IP Addressing (IPv4)

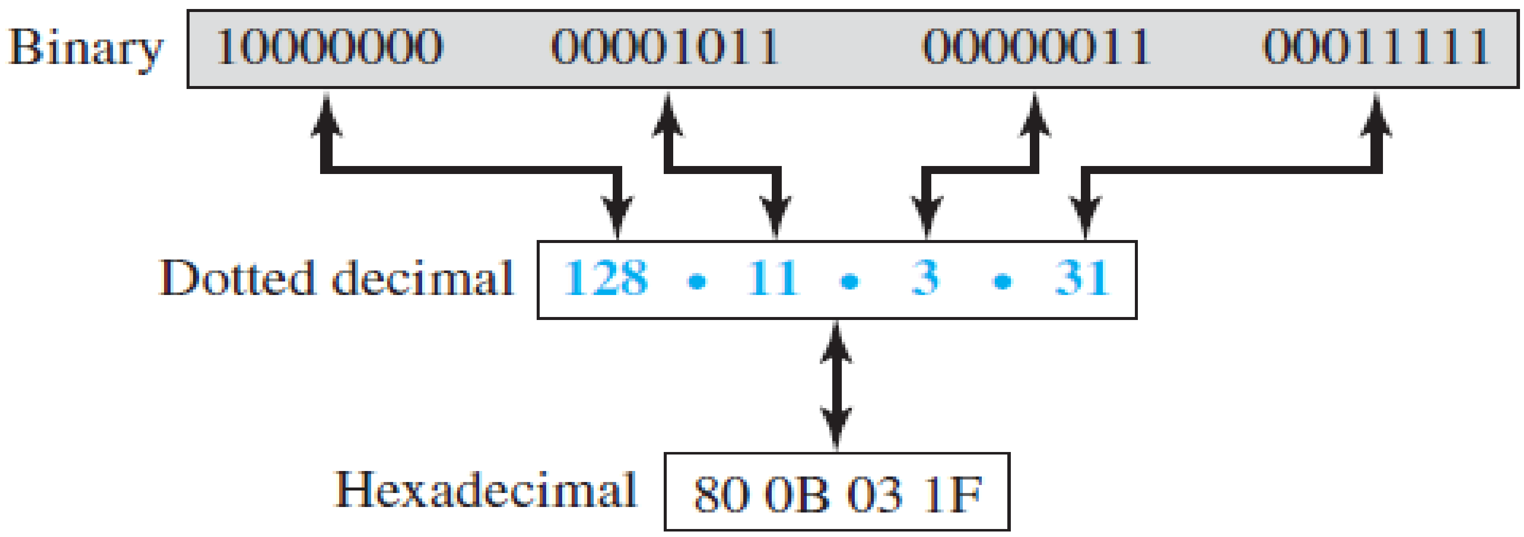
What is an IP Address?

- An IP address is a unique global address for a network interface
- An IP address uniquely and universally defines the connection of a host or a router to the Internet
- IP addresses are unique in the sense that each address defines one, and only one, connection to the Internet
  - If a device has two connections to the Internet, via two networks, it has two IP addresses
- An IPv4 address is a **32 bit long** identifier

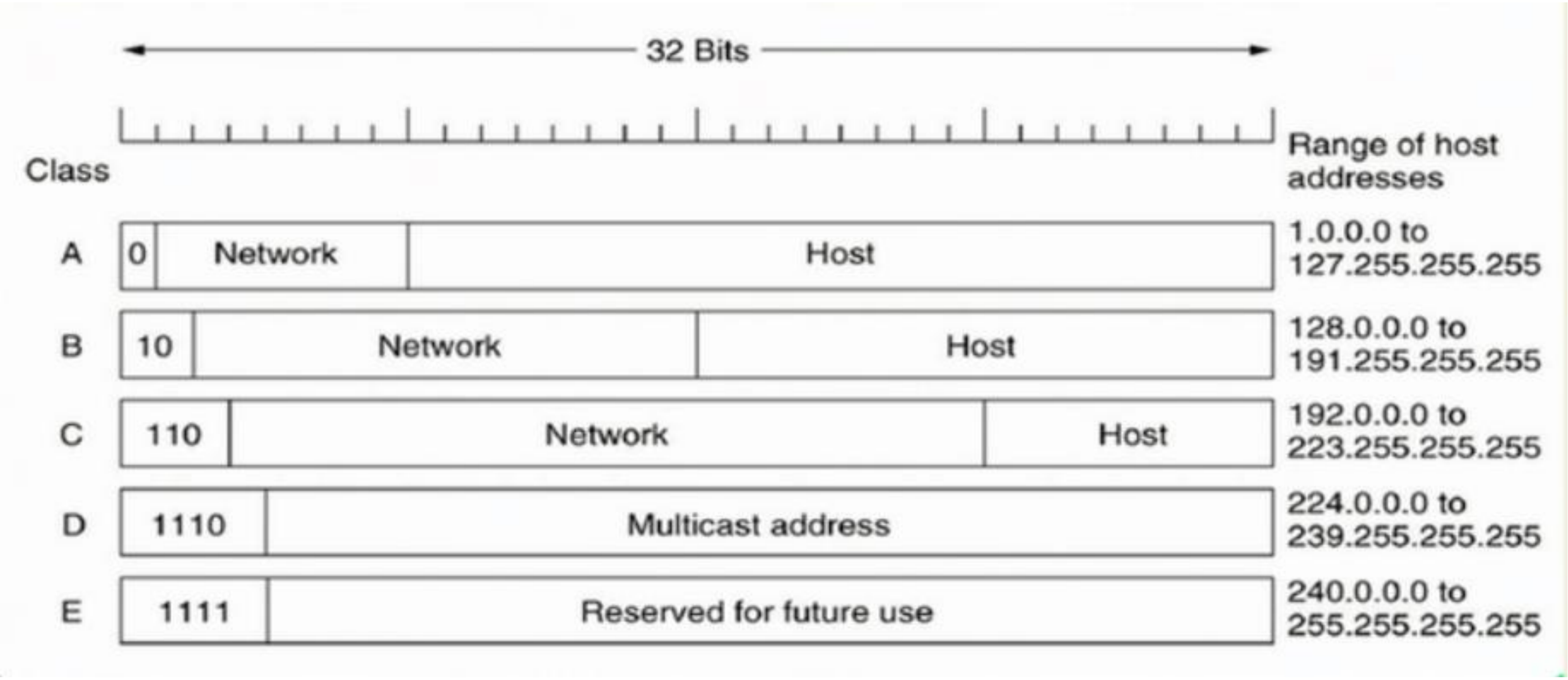
Notations in IPv4 addressing

- Binary and dotted-decimal notations are popular and commonly used
- Hexadecimal notation is not very common. It is often used in network programming

Three different notations in IPv4 addressing



Classful Addressing



- How to identify a class? – Use the first few bits
  - 0 – Class A; 10 – Class B; 110 – Class C; 1110 – Class D; 1111 – Class E

IP Addresses

- **Class A:**
  - Starts with binary 0
  - In class A, the network length is 8 bits, but since the first bit, which is 0, defines the class, we can use only seven bits as the network identifier
  - Therefore, only $2^7 = 128$ (0 to 127) networks are possible in class A
    - All 0 reserved (will be discussed later)
    - 01111111 (127) is also reserved for loopback (will be discussed later)
    - Hence, range: 1.x.x.x to 126.x.x.x

IP Addresses

- **Class B:**
  - Starts with binary 10
  - In class B, the network length is 16 bits, but since the first two bits, which are (10), define the class, we can use only 14 bits as the network identifier
  - Therefore, only 2^{14} = 16,384 networks are possible in class B
  - Range: 128.x.x.x to 191.x.x.x

IP Addresses

- **Class C:**
  - Starts with binary 110
  - The network length is 24 bits
  - Since first three bits define the class, we can use only 21 bits as the network identifier
  - There are 2^{21} = 2,097,152 networks possible in class C
  - Range: 192.x.x.x to 223.x.x.x

IP Addresses

- **Class D:**
  - Starts with binary 1110
  - Class D is not divided into prefix and suffix
  - It is used for multicast addresses
- **Class E:**
  - Starts with binary 1111
  - As in Class D, Class E is not divided into prefix and suffix
  - Reserved for future use

Network Address and Broadcast Address

- **Network address:** Identify a network
  - All 0's in the host address part
  - **Ex-1 (Class A):** 01111110.00000000.00000000.00000000 (126.0.0.0)
  - **Ex-2 (Class B):** 10111101.11101001.00000000.00000000 (189.233.0.0)
- **Broadcast address:** Send the data to **all the hosts** of a network
  - All 1's in the host address part
  - **Ex-1 (Class A):** 01111110.11111111.11111111.11111111 (126.255.255.255)
  - **Ex-2 (Class B):** 10111101.11101001.11111111.11111111 (189.233.255.255)

Subnetting, Supernetting, and Classless Inter-domain Routing (CIDR)

- Suppose, you have to configure a network with 255 hosts. Which IPv4 address class will you use – Class C or Class B?
  - Class C – Not possible
  - Class B – Huge address space is unutilized (using only 255 addresses out of possible $2^{16} - 2$ addresses)

Subnetting, Supernetting, and CIDR

- Split a large network or combine multiple small networks for efficient use of address space
  - **Subnetting:** Divide a large network into multiple small networks
  - **Supernetting:** Combine multiple small networks into a single large network

- **Subnet mask:** Denote the number of bits in the network address part

Subnetting, Supernetting, and CIDR

- **Subnet mask: (Contd…)**
  - An IP address is divided into two parts: network and host parts
  - For example, an IP class A address consists of 8 bits identifying the network and 24 bits identifying the host
  - Like an IP address, a subnet mask also consists of 32 bits. We use it to determine the network part and the host part of an address
  - The 1s in the subnet mask represent a network part, and the 0s represent a host part
  - A subnet mask must always be a series of 1s followed by a series of 0s
  - For example, the default subnet mask for a class A IP address in dotted decimal notation is 255.0.0.0

Classless Addressing (IPv4)

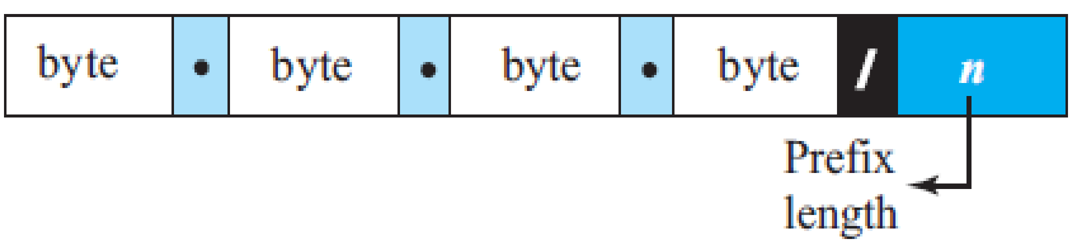
Classless Addressing

- In classless addressing, the class privilege was removed
- In classless addressing, variable-length blocks are used that belong to no classes
  - We can have a block of 1 address, 2 addresses, 4 addresses,128 addresses, and so on
- In classless addressing, the whole address space is divided into variable length blocks
- The prefix in an address defines the block (network); the suffix defines the node (device)
- One of the restrictions is that the number of addresses in a block needs to be a power of 2
- Unlike classful addressing, the prefix length in classless addressing is variable
  - We can have a prefix length that ranges from 0 to 32

Classless Addressing

- **Prefix Length: Slash Notation:**
  - The first question that we need to answer in classless addressing is how to find the prefix length if an address is given
  - In classless addressing, the prefix length is added to the address, separated by a slash
  - The notation is informally referred to as slash notation and formally as classless interdomain routing (CIDR)

Slash notation (CIDR)



Examples:
12.24.76.8/8
23.14.67.92/12
220.8.24.255/25

CIDR: Addressing Format

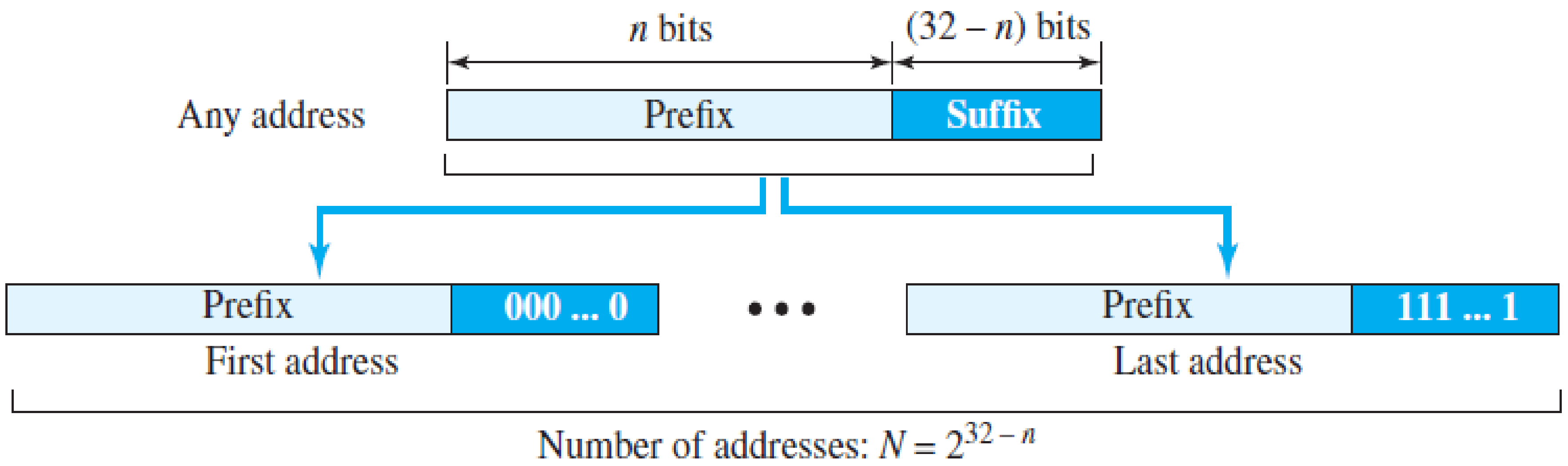
- We write the IP address as 192.180.83.235/12 in CIDR notation
  - The first 12 bits are the network address and rest (32 - 12) = 20 bits are for host address
  - The subnet mask is 255.240.0.0

Extracting Information from an Address

- Given any address in the block, we normally like to know three pieces of information about the block to which the address belongs:
  - The number of addresses, the first address in the block, and the last address
- We can easily find these three pieces of information as (also shown in Figure in the next slide):
  1. The number of addresses in the block is found as $N = 2^{32 - n}$ where n is the prefix length
  2. To find the first address, we keep the n leftmost bits and set the (32 – n) rightmost bits all to 0s
  3. To find the last address, we keep the n leftmost bits and set the (32 – n) rightmost bits all to 1s

Extracting Information from an Address

Information extraction in classless addressing



Extracting Information from an Address

- **Example:** A classless address is given as 167.199.170.82/27. We can find the above three pieces of information as follows:
  - The number of addresses in the network is $2^{32-n} = 2^5 = 32$ addresses
  - The first address can be found by keeping the first 27 bits and changing the rest of the bits to 0s as:

Address: 167.199.170.82/27 10100111 11000111 10101010 01010010

First address: 167.199.170.64/27 10100111 11000111 10101010 01000000

  - The last address can be found by keeping the first 27 bits and changing the rest of the bits to 1s as:

Address: 167.199.170.82/27 10100111 11000111 10101010 01010010

Last address: 167.199.170.95/27 10100111 11000111 10101010 01011111

CIDR: Block Allocation

- One issue in classless addressing is block allocation
- The ultimate responsibility of block allocation is given to a global authority called the Internet Corporation for Assigned Names and Numbers (ICANN)
- However, ICANN does not normally allocate addresses to individual Internet users
- It assigns a large block of addresses to an ISP (or a larger organization), which further allocate addresses to individual Internet users

CIDR: Block Allocation

- For the proper operation of the CIDR, two restrictions need to be applied to the allocated block

  1. The number of requested addresses, N, needs to be a power of 2. The reason is that $N = 2^{32-n}$ or $n = 32 - \log_2 N$. If N is not a power of 2, we cannot have an integer value for n (prefix length)

  2. The requested block needs to be allocated where there is an adequate number of contiguous addresses available in the address space

- **Example:** An ISP has requested a block of 1000 addresses. Since 1000 is not a power of 2, 1024 addresses are granted. The prefix length is calculated as $n = 32 - \log_2 1024 = 22$

  - An available block, 18.14.12.0/22, is granted to the ISP

Special Addresses

- **This-host Address:**
  - The only address in the block 0.0.0.0/32 is called the this-host address
  - It is used whenever a host needs to send an IP datagram but it does not know its own address. Such host uses it as the source address
- **Limited-broadcast Address:**
  - The only address in the block 255.255.255.255/32 is called the limited-broadcast address
  - It is used whenever a router or a host needs to send a datagram to all devices in its own network
  - The routers in the network, however, block the packet having this address as the destination; the packet cannot travel outside the network

Special Addresses

- **Loopback Address:**
  - The block 127.0.0.0/8 is called the loopback address
  - A packet with one of the addresses in this block as the destination address never leaves the host; it will remain in the host
  - Any address in the block is used to test a piece of software in the machine
  - For example, we can write a client and a server program in which one of the addresses in the block is used as the server address. We can test the programs using the same host to see if they work before running them on different computers

Special Addresses

- **Private Addresses:**
  - Four blocks are assigned as private addresses: 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16, and 169.254.0.0/16
  - We use any one of these address blocks when we use NAT (Network Address Translation) method for setting up an organization network

IPv6

IPv6

- The network layer protocol in the TCP/IP protocol suite is currently IPv4.
- Although IPv4 is well designed.
- IPv4 has some deficiencies that make it unsuitable for the fast-growing Internet.
  - The small size of the address space in IPv4.
  - No any built in security features in IPv4.
  - The address depletion of IPv4.
- Internet Protocol version 6 (IPv6) or IP new generation (IPng)

IPv6 Features

- Larger address space.
- Globally unique and hierarchical addressing.
- Optimized routing table using prefixes rather than address classes.
- Support for encapsulation.
- Built-in authentication and encryption.
- It has base header of 40 bytes (fixed).

IPv6 Addressing

- 128 bit address

The colon hexadecimal notation

- Divides the address into eight sections.
- Each made of four hexadecimal digits.
- Separated by colons(:).
- for example:

Binary (128 bits)	1111111011110110	...	1111111100000000
Colon Hexadecimal	FEF6:BA98:7654:3210:ADEF:BBFF:2922:FF00		

- The leading zeros of a section can be omitted.
  - 0074 can be written as 74

IPv6 Addressing

Zero compression

- If there are consecutive sections consisting of zeros only.
- Remove all the zeros and replace them with a double colon (::).

FDEC:0:0:0:0:BBFF:0:FFFF ⟶ FDEC::BBFF:0:FFFF

- allowed only once per address.

Mixed Notation

- colon hex and dotted-decimal notation.
- When an IPv4 address is embedded in an IPv6 address.
- **Example ::130.24.24.18**

CIDR Notation

FDEC::BBFF:0:FFFF/60

IPv6 Addressing : Address Space

- The address space of IPv6 contains $\mathbf{2^{128}}$ addresses.
- Almost 2% percent of the addresses in the space can be assigned to the people on planet Earth.
- Three Address Types
  - Unicast Address
  - Anycast Address
    - A group of computers that all share a single address.
    - Delivered to only one member of the group.
    - IPv6 does not designate a block for anycasting.
  - Multicast Address
    - IPv6 has designated a block for multicasting
- Does not define broadcasting.

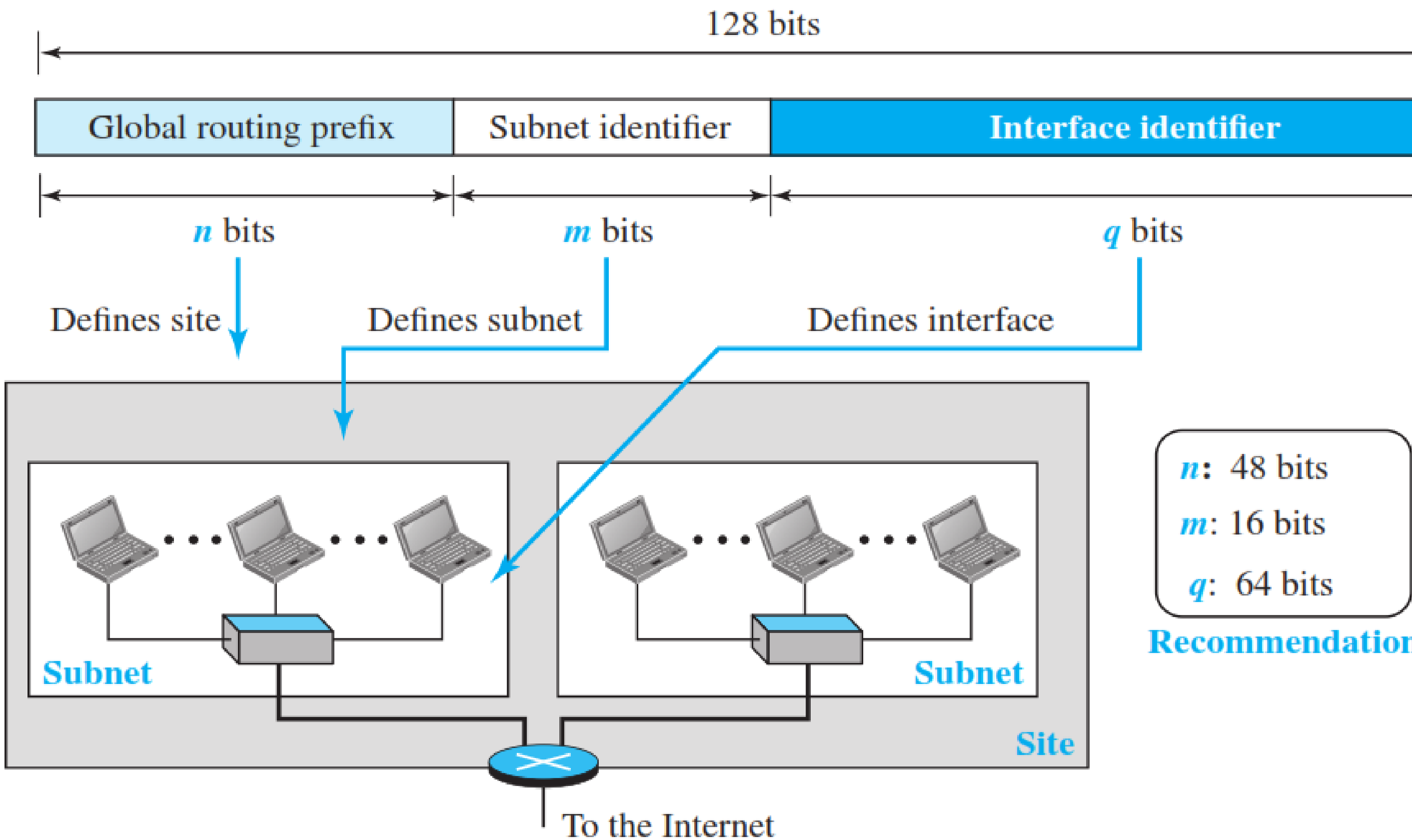
Address Space Allocation

- The address space of IPv6 is divided into several blocks of varying size.
- Most of the blocks are still unassigned and reserved for future use.

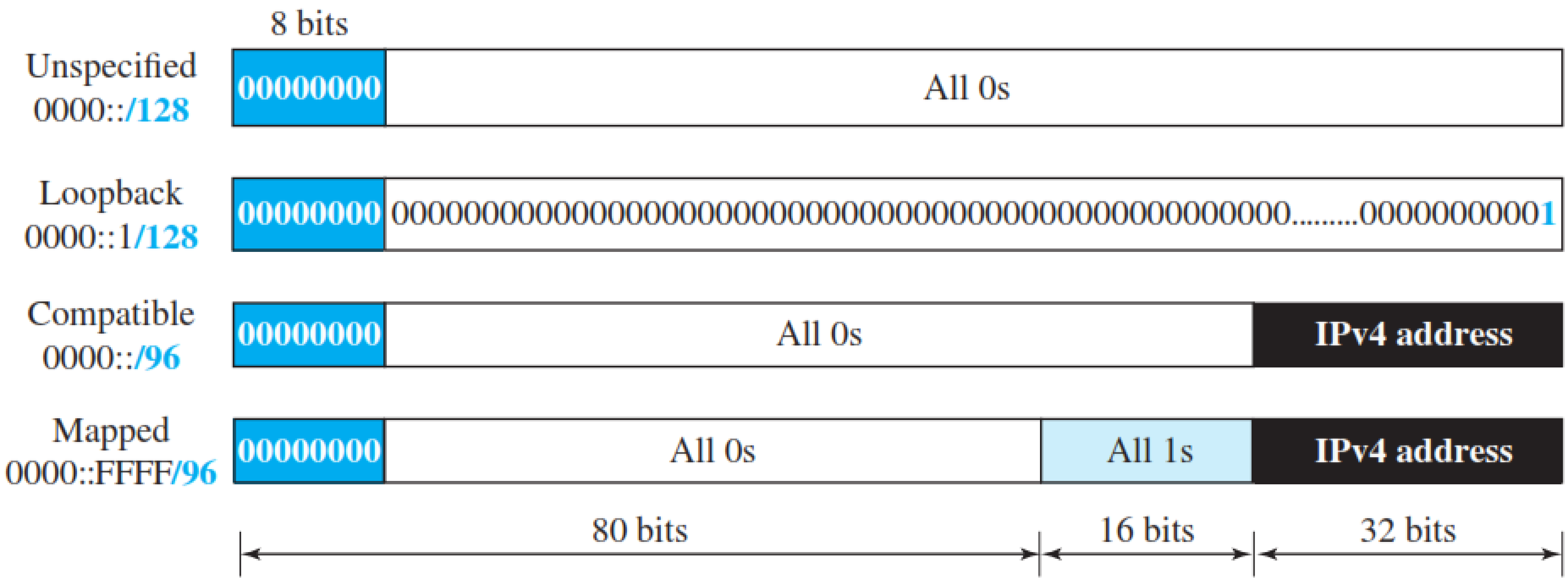
Block prefix	*CIDR*	*Block assignment*	*Fraction*
0000 0000	0000::/8	Special addresses	1/256
001	**2000::/3**	**Global unicast**	**1/8**
1111 110	FC00::/7	Unique local unicast	1/128
1111 1110 10	FE80::/10	Link local addresses	1/1024
1111 1111	FF00::/8	Multicast addresses	1/256

Global Unicast Address

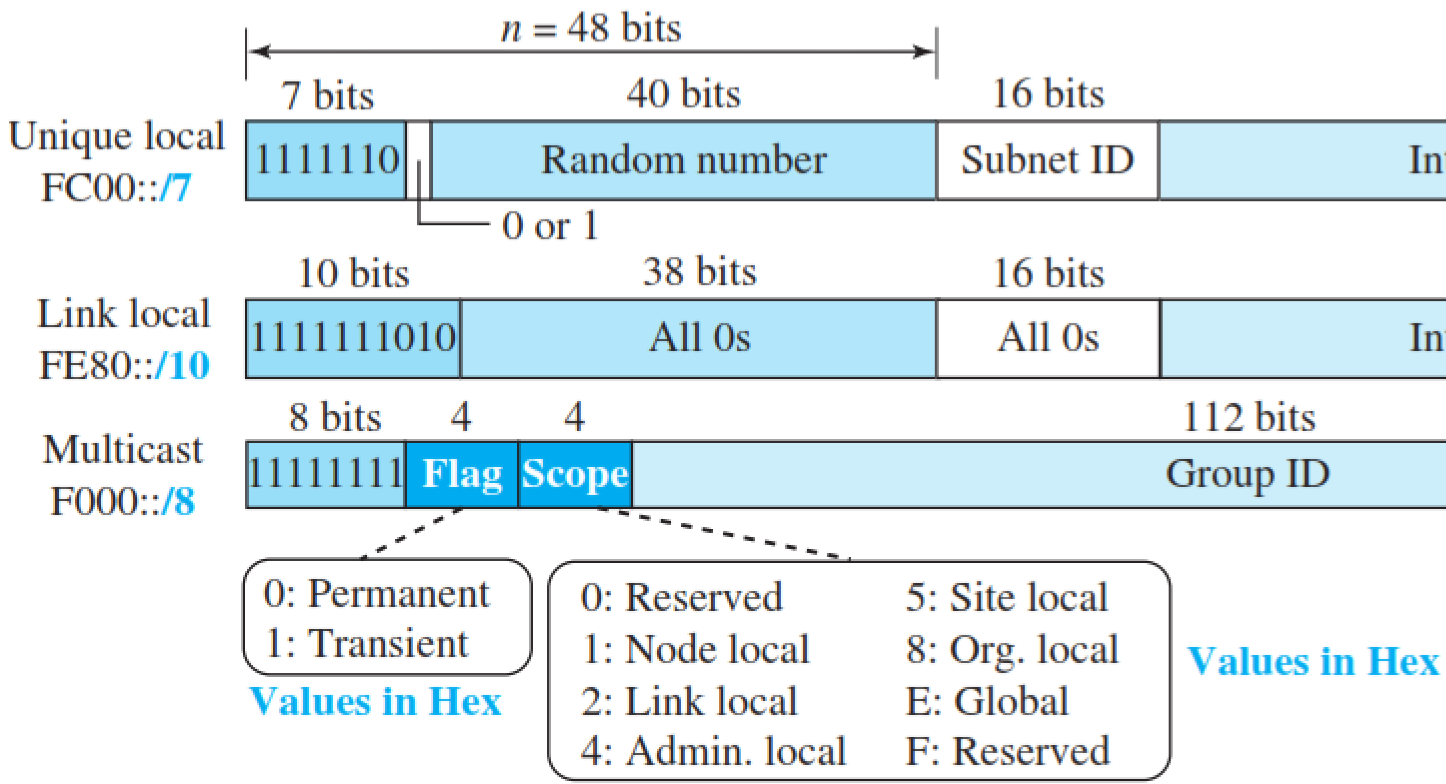
- Used for unicast (one-to-one) communication between two hosts.
- An address in this block is divided into three parts.



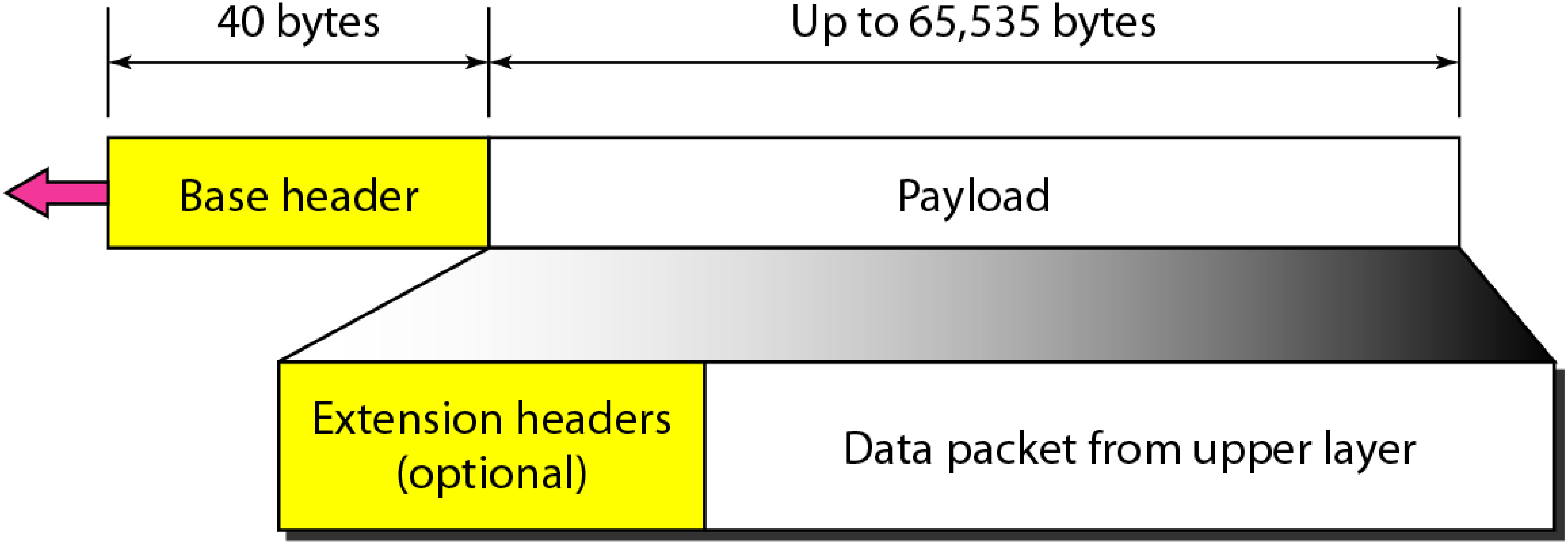
Special Addresses



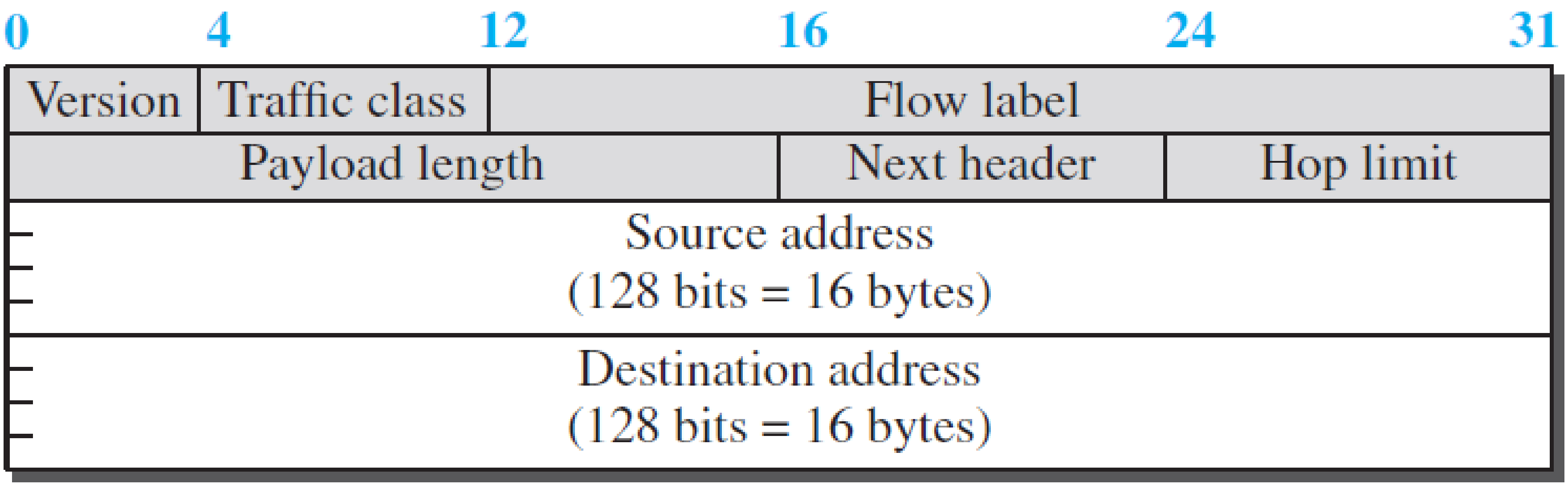
Other Assigned Addresses



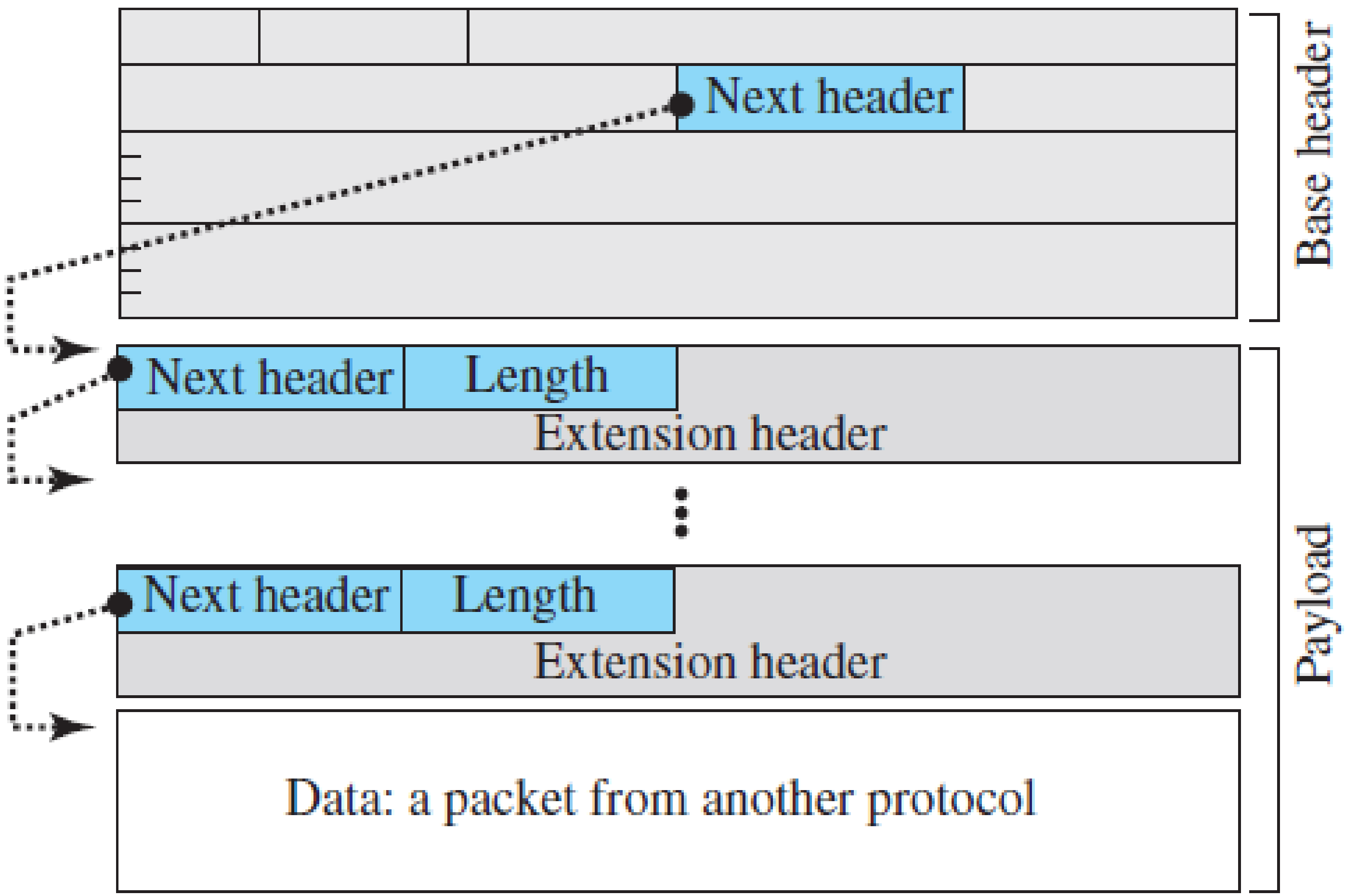
IPv6 Datagram



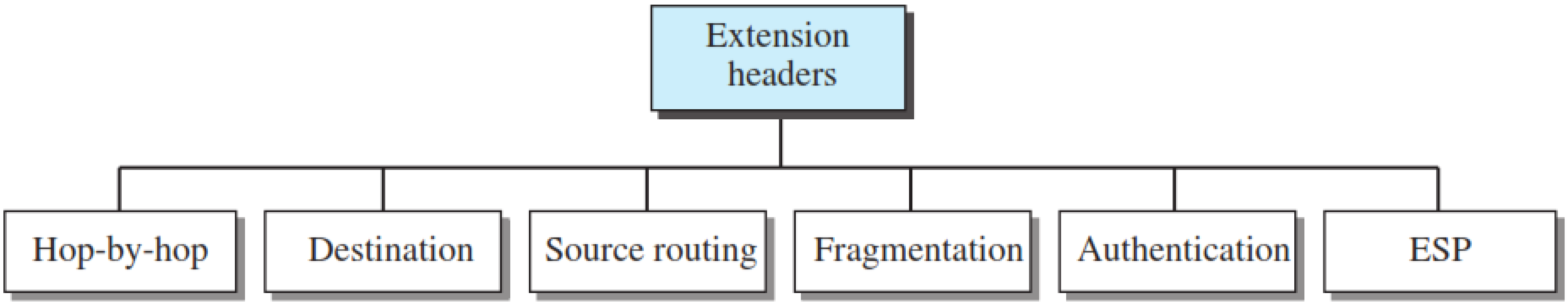
IPv6 Base Header



IPv6 Payload



IPv6 Datagram : Extension Headers



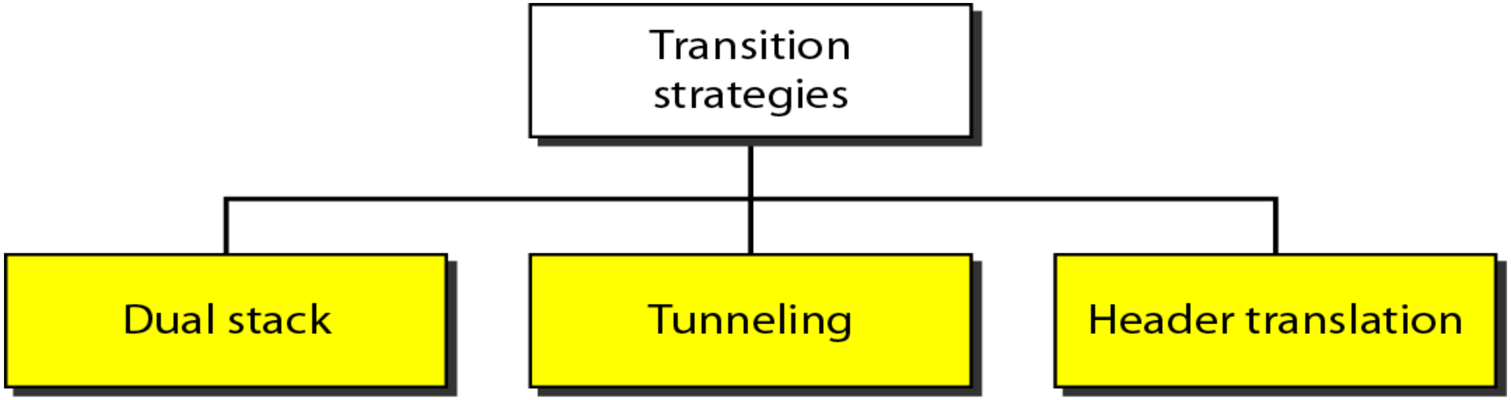
IPv6 vs IPv4 Header

Comparison
1. The header length field is eliminated in IPv6 because the length of the header is fixed in this version.
2. The service type field is eliminated in IPv6. The priority and flow label fields together take over the function of the service type field.
3. The total length field is eliminated in IPv6 and replaced by the payload length field.
4. The identification, flag, and offset fields are eliminated from the base header in IPv6. They are included in the fragmentation extension header.
5. The TTL field is called hop limit in IPv6.
6. The protocol field is replaced by the next header field.
7. The header checksum is eliminated because the checksum is provided by upper-layer protocols; it is therefore not needed at this level.
8. The option fields in IPv4 are implemented as extension headers in IPv6.

Transition from IPv4 to IPv6

- Because of the huge number of systems on the Internet, the transition from IPv4 to IPv6 cannot happen suddenly
- It takes a considerable amount of time before every system in the Internet can move from IPv4 to IPv6
- The transition must be smooth to prevent any problems between IPv4 and IPv6 systems

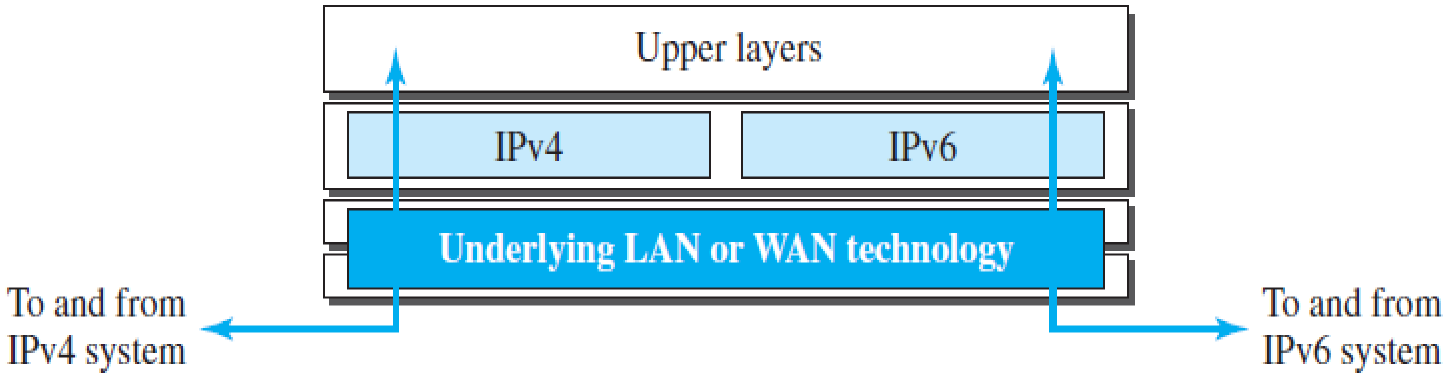
Three transition strategies



Internet Transition: Migrating from IPv4 to IPv6

- **Dual Stack:**
- It is recommended that all hosts, before migrating completely to version 6, have a **dual stack** of protocols during the transition
- A station must run IPv4 and IPv6 simultaneously until all the Internet uses IPv6
- To determine which version to use when sending a packet to a destination, the source host queries the DNS
  - If the DNS returns an IPv4 address, the source host sends an IPv4 packet
  - If the DNS returns an IPv6 address, the source host sends an IPv6 packet

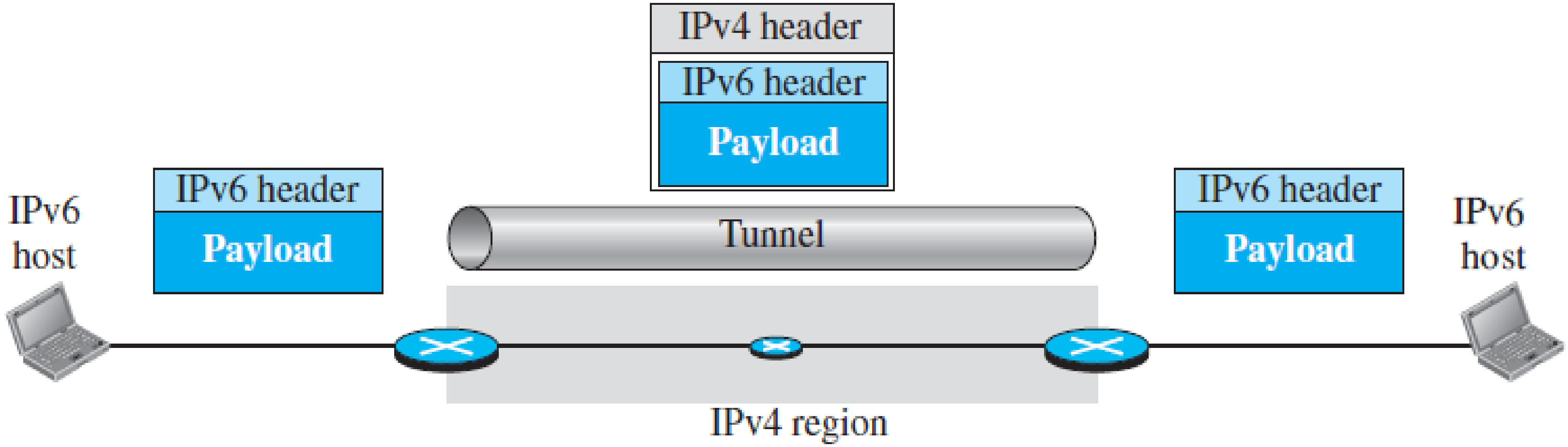
Dual stack



Internet Transition: Migrating from IPv4 to IPv6

- **Tunneling:** It is a strategy used when two computers using IPv6 want to communicate with each other and the packet must pass through a region that uses IPv4
  - To pass through this region, the packet must have an IPv4 address
- So the IPv6 packet is encapsulated in an IPv4 packet when it enters the region, and decapsulated when it exits the region

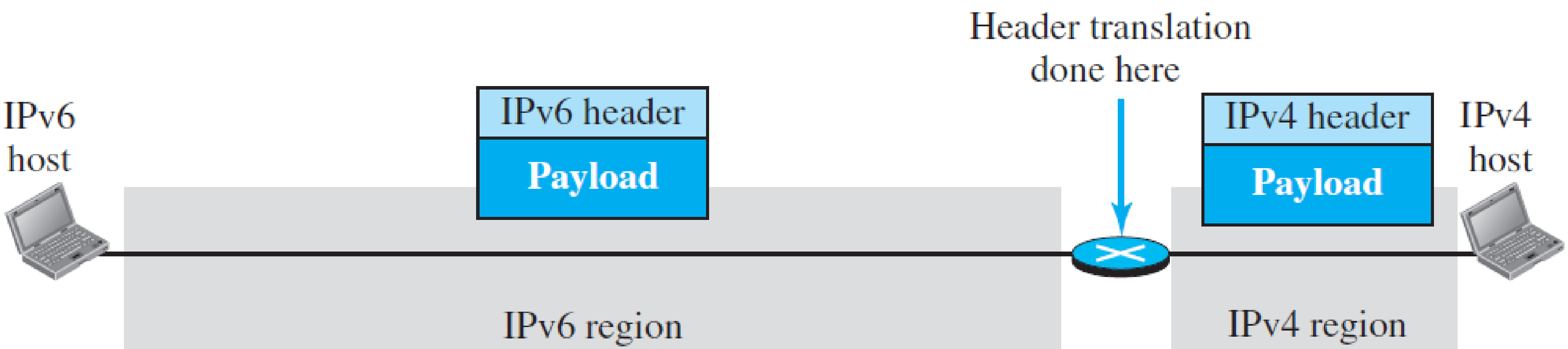
Tunneling strategy



Internet Transition: Migrating from IPv4 to IPv6

- **Header translation:** Header translation is necessary when the majority of the Internet has moved to IPv6 but some systems still use IPv4
- Suppose, the sender wants to use IPv6, but the receiver does not understand IPv6. In this case, the header format must be totally changed through header translation
  - Address must be translated as well during translation
  - Take low order 32 bits (i.e. rightmost 32 bits) for IPv6 to IPv4
  - Append :: FFFF/96 prefix for IPv4 to IPv6

Header translation strategy

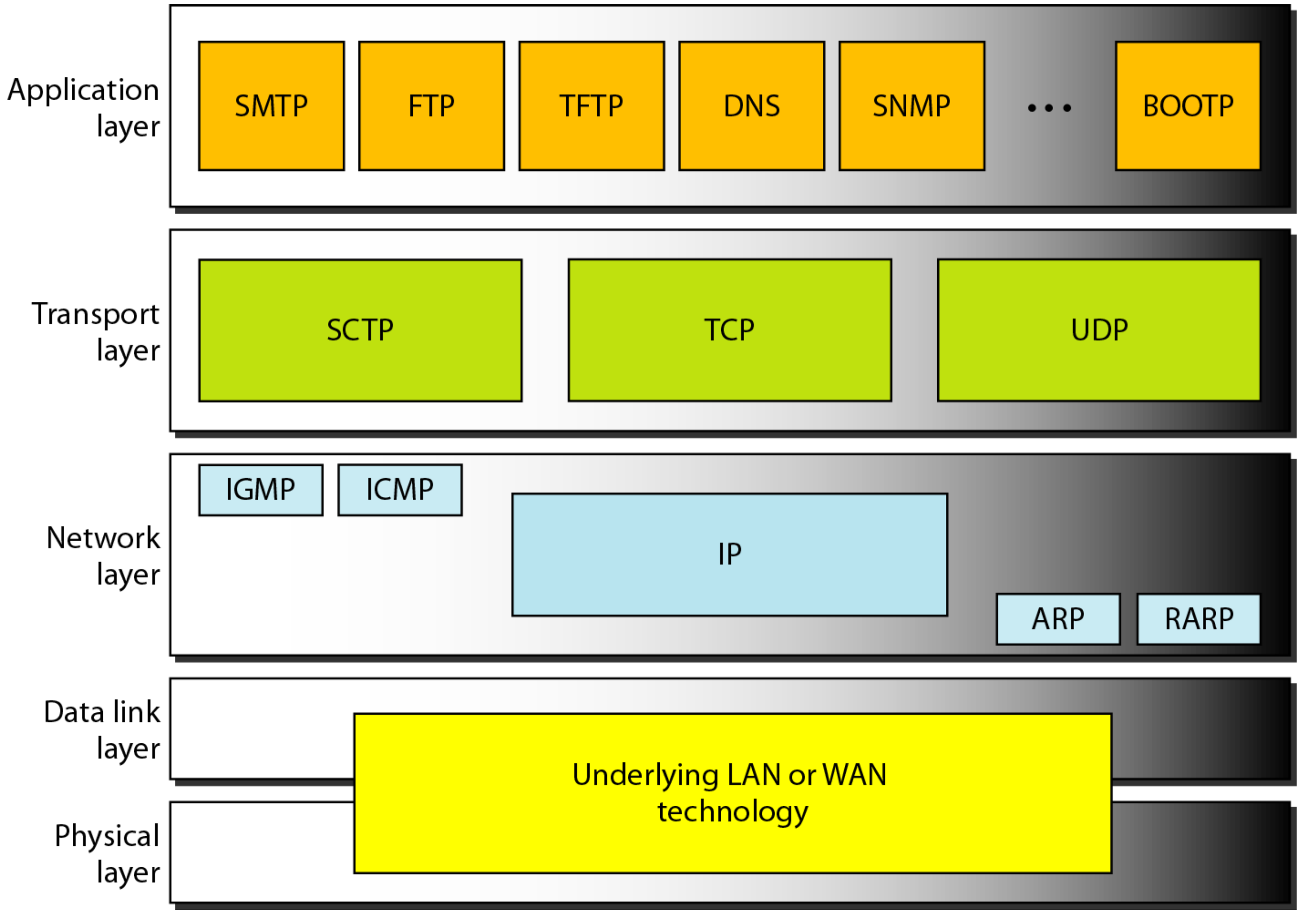


Transport Layer

Transport Layer

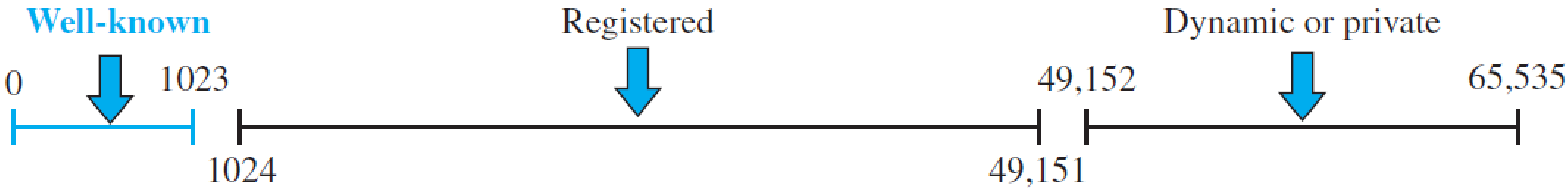
- The transport layer is responsible for process-to-process delivery
- It is responsible for creating the end-to-end connection between hosts for which it mainly uses TCP and UDP
- It is also responsible for congestion control, error control and flow control
- Transport layer protocols:
  - Transmission Control Protocol (TCP)
  - User Datagram Protocol (UDP)
  - Stream Control Transmission Protocol (SCTP)

Position of UDP, TCP, and SCTP in TCP/IP suite



Addressing: Port Numbers

- Transport layer uses port numbers as process identifier.
- The port numbers are integers between 0 and 65,535 (16 bits).
- The client program defines itself with a port number, called the ephemeral port number.
- TCP/IP has decided to use universal port numbers for servers; these are called well-known port numbers.



Addressing: Port Numbers

Port	*Protocol*	*UDP*	*TCP*	*SCTP*	*Description*
7	Echo	√	√	√	Echoes back a received datagram
9	Discard	√	√	√	Discards any datagram that is received
11	Users	√	√	√	Active users
13	Daytime	√	√	√	Returns the date and the time
17	Quote	√	√	√	Returns a quote of the day
19	Chargen	√	√	√	Returns a string of characters
20	FTP-data		√	√	File Transfer Protocol
21	FTP-21		√	√	File Transfer Protocol
23	TELNET		√	√	Terminal Network
25	SMTP		√	√	Simple Mail Transfer Protocol
53	DNS	√	√	√	Domain Name Service
67	DHCP	√	√	√	Dynamic Host Configuration Protocol
69	TFTP	√	√	√	Trivial File Transfer Protocol
80	HTTP		√	√	HyperText Transfer Protocol
111	RPC	√	√	√	Remote Procedure Call
123	NTP	√	√	√	Network Time Protocol
161	SNMP-server	√			Simple Network Management Protocol
162	SNMP-client	√			Simple Network Management Protocol

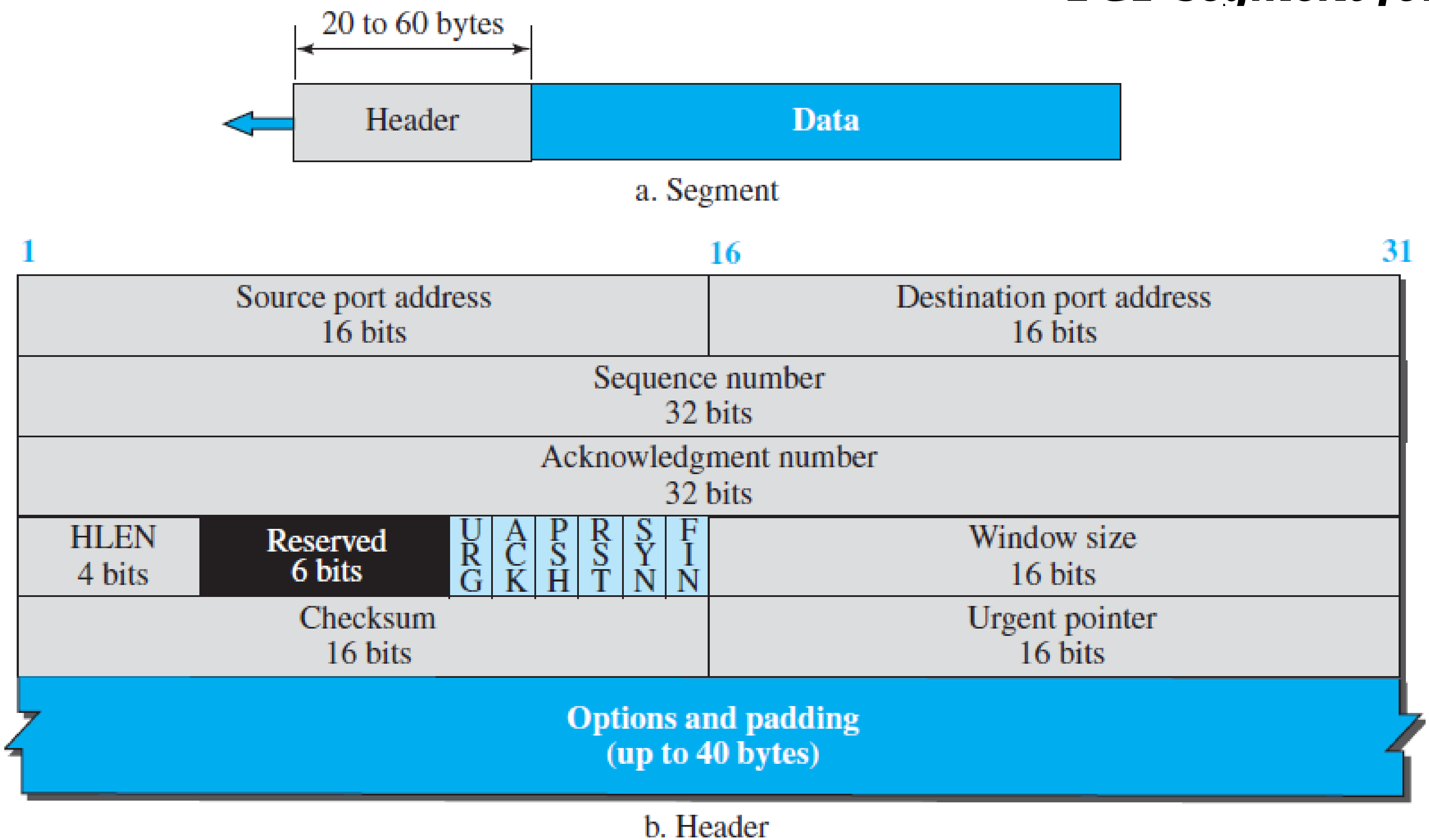
Transmission Control Protocol (TCP)

- TCP was specifically designed to provide a reliable end-to-end byte stream over an unreliable internetwork
- In internetwork - different parts may have widely different topologies, bandwidths, delays, packet sizes, and other parameters
- **TCP Services:**
  - Process-to-Process Communication
  - Stream Delivery Service
  - Full-Duplex Communication
  - Connection-Oriented Service
  - Reliable Service

The TCP Protocol: TCP Segment

- A packet in TCP is called a *segment*

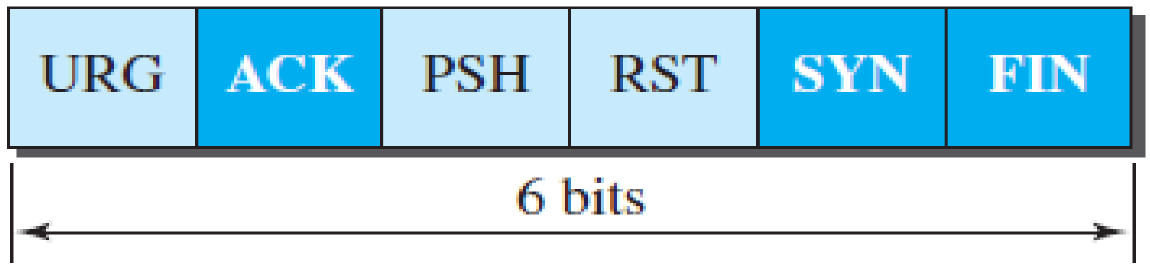
TCP segment format



The TCP Protocol: The Header

1-bit flags (control field):

- **URG** is set to 1 if the urgent pointer is in use
- If **ACK** is 0, the segment does not contain an ACK
- The **PSH** bit indicates PUSHed data.
  - The receiver is hereby kindly requested to deliver the data to the application immediately
- The **RST** bit is used to abruptly reset a connection
- The **SYN** bit is used to establish connections
- The **FIN** bit is used to release a connection



The TCP Protocol: The Header

- The **source port** and **destination port** - identify the local end points of the connection
- Every byte on a TCP connection has its own 32-bit sequence number - a **byte stream** oriented connection
  - The **sequence number** field indicates the byte sequence number of the first data byte contained in that TCP data block
- TCP uses sliding window based flow control - the **acknowledgement number** specifies the next expected byte in order, which acknowledges the cumulative bytes that have been received by the receiver
  - ACK number 31245 means that the receiver has correctly received up to 31244 bytes and expecting for byte 31245

The TCP Protocol: The Header

- **HLEN:** The TCP header length specifies the number of 4-byte words contained in the TCP header
- The **window size** field tells how many bytes may be sent starting at the byte acknowledged
- **Checksum** checksums the whole segment (including header and data)
- The **urgent pointer** is used to indicate a byte offset from the current sequence number at which urgent data are to be found
- The **options field** was designed to provide a way to add extra facilities not covered by the regular header

TCP Segments

- The sending and receiving TCP entities exchange data in the form of **segments**
- A TCP segment consists of a fixed 20 byte header (plus an optional part) followed by zero or more data bytes
- A segment size is restricted by two parameters:
  - IP payload (65515 bytes)
  - Maximum Transfer Unit (MTU) of the link

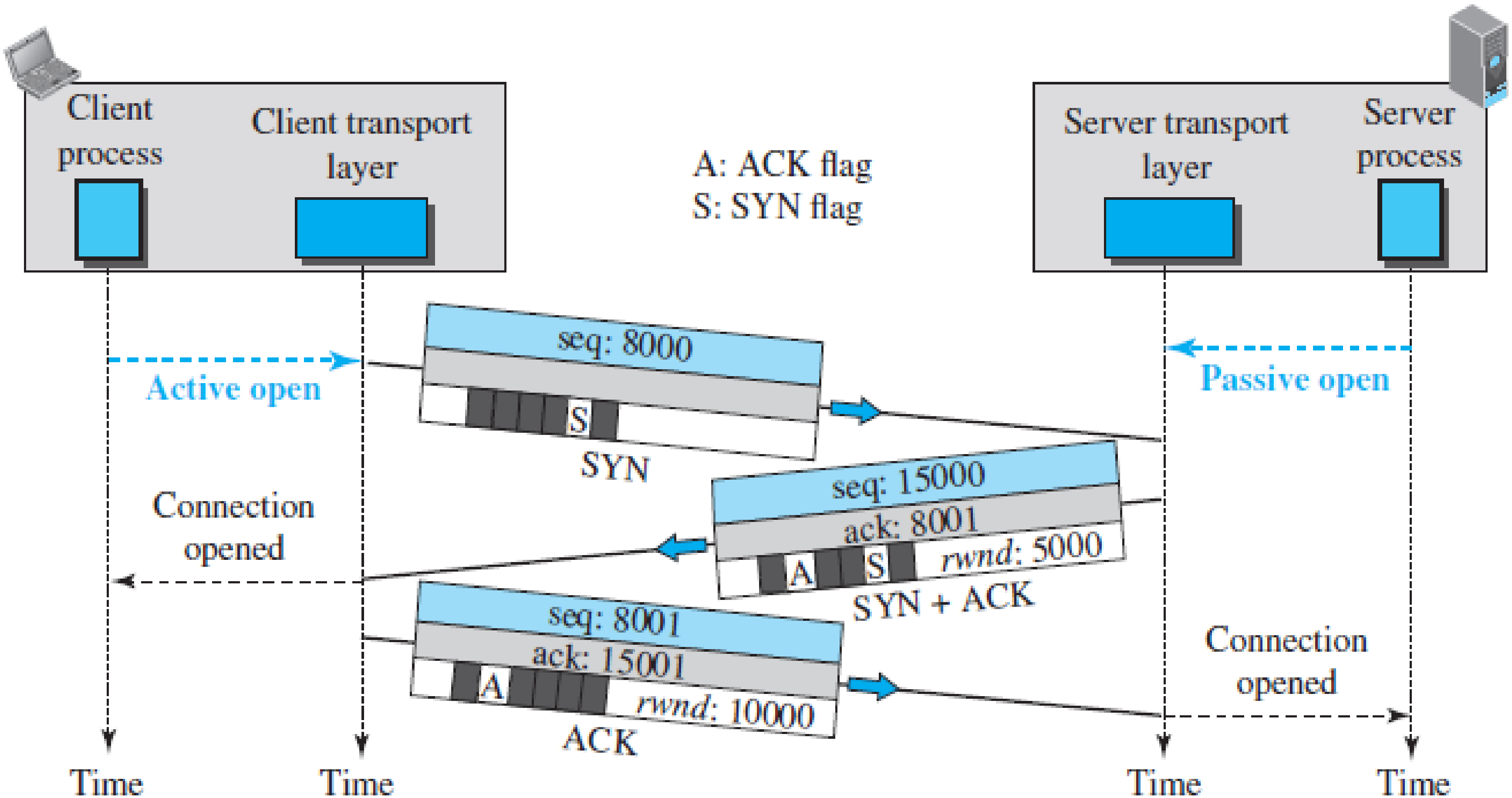
Window Size Field in the TCP Segment Header

- Flow control in TCP is handled using a variable sized sliding window
- The window size field tells how many bytes the receiver can receive based on the current free size at its buffer space
- **What is meant by window size 0?**
  - The receiver does not have a sufficient buffer space, so the sender should stop transmitting further data till it gets good amount of window size advertisement
- TCP acknowledgement – combination of acknowledgement number and window

TCP Connection Establishment

- TCP is connection-oriented
- TCP transmits data in full-duplex mode
- The connection establishment in TCP is called ***three-way handshaking***
- The process starts with the server. The server program tells its TCP that it is ready to accept a connection. This request is called a ***passive open***
- The client program issues a request for an ***active open***. A client that wishes to connect to an open server tells its TCP to connect to a particular server
- TCP can now start the three-way handshaking process

TCP Connection Establishment



Here, SYN segment is a control segment that carries no data. In SYN segment the initial sequence number is a random number. We can

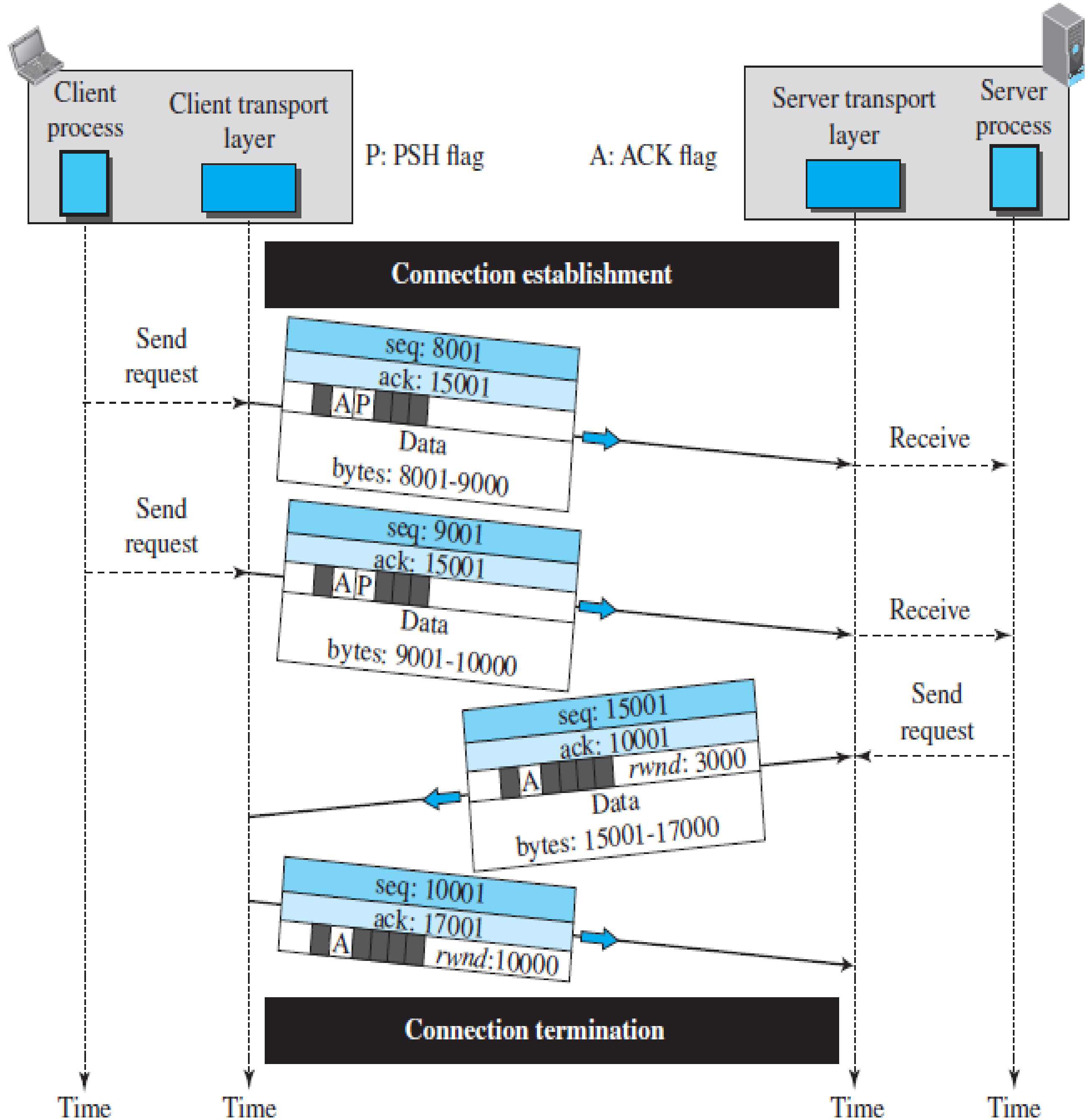
TCP Connection Establishment

- A SYN segment cannot carry data, but it consumes one sequence number

- A SYN + ACK segment cannot carry data, but it does consume one sequence number

- An ACK segment, if carrying no data, consumes no sequence number

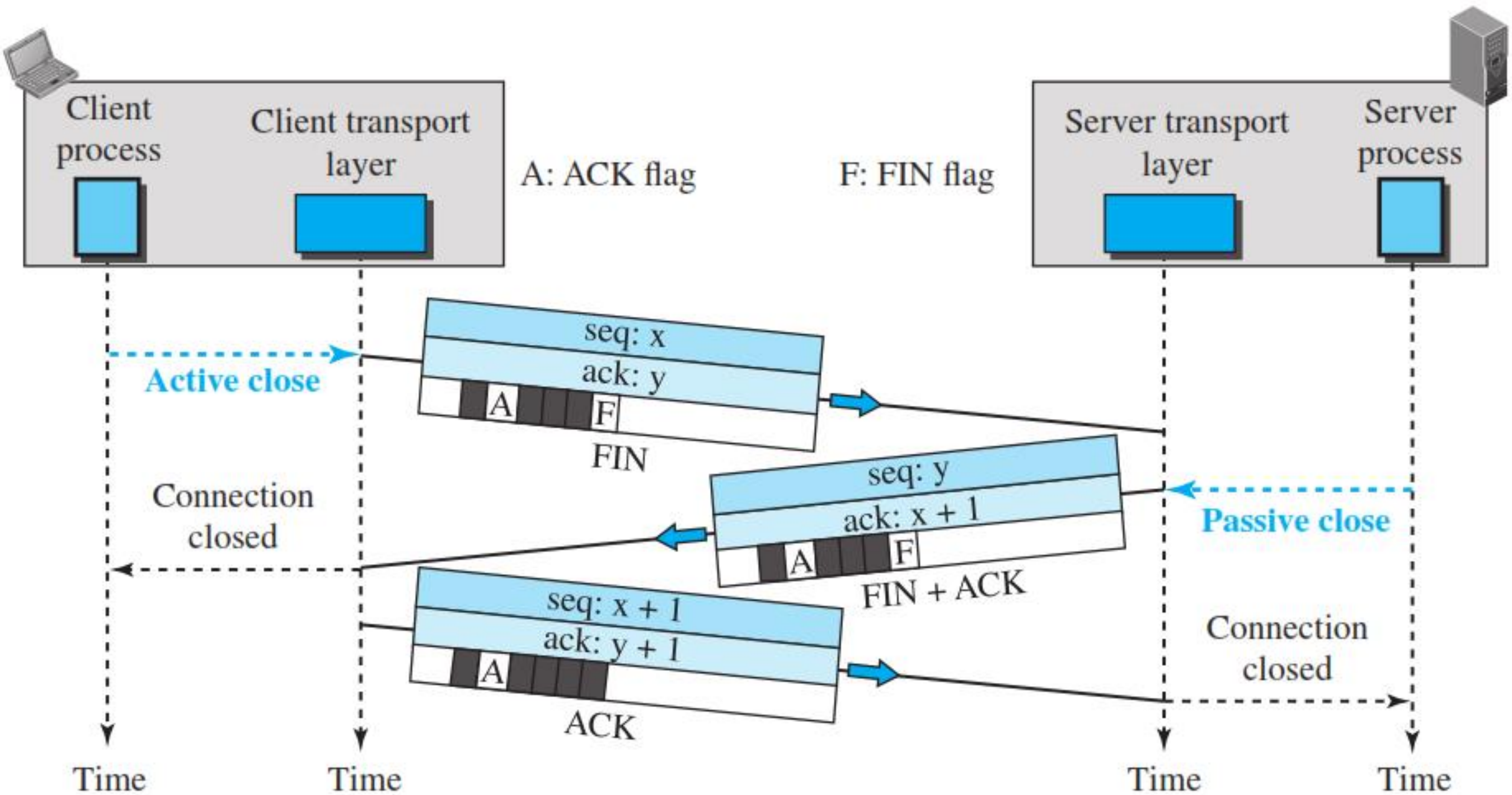
TCP Data transfer



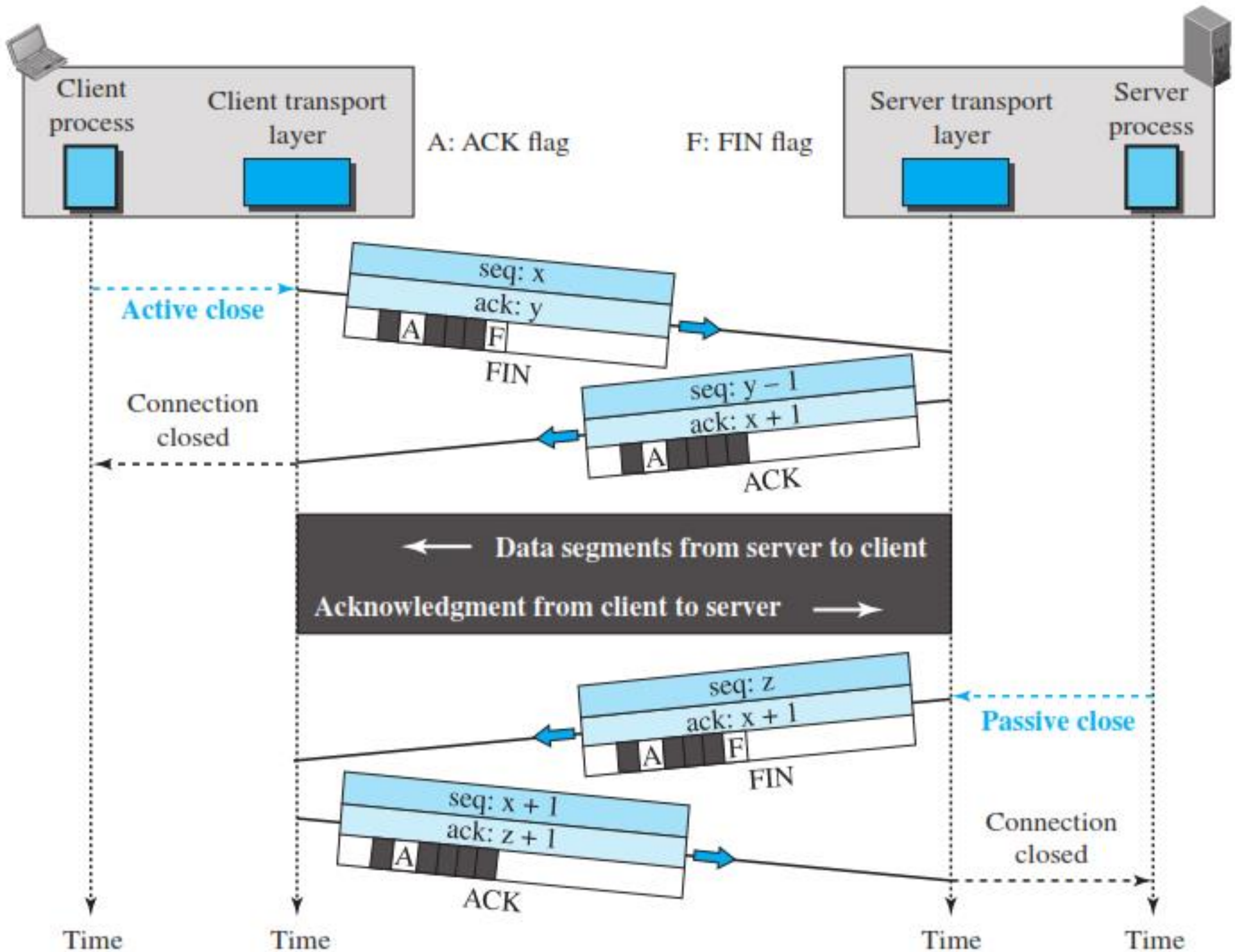
TCP Connection Termination

- Two options for connection termination:
  - Three-Way Handshaking
  - Half-Close
    - one end can stop sending data while still receiving data.

TCP Connection Termination



TCP Connection Termination : Half Close



TCP Connection Termination

- The FIN segment consumes one sequence number if it does not carry data.

- The FIN + ACK segment consumes only one sequence number if it does not carry data.

- ACK segment cannot carry data and consumes no sequence numbers.

User Datagram Protocol (UDP)

- UDP is a connectionless, unreliable transport protocol
- It does not add anything to the services of IP except to provide process-to-process communication

Features:

- Simple protocol
  - A wrapper on top of IP layer
- Fast
  - No flow control, no congestion control
- Provides connectionless services

User Datagram Protocol (UDP)

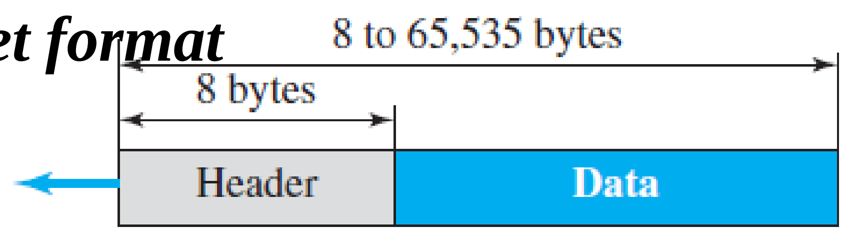
Uses:

- Provide performance
  - No data holding in buffer like TCP
- Have no overhead
- Suitable for short messages

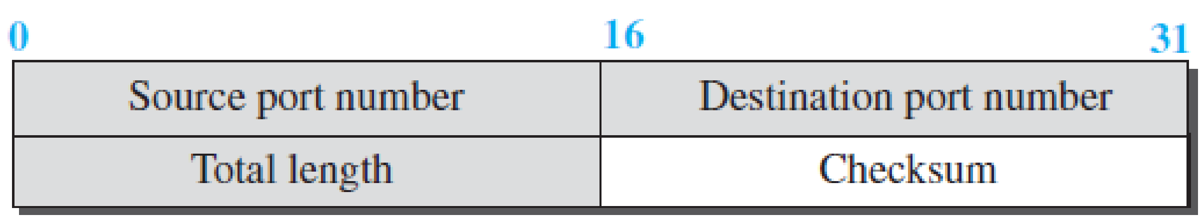
UDP: User Datagram

- A packet in UDP is called ***user datagram***
- UDP has a fixed-size header of 8 bytes
- **Source and destination port numbers:** Identify the local end points of the connection
- **Length:** Defines the total length of the UDP **segment** (header plus data)
- **Checksum:** It checksums the whole segment

User datagram packet format



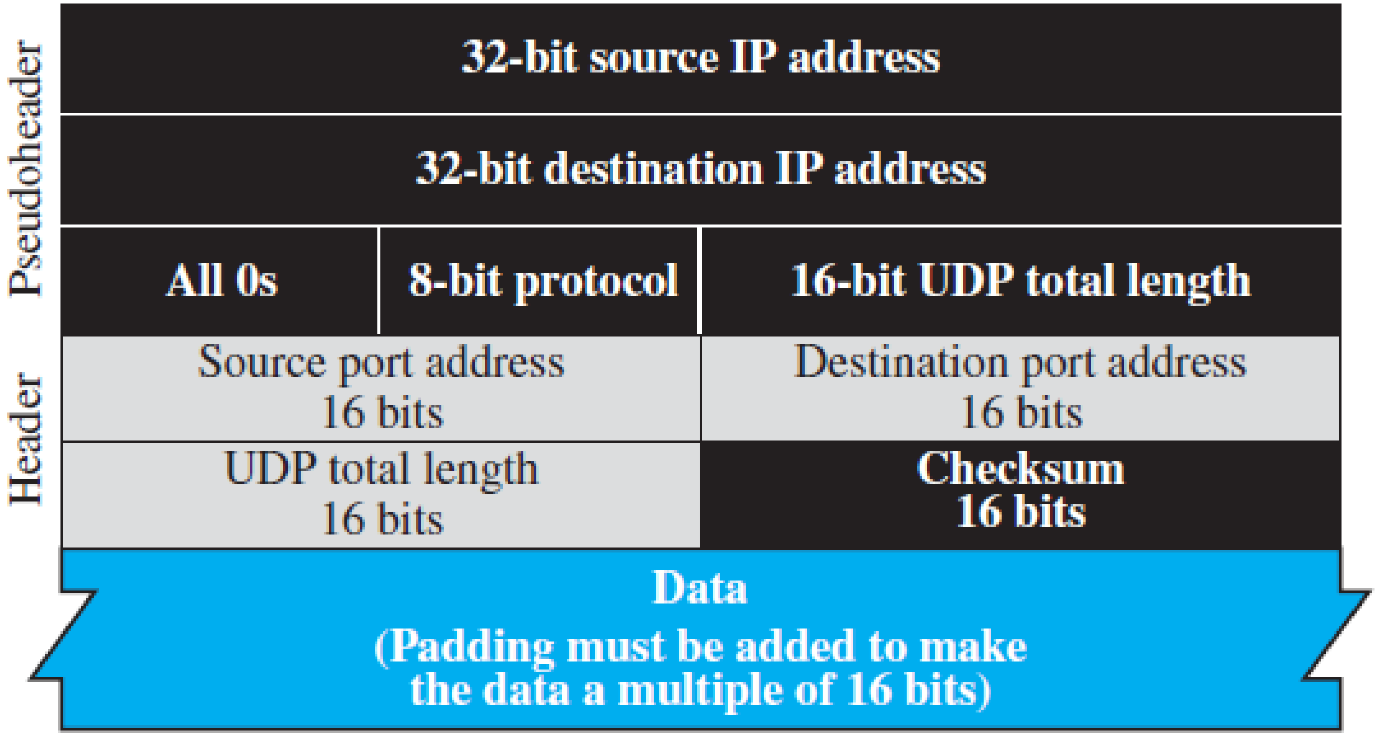
a. UDP user datagram



b. Header format

UDP uses pseudoheader for checksum calculation

- UDP checksum calculation includes three sections: a pseudoheader, the UDP header, and the data coming from the application layer
- The he
  IP



UDP Applications

Protocol	Keyword	Comment
DNS	domain	Simple request response messaging system is faster than TCP
BOOTP/DHCP	Network configuration	Short messaging helps faster configuration
TFTP	File transfer	Lightweight file transfer protocol to transfer small files
SNMP	Network management	Simple UDP protocol easily cut through congestion than TCP
QUIC	Advance transport protocol	UDP provide direct access to IP while TCP can't

TFTP: Trivial File Transfer Protocol

BOOTP: Bootstrap Protocol

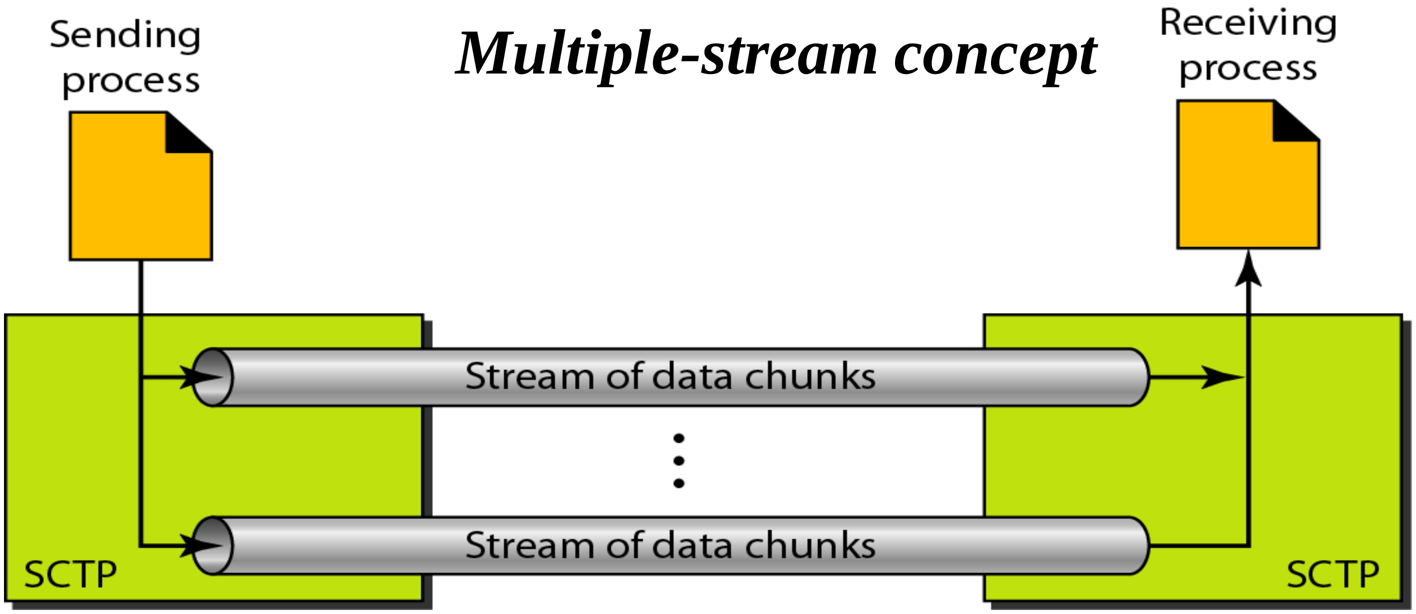
Stream Control Transmission Protocol (SCTP)

- SCTP is a new transport-layer protocol
- It is designed to combine some features of UDP and TCP in an effort to create a better protocol for multimedia communication
- SCTP is a message-oriented, reliable protocol

SCTP Services

- SCTP, like UDP or TCP, provides process-to-process communication
- **Multiple Streams:** SCTP allows **multistream service** in each connection, which is called **association** in SCTP terminology
  - If one of the streams is blocked, the other streams can still deliver their data

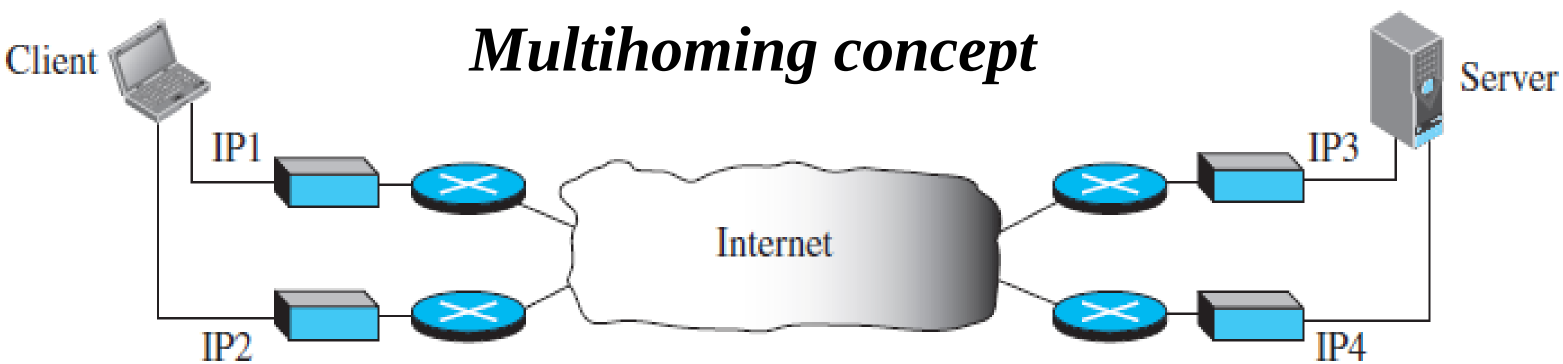
Multiple-stream concept



SCTP Services

- **Multihoming:** An SCTP association, supports **multihoming service.** The sending and receiving host can define multiple IP addresses in each end for an association i.e. SCTP association allows multiple IP addresses for each end
  - When one path fails, other interface can be used for data delivery without interruption

Multihoming concept



SCTP Services

- SCTP offers full-duplex service
- SCTP is a connection-oriented protocol. However, in SCTP, a connection is called an *association*
- SCTP, like TCP, is a reliable transport protocol. It uses an ACK mechanism to check the safe and sound arrival of data

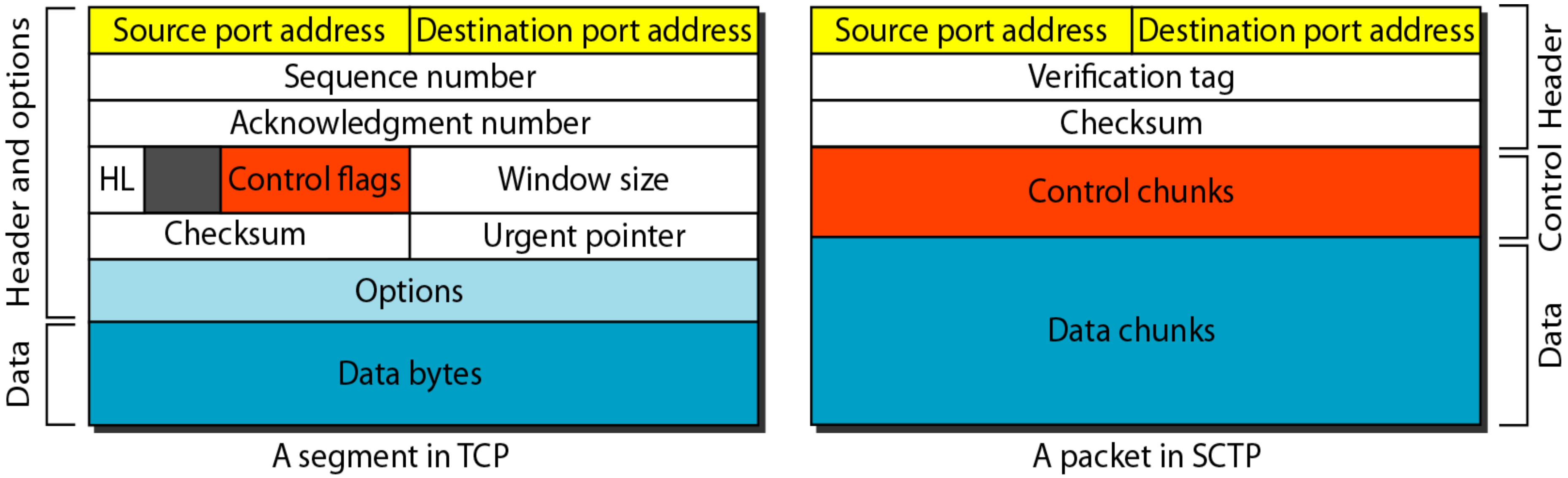
SCTP Features

- **Transmission Sequence Number (TSN):**
  - The unit of data in SCTP is a data chunk
  - In SCTP, a data chunk is numbered using a TSN. It is analogous to sequence number in TCP
- **Stream Identifier (SI):**
  - In SCTP, there may be several streams in each association
  - To distinguish between different streams, SCTP uses an SI
  - Each data chunk must carry the SI in its header so that when it arrives at the destination, it can be properly placed in its stream
- **Stream Sequence Number (SSN):**
  - To distinguish between different data chunks belonging to the same stream, SCTP uses SSNs

SCTP Features

- **Packets:**
  - TCP has segments; SCTP has packets
  - In SCTP, data are carried as data chunks, control information as control chunks. Several control chunks and data chunks can be packed together in a packet

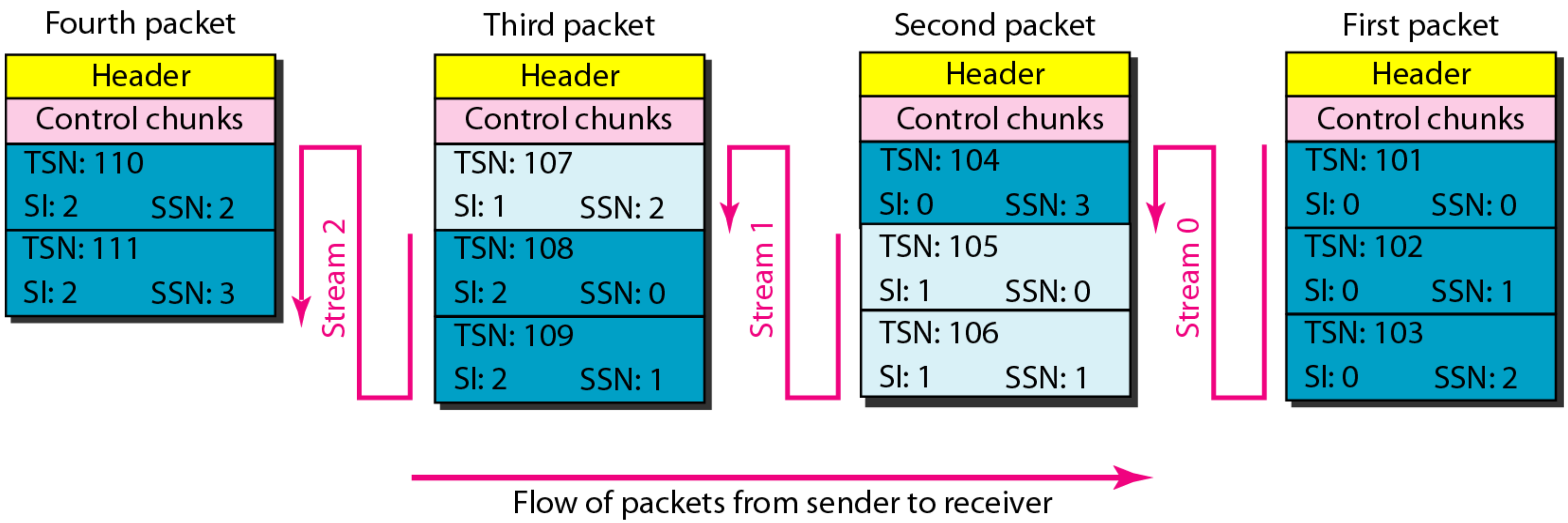
Comparison between a TCP segment and



SCTP Features

- **Packets: (Contd…)**
  - In SCTP, we have data chunks, streams, and packets
  - An association may send many packets, a packet may contain several chunks, and chunks may belong to different streams

Packets, data chunks, and streams

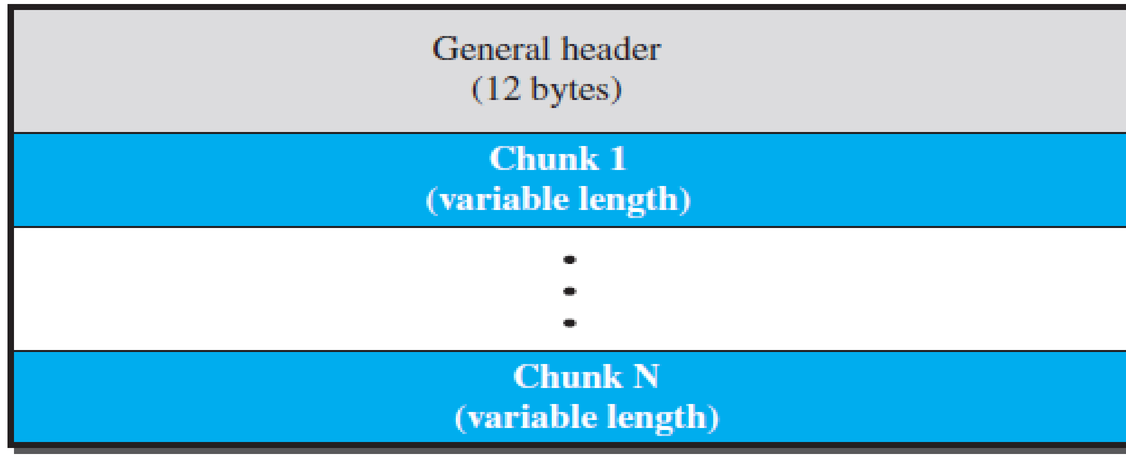


- In SCTP, acknowledgment numbers are used to acknowledge only data chunks; control chunks are acknowledged by other control chunks if necessary

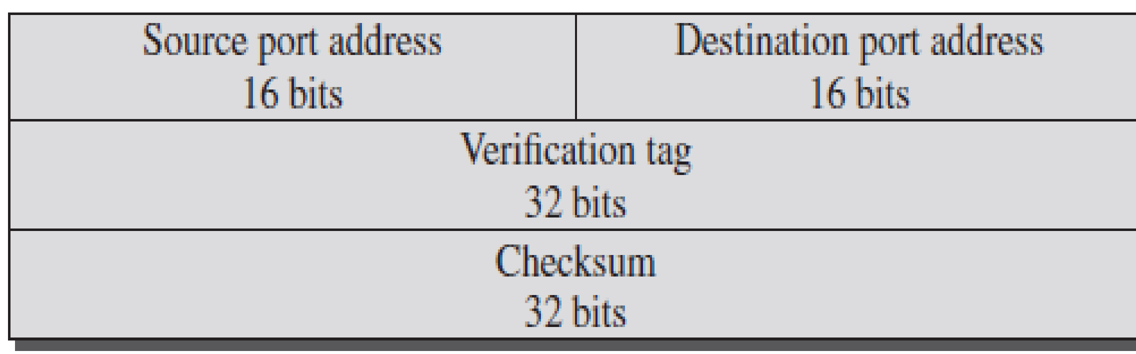
SCTP Packet Format

- In an SCTP packet, control chunks come before data chunks
- A connection in SCTP is called an association
- In header, the verification tag is a 32-bit field that matches a packet to an association
  - It serves as an identifier for the association
  - It is repeated in every packet during the association

SCTP packet format



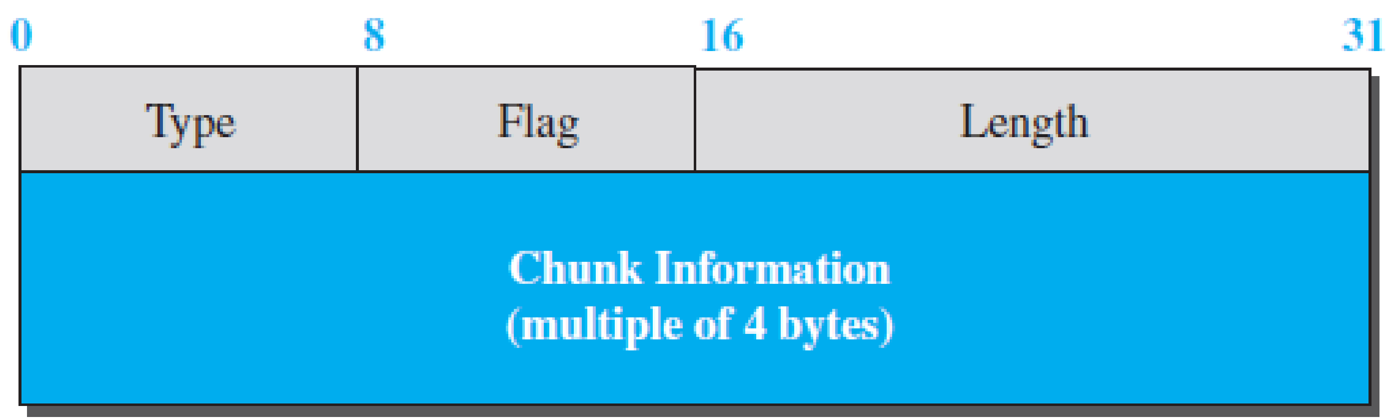
General header



SCTP Packet Format

- **Chunks:**
  - Control information or user data are carried in chunks
  - Chunks have a common layout
  - The first three fields are common to all chunks
  - The type field can define up to 256 types of chunks
  - Flag field defines special flags that a particular chunk may need
  - The length field defines the total size of the chunk, in bytes, including the type, flag, and length fields

Common layout of a chunk



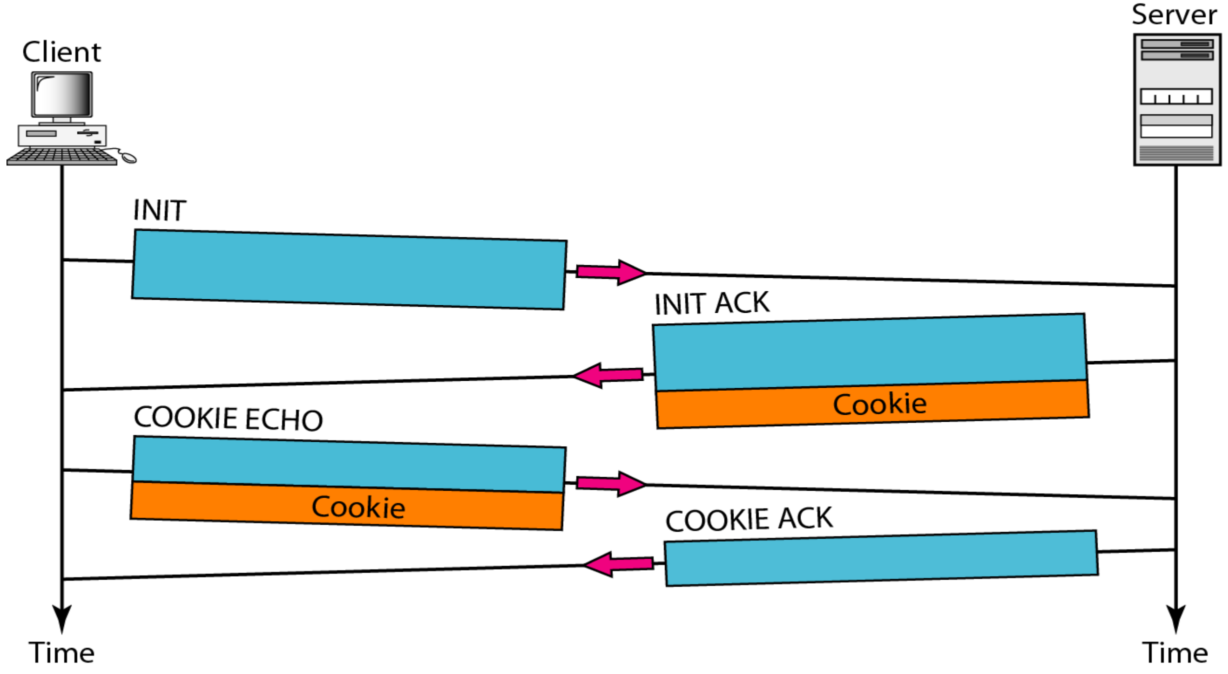
Types of Chunks

Type	*Chunk*	*Description*
0	DATA	User data
1	INIT	Sets up an association
2	INIT ACK	Acknowledges INIT chunk
3	SACK	Selective acknowledgment
4	HEARTBEAT	Probes the peer for liveliness
5	HEARTBEAT ACK	Acknowledges HEARTBEAT chunk
6	ABORT	Aborts an association
7	SHUTDOWN	Terminates an association
8	SHUTDOWN ACK	Acknowledges SHUTDOWN chunk
9	ERROR	Reports errors without shutting down
10	COOKIE ECHO	Third packet in association establishment
11	COOKIE ACK	Acknowledges COOKIE ECHO chunk
14	SHUTDOWN COMPLETE	Third packet in association termination
192	FORWARD TSN	For adjusting cumulating TSN

An SCTP Association

- A connection in SCTP is called an *association*
- **Association Establishment:** Association establishment in SCTP requires a four-way handshake

Four-way handshaking

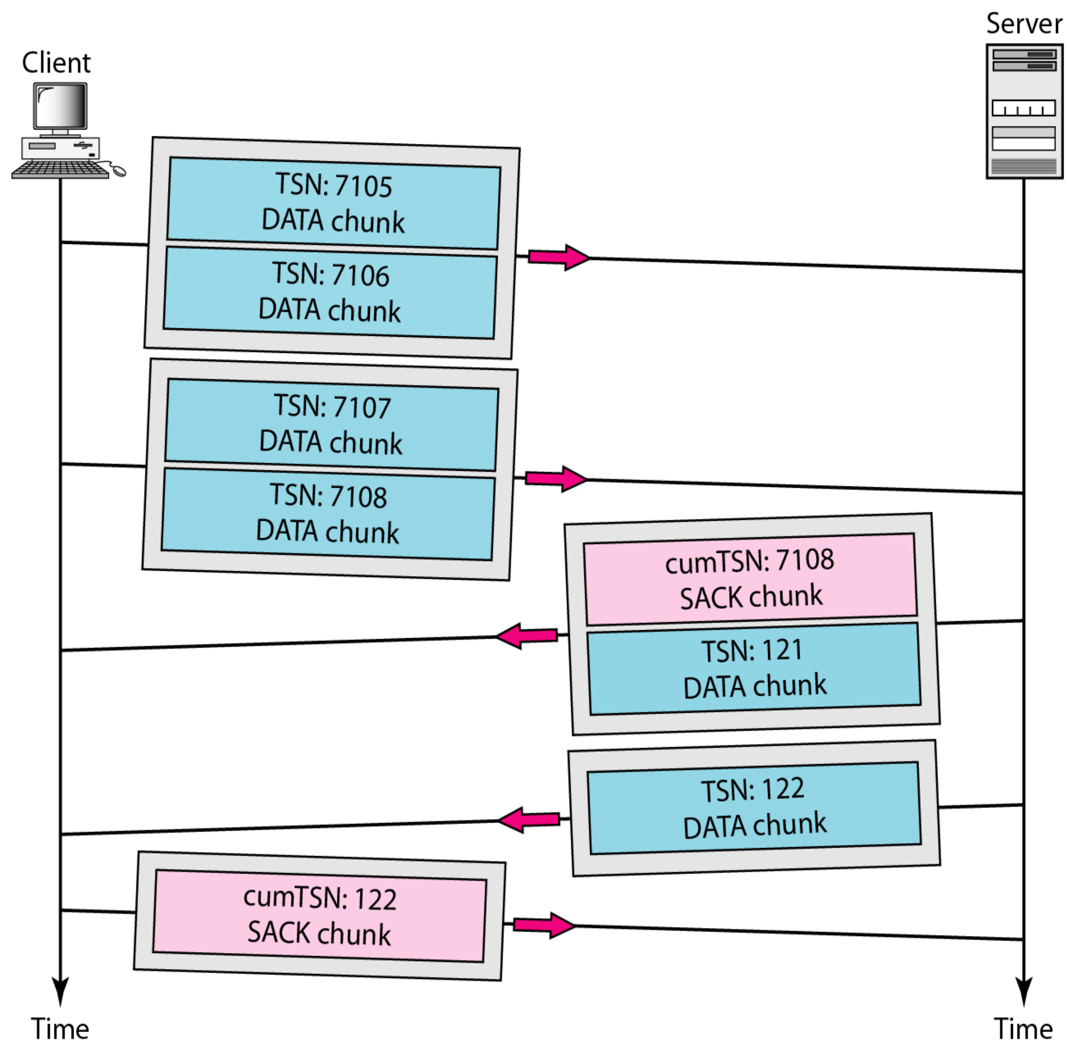


Note: No other chunk is allowed in a packet carrying an INIT or INIT ACK chunk. A COOKIE ECHO or a COOKIE ACK chunk can carry data chunks

An SCTP Association

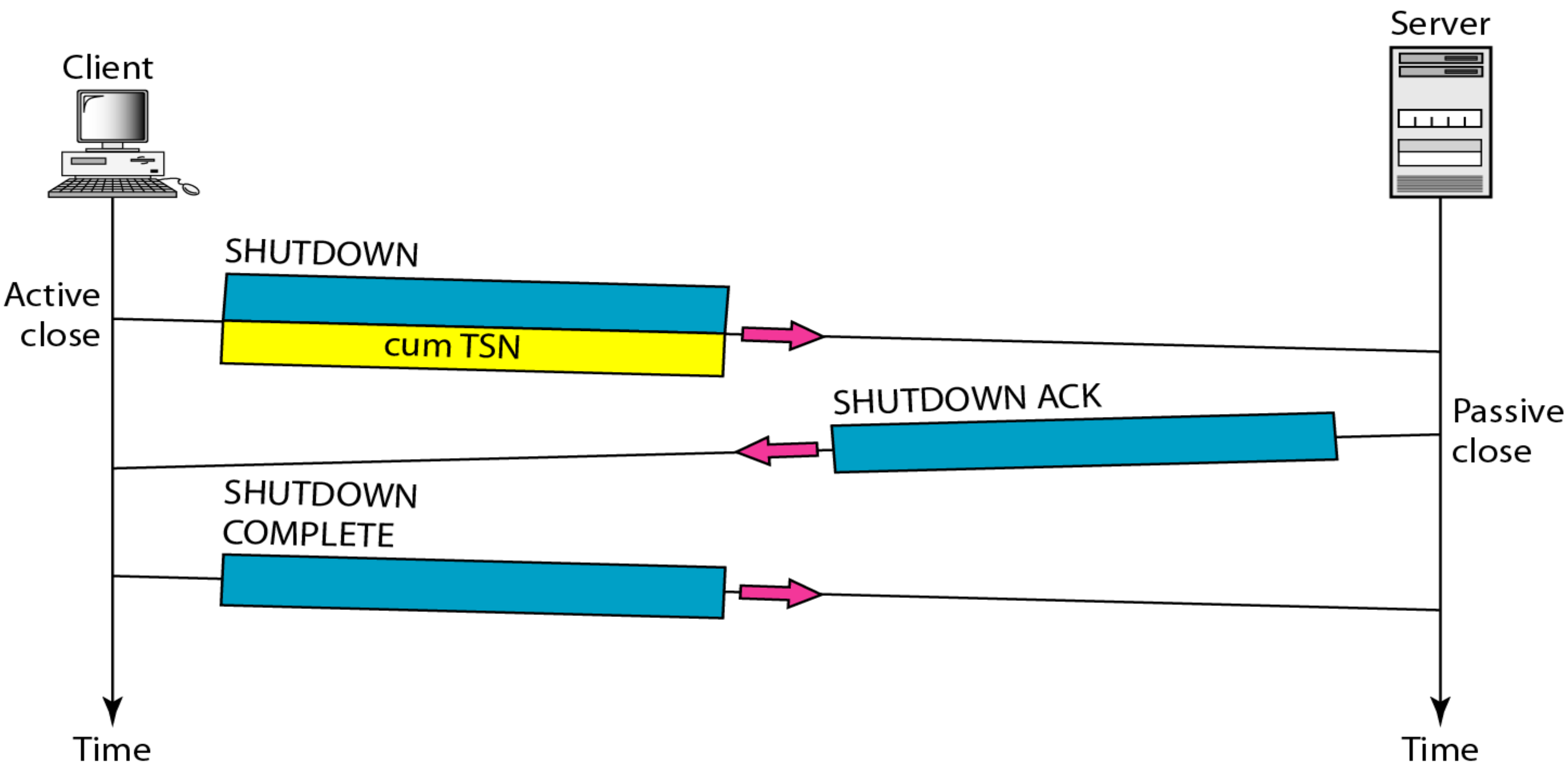
Data Transfer:

- The ACK in SCTP defines the cumulative TSN, the TSN of the last data chunk received in order
- SCTP preserves the boundaries of the message when creating a DATA chunk from a message
- The size of the message does not exceed the MTU of the path



An SCTP Association

Association Termination:



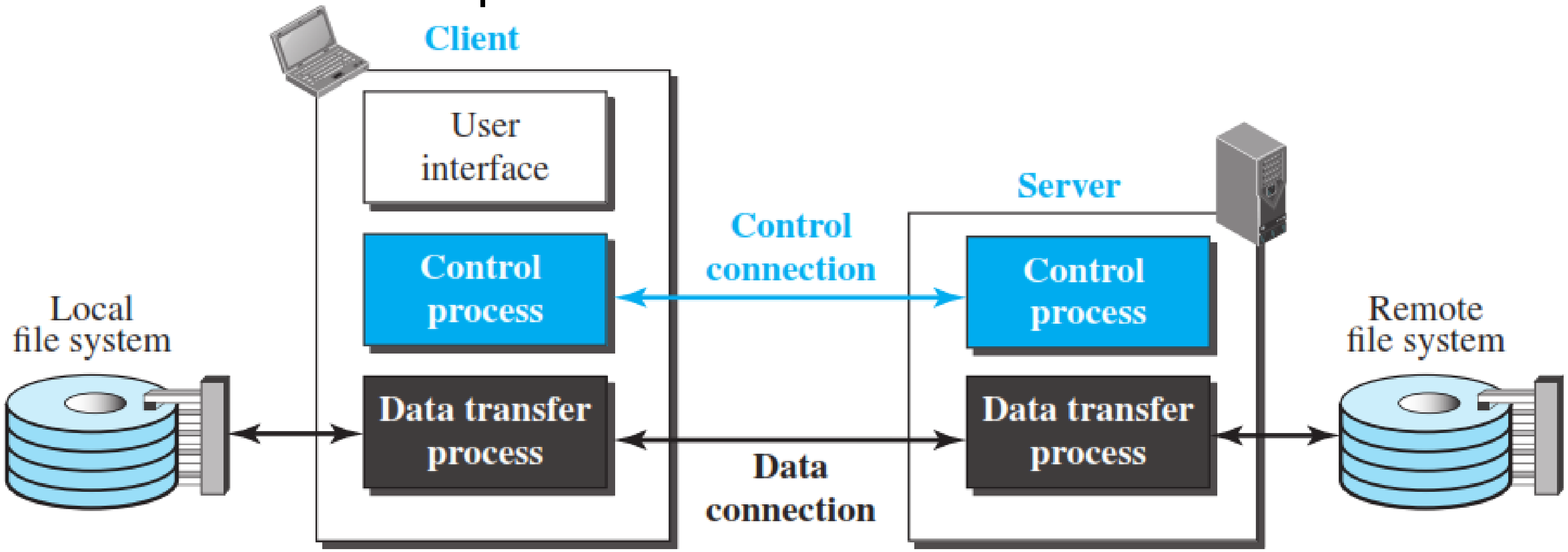
Application Layer

Application Layer

- The application layer provides services to the user.
- The protocols in this layer do not provide services to any other protocols.
- Only receive services from the protocols in the transport layer.
- The application-layer protocols can be both standard and nonstandard.
- Application-Layer Paradigms
  - Client-Server paradigm
  - Peer-to-Peer paradigm
  - Mixed Paradigm

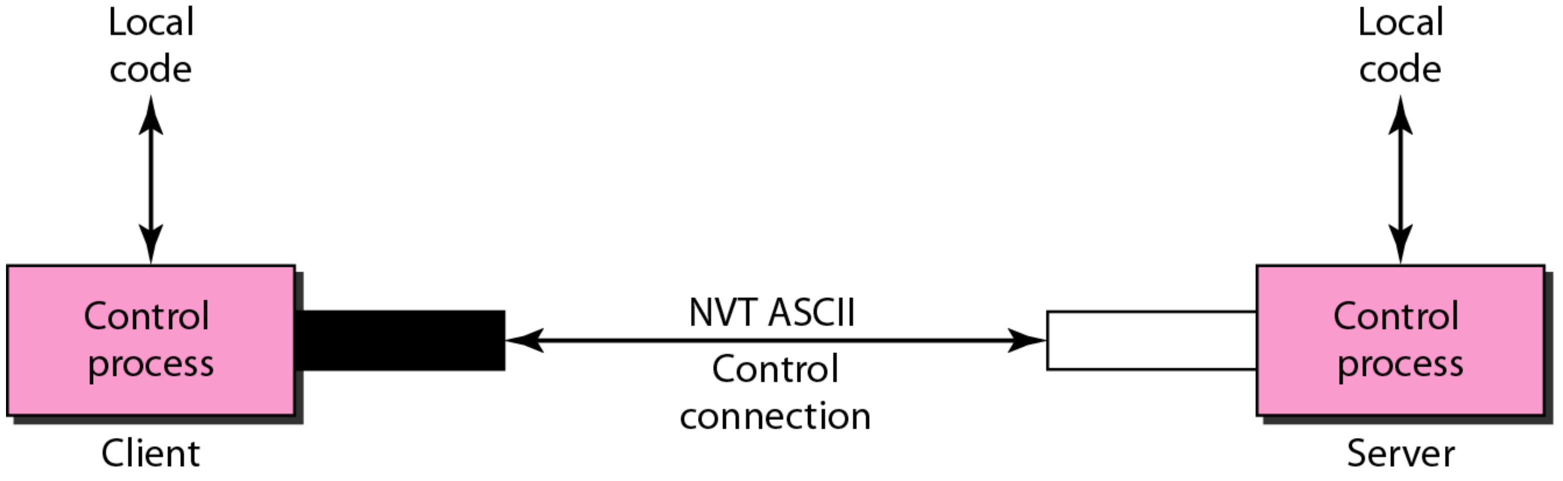
File Transfer Protocol (FTP)

- A standard protocol for transferring files from one computer to another.
- FTP uses the services of TCP.
- It needs two TCP connections.
- The well-known port 21 is used for the control connection.
- The well-known port 20 for the data connection.



FTP : Control Connection

- The control connection remains connected during the entire interactive FTP session
- It uses the NVT ASCII character set.
- Commands are sent from the client to the server.
- Responses are sent from the server to the client.
- A response has two parts: a three-digit number followed by text.



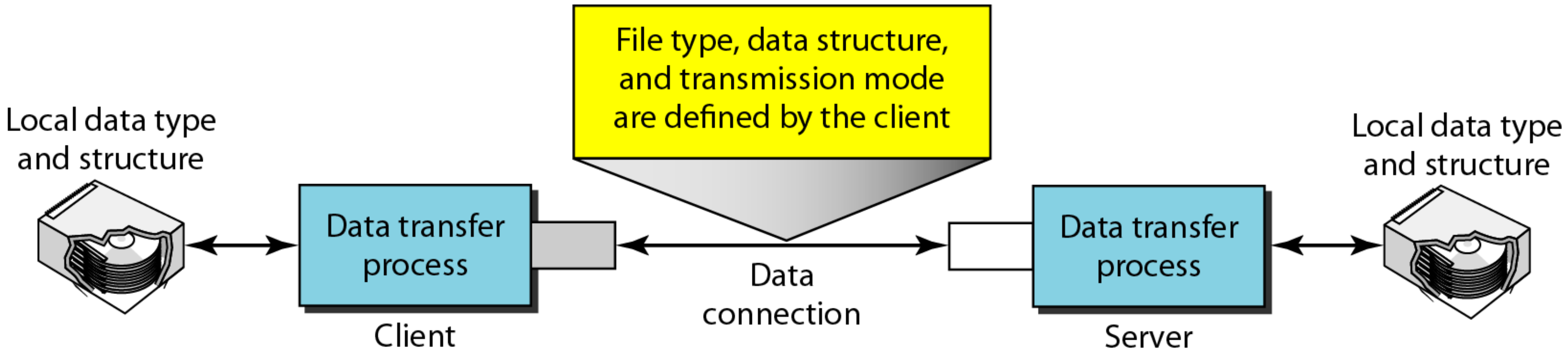
Command	*Argument(s)*	*Description*
ABOR		Abort the previous command
CDUP		Change to parent directory
CWD	Directory name	Change to another directory
DELE	File name	Delete a file
LIST	Directory name	List subdirectories or files
MKD	Directory name	Create a new directory
PASS	User password	Password
PASV		Server chooses a port
PORT	Port identifier	Client chooses a port
PWD		Display name of current directory
QUIT		Log out of the system
RETR	File name(s)	Retrieve files; files are transferred from server to client
RMD	Directory name	Delete a directory
RNFR	File name (old)	Identify a file to be renamed
RNTO	File name (new)	Rename the file
STOR	File name(s)	Store files; file(s) are transferred from client to server
STRU	**F**, **R**, or **P**	Define data organization (**F**: file, **R**: record, or **P**: page)
TYPE	**A**, **E**, **I**	Default file type (**A**: ASCII, **E**: EBCDIC, **I**: image)
USER	User ID	User information
MODE	**S**, **B**, or **C**	Define transmission mode (**S**: stream, **B**: block, or **C**: compressed

FTP : Control Connection

Code	*Description*	*Code*	*Description*
125	Data connection open	**250**	Request file action OK
150	File status OK	**331**	User name OK; password is needed
200	Command OK	**425**	Cannot open data connection
220	Service ready	**450**	File action not taken; file not available
221	Service closing	**452**	Action aborted; insufficient storage
225	Data connection open	**500**	Syntax error; unrecognized command
226	Closing data connection	**501**	Syntax error in parameters or arguments
230	User login OK	**530**	User not logged in

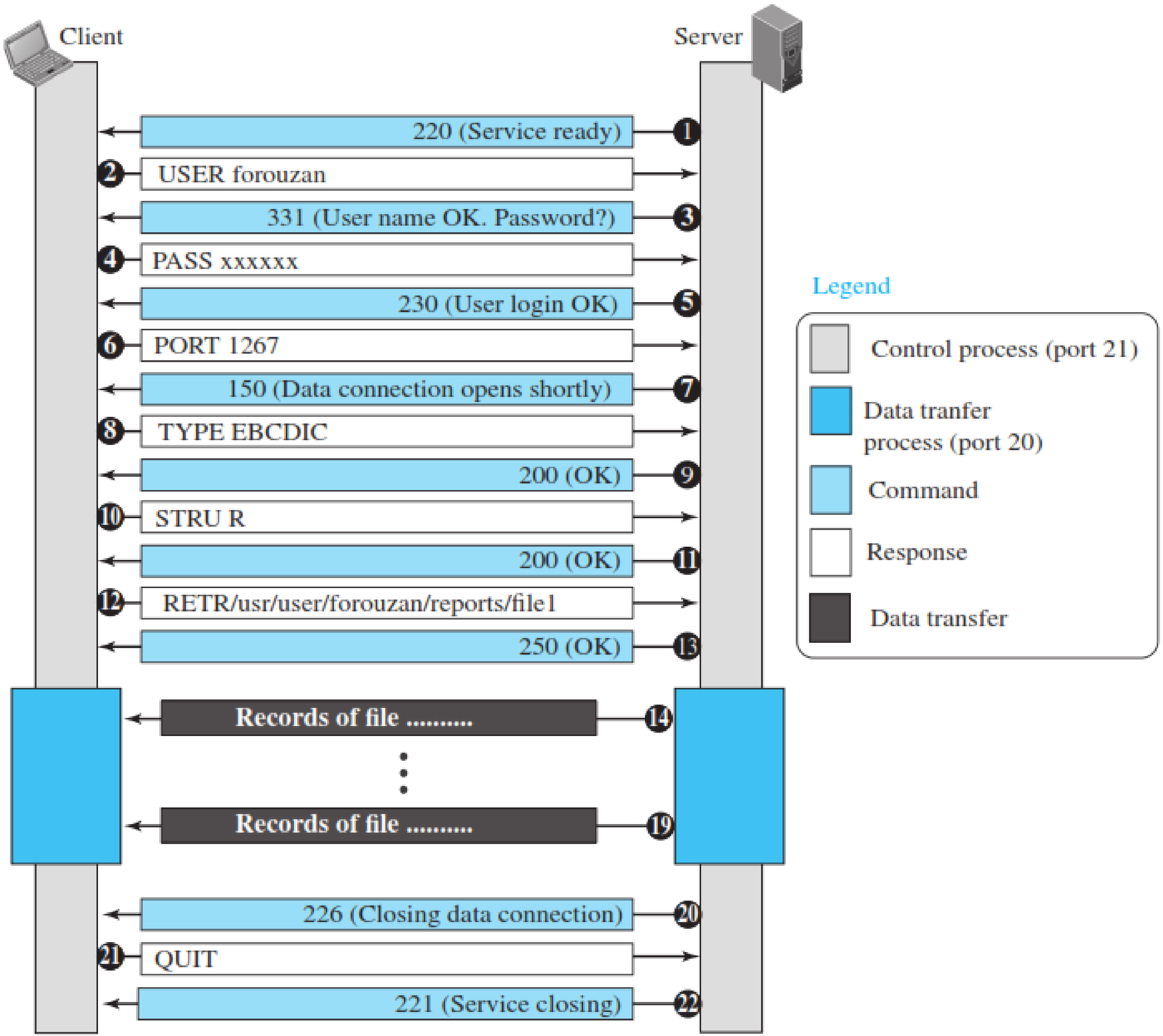
FTP : Data Connection

- The data connection is opened and then closed for each file transfer activity.
- The creation of a data connection
  - The client issues a passive open using an ephemeral port.
  - Using the PORT command the client sends this port number to the server.
  - Server issues an active open using the well-known port 20 and the received ephemeral port number.



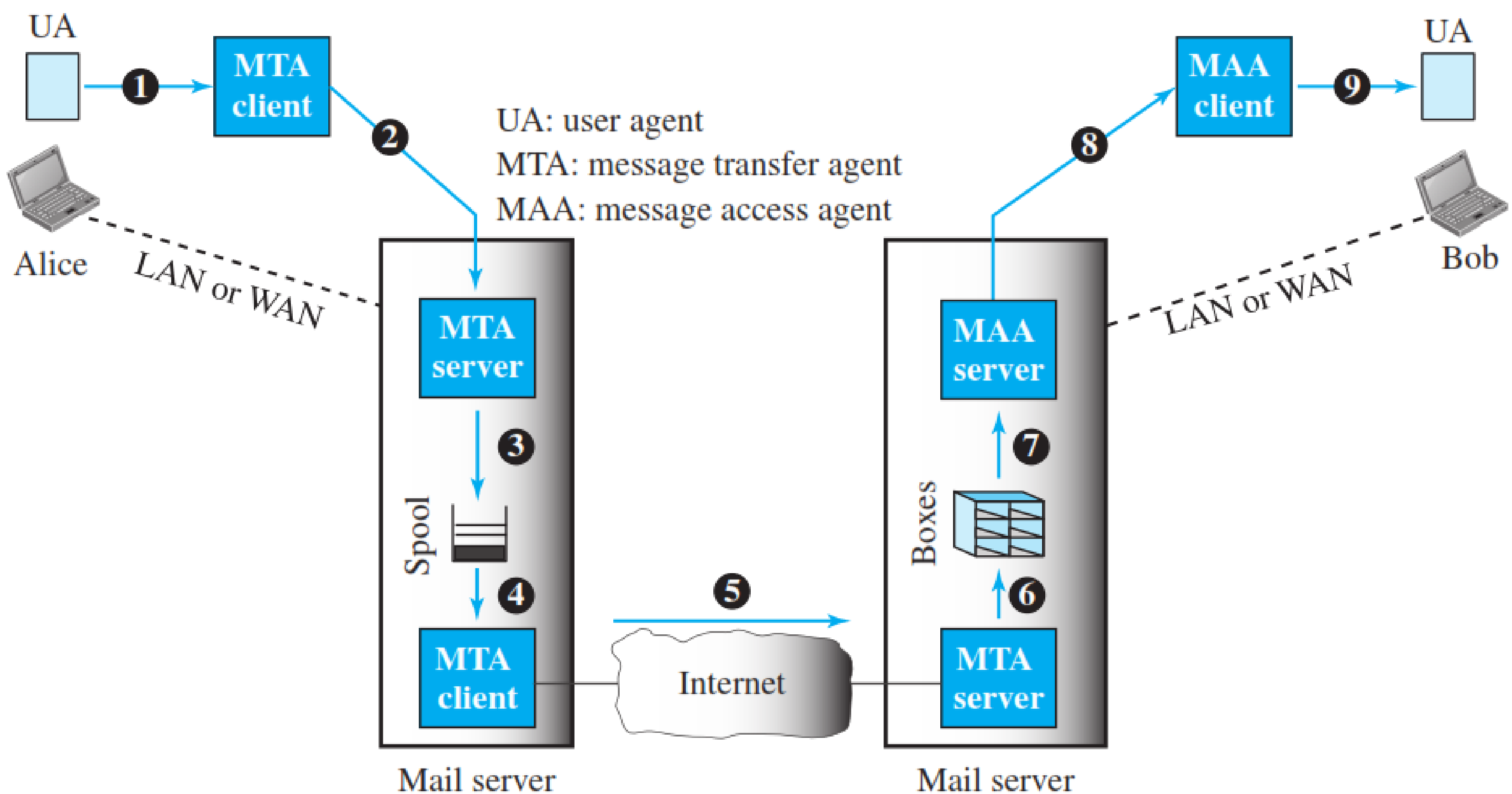
FTP : Data Connection

- The client must define
  - File Type
    - ASCII file
    - EBCDIC file
    - image file
  - Data Structure
    - File structure
    - Record structure
    - Page structure
  - Transmission Mode
    - Stream mode
    - Block mode


Client
Server
220 (Service ready)
USER forouzan
331 (User name OK. Password?)
PASS xxxxxx
230 (User login OK)
PORT 1267
150 (Data connection opens shortly)
TYPE EBCDIC
200 (OK)
STRU R
200 (OK)
RETR/usr/user/forouzan/reports/file1
250 (OK)
Records of file ....
Records of file ....
226 (Closing data connection)
QUIT
221 (Service closing)
Legend
Control process (port 21)
Data tranfer process (port 20)
Command
Response
Data transfer

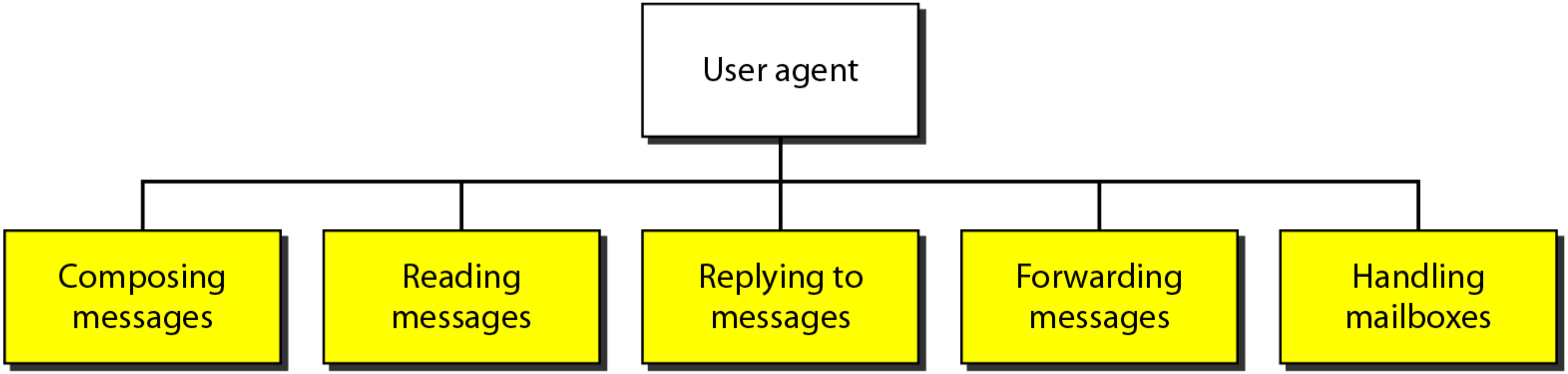
ELECTRONIC MAIL (E-MAIL)

- One of the most popular Internet services is e-mail.
- Its architecture consists of several components.

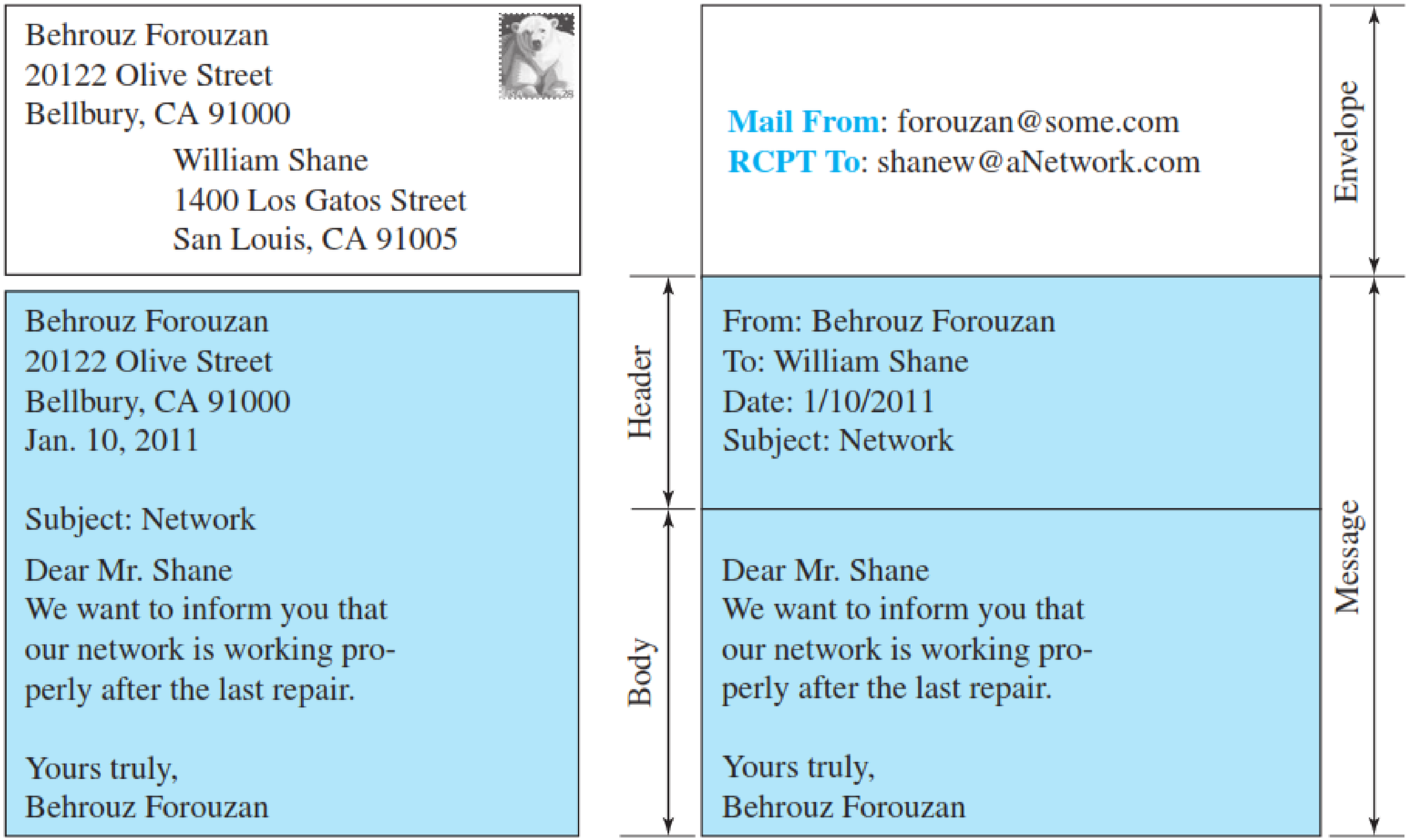


E-MAIL : User Agent

- It provides service to the user to make the process of sending and receiving a message.
- Service provided by User Agent



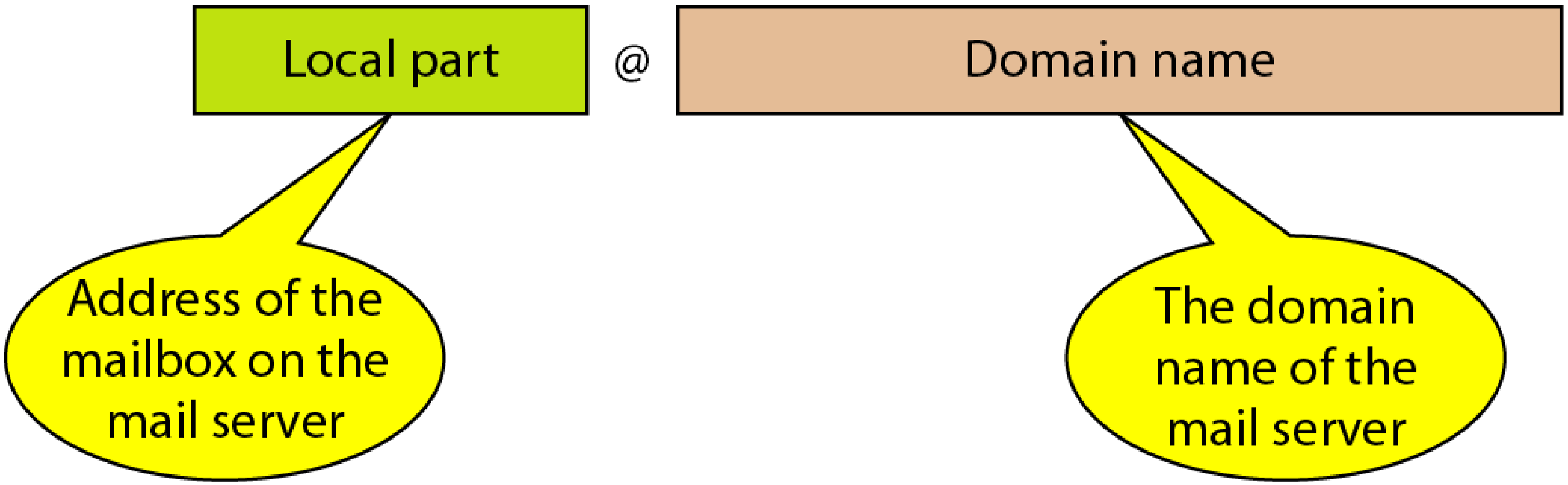
Format of E-mail



Postal mail

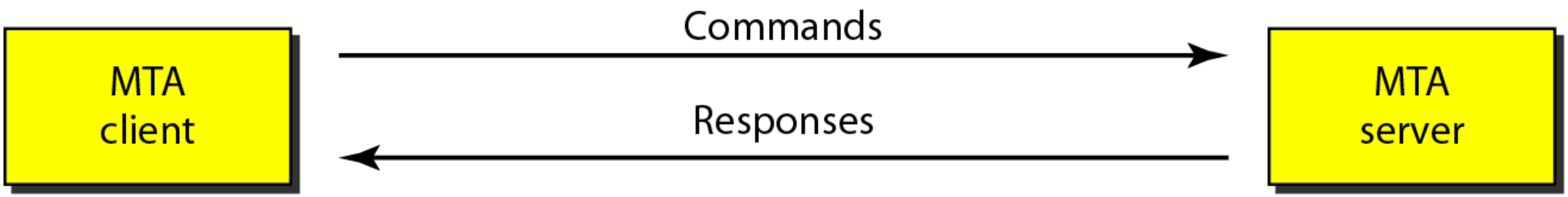
Electronic mail

Format of E-mail Address



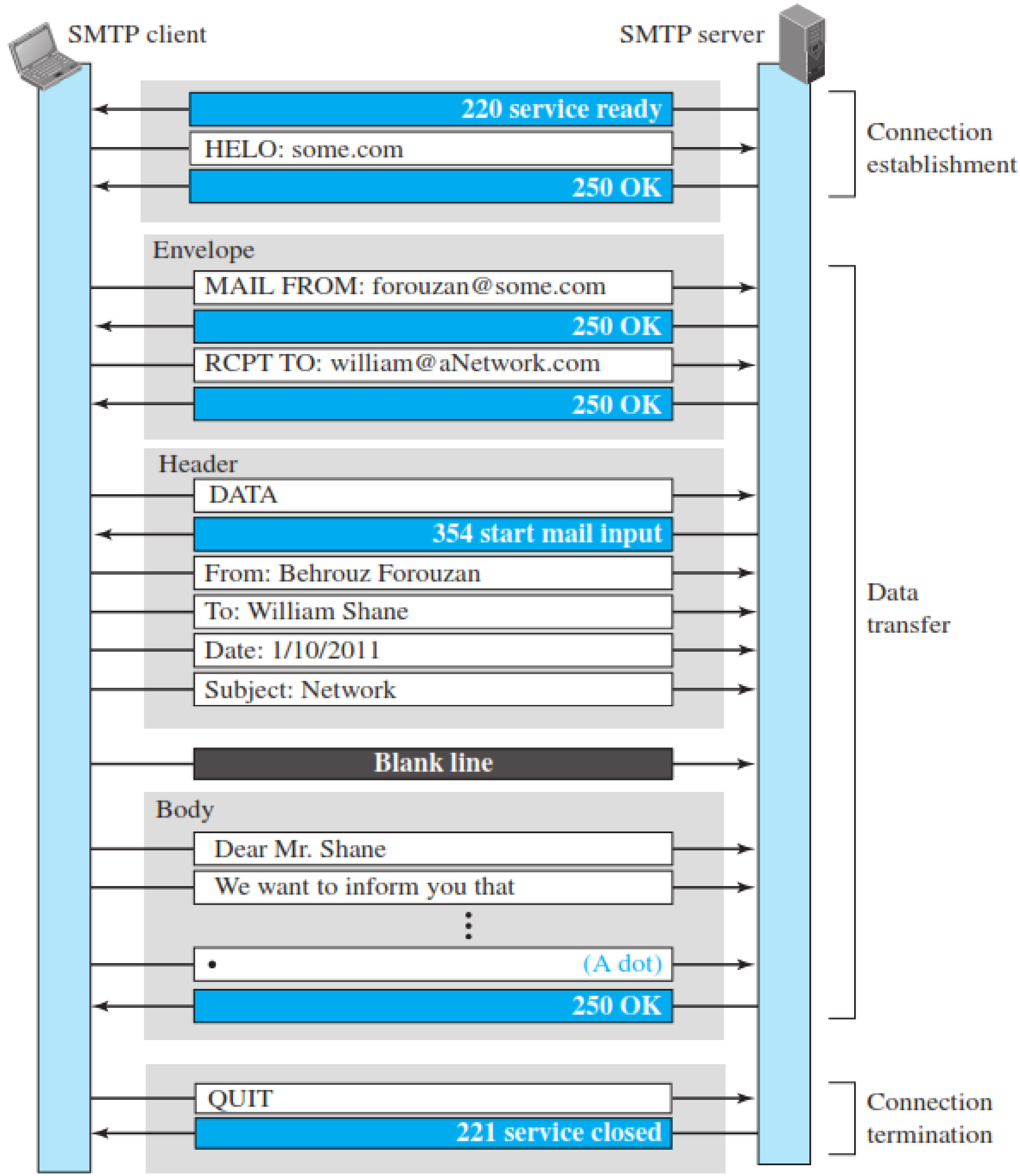
Message Transfer Agent: SMTP

- The formal protocol for MTA client and server is called Simple Mail Transfer Protocol (SMTP).
- SMTP uses commands and responses to transfer messages.



Message Transfer Agent: SMTP

- The process of transferring a mail message occurs in three phases:
  - Connection establishment
  - Mail transfer
  - Connection termination


SMTP client
SMTP server
220 service ready
HELO: some.com
250 OK
Connection establishment
Envelope
MAIL FROM: forouzan@some.com
250 OK
RCPT TO: william@aNetwork.com
250 OK
Header
DATA
354 start mail input
From: Behrouz Forouzan
To: William Shane
Date: 1/10/2011
Subject: Network
Blank line
Body
Dear Mr. Shane
We want to inform you that
(A dot)
250 OK
Data transfer
QUIT
221 service closed
Connection termination

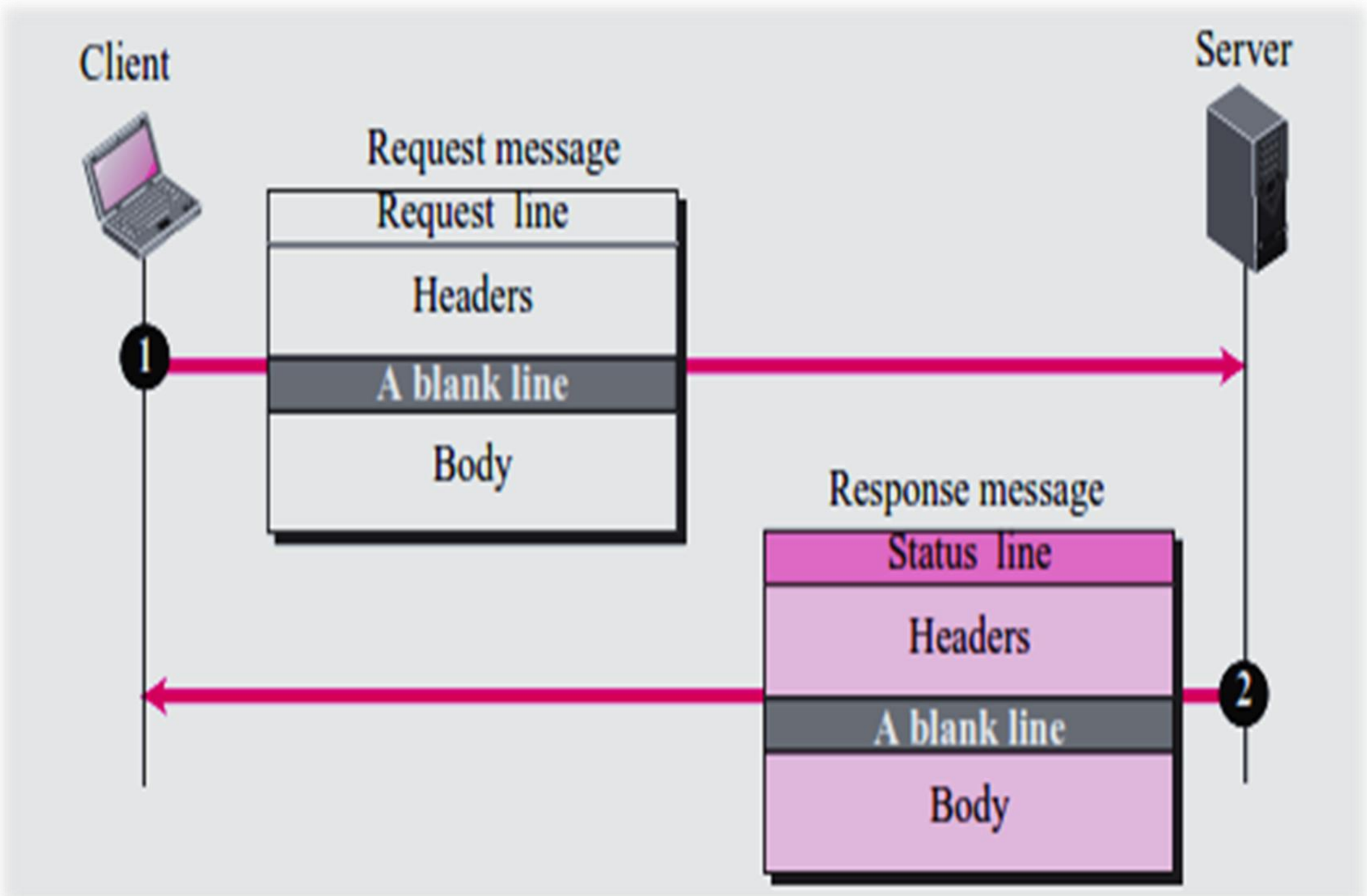
Message Access Agent: POP and IMAP

- The client must pull messages from the server.
- Currently two message access protocols are available:
  - Post Office Protocol, version 3 (POP3)
  - Internet Mail Access Protocol, version 4 (IMAP4)

Hypertext Transfer Protocol (HTTP)

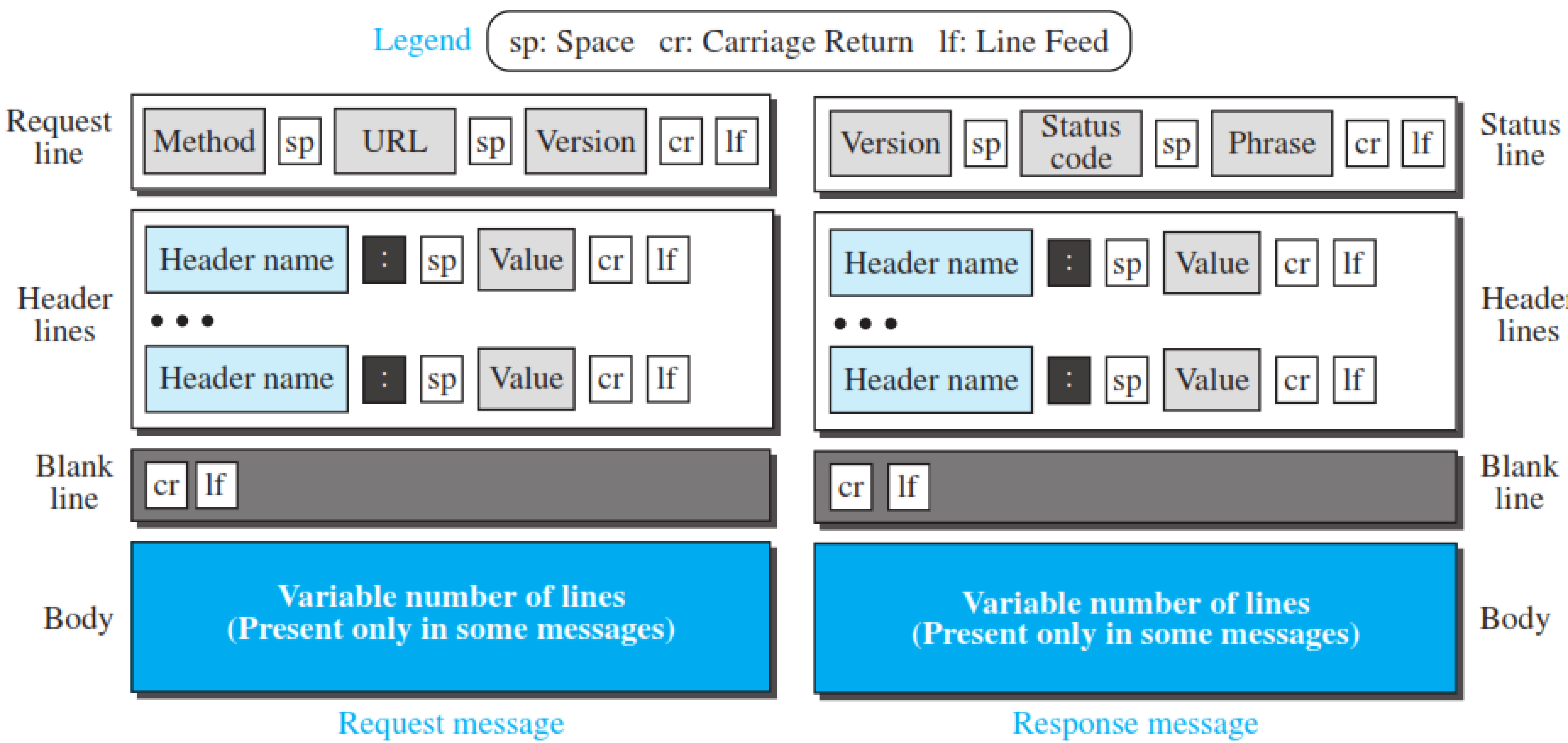
- HTTP is a protocol used mainly to access data on the World Wide Web.
- HTTP uses the services of TCP on well-known port 80.
- The client initializes a HTTP transaction by sending a request.
- The server replies by sending a response.

Hypertext Transfer Protocol (HTTP)



Hypertext Transfer Protocol (HTTP)

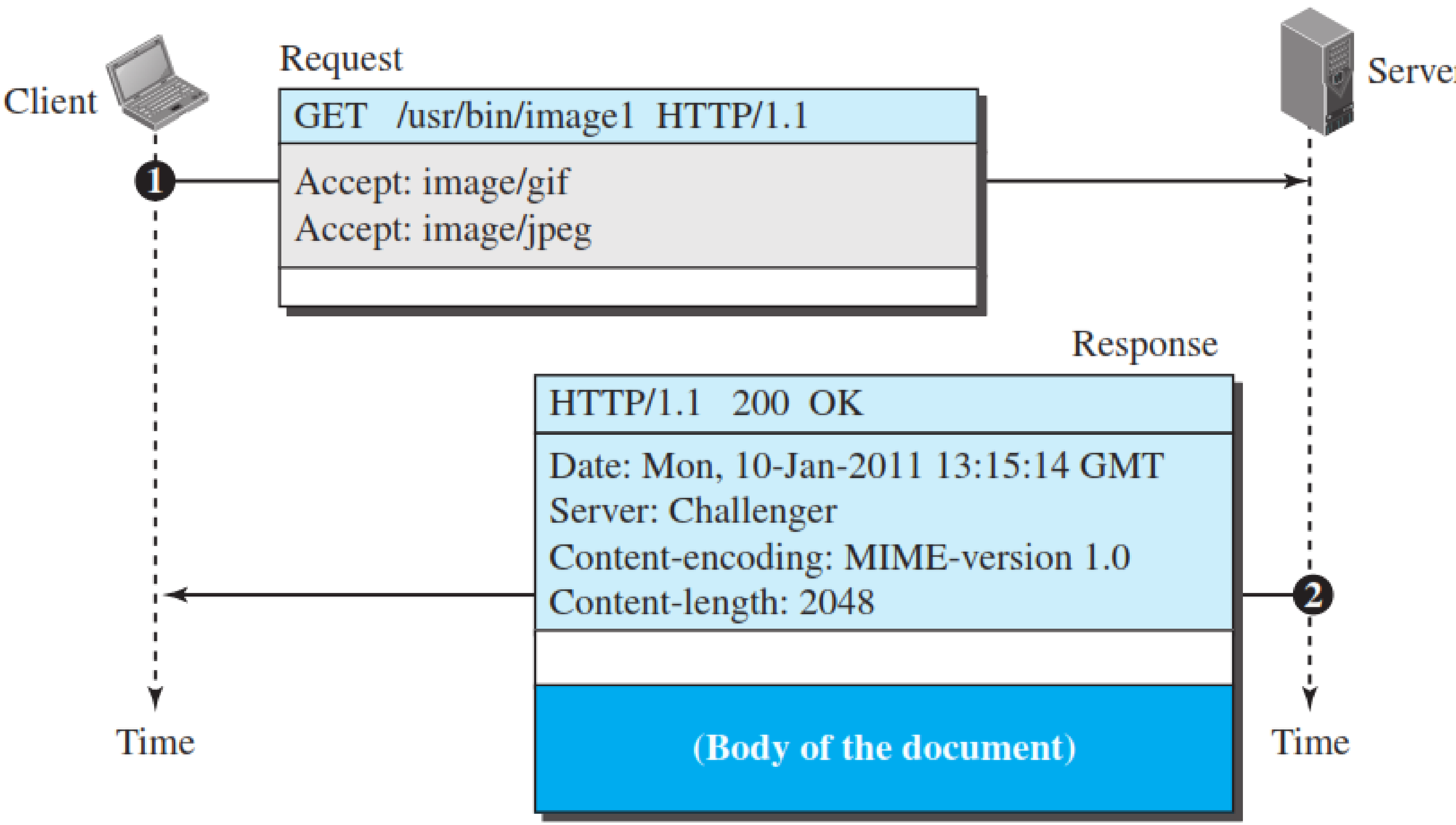
Type of messages:



HTTP : Request Message Methods

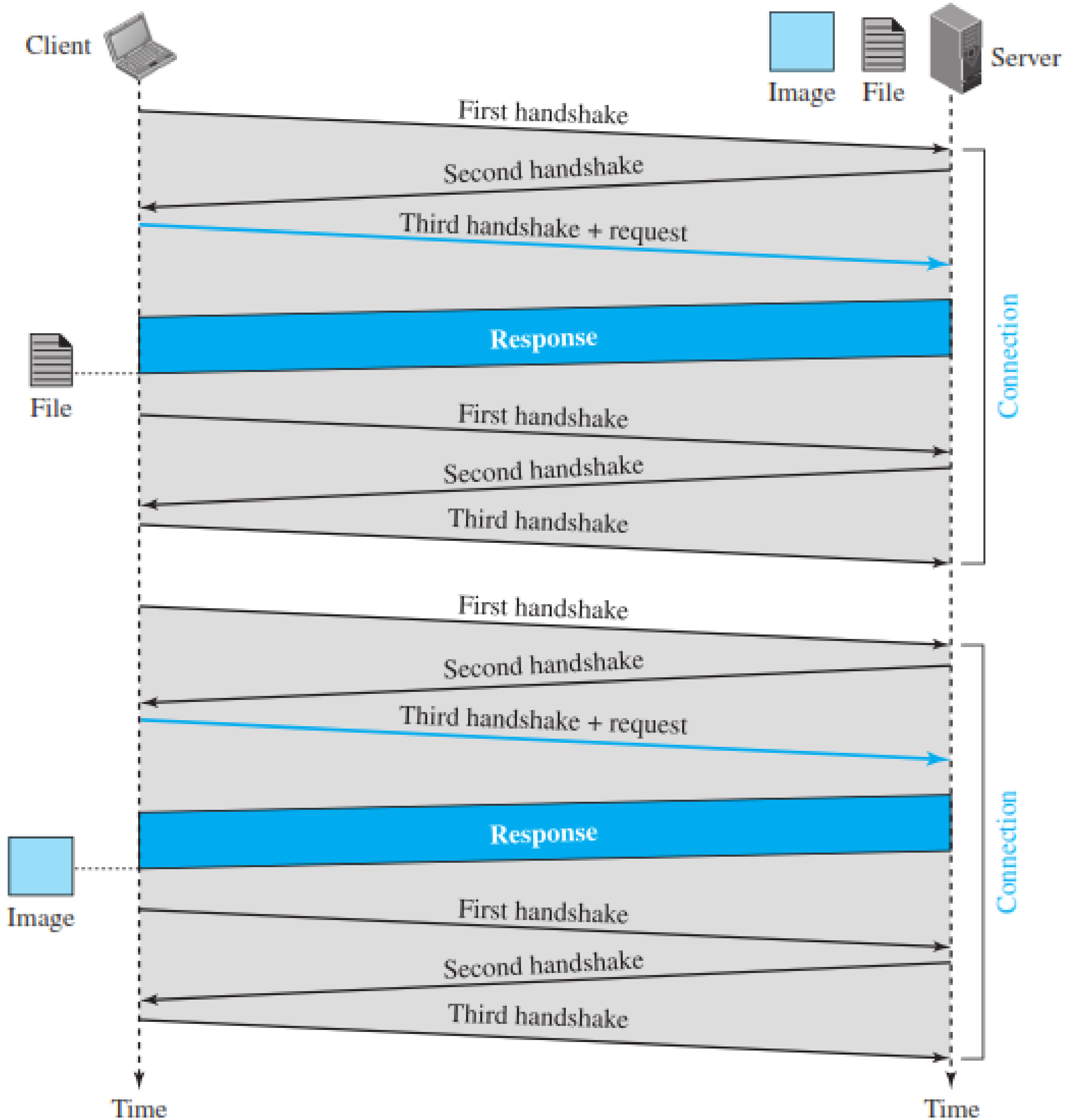
Method	Action
GET	Requests a document from the server
HEAD	Requests information about a document but not the document itself
POST	Sends some information from the client to the server
PUT	Sends a document from the client to the server
TRACE	Echoes the incoming request
CONNECT	Reserved
DELETE	Remove the Web page
OPTIONS	Enquires about available options

HTTP : Example



HTTP : Type of Connections

- **Nonpersistent Connection**
  - One TCP connection is made for each request/response.
- **Persistent Connection**
  - The server leaves the connection open for more requests after sending a response.
  - HTTP version 1.1 specifies a persistent connection by default.


Client
Image
File
Server
First handshake
Second handshake
Third handshake + request
Response
File
First handshake
Second handshake
Third handshake
Connection
First handshake
Second handshake
Third handshake + request
Response
Image
First handshake
Second handshake
Third handshake
Connection
Time
Time

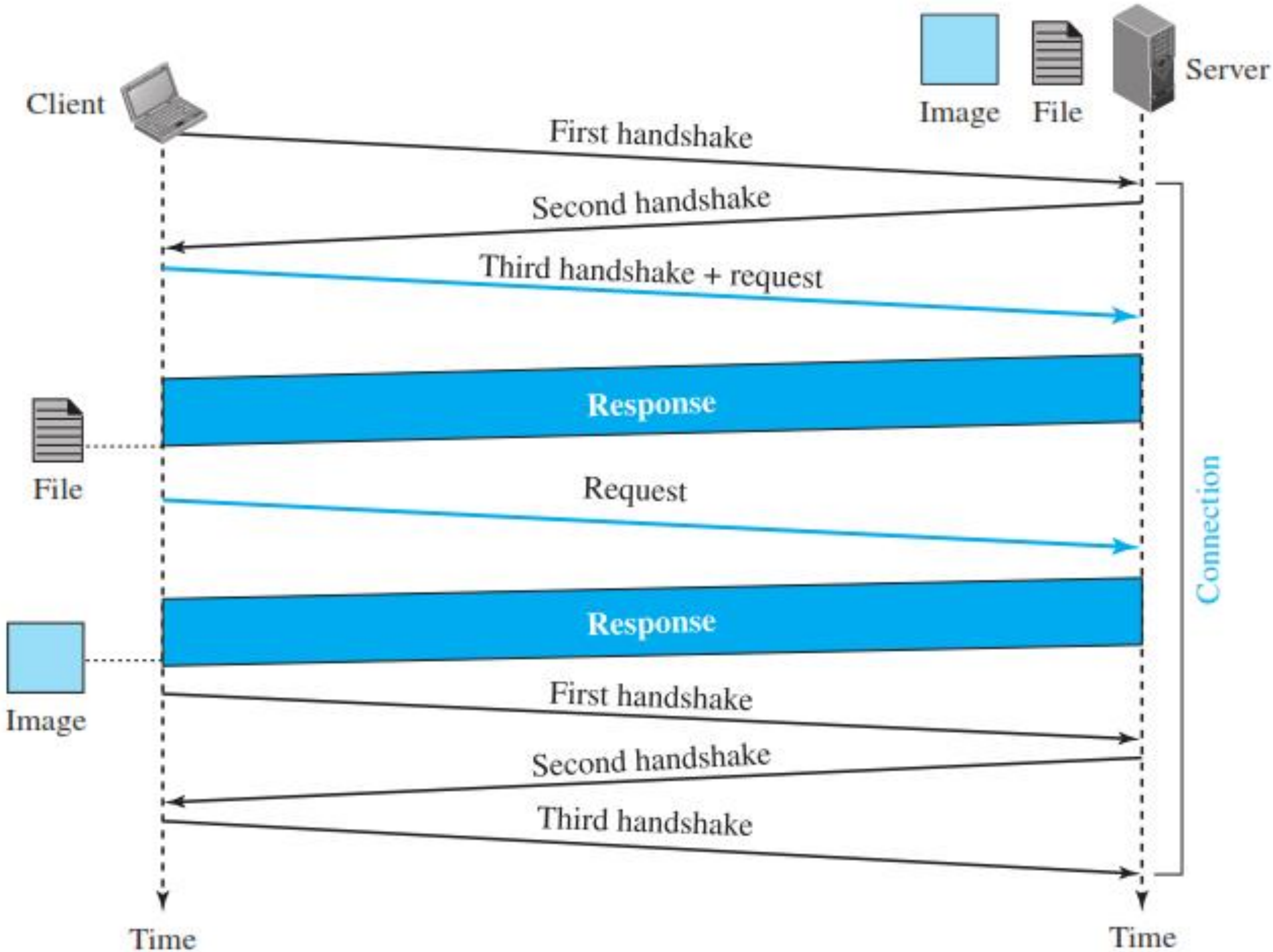
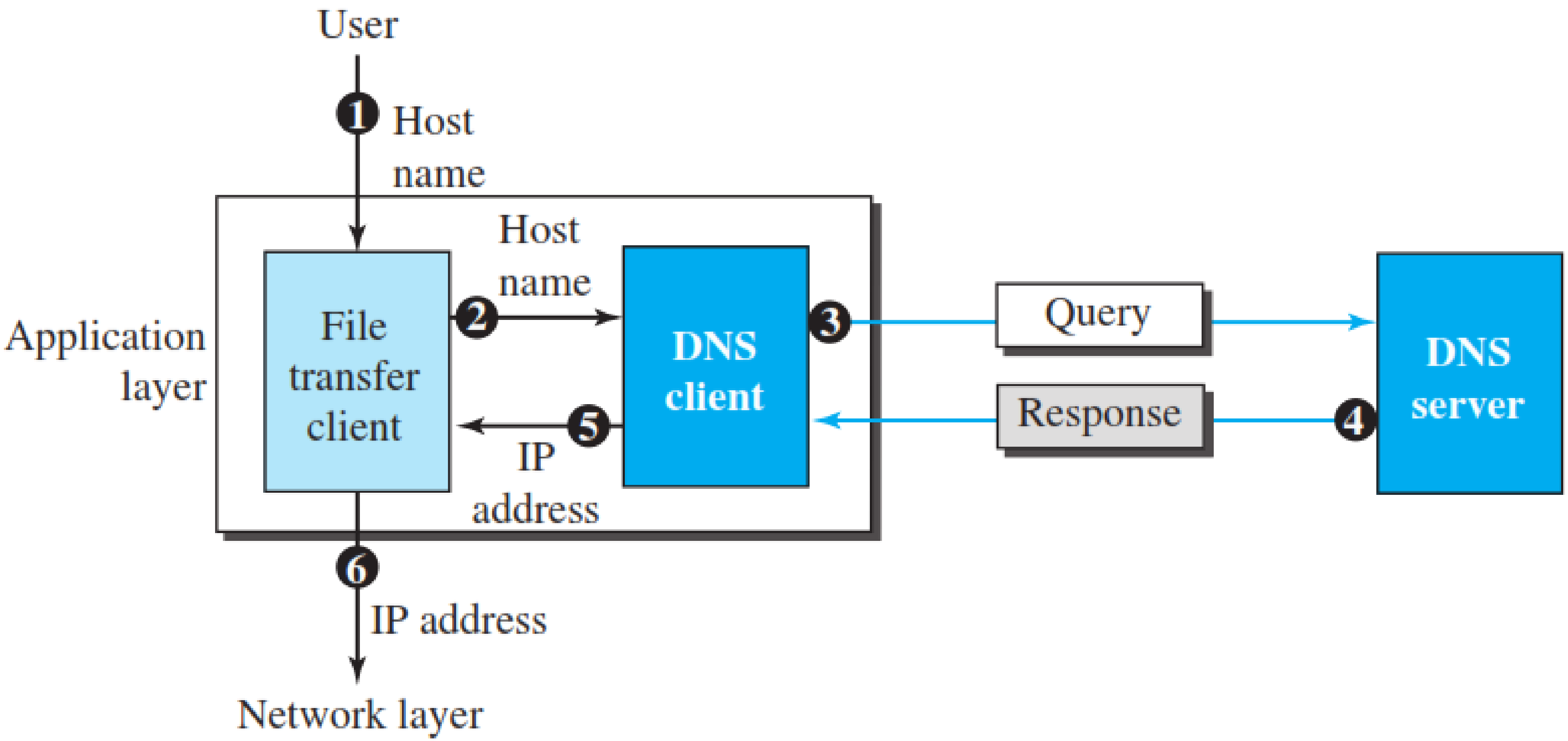


Image
File
Server
Client
First handshake
Second handshake
Third handshake + request
Response
File
Request
Connection
Response
Image
First handshake
Second handshake
Third handshake
Time
Time

Domain Name System (DNS)

- DNS is a client/server application program used to help other application layer programs.
- DNS is used to map a host name in the application layer to an IP address in the network layer.
- WWW uses host name as address of web server at application layer.
- Users can remember name easily.
- TCP/IP protocols uses the IP address of web server to uniquely identify a host.

Domain Name System (DNS)



DNS : Name Space

- DNS uses name space to provide unique name to each host in Internet.
- A name space can be organized in two ways:
  - Flat Name Space
    - A name in this space is a sequence of characters without structure.
    - It can not be used in Internet.
    - Requires a centralized authority to assign names.
  - Hierarchical Name Space
    - A domain name space was designed for Internet that uses a hierarchical name space.

DNS : Name Space

- The names are defined in an inverted-tree structure with the root at the top.
- The tree can have only 128 levels:
  - level 0 (root) to level 127.
- Each node in the tree has a label :
  - a string with a maximum of 63 characters.
- The root label is a null string.
- Children of a node should have different labels.
- Each node in the DNS tree has a domain name.
- A full domain name is a sequence of labels from the node upto root separated by dots (.).

DNS : Domain Name

- Fully Qualified Domain Name (FQDN)
  - If a label is terminated by a null string
  - A DNS server can only match an FQDN to an address.
- Partially Qualified Domain Name (PQDN)
  - If a label is not terminated by a null string

DNS : Distribution of Name Space

- It is not possible to store all domain name on a single server.
- It will increase load on a single server.
- It is not reliable because any failure makes the data inaccessible.
- So, domain names are distributed among many computers called DNS Servers.
- Each server can be responsible (authoritative) for either a large or small domain.
- A server is responsible for or has authority over is called a zone.

DNS : Distribution of Name Space

- Name space is divided among various hierarchical zones.
- The server have a database called a zone file and keeps all the information for every node under that domain.
- A root server is a server whose zone consists of the whole tree.
- There are several root servers, each covering the whole domain name space.
- The root servers are distributed all around the world.

DNS : Distribution of Name Space

- DNS defines two types of servers:
  - Primary Server
    - It is a server that stores a file about the zone for which it is an authority.
    - It is responsible for creating, maintaining, and updating the zone file.
  - Secondary Server
    - It is a server that loads the complete information about a zone from another server.
    - The secondary server neither creates nor updates the zone files.

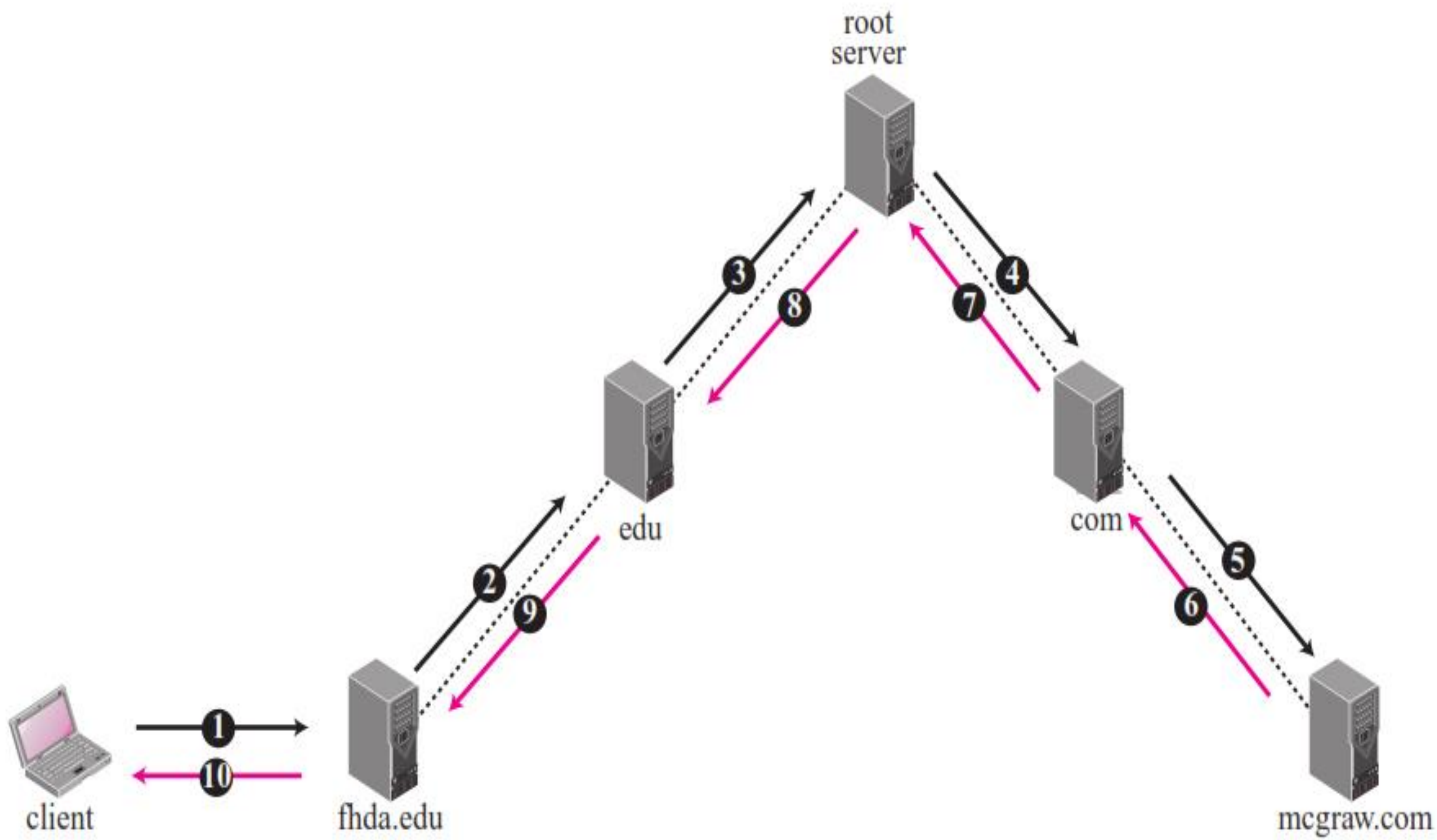
DNS : DNS Resolution

- Mapping a name to an address or an address to a name is called name-address resolution.
- A host that needs to map an address to a name or a name to an address calls a DNS client called a resolver.
- The resolver accesses the closest DNS server with a mapping request.
- After the resolver receives the mapping, it verifies and delivers the result to the process that requested it.
- Resolver can do two type of mapping:
  - Names to Addresses
  - Addresses to Names

DNS : DNS Resolution

- A resolution can be of two types:
  - Recursive Resolution
  - Iterative Resolution

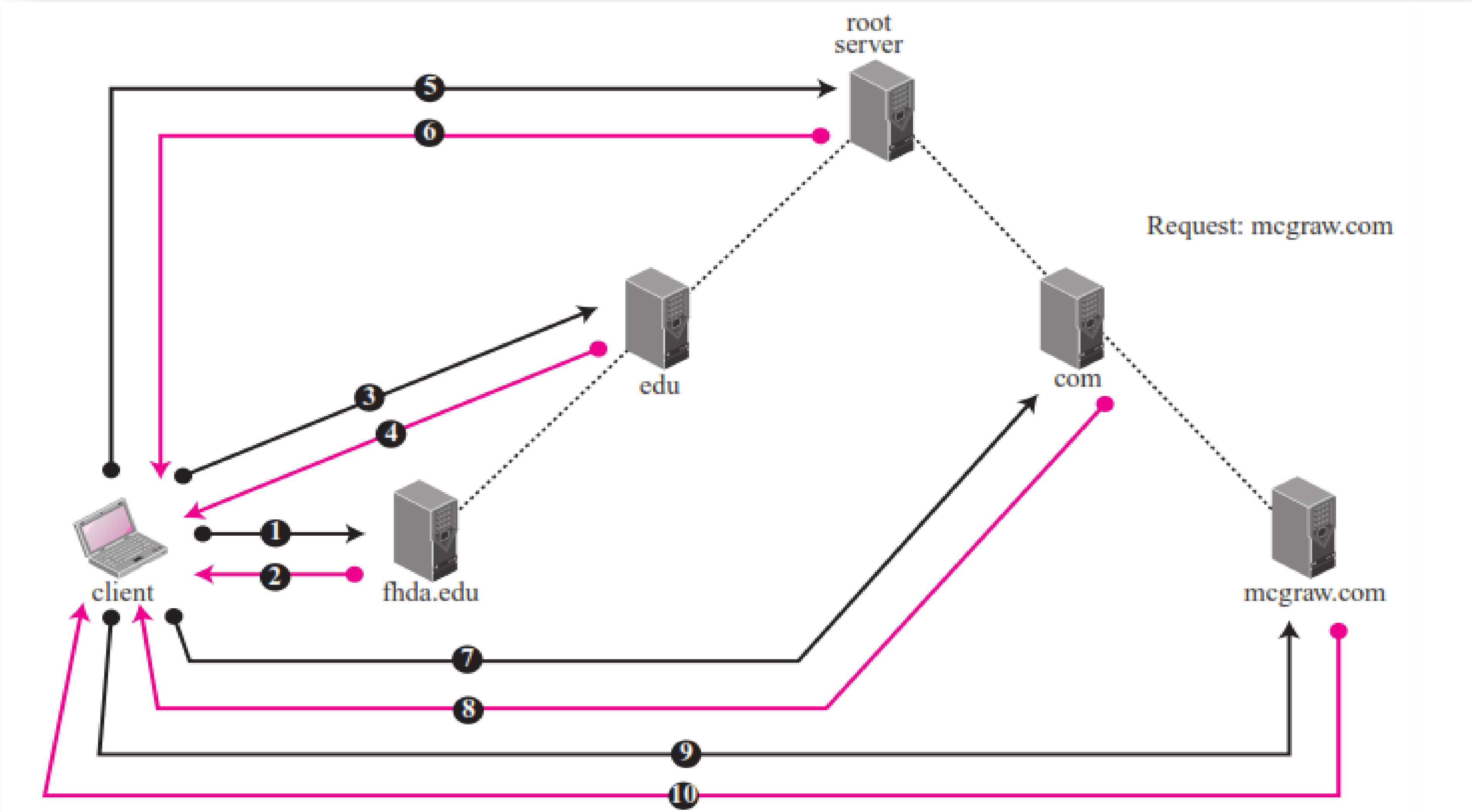
DNS : Recursive Resolution



DNS : DNS Resolution

- The client (resolver) asks for a recursive answer from a nearest local DNS server.
- Client requires the Local Server to give either the requested mapping or an error message.
- If the server is the authority for the domain name, it checks its database and responds.
- If the server is not the authority, it sends the request to another server (the parent usually) and waits for the response.
- When the query is resolved, the response travels back until it reaches the requesting client.

DNS : Iterative Resolution



DNS : Iterative Resolution

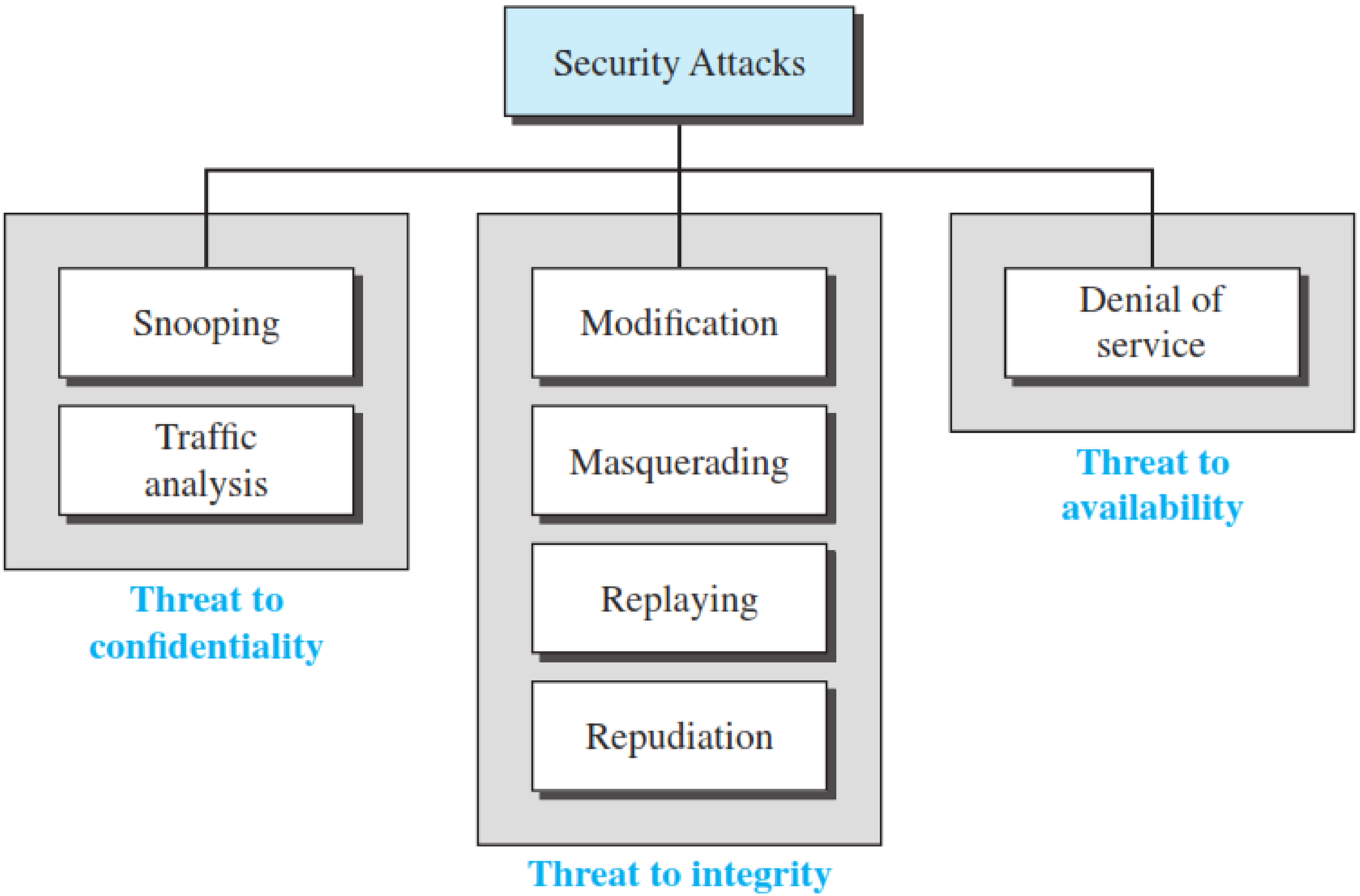
- The client does not ask for a recursive answer, then the mapping can be done iteratively.
- If the server is an authority for the name, it sends the answer.
- If it is not, it returns (to the client) the IP address of the server that it thinks can resolve the query.
- The client is responsible for repeating the query to this second server.
- This process is called iterative because the client repeats the same query to multiple servers.

Network Security

Network Security

- Information is an asset that has a value like any other asset.
- Information needs to be secured when transmitted.
- Network Security goals:
  - Confidentiality - hide from unauthorized access
  - Integrity - protect from unauthorized change (modification)
  - Availability - available to an authorized entity when it is needed
- Attack
  - Attack is malicious (unauthorized) attempts that are carried out by cybercriminals to compromise the security goals.
- Classification of attacks
  - Active Attack - attempts to alter system resources.
  - Passive Attack - make use of information from the system but does not affect system resources.

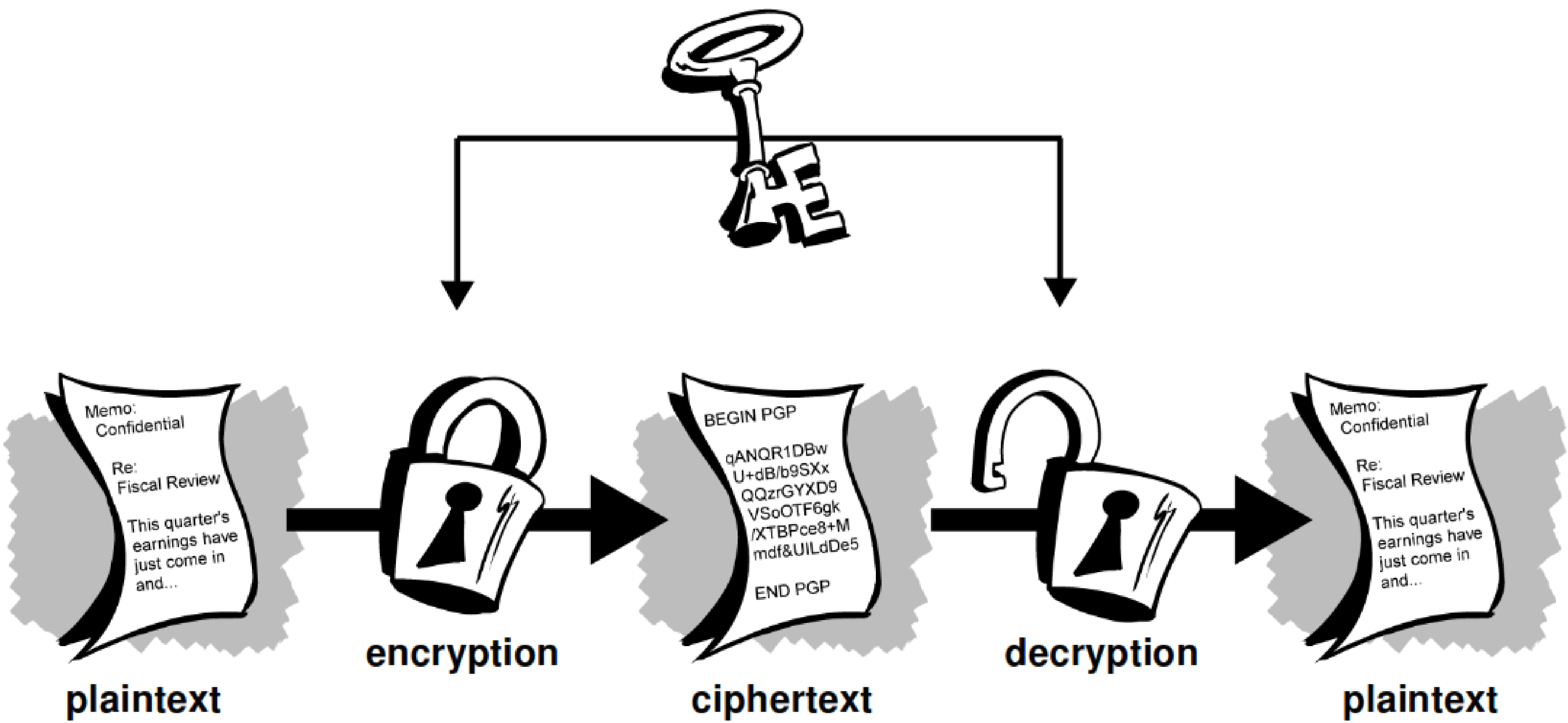
Taxonomy of attacks



Cryptography

- Science and art of transforming messages to make them secure and immune to attacks.
- Data that can be read and understood without any special measures is called **plaintext** or **cleartext**.
- The method of disguising plaintext in such a way as to hide its substance is called **encryption**.
- The scrambled message produced as output of encryption is called **ciphertext**.
- The process of reverting ciphertext to its original plaintext is called **decryption**.

Cryptography

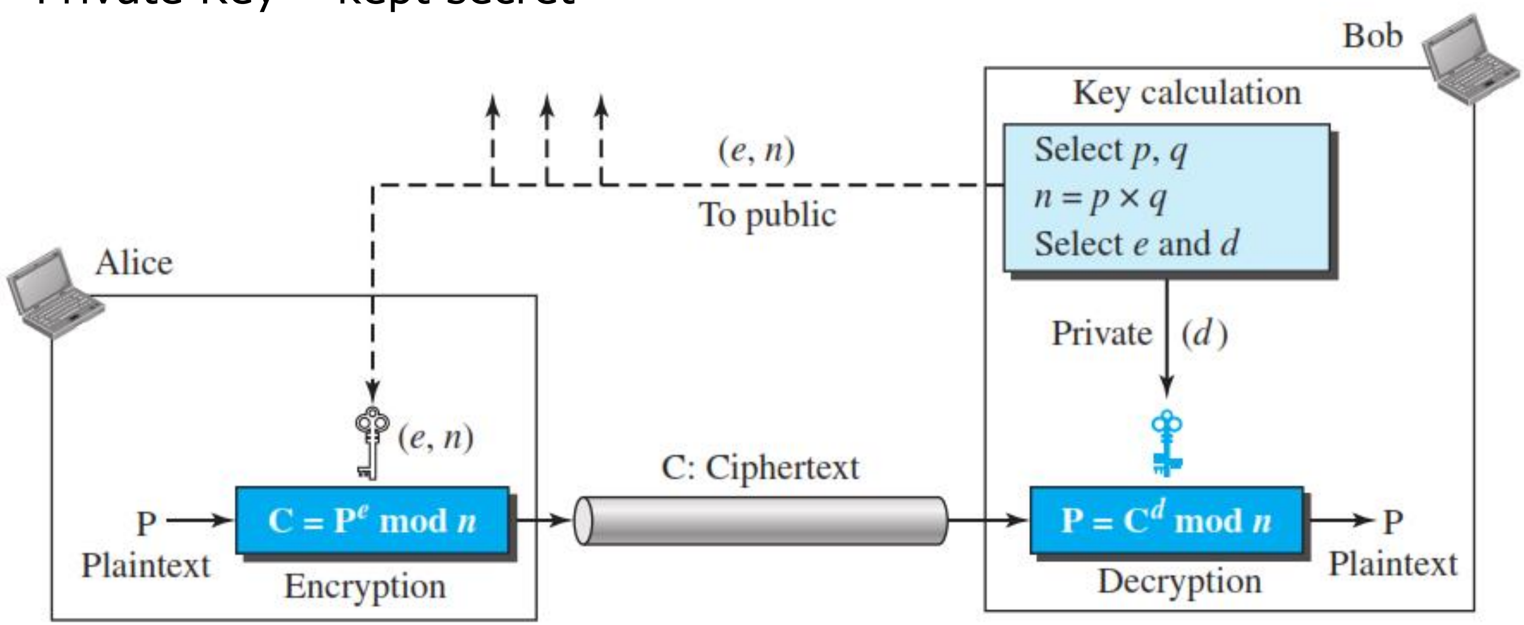


Types of Cryptography

- A cryptographic algorithm, or cipher, is a mathematical function used in the encryption and decryption process.
- A cryptographic algorithm works in combination with a key—a word, number, or phrase—to encrypt the plaintext.
- Types of Cryptography
  - symmetric-key cryptography
  - asymmetric-key cryptography/ public key cryptography

RSA Algorithm

- One of the common public- key algorithms is the RSA cryptosystem.
- Named for its inventors (Rivest, Shamir, and Adleman).
- It works on two different keys.
  - Public Key - given to everyone
  - Private Key – kept secret



RSA Algorithm

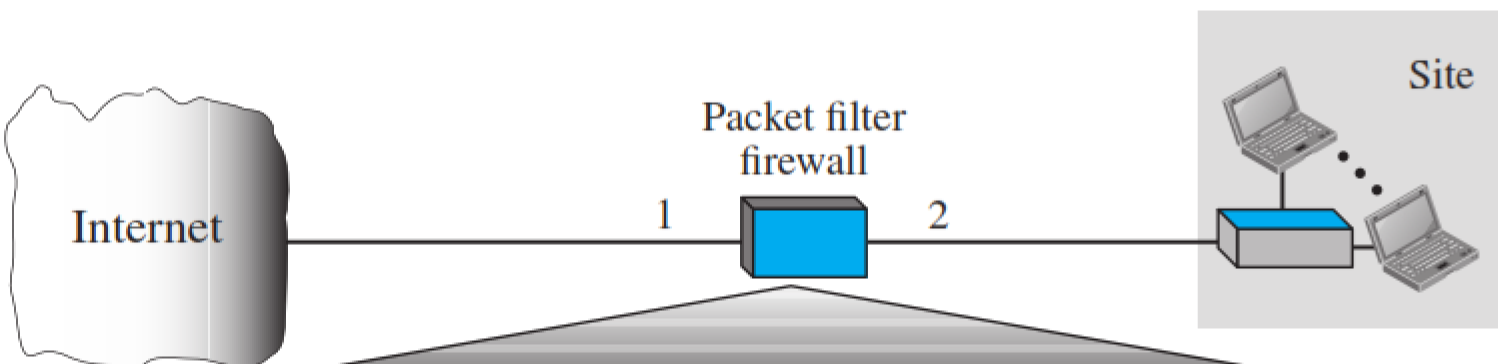
- chooses two large prime numbers, p and q
- calculates $n = p \times q$ and $\Phi(n) = (p - 1) \times (q - 1)$
- select an integer e such that
  - Not be a factor of $\Phi(n)$
  - $1 < e < \Phi(n)$
- Public Key is PK(e,n)
- select an integer d such that
  - $d \times e \bmod \Phi(n) = 1$
- Private Key is PR(d,n)

FIREWALLS

- A firewall is a device (usually a router or a computer) installed between the internal network of an organization and the rest of the Internet.
- It is designed to forward some packets and filter (not forward) others.
- A firewall can be used to deny access to a specific host or a specific service in the organization.
- A firewall is usually classified as
  - packet-filter firewall
  - proxy-based firewall

Packet-Filter Firewall

- It can forward or block packets based on the information in the network-layer and transport-layer headers.
- A packet-filter firewall is a router that uses a filtering table.



Interface	Source IP	Source port	Destination IP	Destination port
1	131.34.0.0	*	*	*
1	*	*	*	23
1	*	*	194.78.20.8	*
2	*	*	*	80

Proxy Firewall

- It can forward or block packets based on the information available in the message itself.
- A proxy firewall filters at the application layer.
- Sometimes called an application gateway.