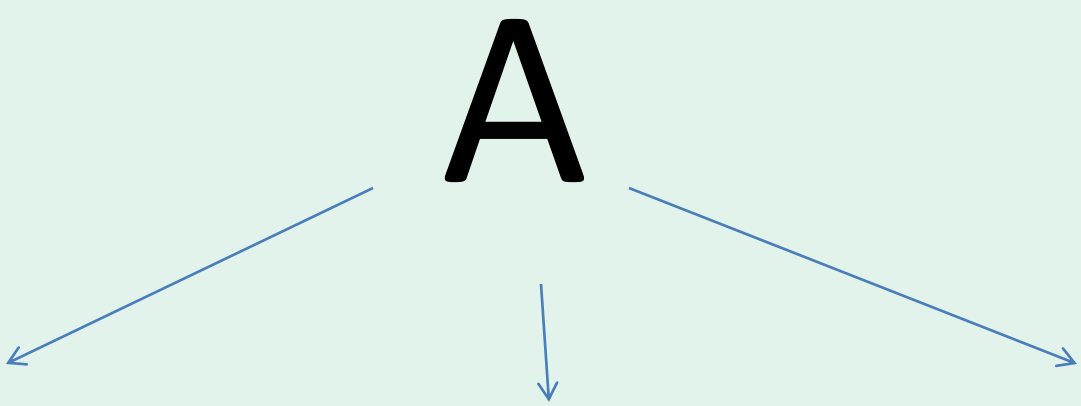
NETWORK SOFTWARE MODELS

UNIT-1(Part-B)

Contents:

- Functionalities of Computer Network
- Introduction to design of Network Software
- OSI Reference Model
- TCP/IP Model
- Comparison between OSI & TCP/IP

Encoding



00001010

ASCII - American Standard Code for Information Interchange

01000001

Extended Binary Coded Decimal Interchange Code (**EBCDIC**)

00000000 00000000 00000000 01000001

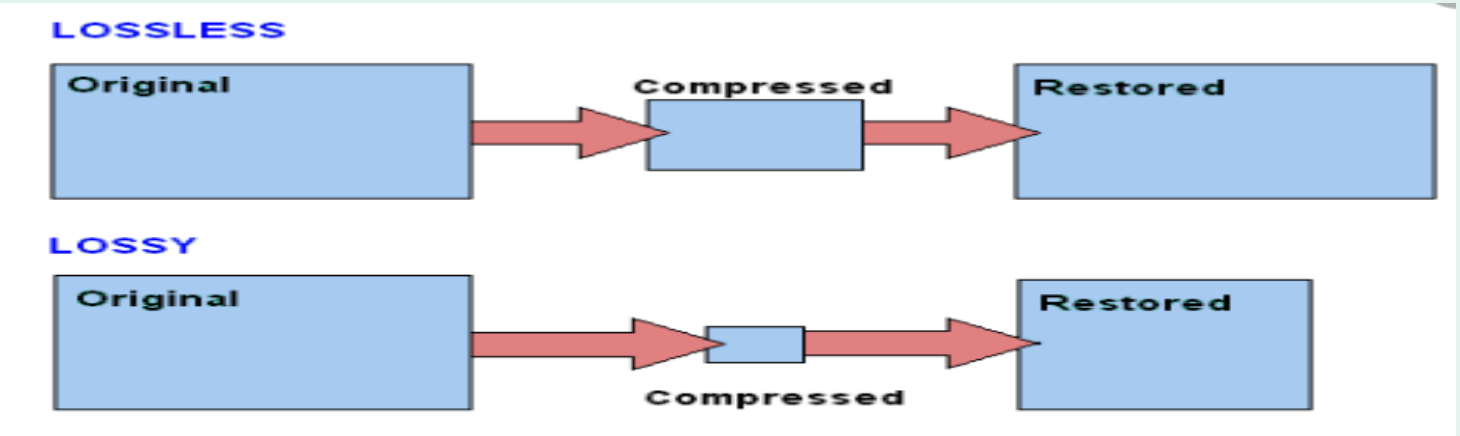
Unicode Text Format(UTF-32)

Type of Service

- Connection Oriented Mechanism
  - Connection Establishment
  - Data Transfer
  - Connection Release
- Connectionless Mechanism
  - Direct Data Transfer

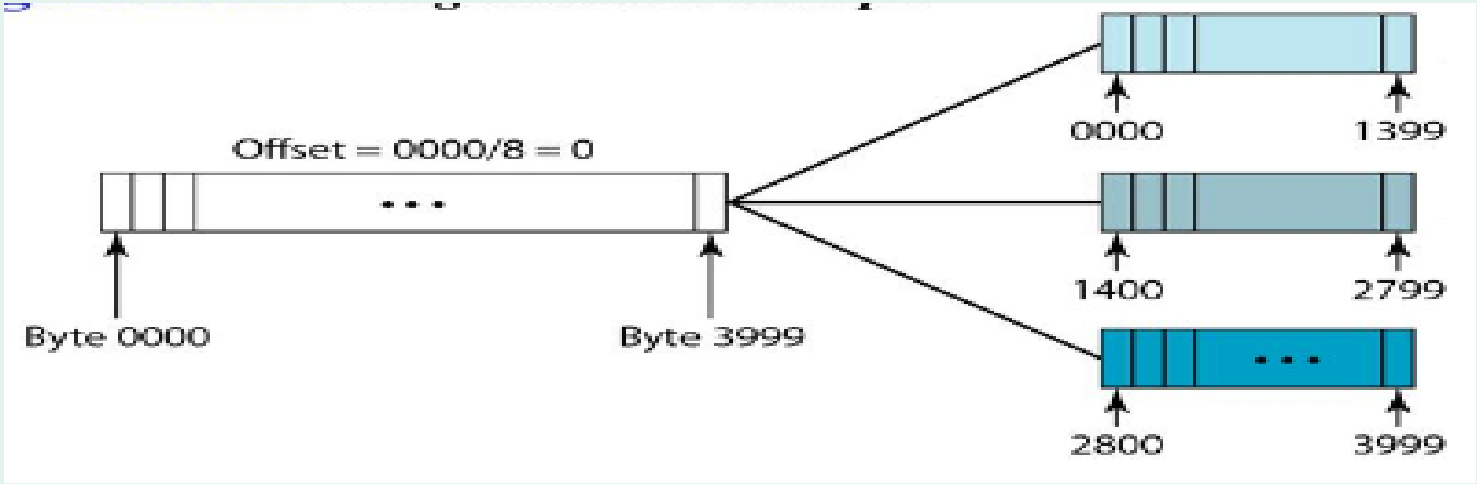
Compression

- **Compression** is the method **computers** use to make files smaller by reducing the number of bits (1's and 0's) used to store the information.
- Compression
  - Lossless Compression
  - Lossy Compression



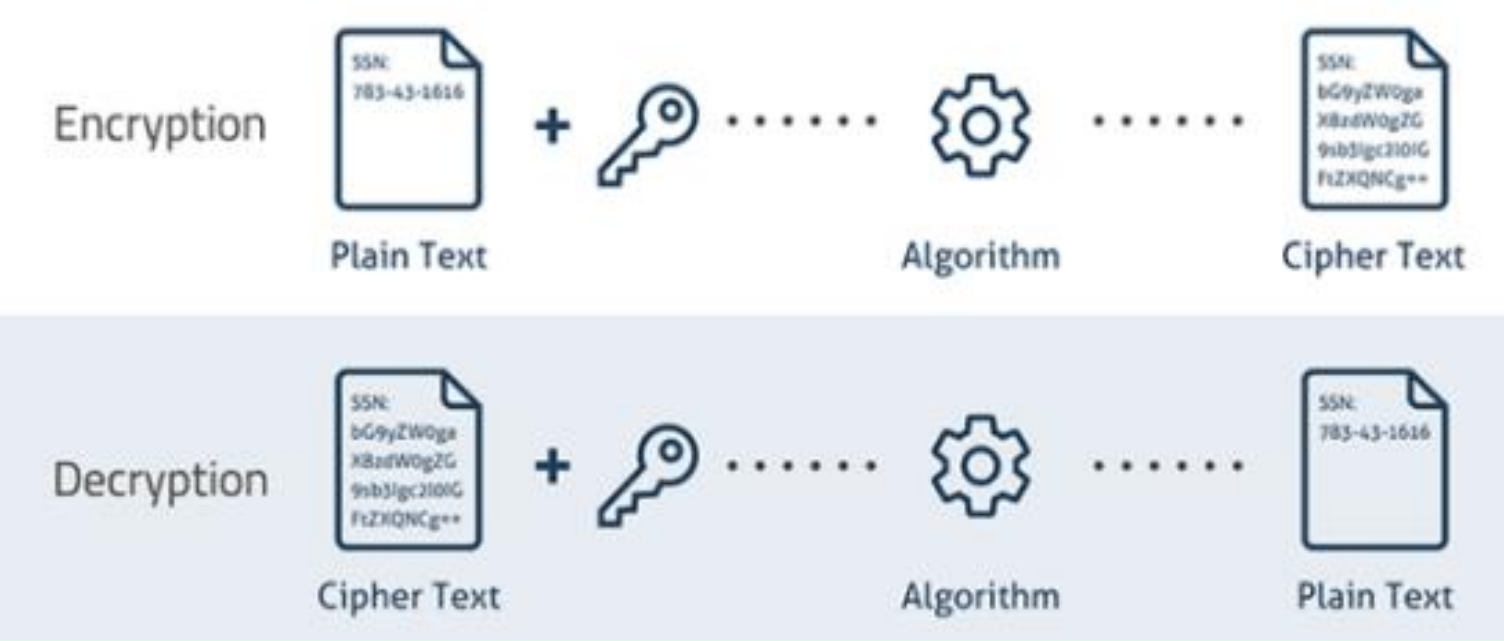
Dividing into Smaller units of data (Segments/Packets/Frames)

- Network Software Support
- Medium Support
- Easy transmission
- Error → Resending



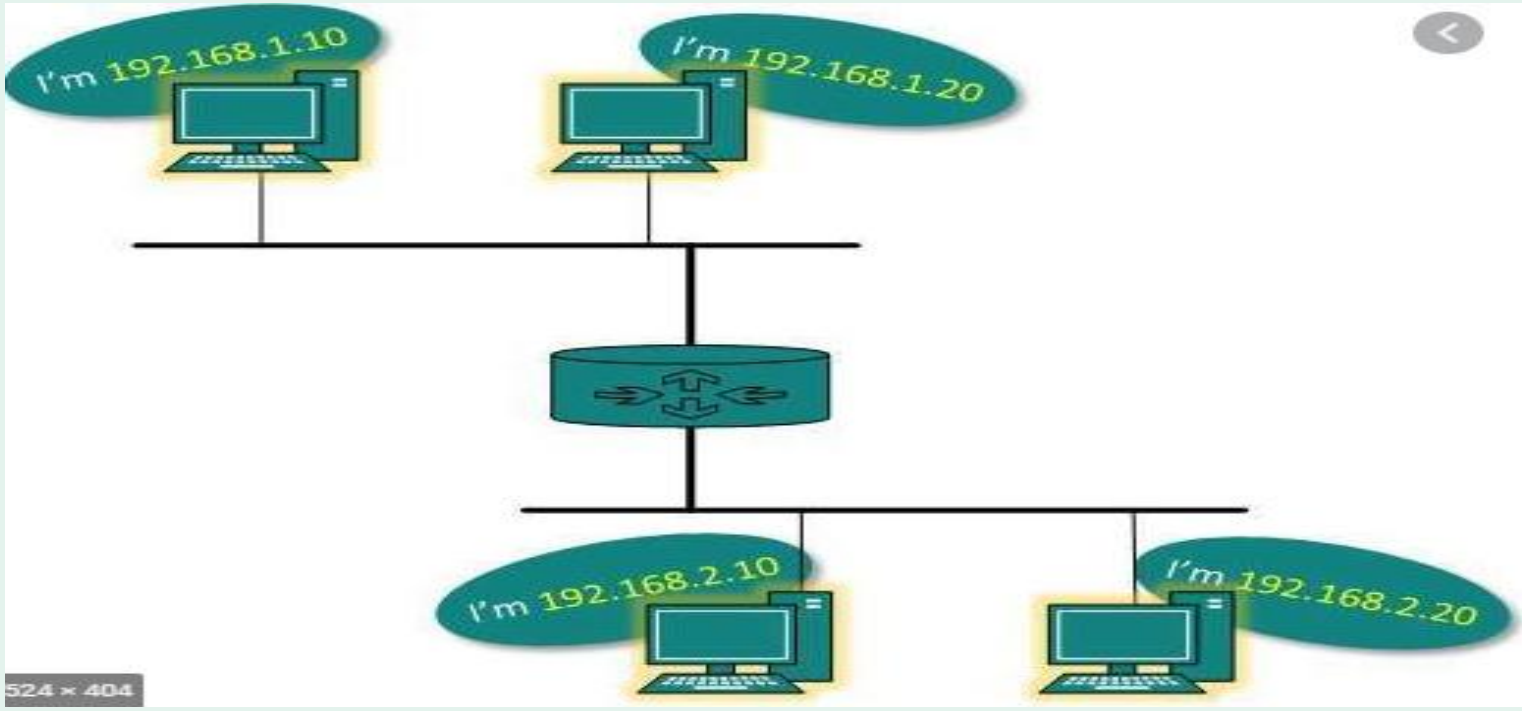
Encryption

- **Encryption** is a process that encodes a message or file so that it can be only be read by certain people



Logical Addressing

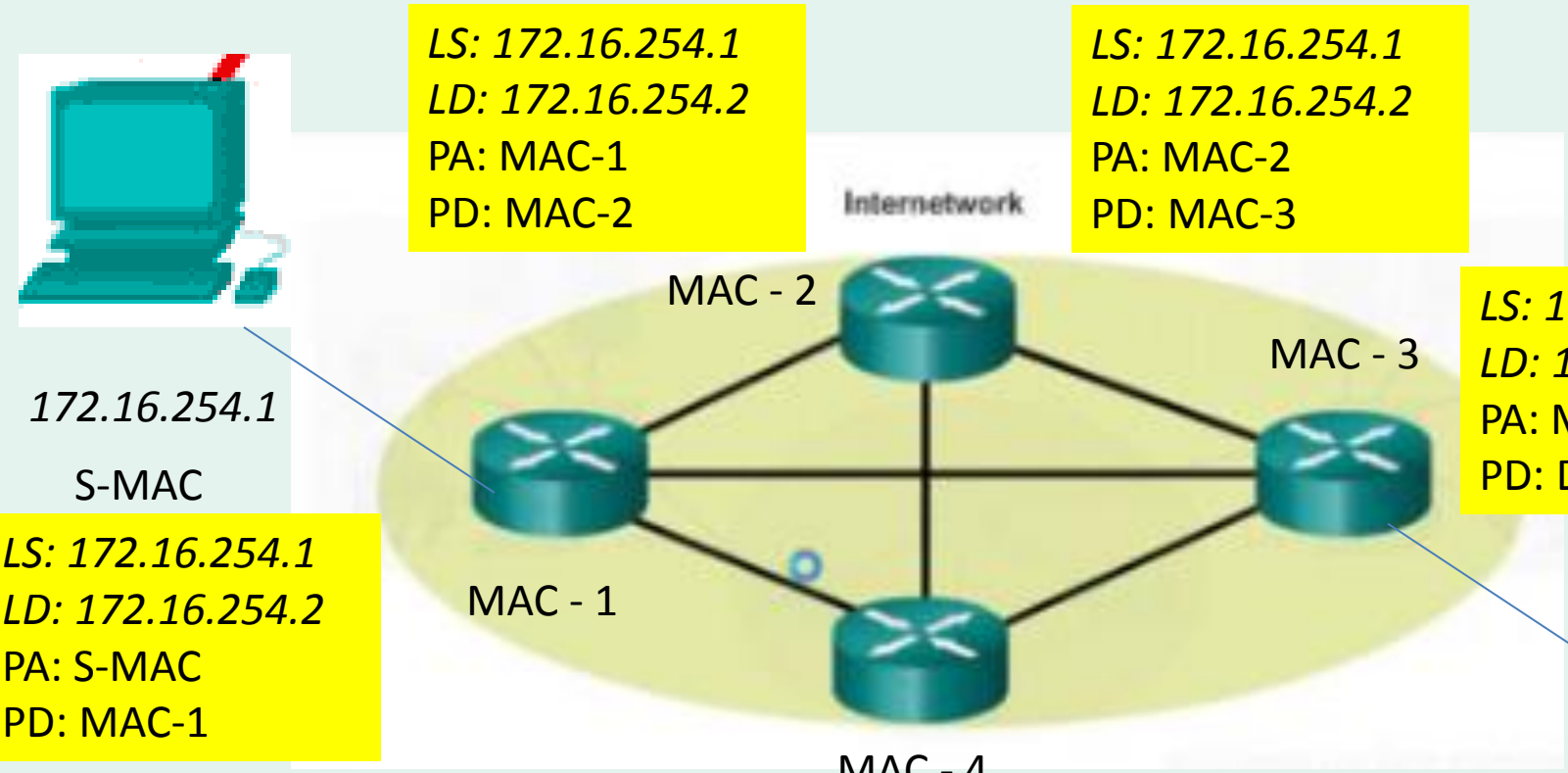
- The method of assigning unique ID to every device connected in network
- End to End communication



- Delivery parameters (cookies)
  - Time
  - When to start & end in Connection Oriented Service
- Delivery confirmation
  - Reliable - Acknowledgement
  - Unreliable - No acknowledgement
- Media
  - Guided
  - Unguided

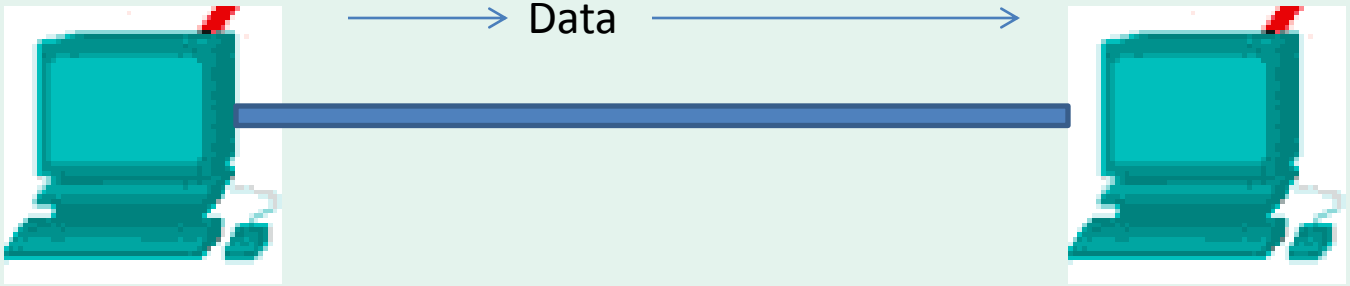
Physical Addressing

- Hop by Hop communication



Service

- Process to Process communication



File Transfer
Web Browser
Database server

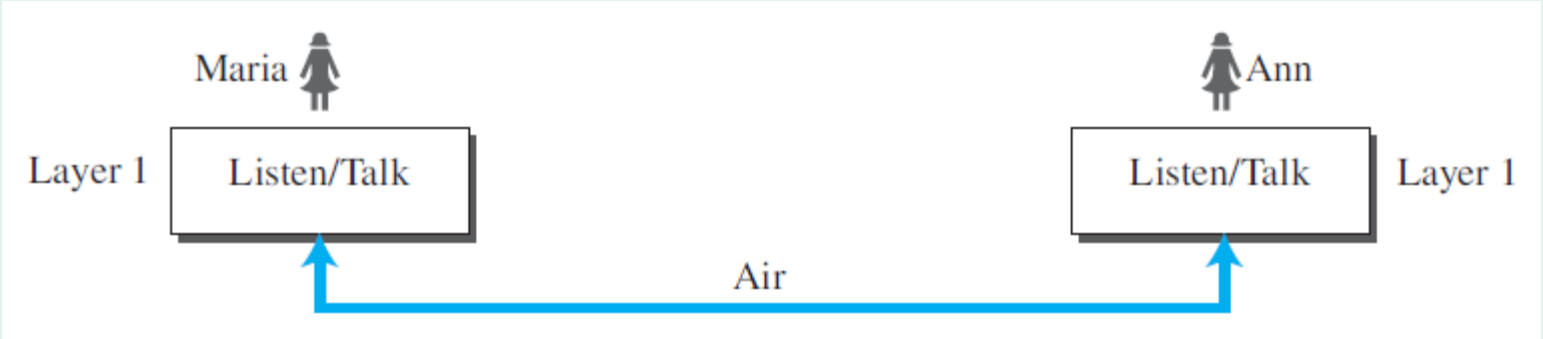
Symmetrical Problems in Network Communication

General Issue	Computer Networks
Language	Encoding (Character / Signal)
Seeking Permission	Connection Oriented/Connectionless
Actual Address	Logical Address
Packing mechanism	Compression
Security	Encryption
Delivery Time	Dialog Control
Delivery Confirmation	Reliability
Postal Package Damage	Error Control (Network/Sender)
Mode of Travel	Media
Intermediary Path	Physical Address
In Person Delivery	Process
Weight	Dividing Data (Segmentation, Packeting, Framing)

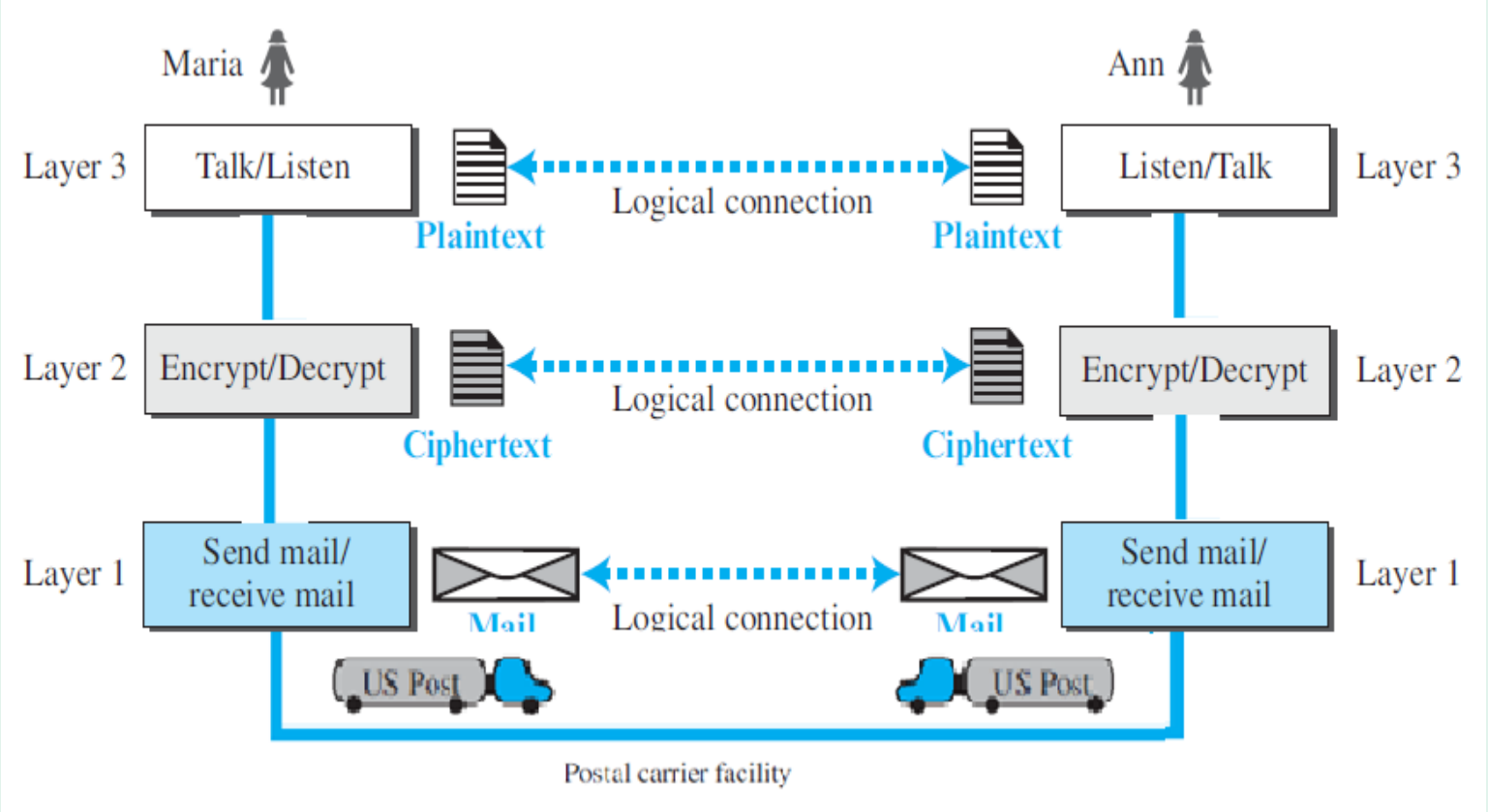
Protocol Layering

- A protocol defines the rules that both the sender and receiver and all intermediate devices need to follow to be able to communicate effectively.
- When communication is simple, we may need only one simple protocol.
- When the communication is complex, we may need to divide the task between different layers, in which case we need a protocol at each layer, or protocol layering.

Scenario 1



Scenario 2



Advantages:

- Enables us to divide a complex task into several smaller and simpler tasks
- It allows us to separate the services from the implementation
- Service gets compromised it is responsibility of that layer which is performing the service
- Intermediate systems that need only some layers, but not all layers(ex. Intermediate mail sender)

Principles in design:

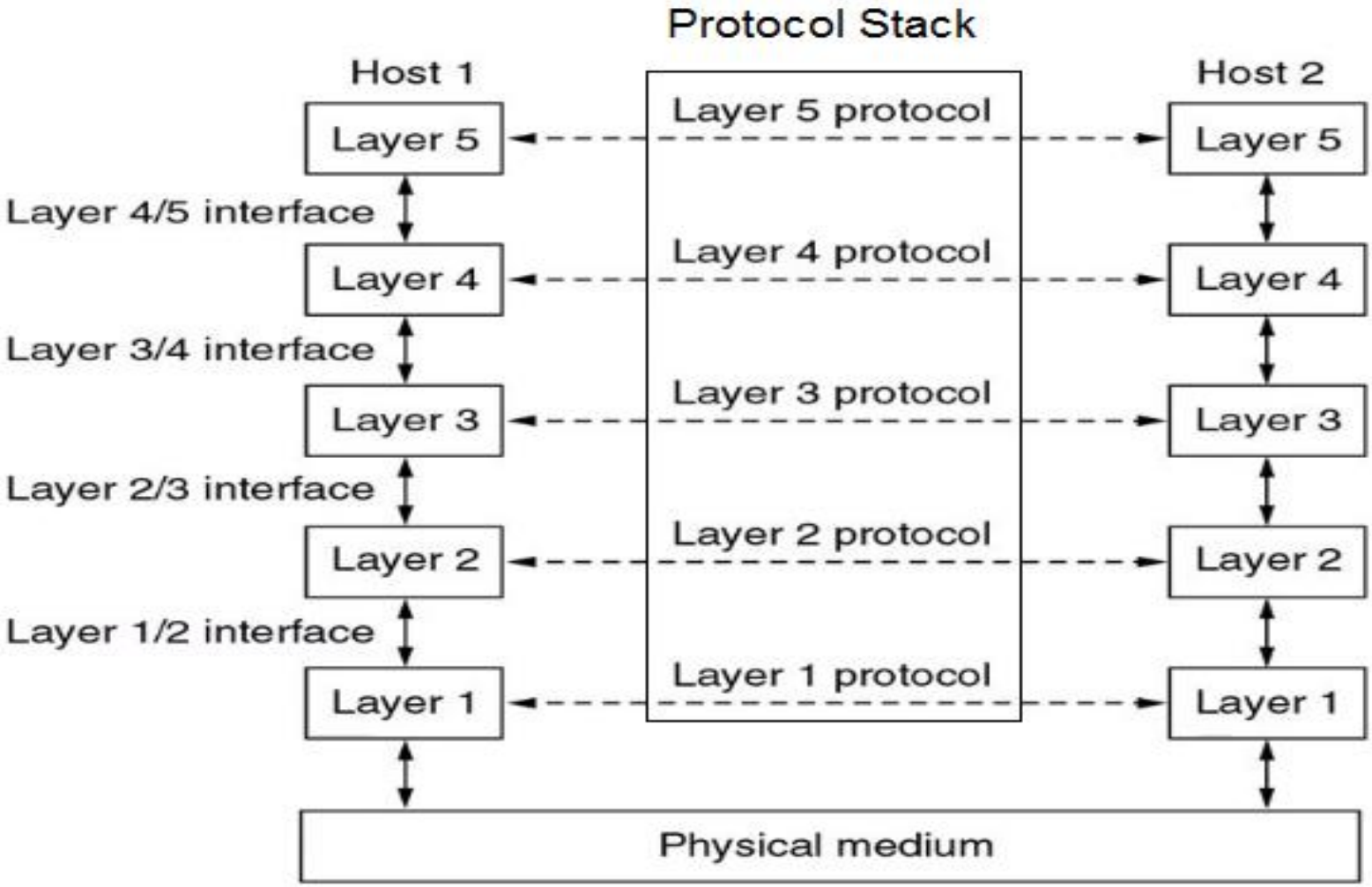
- Make each layer so that it is able to perform two opposite tasks, one in each direction(i.e . talk/listen, encrypt/decrypt, send/receive)
- Two objects under each layer at both sites should be identical (letter, cipher text, piece of mail)

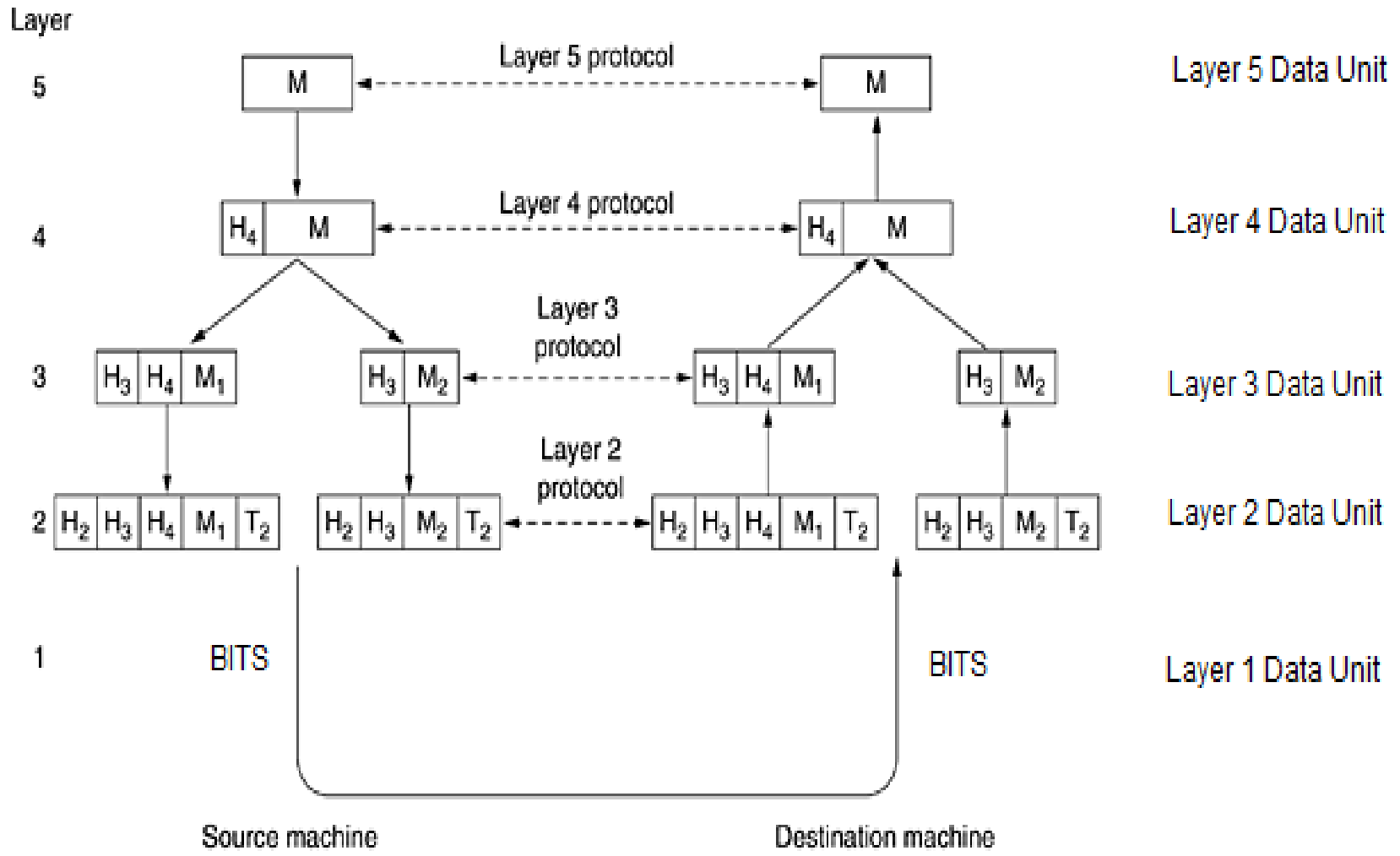
Network Software design

- To reduce their design complexity, most networks are organized as a stack of layers or levels, each one built upon the one below it.
- **A protocol** is an agreement between the communicating parties on how communication is to proceed.
- Each layer passes data and control information to the layer immediately below it, until the lowest layer is reached and vice versa(i.e. hierarchical) .
- Between each pair of adjacent layers is an **interface**. The interface defines which primitive operations and services the lower layer makes available to the upper one.

Network Software design

- A set of layers and protocols is called a **network architecture**. The specification of an architecture must contain enough information to allow an implementer to write the program or build the hardware for each layer so that it will correctly obey the appropriate protocol.
- A list of protocols used by a certain system, one protocol per layer, is called a **protocol stack**.
- A service is formally specified by a set of **primitives (operations)** available to a user process to access the service. These primitives tell the service to perform some action or report on an action taken by a peer entity to the layer above it. (ex. SEND/RECEIVE/ CONNECT/DISCONNECT/LISTEN)
- The set of primitives available depends on the nature of the service being provided.


Protocol Stack
Host 1
Host 2
Layer 5
Layer 5 protocol
Layer 5
Layer 4/5 interface
Layer 4
Layer 4 protocol
Layer 4
Layer 3/4 interface
Layer 3
Layer 3 protocol
Layer 3
Layer 2/3 interface
Layer 2
Layer 2 protocol
Layer 2
Layer 1/2 interface
Layer 1
Layer 1 protocol
Layer 1
Physical medium

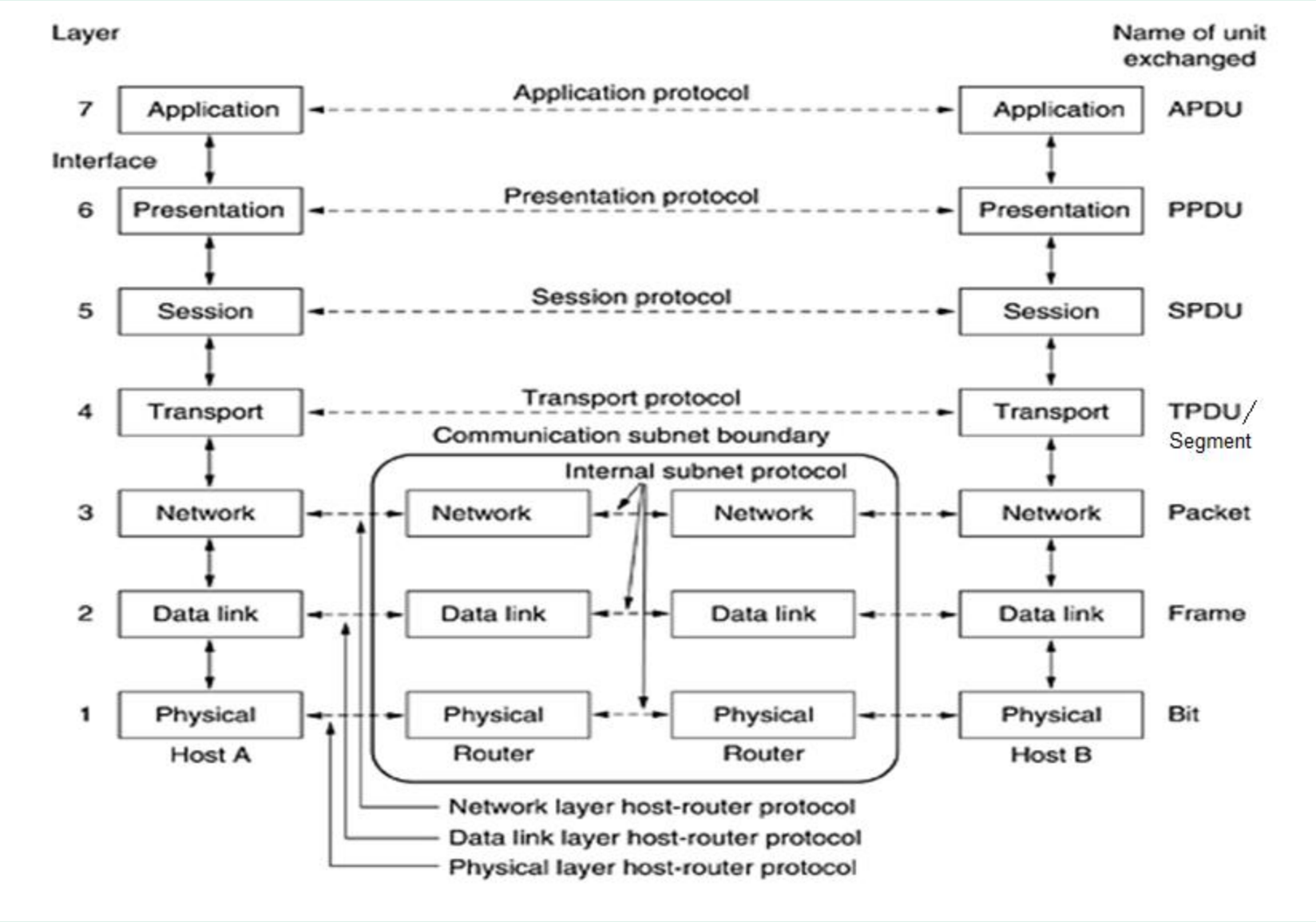

Layer
5
4
3
2
1
M
Layer 5 protocol
M
Layer 5 Data Unit
H4 M
Layer 4 protocol
H4 M
Layer 4 Data Unit
Layer 3 protocol
H3 H4 M1
H3 M2
H3 H4 M1
H3 M2
Layer 3 Data Unit
Layer 2 protocol
H2 H3 H4 M1 T2
H2 H3 M2 T2
H2 H3 H4 M1 T2
H2 H3 M2 T2
Layer 2 Data Unit
BITS
BITS
Layer 1 Data Unit
Source machine
Destination machine

ISO-OSI Reference Model / Open Systems Interconnection(OSI) Model

Principles in Design of OSI Model:

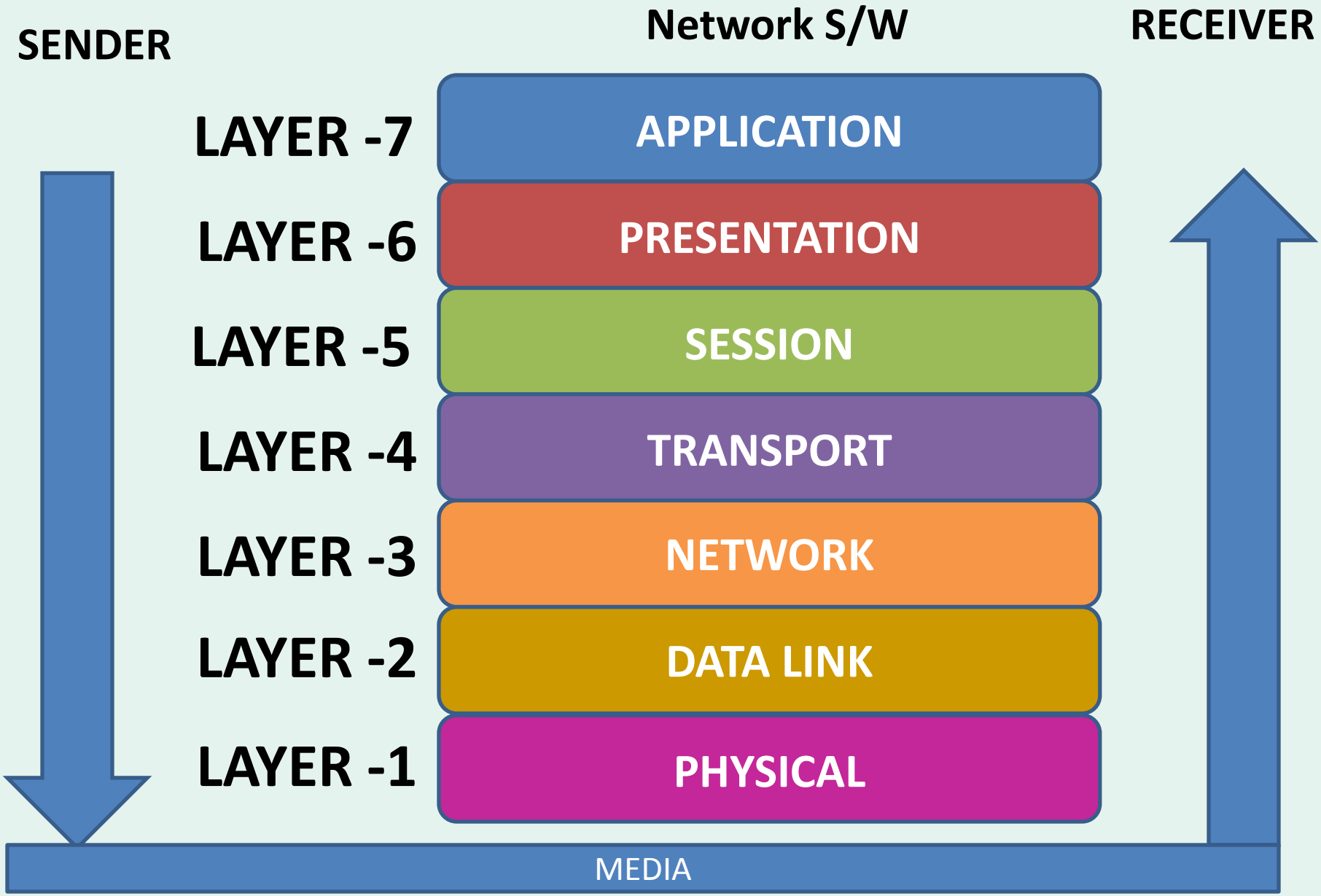
- Layered Architecture
- A layer should be created where a different abstraction is needed.
- The number of layers should be large enough that distinct functions need not be thrown together
- Order of functionality is defined
- Each layer should perform a well-defined function.
- The function of each layer should be chosen with an eye toward defining internationally standardized protocols.
- The layer boundaries should be chosen to minimize the information flow across the interfaces.
- Encapsulation

- OSI Model has 7 Layers



Mnemonic: **P**lease **D**o Not **T**hrow **S**ausage **P**izza **A**way (OR) **A**ll **P**eople **S**eem **T**o **N**eed **D**ata **P**rocessing

OSI Model


SENDER
Network S/W
RECEIVER
LAYER -7
APPLICATION
LAYER -6
PRESENTATION
LAYER -5
SESSION
LAYER -4
TRANSPORT
LAYER -3
NETWORK
LAYER -2
DATA LINK
LAYER -1
PHYSICAL
MEDIA

OSI MODEL

- **APPLICATION**
- PRESENTATION
- SESSION
- TRANSPORT
- NETWORK
- DATA LINK
- PHYSICAL

APPLICATION LAYER (LAYER-7)

- It provides user applications with access to network services
- There are several network services
  - Web Services
  - File Transfer and access
  - Remote Login
  - Mail Services
  - News Services
- Data Unit: Application Protocol Data Unit

OSI MODEL

- APPLICATION
- **PRESENTATION**
- SESSION
- TRANSPORT
- NETWORK
- DATA LINK
- PHYSICAL

PRESENTATION LAYER (LAYER-6)

- This layer is concerned with syntax and semantics of the information transmitted
- The main functions are:
  - Encoding
  - Compression
  - Encryption
- Data Unit: Presentation Protocol Data Unit

OSI MODEL

- APPLICATION
- PRESENTATION
- **SESSION**
- TRANSPORT
- NETWORK
- DATA LINK
- PHYSICAL

SESSION LAYER (LAYER-5)

- This layer allows users on different machines to establish session between them
- Session establishment is application specific
- The main functions are:
  - Dialog control (Half/Full Duplex)
  - Token management (Who)
  - Synchronization (Time)
- Data Unit: Session Protocol Data Unit

OSI MODEL

- APPLICATION
- PRESENTATION
- SESSION
- **TRANSPORT**
- NETWORK
- DATA LINK
- PHYSICAL

TRANSPORT LAYER (LAYER-4)

- This layer is responsible for **Process to Process** communication
- The main functions are:
  - Connection control
  - Port Addressing
  - Segmentation & Reassembly
  - Flow Control at segment level
  - Error control at segment level
  - Multiplexing & Demultiplexing
- Data Unit: Transport Protocol Data Unit (Segment / User Datagram)

OSI MODEL

- APPLICATION
- PRESENTATION
- SESSION
- TRANSPORT
- **NETWORK**
- DATA LINK
- PHYSICAL

NETWORK LAYER (LAYER-3)

- This layer is responsible for **End to End** delivery.
- Mainly relies on communication subnet
- The main functions are:
  - Logical Addressing
  - Dividing data into Packets/Datagrams
  - Routing
  - Congestion Control
- Data Unit: Packet / Datagram

OSI MODEL

APPLICATION

PRESENTATION

SESSION

TRANSPORT

NETWORK

DATA LINK

PHYSICAL

DATALINK LAYER (LAYER-2)

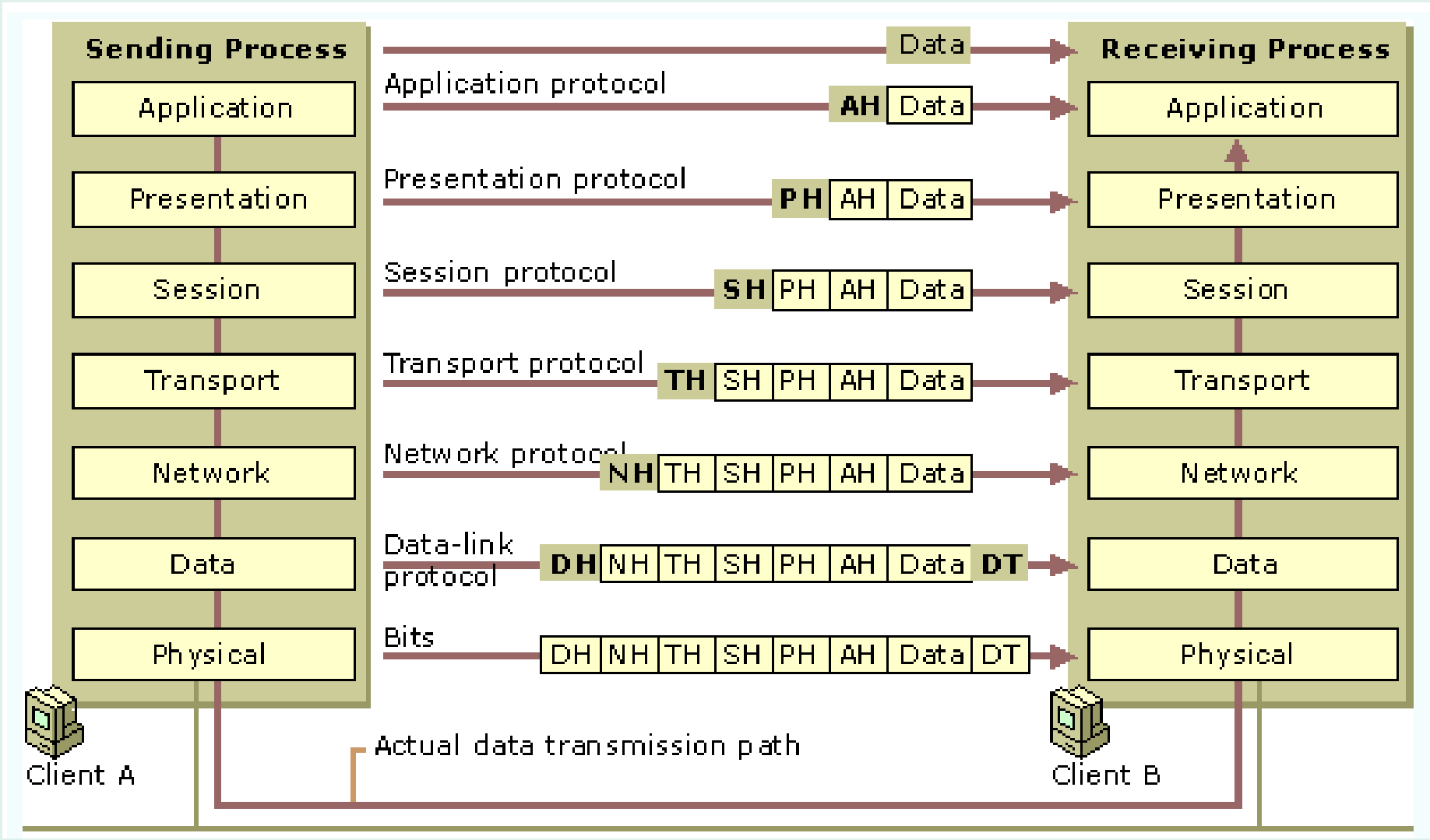
- This layer is responsible for Hop to Hop delivery.
- The main functions are:
  - Physical Addressing
  - Error Control
  - Flow Control at Device Level
  - Access Control
  - Framing
- Data Unit: Frame

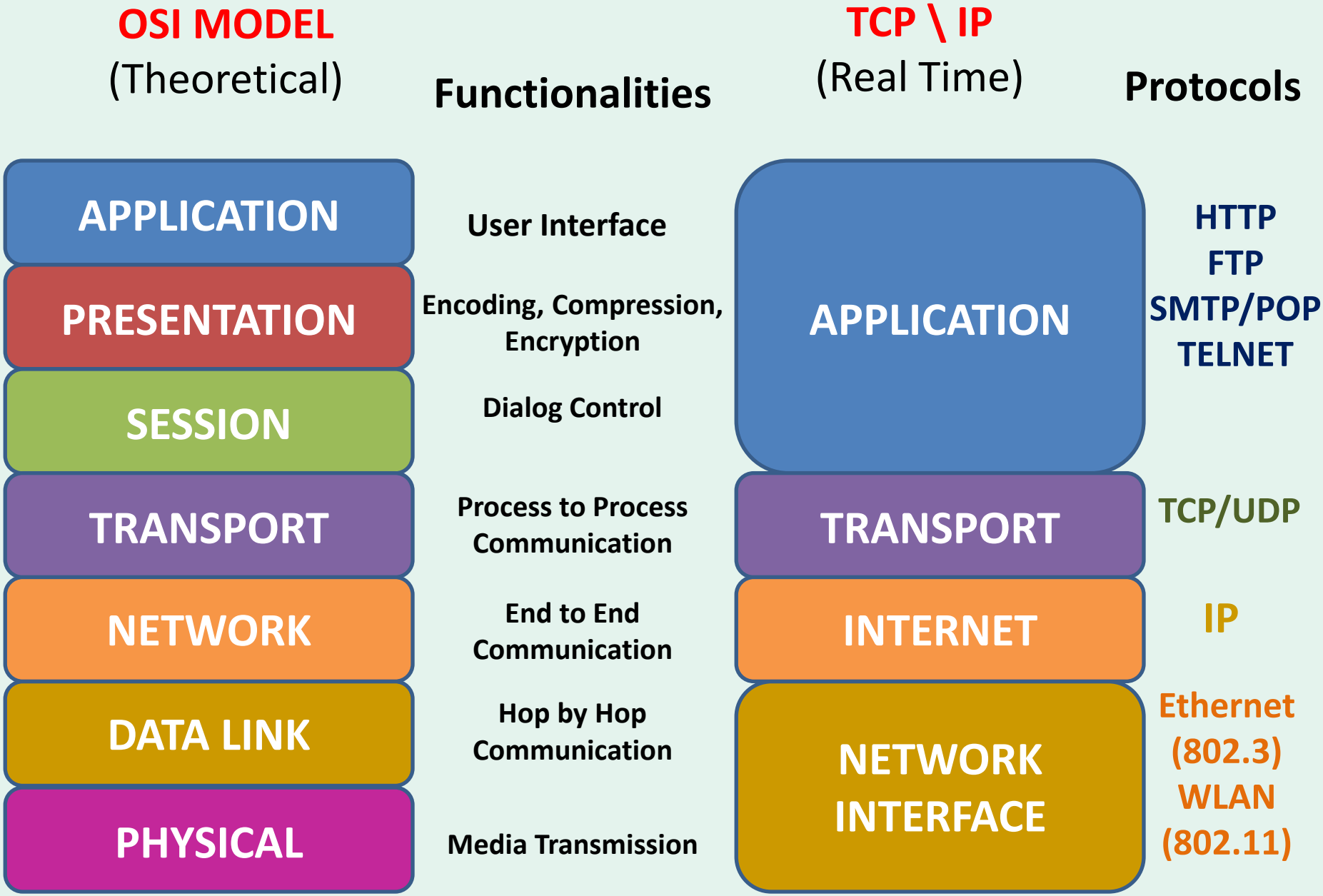
OSI MODEL

- APPLICATION
- PRESENTATION
- SESSION
- TRANSPORT
- NETWORK
- DATA LINK
- **PHYSICAL**

PHYSICAL LAYER (LAYER-1)

- It is responsible for transmitting bit stream over physical medium (i.e. mechanical & electrical specifications of the media)
- The main functions are:
  - Physical characteristics of media
  - Representation of Bits
  - Data transmission rate
  - Synchronization of Bits
- Data Unit: Bits

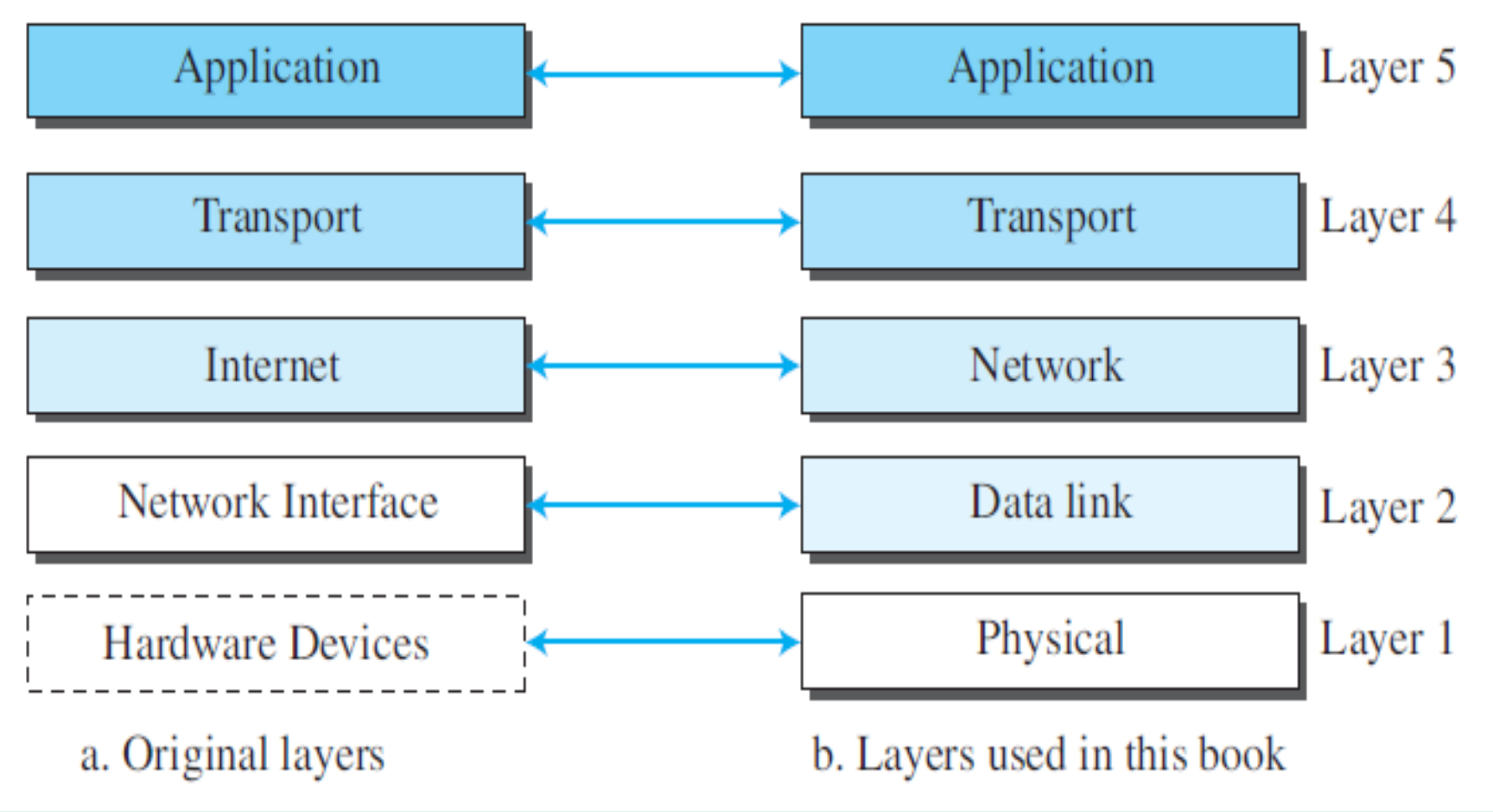

Sending Process
Application
Presentation
Session
Transport
Network
Data
Physical
Data
Application protocol
AH
Data
Presentation protocol
PH
AH
Data
Session protocol
SH
PH
AH
Data
Transport protocol
TH
SH
PH
AH
Data
Network protocol
NH
TH
SH
PH
AH
Data
Data-link protocol
DH
NH
TH
SH
PH
AH
Data
DT
Bits
DH
NH
TH
SH
PH
AH
Data
DT
Receiving Process
Application
Presentation
Session
Transport
Network
Data
Physical
Client A
Actual data transmission path
Client B


OSI MODEL
(Theoretical)
Functionalities
TCP \ IP
(Real Time)
Protocols
APPLICATION
PRESENTATION
SESSION
TRANSPORT
NETWORK
DATA LINK
PHYSICAL
User Interface
Encoding, Compression, Encryption
Dialog Control
Process to Process Communication
End to End Communication
Hop by Hop Communication
Media Transmission
APPLICATION
TRANSPORT
INTERNET
NETWORK INTERFACE
HTTP
FTP
SMTP/POP
TELNET
TCP/UDP
IP
Ethernet (802.3)
WLAN (802.11)

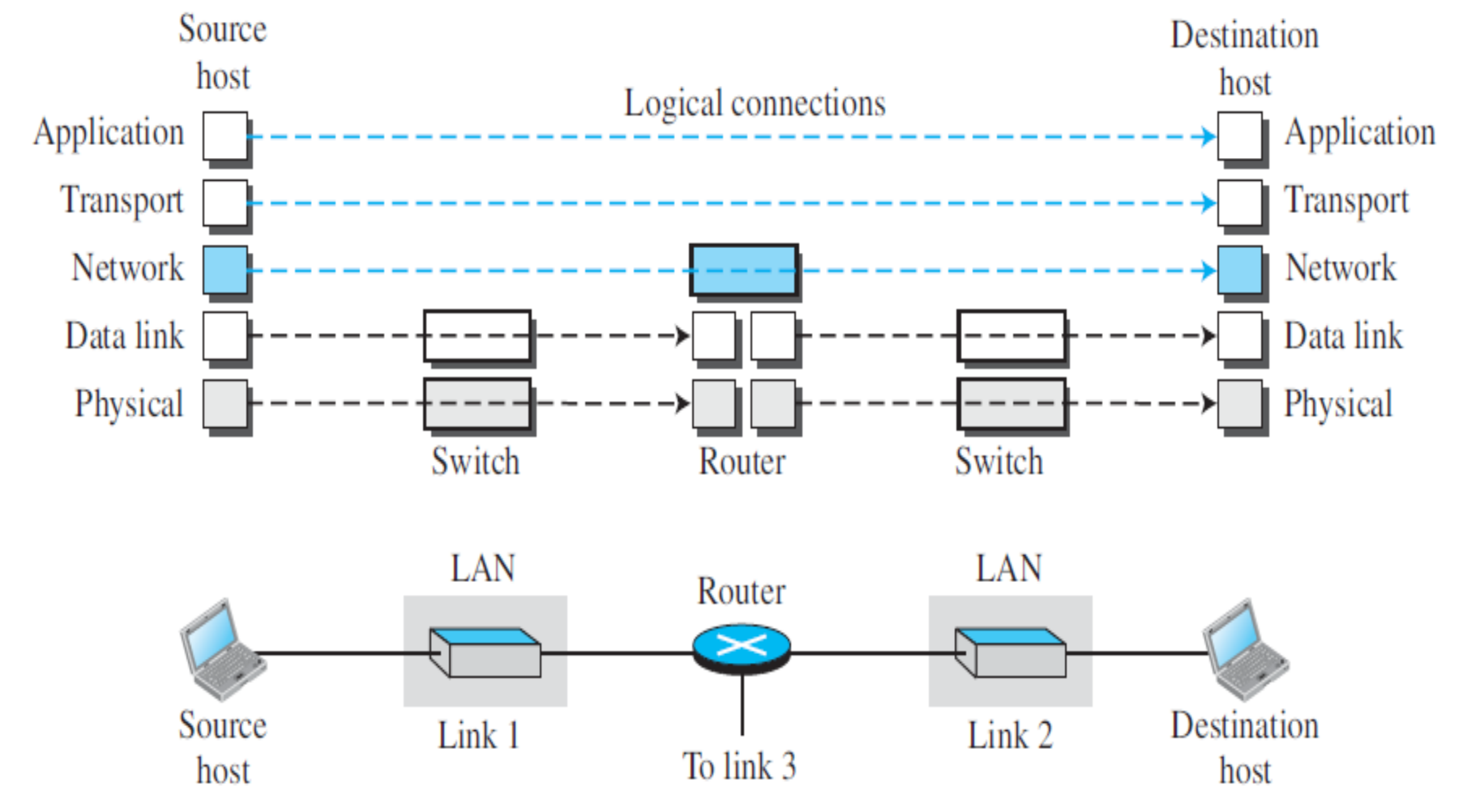
TCP\IP Protocol Suite

- Transmission Control Protocol / Internet Protocol
- Used in the Internet today
- Initially developed as 4 layers but today perceived as 5 layers
- Each Intermediary device is involved with a set of layers depending on the role of the device in the network.
- Functionalities are similar to OSI Model

Layers in TCP/IP



Logical Connection between Layers



Application Layer

- The logical connection between the two application layers is end to end.
- Communication at the application layer is between two processes
- To communicate, a process sends a request to the other process and receives a response.
  - Hypertext Transfer Protocol (HTTP) → Wide Web (WWW).
  - Simple Mail Transfer Protocol (SMTP)→ e-mail
  - File Transfer Protocol (FTP) → Transferring files
  - Terminal Network (TELNET) → remote access
  - Simple Network Management Protocol (SNMP) → Managing Network
  - Domain Name System (DNS) → Finding Network layer Address
  - Internet Group Management Protocol(IGMP) → Groups

Transport layer

- The logical connection between the two transport layers is end to end.
- Giving services to application layer
- Services are either Connection Oriented or Connection less (Based on the Application)
- Transmission Control Protocol (TCP)→ Connection Oriented
  - Flow control , Error control, congestion control
- User Datagram Protocol (UDP) → connectionless protocol
- Stream Control Transmission Protocol (SCTP) → One to Many applications

Network Layer

- The logical connection between two network layers is end-to-end.
- Responsible for Routing i.e. choosing best path for in a network
- Internet Protocol (IP) → Logical addressing
- Internet Control Message Protocol (ICMP) → problems in routing
- Internet Group Management Protocol (IGMP) → managing groups
- Dynamic Host Configuration Protocol(DHCP) → get IP address dynamically
- The Address Resolution Protocol (ARP) → finding link address for a given logical address

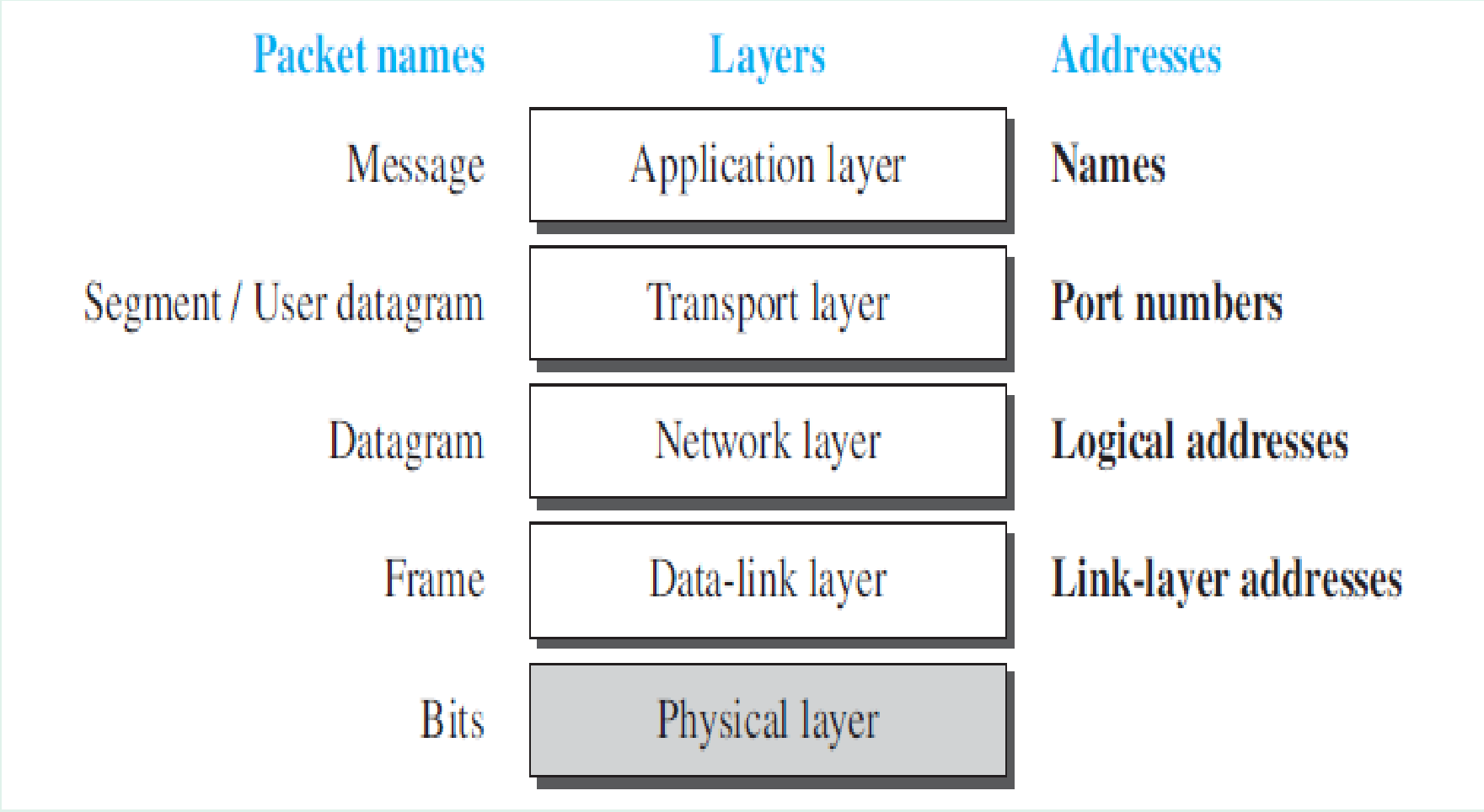
Data link Layer

- The logical connection between two data link layers is hop-to-hop.
- Responsible for link transmission
- Routing select best links from source to destination
- Provides error detection/correction
  - Ethernet → IEEE 802.3 Protocol
  - Wireless LAN→ IEEE 802.11 Protocol
  - Token Ring → IEEE 802.5 Protocol
  - Token Bus → IEEE 802.4 Protocol
  - Bluetooth → IEEE 802.15 Protocol

Physical Layer

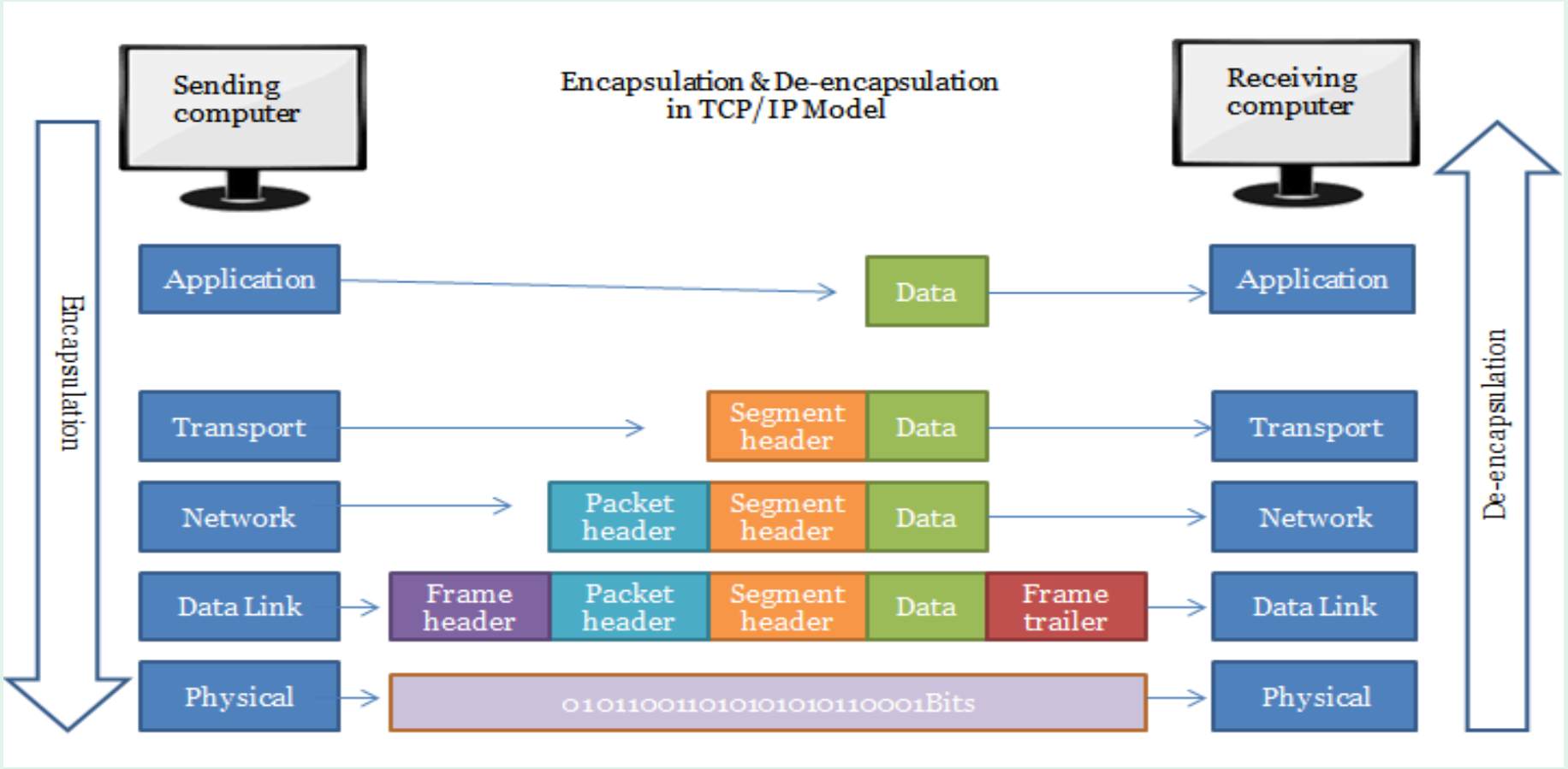
- The logical connection between two physical layers is hop-to-hop.
- The physical layer is responsible for carrying individual bits in a frame across the link
- The bits received in a frame from the data-link layer are transformed and sent through the transmission media
- The logical unit between two physical layers in two devices is a ‘bits’
- There are several protocols that transform a bit to a signal

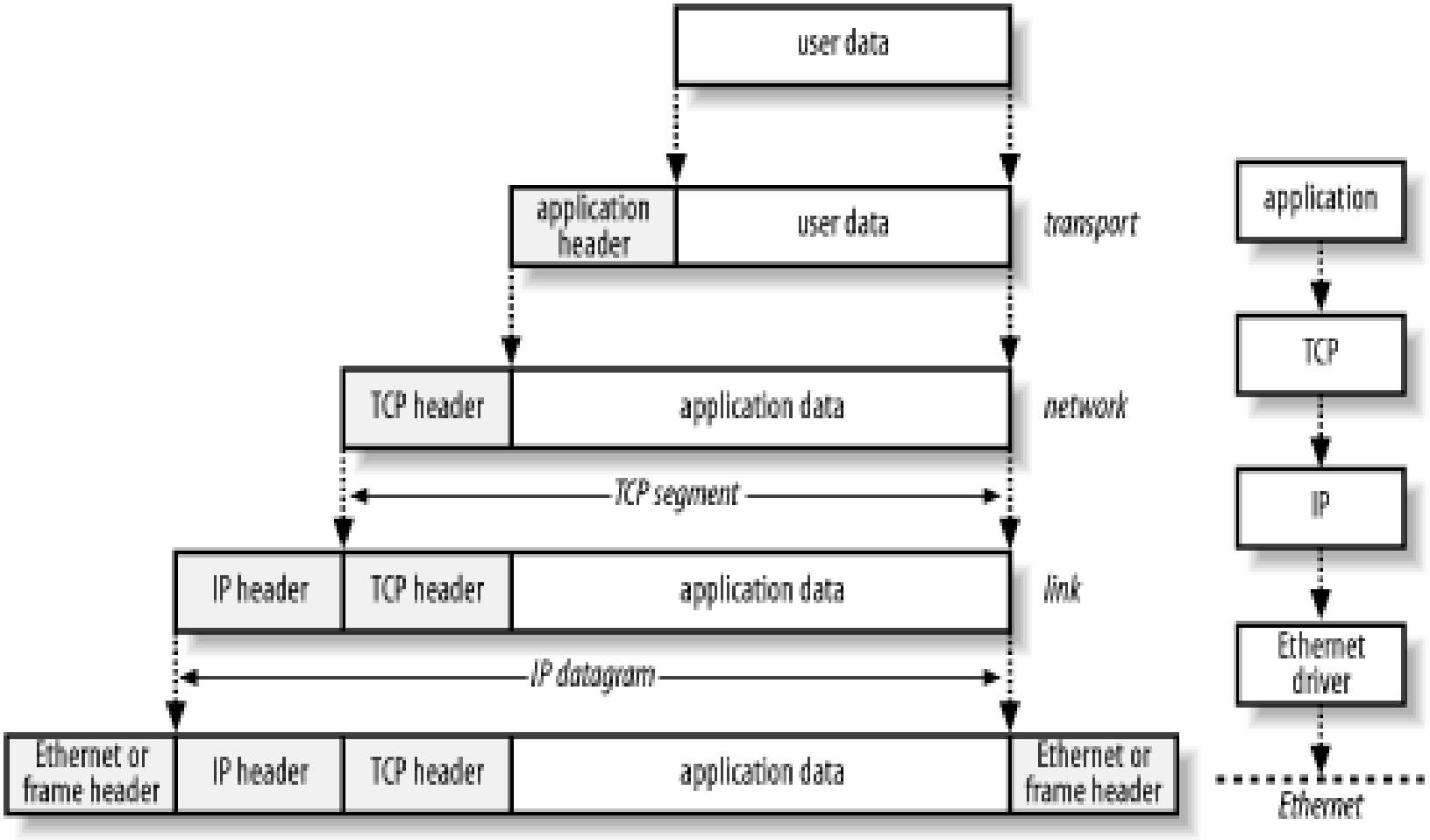
Addressing

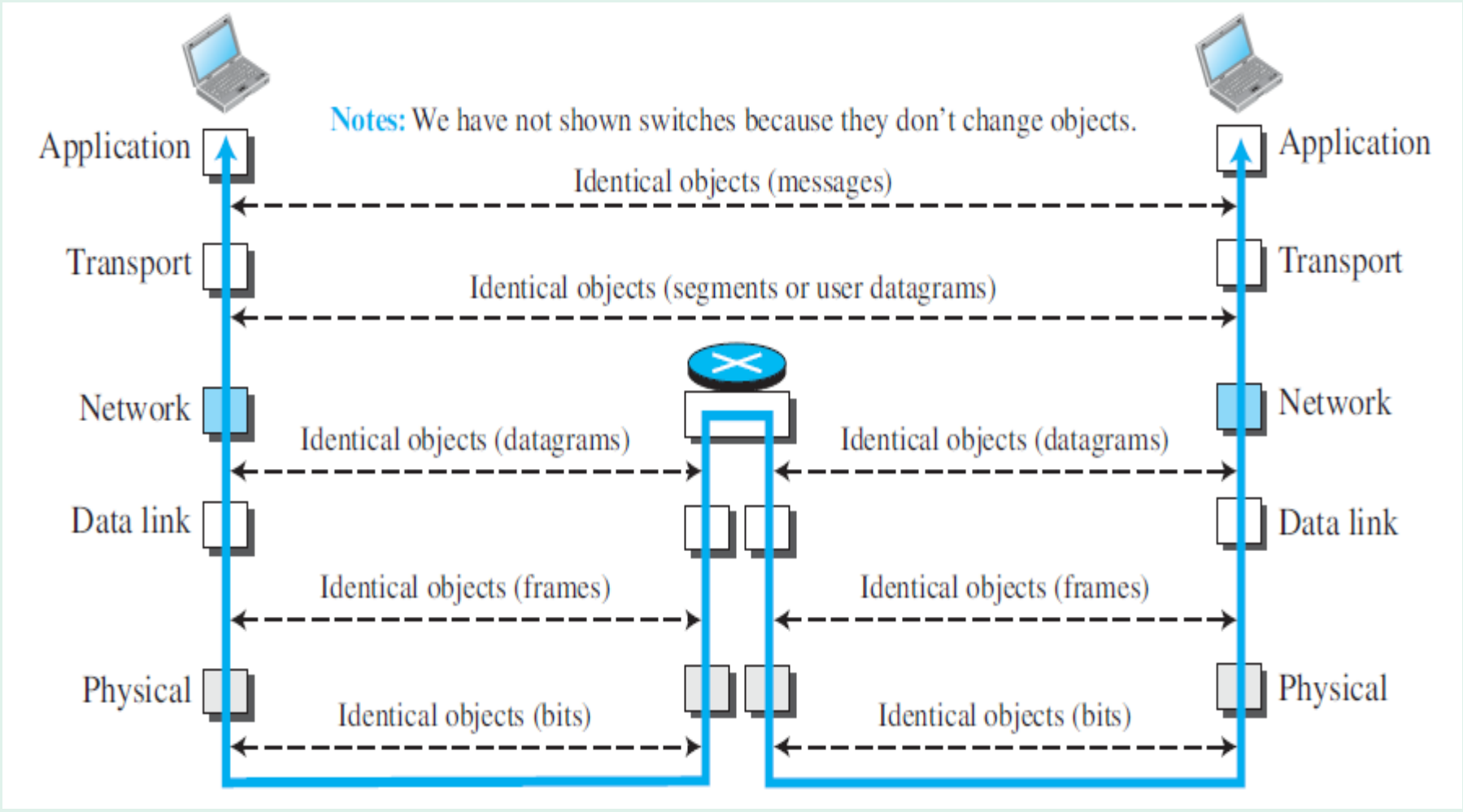

Packet names
Layers
Addresses
Message
Application layer
Names
Segment / User datagram
Transport layer
Port numbers
Datagram
Network layer
Logical addresses
Frame
Data-link layer
Link-layer addresses
Bits
Physical layer

Encapsulation / Decapsulation

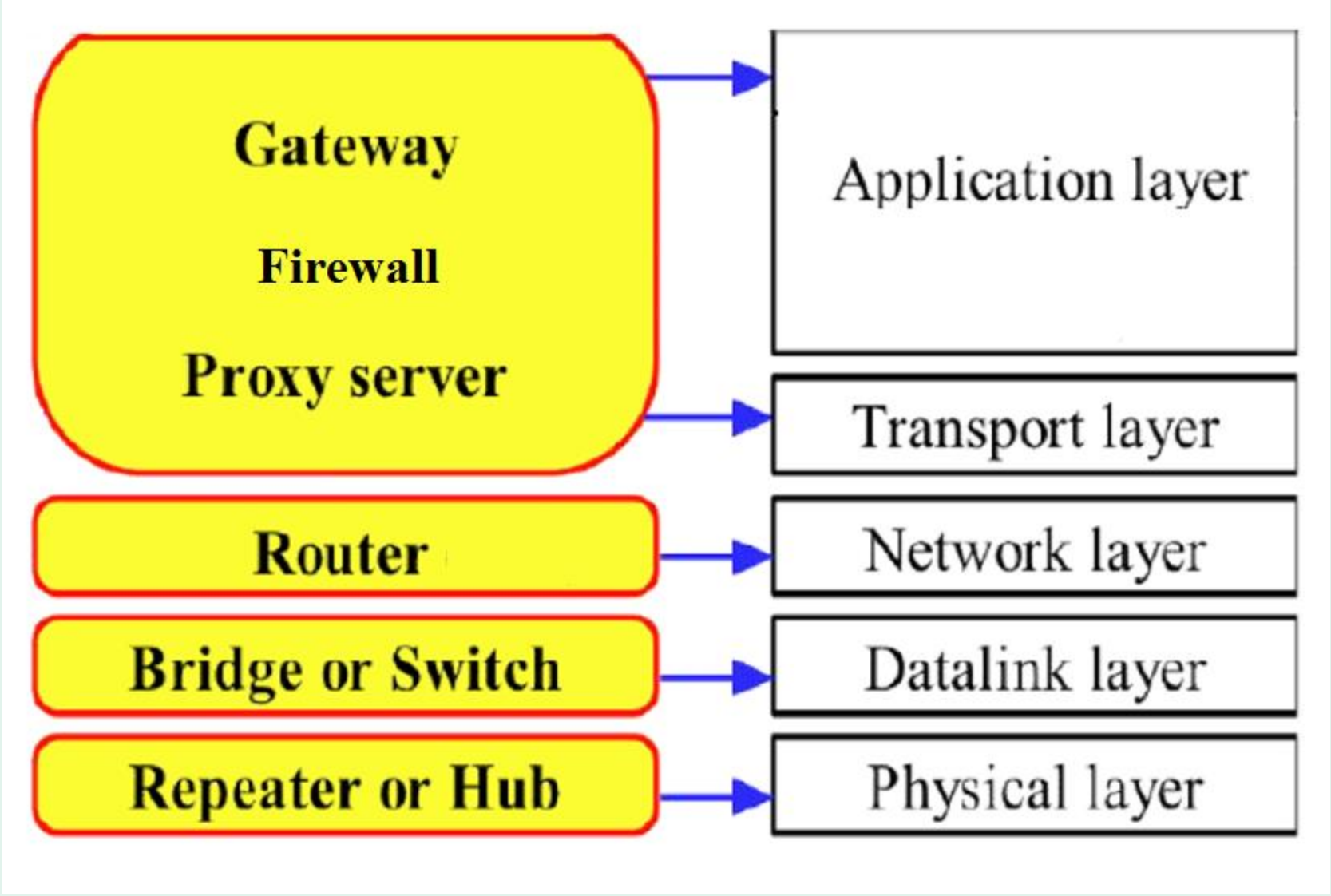
It is used to describe a process of adding / removing headers and trailers around some data.



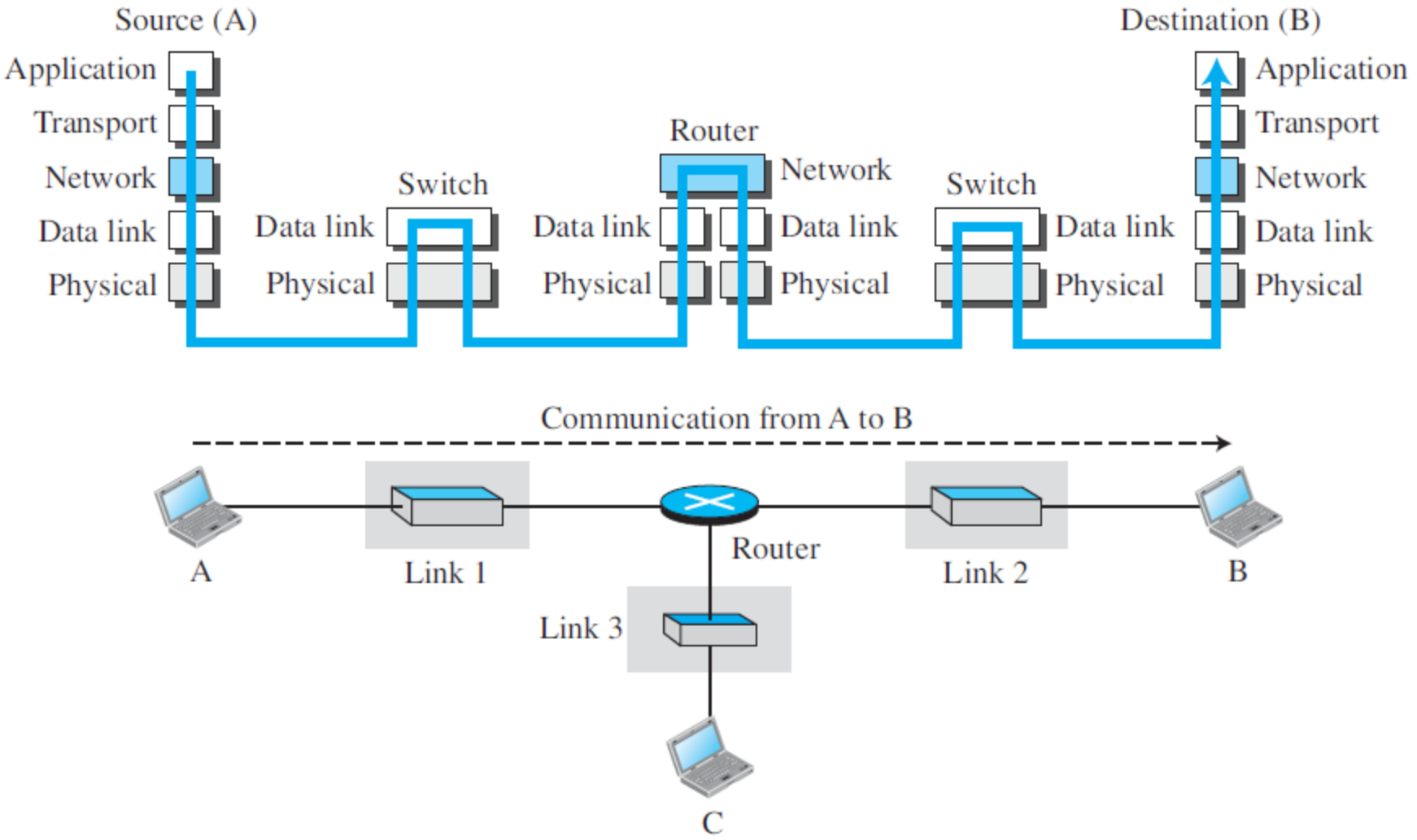

user data
application header
user data
transport
TCP header
application data
network
TCP segment
IP header
TCP header
application data
link
IP datagram
Ethernet or frame header
IP header
TCP header
application data
Ethernet or frame header
application
TCP
IP
Ethernet driver
Ethernet

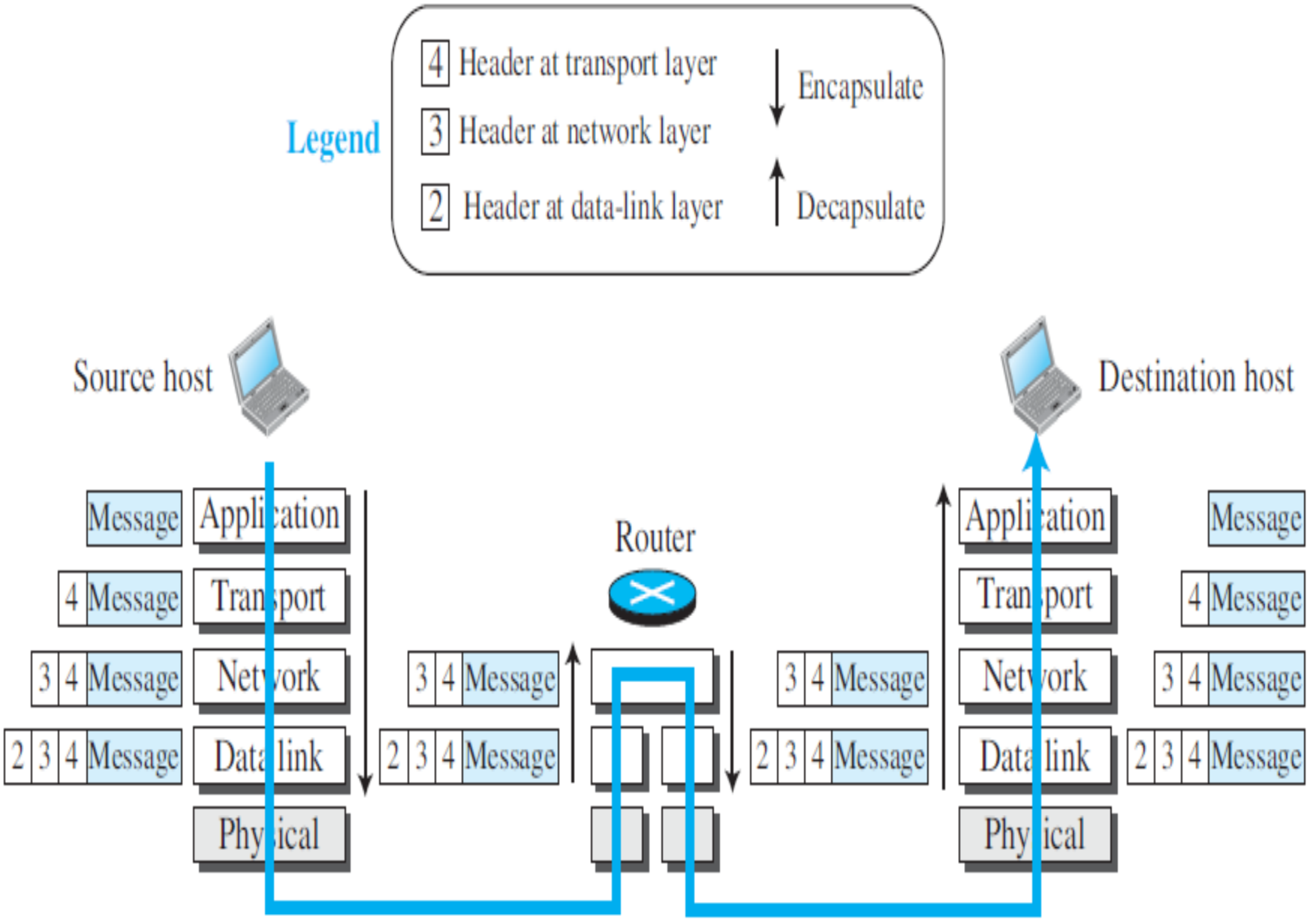

Notes: We have not shown switches because they don't change objects.
Application
Application
Identical objects (messages)
Transport
Transport
Identical objects (segments or user datagrams)
Network
Network
Identical objects (datagrams)
Identical objects (datagrams)
Data link
Data link
Identical objects (frames)
Identical objects (frames)
Physical
Physical
Identical objects (bits)
Identical objects (bits)

Intermediary Devices



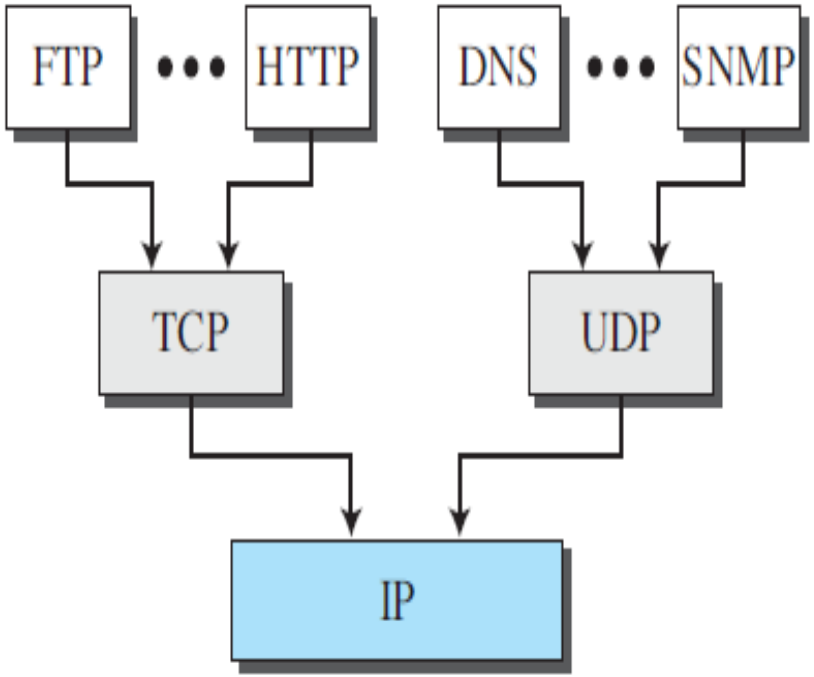
Communication through Internet



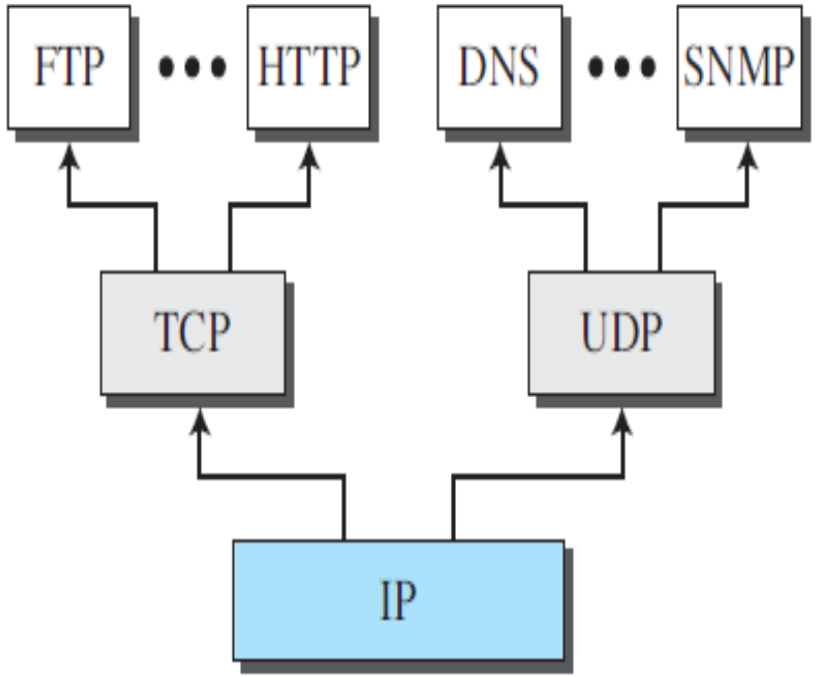

Legend
4 Header at transport layer
3 Header at network layer
2 Header at data-link layer
Encapsulate
Decapsulate
Source host
Destination host
Router
Message
Application
4 Message
Transport
3 4 Message
Network
2 3 4 Message
Data link
Physical

Multiplexing / Demultiplexing

- Process of simultaneously sending data of several upper layer protocols together



a. Multiplexing at source



b. Demultiplexing at destination

OSI v/s TCP/IP

- The OSI has seven layers while the TCP/IP has four layers.
- OSI model is a generic model that is based upon functionalities of each layer. TCP/IP model is a protocol-oriented standard.
- OSI model distinguishes the three concepts, namely: services, interfaces, and protocols. TCP/IP does not have a clear distinction between these three.
- OSI model gives guidelines on how communication needs to be done, while TCP/IP protocols layout standards on which the Internet was developed. So, TCP/IP is a more practical model.
- In OSI, the model was developed first and then the protocols in each layer were developed. In the TCP/IP suite, the protocols were developed first and then the model was developed.

Lack of OSI Model's Success

- OSI was completed when TCP/IP was fully in place and a lot of time and money had been spent on the suite; changing it would cost a lot.
- Some layers in the OSI model were never fully defined i.e. Presentation & Session Layer
- It did not show a high enough level of performance to entice the Internet authority