TCP/IP Networks

Dr. Miled M. Tezeghdanti

December 17, 2010

Outline

- TCP/IP
  - IP
  - ARP
  - ICMP
  - TCP/UDP

History

- ARPANET
  - 1969: 4 workstations, Backbone (50 Kbps)
  - ARPA "Advanced Research Project Agency", DoD (1957)
  - DARPA "The Defense Advanced Research Project Agency" (1973)
  - NCP "Network Control Protocol"
- TCP/IP
  - 1973
  - Vint Cerf (Stanford), Bob Khan (DARPA)
    - 1974, first use of the "Internet" word in their paper "Transmission Control Protocol"
  - Use of TCP/IP in ARPANET in 1976
- ARPANET (1984)
  - MILNET, ARPANET(Internet)

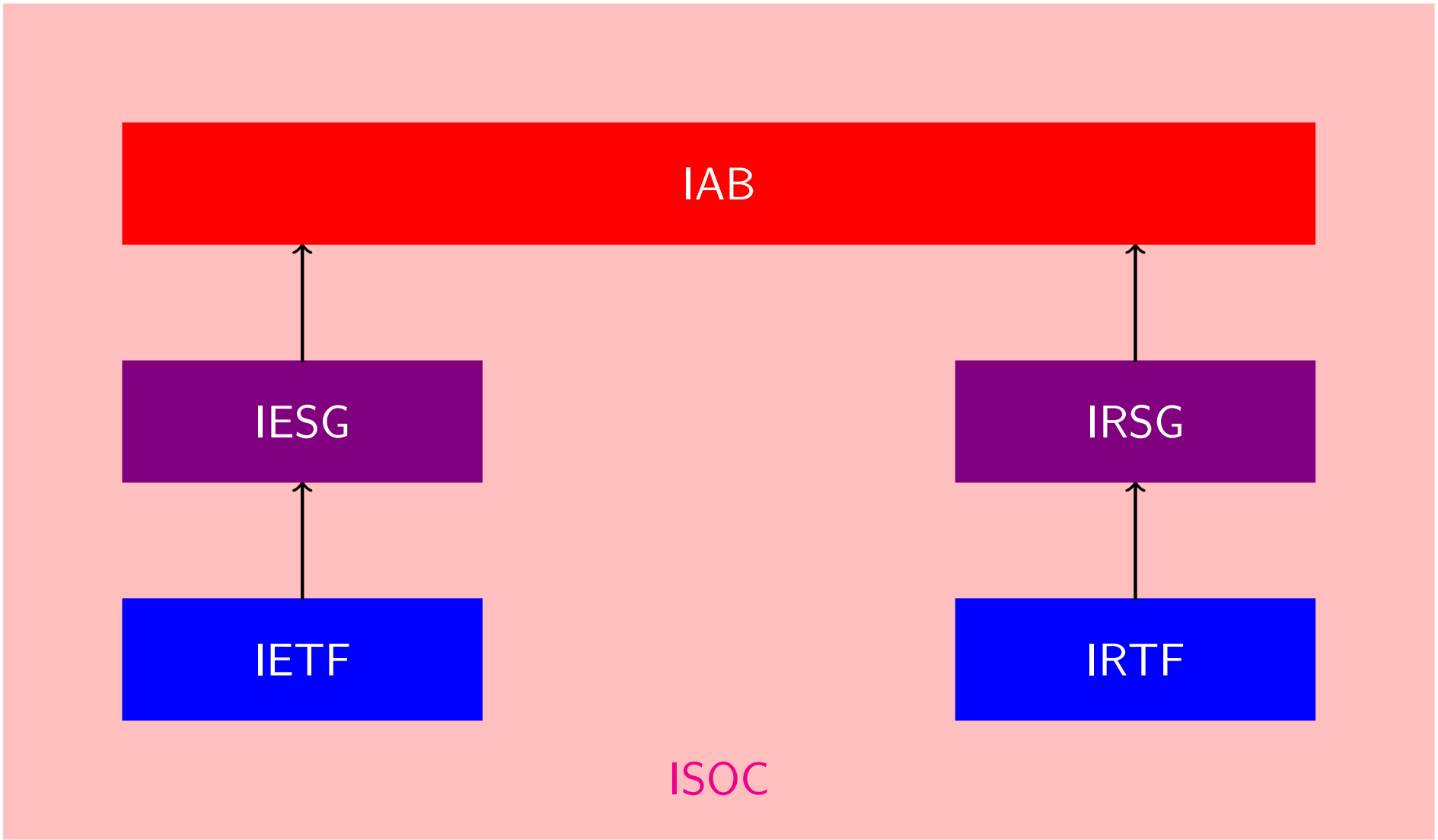
Internet Organizations

- IAB : Internet Architecture Broad (1983)
  - Design, Engineering, and Management of Internet
- IETF : Internet Engineering Task Force (1986)
  - Technical Development of Internet
  - Working Groups
    - Example: ospf (Open Shortest Path First IGP)
  - Managed by IESG: Internet Engineering Steering Group
- IRTF : Internet Research Task Force (1986)
  - Research and long-term development of Internet
  - Research Groups
  - Managed by IRSG: Internet Research Steering Group
    - Example: Routing Research Group

Internet Organizations

- ISOC: Internet Society (1992)
  - Internet Promotion
  - Contains IAB, IETF, and IRTF
- W3C: World Wide Web Consortium (1994)
  - Tim Berners-Lee
  - CERN
  - DARPA, European Commission
- ICANN: The Internet Corporation for Assigned Names and Numbers (1998)
  - Successor of IANA: Internet Assigned Numbers Authority
  - It is the highest international authority for all questions related to domain names, addresses, and protocols.

Internet Organizations

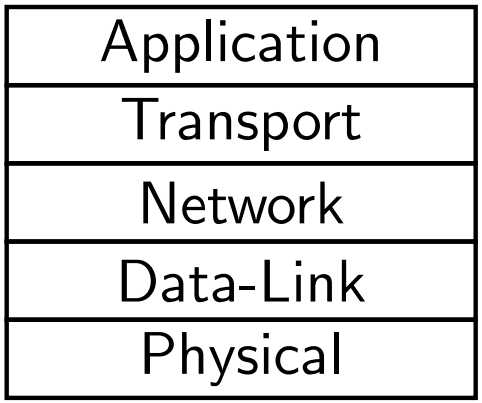


Standards

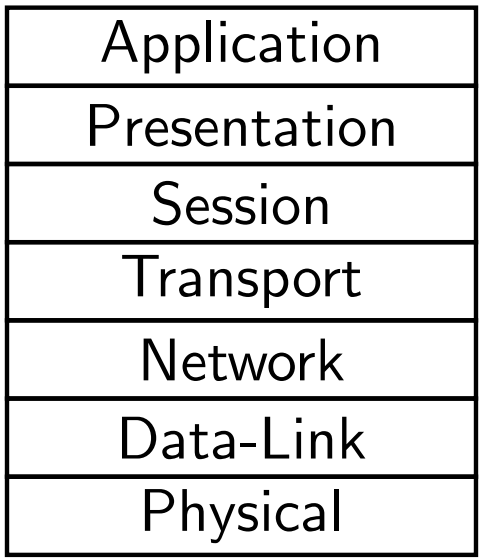
- RFC : Request For Comments
  - RFC 2328
- RFC
  - Internet Draft
  - RFC
    - Prototype
    - Experimental
    - Informational
    - Historic
    - Standard

    1. Proposed Standard
    2. Draft Standard
    3. Internet Standard

TCP/IP Model

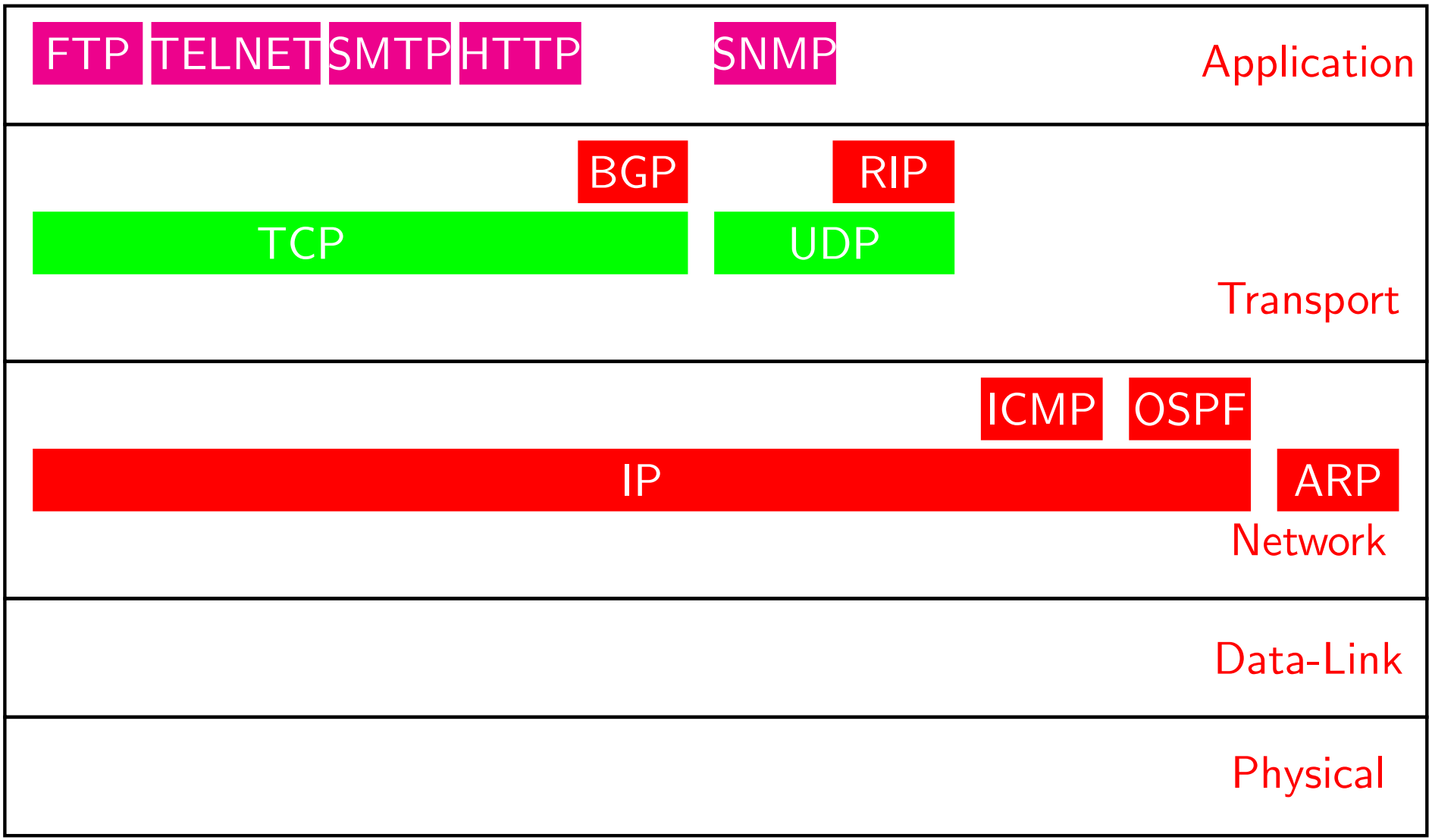


TCP/IP Model



OSI Model

TCP/IP Stack


FTP
TELNET
SMTP
HTTP
SNMP
Application
BGP
RIP
TCP
UDP
Transport
ICMP
OSPF
IP
ARP
Network
Data-Link
Physical

IP Protocol

- Internet Protocol
- RFC 791
- Network Layer
- Interconnection of networks
  - Ethernet
  - Token Bus
  - Token Ring
- Hardware is hidden by the Network layer
  - some exceptions like MTU: Maximum Transmission Unit

Functions provided by the IP layer

- Addressing
- Routing
- Forwarding
- Fragmentation and Reassembly
- Error Notification

IP

- Sending and Receiving of packets
- No retransmissions
  - IP does not provide a reliable forwarding service
- Packets may be:
  - lost
  - dropped
  - duplicated
  - delayed
  - corrupted
  - delivered out of order
- Best effort service
  - Network does his best effort to forward packets

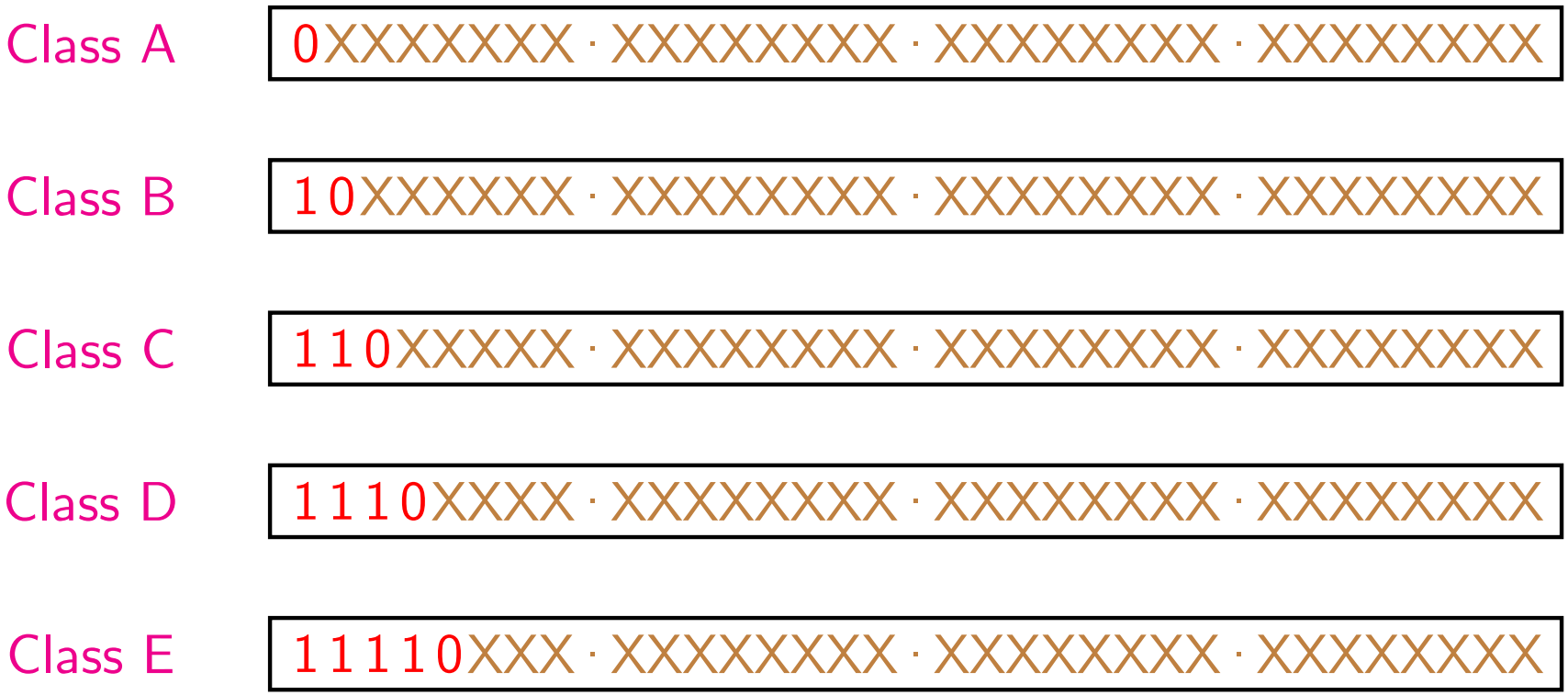
IP Addressing

- An IP address is represented on 32 bits (4 bytes)
- Every equipment has an IP address which identifies it in a unique manner on the network
- IP Address Representation
  - Dotted-Decimal Representation
  - 4 decimal numbers separated by decimal point
  - Value between 0 and 255 for each number
- Example
  - 10000011000100011100000100000001
  - 10000011.00010001.11000001.00000001
  - 131.17.193.1

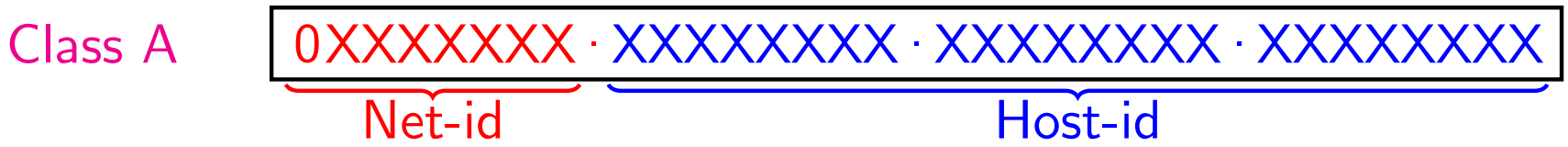
Addressing

- With 32 bits, we can have 232 different IP addresses
- Addressing space is divided in many classes
- An address is divided in two parts
  - The first part represents the address of the network connected to the host (workstation)
  - The second part represents the address of the host (workstation) on the network
  - Hosts connected to the same network have the same network address (first part of the address is the same for all these hosts)
- Mask
  - 32 bits
  - Allows the distinction between the network part and the host part of the address (1 if the bit belongs to the net-id, 0 otherwise)

Addressing


Class A
0XXXXXXX · XXXXXXXX · XXXXXXXX · XXXXXXXX
Class B
10XXXXXX · XXXXXXXX · XXXXXXXX · XXXXXXXX
Class C
110XXXXX · XXXXXXXX · XXXXXXXX · XXXXXXXX
Class D
1110XXXX · XXXXXXXX · XXXXXXXX · XXXXXXXX
Class E
11110XXX · XXXXXXXX · XXXXXXXX · XXXXXXXX

Addressing: Class A

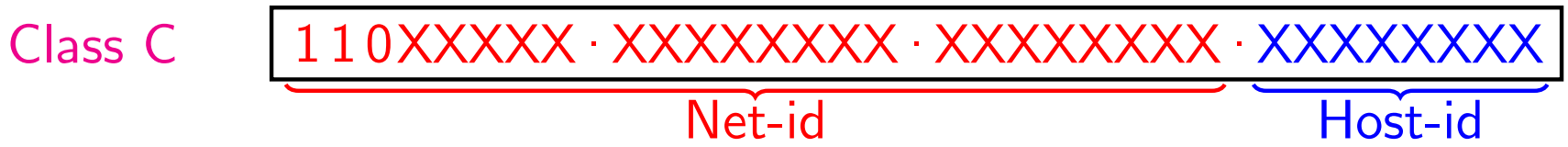


- Net-id: 8 bits
  - $2^7 - 2 = 126$ class A networks
  - $1 \ldots 126$ (0 and 127 reserved)
- Host-id: 24 bits
  - $2^{24} - 2 = 16777214$ hosts
- Example: 12.5.2.3

Addressing: Class B

- Net-id: 16 bits
  - $2^{14} = 16384$ class B networks
  - $128 \ldots 191$
- Host-id: 16 bits
  - $2^{16} - 2 = 65534$ hosts
- Example: 130.20.6.1

Addressing: Class C



- Net-id: 24 bits
  - $2^{21} = 2097152$ class C networks
  - $192 \ldots 223$
- Host-id: 16 bits
  - $2^8 - 2 = 254$ hosts
- Example: 195.16.26.17

Private Addressing

- Private Address: is an IP address that cannot be used to interconnect a host to Internet
- Private addresses:
  - Class A
    - 10.0.0.0 - 10.255.255.255
  - Class B
    - 172.16.0.0 - 172.31.255.255
  - Class C
    - 192.168.0.0 - 192.168.255.255
- Examples
  - 10.1.3.2
  - 172.16.8.17
  - 192.168.20.39

Loop-back and Multicast Addresses

- 127.0.0.1
  - Loop-back address
  - Inter-process communication on the same host
  - Test the protocol stack without having a network connection
  - Packets sent to this address will be directly sent to the local host
- Class D: Multicast addresses
  - 224.0.0.5: All OSPF Routers on the same LAN
- Class E: reserved for future use

Specific Addresses

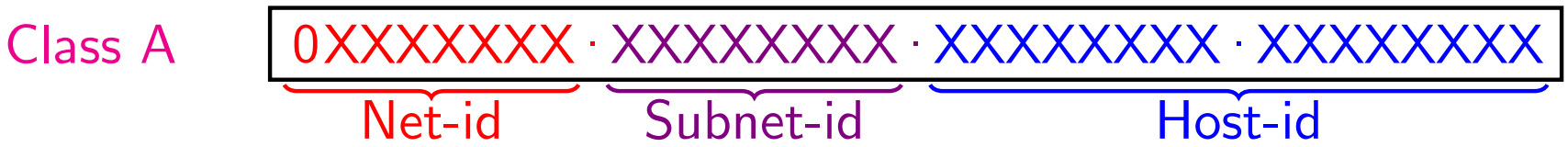
- 0.0.0.0
  - An IP address used by a host when it does not know its IP address (This host)
- 0.A.A.A, 0.0.B.B, 0.0.0.C
  - When the network-id is equal to 0, this indicates the network that is directly connected to the host (This host on this network)
  - 0.5.3.4, 0.0.75.3, 0.0.0.13
- 255.255.255.255
  - Broadcast Address on the LAN
- A.255.255.255, B.B.255.255, C.C.C.255
  - Broadcast Address on a distant network (A.0.0.0, B.B.0.0, C.0.0.0)
  - 12.255.255.255, 130.24.255.255, 195.15.63.255

IP Subnet

- Hosts on the same network must have the same network address
  - Class A: 16.12.85.1 and 16.18.74.12 are on the same IP network
  - Class B: 131.16.74.8 and 131.16.5.5 are on the same IP network
  - Class C: 194.3.5.4 and 194.3.5.6 are on the same IP network
- A class A network may contain up to 16777214 hosts
- A class B network may contain up to 65534 hosts
- To simplify the management of class A and B networks, the concept of subnet was introduced.

IP Subnet

- We can split a class A IP network to 256 different class B subnets (actually 254 class B subnets)
  - We use the 8 most significant bits of the host-id to address the subnet
  - 16.0.0.0: class A network
    - 16.1.0.0: first subnet
    - 16.2.0.0: second subnet
    - ...
    - 16.254.0.0: 254th subnet



IP Subnet

- We can split a class B IP network to 256 different class B subnets (actually 254 class C subnets)
  - We use the 8 most significant bits of the host-id to address the subnet
  - 131.23.0.0: class B network
    - 131.23.1.0: first subnet
    - 131.23.2.0: second subnet
    - ...
    - 131.23.254.0: 254th subnet

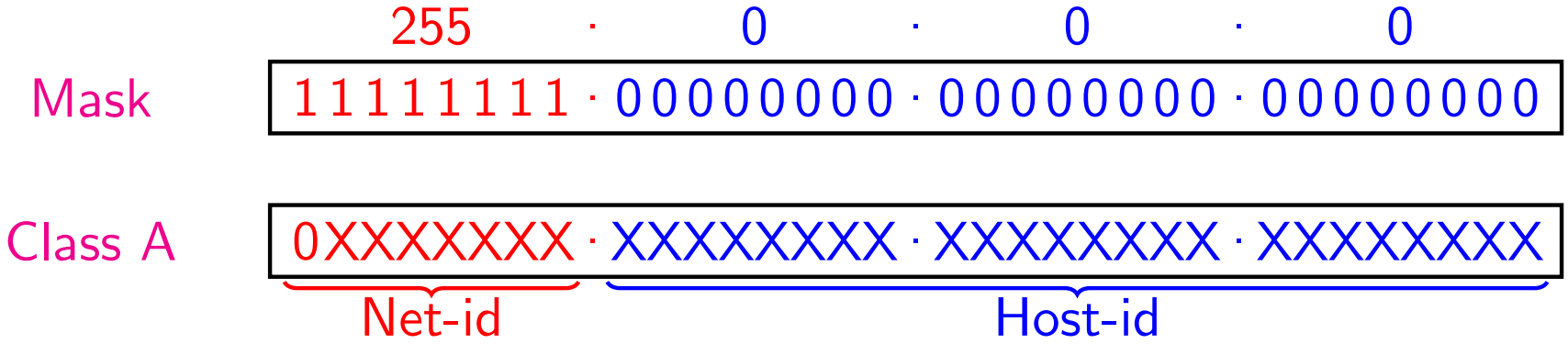
Mask

- We need a supplementary mechanism to distinguish between the network-id (network and subnet) and the host-id
  - Before, it is sufficient to determine the class of the address to distinguish between the network-id part and the host-id part
- Network Mask: distinguish between the net-id (network and subnet) and the host-id
  - 32 bits (same size as an IP address)

Mask

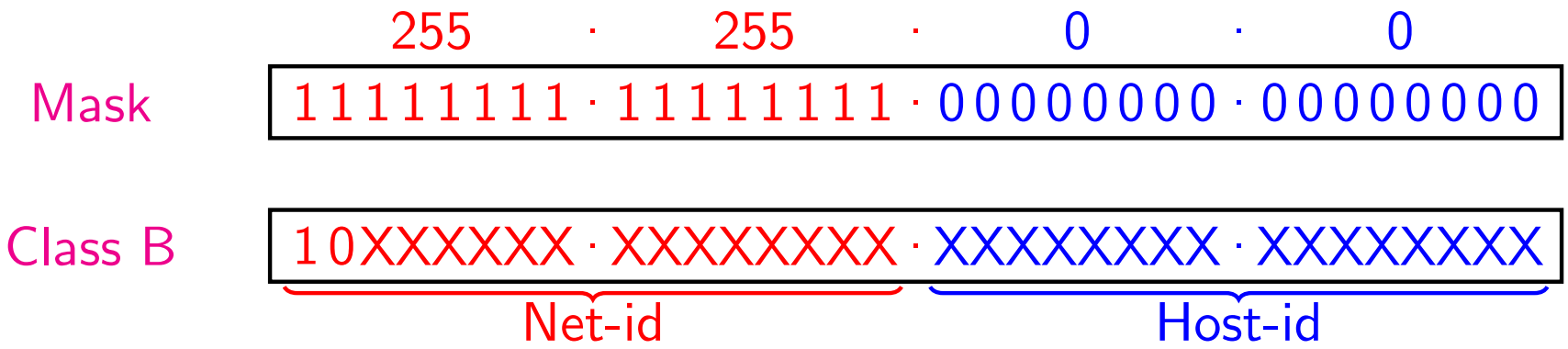
- How we compute the network mask?
- For each bit of order i (i = 0..31) of the IP address,
  - Affect to the bit i of the mask the value
    - 1 if the bit of order i is in the net-id (network and subnet) part
    - 0 if the bit of order i is in the host-id part
- Network mask is represented in the same manner as an IP address
  - Example: 255.255.0.0

Class A Mask



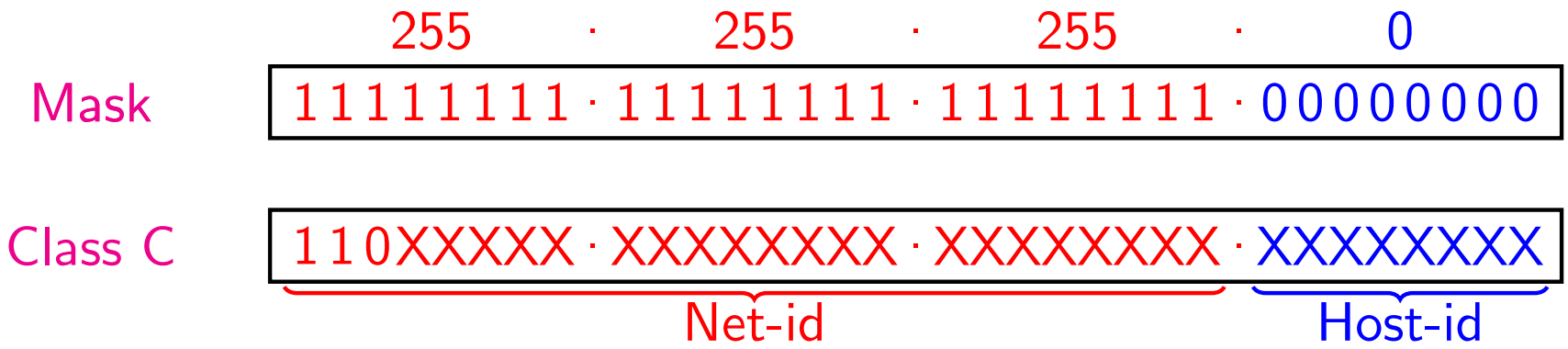
- Example
  - 16.0.0.0
    - Mask 255.0.0.0
  - 23.0.0.0
    - Mask 255.0.0.0

Class B Mask



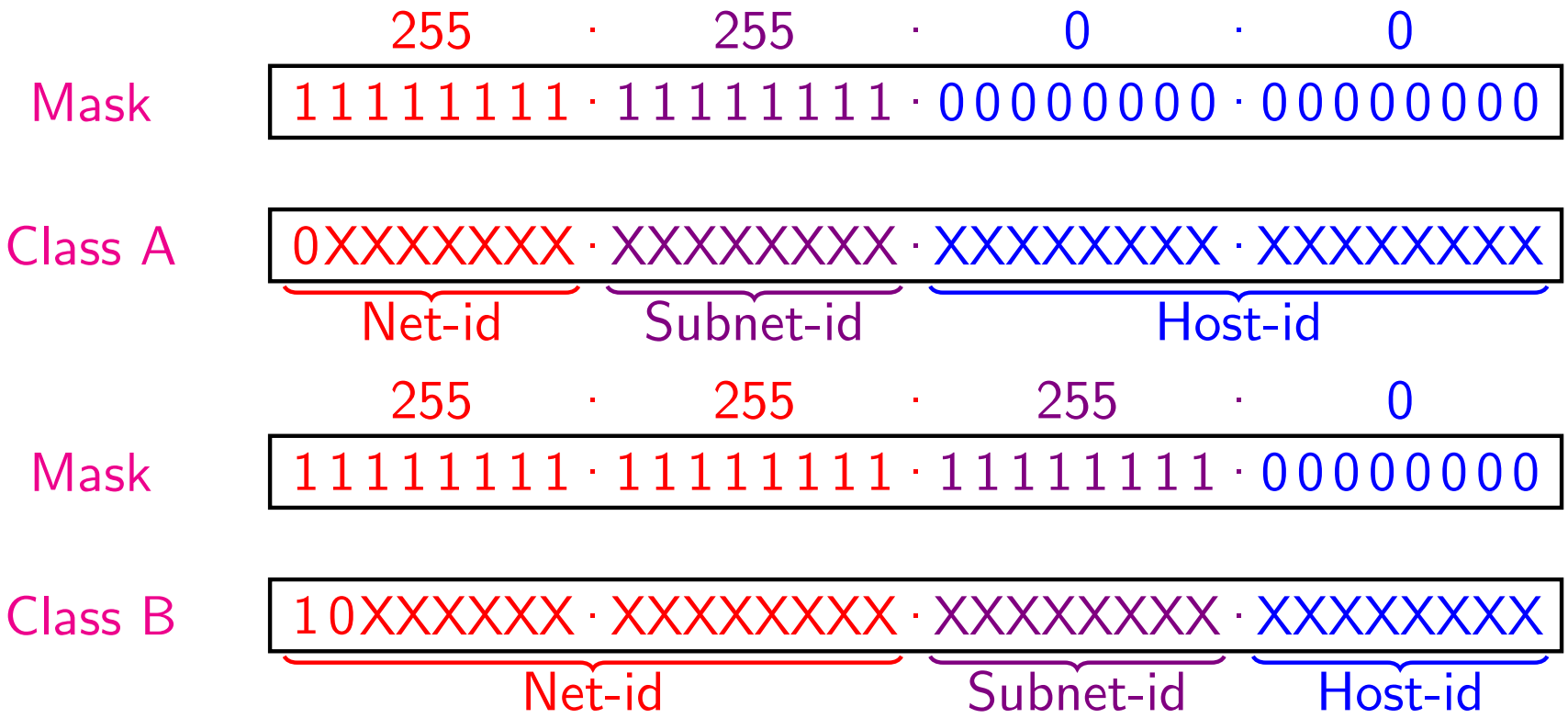
- Example
  - 131.23.0.0
    - Mask 255.255.0.0
  - 136.74.0.0
    - Mask 255.255.0.0

Class C Mask



- Example
  - 195.12.14.0
    - Mask 255.255.255.0
  - 196.72.53.0
    - Mask 255.255.255.0

Subnet Mask


255 · 255 · 0 · 0
Mask
11111111 · 11111111 · 00000000 · 00000000
Class A
0XXXXXXX · XXXXXXXX · XXXXXXXX · XXXXXXXX
Net-id
Subnet-id
Host-id
255 · 255 · 255 · 0
Mask
11111111 · 11111111 · 11111111 · 00000000
Class B
10XXXXXX · XXXXXXXX · XXXXXXXX · XXXXXXXX
Net-id
Subnet-id
Host-id

Subnet Mask

- Class A Subnets
  - 16.0.0.0: class A network, mask 255.0.00
    - 16.1.0.0: first subnet, mask 255.255.0.0
    - 16.2.0.0: second subnet, mask 255.255.0.0
    - ...
    - 16.254.0.0: 254th subnet, mask 255.255.0.0
- Class B Subnets
  - 131.23.0.0: class B network, mask 255.255.0.0
    - 131.23.1.0: first subnet, mask 255.255.255.0
    - 131.23.2.0: second subnet, mask 255.255.255.0
    - ...
    - 131.23.254.0: 254th subnet, mask 255.255.255.0

Variable Length Mask

- Example
  - We have the class B address 131.23.0.0 that we want to use to address 10 different subnets each one containing more than 256 hosts.
  - Solution
    - In fact, we need only 4 bits to address the different subnets, so we can let 12 bits for addressing hosts on subnets. If we take 8 bits as usual, we can not address hosts on subnets that have more than 256 hosts connected to them.
    - Network mask is computed using the same algorithm
    - Mask = 11111111.11111111.11110000.00000000
    - Mask = 255.255.240.0

Variable Length Mask

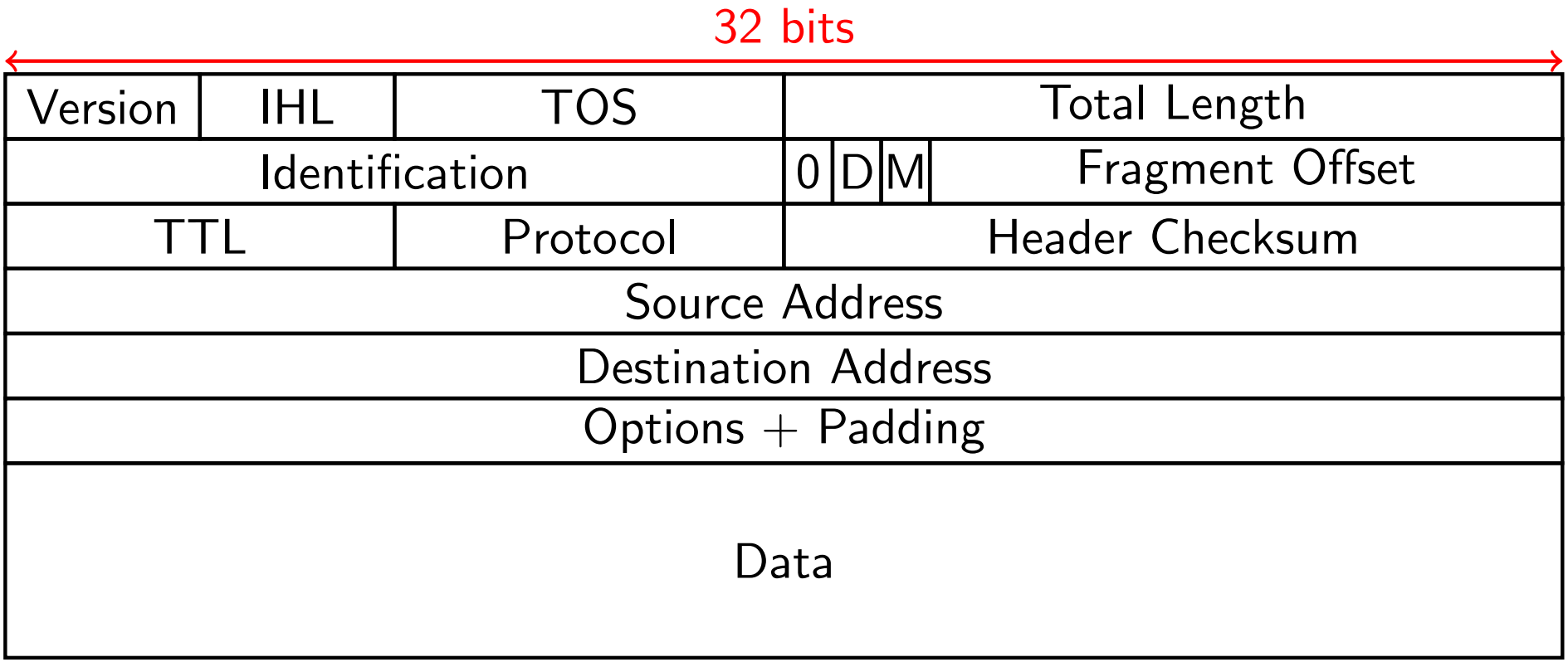
- The 10 subnets have all the same mask 255.255.240.0
  - 0000 et 1111 are not used
  - 131.23.16.0 = 10000011.00010111.00010000.00000000
  - 131.23.32.0 = 10000011.00010111.00100000.00000000
  - 131.23.48.0 = 10000011.00010111.00110000.00000000
  - 131.23.64.0 = 10000011.00010111.01000000.00000000
  - 131.23.80.0 = 10000011.00010111.01010000.00000000
  - 131.23.96.0 = 10000011.00010111.01100000.00000000
  - 131.23.112.0 = 10000011.00010111.01110000.00000000
  - 131.23.128.0 = 10000011.00010111.10000000.00000000
  - 131.23.144.0 = 10000011.00010111.10010000.00000000
  - 131.23.160.0 = 10000011.00010111.10100000.00000000

Variable Length Mask

- With a variable length mask, we can also split a class C address to many subnets
- Example
  - 195.5.6.0
    - 2 subnets

1. We need 2 bits (all 0s and all 1s are not used)
2. Mask: 11111111.11111111.11111111.11000000
3. Mask: 255.255.255.192
4. 195.5.6.64
5. 195.5.6.128

IP Packet Format



IP Packet Format

- Version
  - 4 bits
  - Protocol Version
  - Current Version: 4
  - IPv4
- IHL: IP Header Length
  - 4 bits
  - Size of the IP header in 32 bit words
  - Determines the start of the Data field
  - Minimal size is 5

IP Packet Format

- TOS: Type Of Service
  - 8 bits
  - Indicates the type of service requested by the packet
  - Not used
  - Always set to 00000000
  - New re-use of TOS field (Diff-serv Architecture)
- Total Length
  - 16 bits
  - Total length (Header + payload) of the IP packet expressed in bytes
  - Maximal Length of an IP packet: 64 Kbytes

IP Packet Format

- Identification
  - 16 bits
  - Identifies the IP packet
  - It allows the identification of fragments of the same packet
- Fragment Offset
  - 13 bits
  - It indicates the position of the current fragment from the first fragment
  - The offset is measured in 8 byte words (64 bits)
  - The offset of the first fragment is 0

IP Packet Format

- DF: Don't Fragment
  - One bit
  - If DF = 1, don't fragment the packet
  - If DF = 0, the packet may be fragmented when it is needed
- MF: More Fragments
  - One bit
  - If MF = 0, It is the last fragment of the packet
  - If MF = 1, there is more fragments after this fragment

IP Packet Format

- TTL: Time To Live
  - 8 bits
  - Time remained for the packet before it will be dropped by the network if it doesn't reach its destination
  - Each traversed router decrements the packet TTL by 1 and drops the packet if its TLL equals 0
  - It assures that the packet won't loop indefinitely in the network
- Protocol
  - 8 bits
  - It determines the protocol that must process data transported by the IP packet
  - TCP $= 6$
  - UDP $= 17$

IP Packet Format

- Header Checksum
  - 16 bits
  - Error control limited to the header of the packet
  - 1's Complement
  - Since TTL value changes from hop to hop, Checksum is checked and computed at each processing of the IP header
  - Algorithm:
    - Checksum is the 1's complement of the sum over 16 bits of 1's complements of all 16 bit words of the IP header, including Checksum (Checksum value used in the computation is 0)
  - Algorithm simple and easy to implement

IP Packet Format

- Source Address
  - 32 bits
  - IP address of the host that sends the packet
  - Always, it is a unicast address
  - Over Internet, source address must be a public domain address
- Destination Address
  - 32 bits
  - IP address of the host that will receive the packet
  - May be a unicast/multicast/broadcast address
  - Over Internet, destination address must be a public domain address

IP Packet Format

- Options
  - Type, Length, Value (TLV Encoding)
  - Record Route
  - Explicit Route
- Padding
  - Used to have an IP header length multiple of 32 bit word
  - Padding bytes are set to 0

IP Forwarding

- IP Forwarding
  - It is the set of operations performed by a router over an IP packet in order to send it towards its destination
  - Router
    - Equipment that supports the IP stack and has many network interfaces and performs packet forwarding
    - A workstation may have many network interfaces and supports the IP stack without playing the role of router

1. Multihoming

IP Forwarding

- Each router that receives an IP packet performs following operations over the packet
  - It checks the Checksum, the packet is dropped if an error is detected
  - It decrements TTL by 1 and drops the packet if the TTL becomes 0
  - It computes the new Checksum
  - It looks the routing table to determine the next hop that is on the route of the packet towards its destination
  - If it does not find required routing information to send the packet towards its destination, it drops the packet
  - It sends the packet towards its destination and eventually to its destination if it is on the same network

Routing Table

- Contains required information to forward IP packets towards their destinations
- The routing table may be populated
  - Manually by the network administrator
    - Static Routing
    1. Route command under Unix
  - Automatically using a routing protocol
    - Dynamic Routing
    1. RIP, OSPF, BGP

Routing Table

Destination	Next Hop	Cost
12.0.0.0	196.46.7.2	4
133.15.0.0	198.19.63.2	1
196.46.7.0	196.46.7.1	1
198.19.63.0	198.19.63.1	1

CIDR

- Classless Inter Domain Routing
  - A new addressing scheme that allows
    - Efficient address allocation
    - Routing table size reduction
    - A solution to B addresses' shortcomings
  - Class concept is no longer used
  - Problem:
    - Affect IP addresses to an IP network having 1000 hosts
    - A class C address is not sufficient
    - A class B address may solve the problem, but!

    1. Wasting
    2. There is not enough available class B addresses

  - Solution: Use 4 class C addresses

CIDR

- The attribution of many class C addresses to a company avoids the wasting of IP addresses
  - Explosion of the size of routing tables
    - An entry for each class C network
    - $221 = 2097152$ different class C networks
  - Remedy
    - Class C addresses allocated to a given company must be contiguous in order to use the super-netting concept
    - Replace many contiguous network addresses by a single address accompanied by the number of bits starting from left that are identical for all addresses

1. This new presentation is called IP prefix and the number of bits is called prefix length

CIDR

- Example
  - The following 4 class C addresses are contiguous
    - 195.15.16.0, 195.15.17.0, 195.15.17.0, 195.15.18.0
  - Correspondent prefix is 195.15.16.0/22
  - The prefix length is 22
    - The 22 left bits of the 4 addresses are identical
  - The 4 addresses will be represented by a single entry in the routing table
- The same concept is generalized to reduce the size of the routing table (100000 entries)
  - Aggregation: allows the substitution of many routing table entries by a single prefix if all entries have the same next-hop

CIDR

- Can you do better!
  - Sometimes many routing table entries are contiguous addresses and they have the same next-hop except one or two entries that have a different next-hop
  - How can you solve the problem?
  - Use of the LMA
    - Longest Matching Algorithm
    - We replace all entries having the same next-hop by the correspondent prefix
    - We leave routing table entries that have a different prefix unchanged
    - It is the routing table entry that has the longest matching bits with the destination address starting from the left that will be used to forward the packet

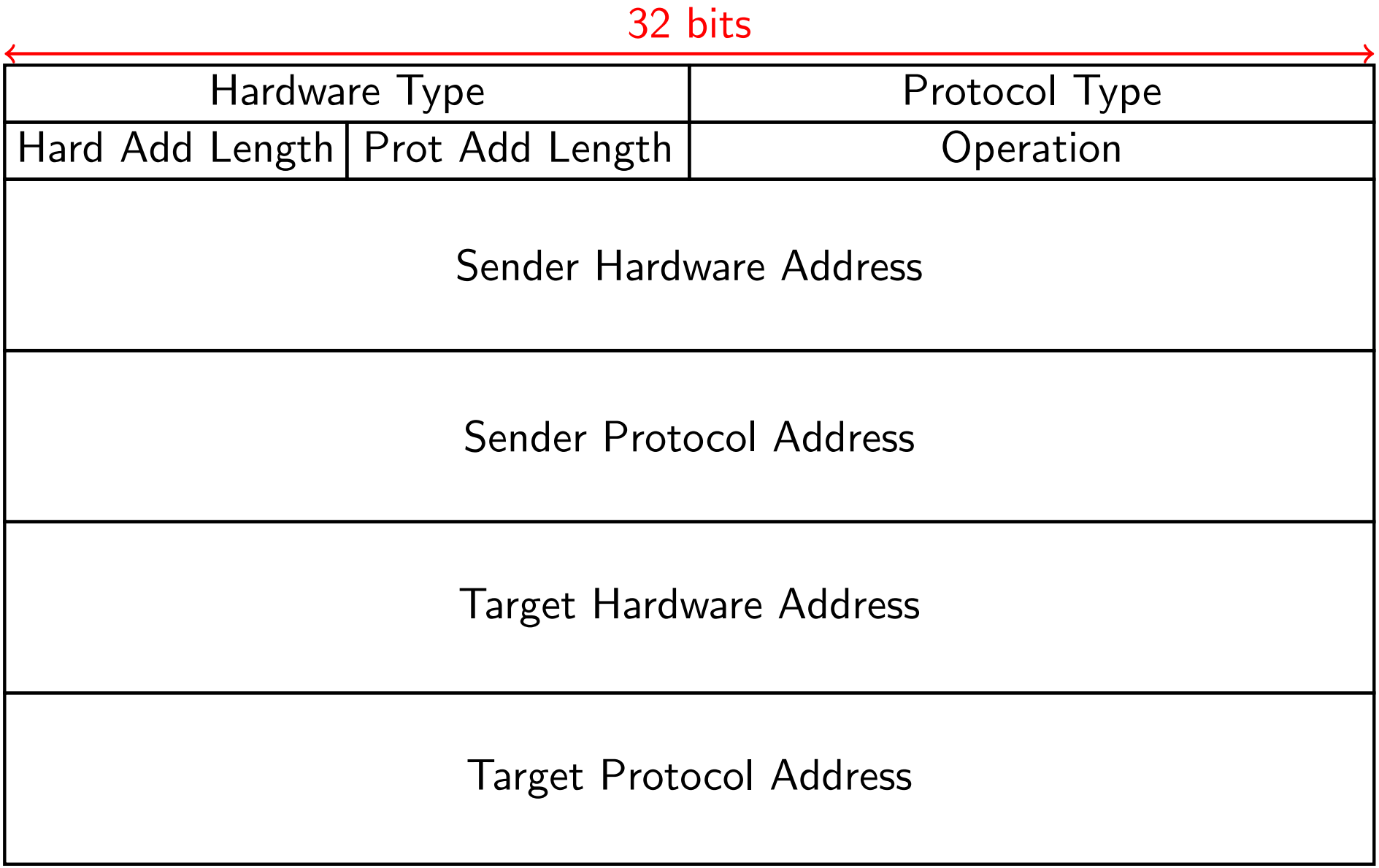
ARP

- Address Resolution Protocol
- RFC 826
  - Determines the physical address correspondent to an IP address when needed
    - MAC address of the next-hop router required by an intermediate router to forward packet towards its destination
    - MAC address of the destination required by the last router to forward packet to its destination
  - Example: over an Ethernet network, find the MAC address of the host having the IP address 131.25.64.3
    - ARP messages are encapsulated in an Ethernet frame (Type = 0x806)
  - Broadcast of an ARP Request
    - Ethernet destination address = FF:FF:FF:FF:FF:FF
  - The host having the same IP address replies with an ARP Reply message that contains its MAC address
    - ARP Reply message is sent only to the host that had sent the ARP Request

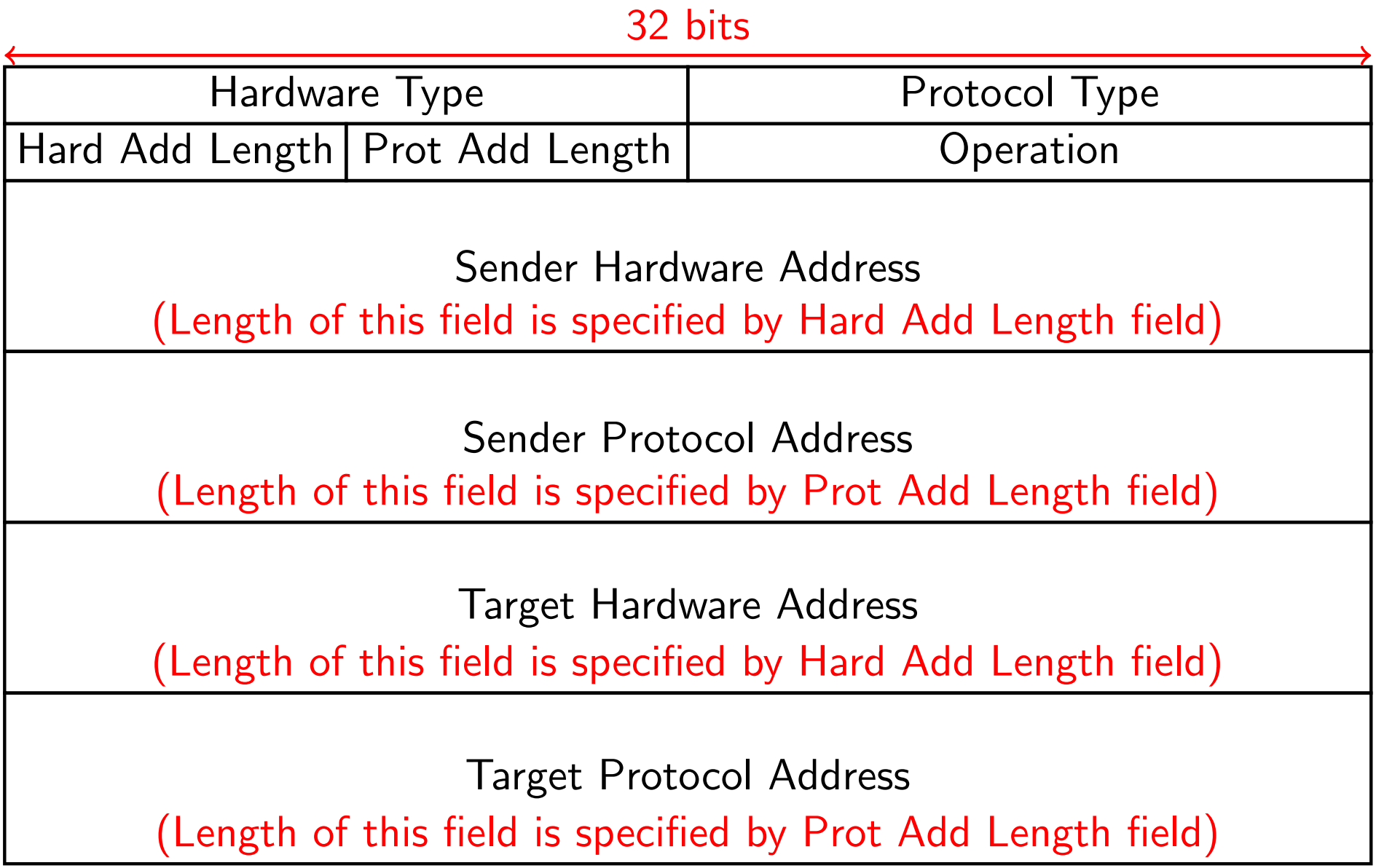
ARP

- Responses are saved in ARP table that contains correspondences between IP addresses and MAC addresses
- Each table entry has a limited time to live (10 to 20 minutes)
- If a host A wants to communicate with a host B, it looks up the MAC address of the host B in its ARP table
  - If it doesn't find it, it sends an ARP Request message over the network to get to the IP address of the host B that must reply with an ARP Reply message
- An ARP server may be used
  - The server answers all requests
  - The server must know all IP and MAC addresses of the network

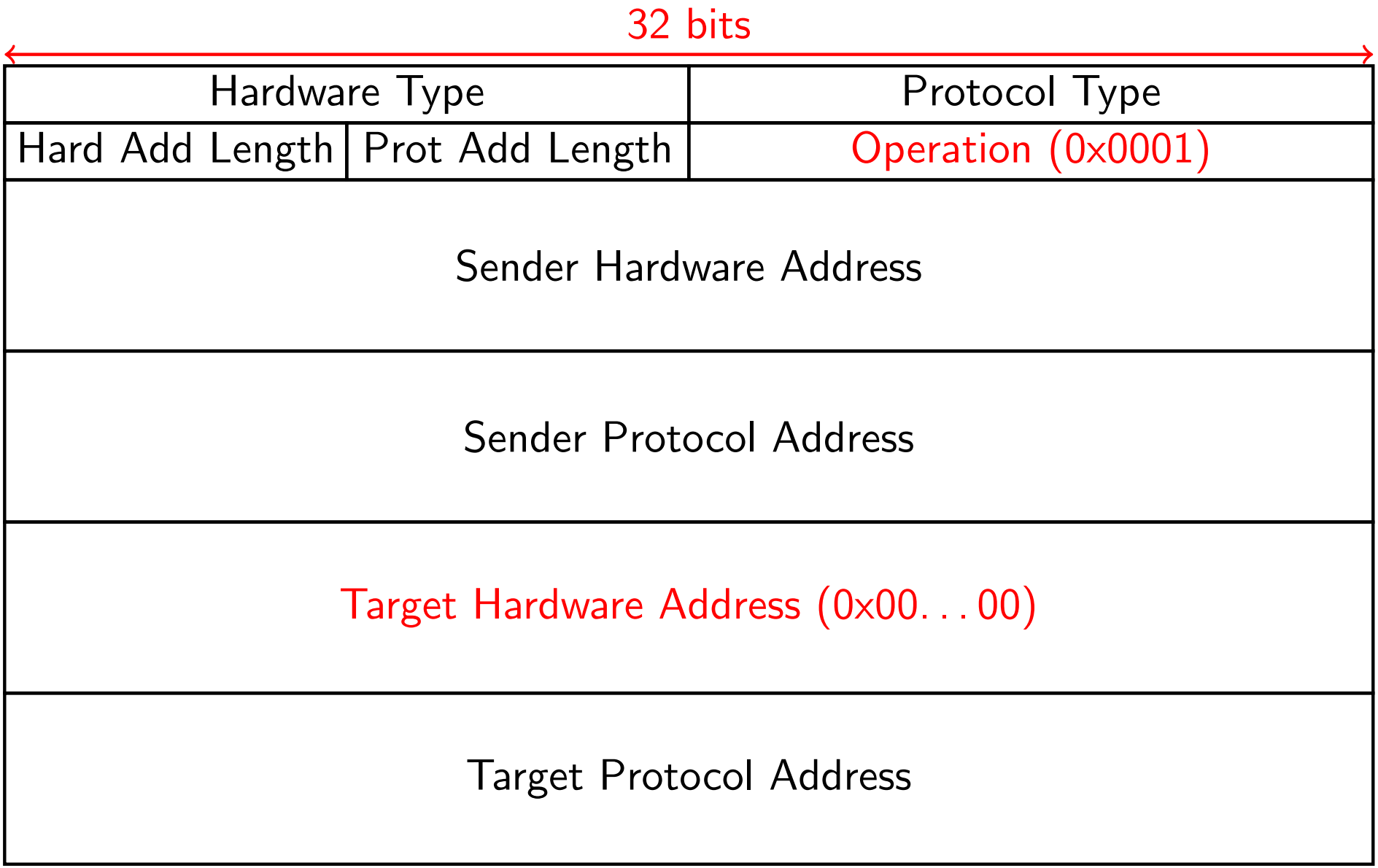
ARP Packet Format



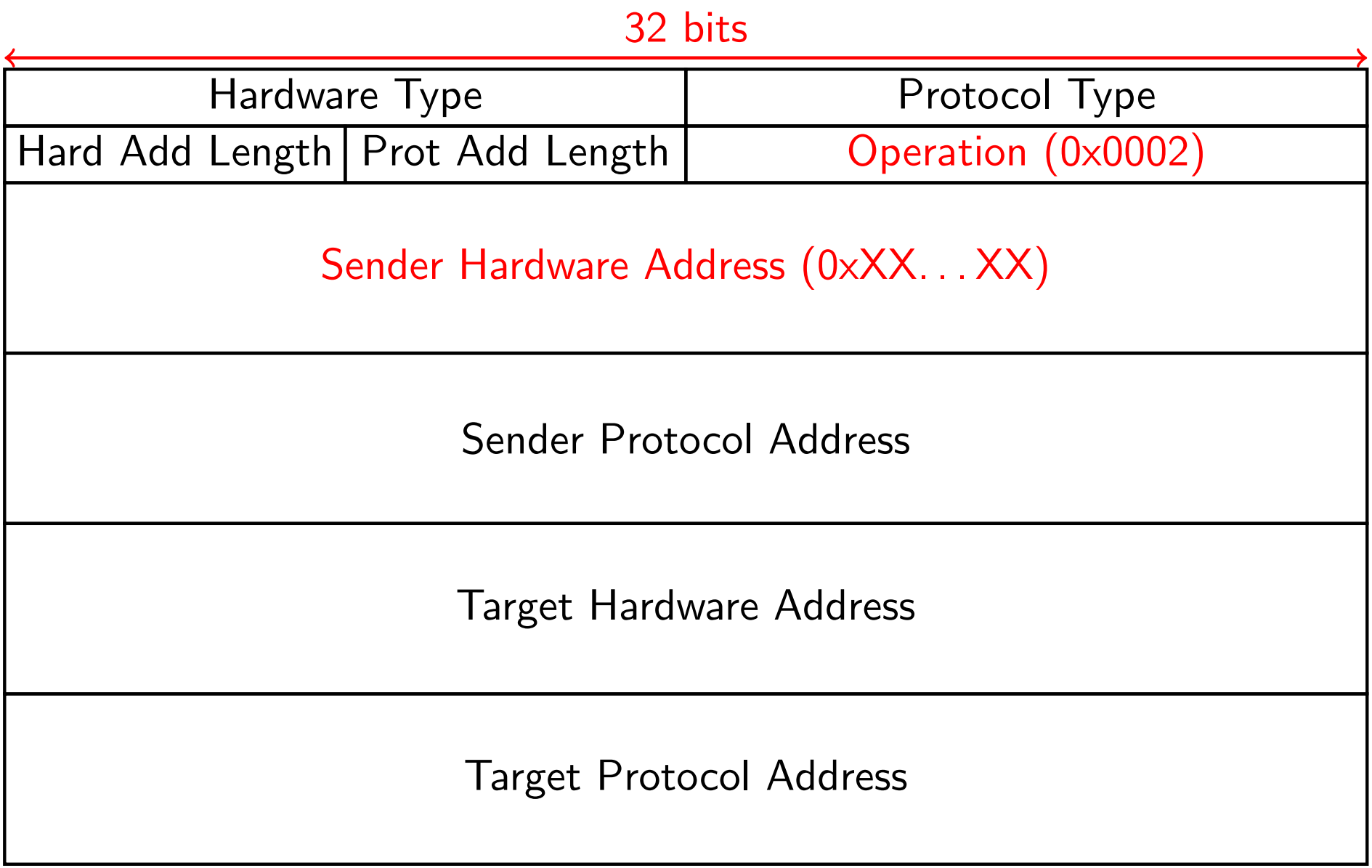
ARP Packet Format



ARP Request

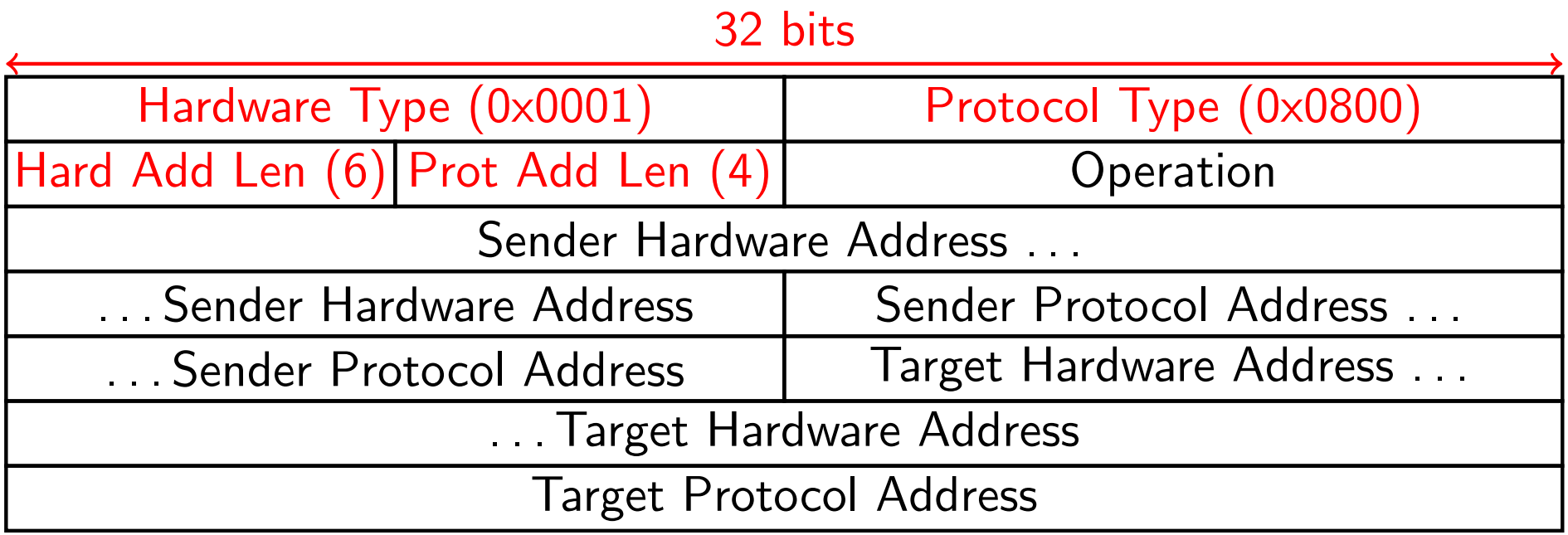

32 bits
Hardware Type
Protocol Type
Hard Add Length
Prot Add Length
Operation (0x0001)
Sender Hardware Address
Sender Protocol Address
Target Hardware Address (0x00...00)
Target Protocol Address

ARP Reply


32 bits
Hardware Type
Protocol Type
Hard Add Length
Prot Add Length
Operation (0x0002)
Sender Hardware Address (0xXX...XX)
Sender Protocol Address
Target Hardware Address
Target Protocol Address

ARP Packet Format/IP Over Ethernet

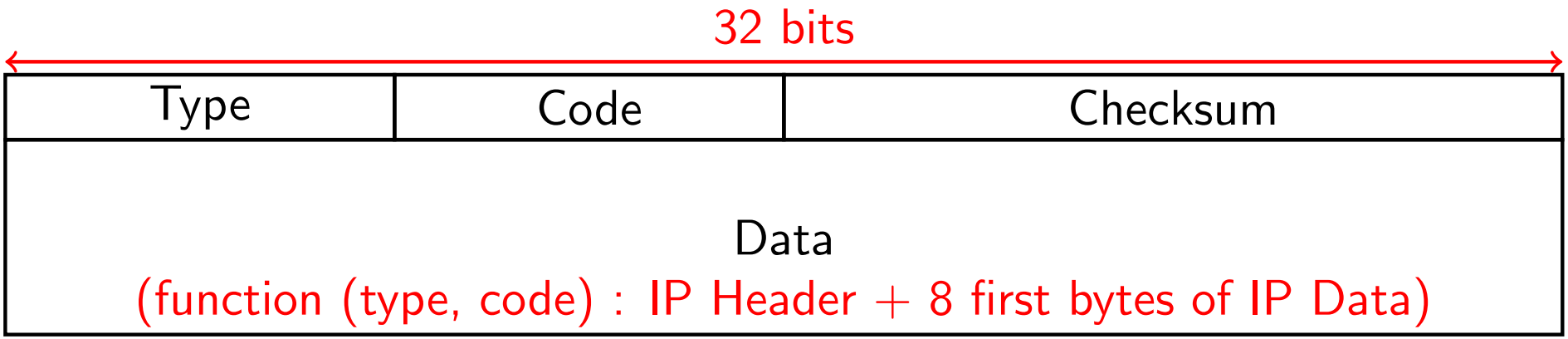
- Ethernet: Hardware Type = 0x0001
- IP: Protocol Type = 0x0800
- Hard Add Length: 6 (Length of an Ethernet Address)
- Prot Add Length: 4 (Length of an IP Address)



ICMP

- Internet Control Message Protocol
- RFC 792
- Allows the notification of errors to the source
- Encapsulated in IP
  - Protocol = 1
- Types
  - Echo Request/Echo Reply
  - Destination Unreachable
  - Redirect
  - Time Exceeded
  - ...

ICMP Packet Format



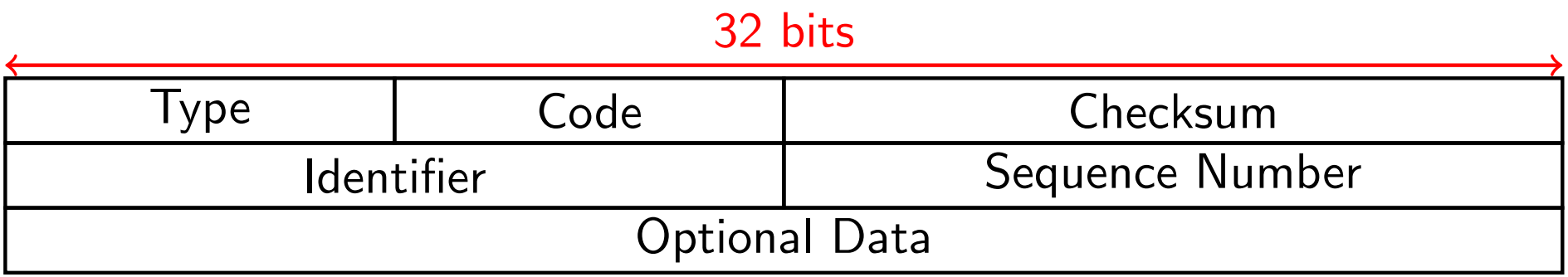
- Type: 8 bits
  - 15 different types
- Code: 8 bits
  - Sub-types for each type
- Checksum

ICMP

- Destination Unreachable
  - Network Unreachable
    - Sent by a router that cannot reach the destination network
  - Host Unreachable
    - Sent by a router on the same network as the destination host that cannot reach the destination host
  - Port Unreachable
    - Sent by the destination host when it cannot reach the destination process
  - Protocol Unreachable
    - Sent by the destination host when it cannot recognize the protocol
  - Fragmentation Needed and 'Don't Fragment' bit set
  - Source Route failed

Ping Utility

- ping "IP address"
  - Unix Command: ping 195.16.84.12
  - Checks the operation of a distant host
  - Checks if the distant host is reachable
  - ICMP Echo Request/ Echo Reply
  - Sequence Number field allows the determination of the number of lost packets
  - Identifier field allows the parallel execution of many ping programs between two hosts



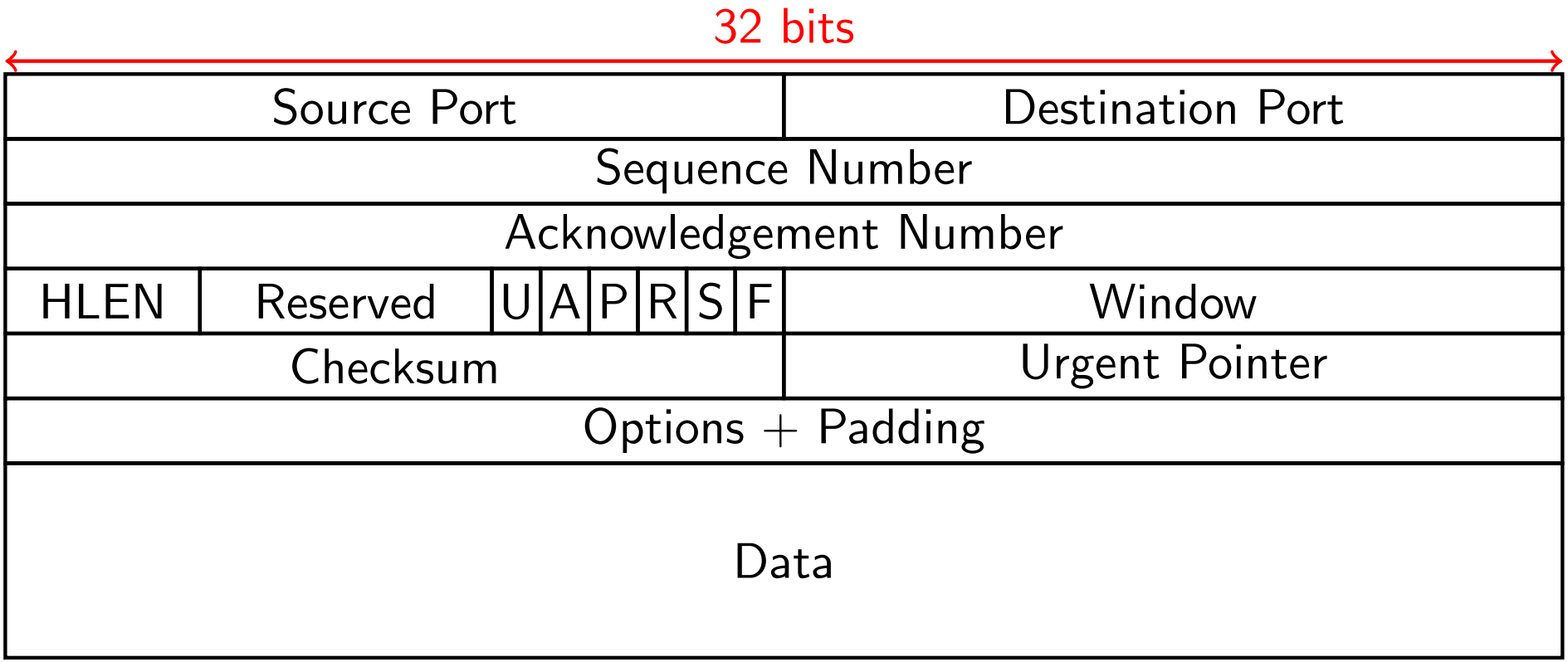
Traceroute Utility

- traceroute "IP address"
  - Unix Command: traceroute 195.16.84.12
  - Determines the whole path followed by packets to reach a particular destination
- Algorithm
  - The source sends an IP packet (UDP packet) to the destination address with a TTL 1
  - The packet will be dropped by the first router on the path towards the destination
  - The router that dropped the packet sends an ICMP Time Exceeded message to the source which uses it to determine the first router on the path towards the destination
  - The source sends a second IP packet to the destination address with a TTL 2. The packet will be dropped by the second router on the path to the destination. This royter will send an ICMP Time Exceeded message to the source that uses it to determine the second router on the path
  - The source repeats the same procedure by incrementing the TTL until it receives an ICMP Error Message (ICMP Port Unreachable Message) from the destination.

TCP

- Transport Control Protocol
- RFC 793
- Encapsulated in IP
  - Protocol = 6
- Connection Oriented Service
  - Reliable (Error, Loss, and Duplicates Management)
  - In Order Delivery
  - Full-Duplex
- Multiplexing
  - Many applications on the same host may communicate at the same time
- T-PDU: segment

TCP


32 bits
Source Port
Destination Port
Sequence Number
Acknowledgement Number
HLEN
Reserved
U
A
P
R
S
F
Window
Checksum
Urgent Pointer
Options + Padding
Data

TCP Segment Format

- Source Port
  - 16 bits
  - Port source
  - Indicates the port number on which the sender is listening
- Destination Port
  - 16 bits
  - Port destination
  - Indicates the port number on which the destination is listening
- The fields (Source Address, Destination Address, Protocol, Source Port, Destination Port) identify in a unique manner each connection

TCP Segment Format

- Sequence Number
  - 32 bits
  - Indicates the sequence number of the first data byte
- Acknowledgement Number
  - 32 bits
  - Indicates the number of the next data byte that the sender is ready to receive from the other side of the connection
  - Acknowledgement of all previous bytes

TCP Segment Format

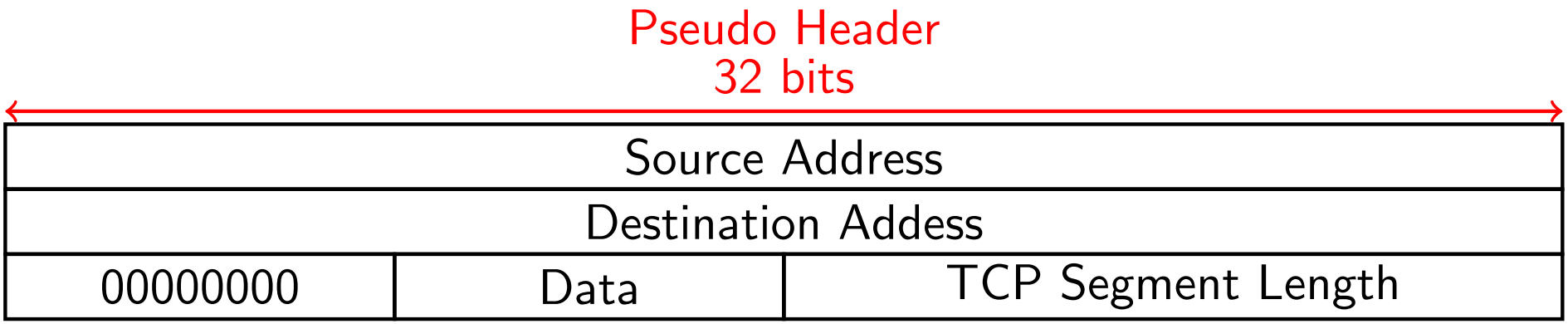
- HLen
  - 4 bits
  - Header Length of the TCP segment header in 32 bit words
  - Minimal length is 5 (No options)
- TCP Flags
  - URG: Urgent, indicates the presence of urgent data in the segment
  - ACK: Acknowledgement, indicates that the segment is an acknowledgement segment
  - PSH: Push, when it is set, data must be delivered to the higher layer immediately
  - RST: Reset, reset the TCP connection
  - SYN: Synchronize, indicates a connection setup segment
  - FIN: Fin, indicates a connection release segment

TCP Segment Format

- Window
  - 16 bits
  - TCP Window
  - Indicates the number of bytes that the receiver is ready to receive
  - Flow Control

TCP Segment Format

- Checksum
  - 16 bits
  - Error control over the whole TCP segment (header + data) + a pseudo-header
    - Violation of layering concept
  - A zero byte is added to the end of the segment if the size of the segment is odd
  - Same algorithm as with IP Checksum



TCP Segment Format

- Urgent Pointer
  - 16 bits
  - Pointer to the urgent Data
  - When the flag URG is set, this field contains a pointer to the urgent data

TCP Segment Format

- Source port and destination port allows multiplexing
- Two different TCP connections have different (source port, destination port) pairs
- Server listens passively over a well known port waiting for connection requests from clients
  - Telnet Server: 23
  - Web Server: 80
  - FTP: 20 and 21
    - 20 for commands
    - 21 for data transfer

TCP Connection Setup

- Server listens passively over a particular port number
  - Server uses a well known port
    - Telnet: 23
    - FTP: 20 for commands and 21 for data
    - HTTP: 80
- Connection is established after the exchange of 3 segments between the client and the server
  - 3-way Handshake
    - Connection setup segments are retransmitted after a timer expiration if no acknowledgement is received (the timer is relative to a TCP connection setup which is different from the timer used for data retransmission)

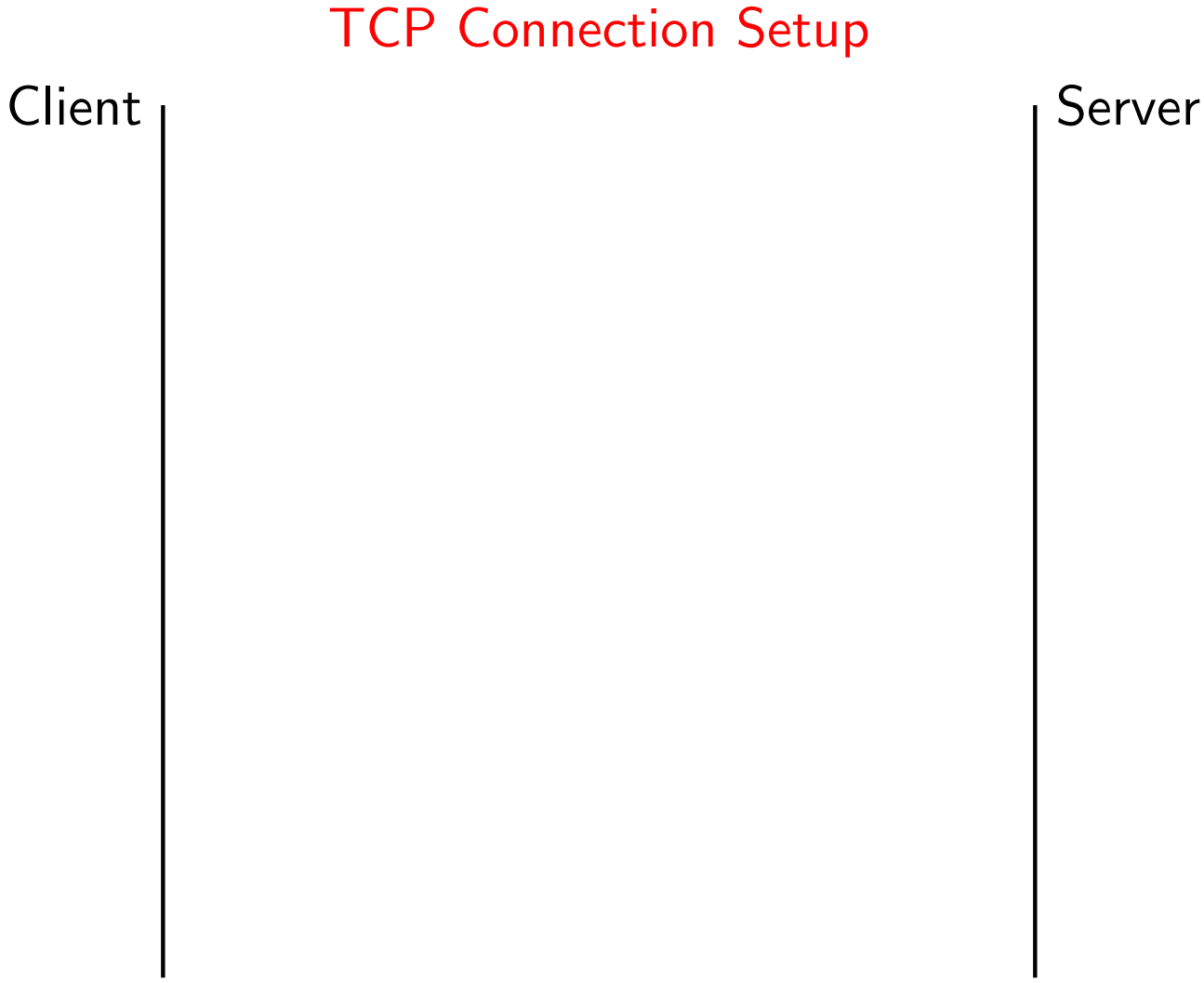
TCP Connection Setup

- The client sends a TCP segment
  - The operating system allocates a free source port to the client
  - The SYN flag of the segment is set to 1
  - Destination port contains the port number on which the server is listening (Telnet: port 23)
  - The first sequence number X is randomly selected (security reasons)
- The server replies by a TCP segment
  - SYN and ACK flags are set to 1
  - The first sequence number Y is randomly selected
  - The acknowledgement number contains the value X+1 (i.e. the server is waiting for the byte having the sequence number X+1 from the client)
  - Destination port is equal to the source port of the segment received from the client

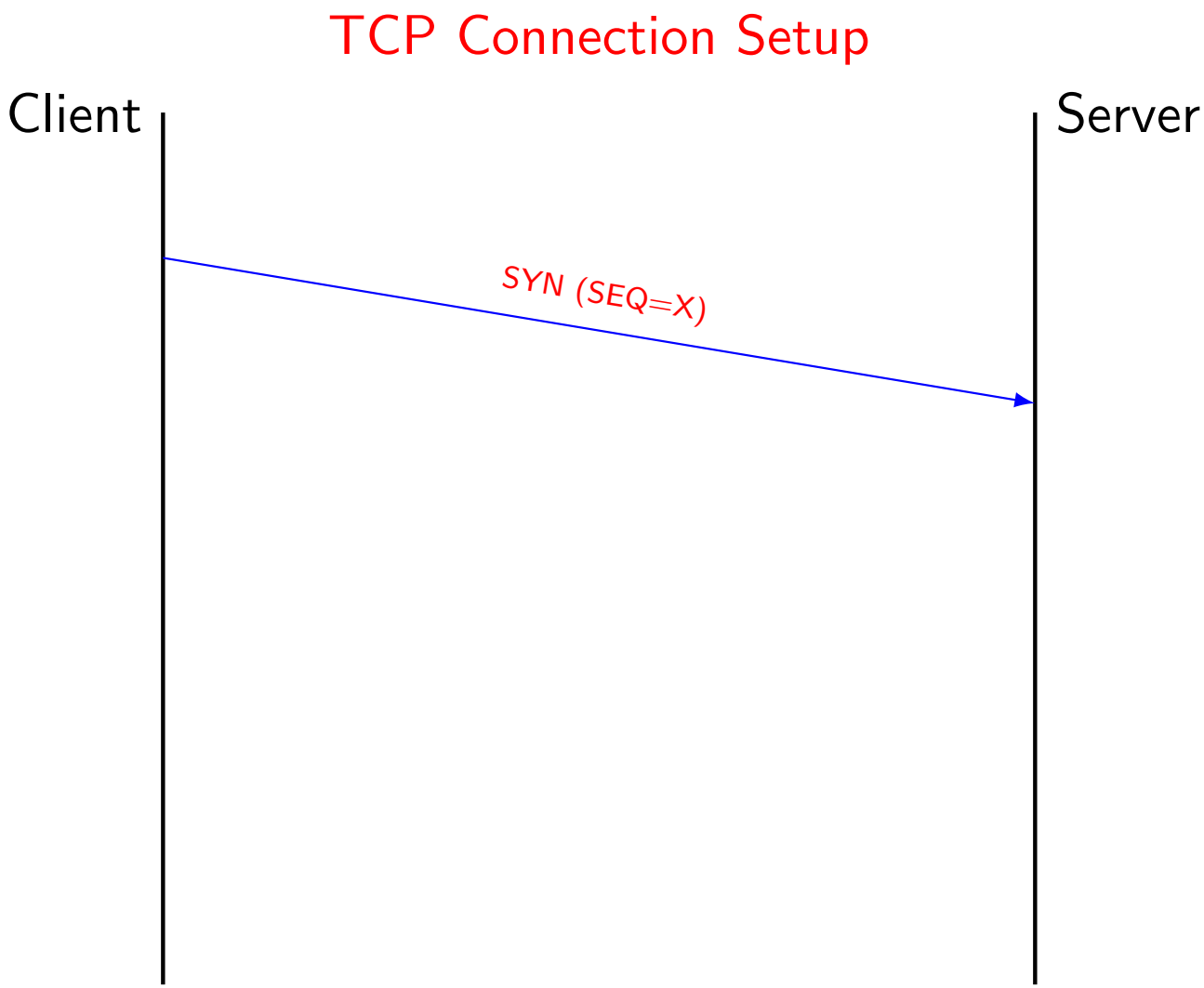
TCP Connection Setup

- The client replies by a TCP segment
  - The ACK flag is set to 1
  - The sequence number is set to X+1
  - The acknowledgement number is set to Y+1
- The sequence number is randomly selected for each connection
  - To avoid confusion with previous connections
  - For security reasons and to prevent against some attacks

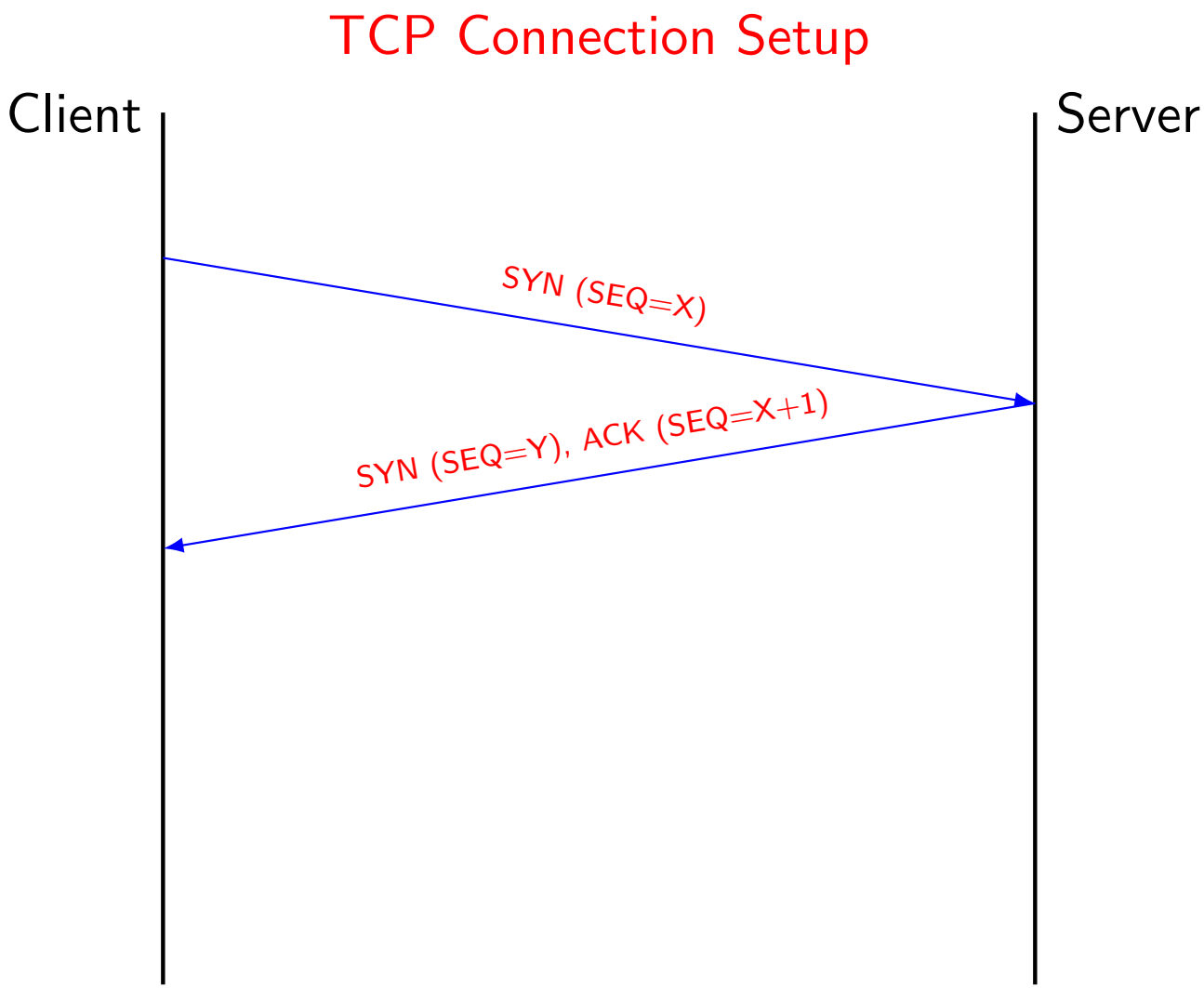
TCP Connection Setup



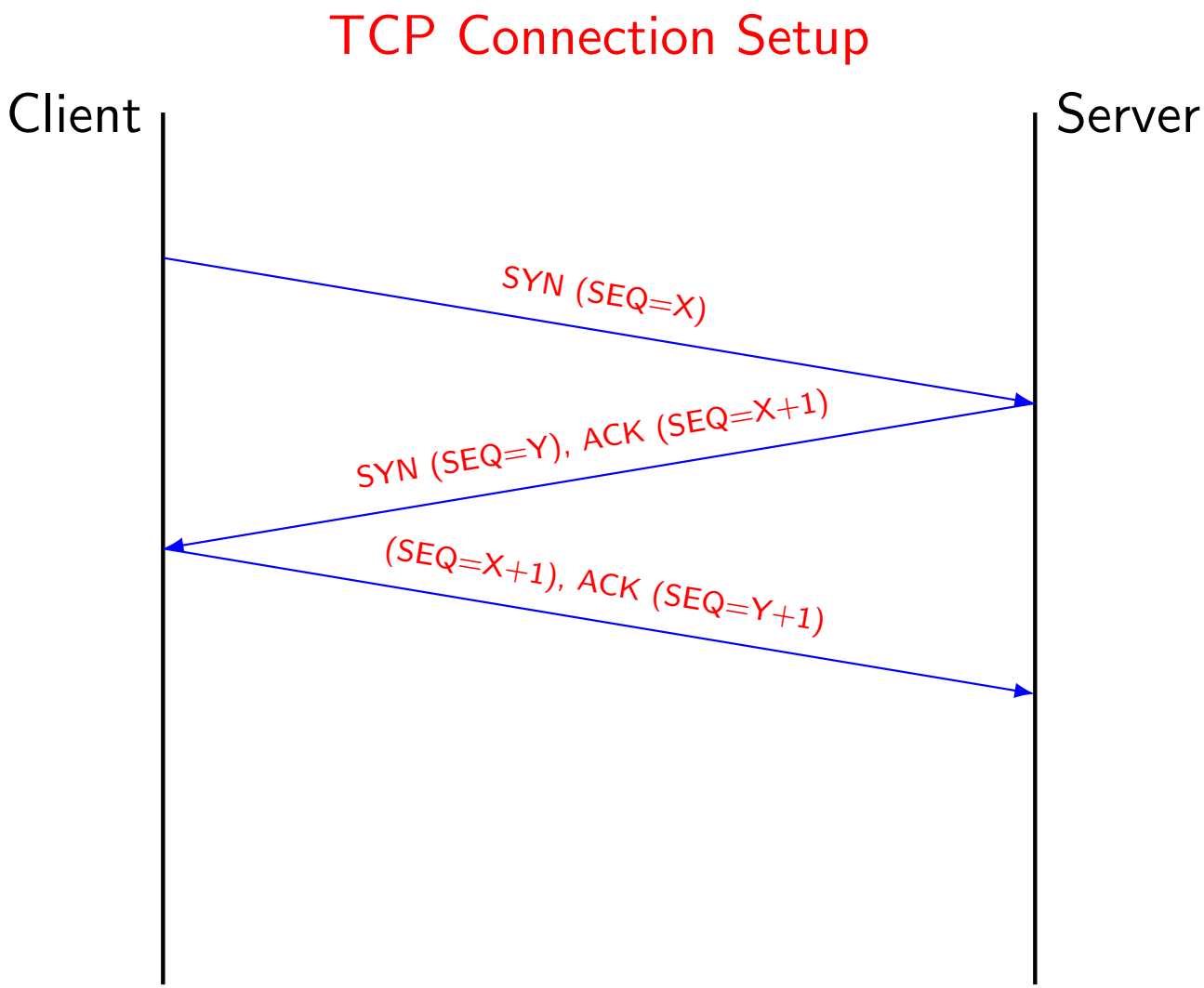
TCP Connection Setup



TCP Connection Setup



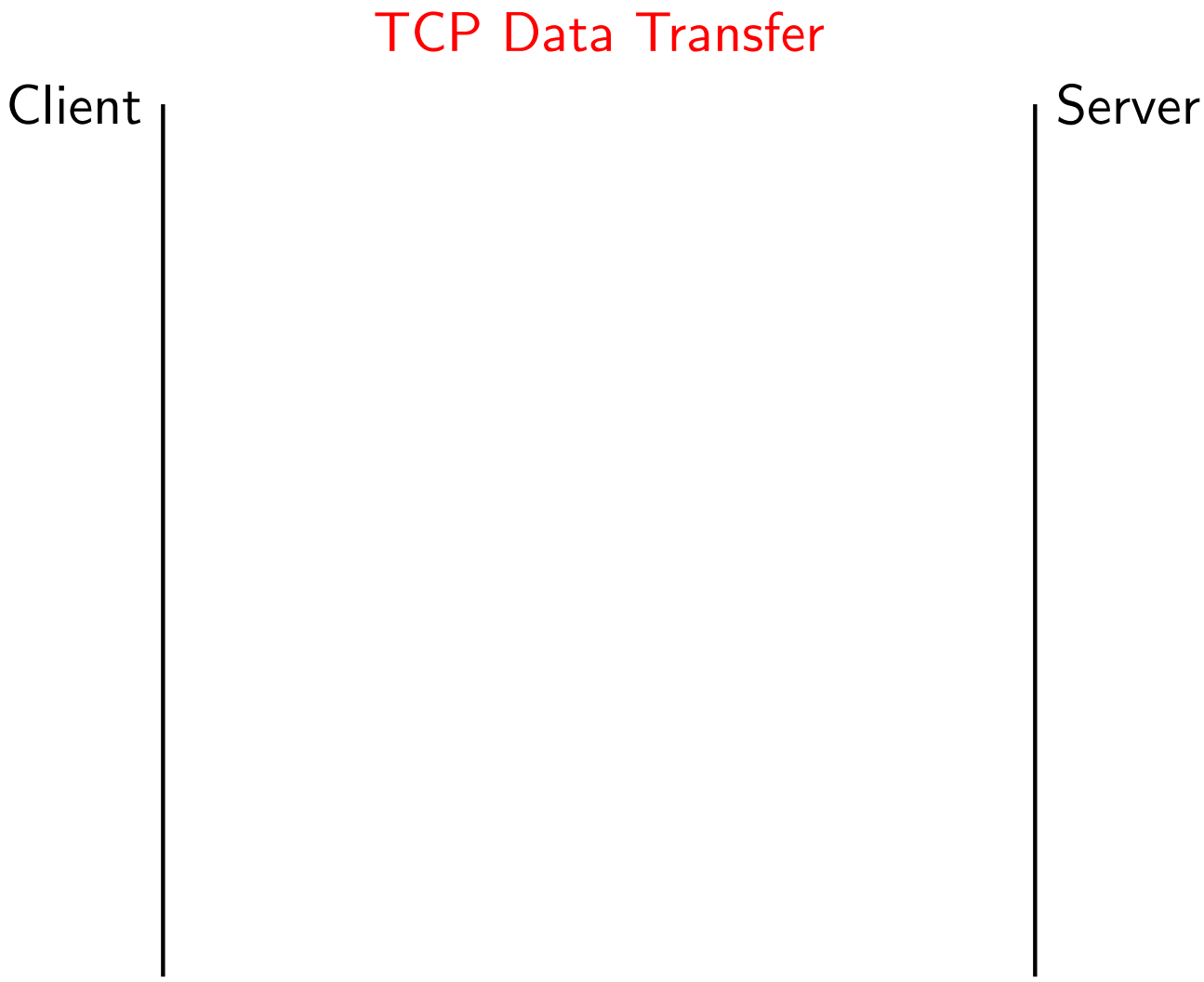
TCP Connection Setup



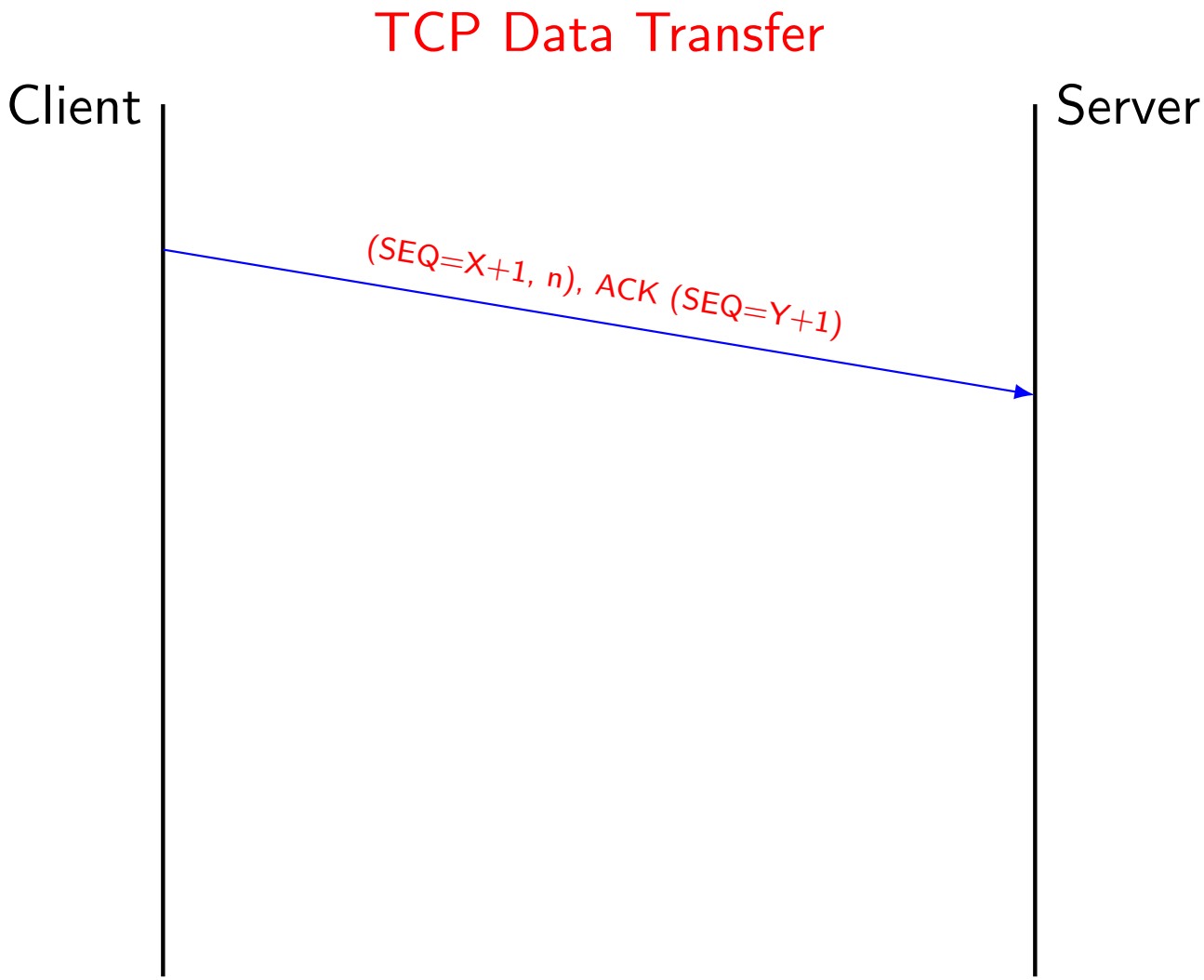
Data Transfer

- After the connection setup, the two parts may start data exchange
- The connection is full-duplex
- Each part must not send more than what allowed by the flow control window
- If no acknowledgement is received before the retransmission timer, the sender retransmits the same segment
- Acknowledgements may be sent with data
  - Piggybacking

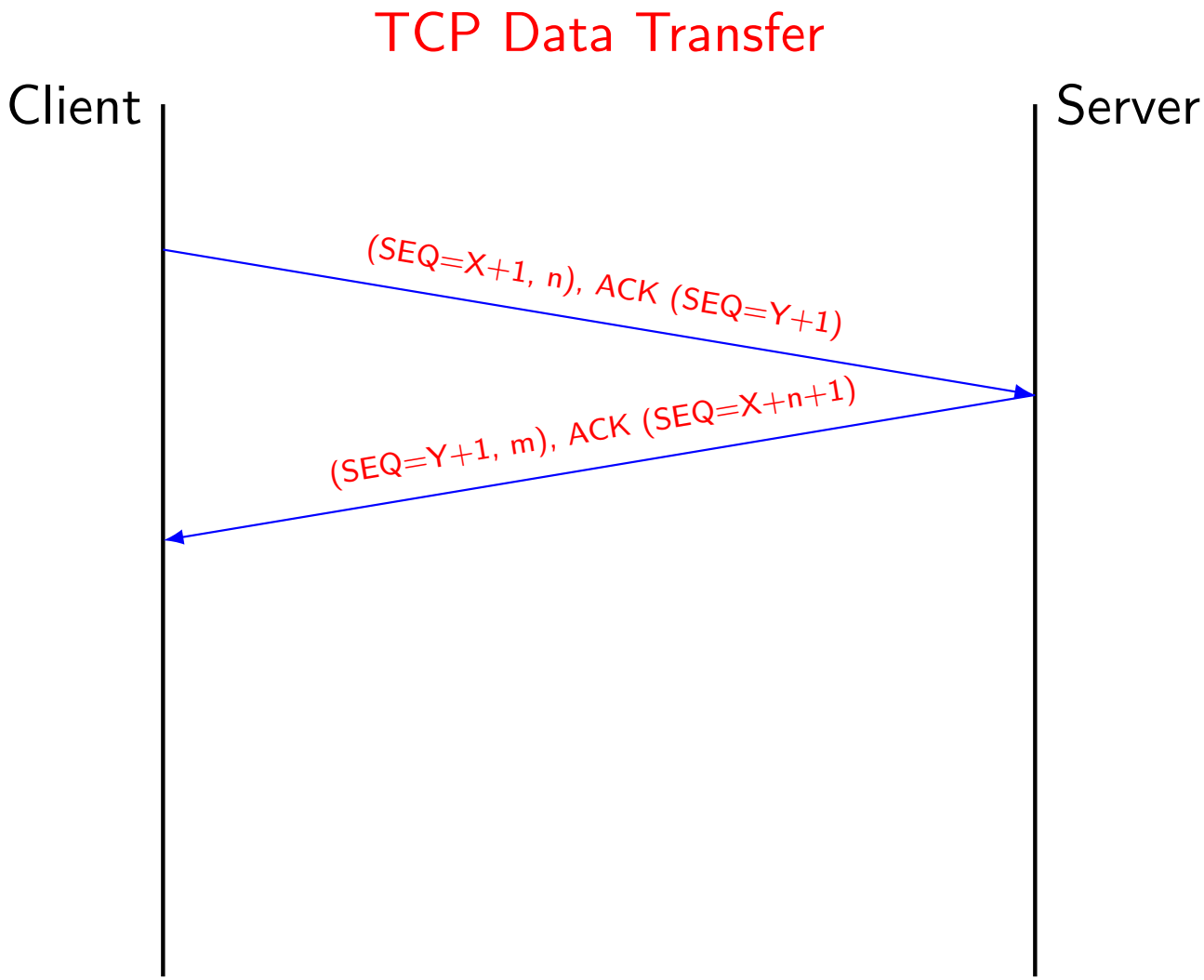
Data Transfer



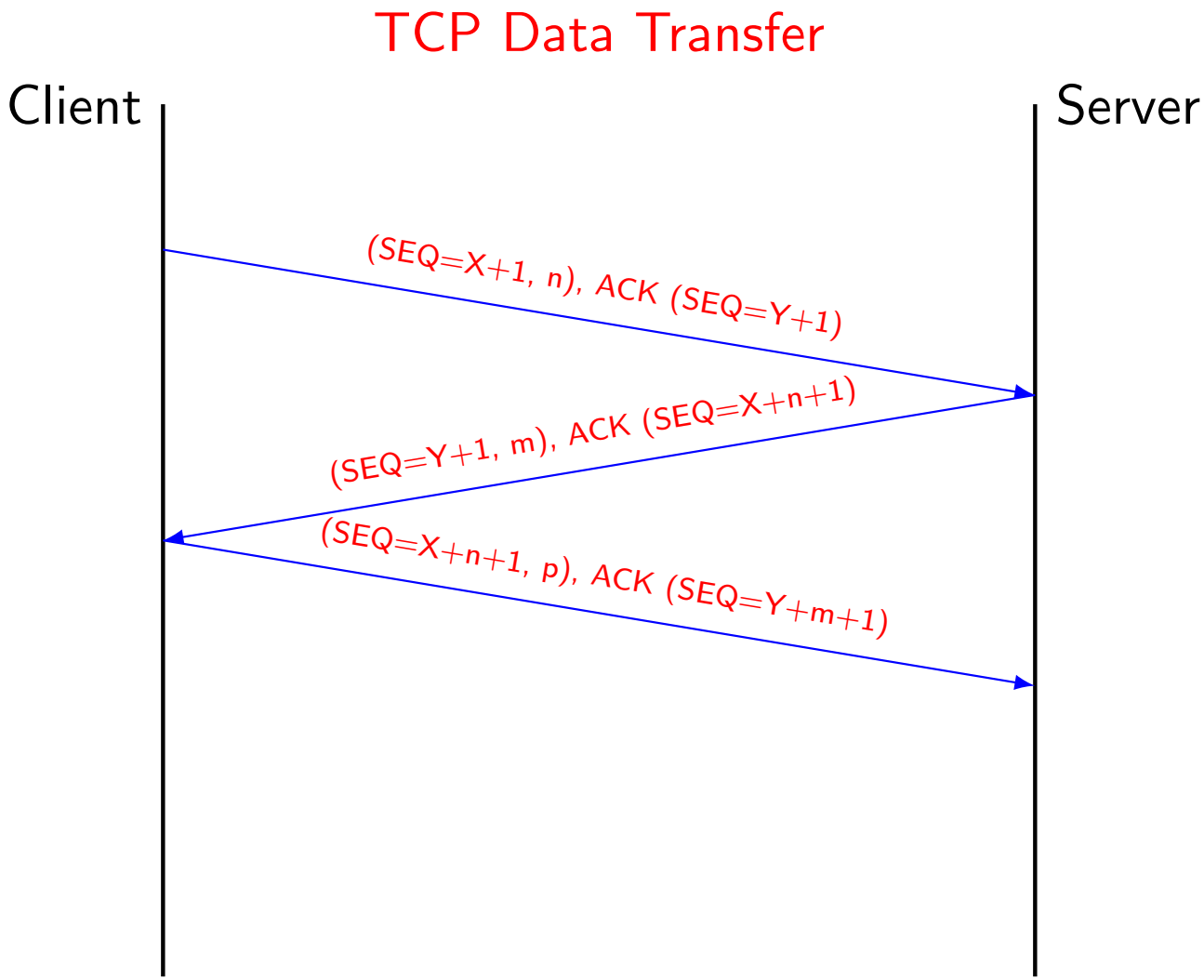
Data Transfer



Data Transfer



Data Transfer



TCP Connection Release

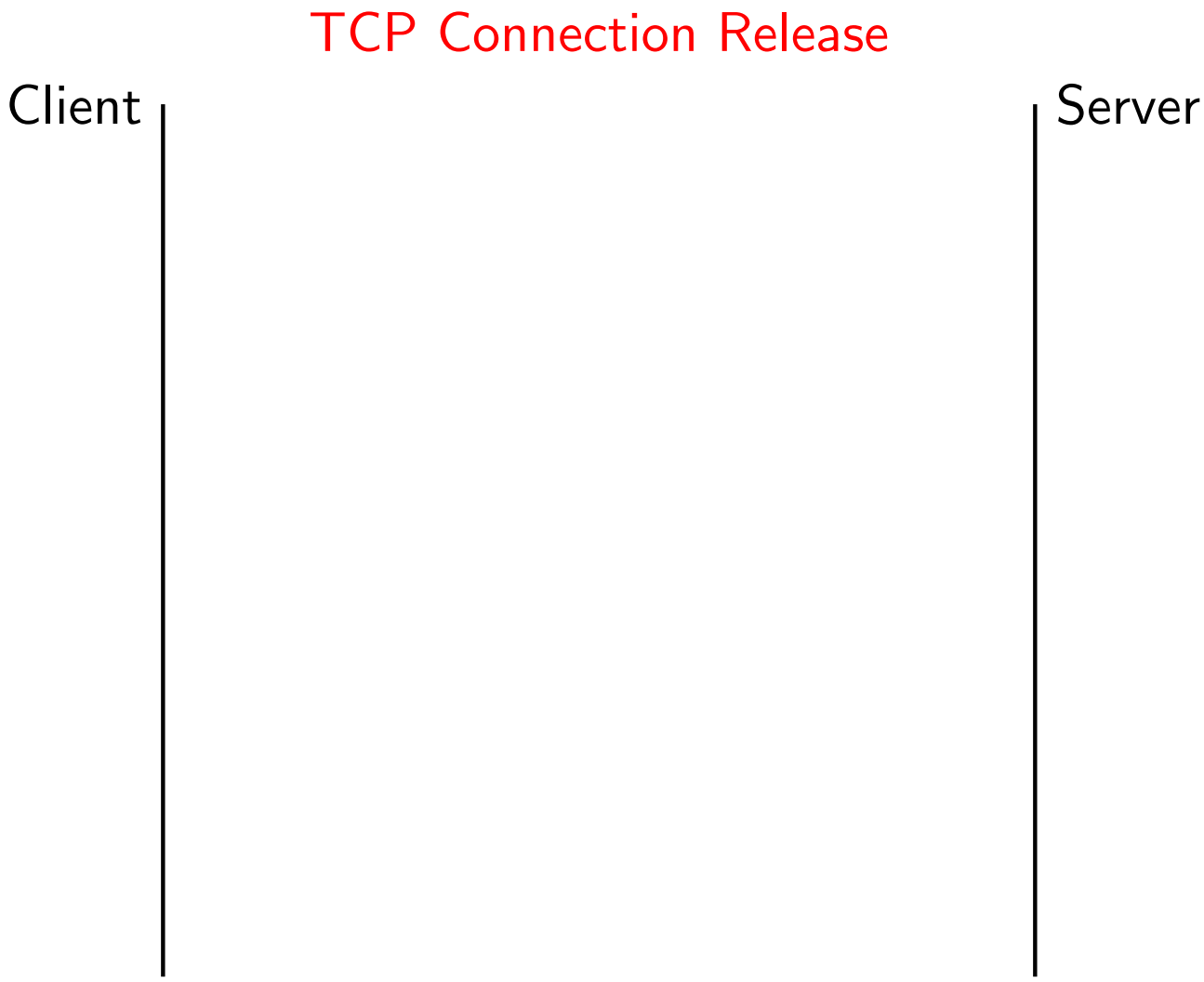
- Symmetric Release
  - Full-Duplex Connection
    - Two distinct (separate) unidirectional connections
    - Each process release its connection when it has no data to send

    1. The process that has no data to send sends a TCP segment with the FIN flag set to 1
    2. The other process acknowledges with a TCP segment having the flag ACK set to 1
    3. The first process may always receive data sent by the other process
    4. When the second process has no data to send, it sends a TCP segment with the flag FIN set to 1
    5. The first process must acknowledges by a TCP segment with the flag ACK set to 1

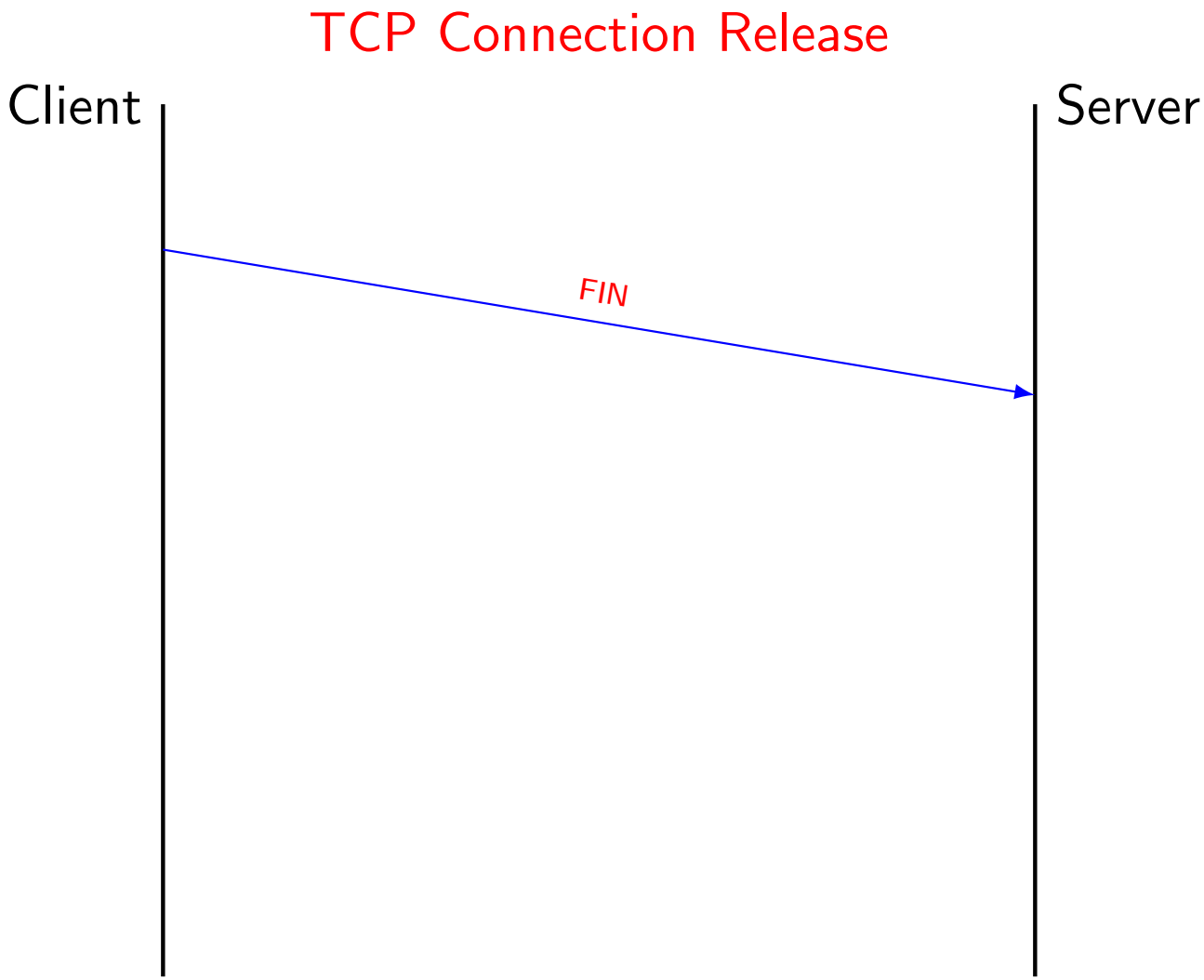
TCP Connection Release

- TCP connection release is done in 4 steps
  - 1 - A TCP segment TCP with the flag FIN set to 1
  - 2 - A TCP segment in the other direction with the flag ACK set to 1
  - 3 - A TCP segment with the flag FIN set to 1
  - 4 - A TCP segment with the flag ACK set to 1
- The connection release may be done in 3 steps if the second process has finished the transmission of its data by sending a TCP segment with FIN and ACK flags set to 1

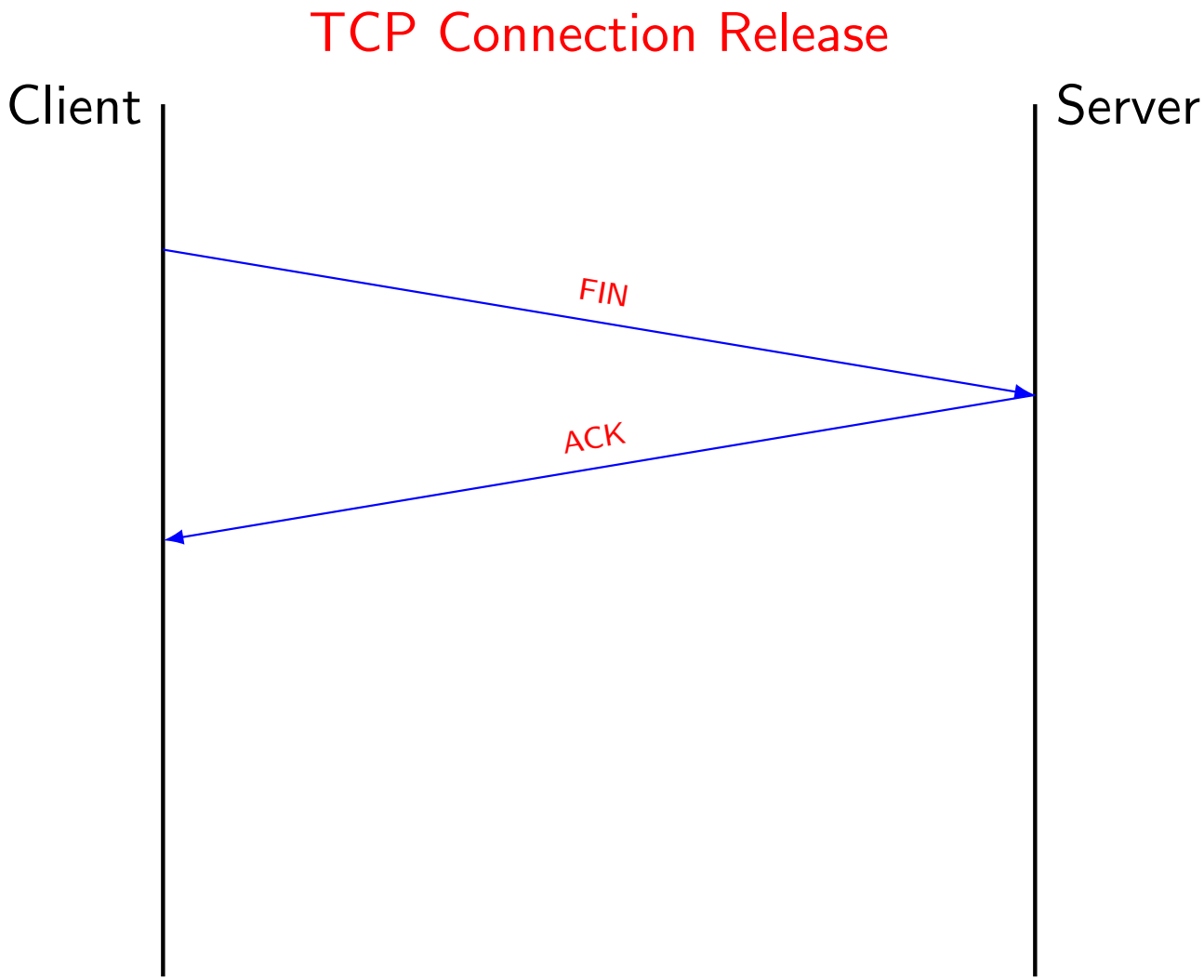
TCP Connection Release



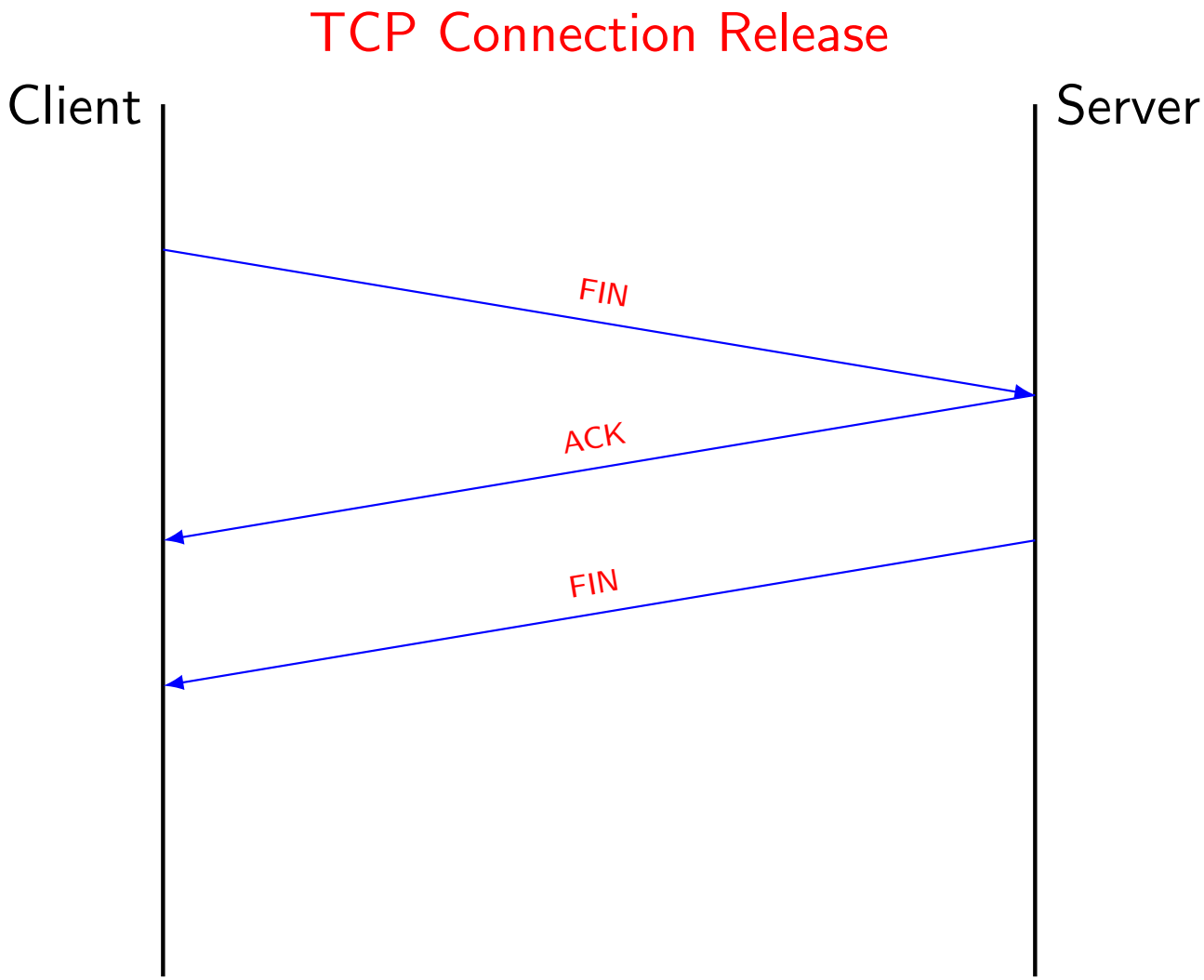
TCP Connection Release



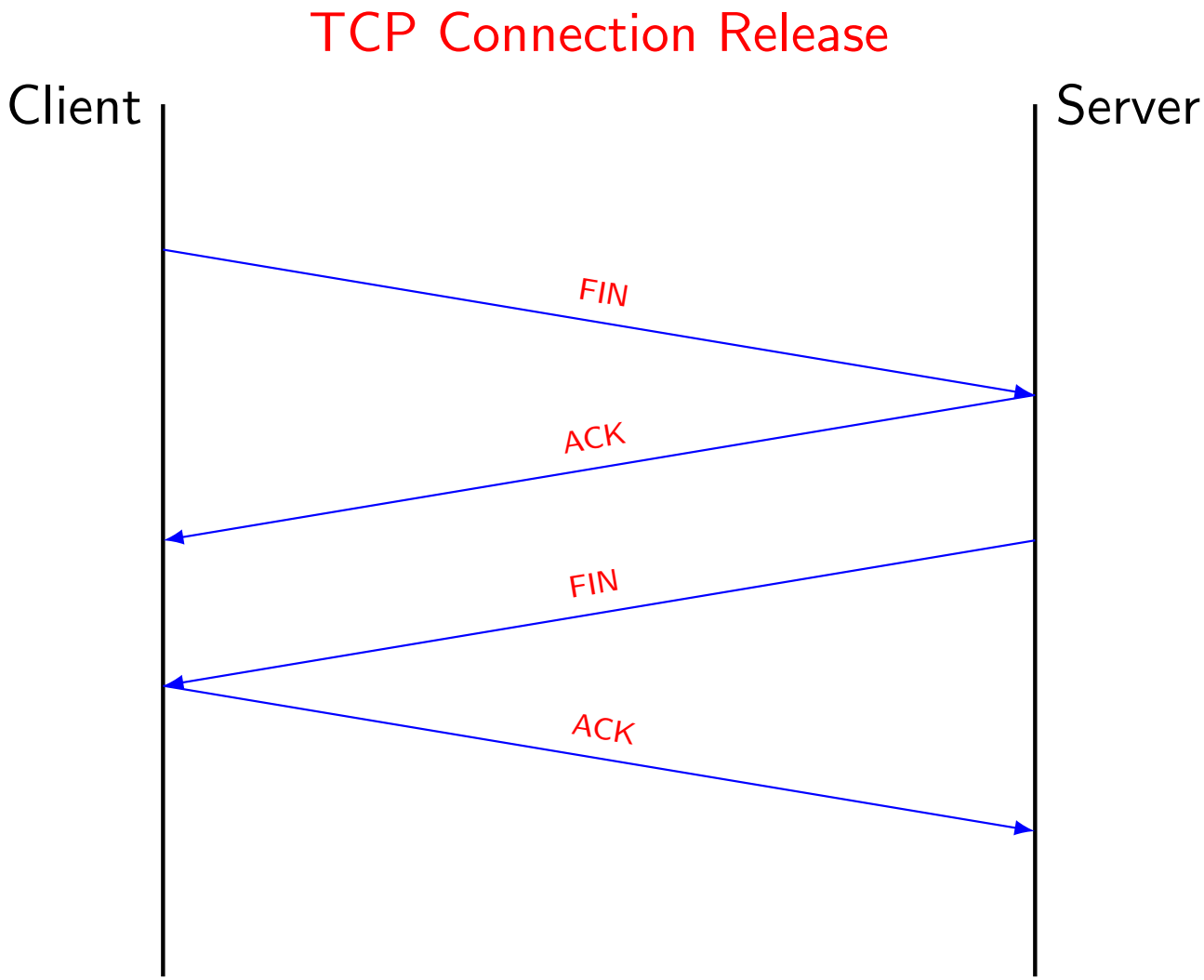
TCP Connection Release



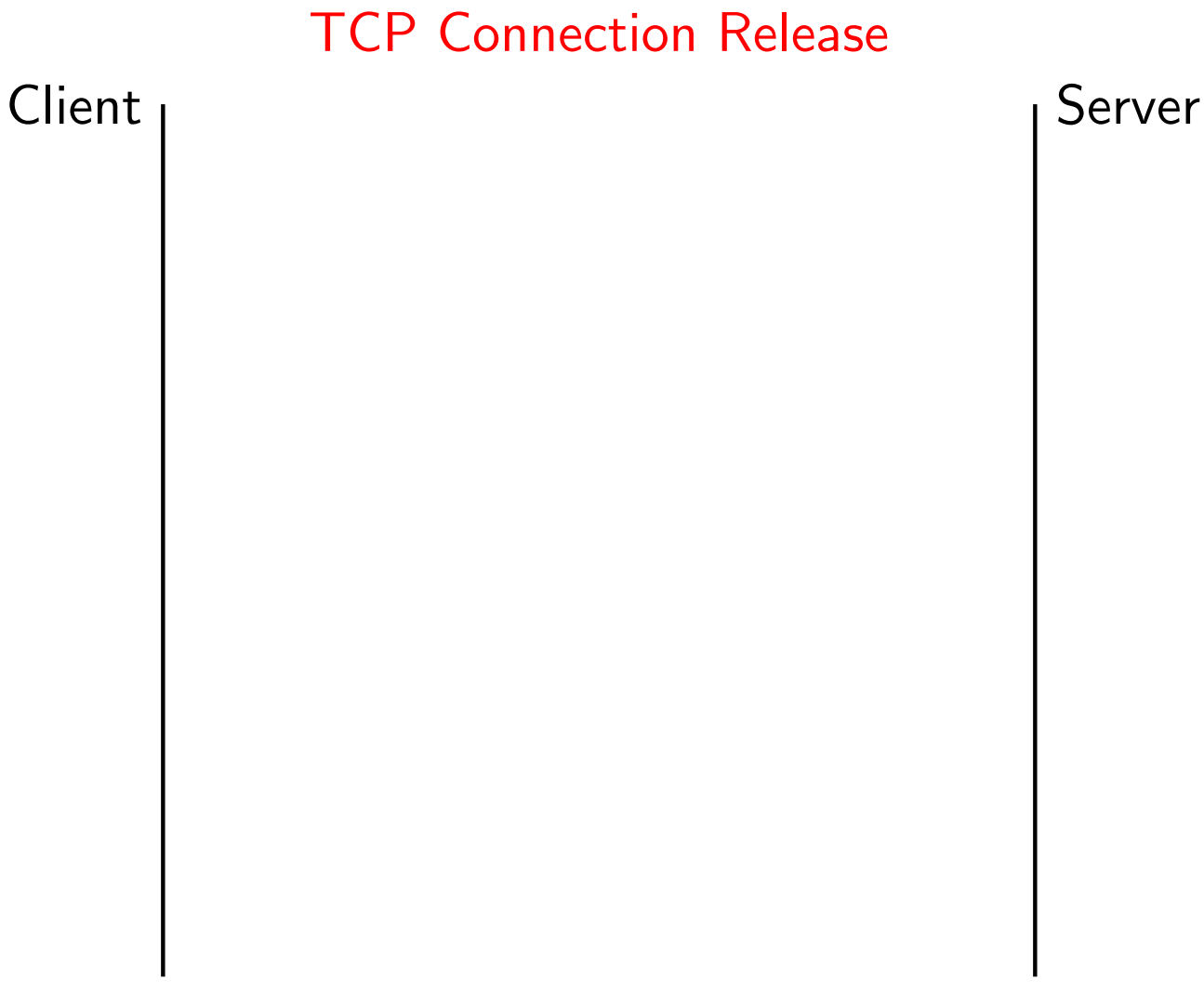
TCP Connection Release



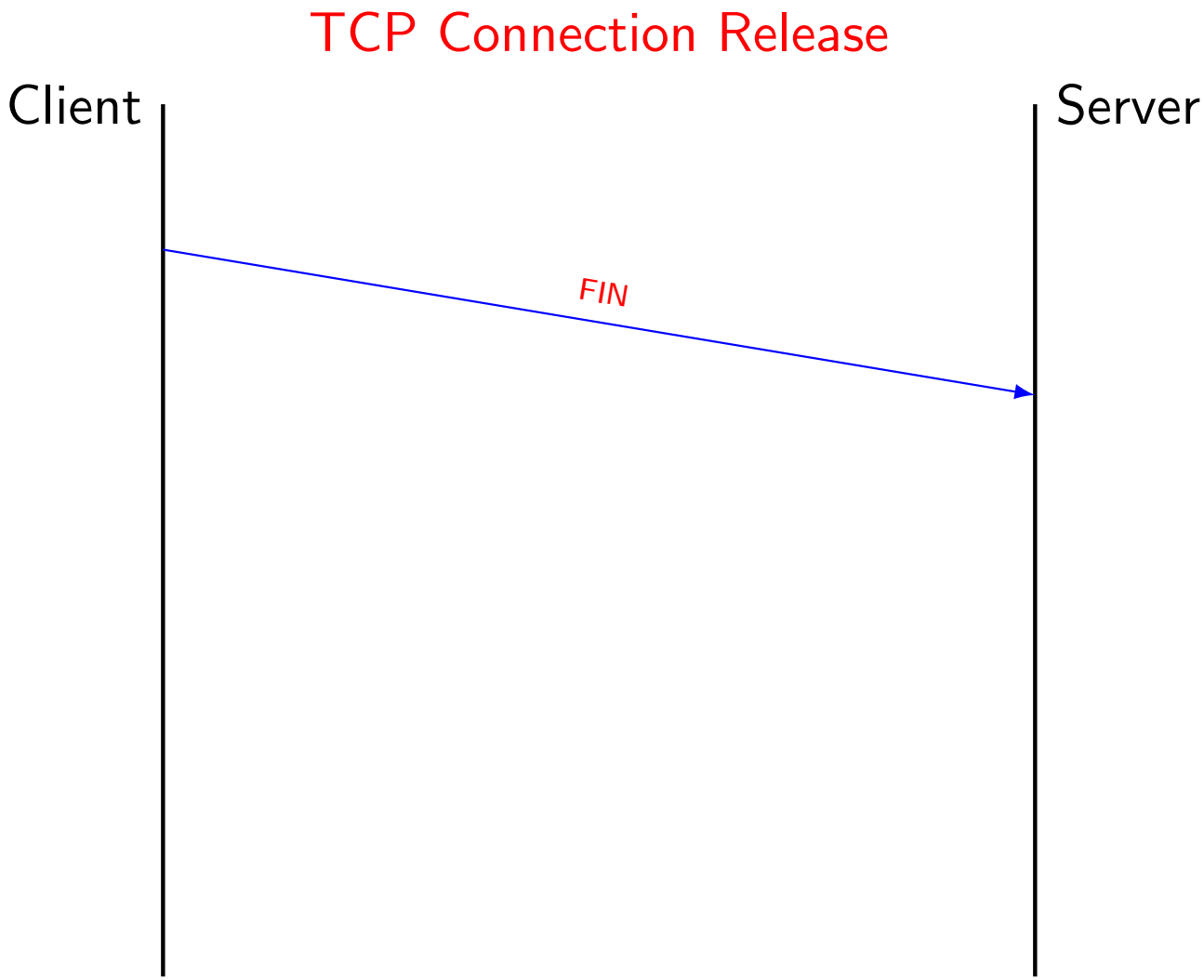
TCP Connection Release



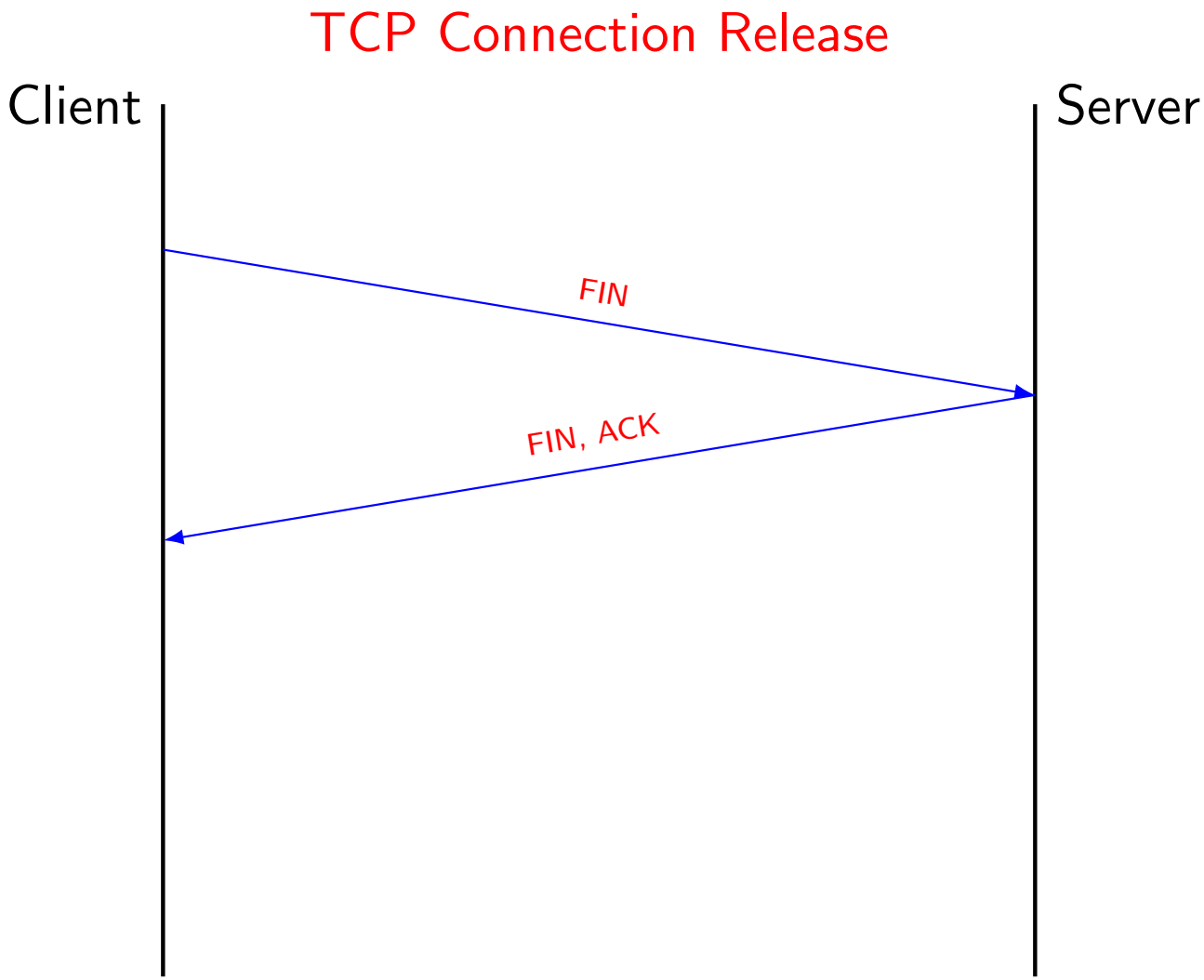
TCP Connection Release



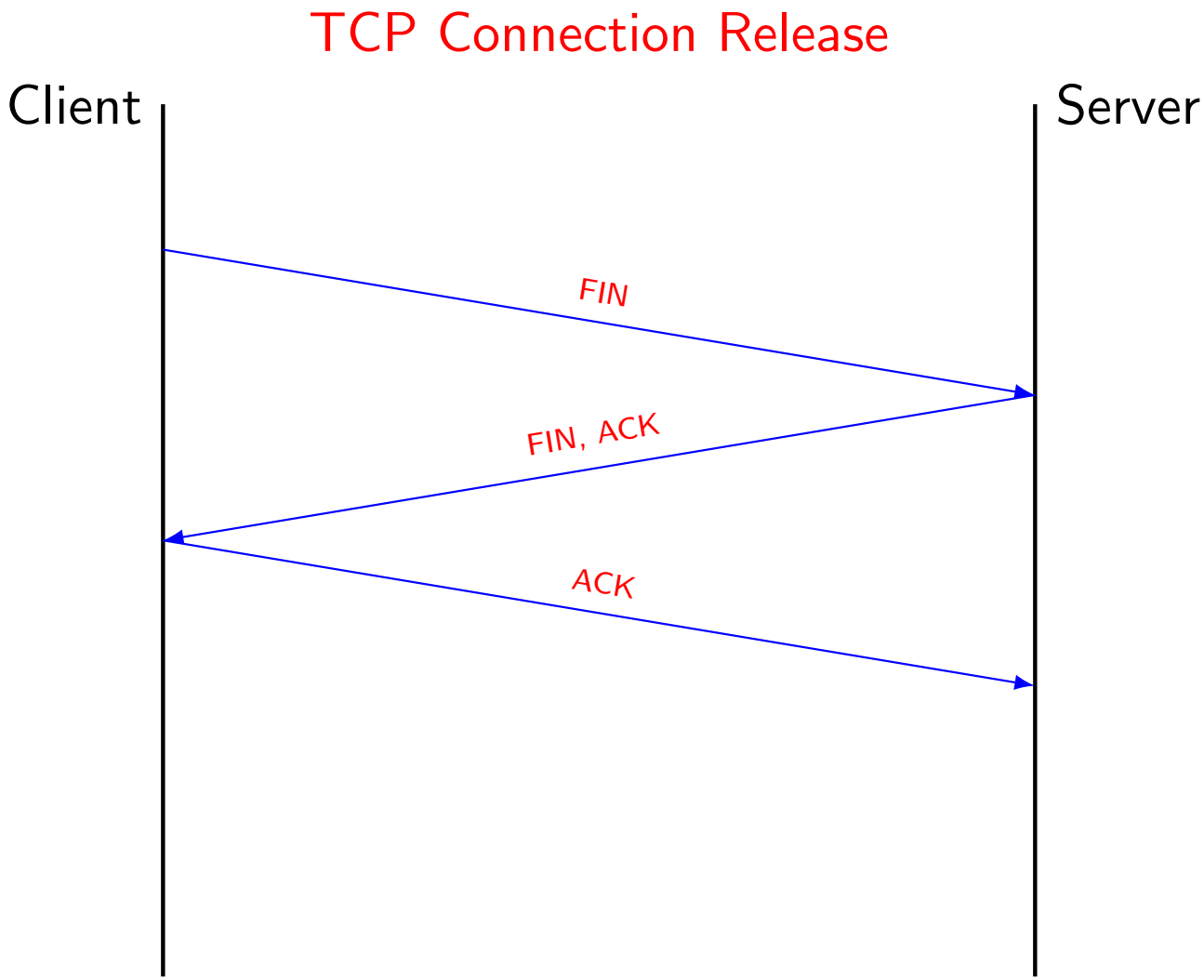
TCP Connection Release



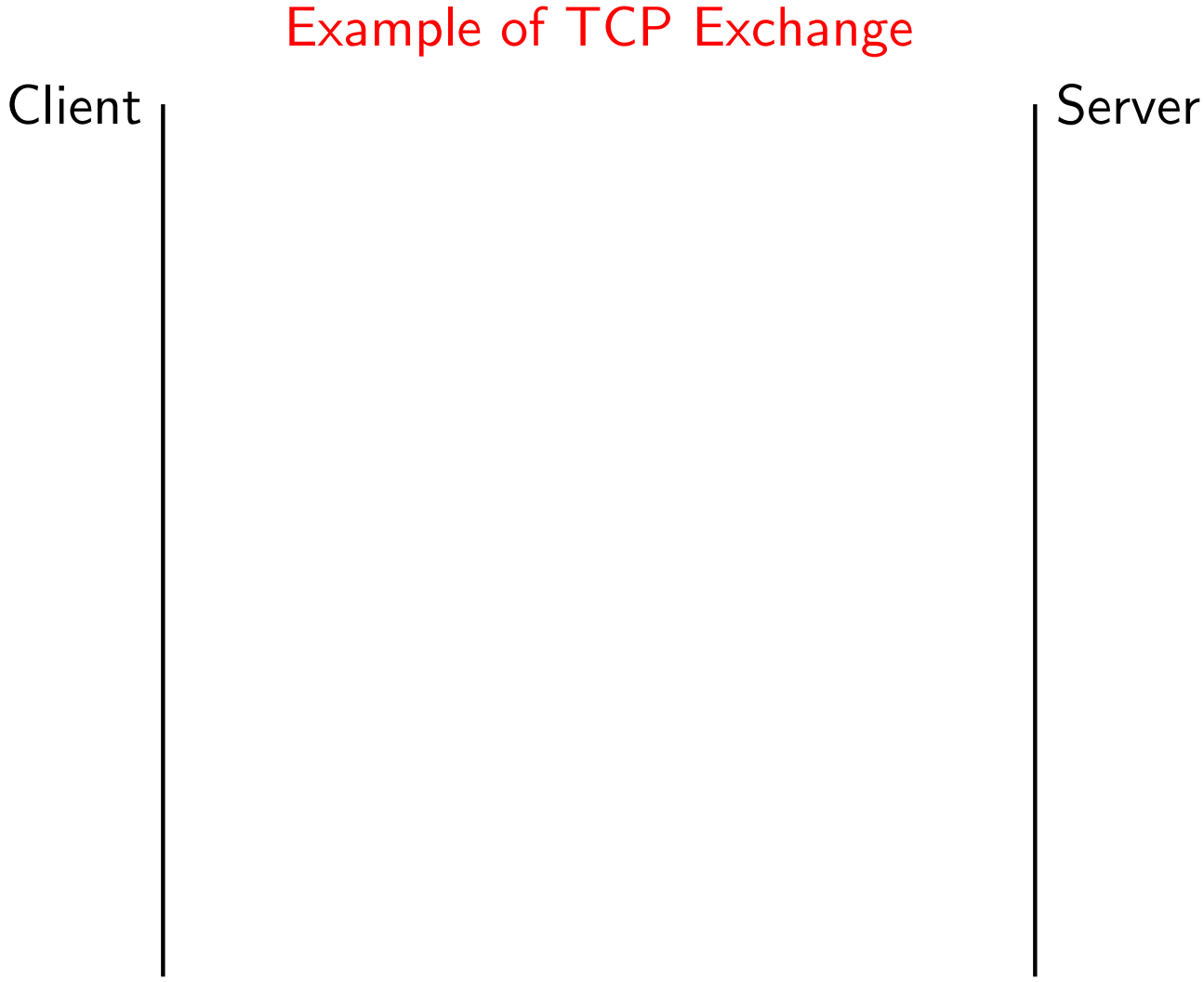
TCP Connection Release



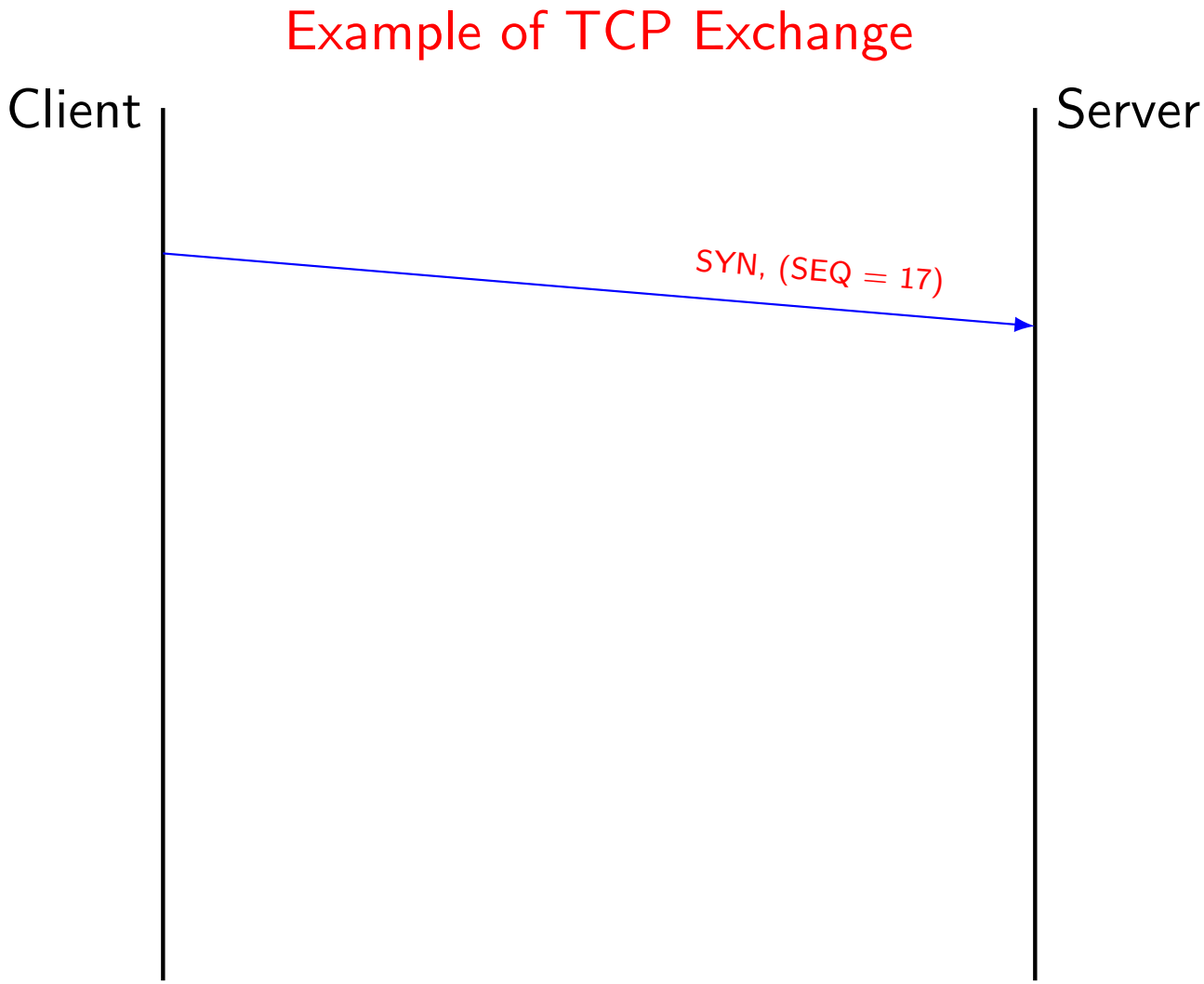
TCP Connection Release



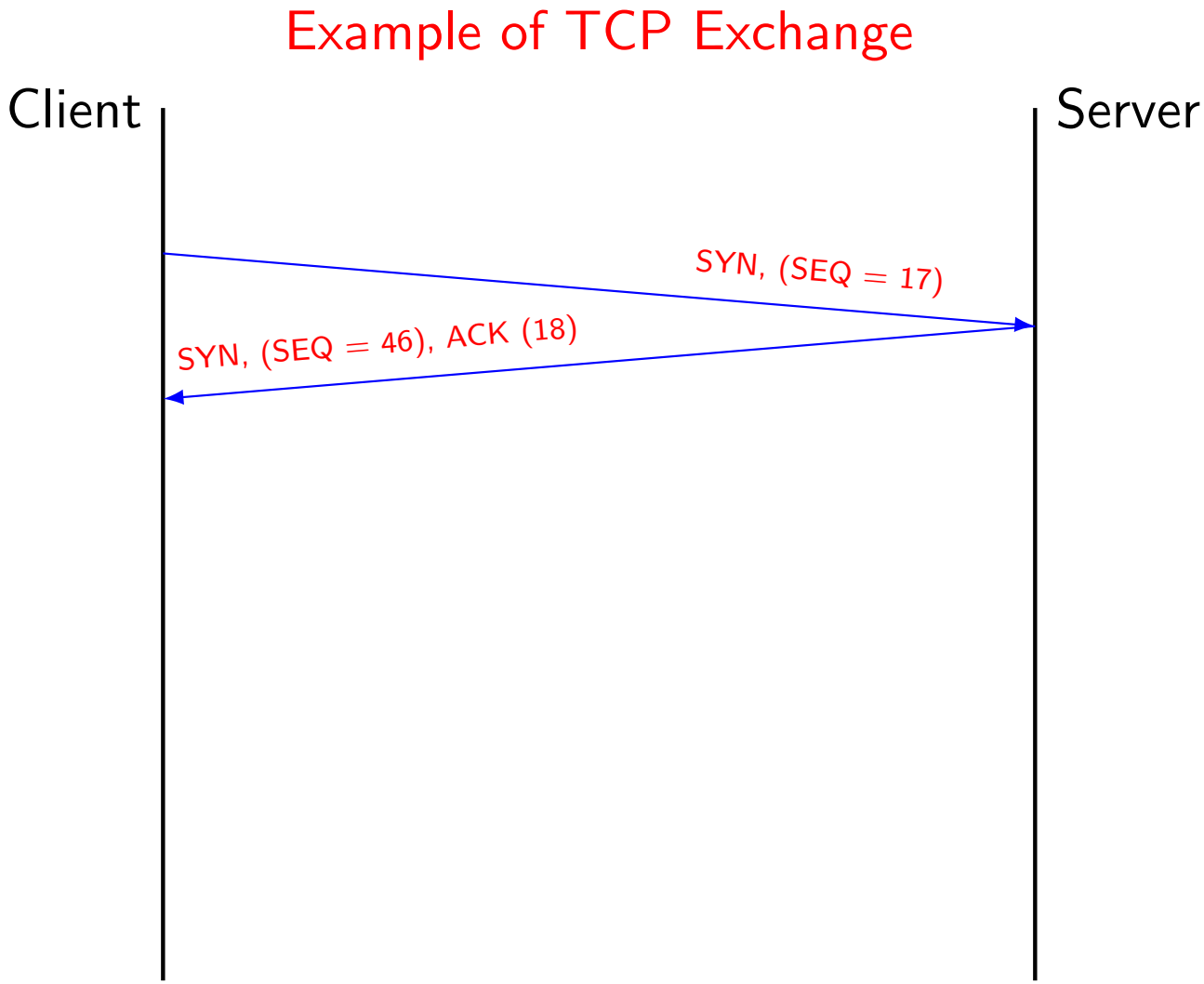
Example of TCP Exchange



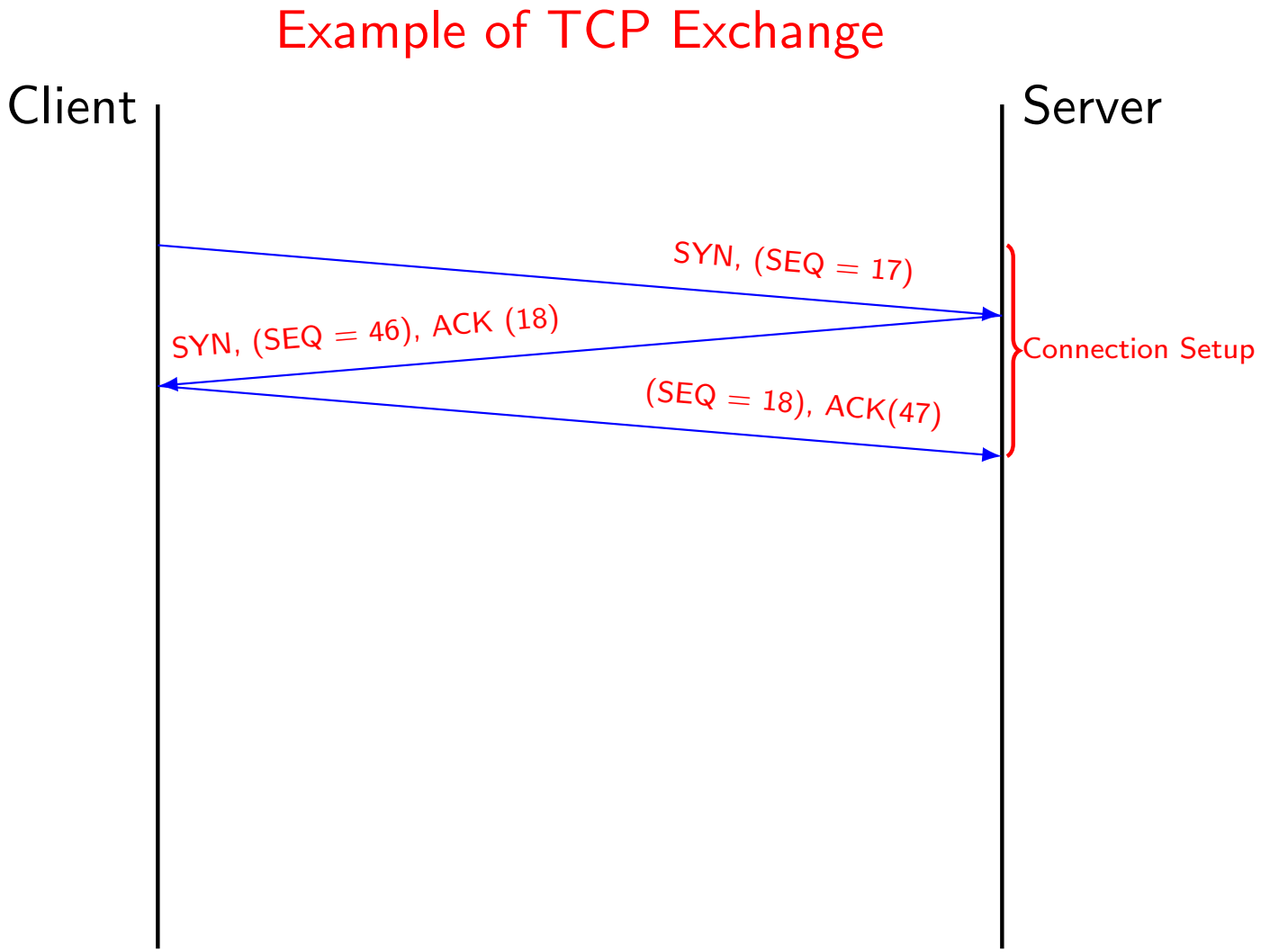
Example of TCP Exchange



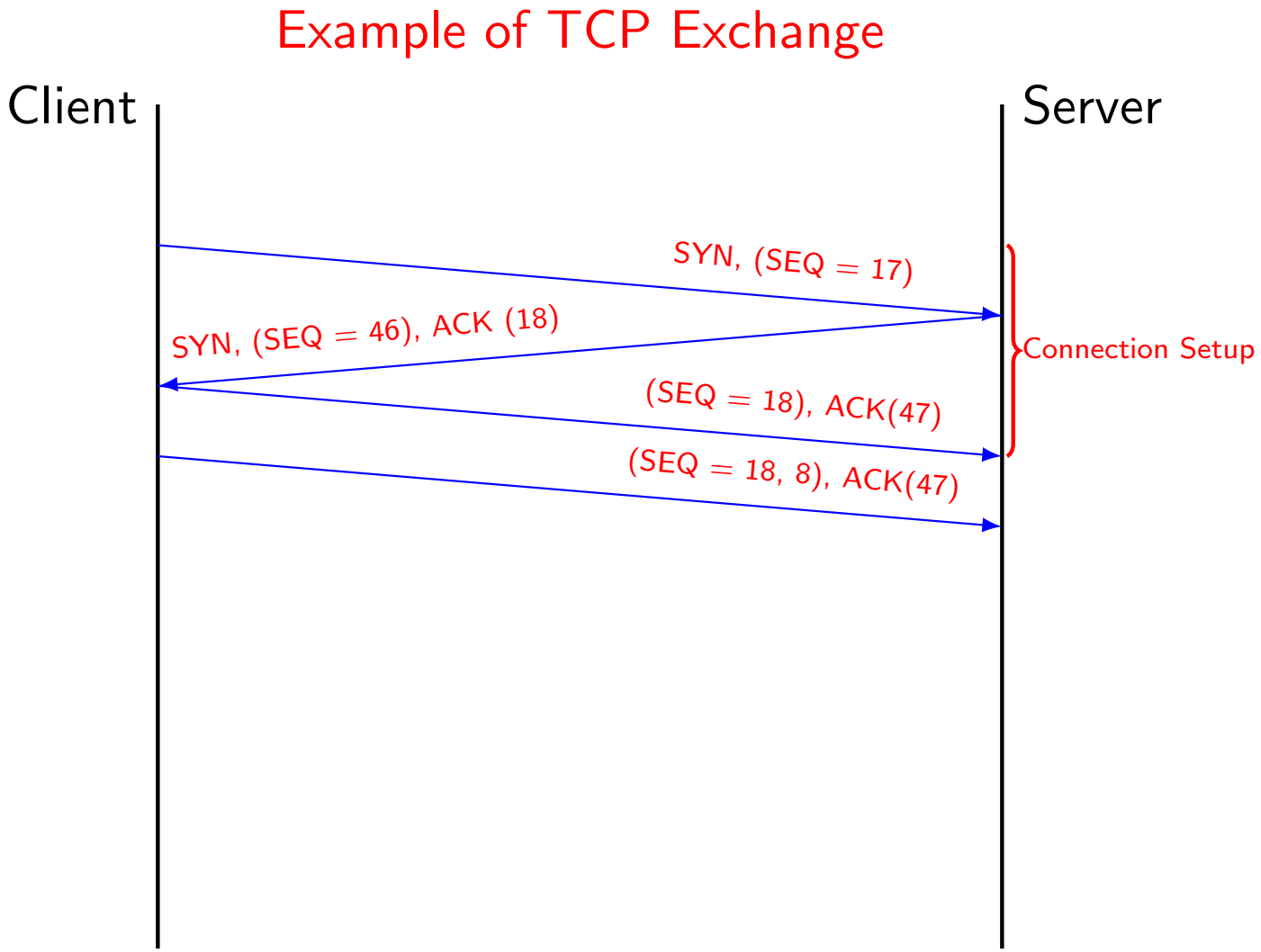
Example of TCP Exchange



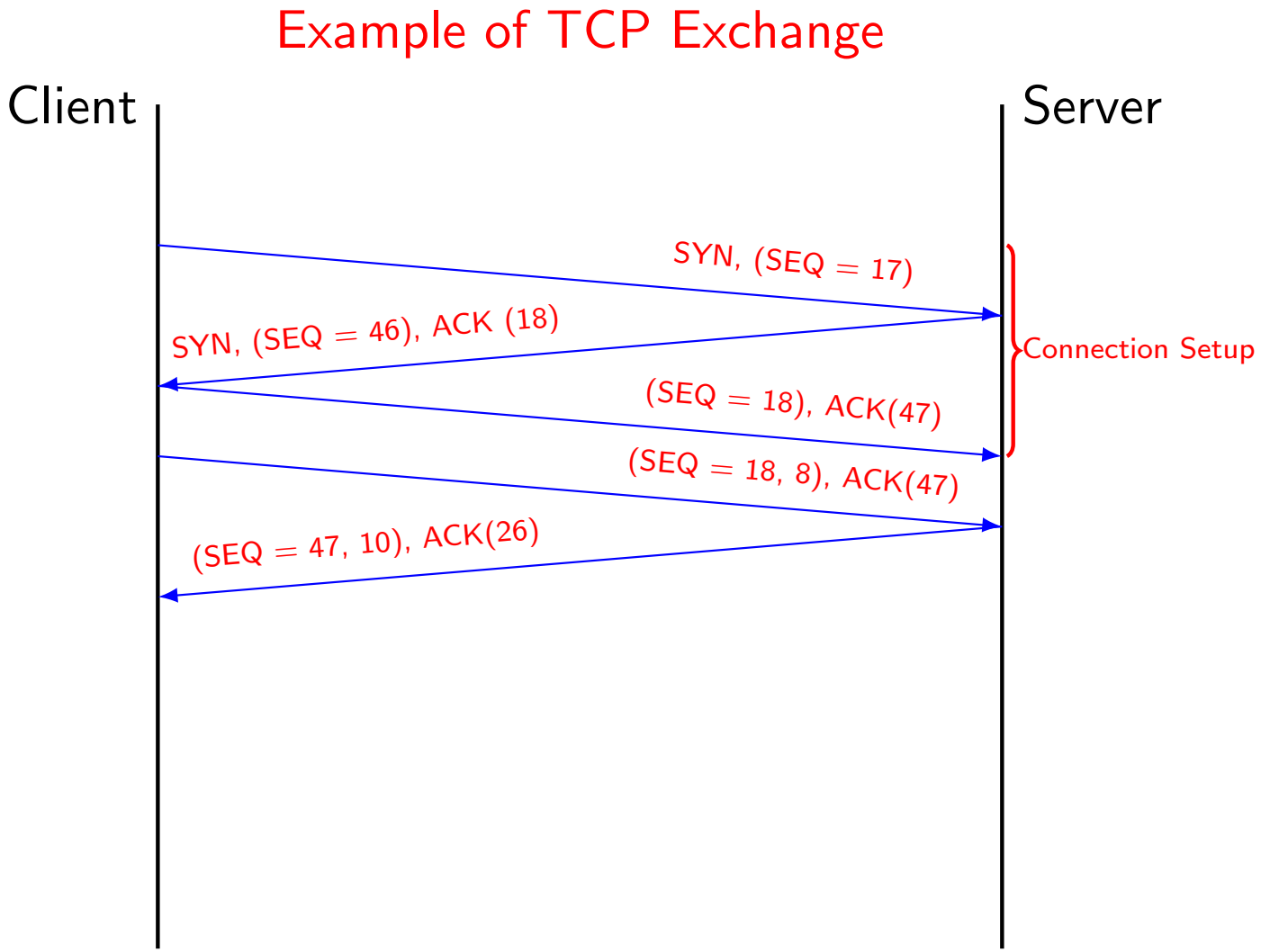
Example of TCP Exchange



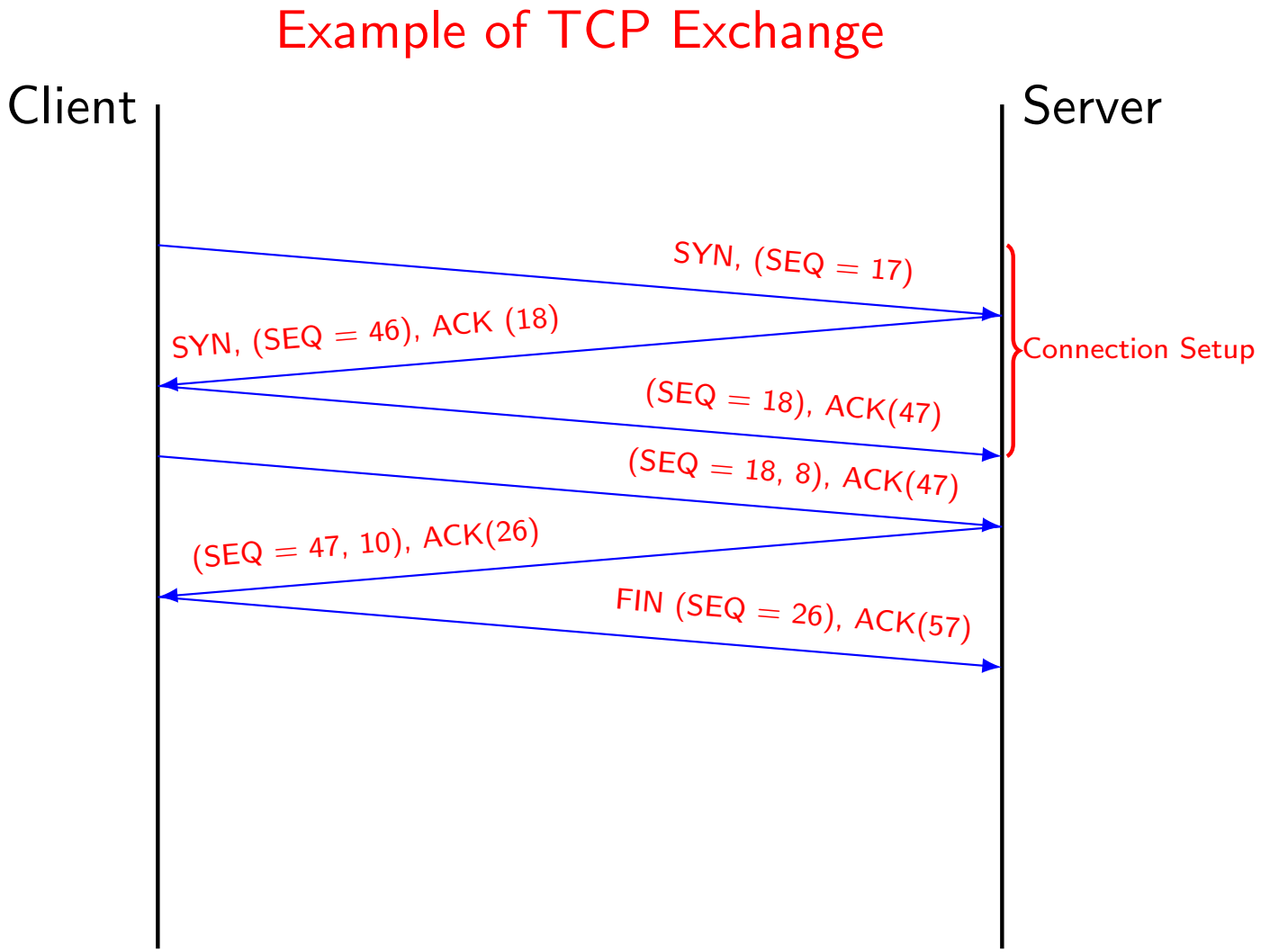
Example of TCP Exchange



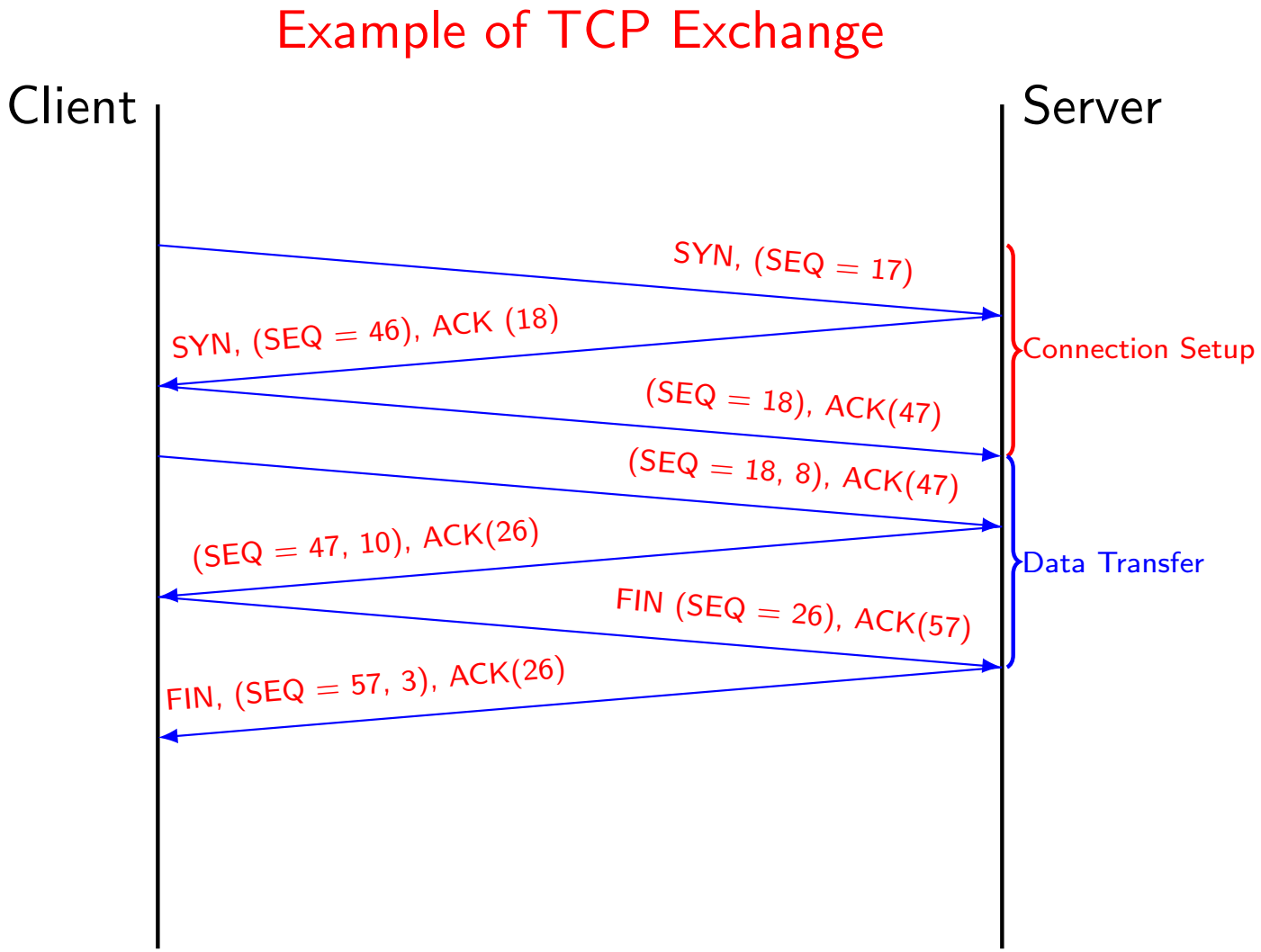
Example of TCP Exchange



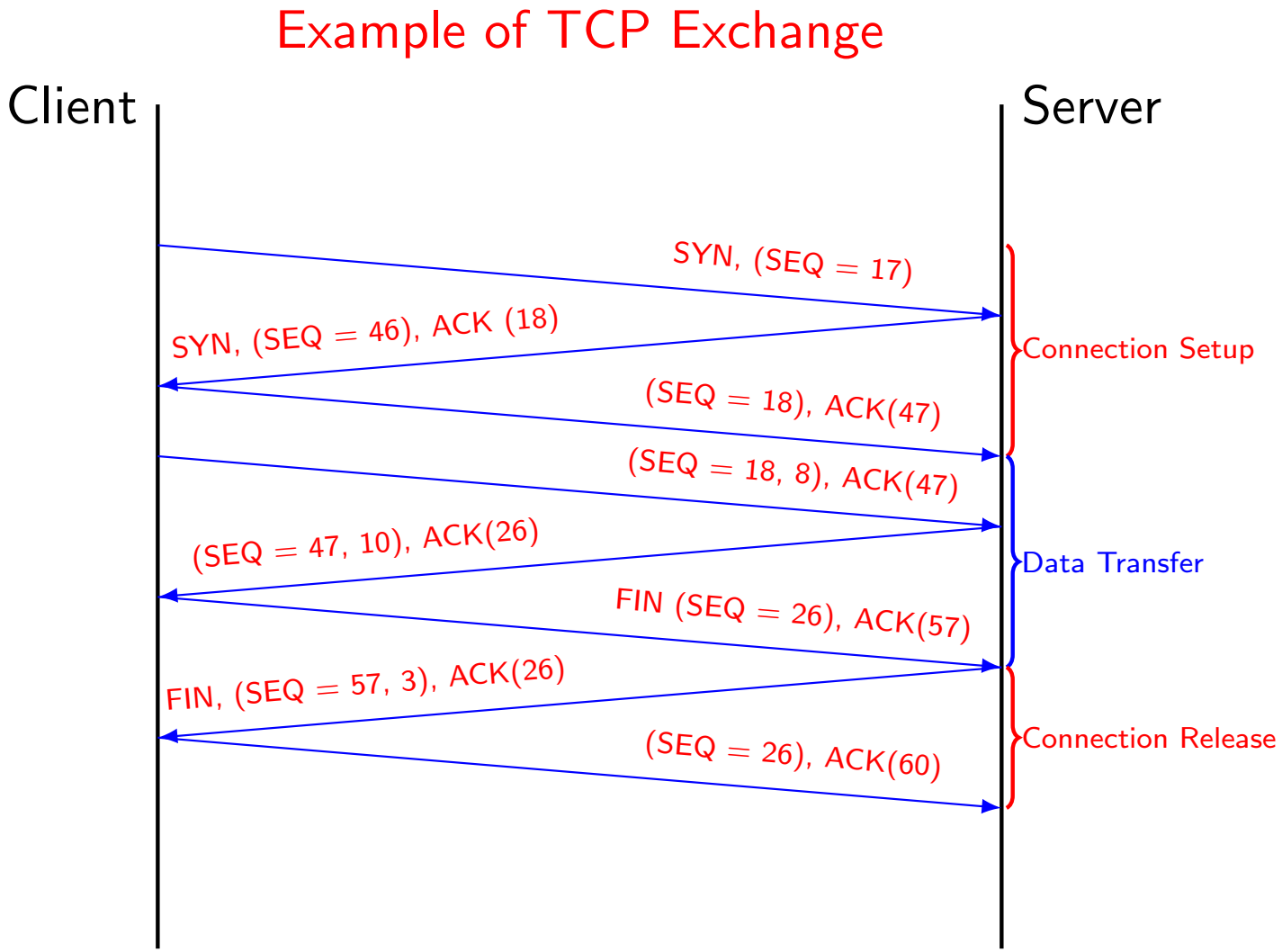
Example of TCP Exchange



Example of TCP Exchange



Example of TCP Exchange



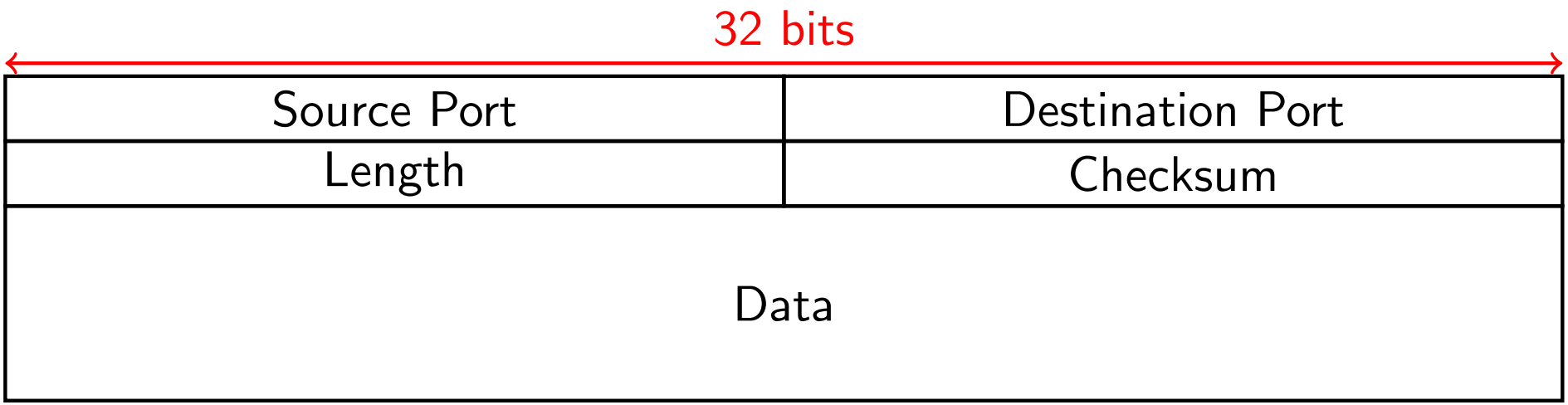
Congestion Control in TCP

- The TCP protocol reduces its traffic when it notices a segment loss
  - The sender maintains a congestion window with a value that depends on the network load
  - The sender maintains at the same time a flow control window where its value is managed by the receiver using the Window field of the TCP segment
  - The window with the minimal size determines the number of bytes that the sender can sends without receiving an acknowledgement

UDP

- User Datagram Protocol
- RFC 768
- Encapsulated in IP
  - Protocol = 17
- Connectionless Service
  - No retransmissions
  - Data may be delivered out of order
  - No error control (optional)
- Multiplexing
- The UDP protocol is useful for applications
  - that have negligible traffic to exchange
  - that are based on the request/reply model (SNMP, Time)

UDP



- Source Port, Destination port: same as in TCP
- Length: total length of the UDP packet expressed in bytes
- Checksum:
  - Same algorithm as in TCP (header + data + pseudo-header)
  - Optional

TCP/IP Services

- Telnet (RFC 854)
  - Virtual Terminal
  - Uses the TCP protocol
  - Port: 23
- FTP (RFC 959)
  - File Transfer Protocol
  - Uses the TCP protocol
  - Ports: 20 and 21
    - 20: FTP

    1. Commands

    - 21: FTP-DATA

    1. Data Transfer

TCP/IP Services

- Electronic Mail
  - Email
  - SMTP protocol (RFC 788)
    - Simple Message Transfer Protocol
    - Uses the TCP protocol
    - Port: 25
- HTTP (RFC 1945)
  - Hyper-Text Transfer Protocol
  - Uses the TCP protocol
  - Port: 80
- SNMP (RFC 1157)
  - Simple Network Management Protocol
  - Uses the UDP protocol