


Computer Networks

CMSC 417 : Spring 2024

COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

Topic: Internetworking: ICMP
&
Transport Layer Protocols (UDP, TCP)
(Textbook chapter 5)

Nirupam Roy

Tu-Th 2:00-3:15pm
CSI 2117

March 5th, 2024




Internet Control Message Protocol (ICMP)

Internet Control Message Protocol (ICMP)

- Defines a collection of error messages that are sent back to the source host.

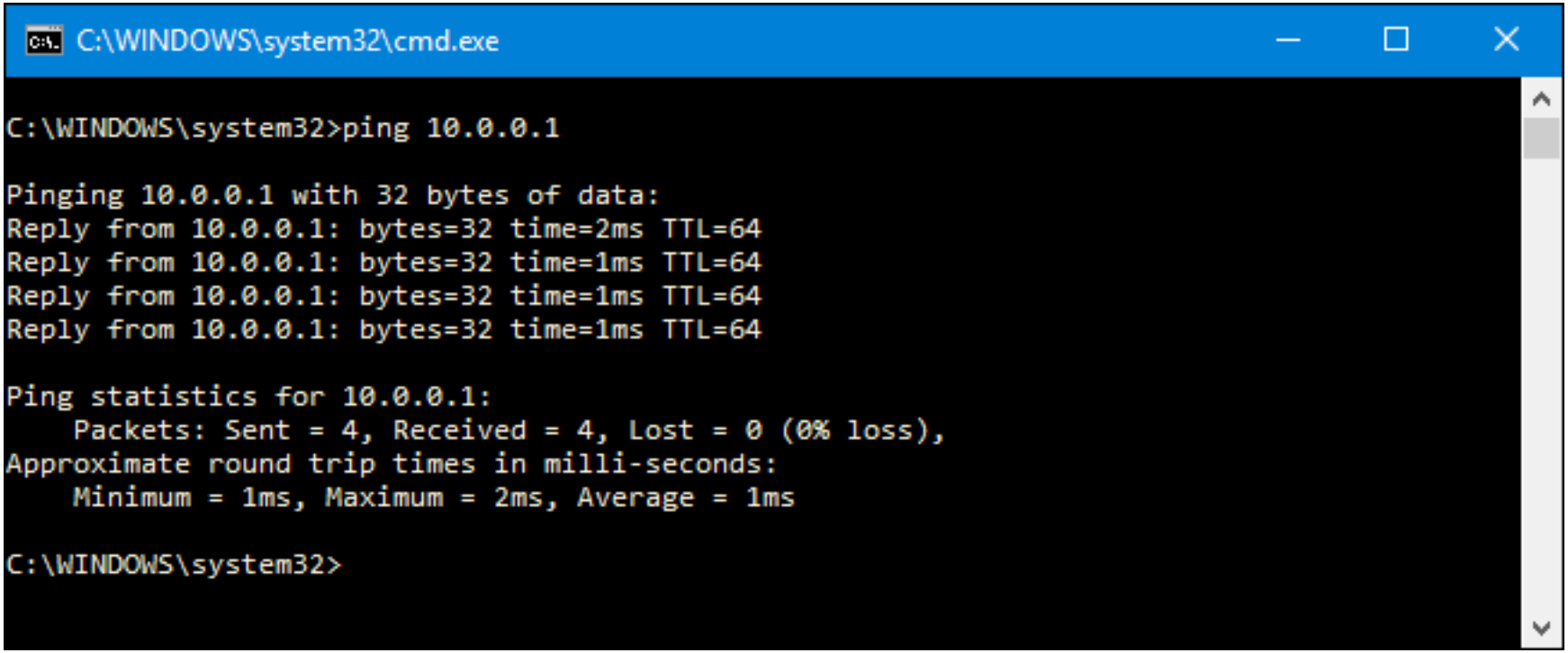
- E.g., ICMP messages are sent whenever a router or host is unable to process an IP datagram successfully
  - Destination host unreachable due to link /node failure
  - Reassembly process failed
  - TTL had reached 0 (so datagrams don't cycle forever)
  - IP header checksum failed

- ICMP-Redirect
  - From router to a source host
  - With a better route information

ICMP message types

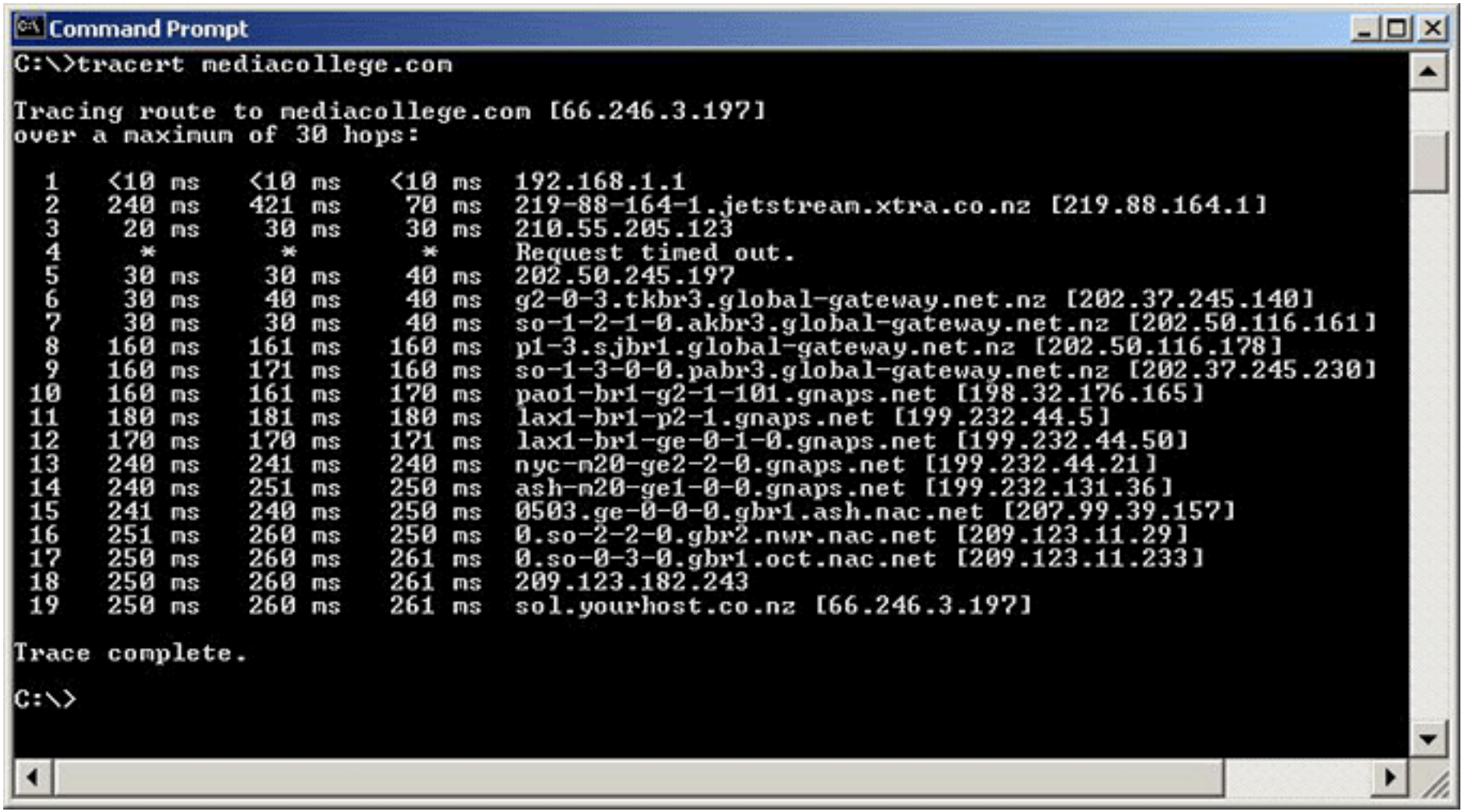
ICMP Type	Code	Description
0	0	echo reply (to ping)
3	0	destination network unreachable
3	1	destination host unreachable
3	2	destination protocol unreachable
3	3	destination port unreachable
3	6	destination network unknown
3	7	destination host unknown
4	0	source quench (congestion control)
8	0	echo request
9	0	router advertisement
10	0	router discovery
11	0	TTL expired
12	0	IP header bad

Ping example

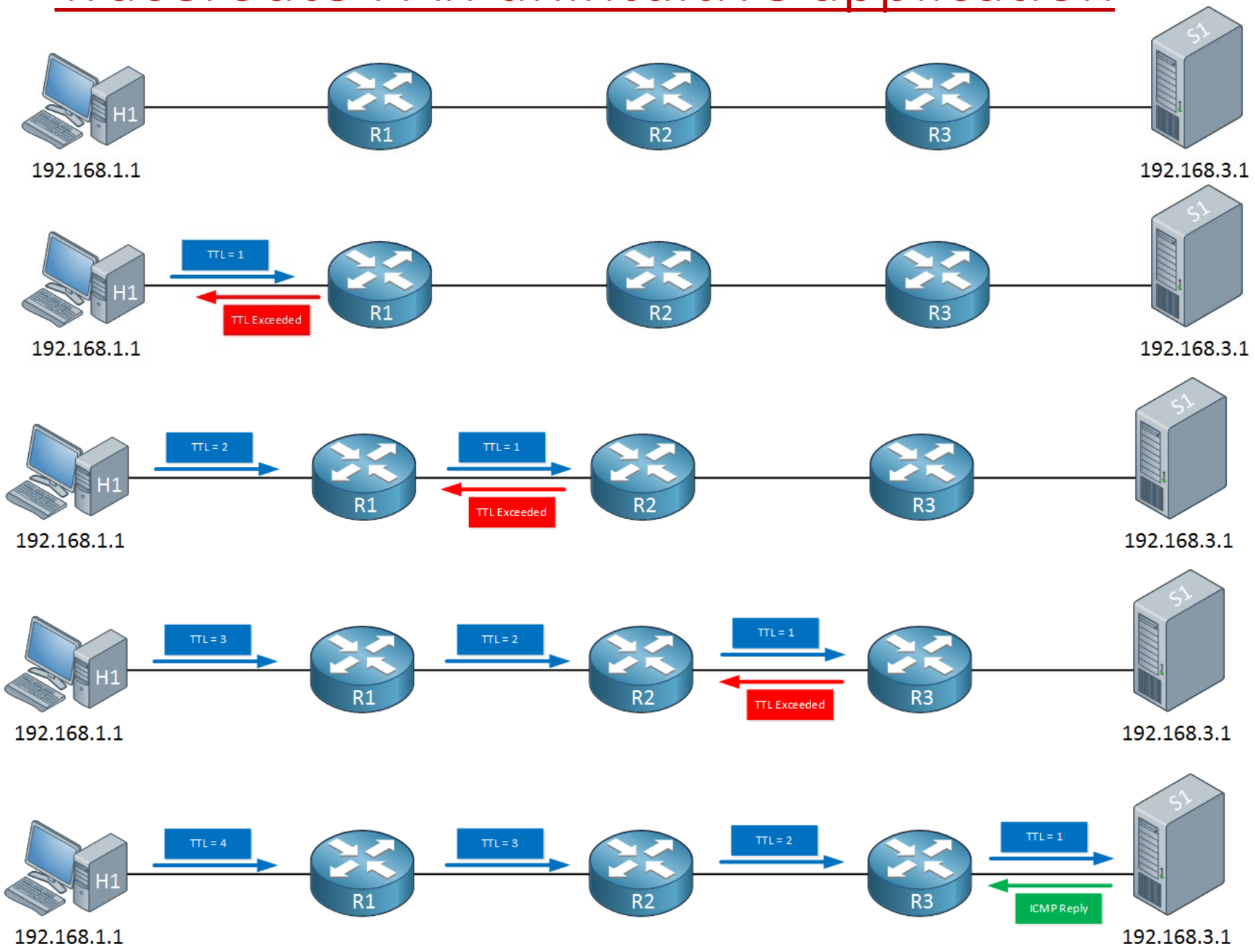


Traceroute : An unintuitive application using ICMP

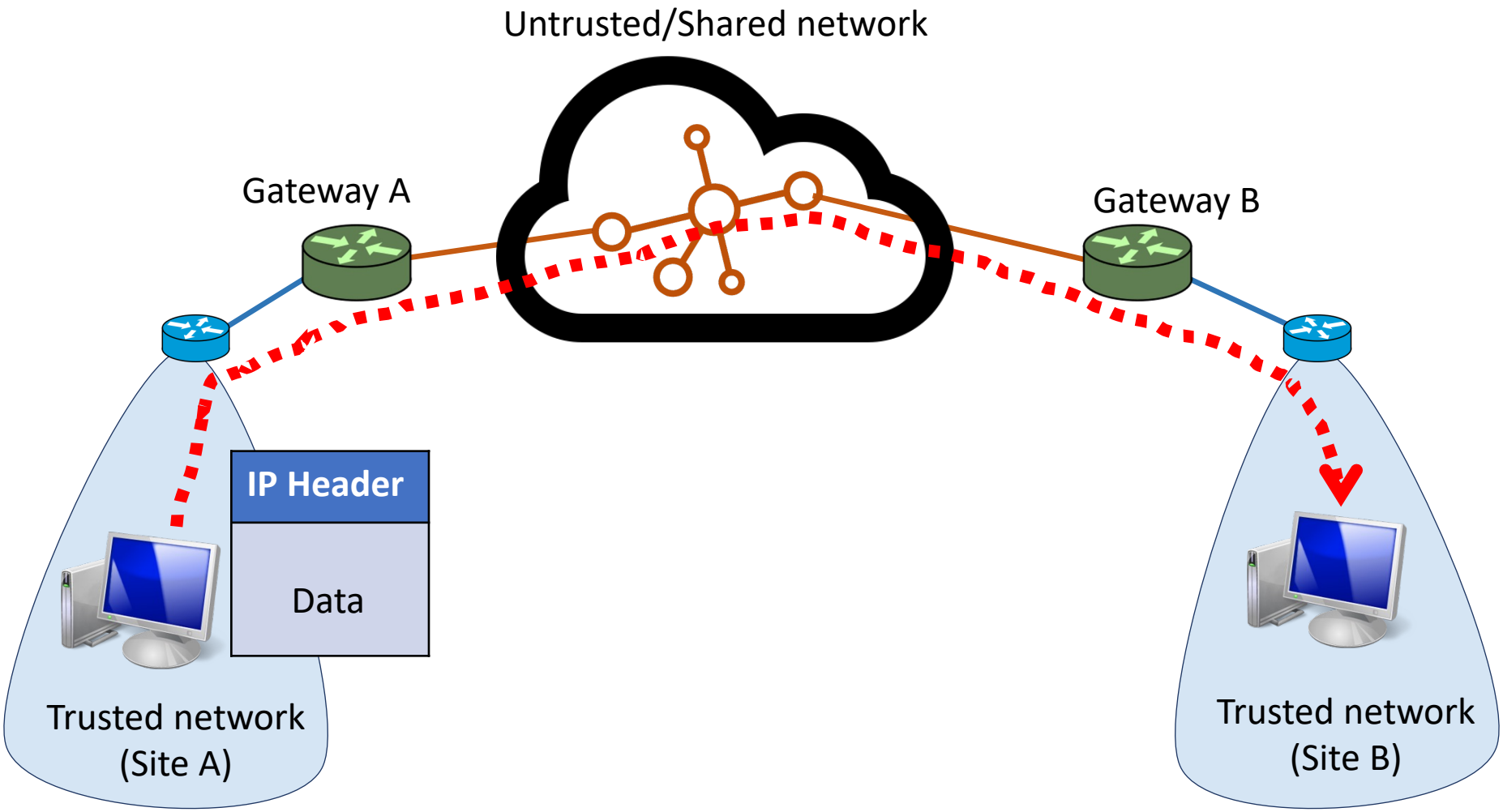
Traceroute example

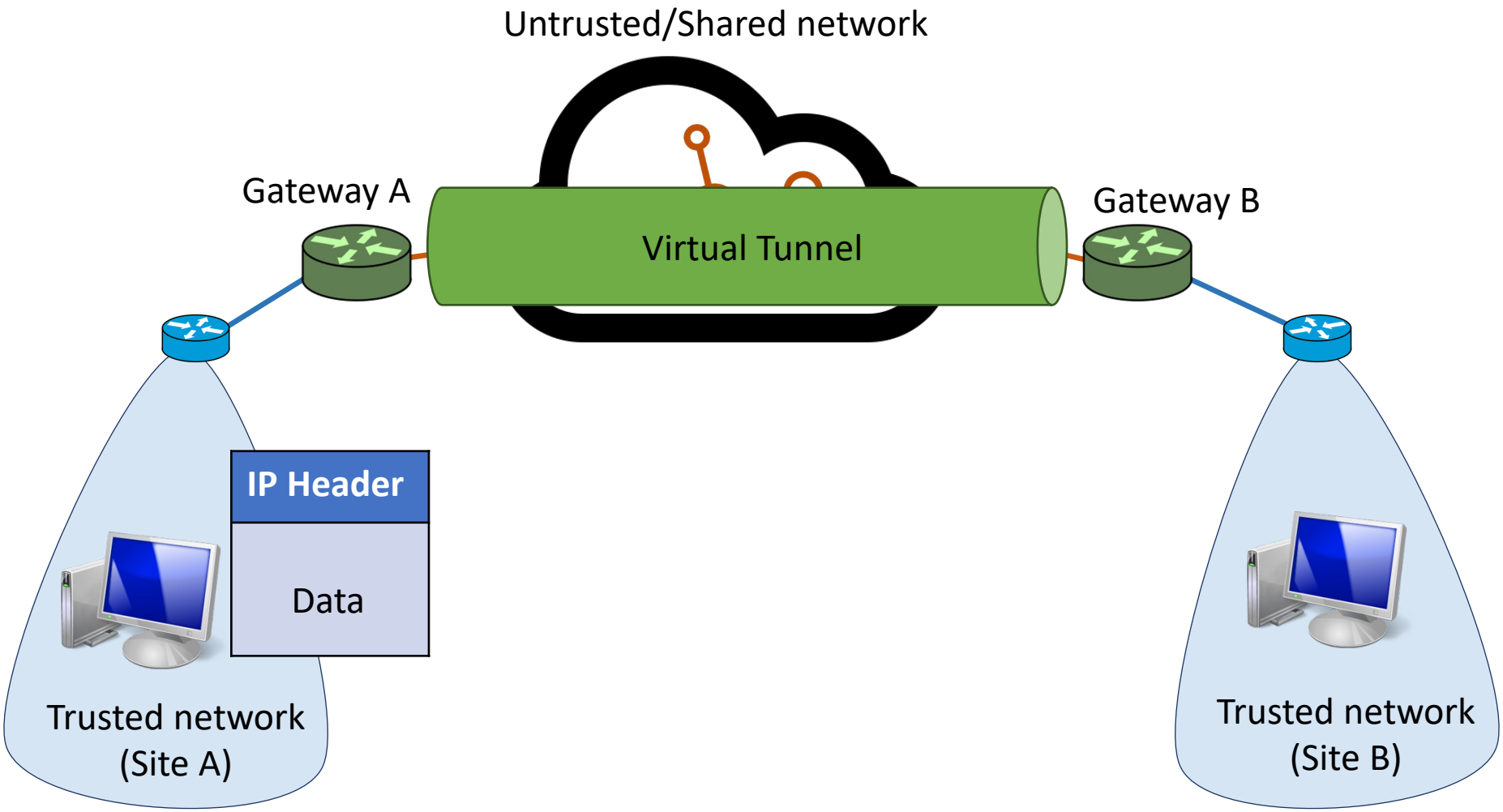


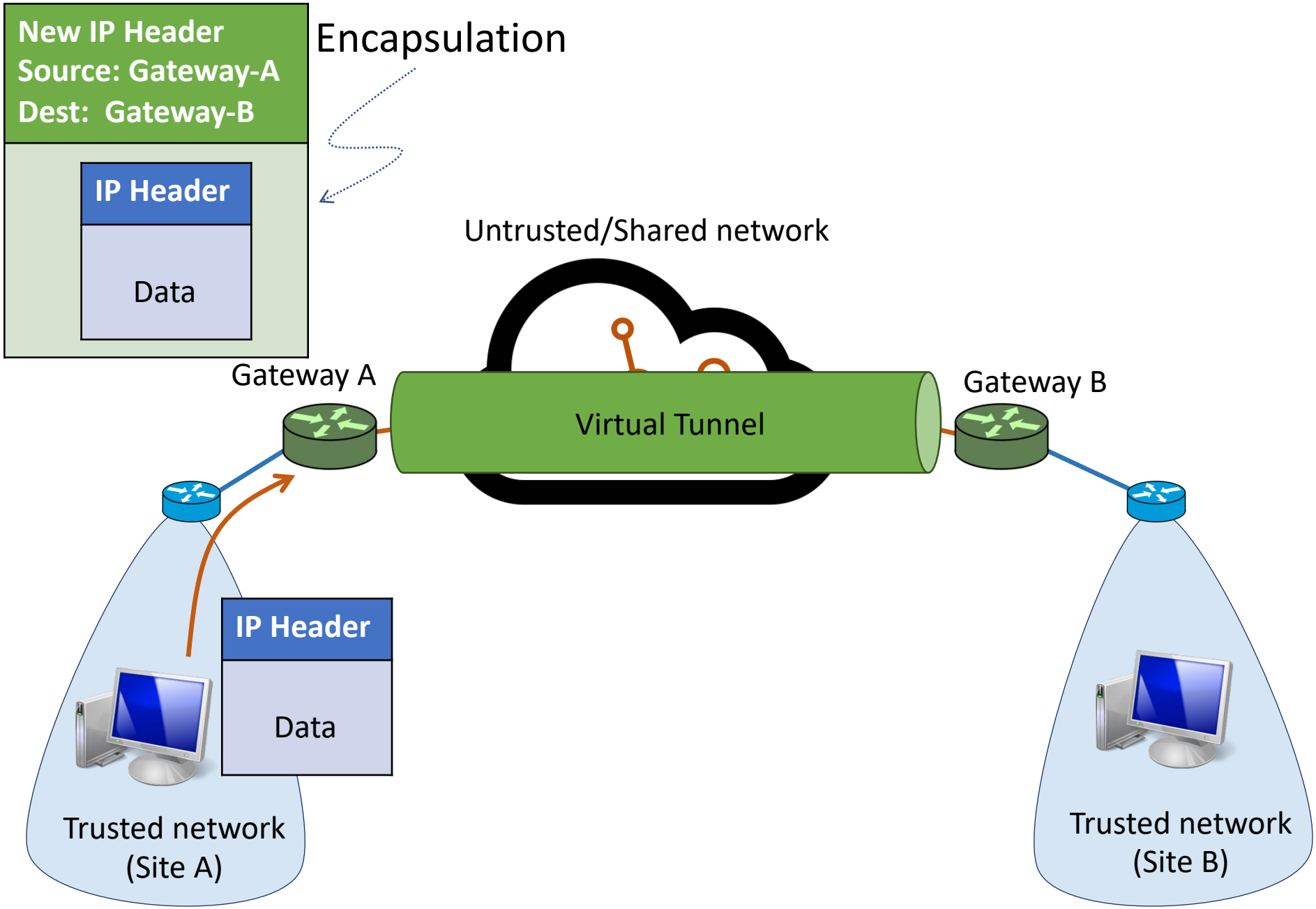
Traceroute : An unintuitive application

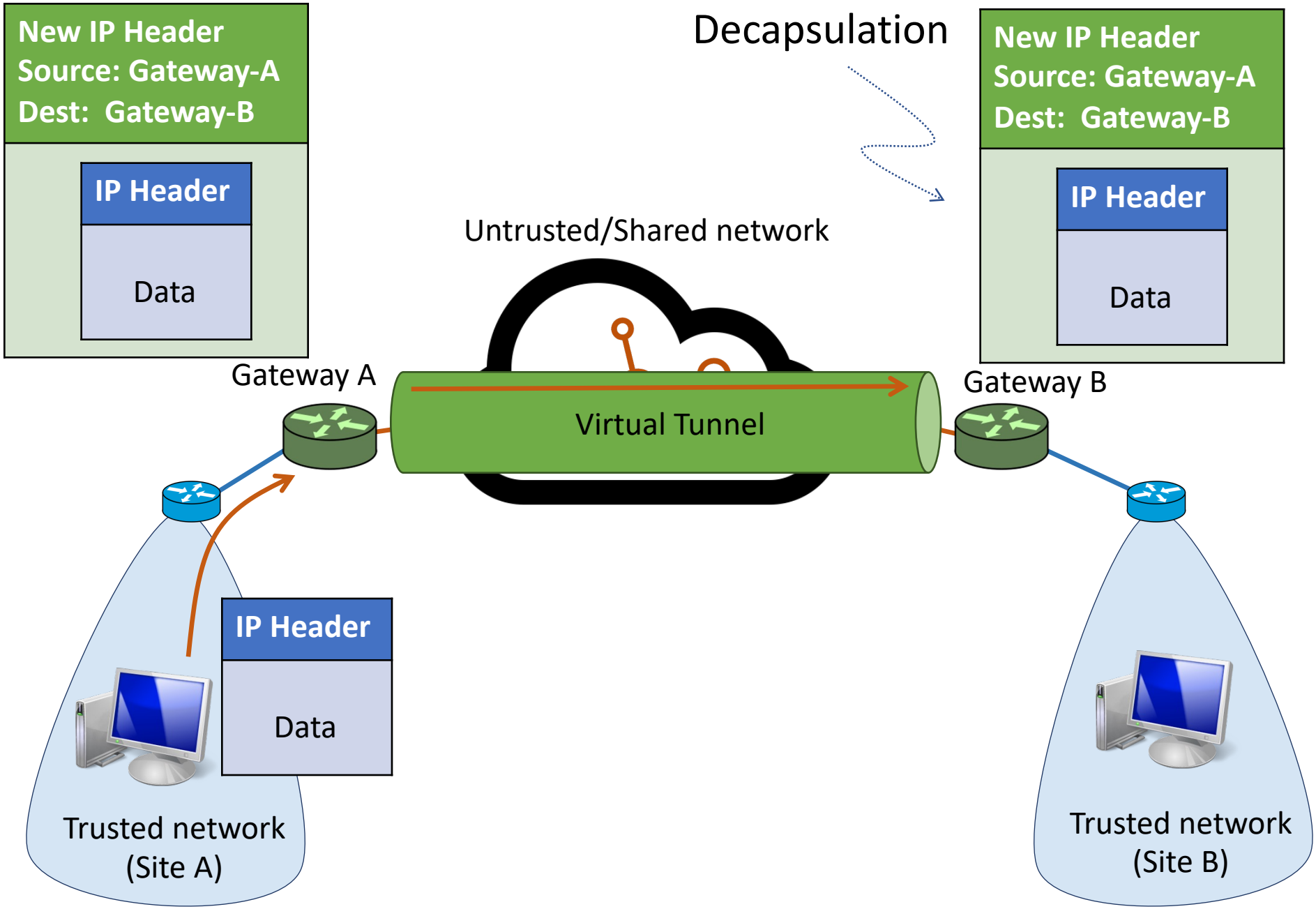


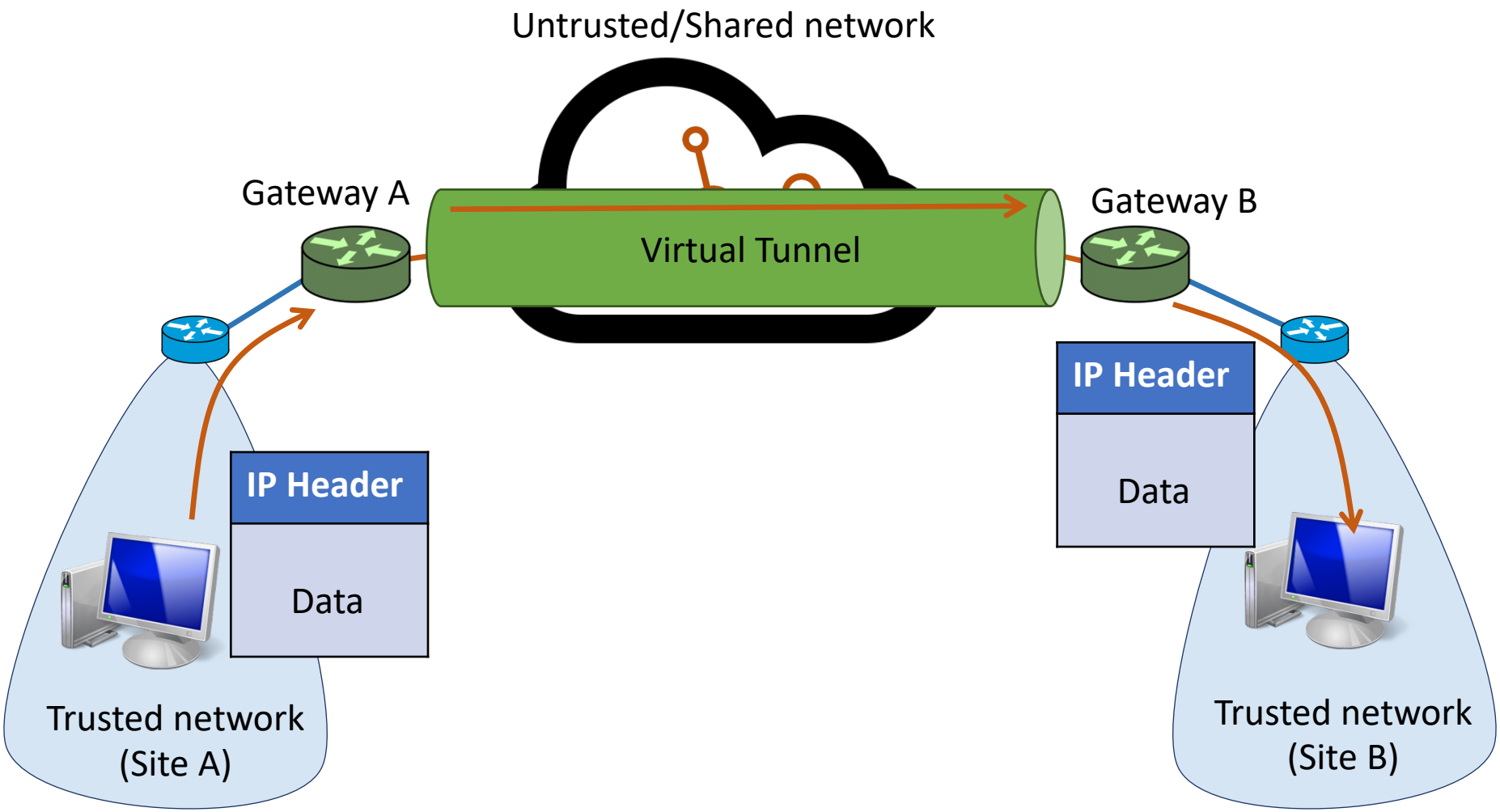
Virtual Networks and Tunnels

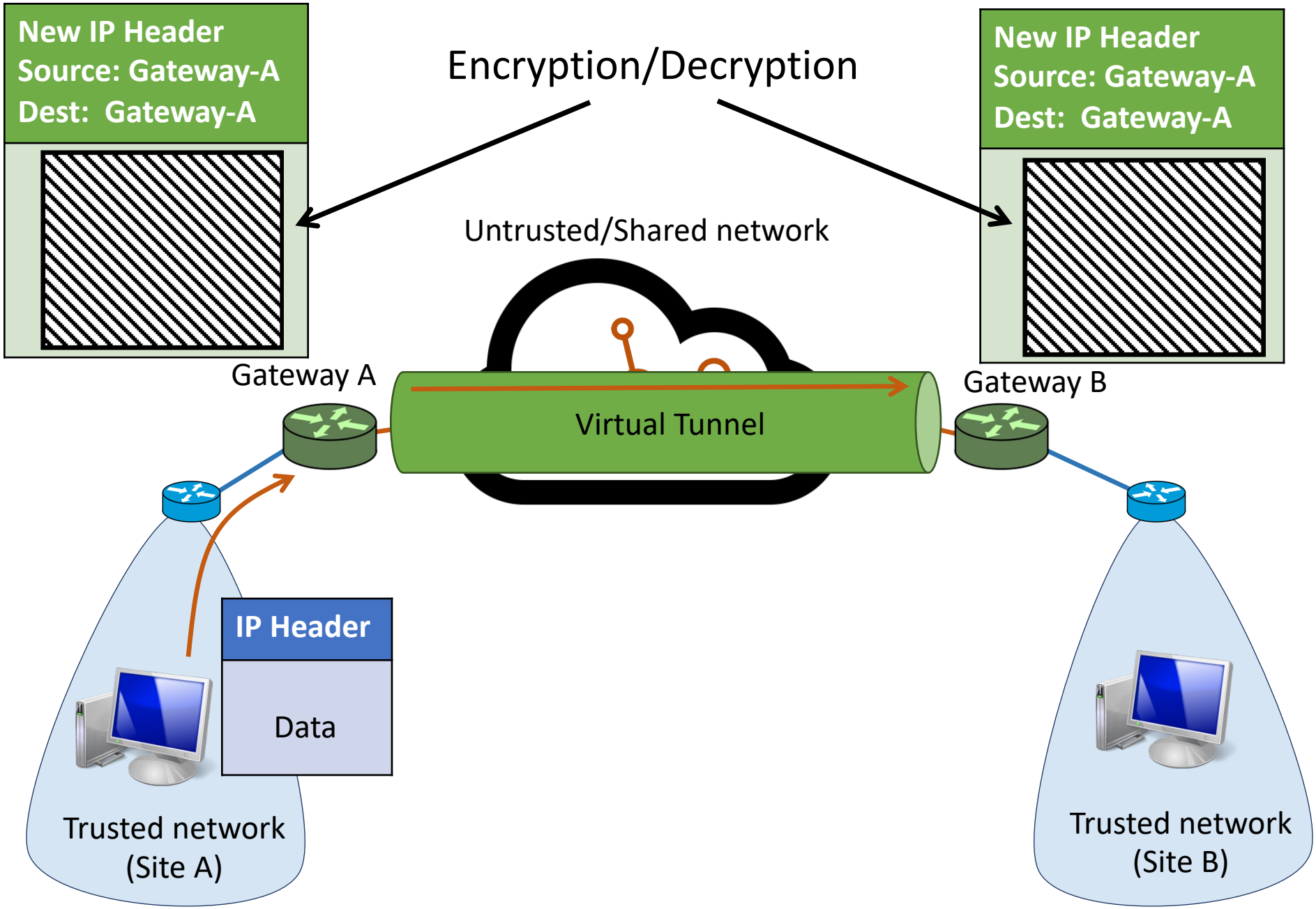

Untrusted/Shared network
Gateway A
Gateway B
IP Header
Data
Trusted network
(Site A)
Trusted network
(Site B)

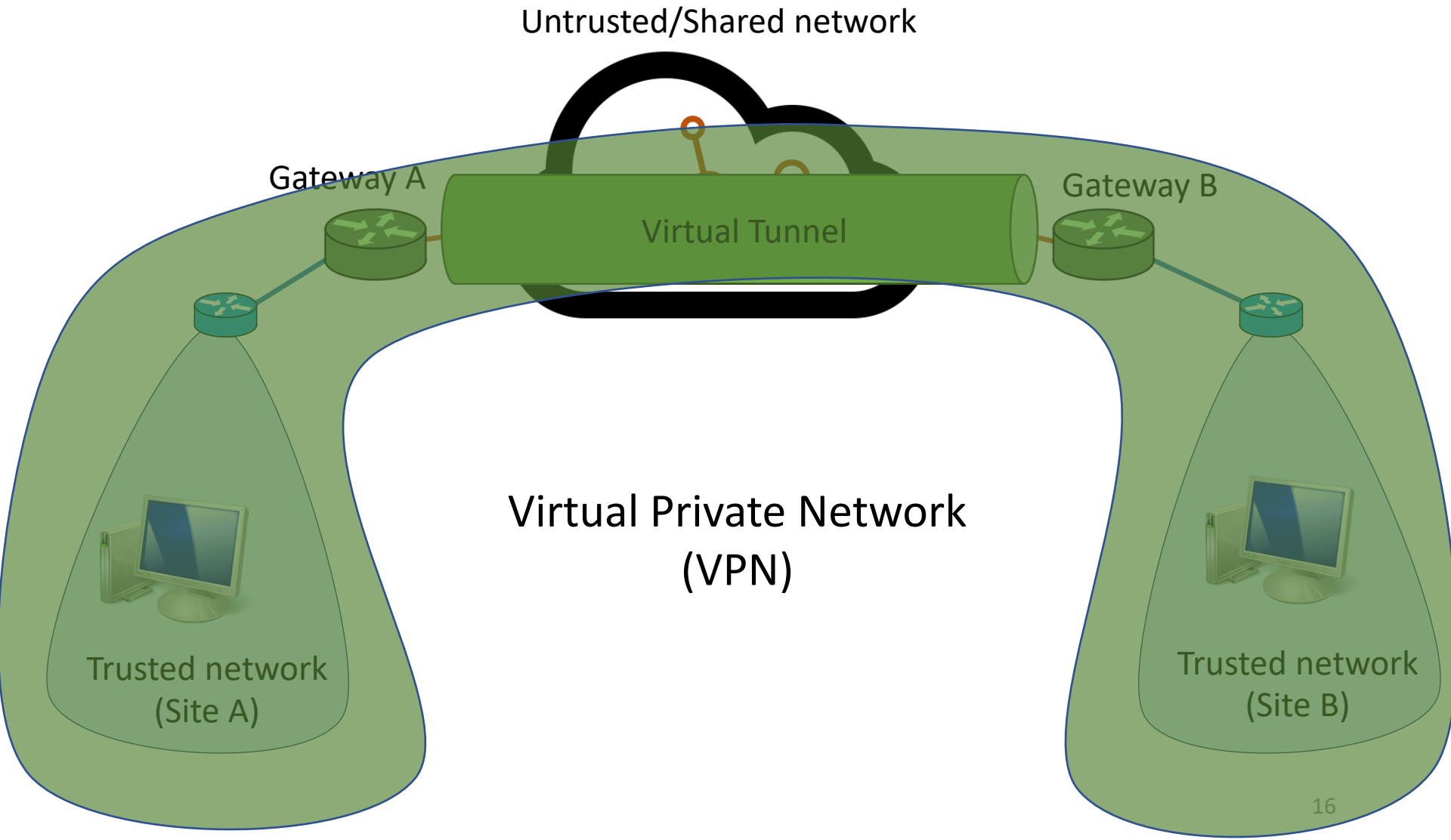

Untrusted/Shared network
Gateway A
Virtual Tunnel
Gateway B
IP Header
Data
Trusted network
(Site A)
Trusted network
(Site B)


New IP Header
Source: Gateway-A
Dest: Gateway-B
IP Header
Data
Encapsulation
Untrusted/Shared network
Gateway A
Virtual Tunnel
Gateway B
IP Header
Data
Trusted network
(Site A)
Trusted network
(Site B)


New IP Header
Source: Gateway-A
Dest: Gateway-B
IP Header
Data
Decapsulation
New IP Header
Source: Gateway-A
Dest: Gateway-B
IP Header
Data
Untrusted/Shared network
Gateway A
Virtual Tunnel
Gateway B
IP Header
Data
Trusted network
(Site A)
Trusted network
(Site B)


Untrusted/Shared network
Gateway A
Virtual Tunnel
Gateway B
IP Header
Data
IP Header
Data
Trusted network
(Site A)
Trusted network
(Site B)


New IP Header
Source: Gateway-A
Dest: Gateway-A
Encryption/Decryption
New IP Header
Source: Gateway-A
Dest: Gateway-A
Untrusted/Shared network
Gateway A
Virtual Tunnel
Gateway B
IP Header
Data
Trusted network
(Site A)
Trusted network
(Site B)


Untrusted/Shared network
Gateway A
Virtual Tunnel
Gateway B
Virtual Private Network
(VPN)
Trusted network
(Site A)
Trusted network
(Site B)

Why do we need virtual networks or tunnels?

1. Security
2. Special capabilities between routers (e.g., multicast)
3. Supporting heterogeneity

Why do we need virtual networks or tunnels?

1. Security
2. Special capabilities between routers (e.g., multicast)
3. Supporting heterogeneity

Disadvantages:

1. Increases packet length
   a) Wastage of bandwidth
   b) More processing
   c) Fragmentation
2. Increases management cost

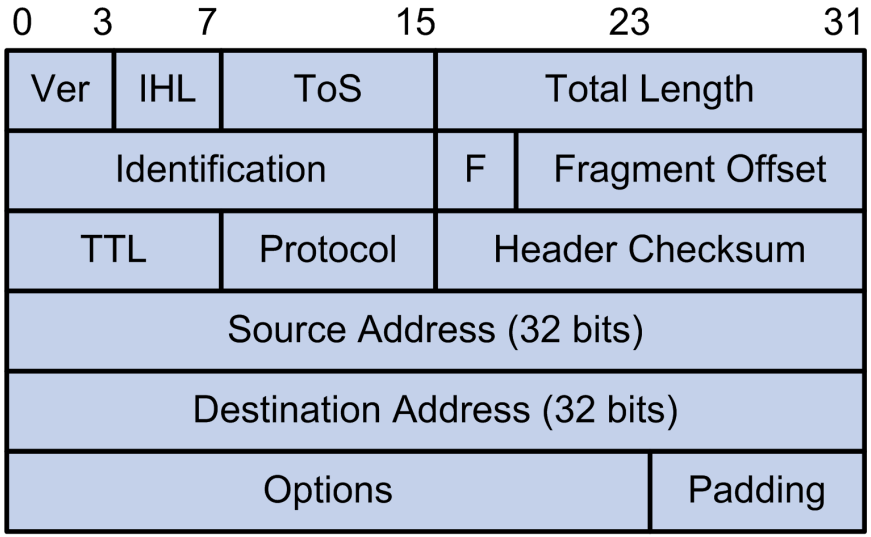
IPv6

1) Address and packet format

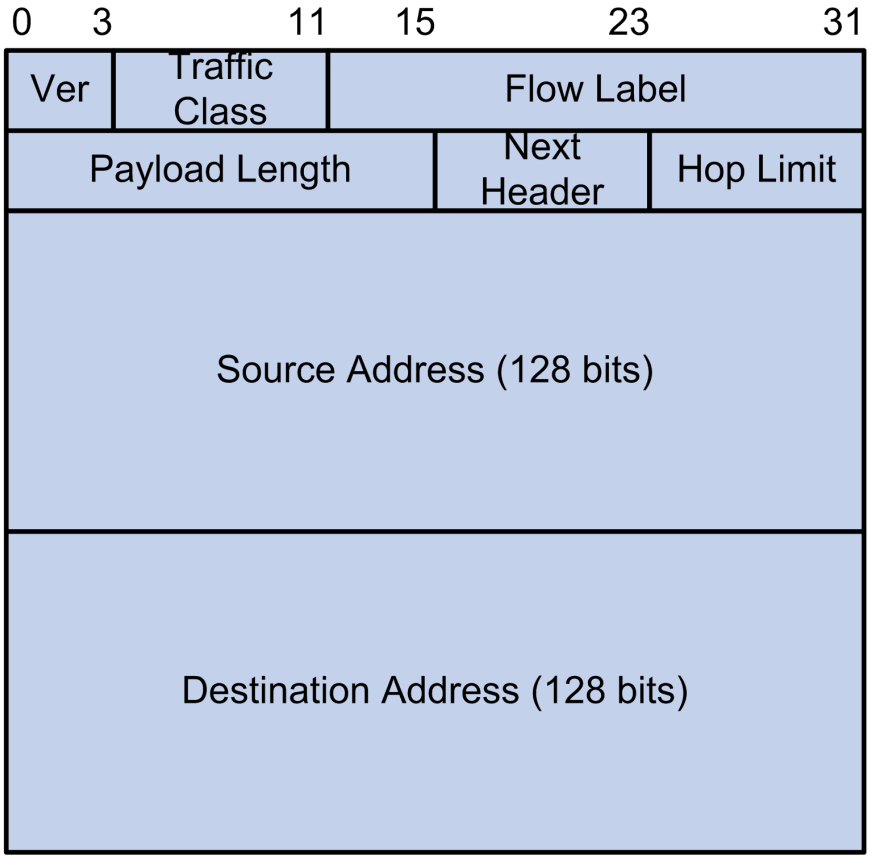
2) Unicast, Multicast, Anycast addressing

3) IPV4 and IPV6 coexistence

IPv6: Address and packet format



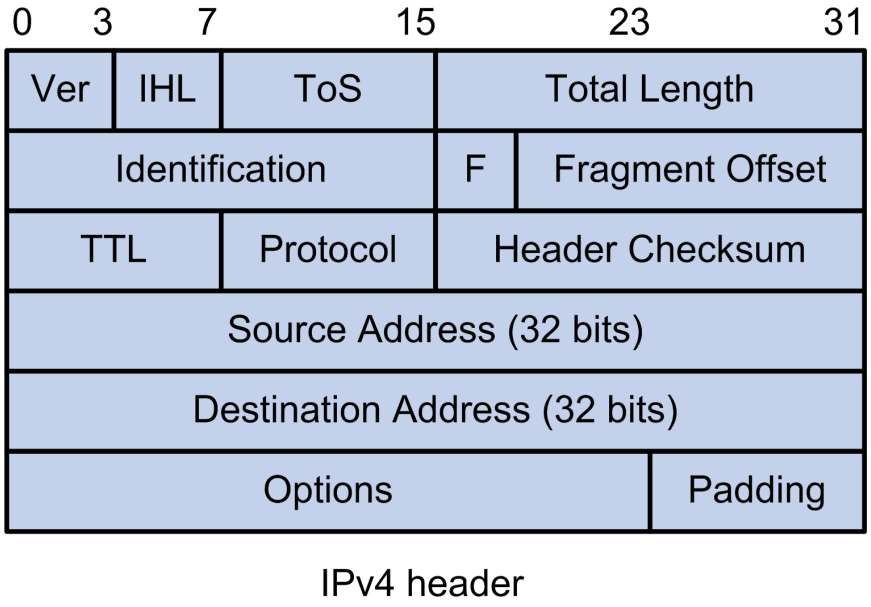
IPv4 header



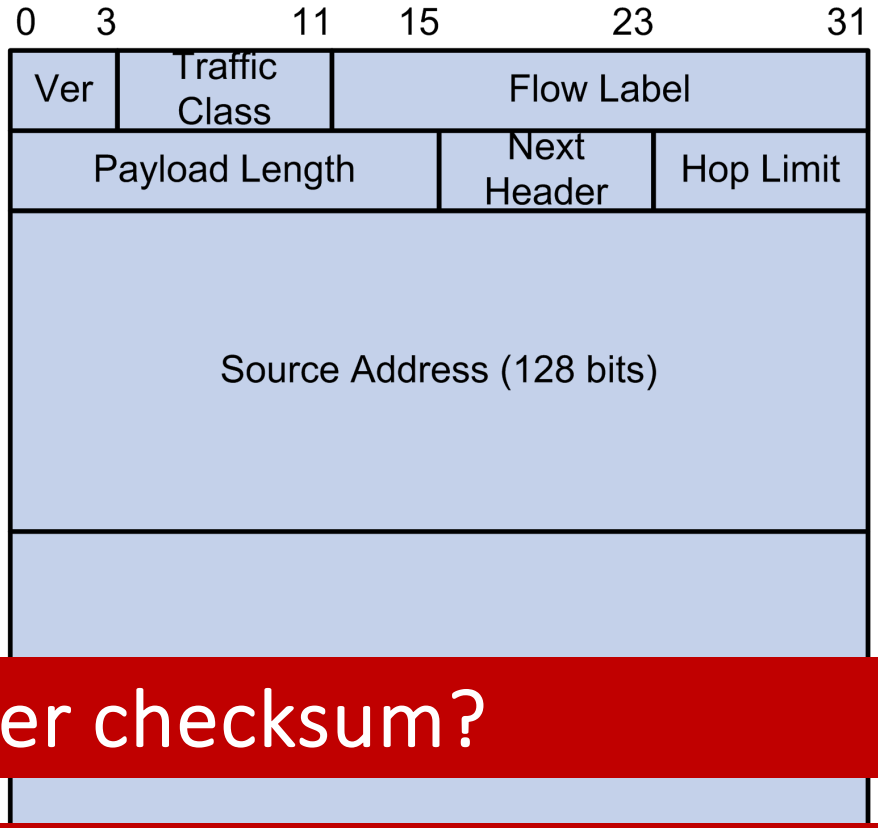
Basic IPv6 header

Pic courtesy: https://techhub.hpe.com

IPv6: Address and packet format



IPv4 header



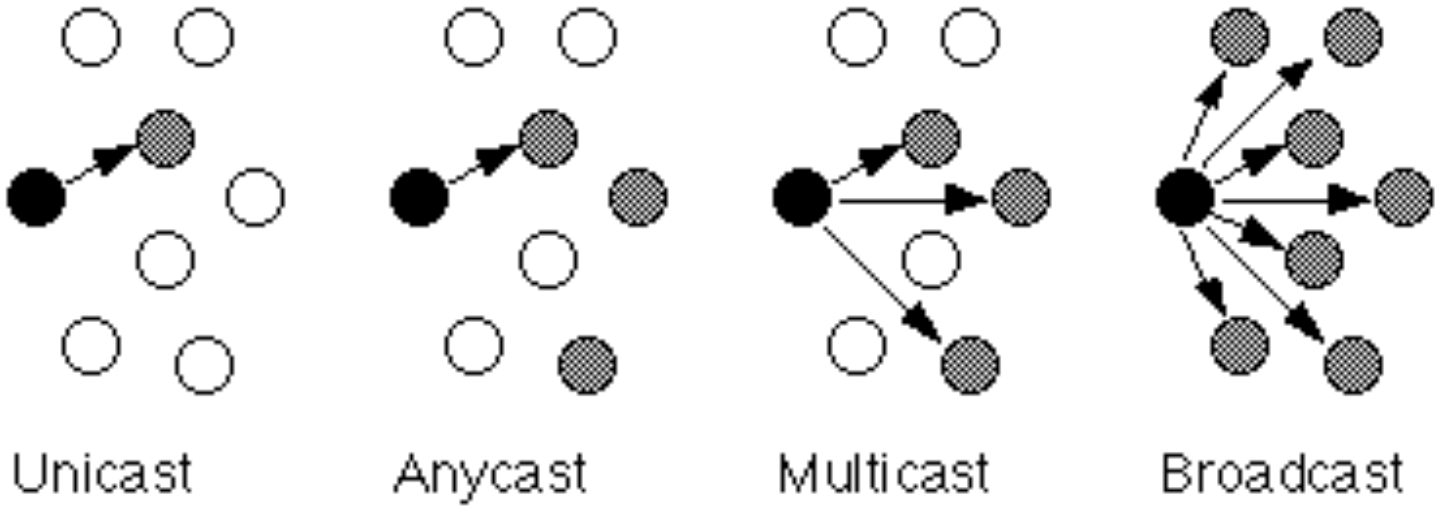
Q 1. Why is there no header checksum?

Q 2. Why is there no fragmenting related option?

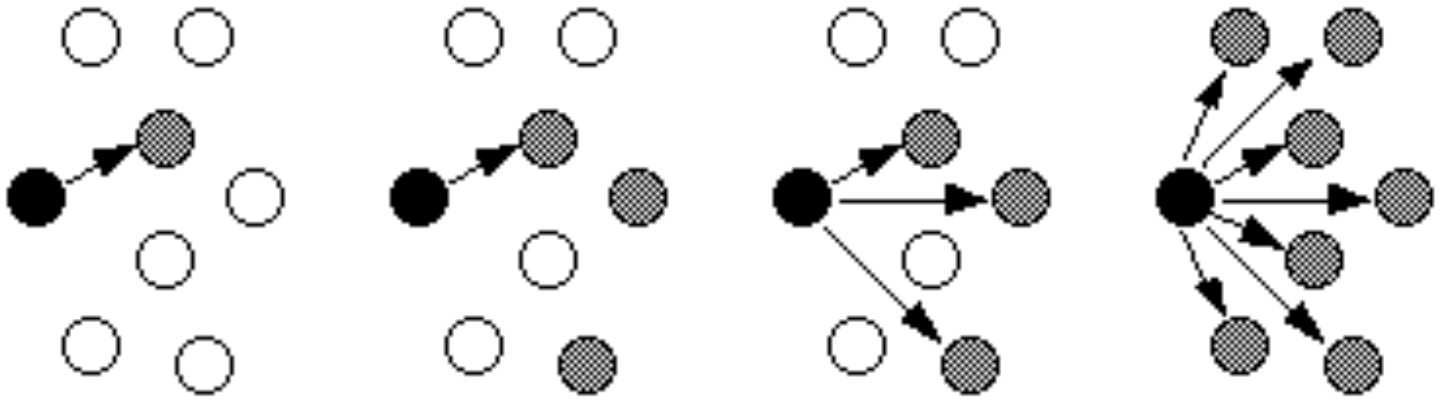
Q 3. Is 128-bit IP address big enough?

Pic courtesy: https://techhub.hpe.com

Unicast, Multicast, Anycast, Broadcast

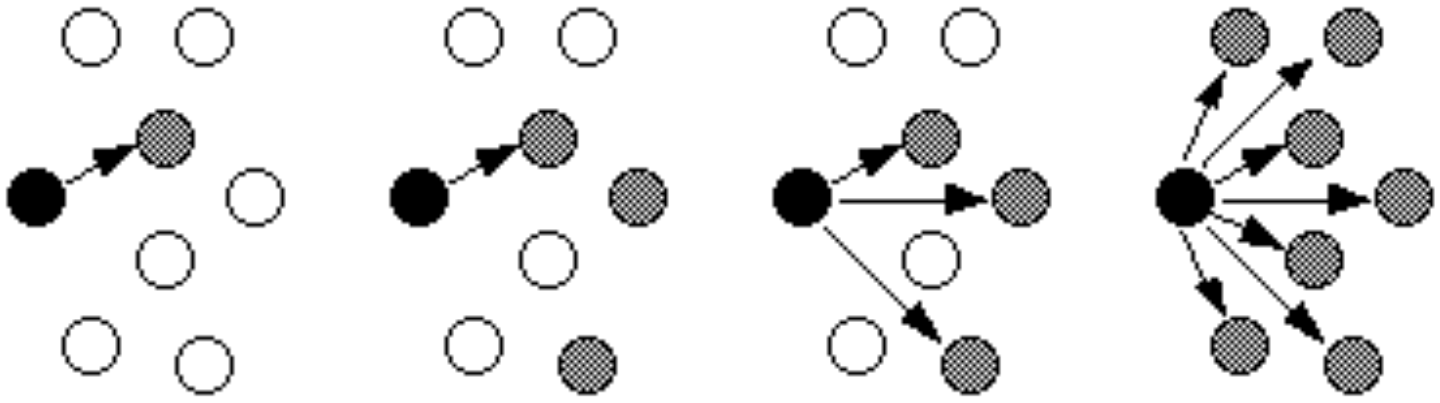


Unicast, Multicast, Anycast, Broadcast



IPv4 **IPv4** **IPv4**

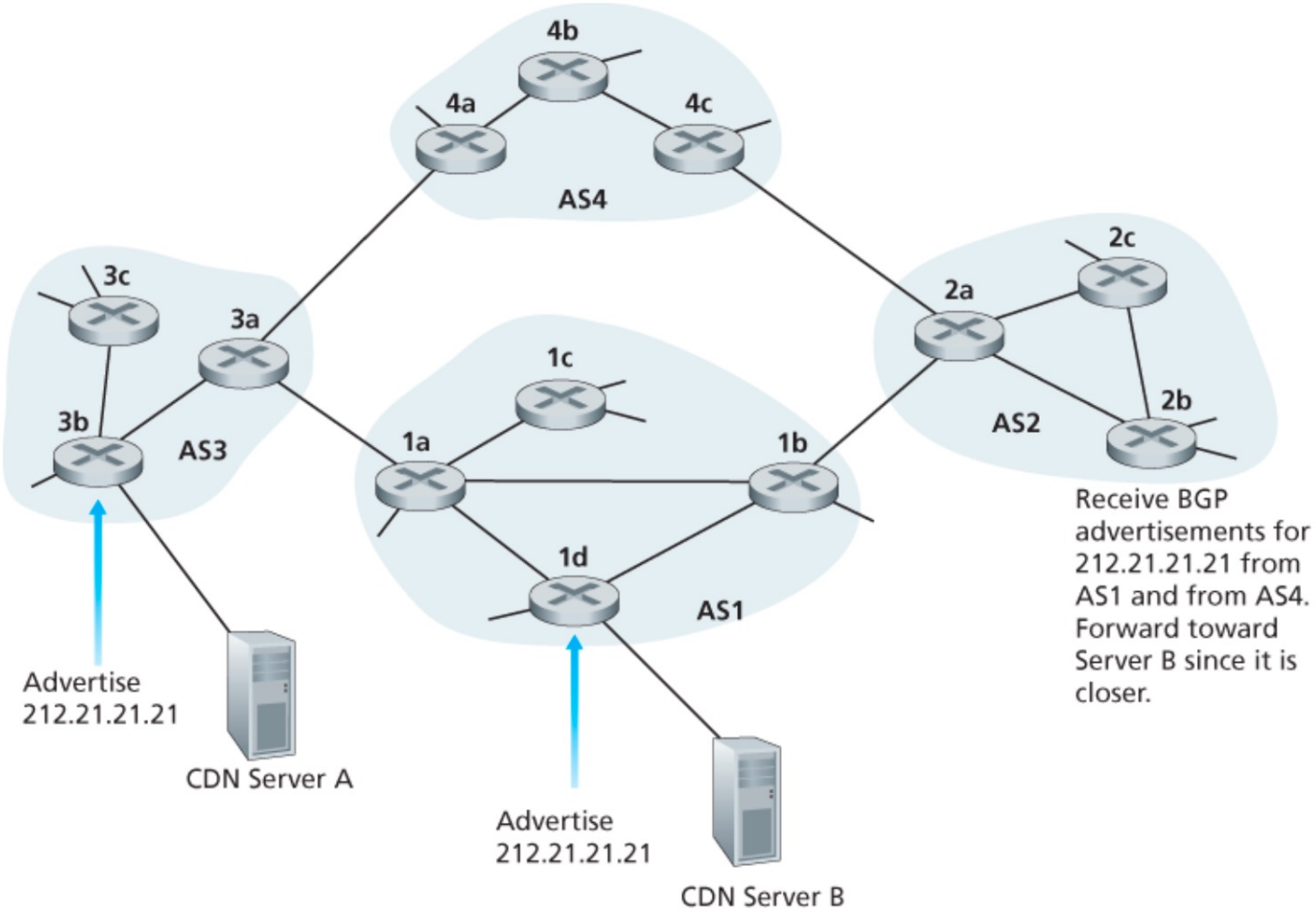
Unicast, Multicast, Anycast, Broadcast



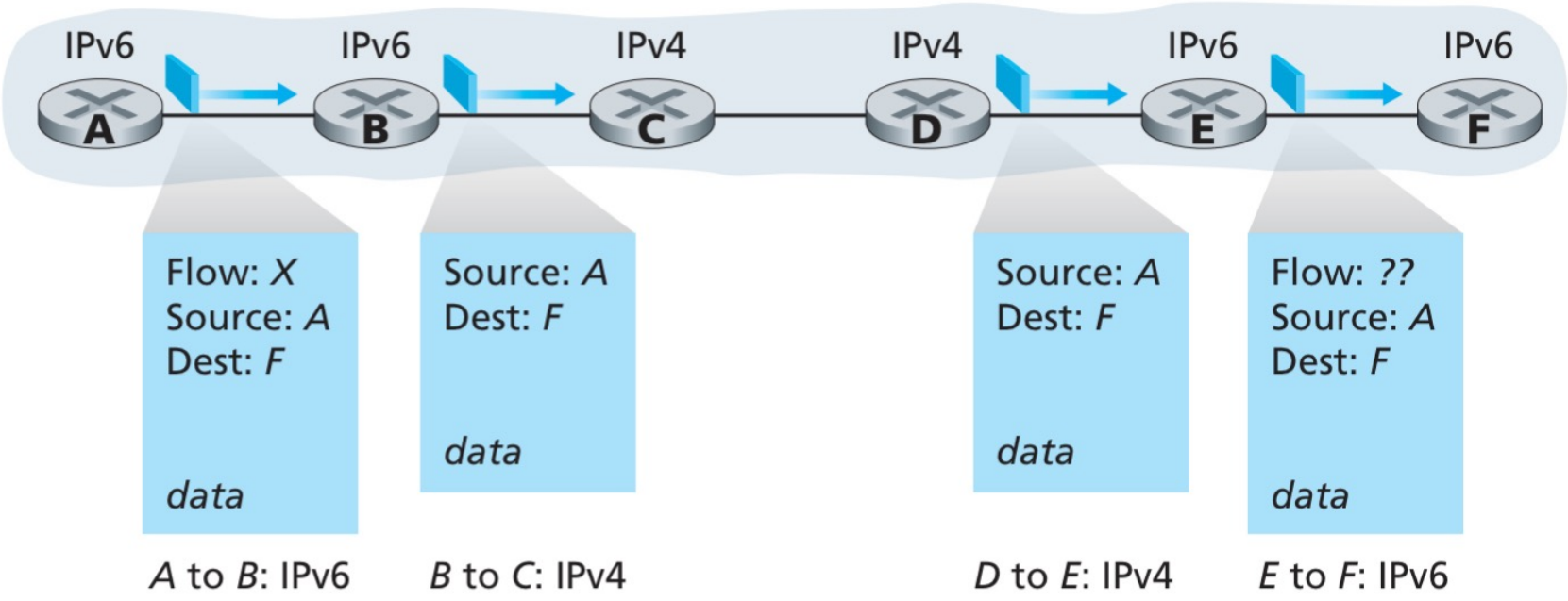
Unicast	Anycast	Multicast	Broadcast
IPv4		IPv4	IPv4
IPv6	IPv6	IPv6	

Unicast, Multicast, Anycast, Broadcast

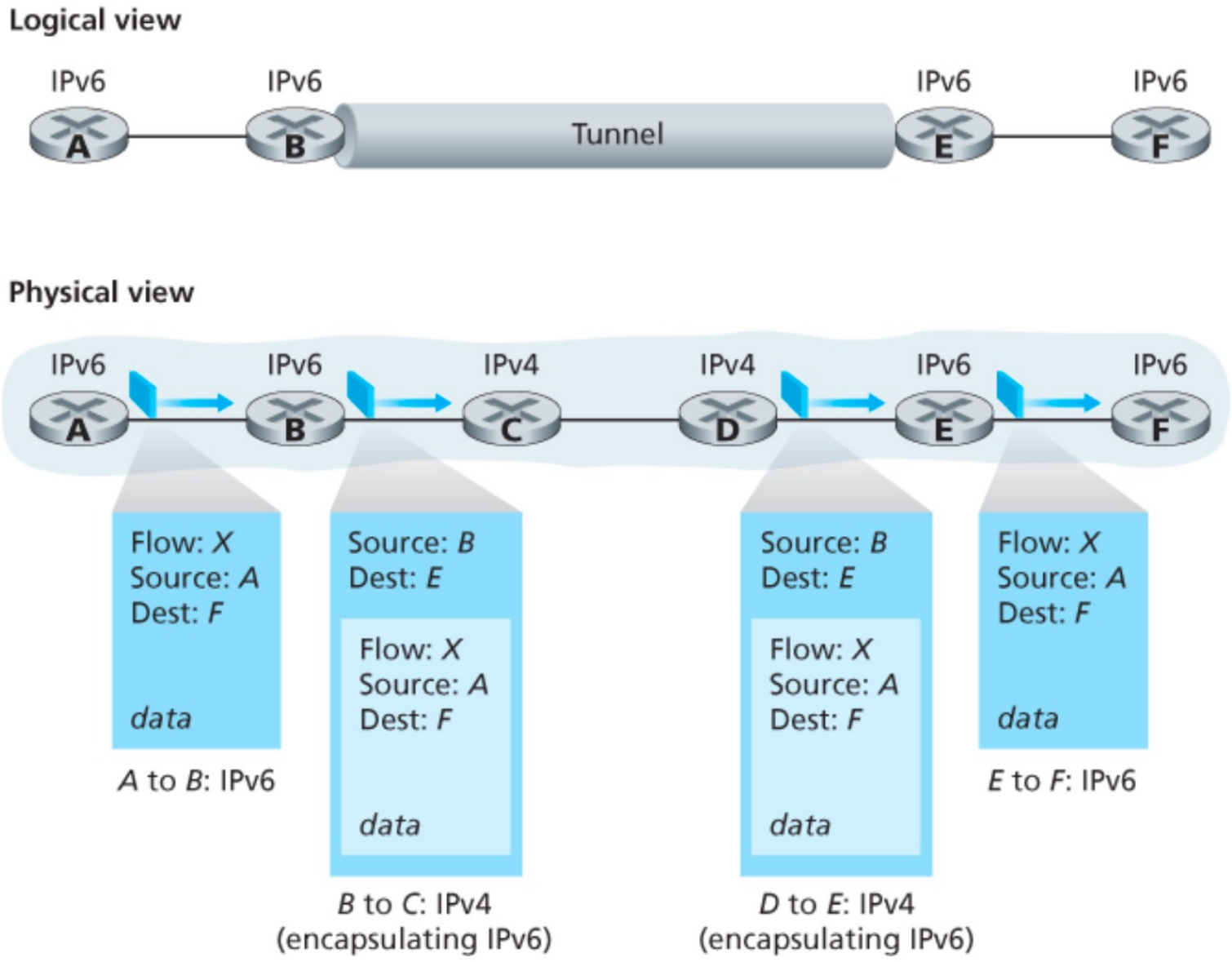
IP: Anycast example



IPv4 and IPv6 Co-existence



IPv4 and IPv6 Co-existence






Computer Networks

CMSC 417 : Spring 2023

COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

Topic:
Transport Layer Protocols (UDP, TCP)
(Textbook chapter 5)

Nirupam Roy

Tu-Th 2:00-3:15pm
CSI 2117

March 5th, 2024



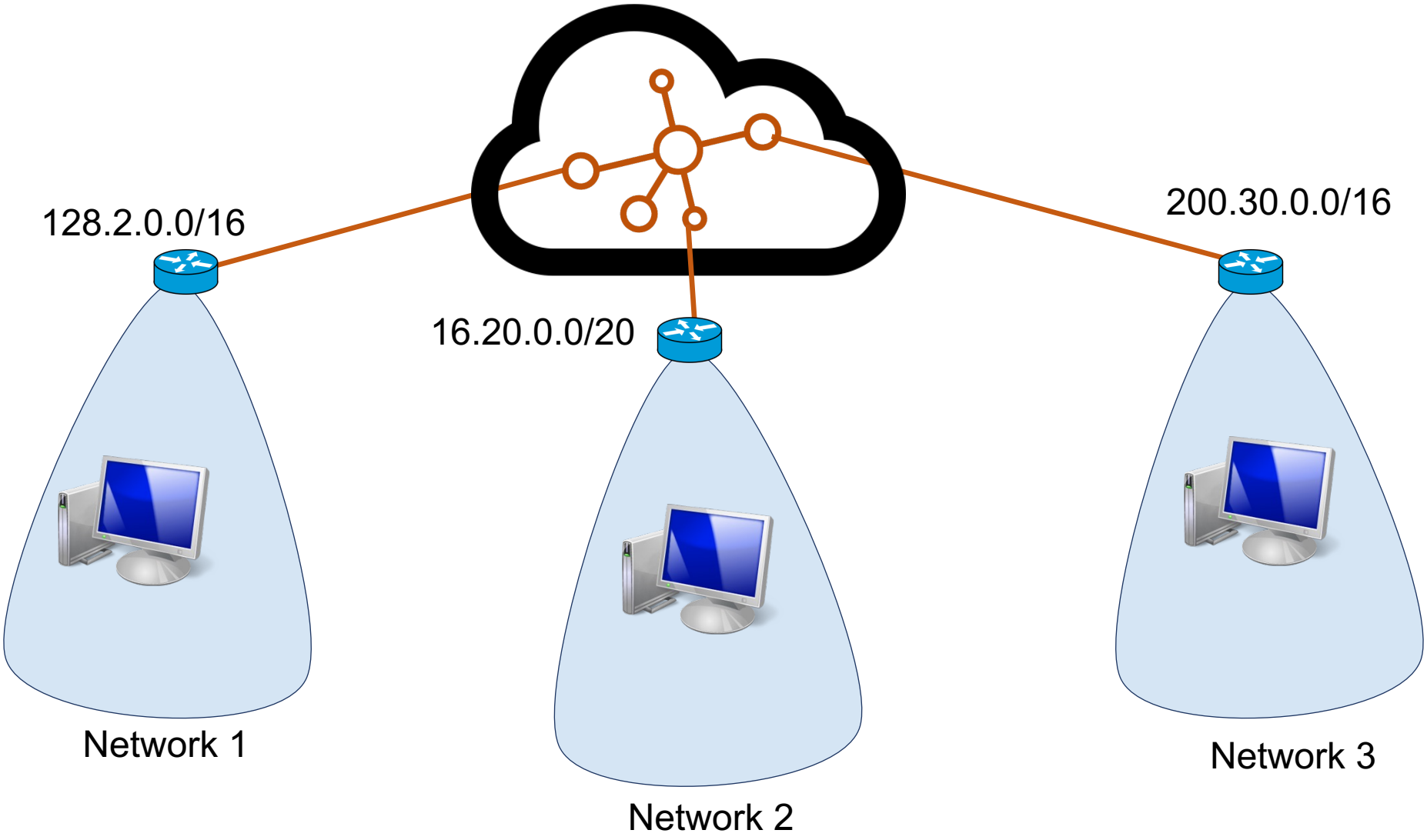

Transport layer: From the lens of two perspectives

(1) Abstraction of clients. (machines/networks/processes)

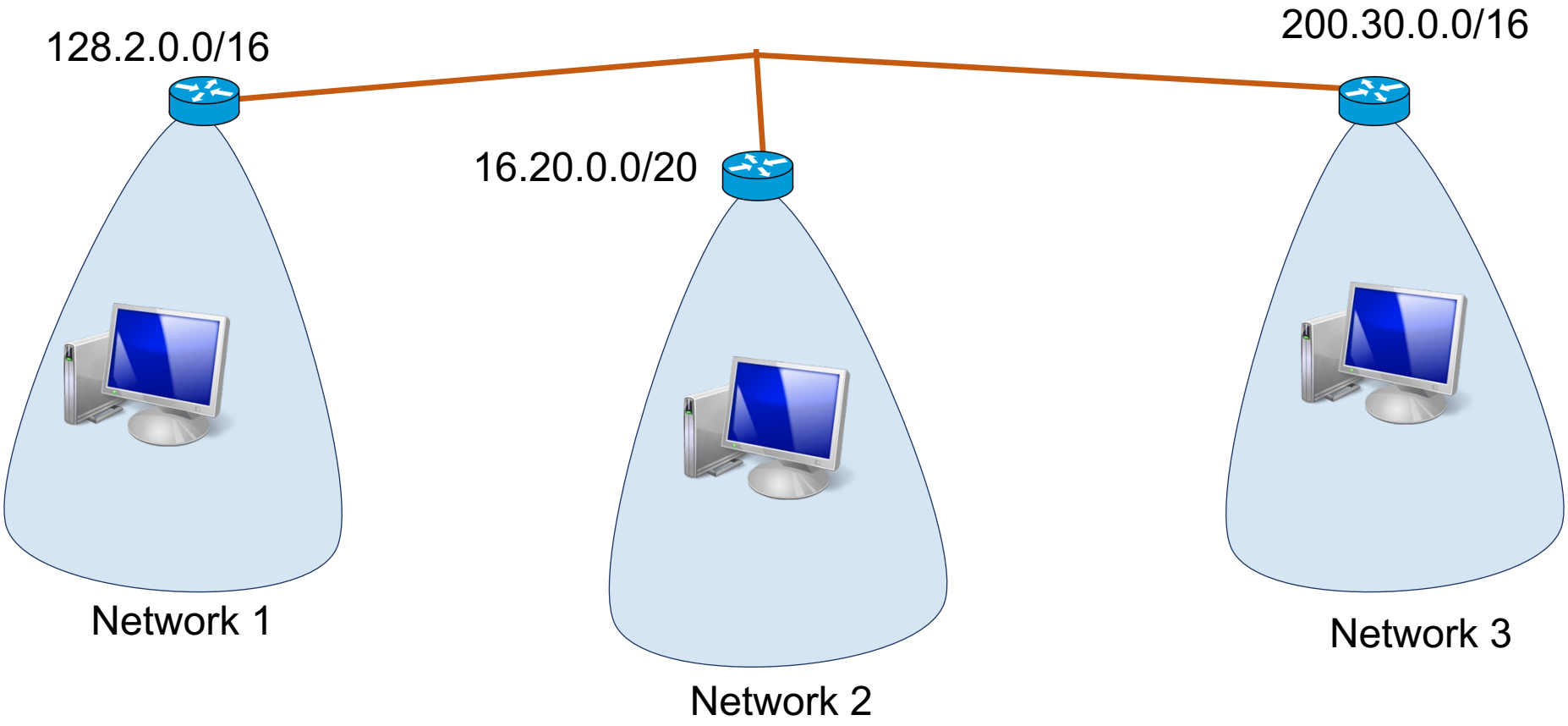
(2) Abstraction of services.

(1) Abstraction of clients

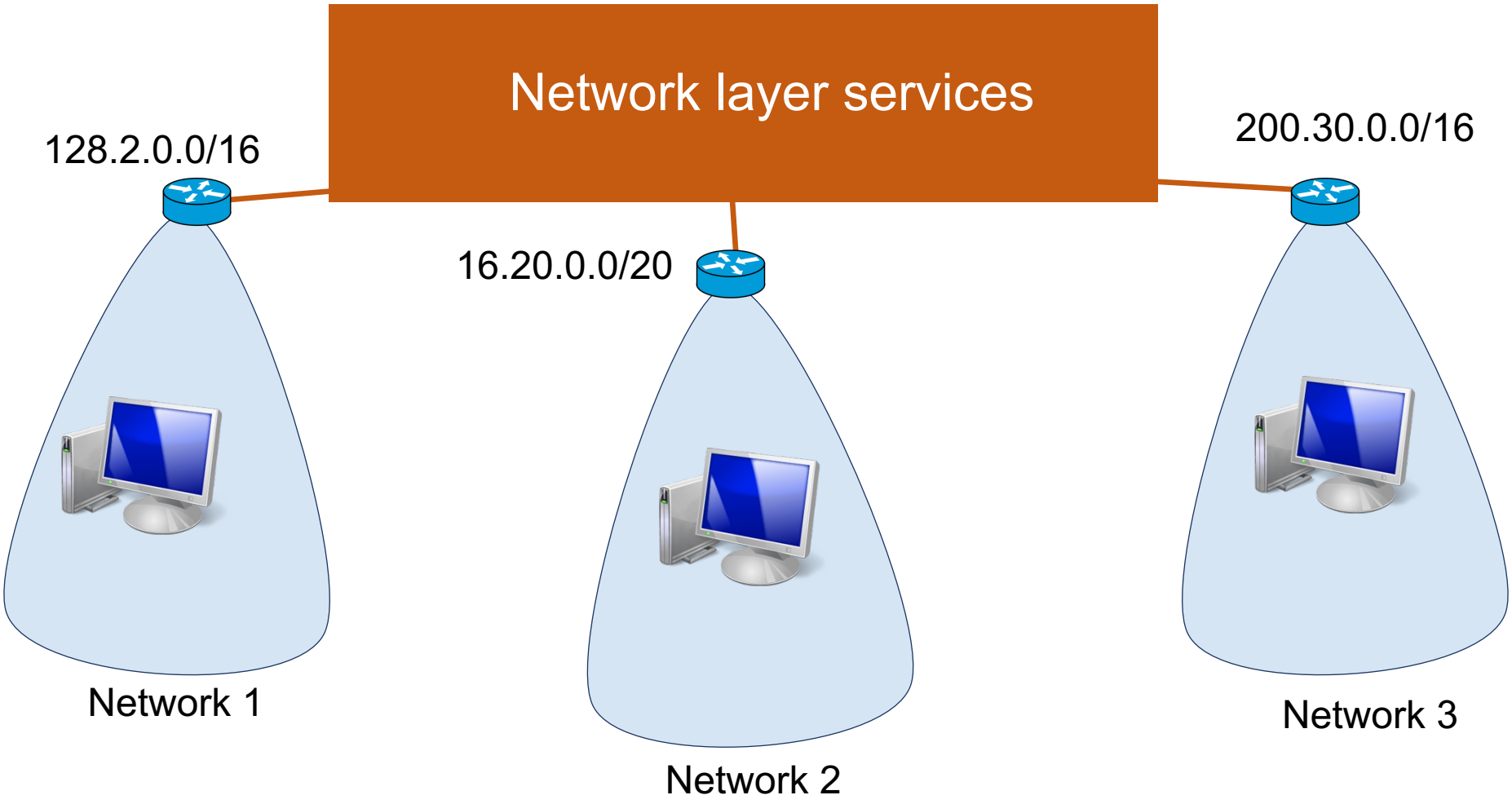
Network layer abstraction



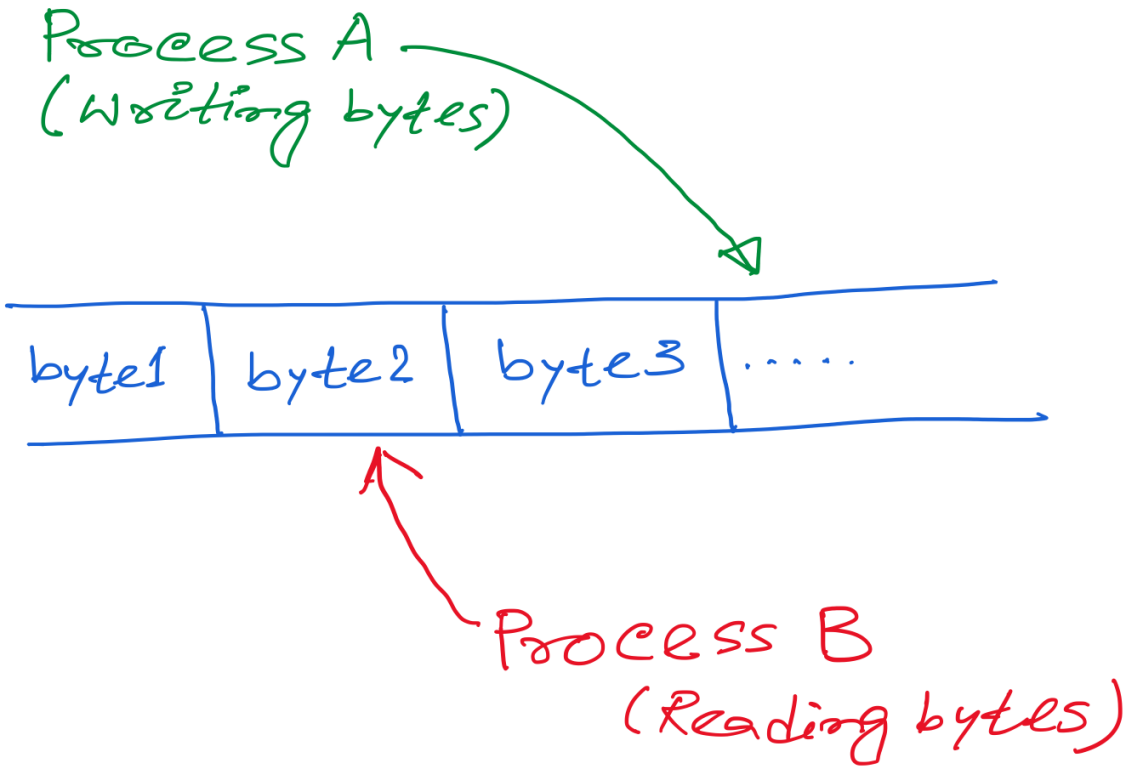
Network layer abstraction



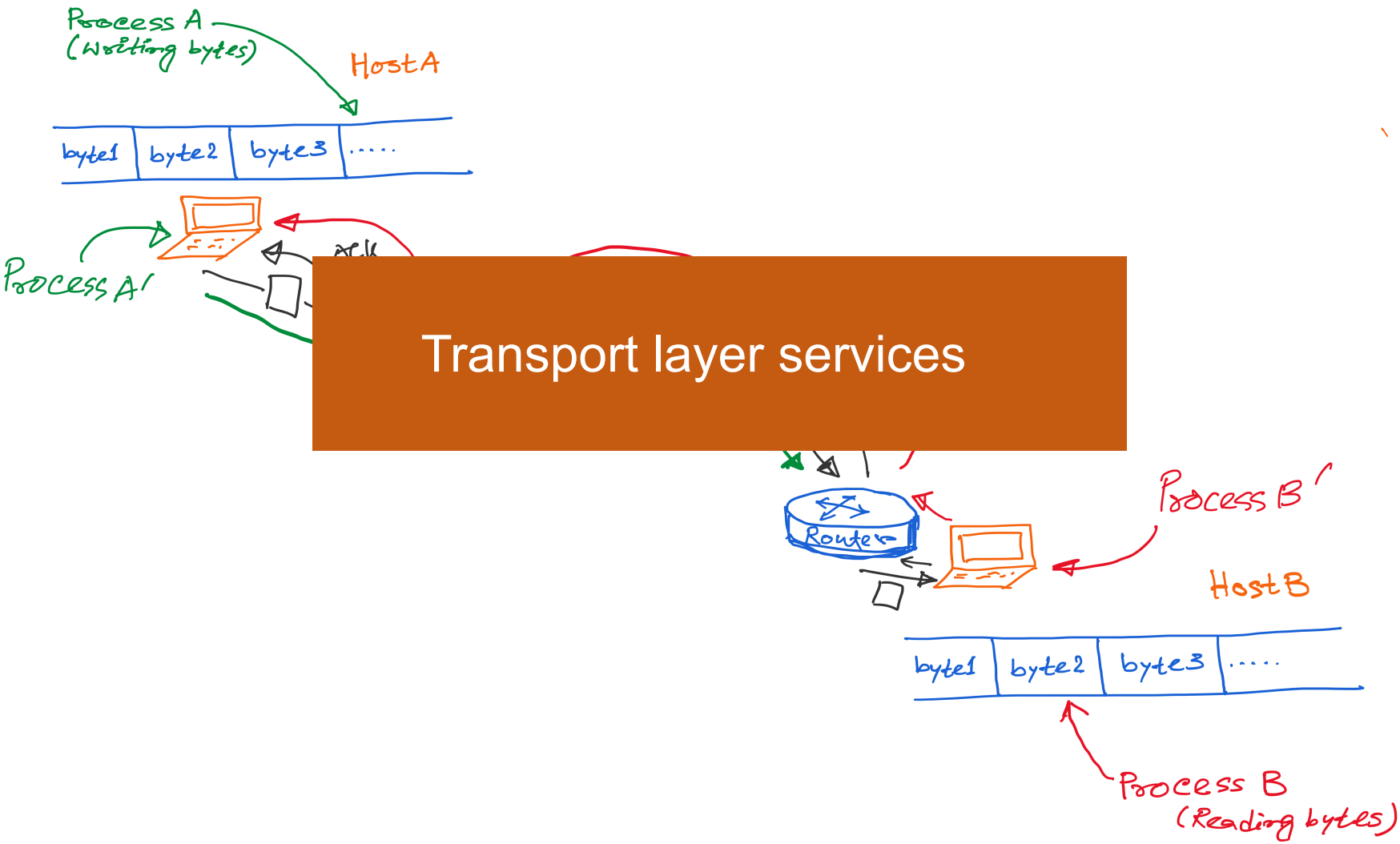
Network layer abstraction

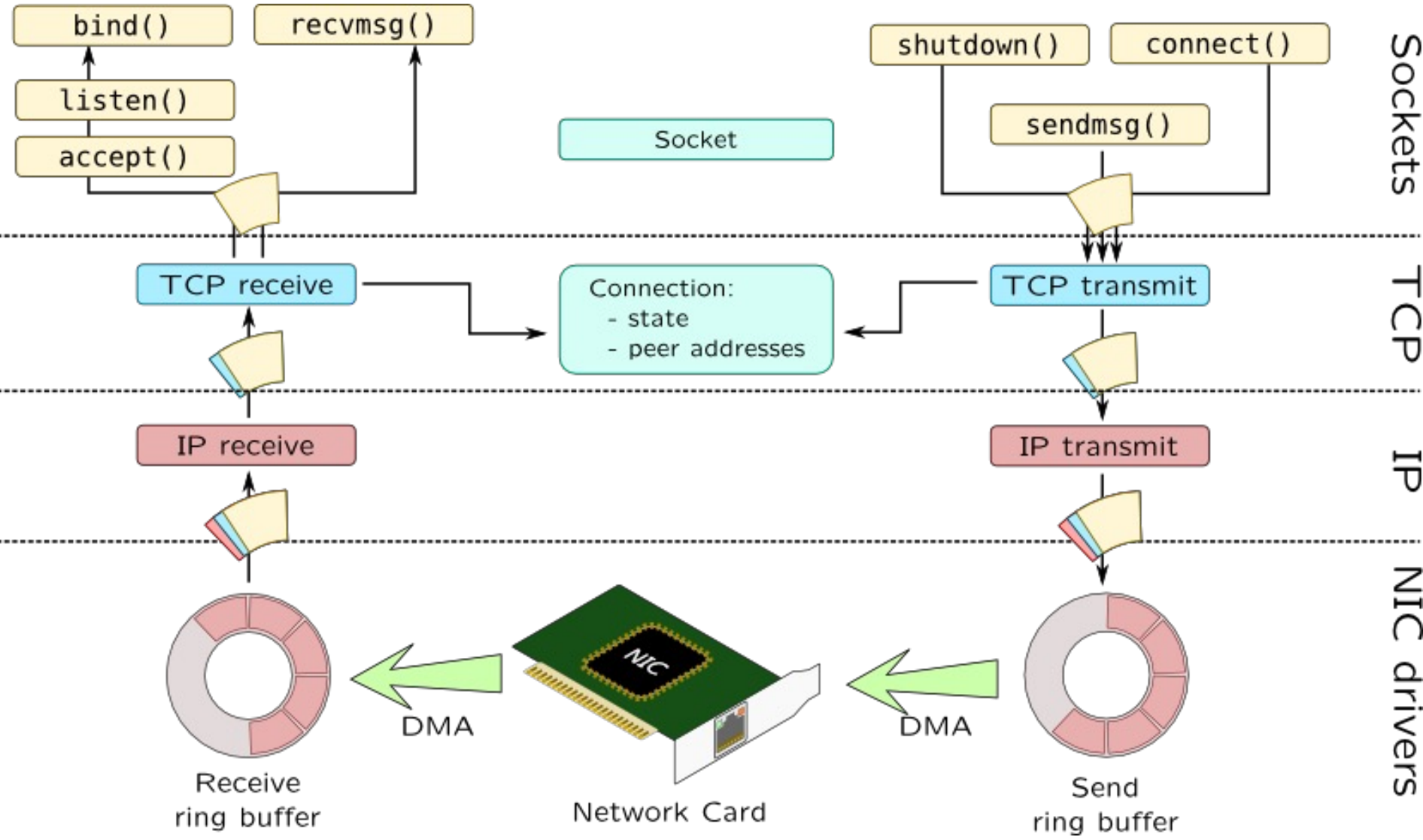


Transport layer abstraction

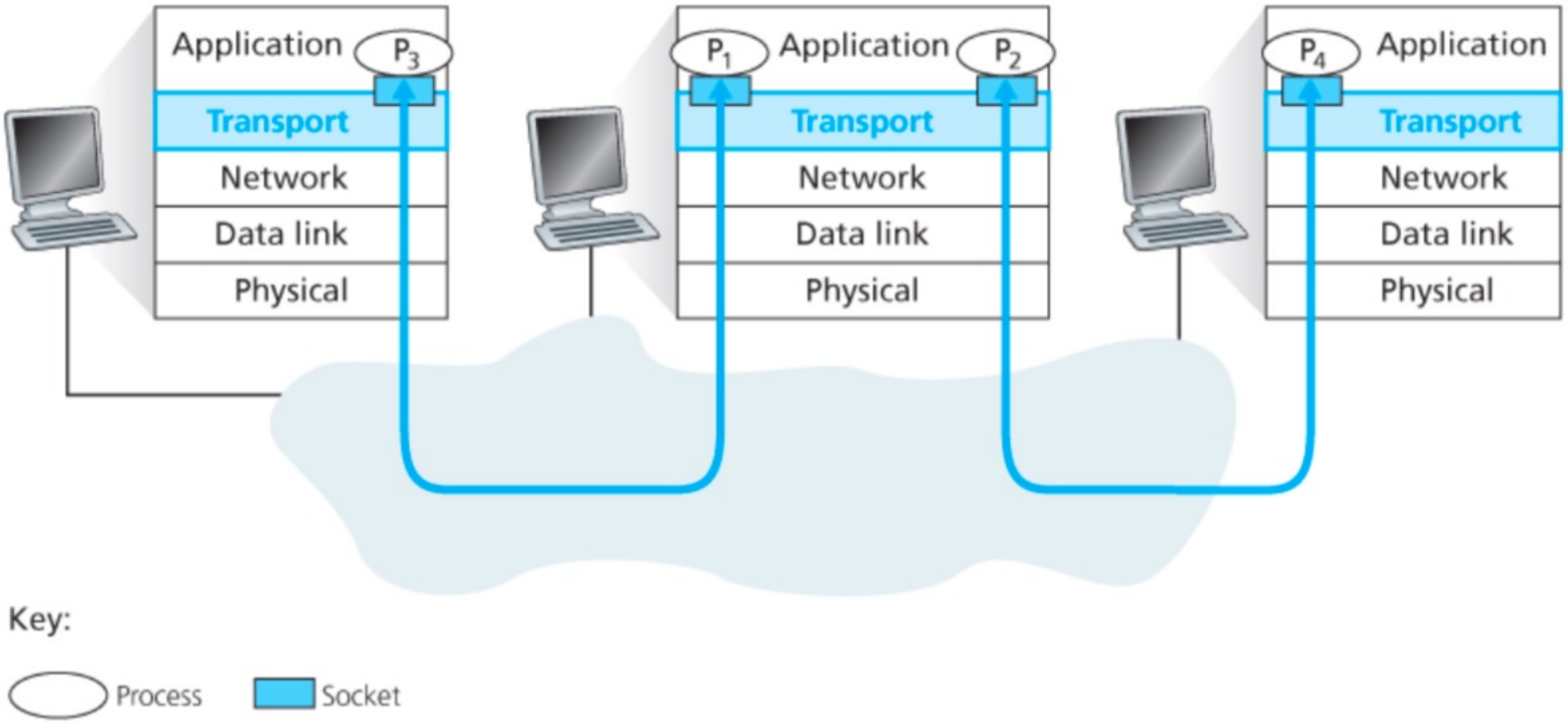


Transport layer abstraction




bind()
recvmsg()
listen()
accept()
Socket
shutdown()
connect()
sendmsg()
Sockets
TCP receive
Connection:
- state
- peer addresses
TCP transmit
TCP
IP receive
IP transmit
IP
NIC drivers
DMA
NIC
DMA
Receive
ring buffer
Network Card
Send
ring buffer

Connection between processes

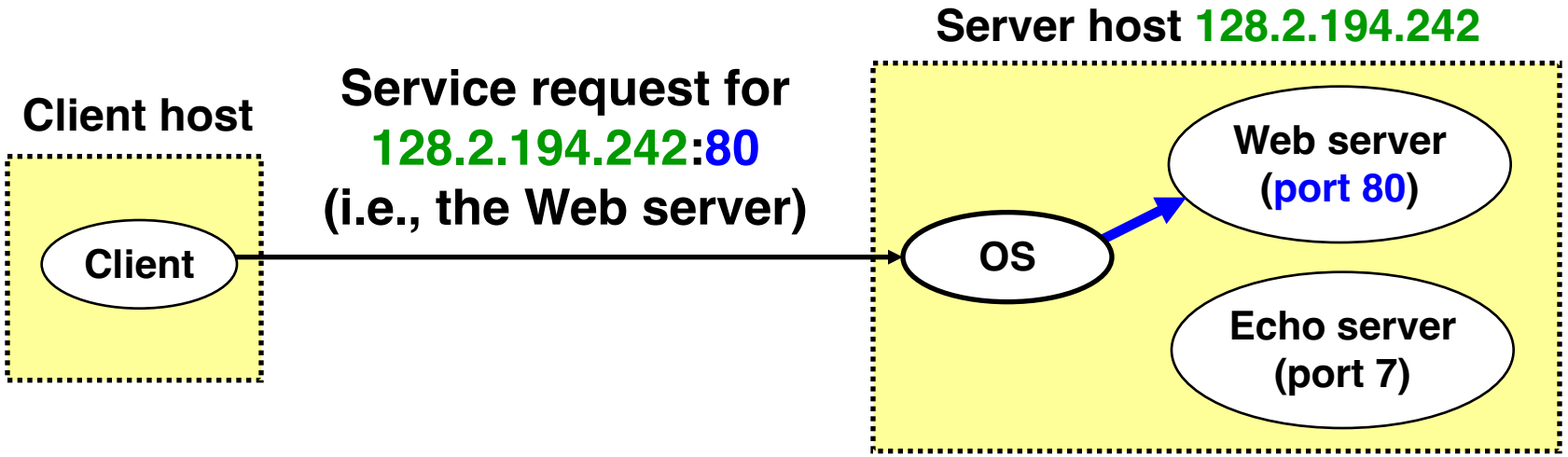


Transport Protocols

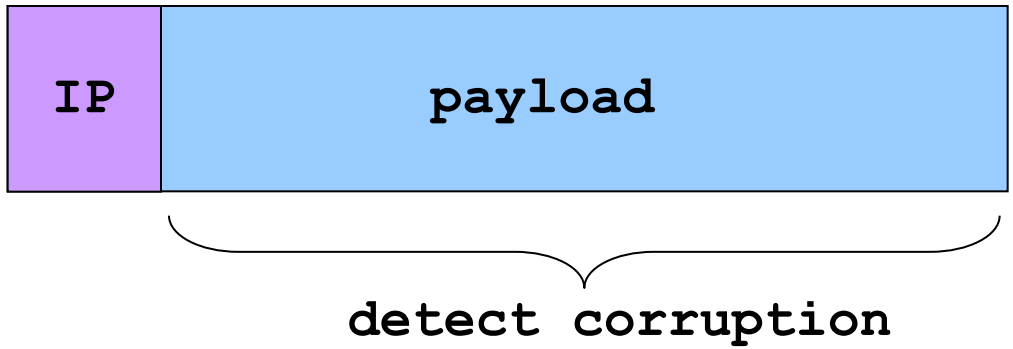
- Logical communication between processes
  - Sender divides a message into segments
  - Receiver reassembles segments into message
- Transport services
  - (De)multiplexing packets
  - Detecting corrupted data
  - Optionally: reliable delivery, flow control, …

Two Basic Transport Features

- **Demultiplexing:** port numbers



- **Error detection:** checksums



Multiplexing/demultiplexing

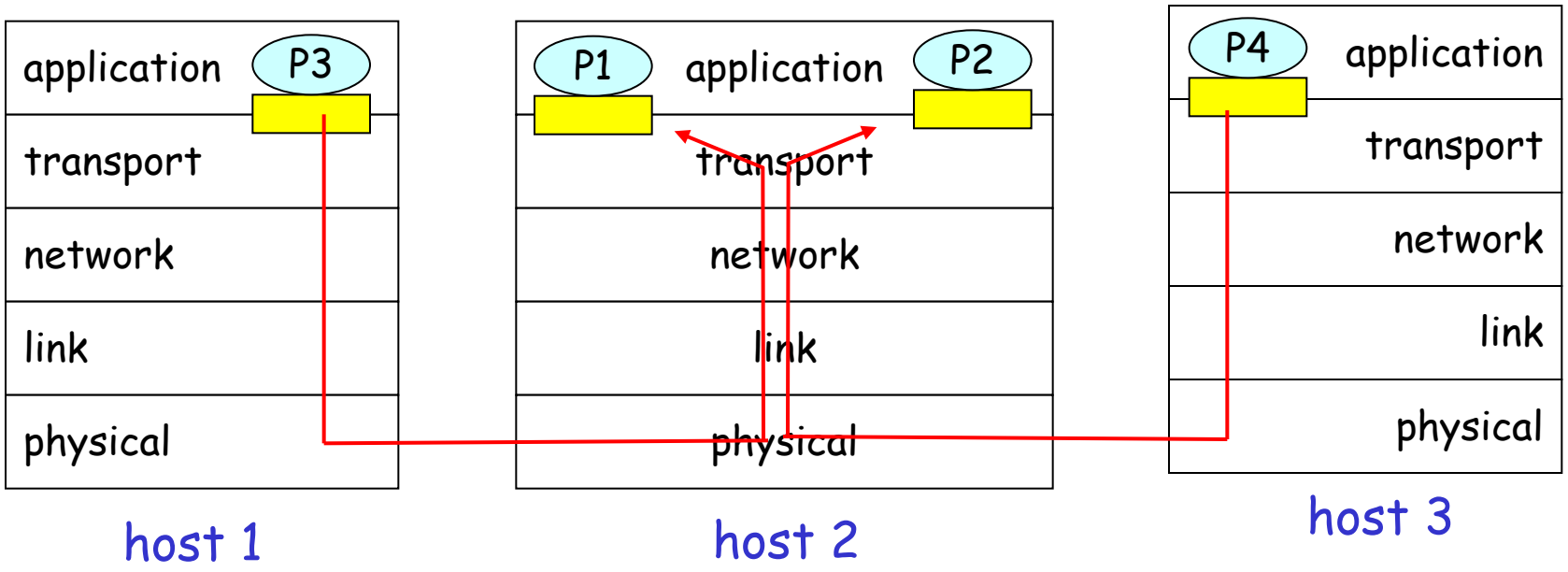
Demultiplexing at rcv host:

delivering received segments to correct socket

Multiplexing at send host:

Data headers helps flows from multiple sockets to use common the network channel, Info in header later helps Demultiplexing.

= socket = process

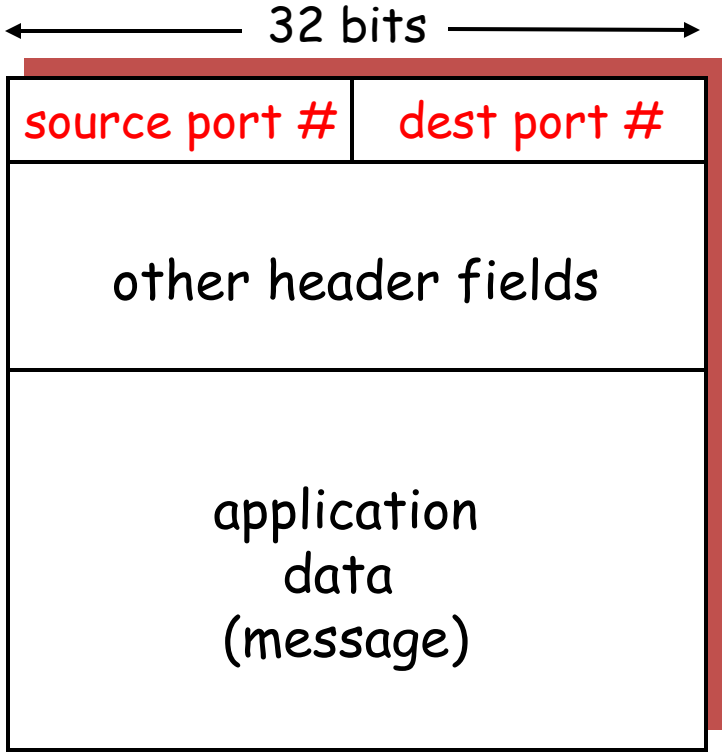


How demultiplexing works

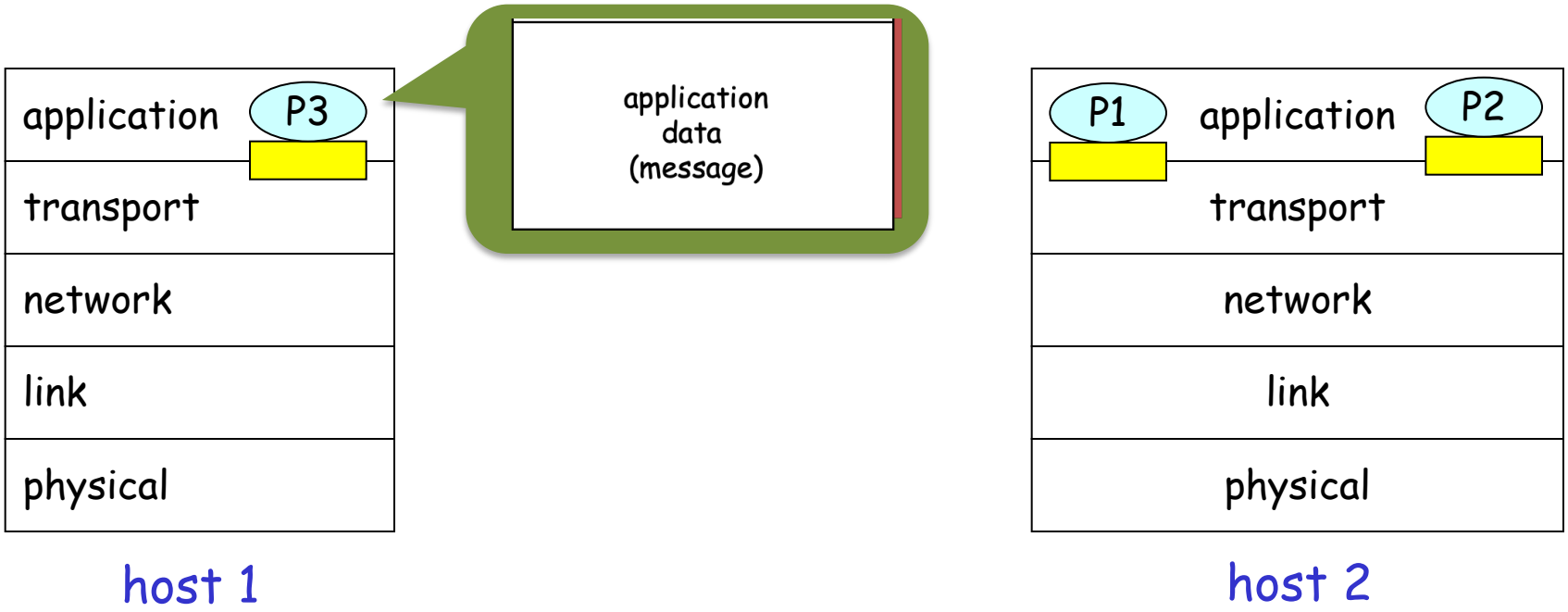
1) host receives IP datagrams

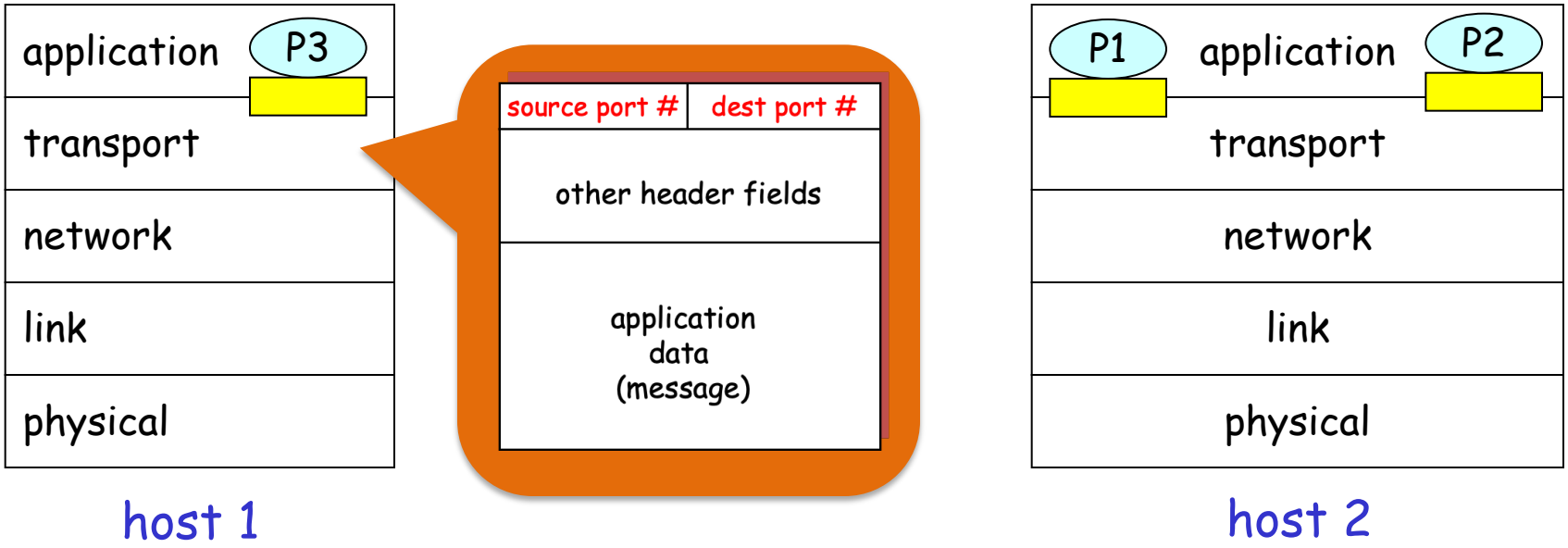
- each datagram has source IP address, destination IP address
- each datagram carries 1 transport-layer segment
- each segment has source, destination port number

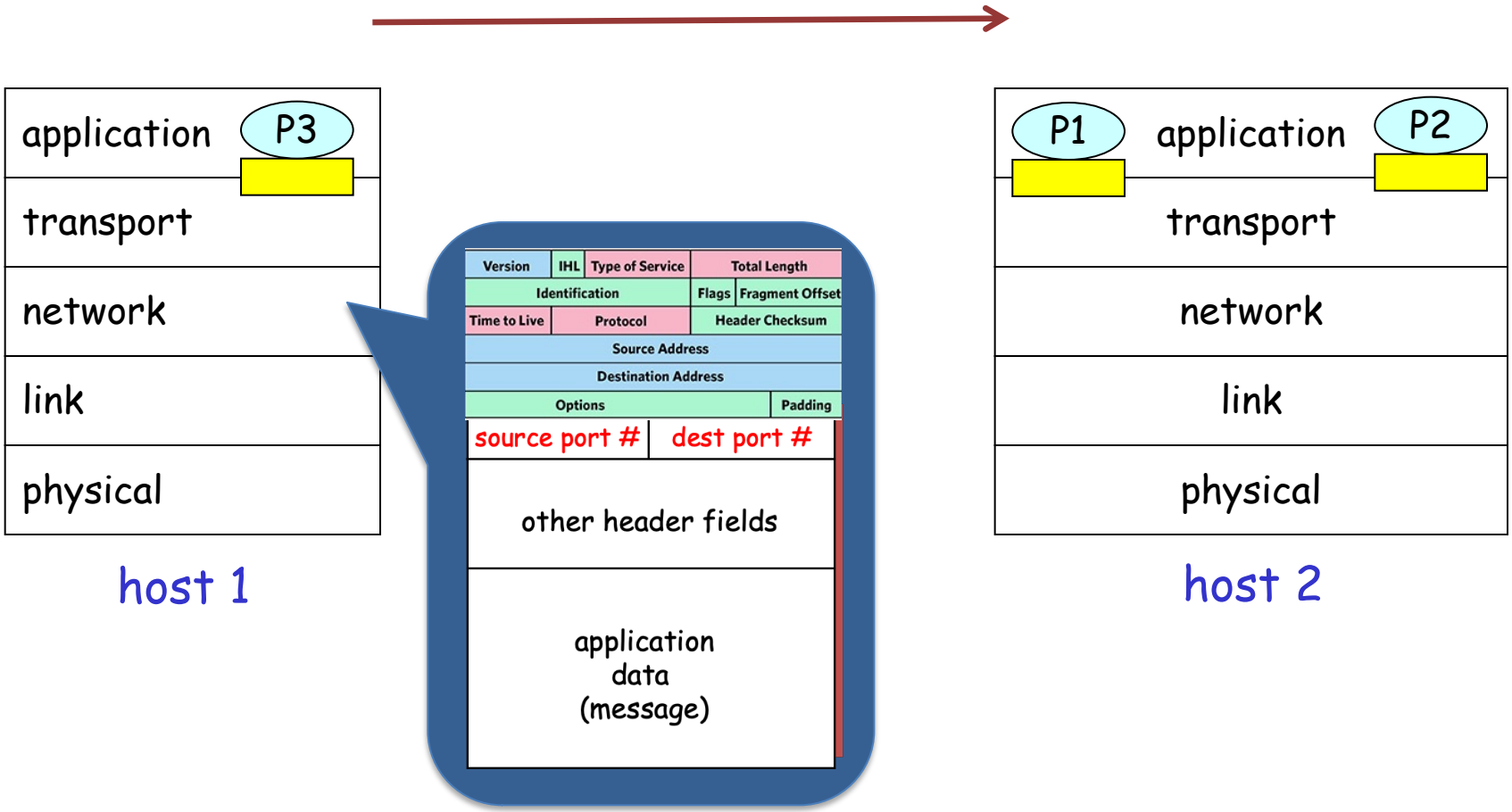
2) host uses IP addresses & port numbers to direct segment to appropriate socket

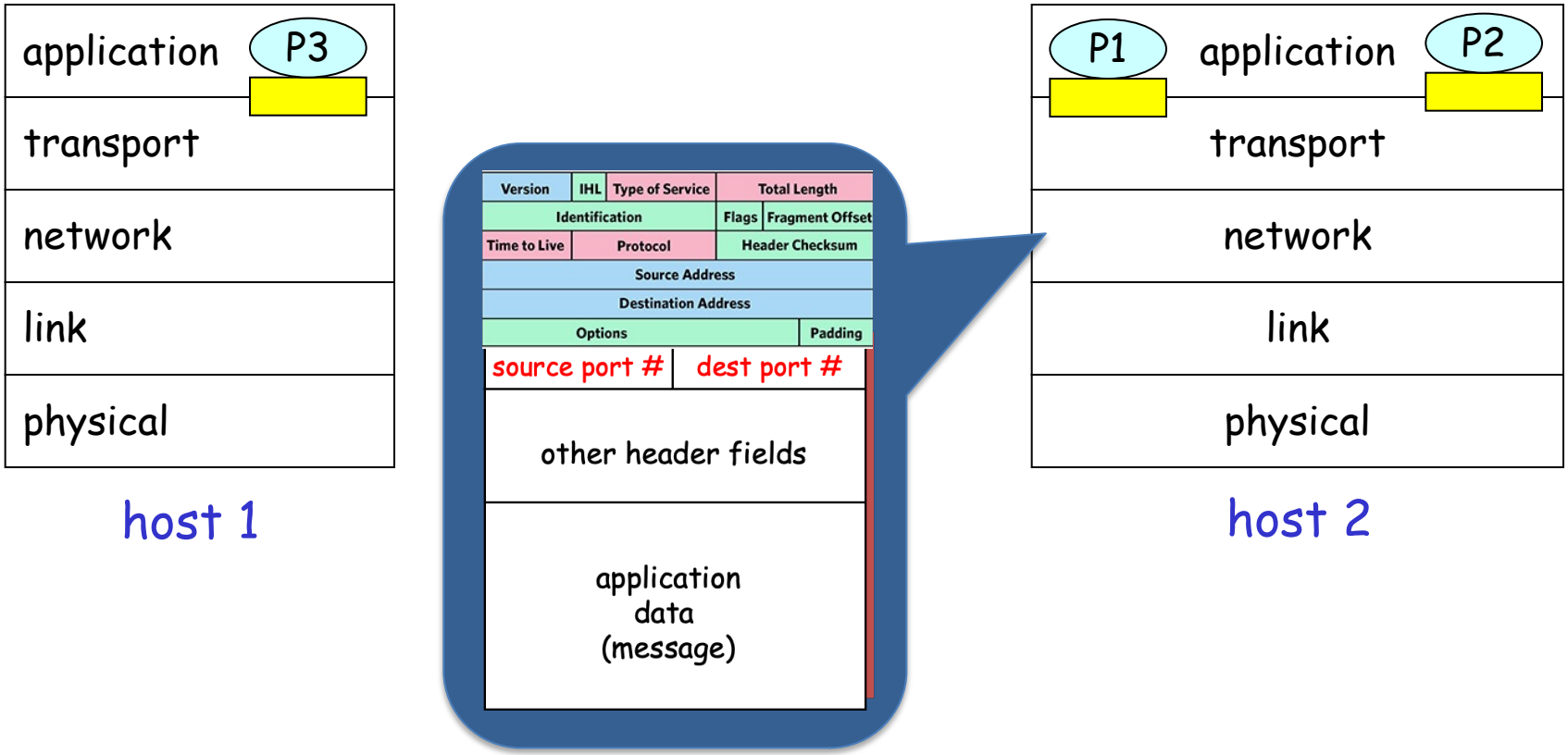


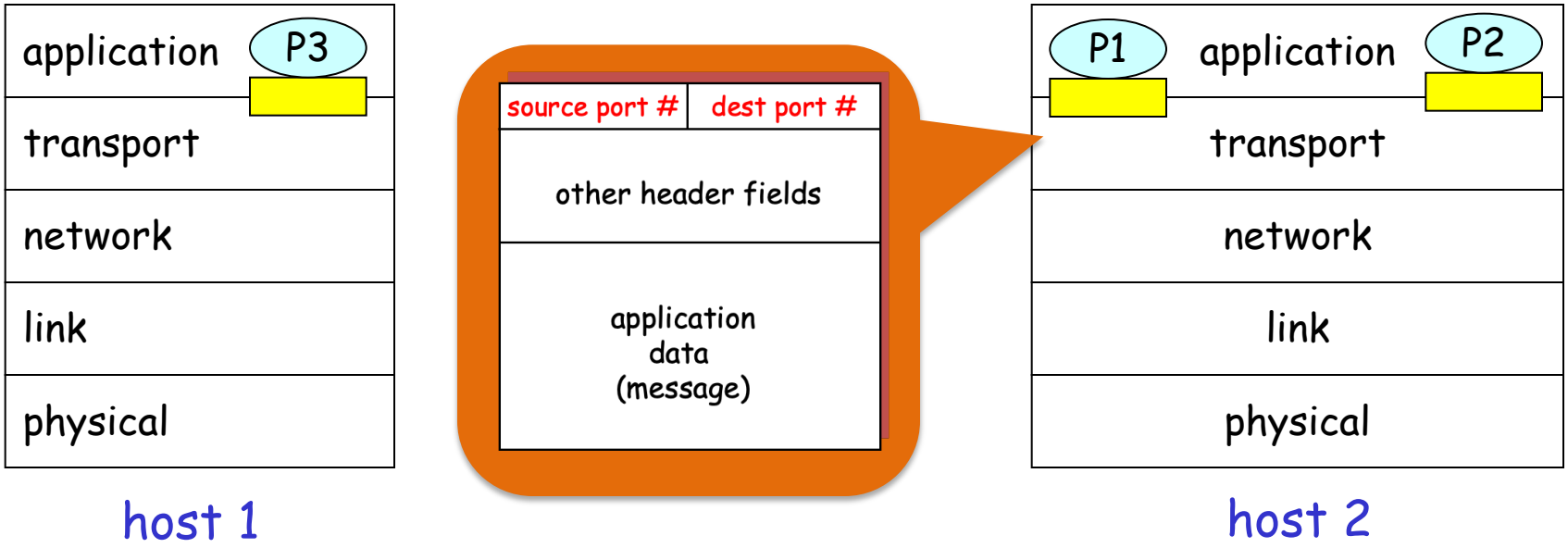
TCP/UDP segment format


application
P3
transport
network
link
physical
application
data
(message)
P1
application
P2
transport
network
link
physical
host 1
host 2


application
P3
transport
network
link
physical
source port #
dest port #
other header fields
application
data
(message)
P1
application
P2
transport
network
link
physical
host 1
host 2


application
P3
transport
network
link
physical
host 1
Version
IHL
Type of Service
Total Length
Identification
Flags
Fragment Offset
Time to Live
Protocol
Header Checksum
Source Address
Destination Address
Options
Padding
source port #
dest port #
other header fields
application
data
(message)
P1
application
P2
transport
network
link
physical
host 2


application
P3
transport
network
link
physical
host 1
Version
IHL
Type of Service
Total Length
Identification
Flags
Fragment Offset
Time to Live
Protocol
Header Checksum
Source Address
Destination Address
Options
Padding
source port #
dest port #
other header fields
application
data
(message)
P1
application
P2
transport
network
link
physical
host 2


application
P3
transport
network
link
physical
host 1
source port #
dest port #
other header fields
application
data
(message)
P1
application
P2
transport
network
link
physical
host 2