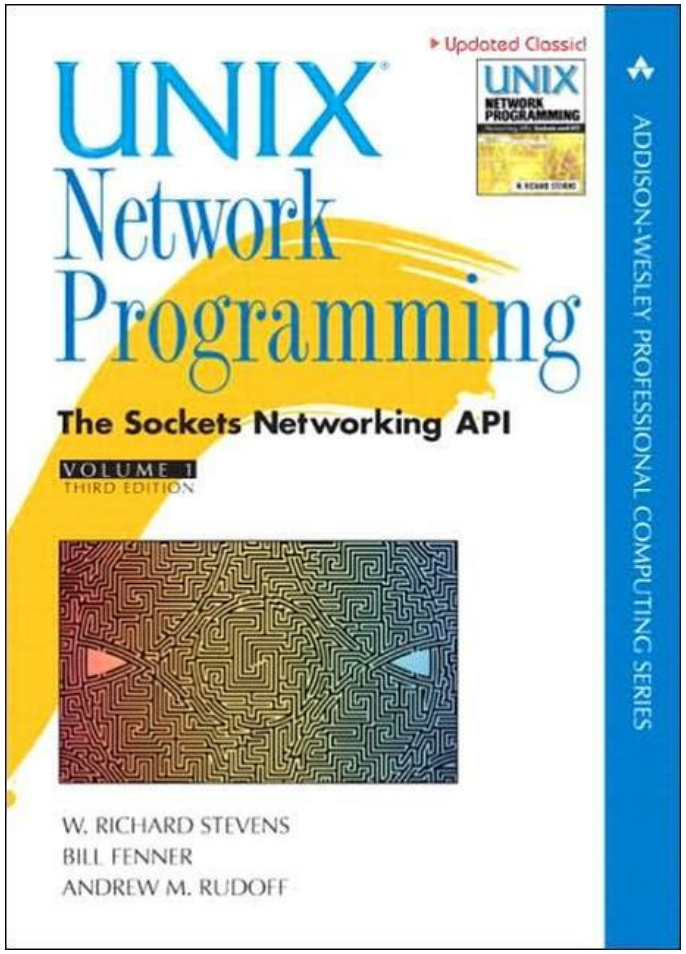
Unix Network Programming

② The Transport Layer: TCP, UDP, and SCTP



Chapter2 : The Transport Layer: TCP, UDP, and SCTP

__2.1 Introduction
__2.2 The Big Picture
__2.3 User Datagram Protocol (UDP)
__2.4 Transmission Control Protocol (TCP)
__2.5 Stream Control Transmission Protocol (SCTP)
__2.6 TCP Connection Establishment and Termination
__2.7 TIME WAIT State
__2.8 SCTP Association Establishment and Termination
__2.9 Port Numbers
__2.10 TCP Port Numbers and Concurrent Servers
__2.11 Buffer Sizes and Limitations
__2.12 Standard Internet Services
__2.13 Protocol Usage by Common Internet Applications
__2.14 Summary

2.1 Introduction

- This Chapter focuses on the transport layer : TCP, UDP, SCTP(StreamControlTransmission)

- Charistic of Protocol

  * UDP : simple, unreliable datagram, provides * message boundaries (explained next slide)

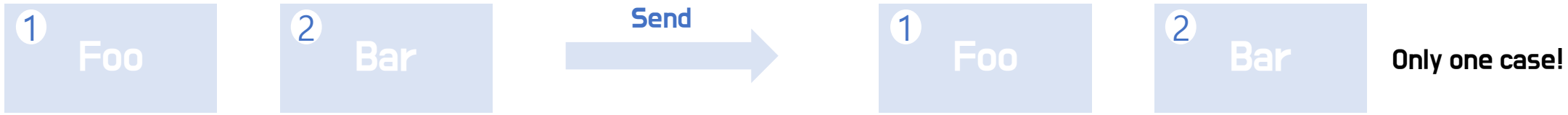
  * TCP : sophisticated, reliable byte-stream protocol

  * SCTP : similar to TCP but, it also provides * message boundaries,
    transport-level support for multihoming, and a way to minimize head-of-line blocking.

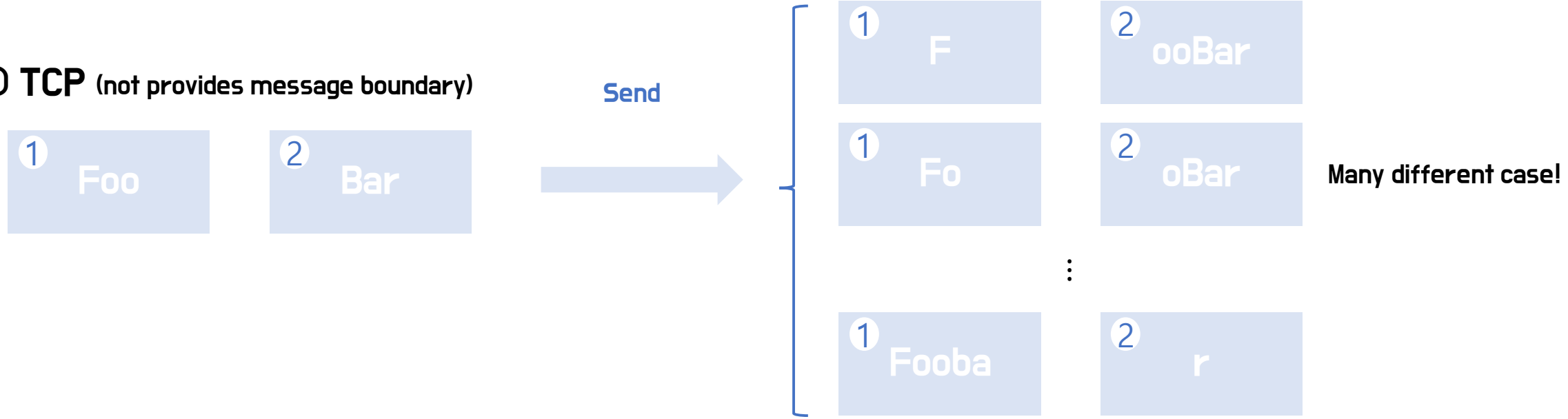
2.1 Introduction

* Message boundary

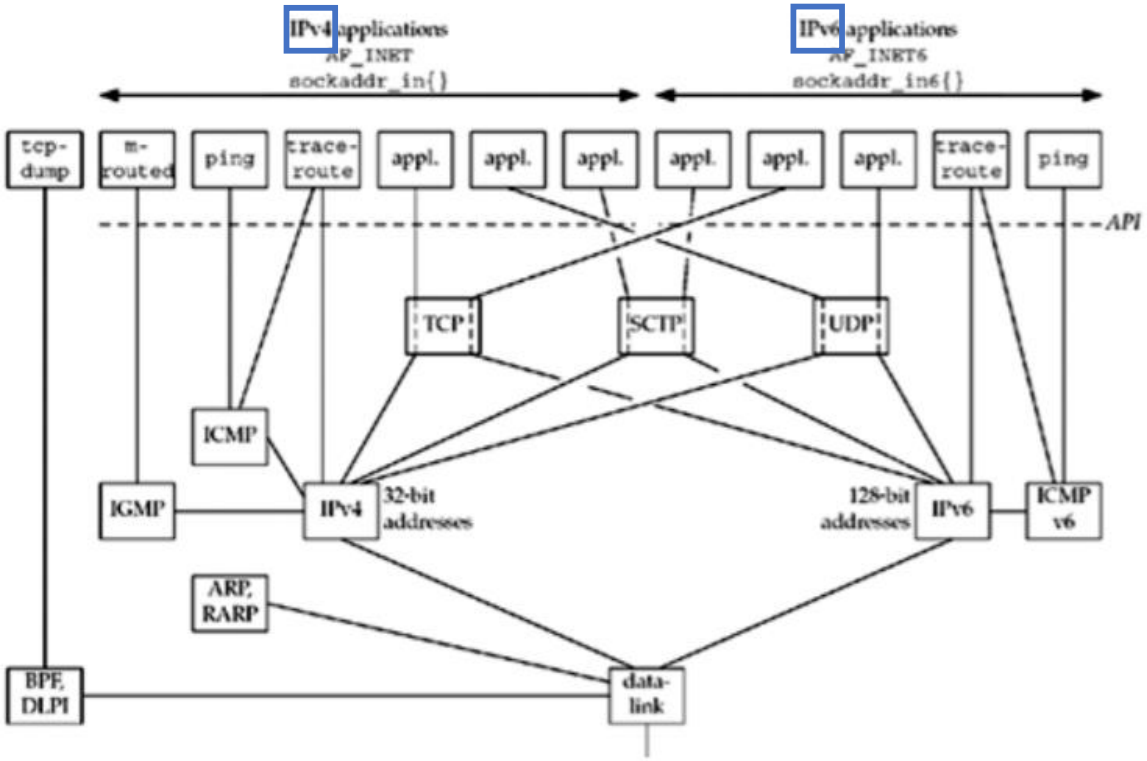
① UDP (provides message boundary)



② TCP (not provides message boundary)



2.2 The Big Picture



Overview of TCP/IP Protocol

* IPv4 vs IPv6

	IPv4	IPv6
Adress Family	AF_INET	AF_INET6
structure	sockaddr_in	sockaddr_in6
Address	32-bit	128-bit

Internet Protocol

- **IPv4** : Internet Protocol version 4. 32-bit address. IPv4 provides UCP. UDP. SCTP. ICMP. IGMP.
- **IPv6** : Internet Protocol version 6. 128-bit address. IPv6 provides TCP. UDP. SCTP. ICMPv6.
- **TCP** : Transmisiion Control Protocol. Connection-oriented protocol. Full-duplex byte stream.
  TCP can use IPv4 or IPv6.
- **UDP** : User Datagram Protocol. Connectionliss protocol.
- **SCTP** : Stream Control Transmission Protocol. Connection-oriented protocol.

=> Each Internet Protocol is defined by one or more documents called a Request for Comments (RFC)

"IPv4/IPv6 host" and "dual-stack host" is used to denote hosts that support both IPv4 and IPv6.

2.3 User Datagram Protocol (UDP)

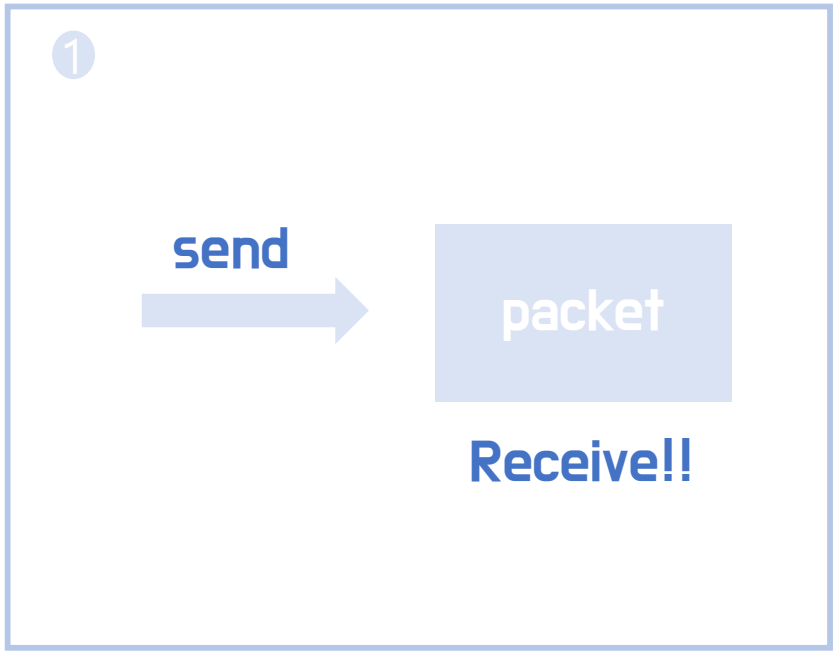
- UDP is a simple transport-layer protocol
- UDP is no guarantee that a UDP datagram will ever reach its final destination

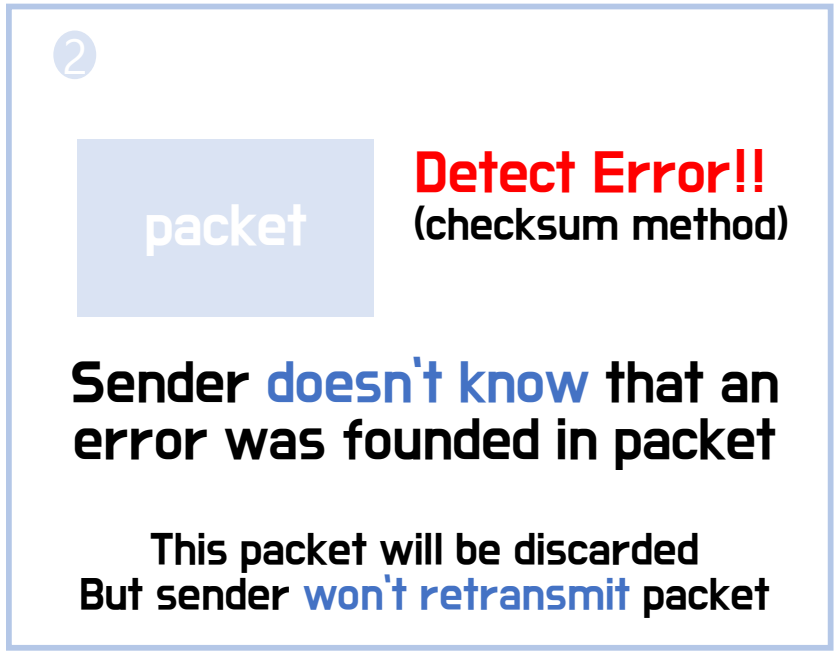
  => * Problem1 (lack of realibility)

2.3 User Datagram Protocol (UDP)

- UDP is a simple transport-layer protocol
- UDP is no guarantee that a UDP datagram will ever reach its final destination

  => * Problem2 (lack of realibility)

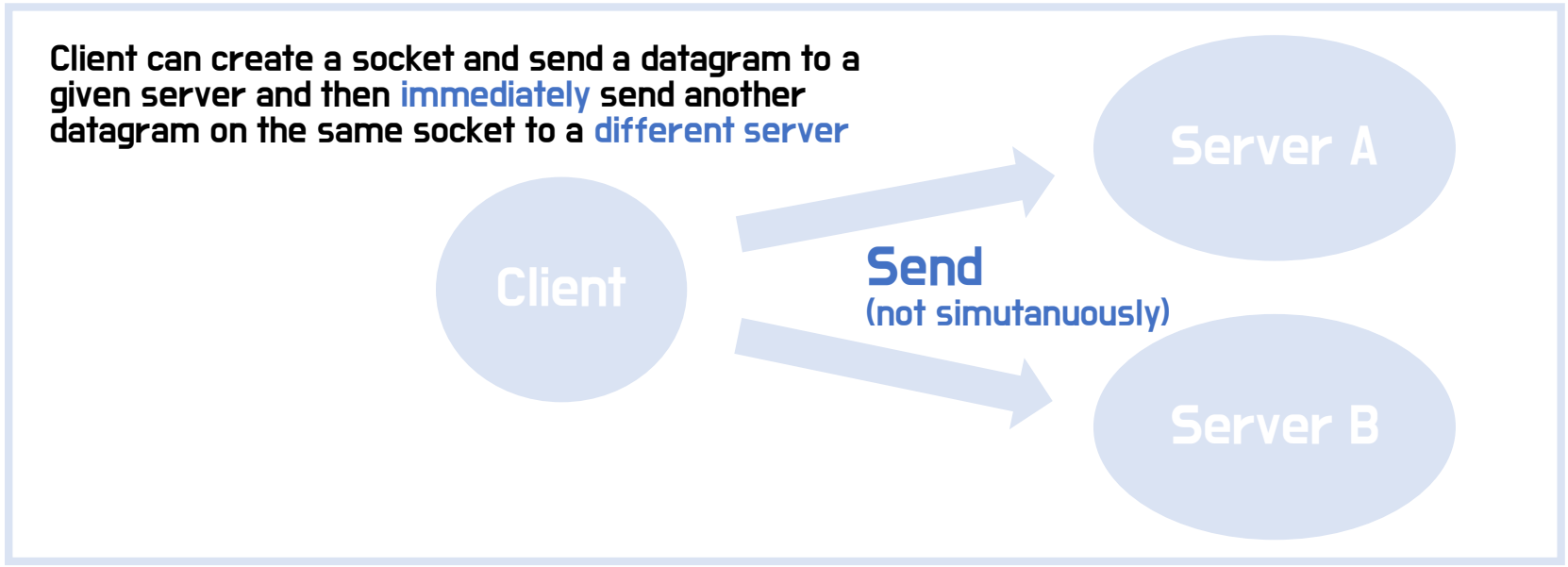




2.3 User Datagram Protocol (UDP)

- UDP is a simple transport-layer protocol
- UDP is no guarantee that a UDP datagram will ever reach its final destination
- UDP provides a connectionless service

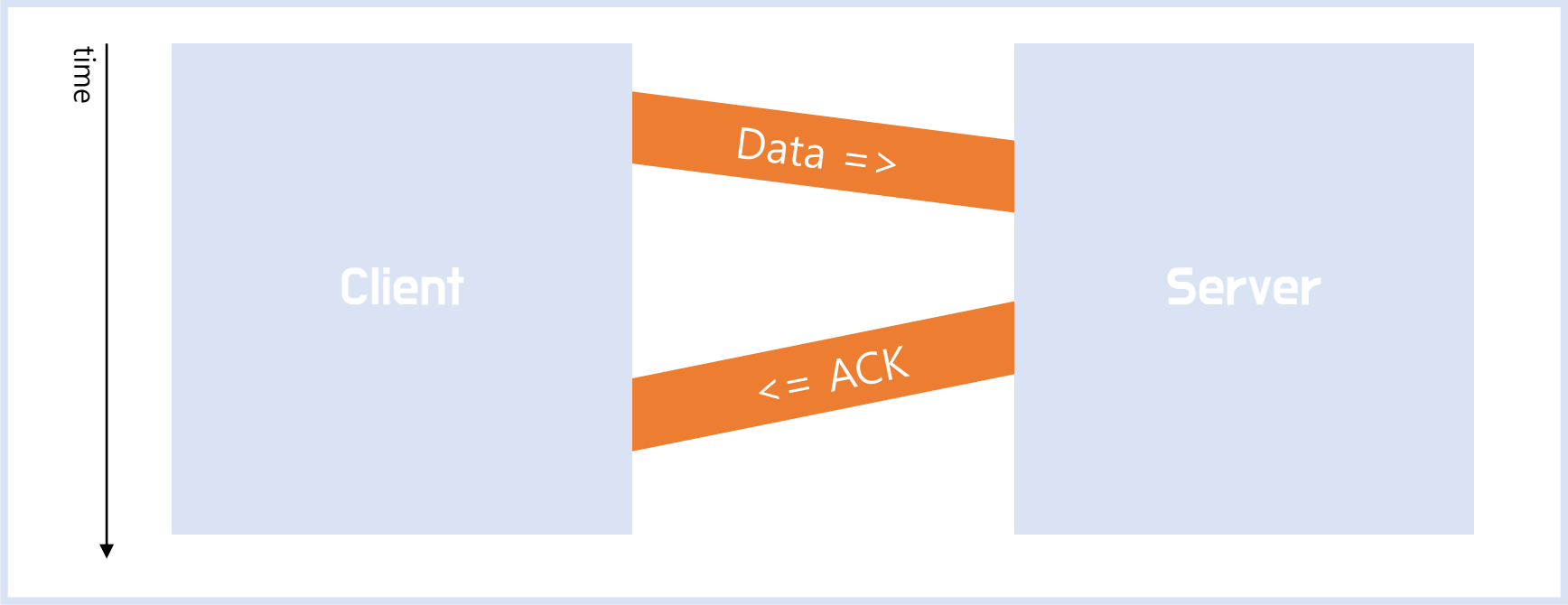
  => there need not be any long-term relationship between client and server



2.4 Transmission Control Protocol (TCP)

- TCP provides connections between client and servers
- TCP also provides reliability.

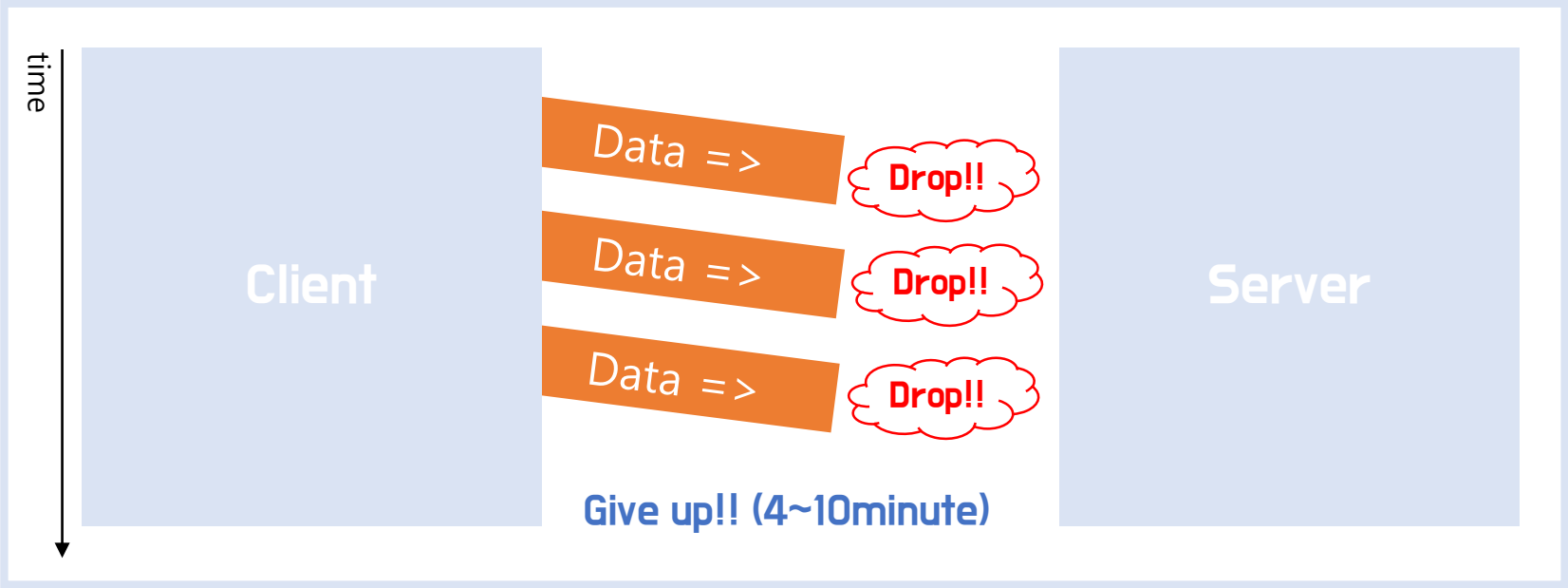
  => when TCP sedns data to the other end. it requires an acknowledgement(ACK) in return.



2.4 Transmission Control Protocol (TCP)

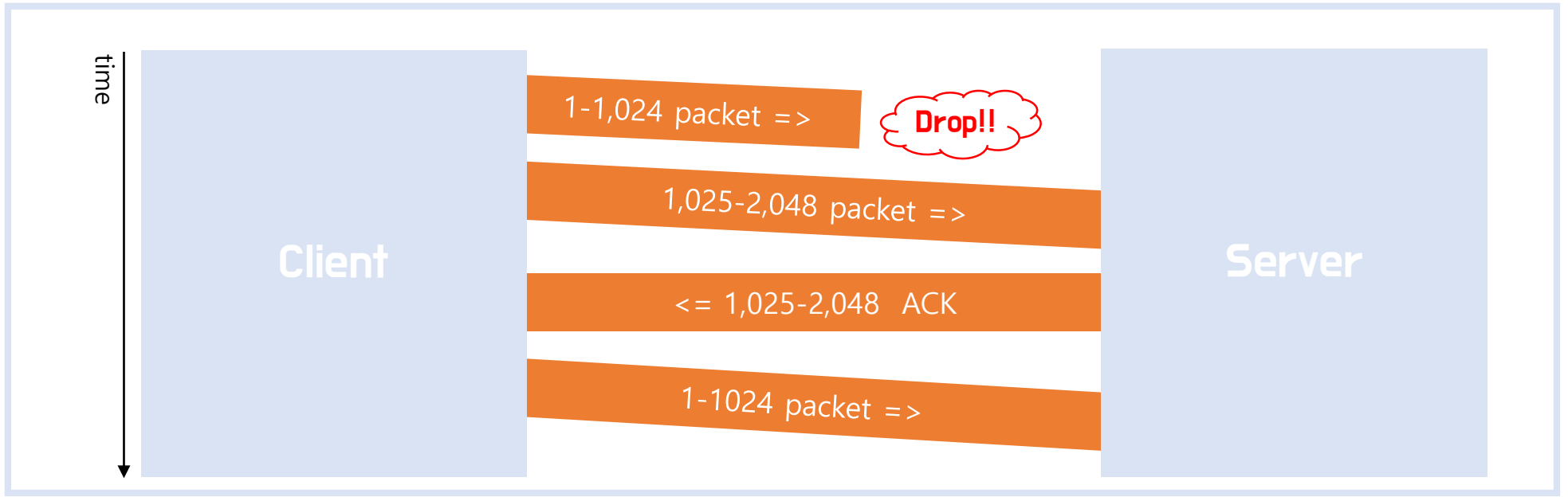
- TCP provides connections between client and servers.
- TCP also provides reliability.
- TCP doesn't guarantee that the data will be received by the other endpoint.

=> TCP delievers data to the other endpoint if possible.



2.4 Transmission Control Protocol (TCP)

- TCP also sequences the data by associating a sequence number with every byte.



2.4 Transmission Control Protocol (TCP)

- If TCP receives duplicate data from its peer, it can detect that the data has been duplicated, and discard the duplicate data.

2.4 Transmission Control Protocol (TCP)

- TCP provides flow control.

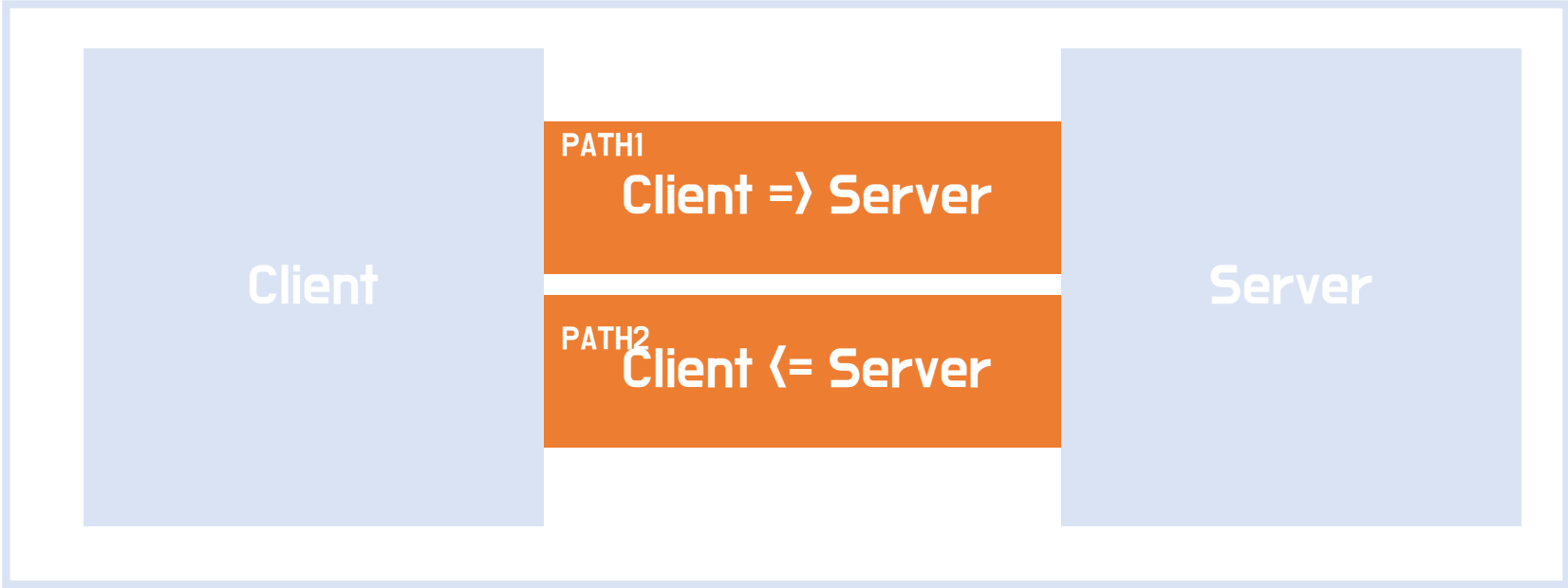
  => TCP always tell its peer exactly how many bytes of data it is willing to accept from the peer at any one time. ➔ this is called the window!!

* Window = Buffer_capability - Current_remain_in_buffer

2.4 Transmission Control Protocol (TCP)

- TCP connection is full-duplex.

  => application can send and receive data in both direction.



Summary of TCP

- TCP provides connections between client and servers.
- TCP also provides reliability.
- TCP doesn't guarantee that the data will be received by the other endpoint.
- TCP also sequences the data by associating a sequence number with every byte.
- If TCP receives duplicate data from its peer, it can detect that the data has been duplicated, and discard the duplicate data.
- TCP provides flow control.
- TCP connection is full-duplex.

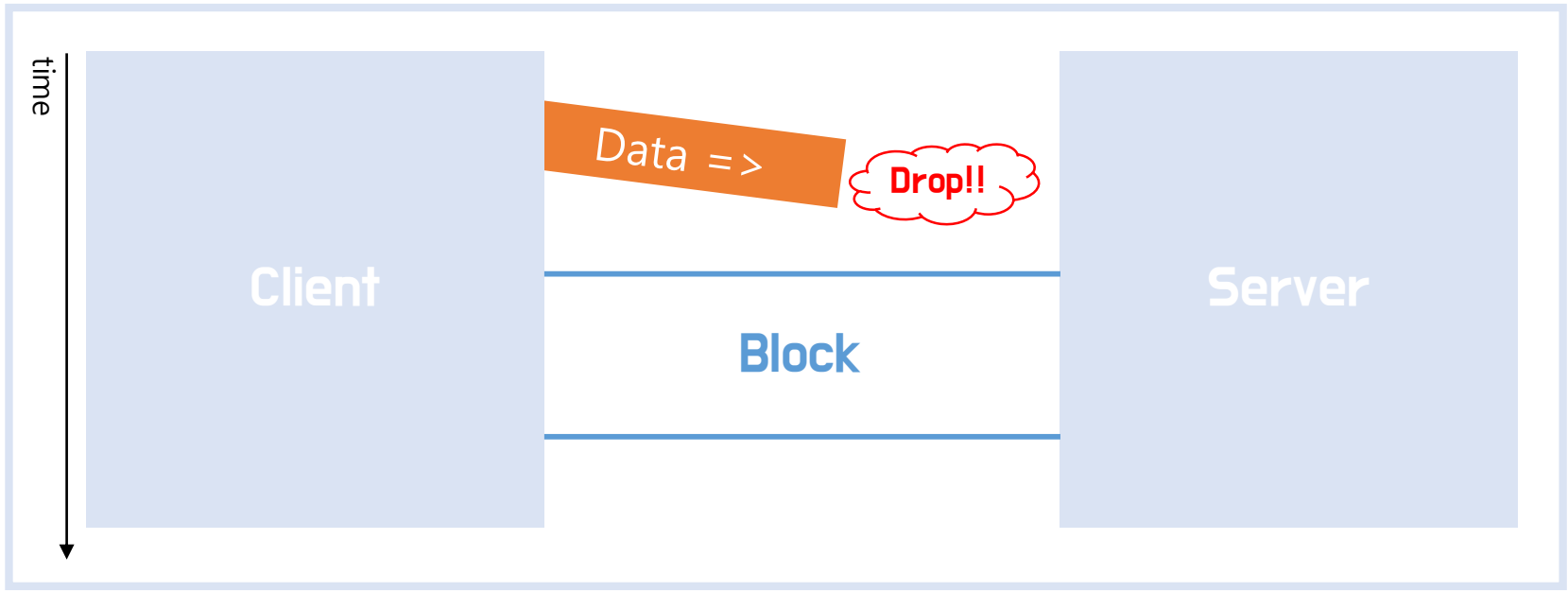
2.5 Stream Control Transmission Protocol (SCTP)

- SCTP provides applications with reliability, sequencing, flow control, full-duplex data transfer like TCP.
- The word "association" is used in SCTP instead of "connection" to avoid connotation.
- Unlike TCP, SCTP is message-oriented. (it provides sequenced delivery of individual records.)

2.5 Stream Control Transmission Protocol (SCTP)

- SCTP can provide multiple streams between connection endpoints.

  => Unlike TCP, when packet loss occurs, all future data transmission is blocked.
  until the loss is repaired.

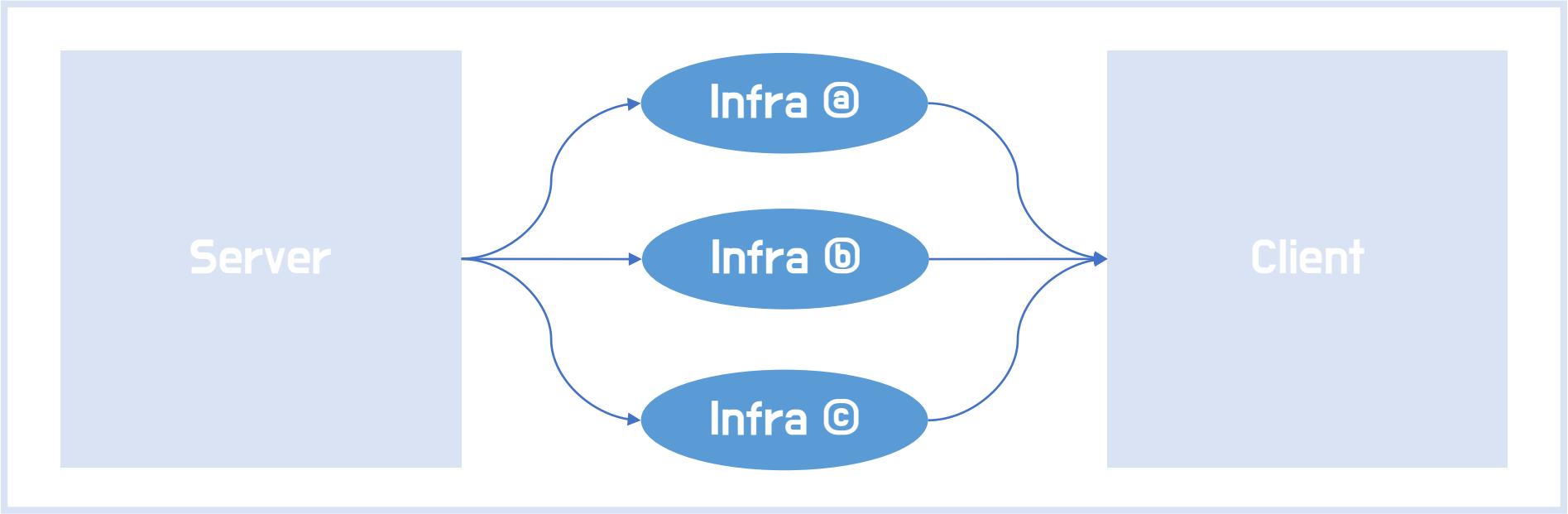


2.5 Stream Control Transmission Protocol (SCTP)

- SCTP also provides a multihoming feature

  => endpoint can have multiple redundant network connections, where each of these networks has a different connection to the Internet infrastructure.

  ➔ this feature provide increased robustness against network failure.



Summary of SCTP

- SCTP provides applications with reliability, sequencing, flow control, full-duplex data transfer like TCP.
- The word "association" is used in SCTP instead of "connection" to avoid connotation.
- Unlike TCP, SCTP is message-oriented. (it provides sequenced delivery of individual records.)
- SCTP can provide multiple streams between connection endpoints.

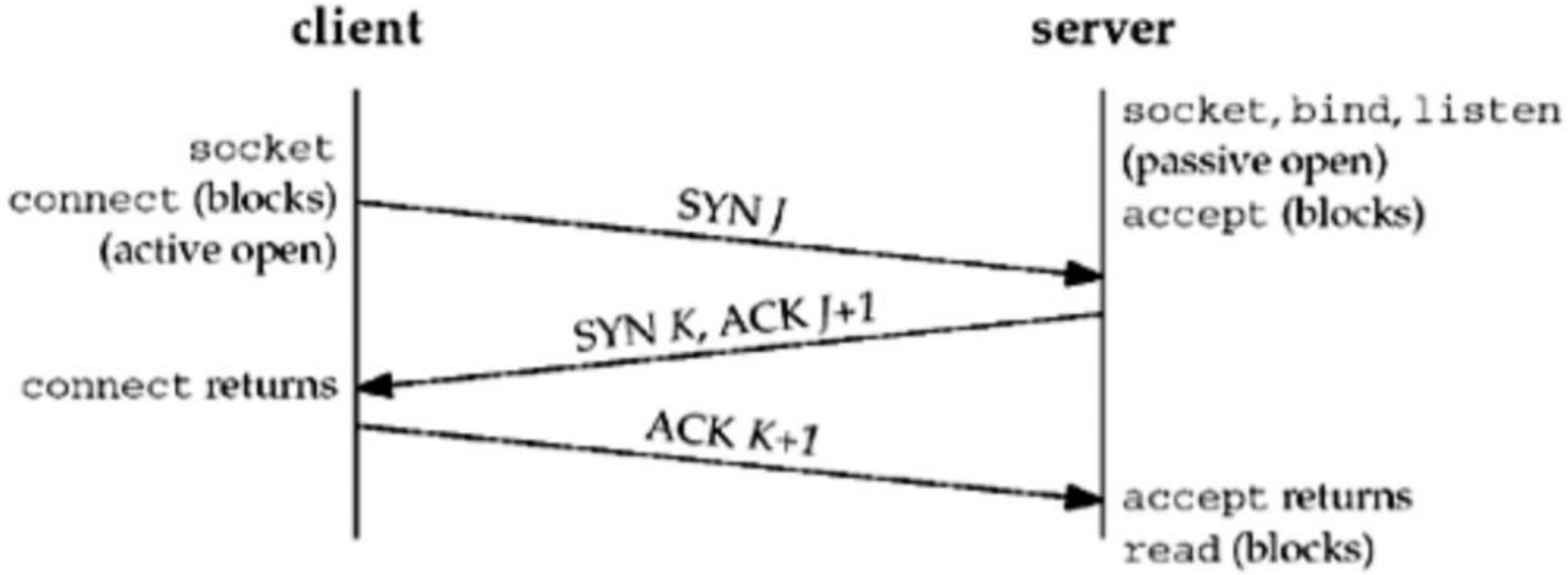
  => Unlike TCP, when packet loss occurs, all future data transmission is blocked. until the loss is repaired.

- SCTP also provides a multihoming feature.

  => endpoint can have multiple redundant network connections, where each of these networks has a different connection to the Internet infrastructure.

2.6 TCP Connection Establishment and Termination

- **Three-Way Handshake** (TCP's connection establishing)
  The minimum number of packets required for this exchange is three; hence,
  this is called TCP'S Three-way handshake

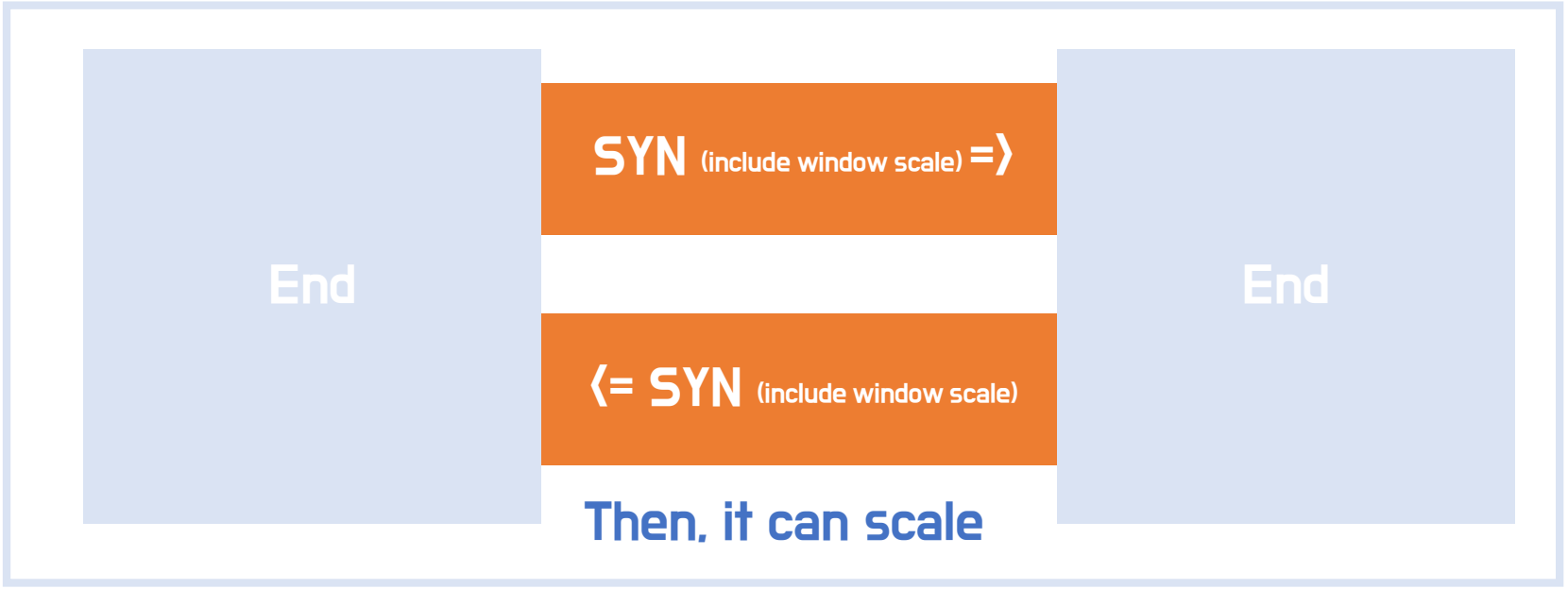


2.6 TCP Connection Establishment and Termination

- **TCP Options**

  - MSS : announces its maximum segment size. TCP_MAXSEG socket option.

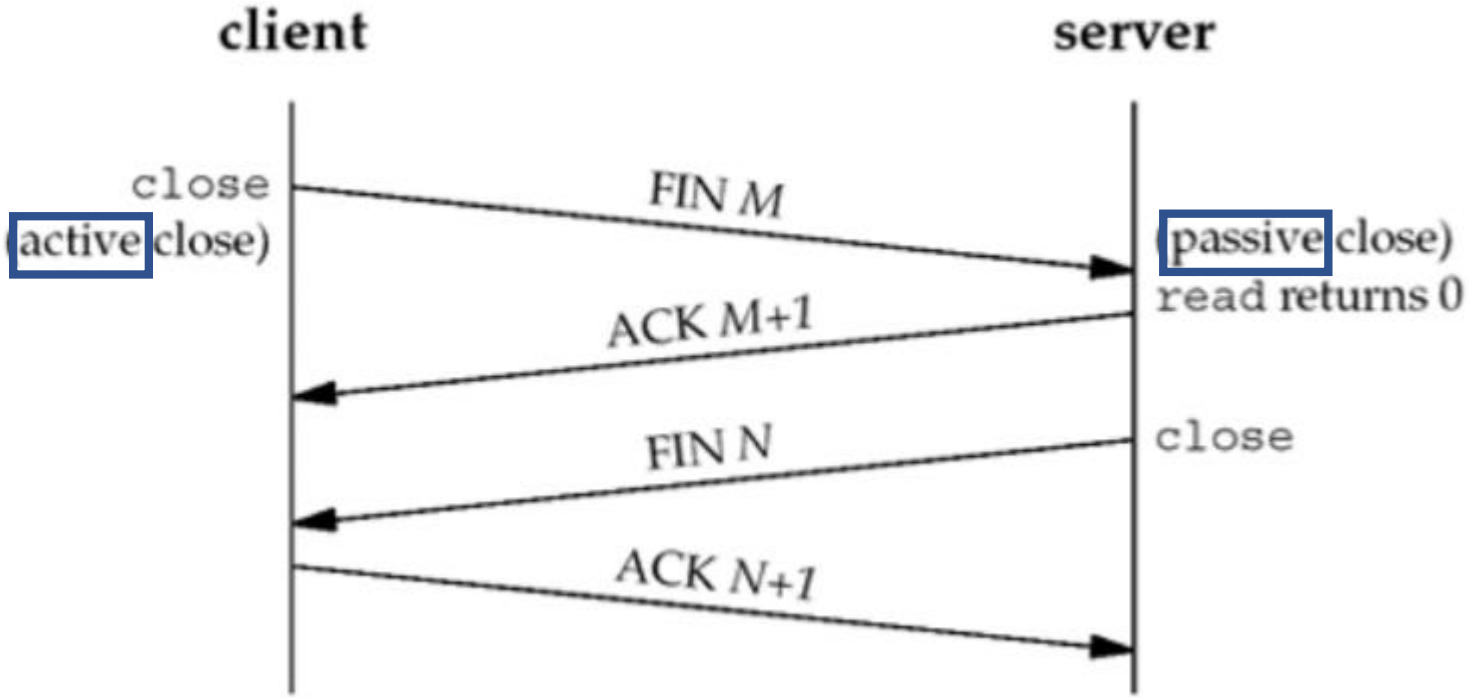
  - Window scale : if you want to make the size of the accepted window larger than 65,535. SO_RCPBUF socket option.
    But, it can scale its windows only if the other end also sends the option with its SYN.



  - Timestamp : to prevent possible data corruption caused by old, delayed, or duplicated segments.

2.6 TCP Connection Establishment and Termination

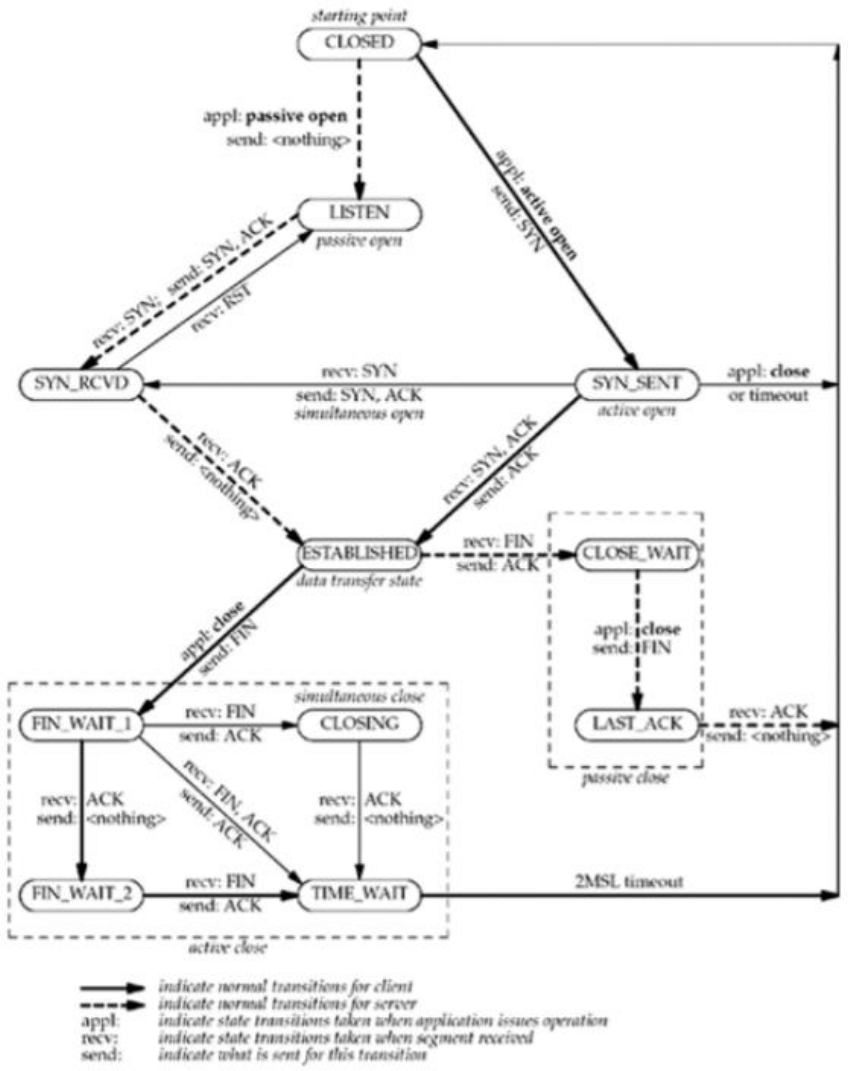
- **TCP Connection Termination**
  it takes four terminate a connection.



It is possible for data to flow from the end doint the passive close to the end doing the active close
=> half-close association.

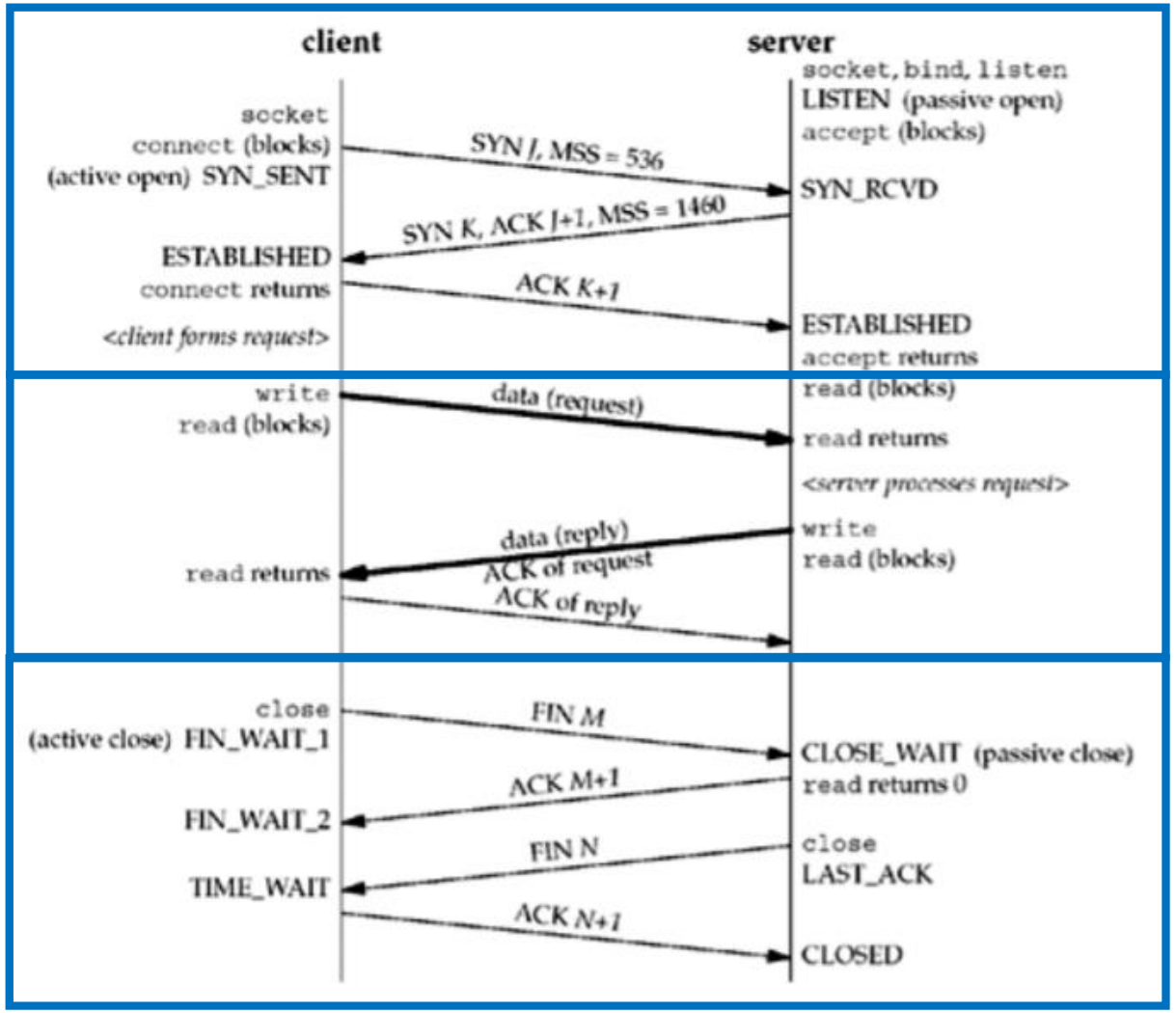
2.6 TCP Connection Establishment and Termination

- **TCP State transition Diagram**



2.6 TCP Connection Establishment and Termination

- **Watching the Packets**



Three-Way Shake

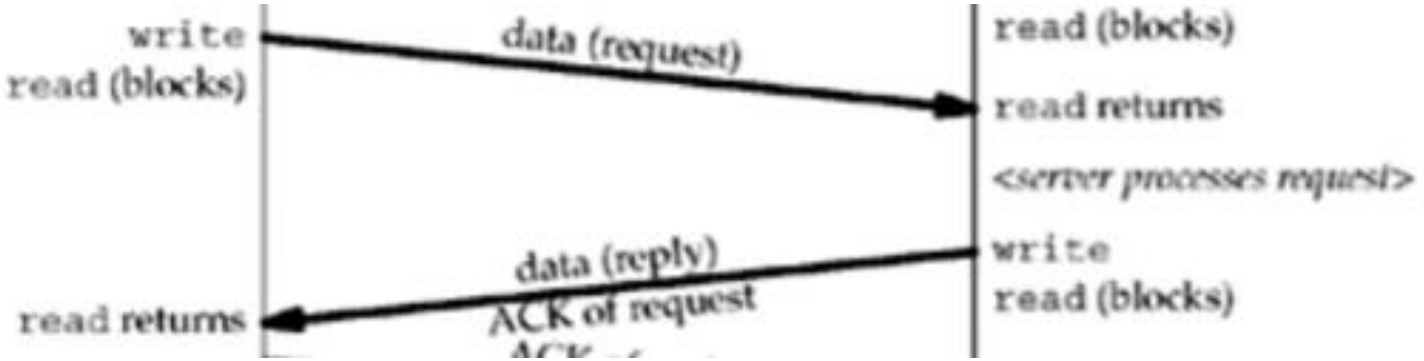
Data Transfer

Termination

What if UDP was used instead?

2.6 TCP Connection Establishment and Termination

- **Watching the Packets** (if UDP was used instead)

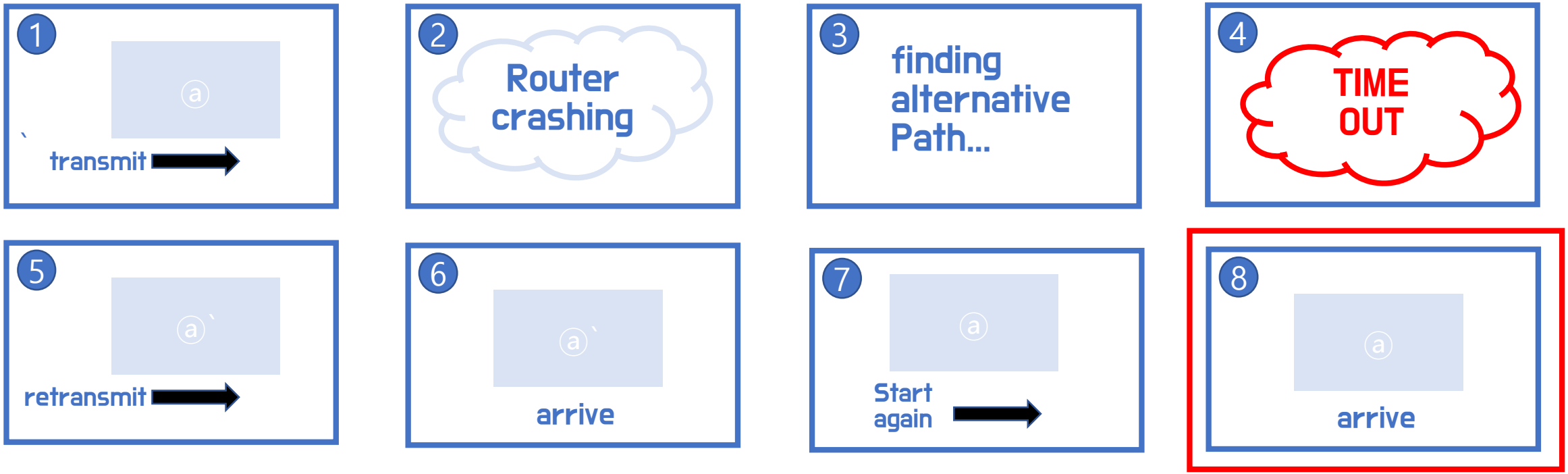


=> Only two packets would be exchanged

This is why many applications still use UDP even though it is not reliable.

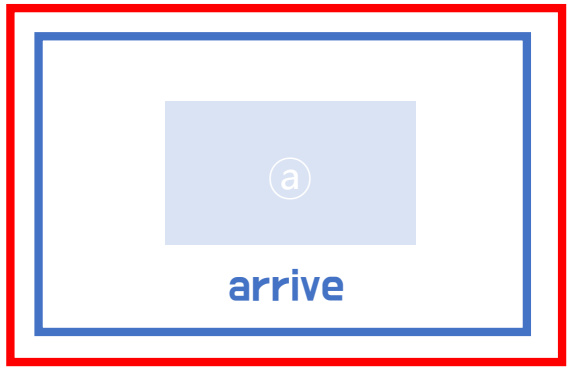
2.7 TIME_WAIT State

- **MSL** (maximum segment lifetime)
  The MSL is the maximum amount of time that any given IP datagram can live in a network.
- The duration that this endpoint remains in TIME_WAIT state is twice the MSL. Called 2MSL.
- Packet gets "lost" in a network.



2.7 TIME_WAIT State

- **MSL** (maximum segment lifetime)
  The MSL is the maximum amount of time that any given IP datagram can live in a network.
- The duration that this endpoint remains in TIME_WAIT state is twice the MSL. Called 2MSL.
- Packet gets "lost" in a network.



=> called "lost duplicate" or "wandering duplicate"

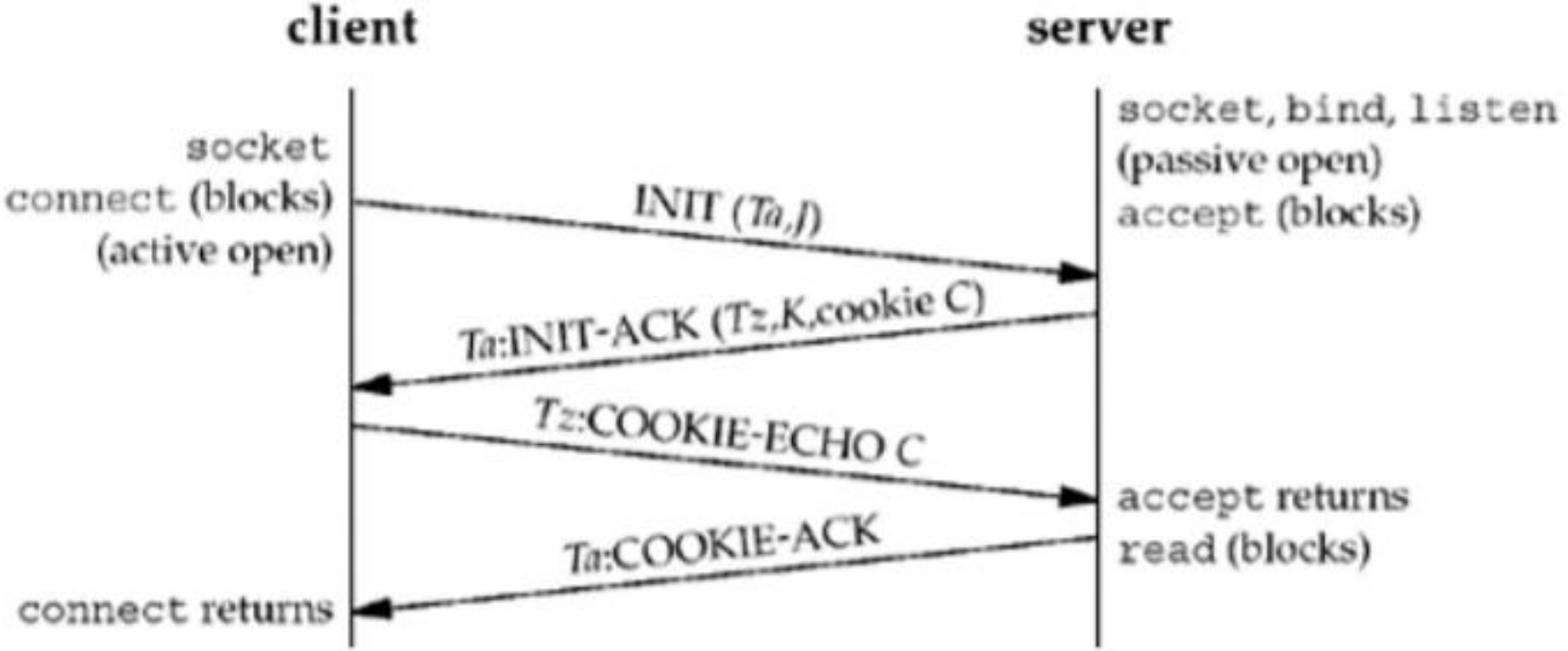
There are two reason for the TIME_WAIT state:

1. To implement TCP'S full-duplex connection termination reliably
2. To allow old duplicate segment to expire in the network

2.8 SCTP Association Establishment and Termination

- **Four-Way Handshake** (SCTP's connection establishing)
  The minimum number of packets required for this exchange is four; hence,
  this is called SCTP'S four-way handshake.

  * state cookie : contains all of the state that the server needs to ensure that association is valid, etc..

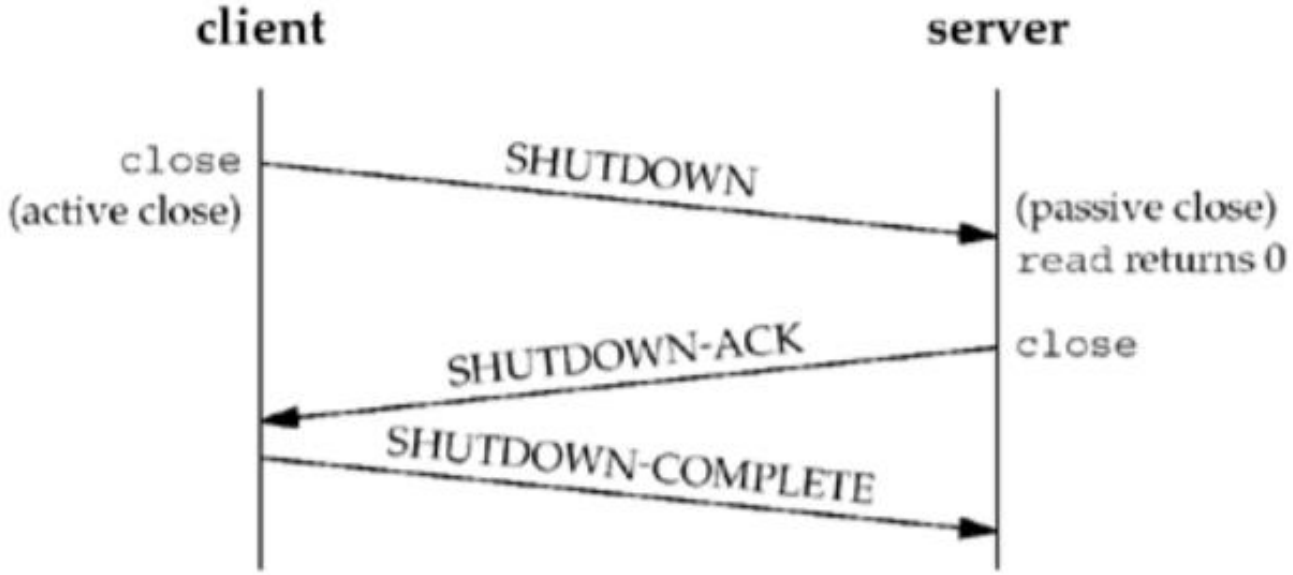


The four-way handshake is used in SCTP to avoid a form of denial-of-service attack. We will discuss later.

2.8 SCTP Association Establishment and Termination

- **Association Termination**

  Unlike TCP, SCTP does not permit a "half-closed" association.
  SCTP doesn't have a TIME_WAIT state like TCP.
  => due to use of verification tags.



2.8 SCTP Association Establishment and Termination

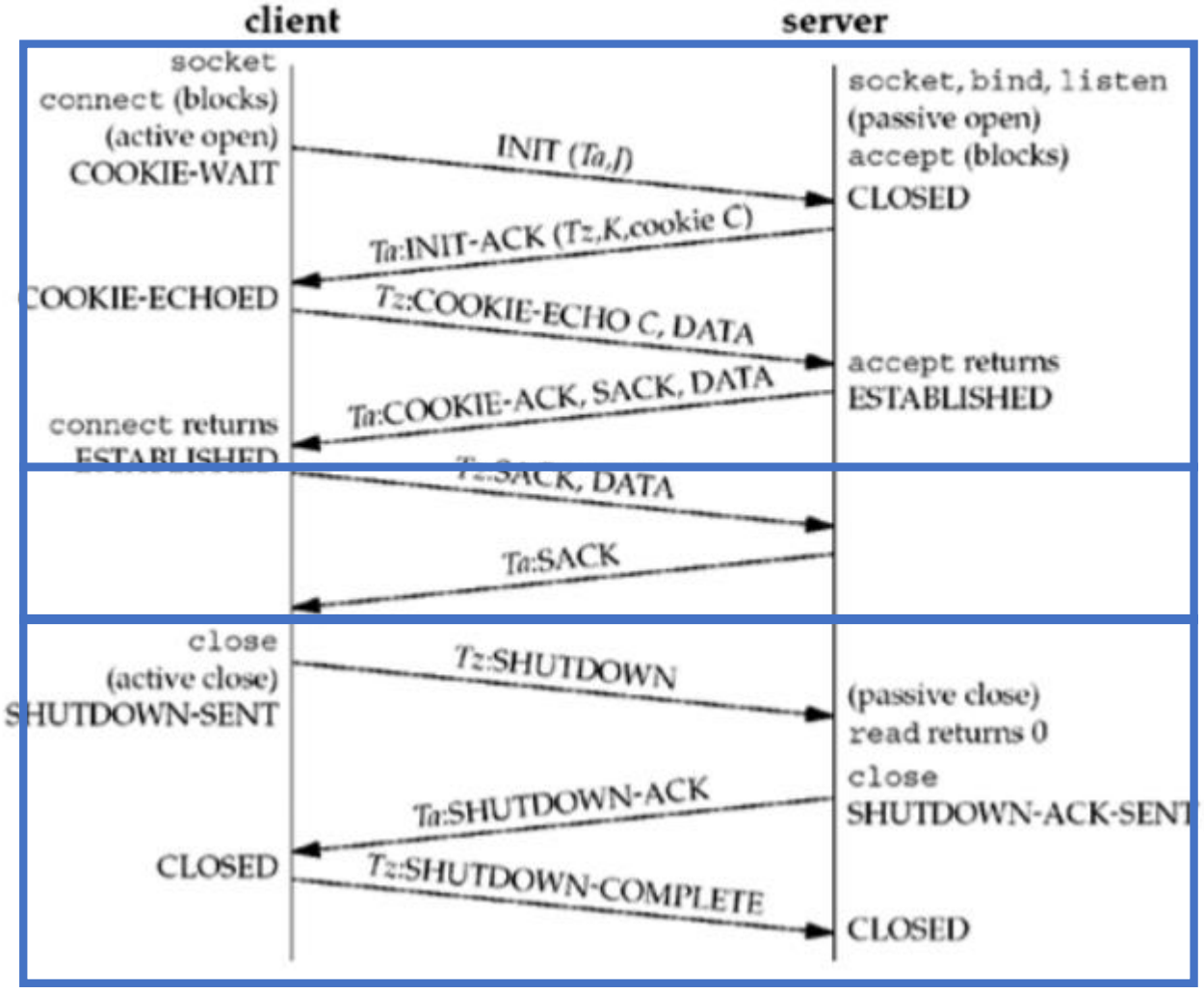
- **SCTP State transition Diagram**

2.8 SCTP Association Establishment and Termination

- **Watching the Packets**

* "chunk"

  INIT, INIT-ACK, COOKIE-ECHO etc..



2.9 Port Numbers

- **Multiple process can be using any given transport : UDP, SCTP, TCP.**

  => all three transport layer use 16-bit integer port numbers to differentiate between the processes.

- **IANA : Internet Assigned Numbers Authority -> maintain a list of port number assignment**

- **Port Numbers**

  The well-known ports : 0~1023. These port numbers are controlled and assigned by the IANA.

  The registered ports : 1024~49151. These are not controlled by the IANA, but the IANA registeres and list the uses of these ports as a convenience to the community

  The dynamical private ports : 49152~65535. nothing about these ports. We call ephemeral(temporal) ports.

- **Socket Pair**

  four-tuple that defines the two endpoints.

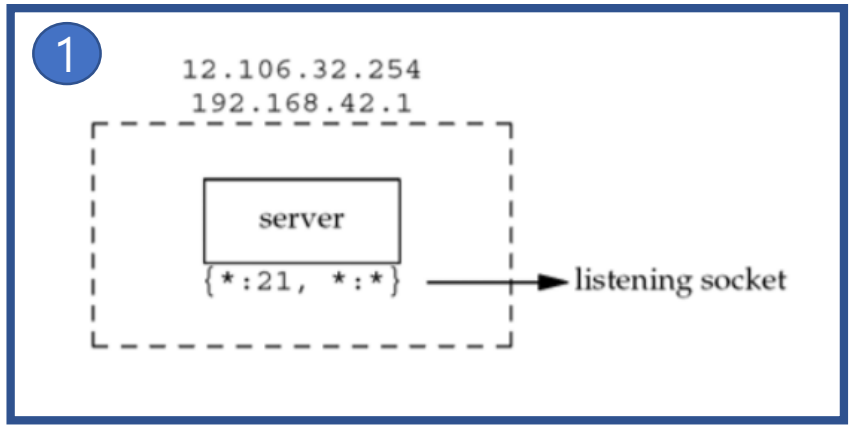
  { local IP address : local port , foreign IP address : foreign port }

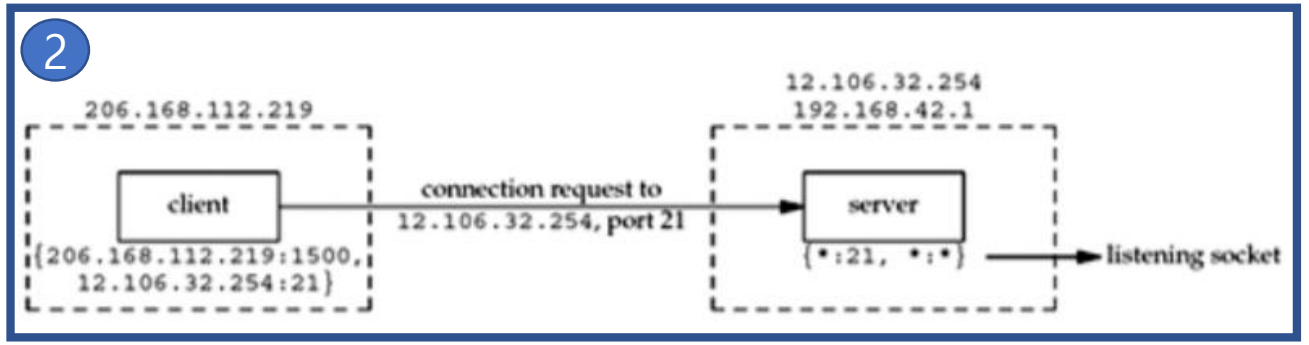
2.10 TCP Port Numbers and Concurrent Servers

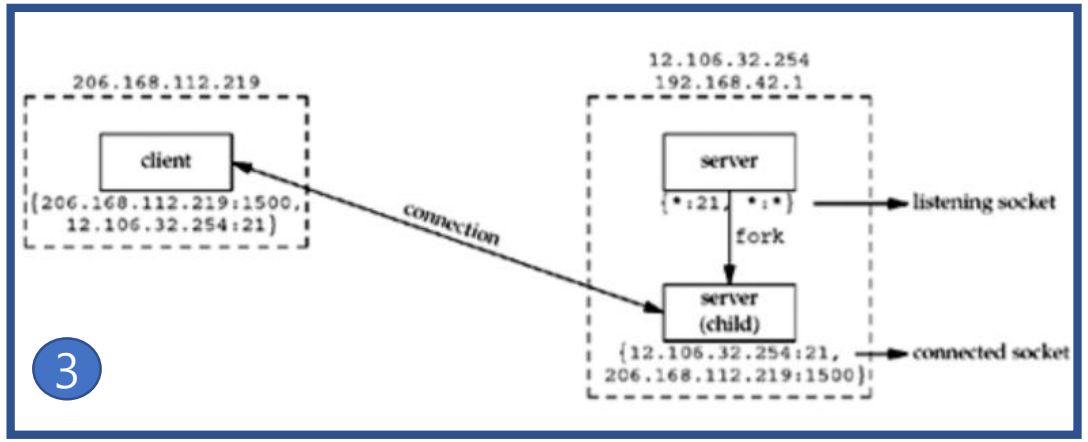
- using the notation {* : 21, * : *} to indicate the server's socket pair. (listening socket)
- When we specify the local IP address as an asterisk(*), it is called the wildcard character.
- It means "any" choice. -> INADDR_ANY

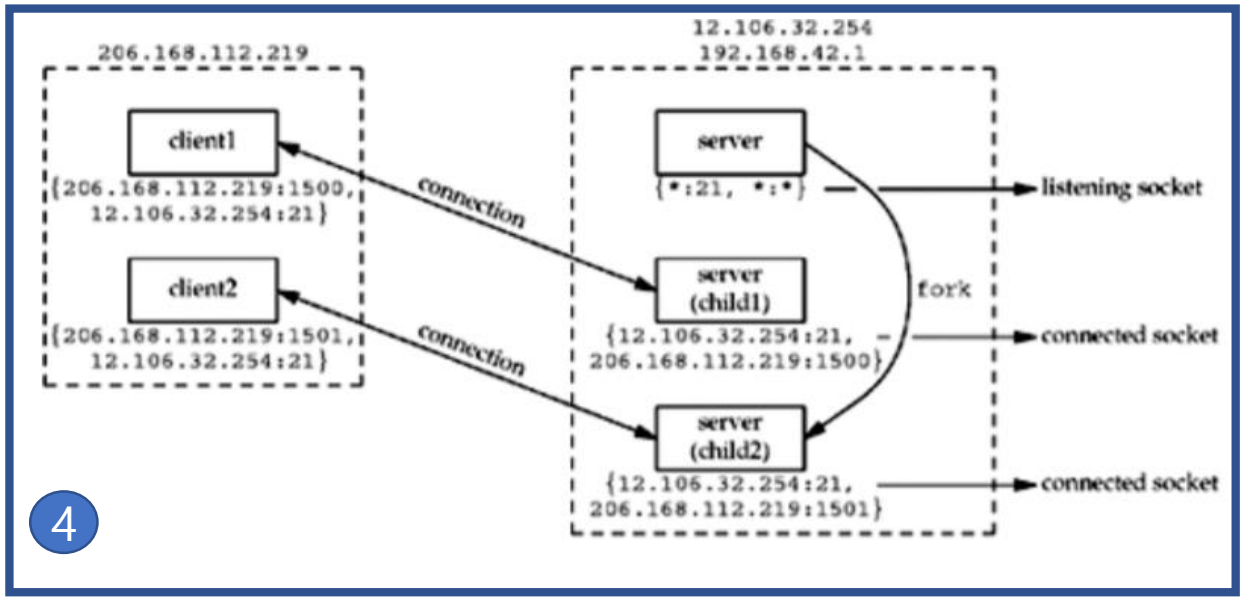
2.10 TCP Port Numbers and Concurrent Servers

- **Connection process** (client and server)









2.11 Buffer sizes and Limitations

- MTU : maximum transmission unit
- MTU is dictated by the Hardware ex) Ethernue MTU is 1500bytes.
- Smallest MTU is called path MTU.
- If it is greater than MTU, IP will proceed with fragmentation.
  Reassembled after reaching final destination.

- DF (Don't Fragment) bit : if it is written, it is not fragmented.
- Minimum reassembly buffer size
  minimum datagram size. -> guaranteed any implementation must support. IPv4 576 bytes, IPv6 1500 bytes.
- MSS (maximum segment size)
  maximum size to send per segment. That announces to the peer TCP the maximum amount of TCP data
  that the peer can send per segment.

2.11 Buffer sizes and Limitations

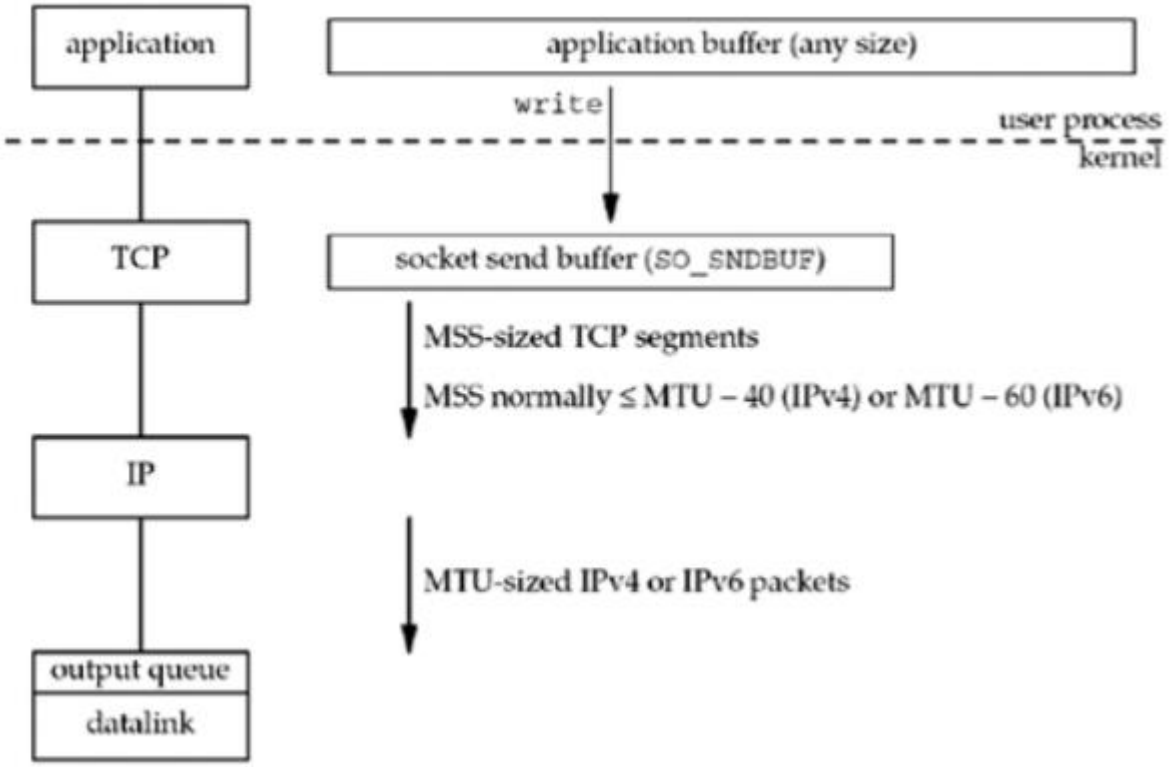
- **TCP Output**

SO_SNDBUF : can change size of this buffer

if socket send buffer inefficient.
process will sleep.
(normal default of a blocking socket)

if TCP receives ACK. TCP will delete
data in socket send buffer.

TCP sends the data s chunk of MSS sizes.
MSS is known by the peer or 536 bytes
if there is no specific configuration.

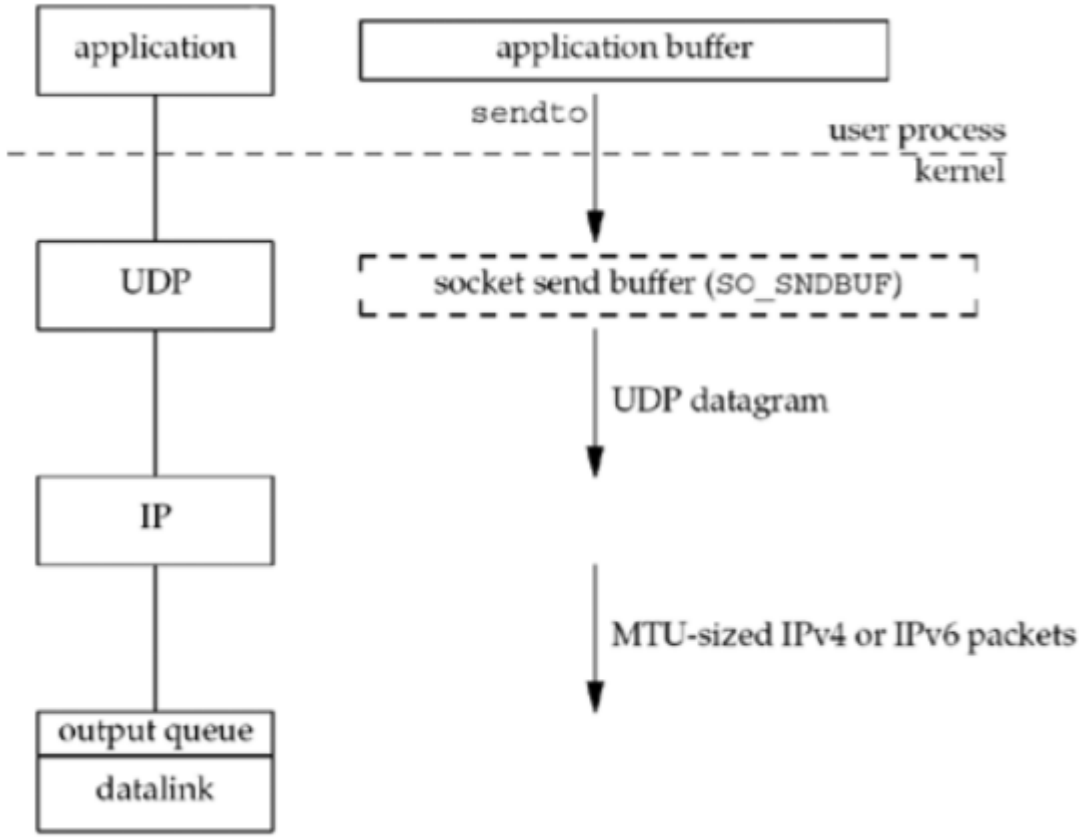


2.11 Buffer sizes and Limitations

- **UDP Output**

  for UDP, socket send buffer does not exist.

  UDP adds an 8-byte header and passes the datagram to IP. IPv4 or IPv6 with additional header send to the datalink output queue.



2.12~2.14는 교재참고