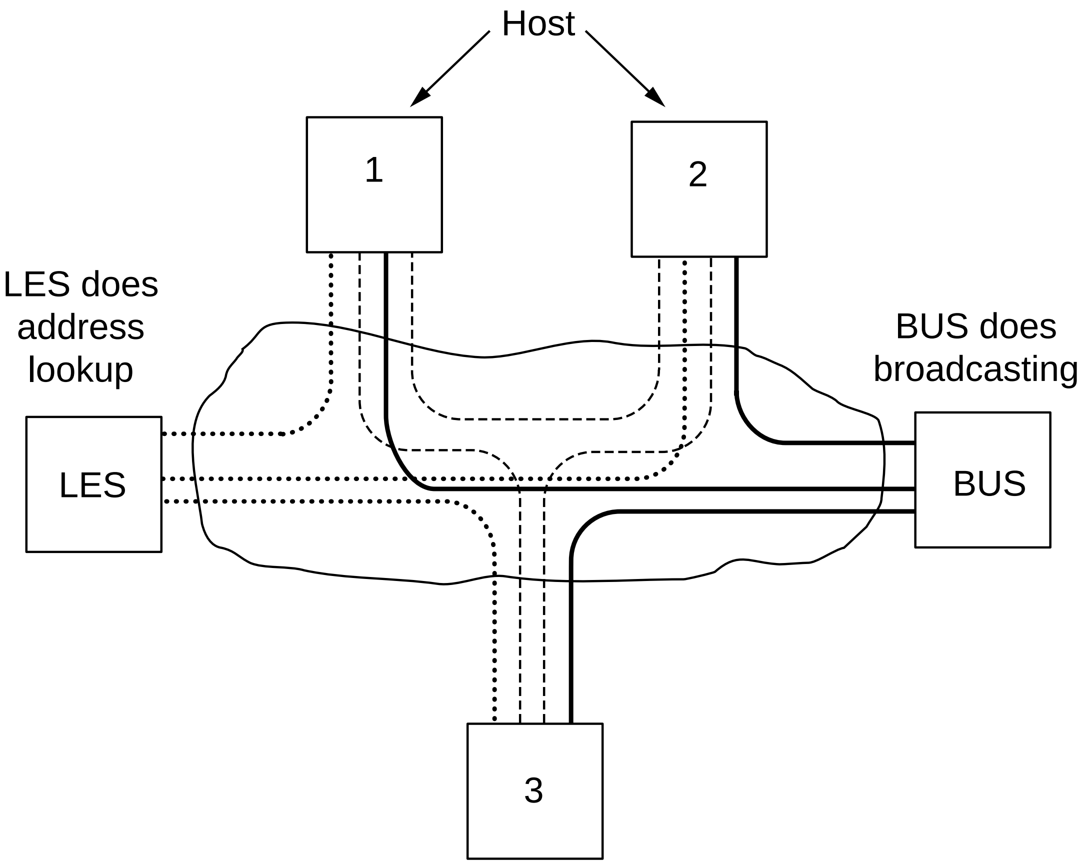


Fig. 5-76. ATM LAN emulation.

From: *Computer Networks*, 3rd ed. by Andrew S. Tanenbaum, © 1996 Prentice Hall