UNIT – IV

Network Layer in the Internet
&
Transport Layer

Internetworking

Internetworking joins multiple, different networks into a single larger network

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

How Networks Differ

Differences can be large; complicates internetworking

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

How networks can be connected

In the Physical Layer

Networks can be connected by repeaters or hubs

It move the bits from one network to an identical network

Most are the analog devices

In the data link layer

Bridges and switches are used

They can accept frames, examine the MAC address and forward the frames to a different network.

In the Network Layer:

Routers that can connect two networks.

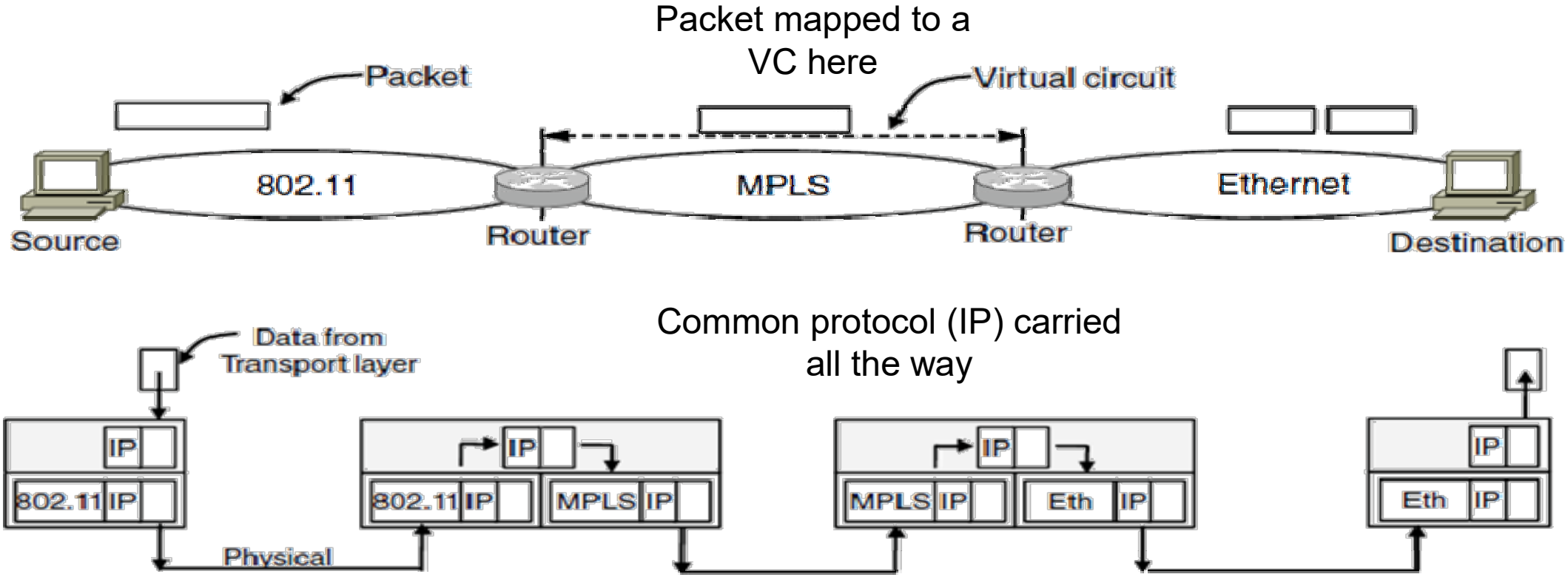
If the connected networks are dissimilar then the router may be able to translate between the packet formats.

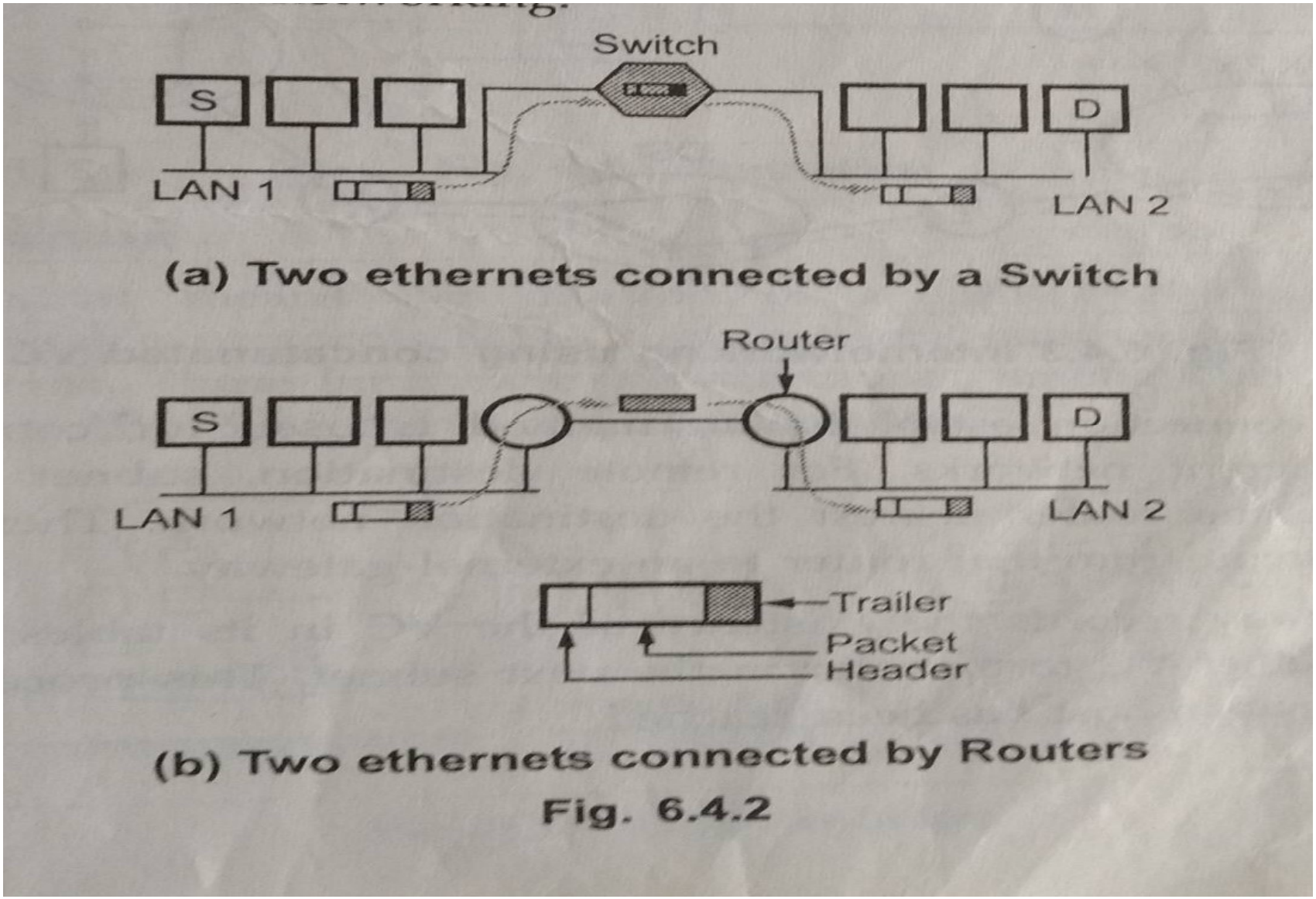
In the transport Layer:

Transport gateway interface between two transport connections

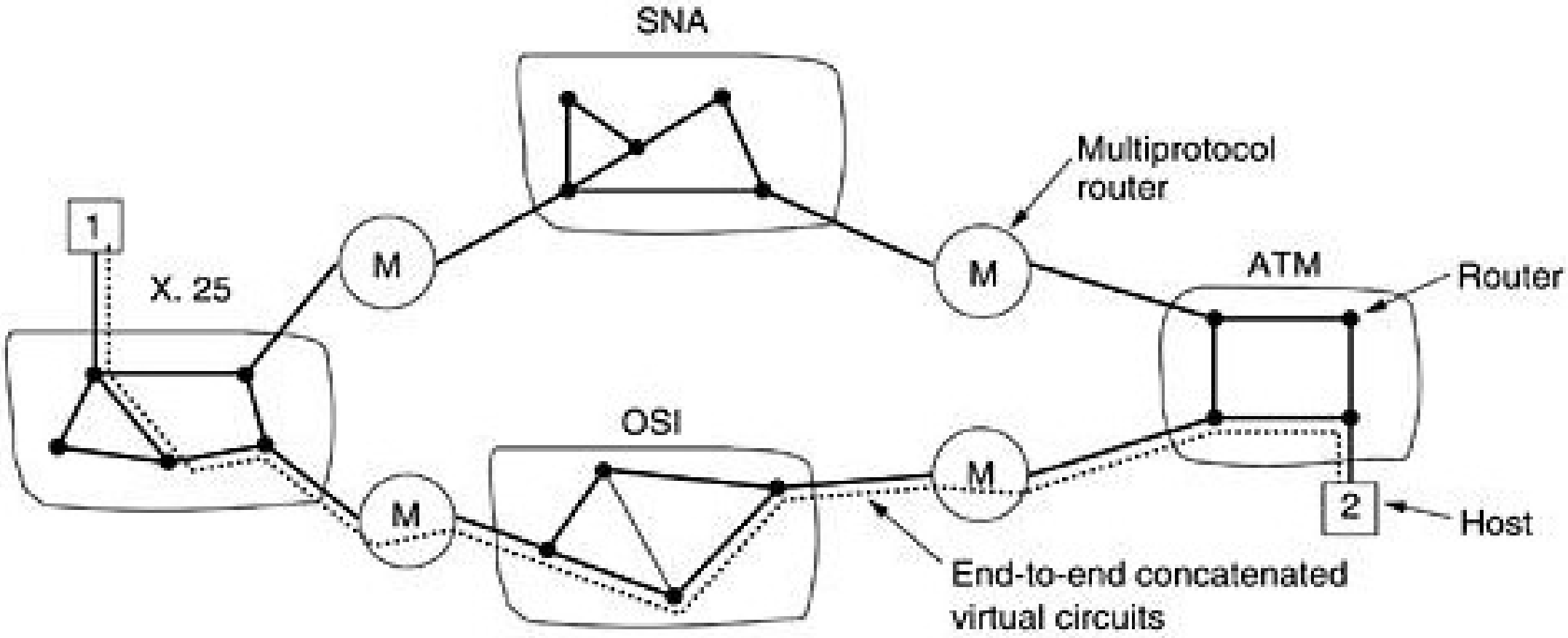
How Networks Can Be Connected

Internetworking based on a common network layer – IP

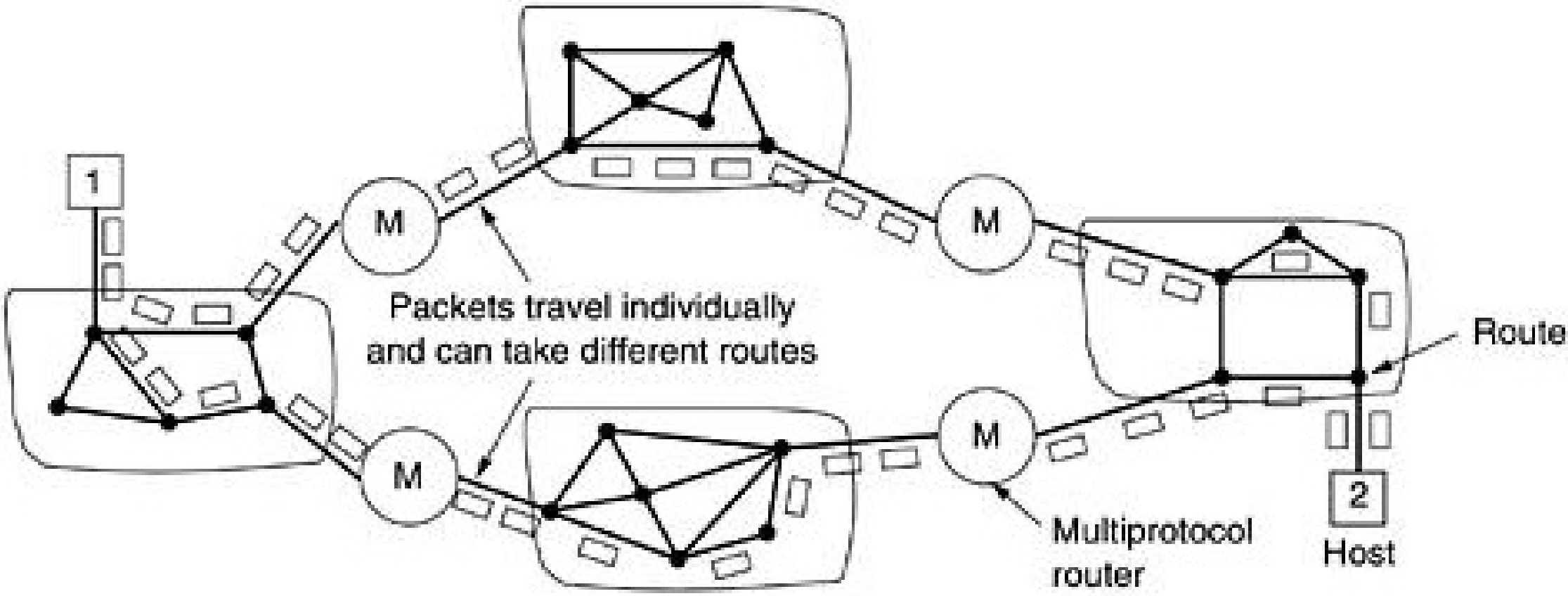



Switch
S
D
LAN 1
LAN 2
(a) Two ethernets connected by a Switch
Router
S
D
LAN 1
LAN 2
Trailer
Packet
Header
(b) Two ethernets connected by Routers
Fig. 6.4.2

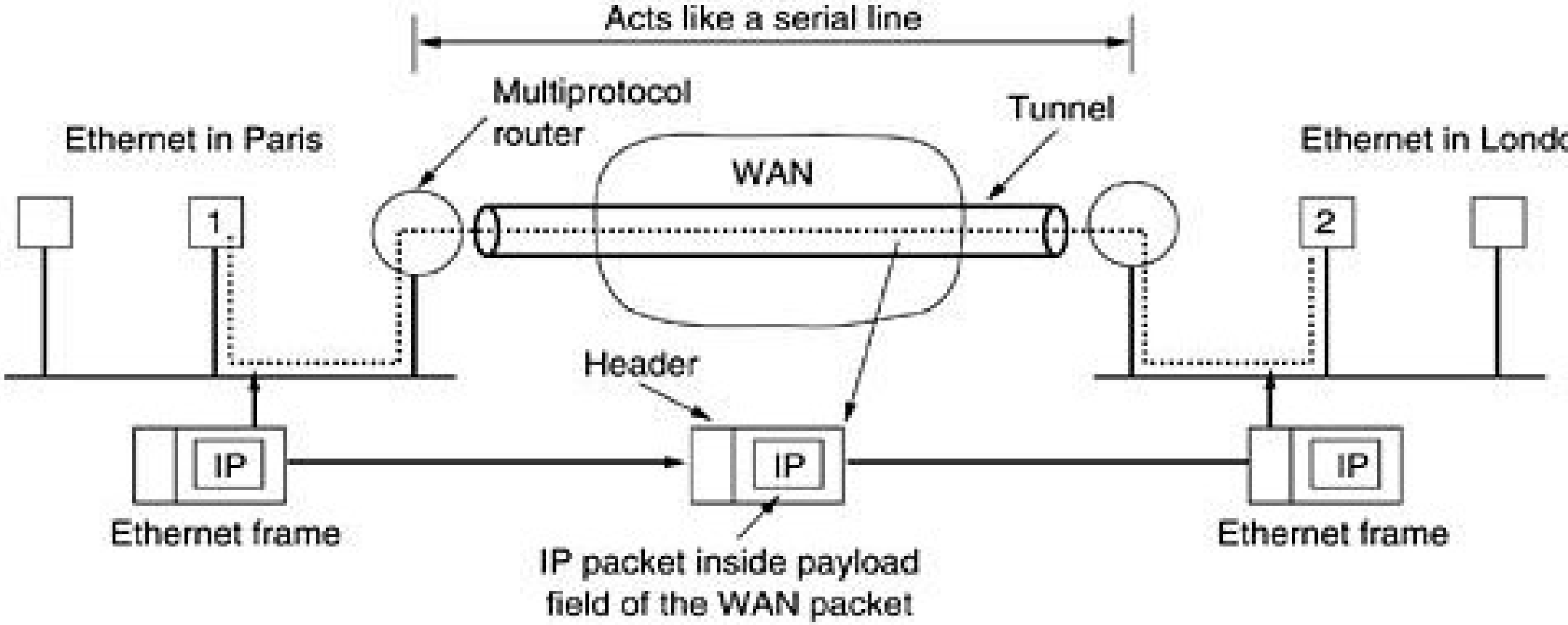
Concatenated virtual circuits



Connectionless internetworking



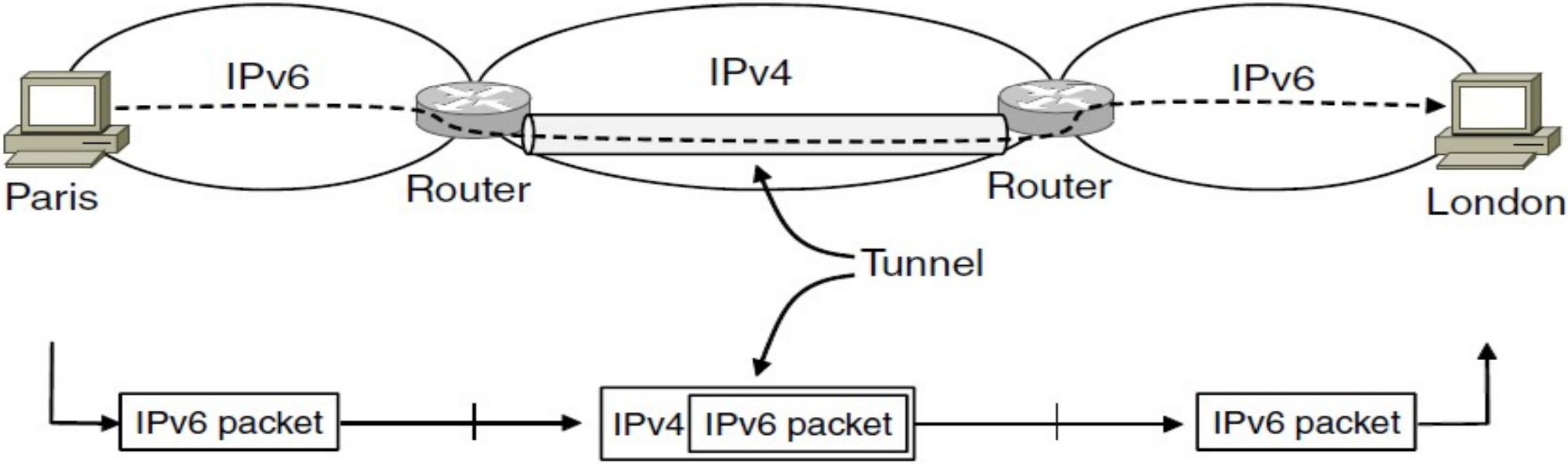
TUNNELING


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

Tunneling (1)

Connects two networks through a middle one

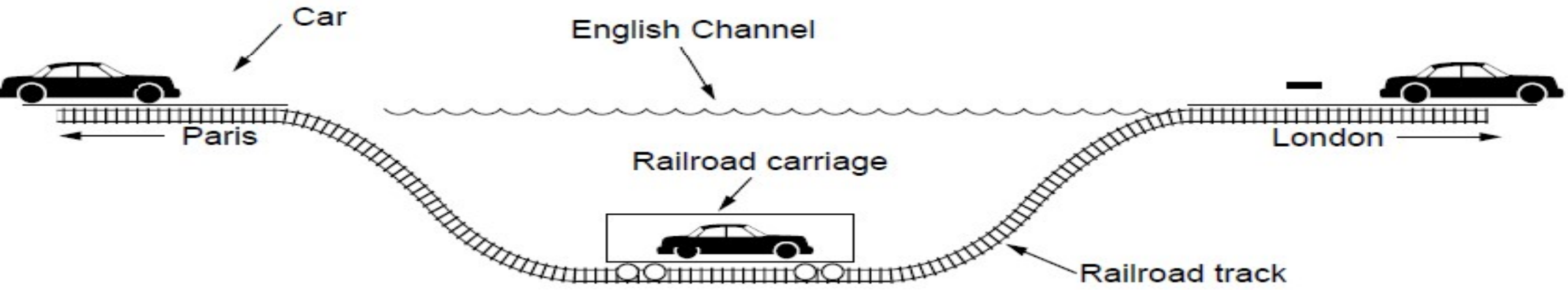
- Packets are encapsulates over the middle



Tunneling (2)

Tunneling analogy:

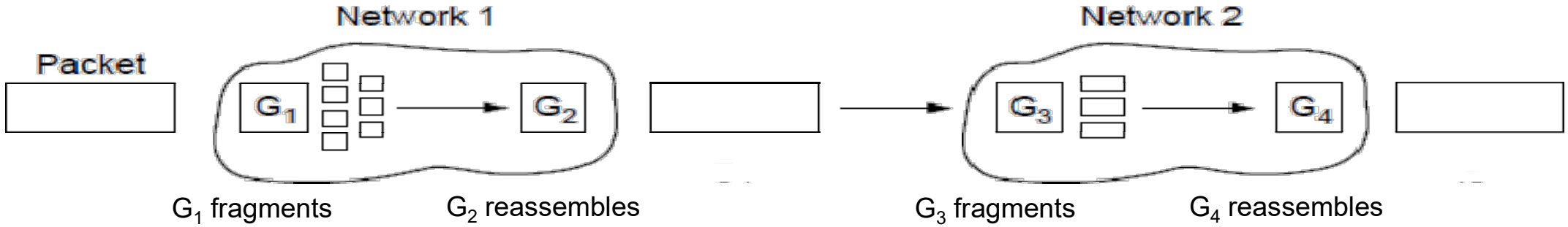
- tunnel is a link; packet can only enter/exit at ends



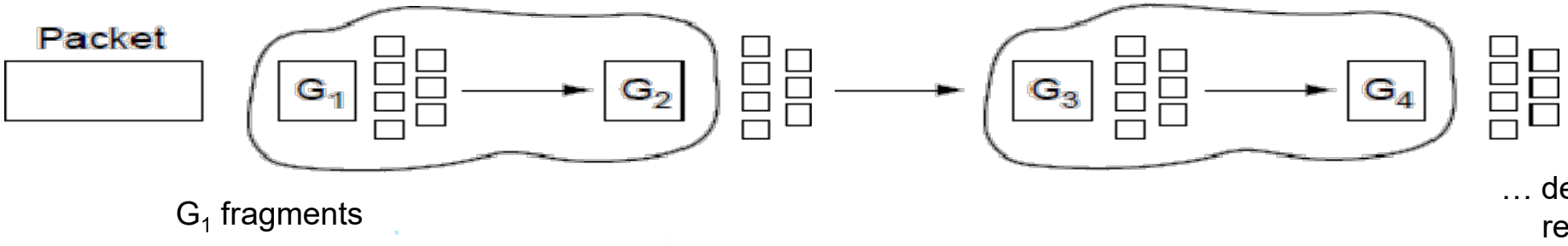
Packet Fragmentation (1)

Networks have different packet size limits for many reasons

- Large packets sent with fragmentation & reassembly



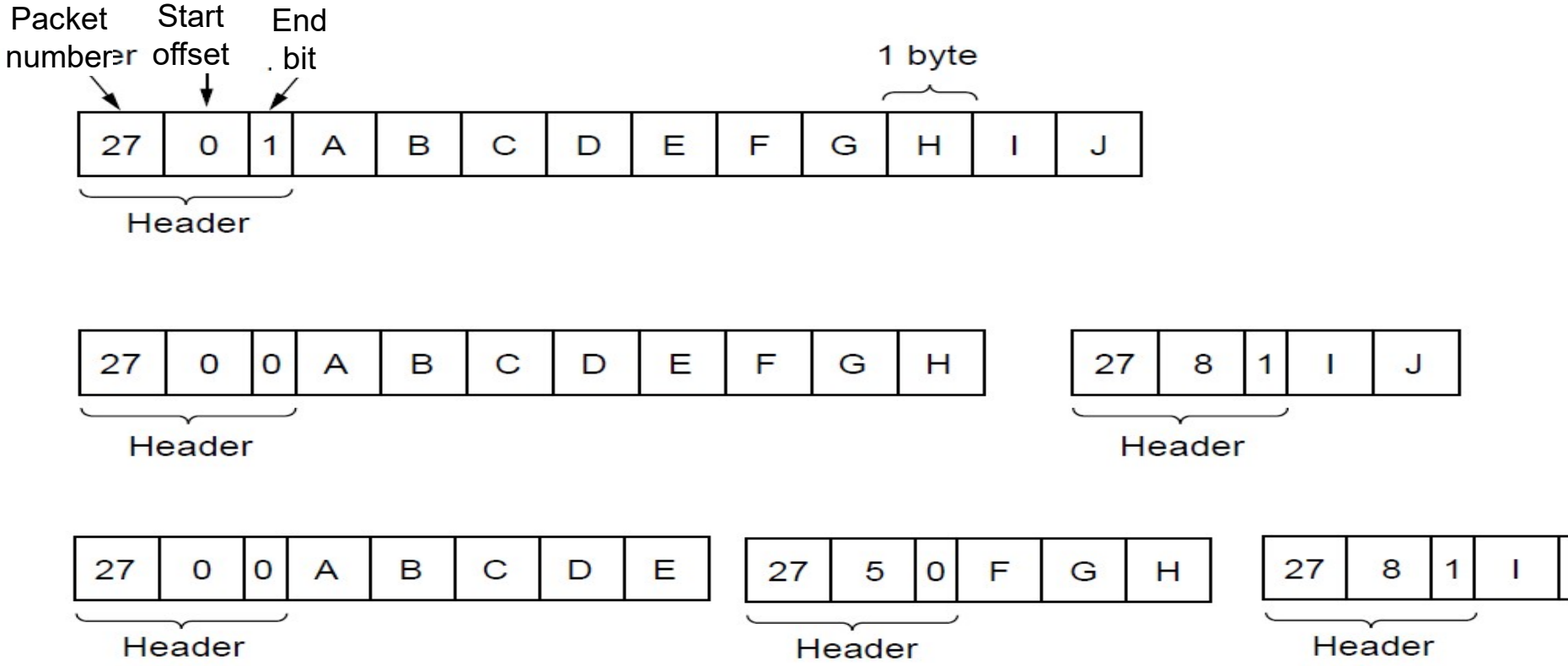
Transparent – packets fragmented / reassembled in each network



Non-transparent – fragments are reassembled at destination

Packet Fragmentation (2)

Example of IP-style fragmentation:



iginal packet:
0 data bytes)

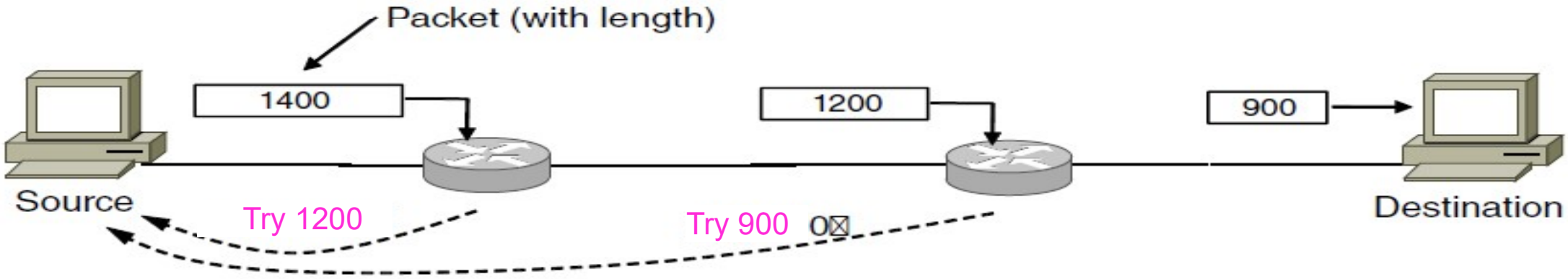
Fragmented:
o 8 data bytes)

e-fragmented:
(to 5 bytes)

Packet Fragmentation (3)

Path MTU Discovery avoids network fragmentation

- Routers return MTU (Max. Transmission Unit) to source and discard large packets

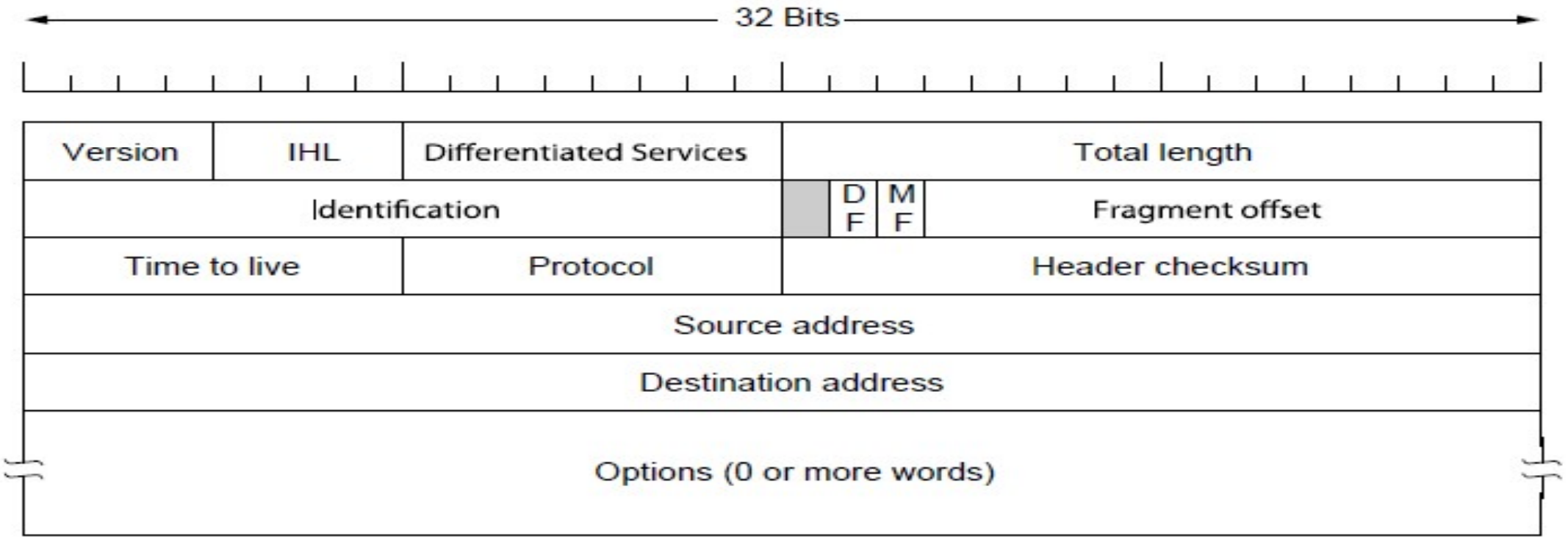


Network Layer in the Internet (1)

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

IP Version 4 Protocol (1)

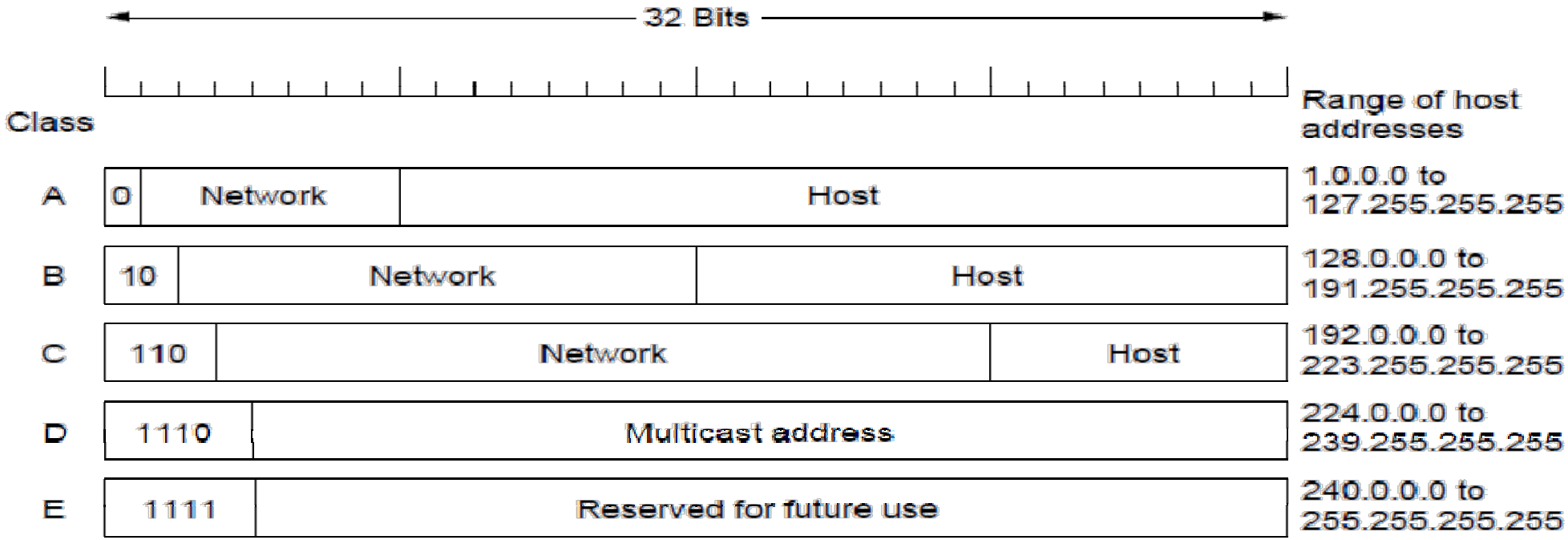
IPv4 (Internet Protocol) header is carried on all packets and has fields for the key parts of the protocol:



IP Addresses (1) – Classful Addresin

Old addresses came in blocks of fixed size (A, B, C)

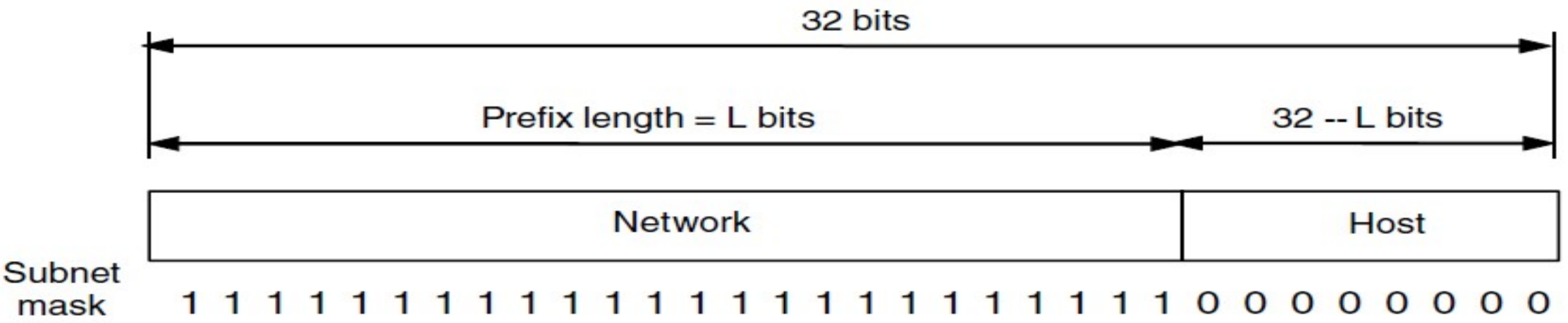
- Carries size as part of address, but lacks flexibility
- Called classful (vs. classless) addressing



IP Addresses (2) – Prefixes

Addresses are allocated in blocks called prefixes

- Prefix is determined by the network portion
- Has 2^L addresses aligned on 2^L boundary
- Written address/length, e.g., 18.0.31.0/24



IP Addresses (3) – Prefixes

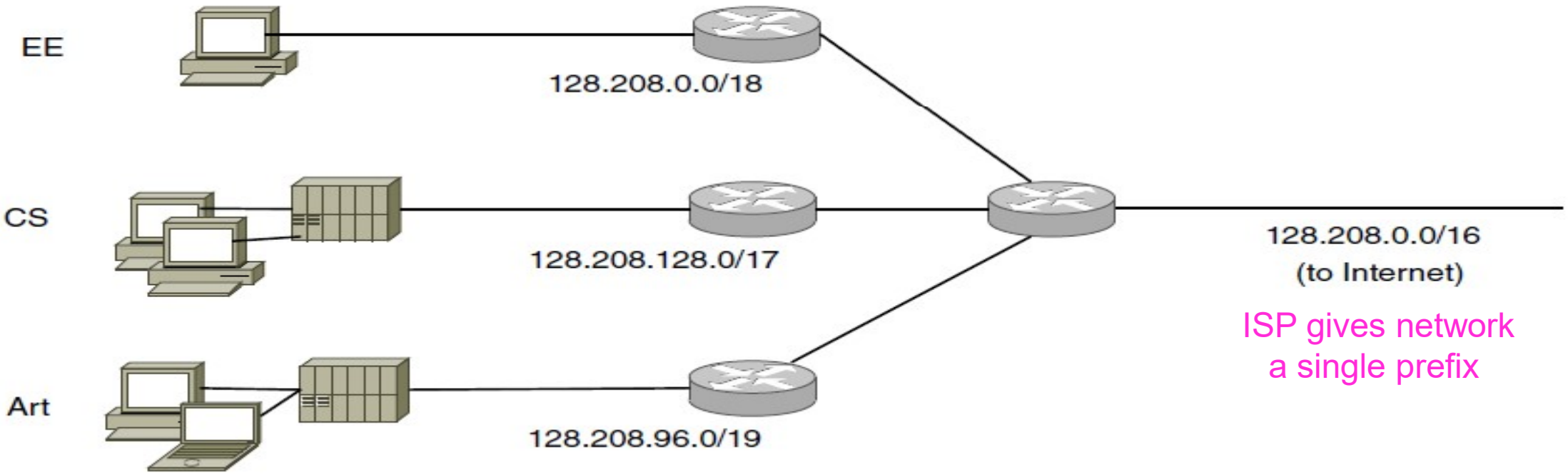
	Private address range	
Class	start address	finish address
A	10.0.0.0	10.255.255.255
B	172.16.0.0	172.31.255.255
C	192.168.0.0	192.168.255.255

	Public address range	
Class	start address	finish address
A	0.0.0.0	126.255.255.255
B	128.0.0.0	191.255.255.255
C	192.0.0.0	223.255.255.255
D	224.0.0.0	239.255.255.255
E	240.0.0.0	254.255.255.255

IP Addresses (4) – Subnets

Subnetting splits up IP prefix to help with management

- Looks like a single prefix outside the network

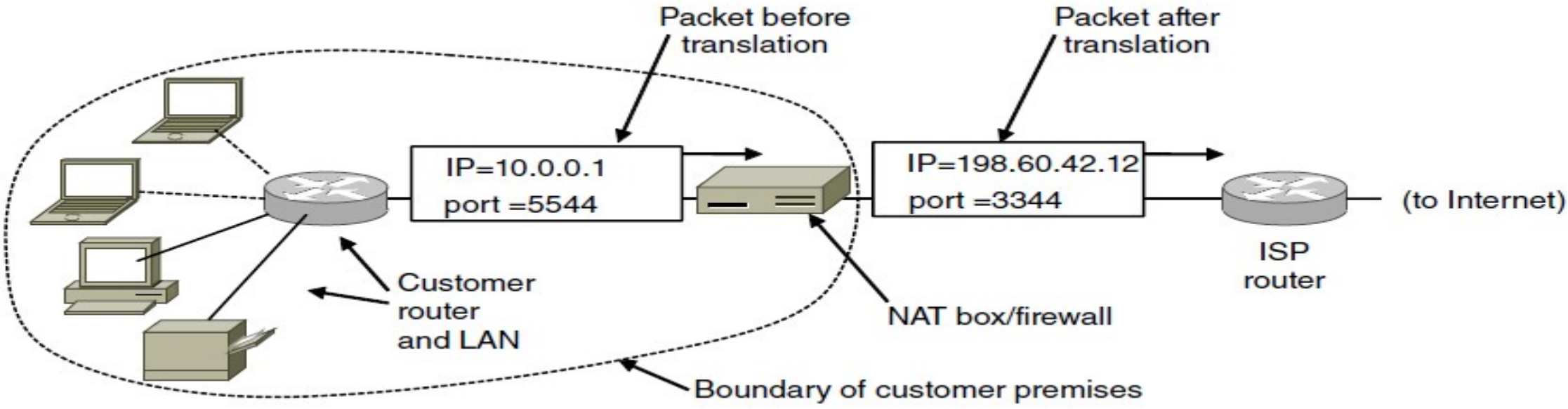


Network divides it into subnets internally

IP Addresses (5) – NAT

NAT (Network Address Translation) box maps one external IP address to many internal IP addresses

- Uses TCP/UDP port to tell connections apart
- Violates layering; very common in homes, etc.



IP Version 6 (1)

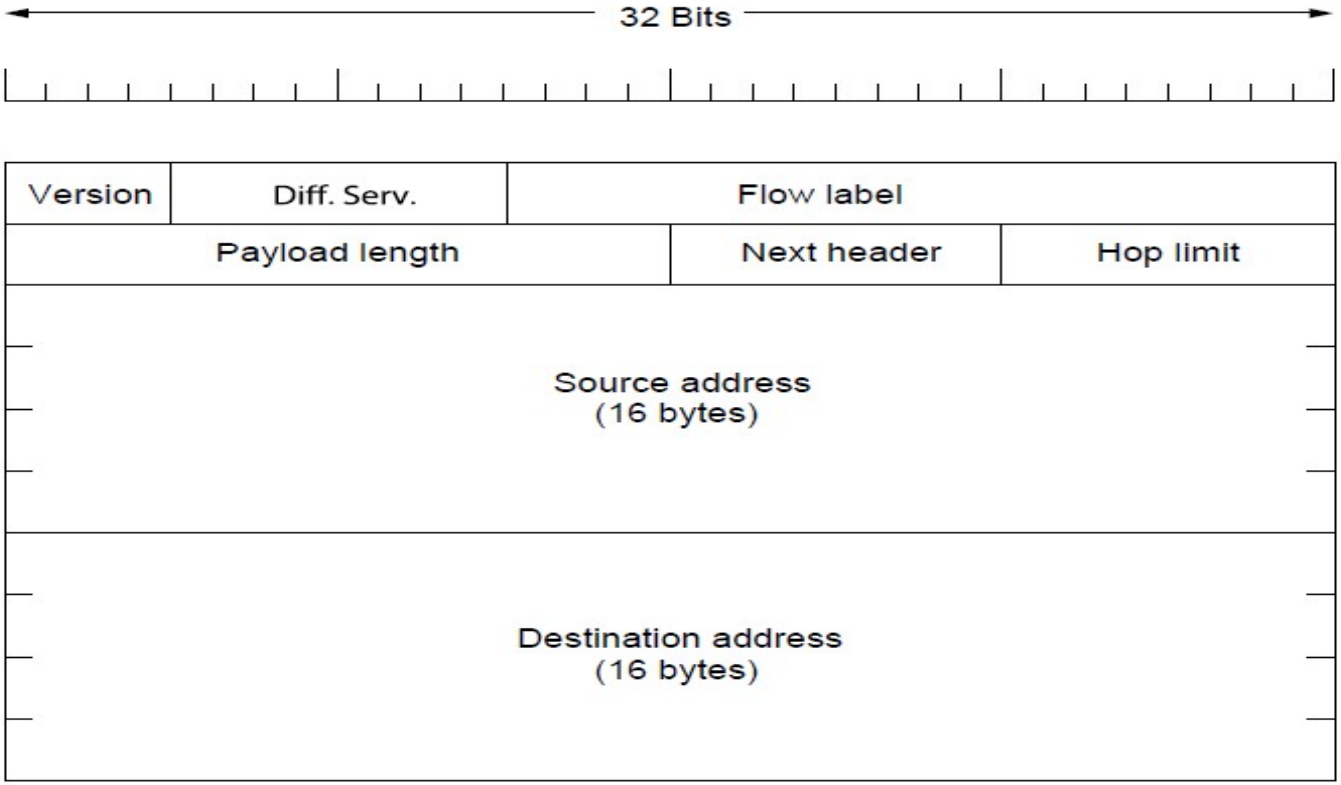
Major upgrade in the 1990s due to impending address exhaustion, with various other goals:

- Support billions of hosts
- Reduce routing table size
- Simplify protocol
- Better security
- Attention to type of service
- Aid multicasting
- Permit coexistence of old, new protocols, …

Deployment has been slow & painful, but may pick up pace now that addresses are all but exhausted.

IP Version 6 (2)

IPv6 protocol header has much longer addresses (128 vs. 32 bits) and is simpler (by using extension headers)



IP Version 6 (3)

IPv6 extension headers handles other functionality

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

Internet Control Protocols (1)

IP works with the help of several control protocols:

- ICMP: (Internet Control Message Protocol)
- ARP: (Address Resolution Protocol)
- RARP: (Reverse Address Resolution Protocol)
- DHCP: (Dynamic Host Configuration Protocol) Assigns a local IP address to a host
  - Gets host started by automatically configuring it
  - Host sends request to server, which grants a lease

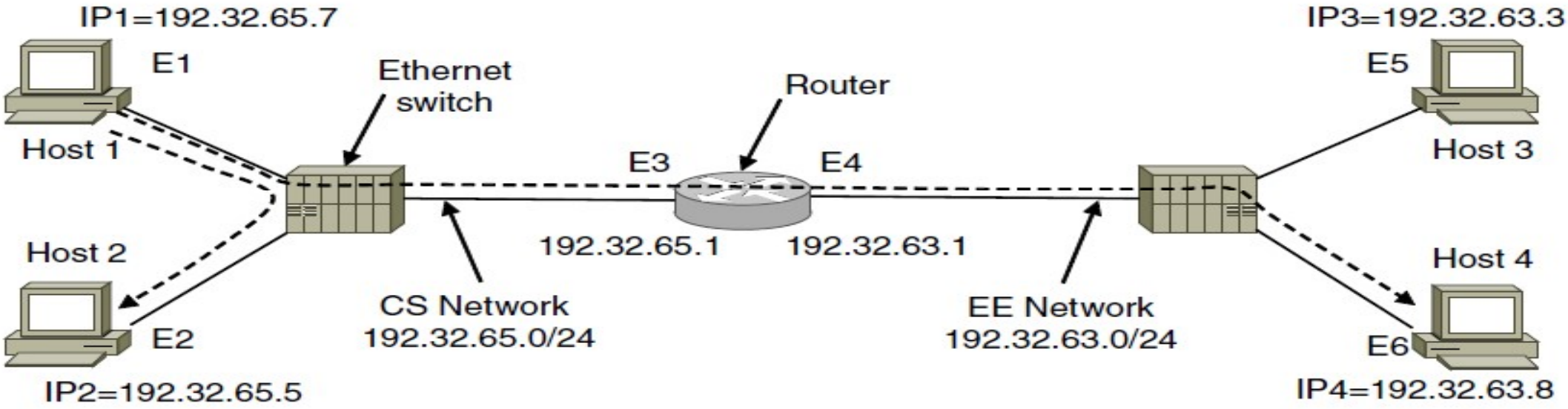
Internet Control Protocols (2)

Main ICMP (Internet Control Message Protocol) 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 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

Internet Control Protocols (3)

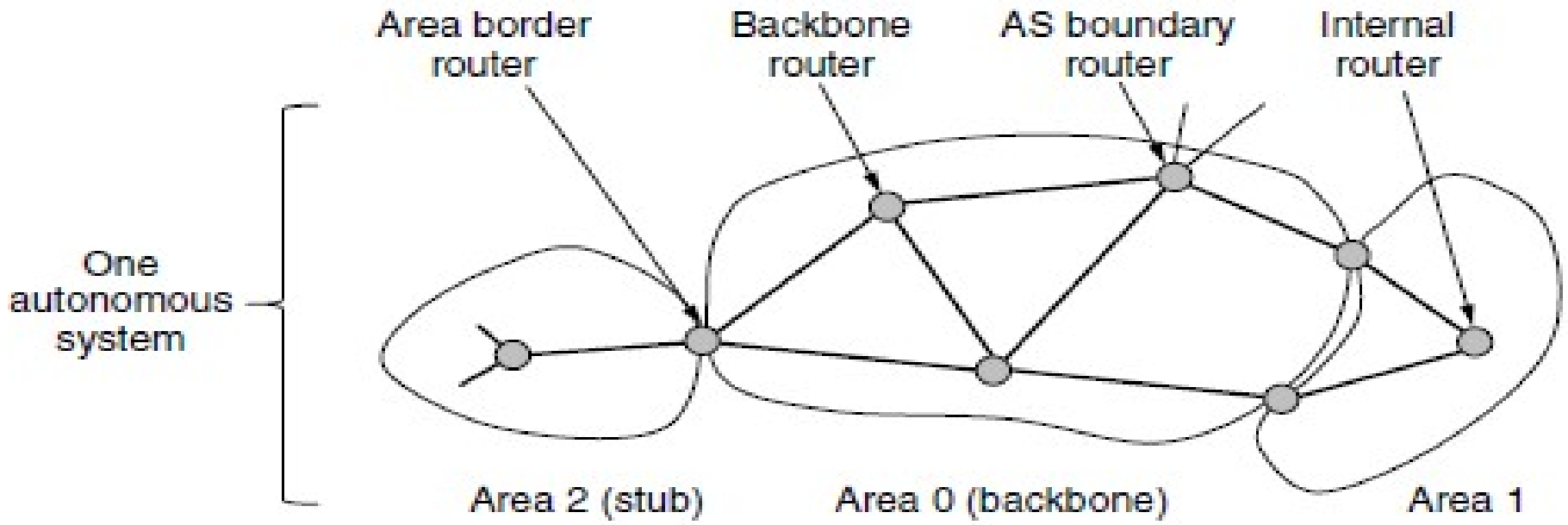
ARP (Address Resolution Protocol) lets nodes find target Ethernet addresses [pink] from their IP addresses



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

OSPF – An Interior Gateway Routing Protocol

- OSPF Open Shortest Path First
- IS-IS (Intermediate System to Intermediate System)



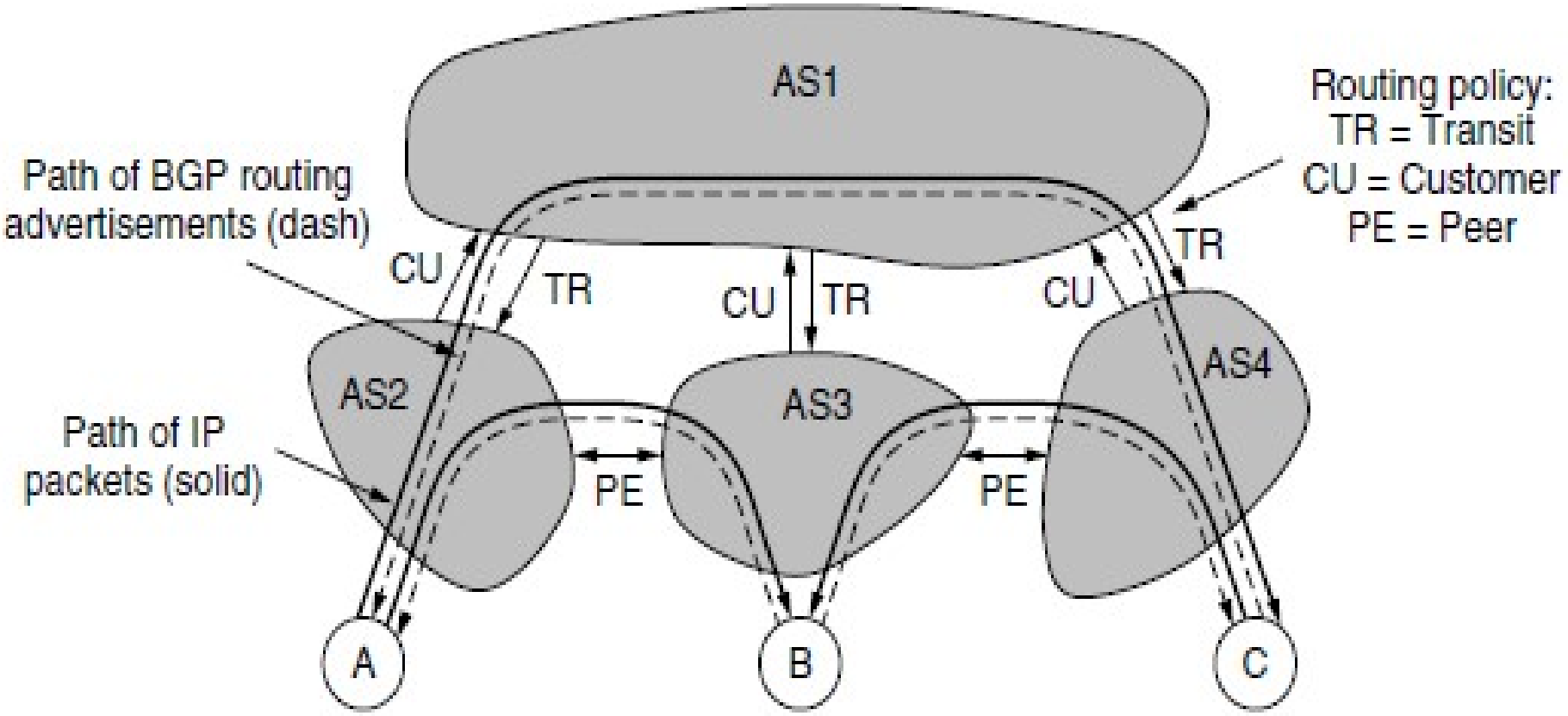
OSPF Messages

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

- Possible Routing Constraints are:
  - Do not carry commercial traffic on the Educational network.
  - Never send traffic from the Pentagon on a route through Iraq.
  - Don’t use AT&T in Australia because performance is poor.
  - Traffic starting or ending at Apple should not transit Google.

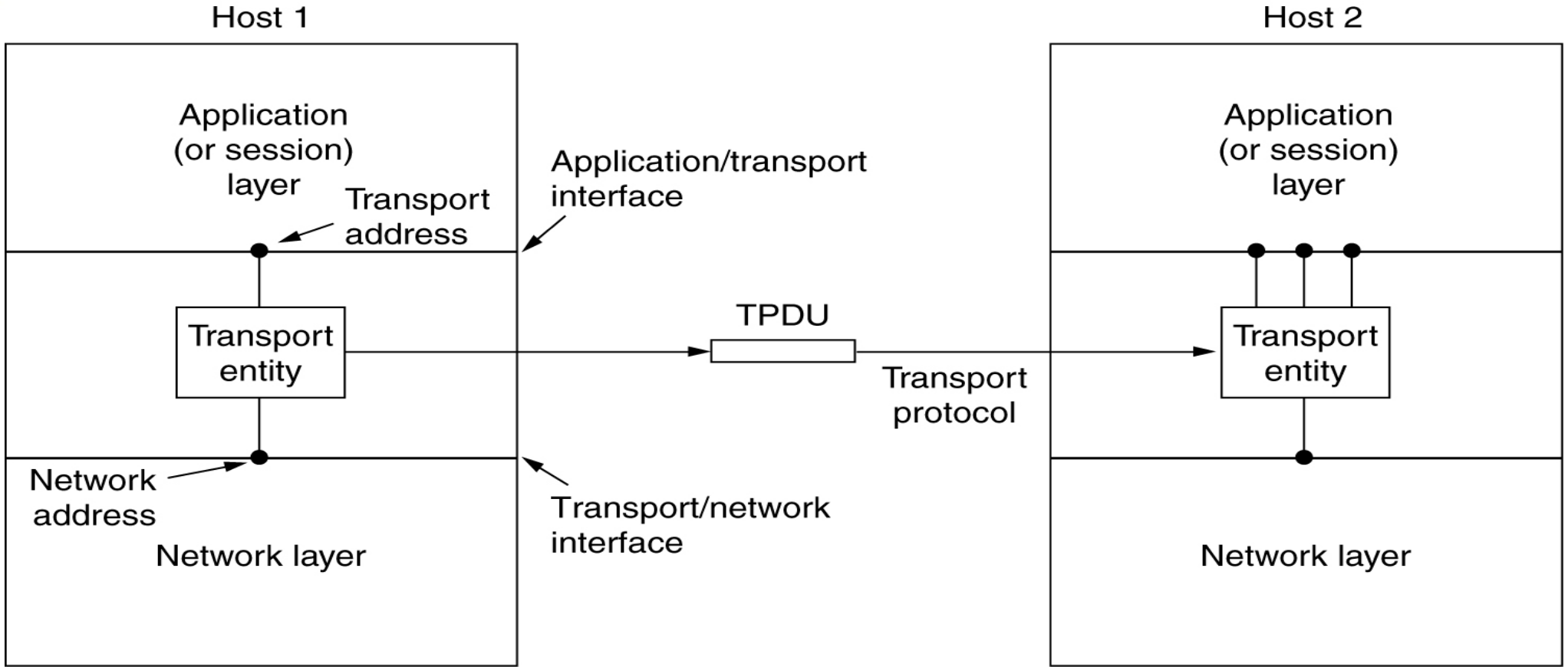
Autonomous Systems



The Transport Service

- Services Provided to the Upper Layers
- Transport Service Primitives
- Berkeley Sockets

Services Provided to the Upper Layers



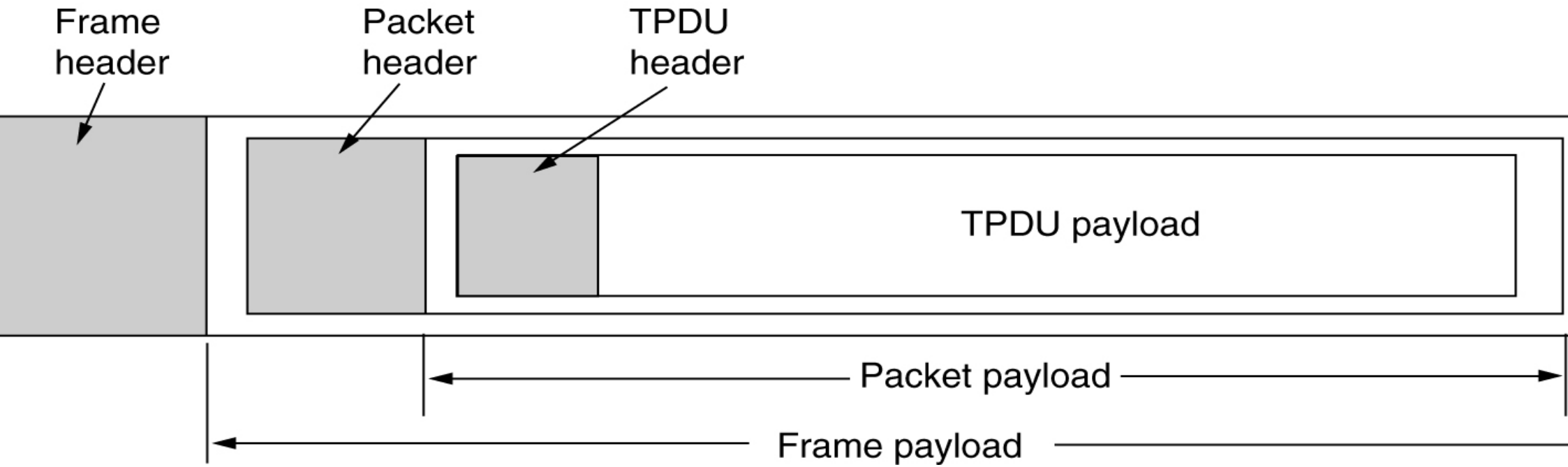
The network, transport, and application layers.

Transport Service Primitives

Primitive	Packet sent	Meaning
LISTEN	(none)	Block until some process tries to connect
CONNECT	CONNECTION REQ.	Actively attempt to establish a connection
SEND	DATA	Send information
RECEIVE	(none)	Block until a DATA packet arrives
DISCONNECT	DISCONNECTION REQ.	This side wants to release the connection

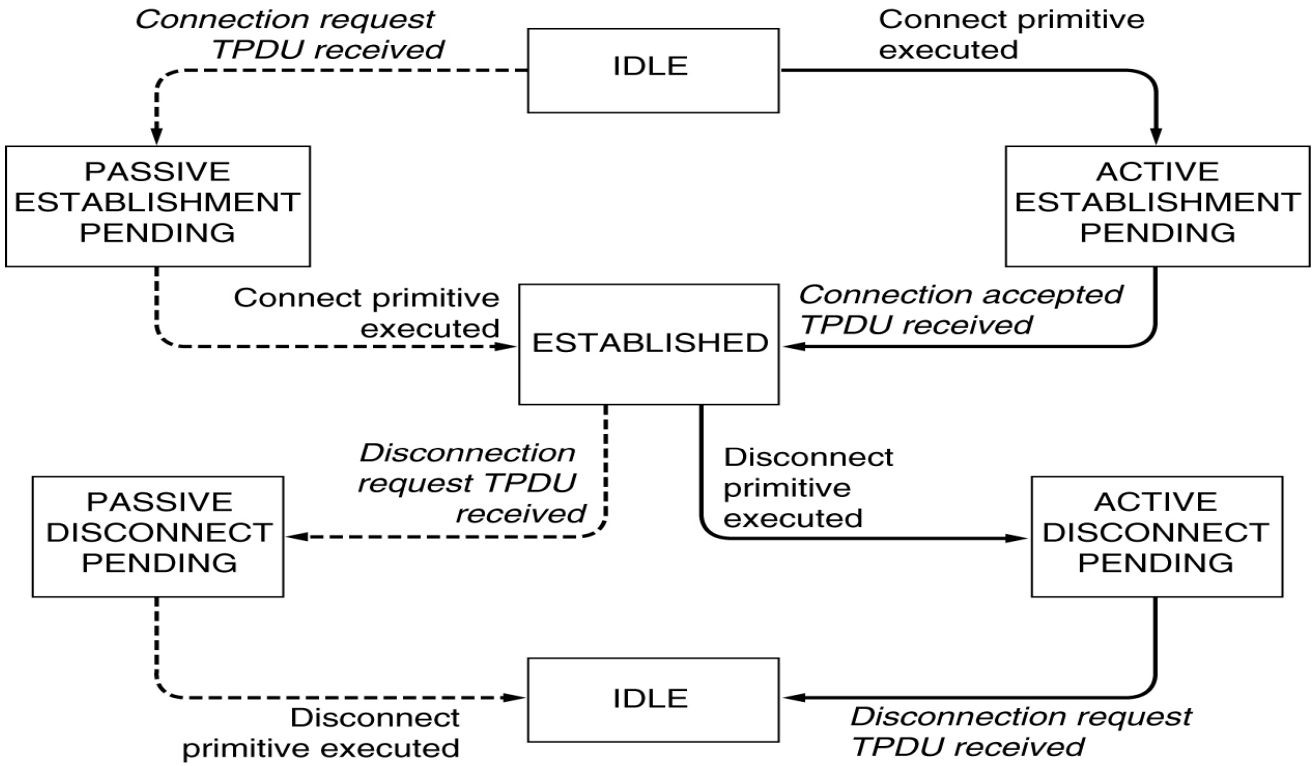
The primitives for a simple transport service.

Transport Service Primitives (2)



The nesting of TPDUs, packets, and frames.

Transport Service Primitives (3)



state diagram for a simple connection management scheme. Transitions labeled in italic
e caused by packet arrivals. The solid lines show the client's state sequence. The dashe
nes show the server's state sequence.

Berkeley Sockets

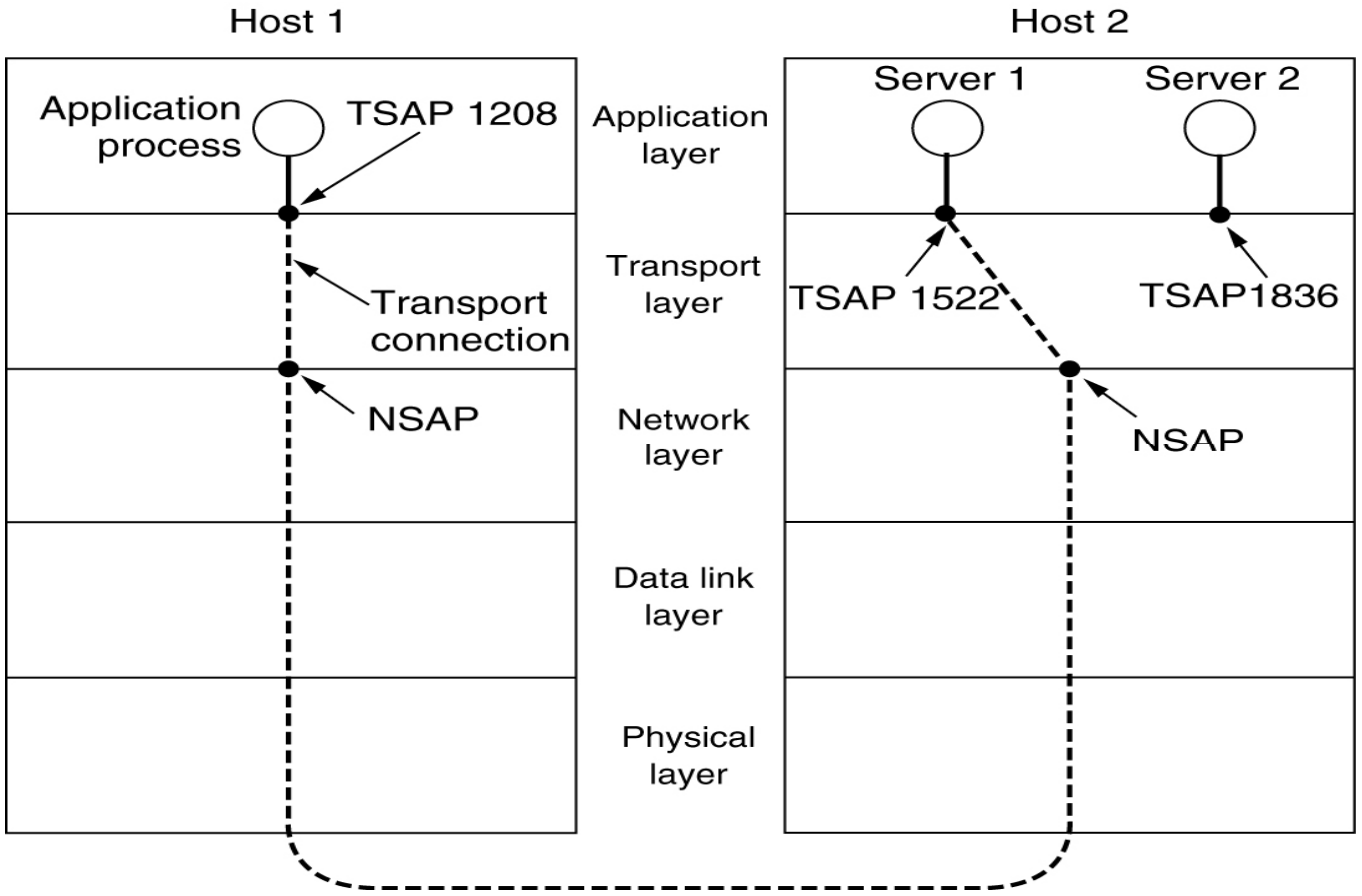
Primitive	Meaning
SOCKET	Create a new communication end point
BIND	Attach a local address to a socket
LISTEN	Announce willingness to accept connections; give queue size
ACCEPT	Block the caller until a connection attempt arrives
CONNECT	Actively attempt to establish a connection
SEND	Send some data over the connection
RECEIVE	Receive some data from the connection
CLOSE	Release the connection

The socket primitives for TCP.

Elements of Transport Protocols

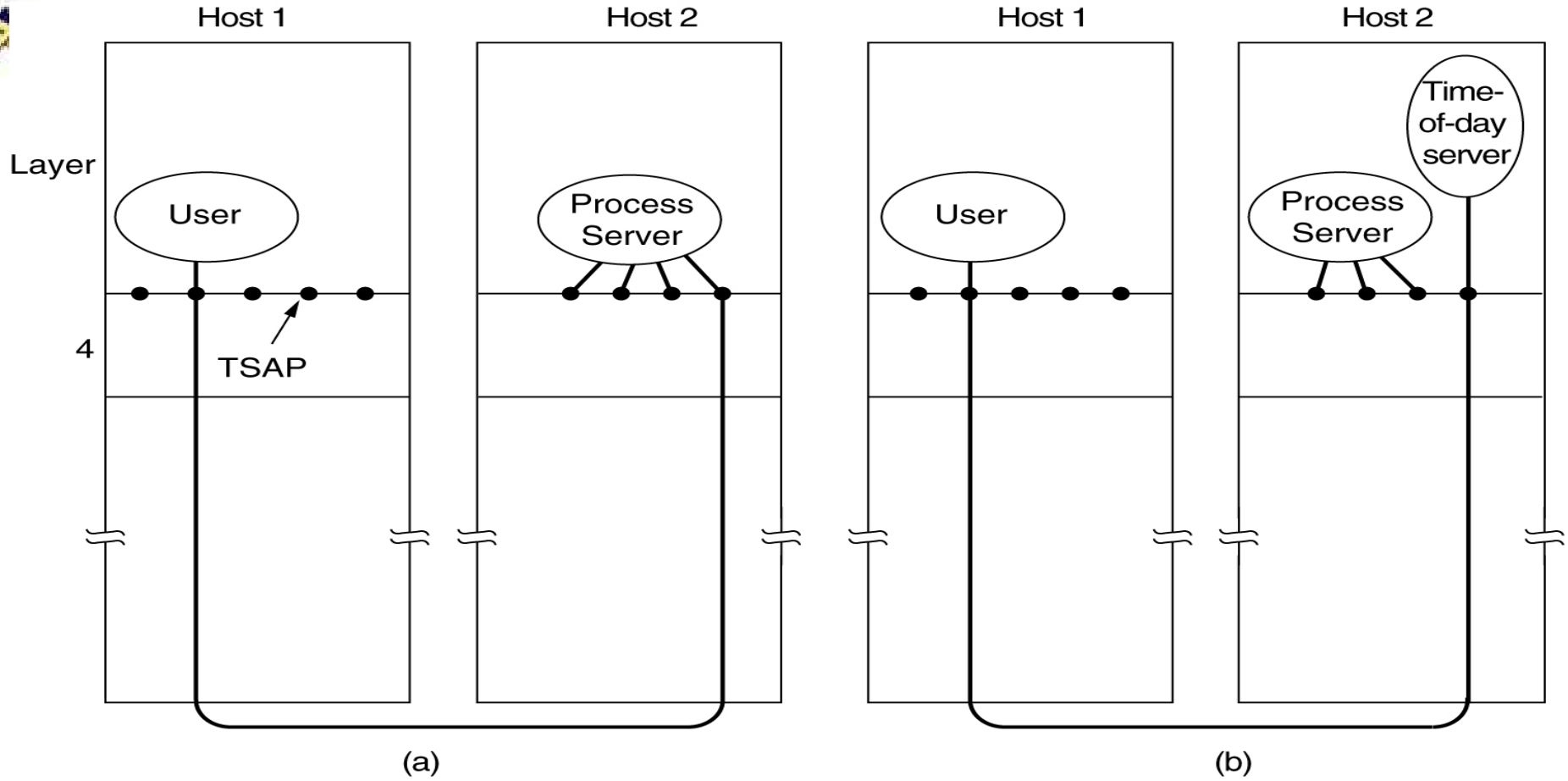
- Addressing
- Connection Establishment
- Connection Release
- Flow Control and Buffering
- Multiplexing
- Crash Recovery

Addressing



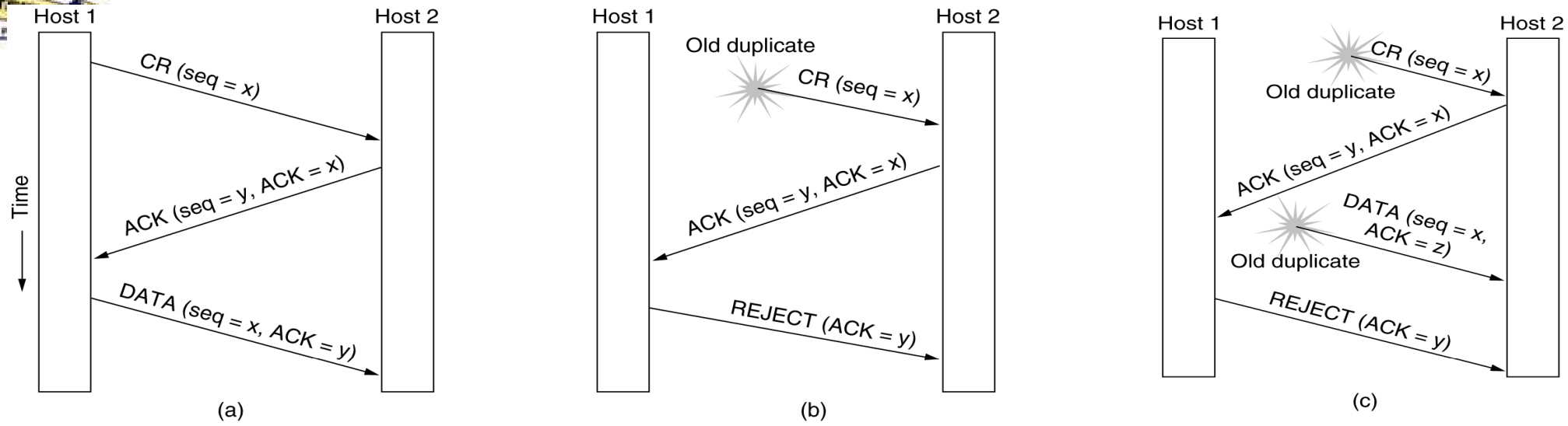
TSAPs, NSAPs and transport connections.

Connection Establishment



How a user process in host 1 establishes a connection with a time-of-day server in host 2.

Connection Establishment (3)



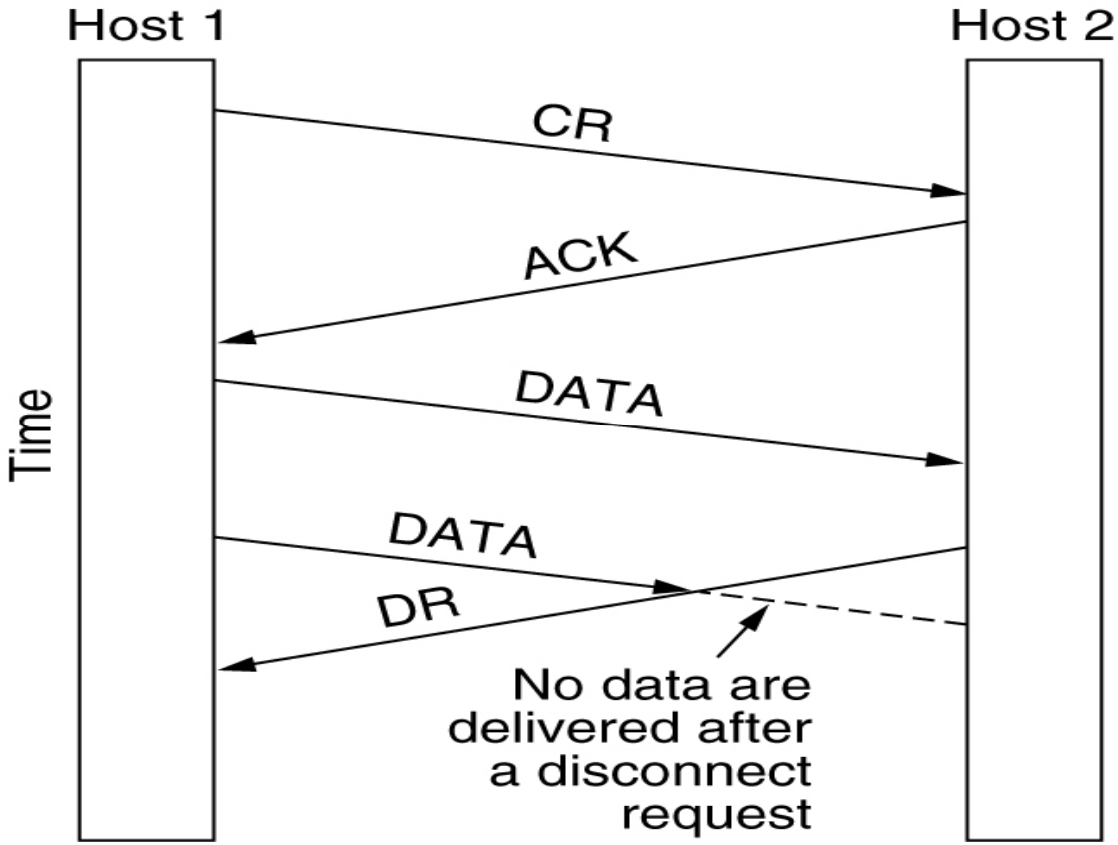
Three protocol scenarios for establishing a connection using a three-way handshake. CR denotes CONNECTION REQUEST.

(a) Normal operation,

(b) Old CONNECTION REQUEST appearing out of nowhere.

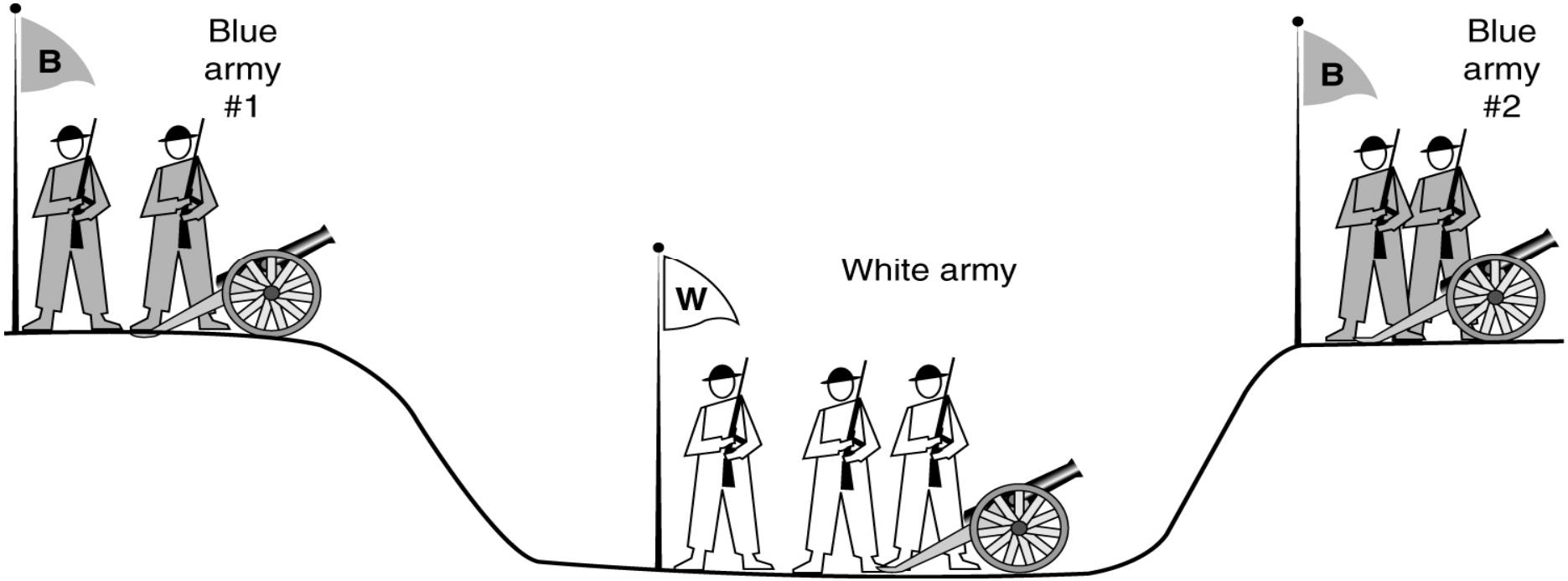
(c) Duplicate CONNECTION REQUEST and duplicate ACK.

Connection Release



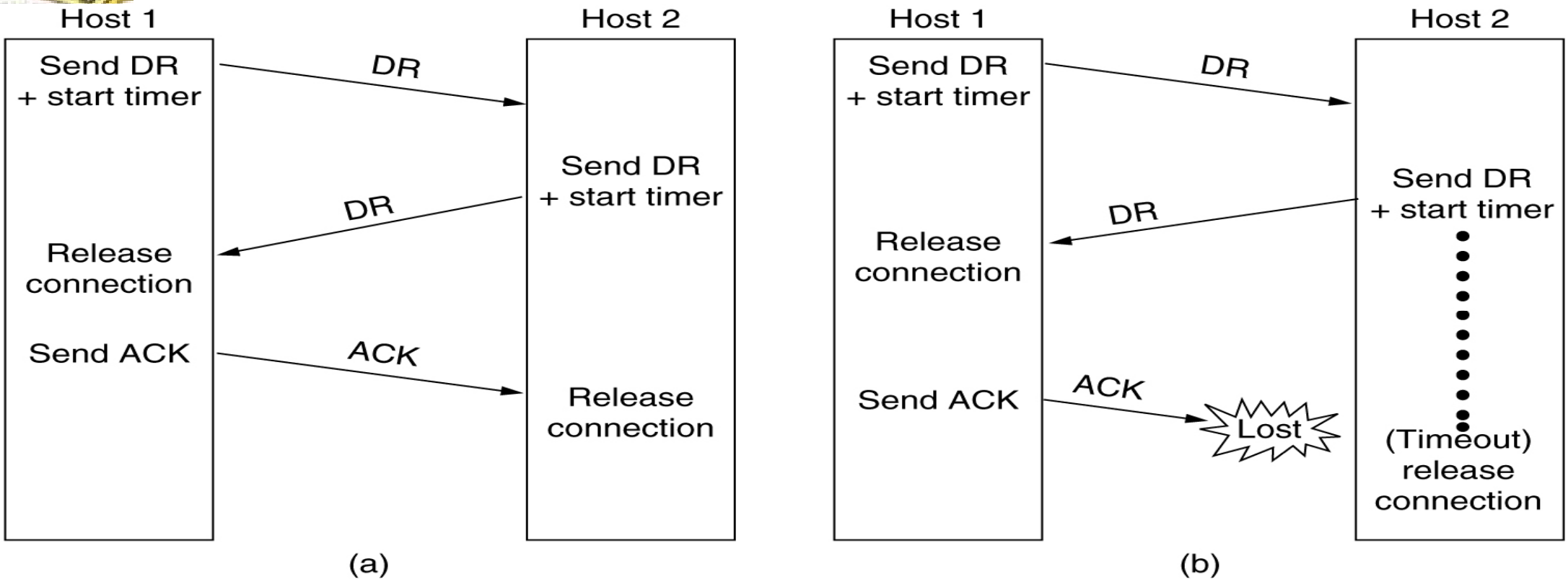
Abrupt disconnection with loss of data.

Connection Release (2)



The two-army problem.

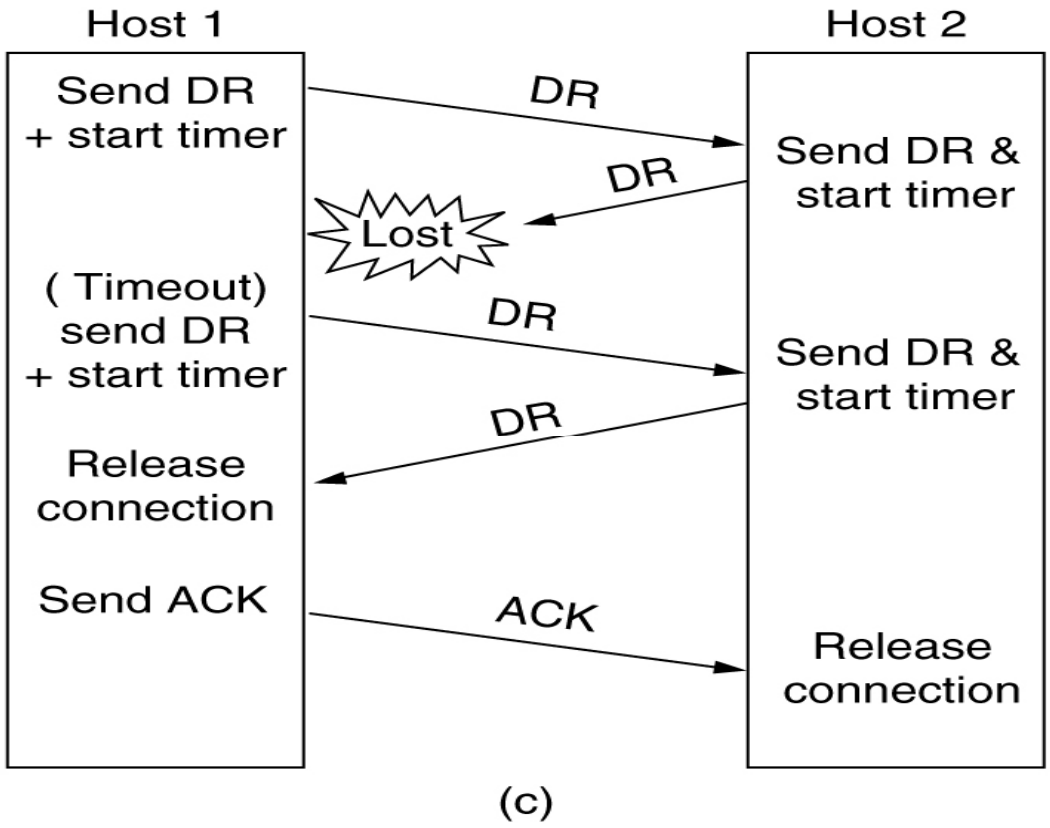
Connection Release (3)

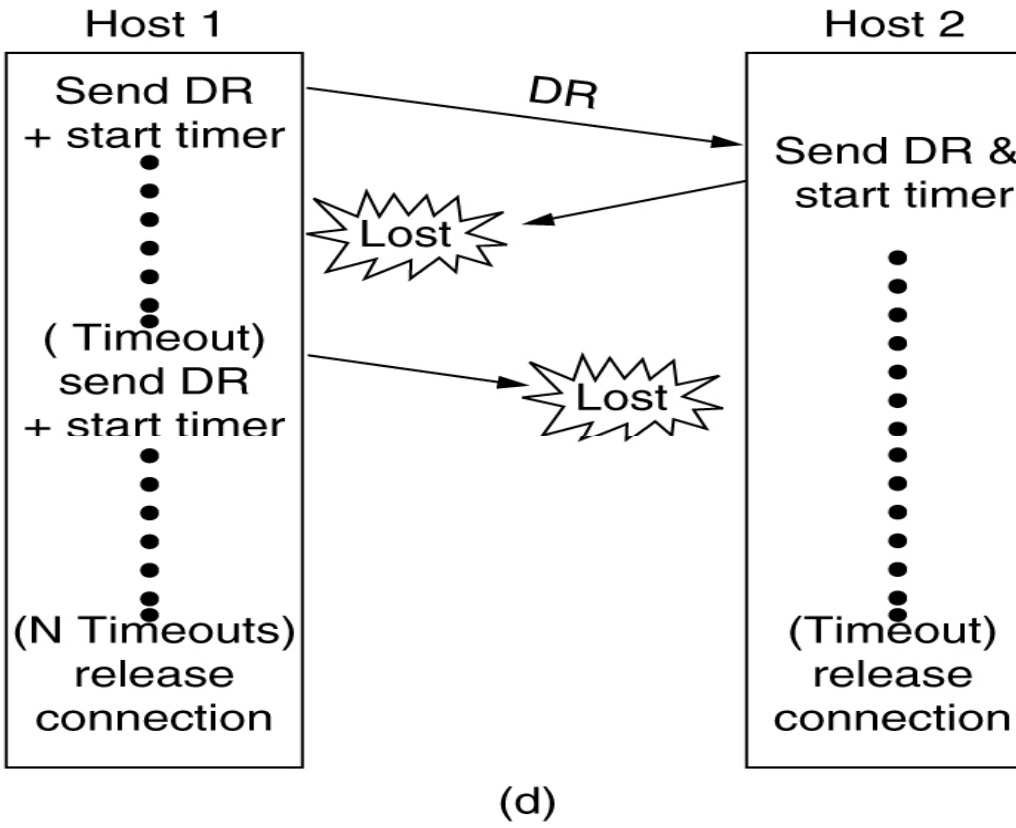


Four protocol scenarios for releasing a connection.

(a) Normal case of a three-way handshake. (b) final ACK lost.

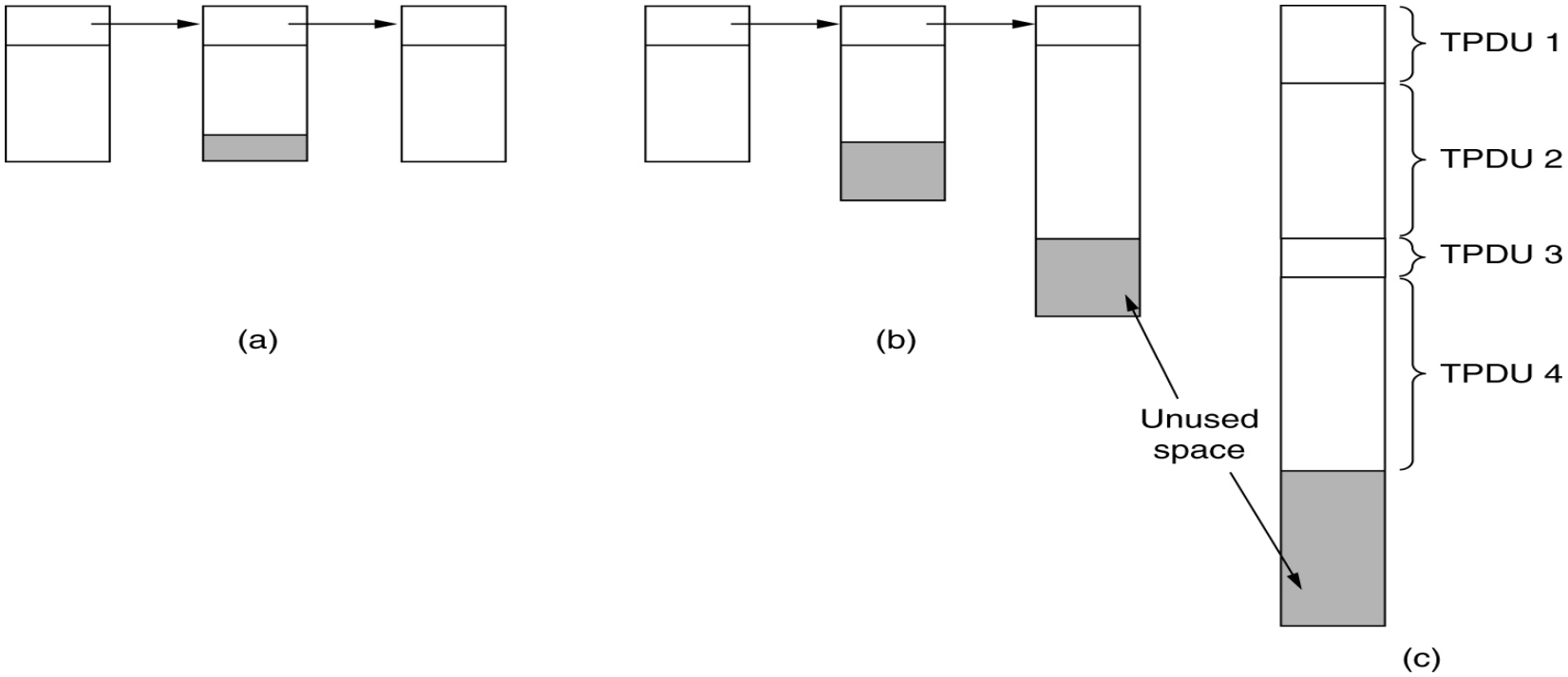
Connection Release (4)





(c) Response lost. (d) Response lost and subsequent DRs lost.

Flow Control and Buffering



(a) Chained fixed-size buffers. (b) Chained variable-sized buffers.
(c) One large circular buffer per connection.

Flow Control and Buffering (2)

	A	Message	B	Comments
1	→	< request 8 buffers>	→	A wants 8 buffers
2	←	<ack = 15, buf = 4>	←	B grants messages 0-3 only
3	→	<seq = 0, data = m0>	→	A has 3 buffers left now
4	→	<seq = 1, data = m1>	→	A has 2 buffers left now
5	→	<seq = 2, data = m2>	• • •	Message lost but A thinks it has 1 left
6	←	<ack = 1, buf = 3>	←	B acknowledges 0 and 1, permits 2-4
7	→	<seq = 3, data = m3>	→	A has 1 buffer left
8	→	<seq = 4, data = m4>	→	A has 0 buffers left, and must stop
9	→	<seq = 2, data = m2>	→	A times out and retransmits
10	←	<ack = 4, buf = 0>	←	Everything acknowledged, but A still blocked
11	←	<ack = 4, buf = 1>	←	A may now send 5
12	←	<ack = 4, buf = 2>	←	B found a new buffer somewhere
13	→	<seq = 5, data = m5>	→	A has 1 buffer left
14	→	<seq = 6, data = m6>	→	A is now blocked again
15	←	<ack = 6, buf = 0>	←	A is still blocked
16	• • •	<ack = 6, buf = 4>	←	Potential deadlock

Dynamic buffer allocation. The arrows show the direction of transmission. An ellipsis (…) indicates a lost TPDU.

Crash Recovery

Strategy used by sending host	Strategy used by receiving host: First ACK, then write			Strategy used by receiving host: First write, then ACK		
	AC(W)	AWC	C(AW)	C(WA)	W AC	WC(A)
Always retransmit	OK	DUP	OK	OK	DUP	DUP
Never retransmit	LOST	OK	LOST	LOST	OK	OK
Retransmit in S0	OK	DUP	LOST	LOST	DUP	OK
Retransmit in S1	LOST	OK	OK	OK	OK	DUP

OK = Protocol functions correctly
DUP = Protocol generates a duplicate message
LOST = Protocol loses a message

Different combinations of client and server strategy.