Data Communications and Networking Fourth Edition

Forouzan

# Chapter 2

# Network Models

# Review of Data Communications

- **What is communication?**
  - **The exchange of information between individuals using a common set of symbols, signs, behavior or language.**
- **What is data communication?**
  - **The exchange of data between two devices via some form of transmission medium such as a wire cable**
- **What is Computer Network?**
  - **A set of devices connected by communication links**
    하드웨어와 소프트웨어, 케이블링의 조합으로 여러 컴퓨터 장치들이 서로 통신할 수 있게끔 하는 것 (의미)
- **What is basic components of data communication system?**
  - **Message/Sender/Receiver/Medium/Protocol**
- **What is the difference between Data Communication and Network?**
  - **Computer Network=데이터 통신 + network Intelligence (topology)**

# Protocol Concepts

**Jin Seek Choi**

jinseek@hanyang.ac.kr

**Recommended Reading**

*nms.lcs.**mit**.edu/6829-papers/darpa-**internet**.pdf*

# 목표 (Objectives)

- 목표
  - What is protocol?
  - Layering
  - End-to-End Arguments –encapsulation
  - A Case Study: the Internet (TCP/IP Layering)

- 질문
  - Protocol이란 무엇인가?
  - Internet Protocols이란?
  - Internet Protocols의 특징은?
  - Internet Protocols의 기능을 계층별로 설명하시오?

# *Protocols and Standards

- **Protocol:** **A set of rules** **that governs data communication** 다른 종류의 정보기기**(entity)** 사이의 원활한 통신을 가능하게 하는 약속된 통신 규약 **(format, order of messages, actions)**

- 표준 **Protocols**
  - **TCP(Transmission Control Protocol)**
  - **IP(Internet Protocol)**
- 표준화 기구
  - **ITU-T**
  - **IETF**
  - **ISO**
  - **FCC**
  - **IEEE**

# What's a protocol?

**human protocols:**

- "what's the time?"
- "I have a question"
- introductions

… **specific** messages sent

… **specific** actions taken when messages received, or other events

**Internet protocol:**

- machines rather than humans
- all communication activity in Internet governed by network protocols

***Network protocols define format, order of messages sent and received among network entities, and actions taken on message transmission, receipt***

# 2-2~2.3 An example of ISO Protocol

## The OSI Model

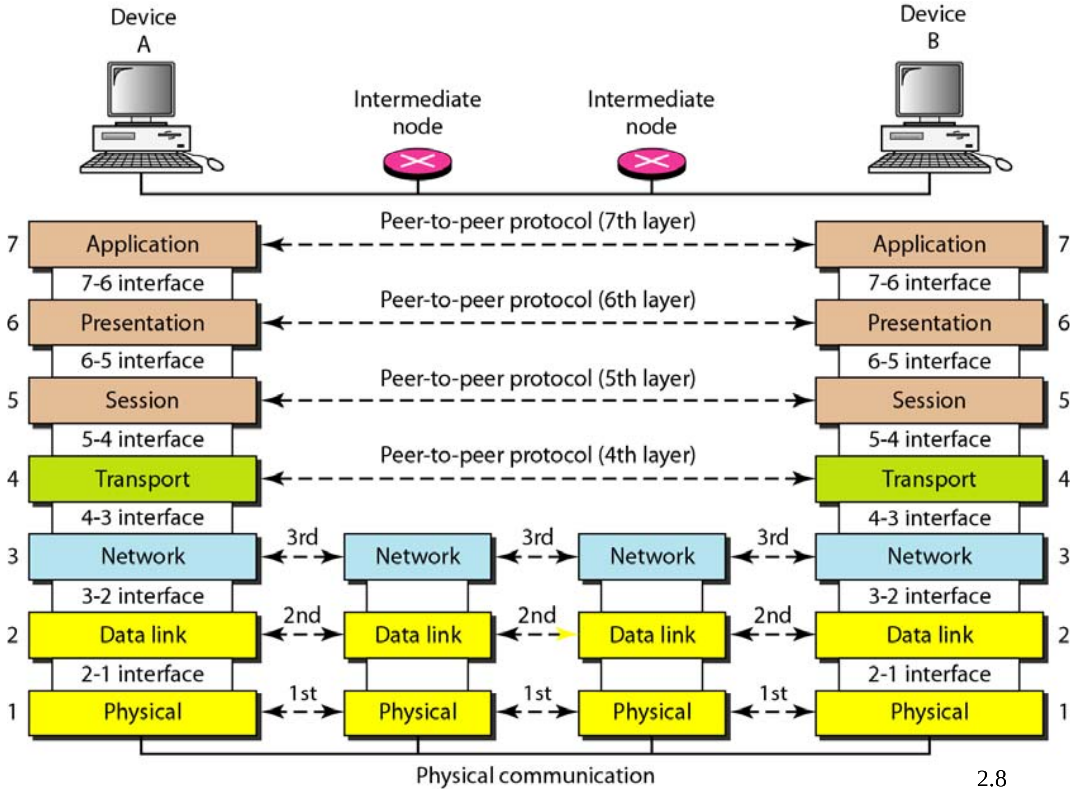
## Layers in the Model

# 2-2 THE OSI MODEL

*Established in 1947, the International Standards Organization (ISO) is a multinational body dedicated to worldwide agreement on international standards. An ISO standard that covers all aspects of network communications is the Open Systems Interconnection (OSI) model. It was first introduced in the late 1970s.*

## *Topics discussed in this section:*

Layered Architecture
Peer-to-Peer Processes
Encapsulation



# 2-2 THE OSI MODEL (2)

## Layered & Encapsulation

Application Layer

Presentation Layer

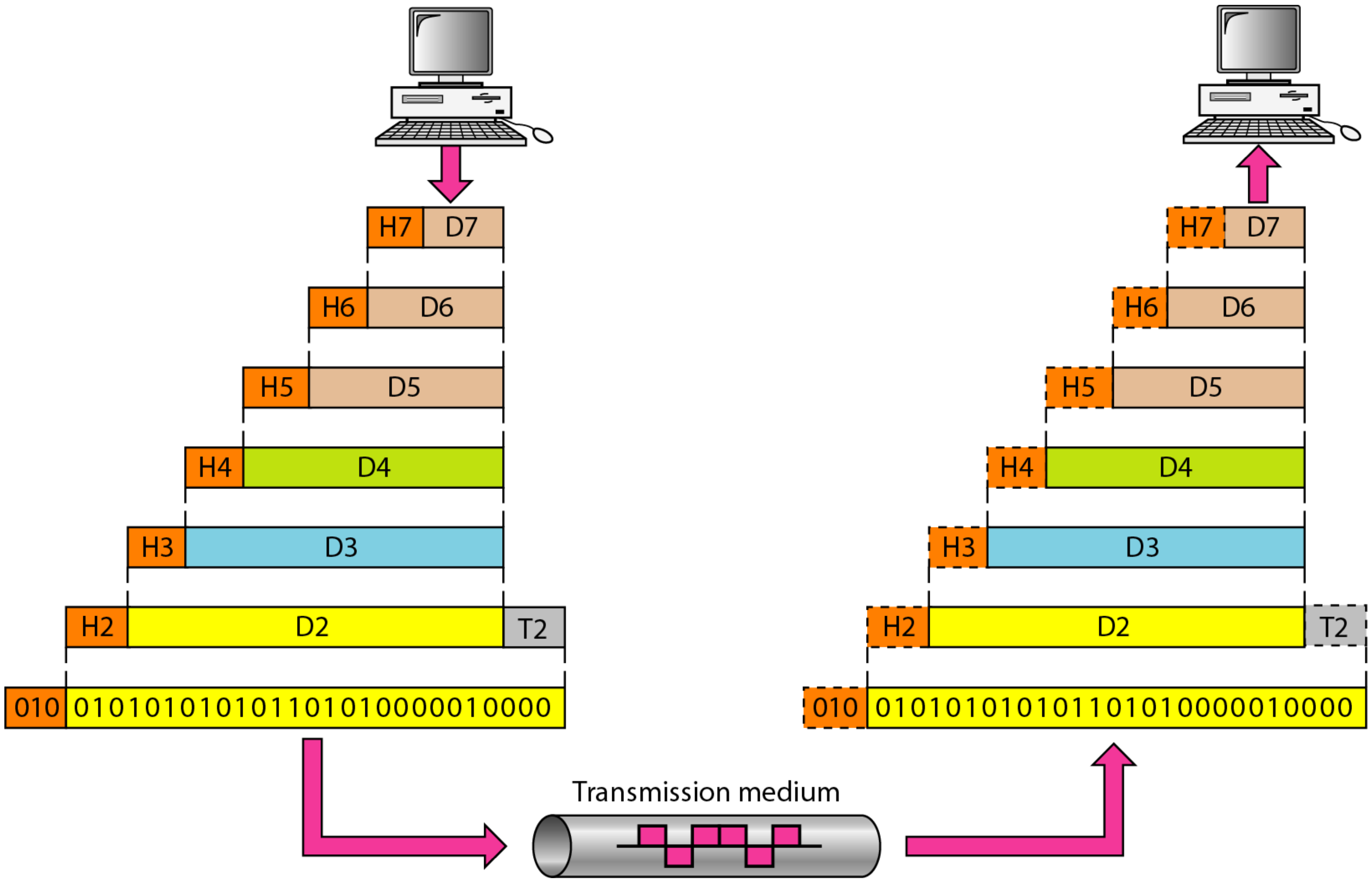
Session Layer

Transport Layer

Network Layer

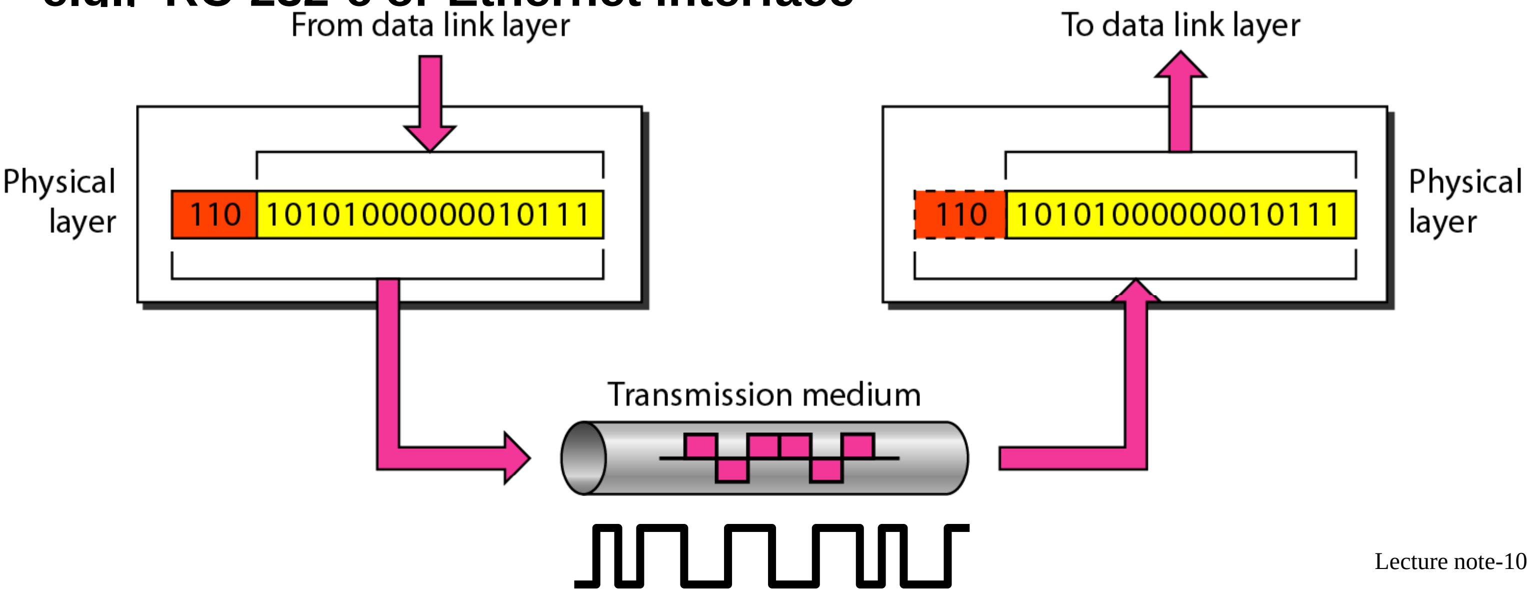
Data Link Layer

Physical Layer



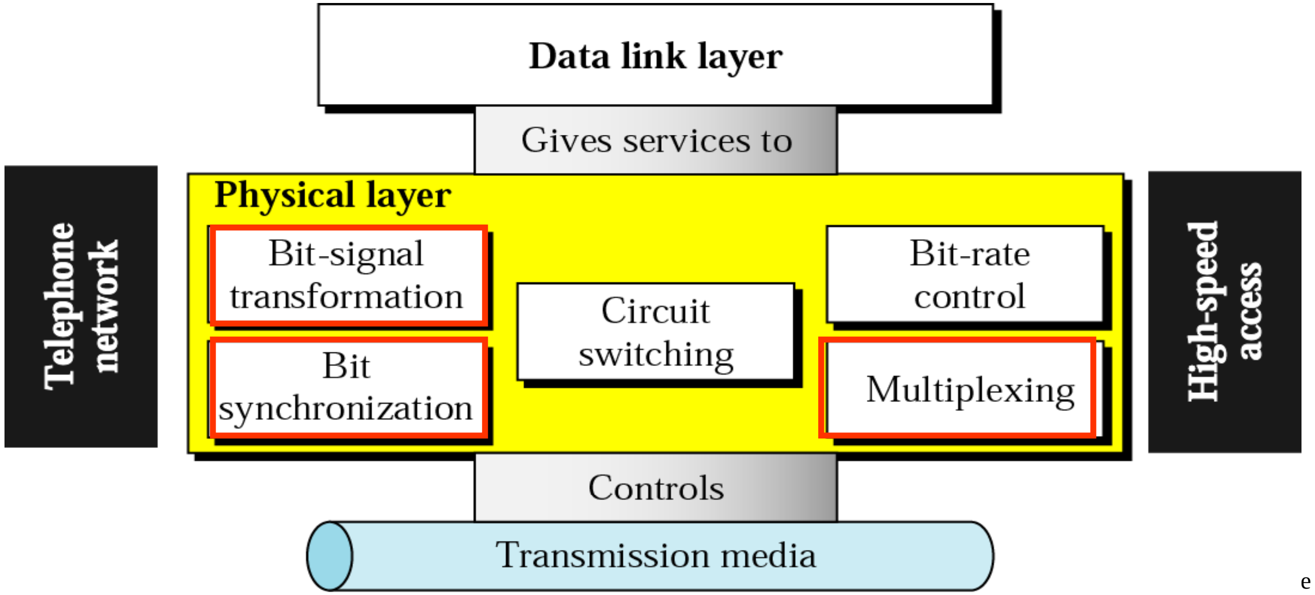
# PHYSICAL LAYER

- **Responsible for transmitting individual bits from one node to the next**
- **Function**
  - **Physical characteristics of interfaces and media**
  - **Representation of bits**
  - **Data rate**
  - **Synchronization of bits**
- **e.g., RC-232-c or Ethernet interface**



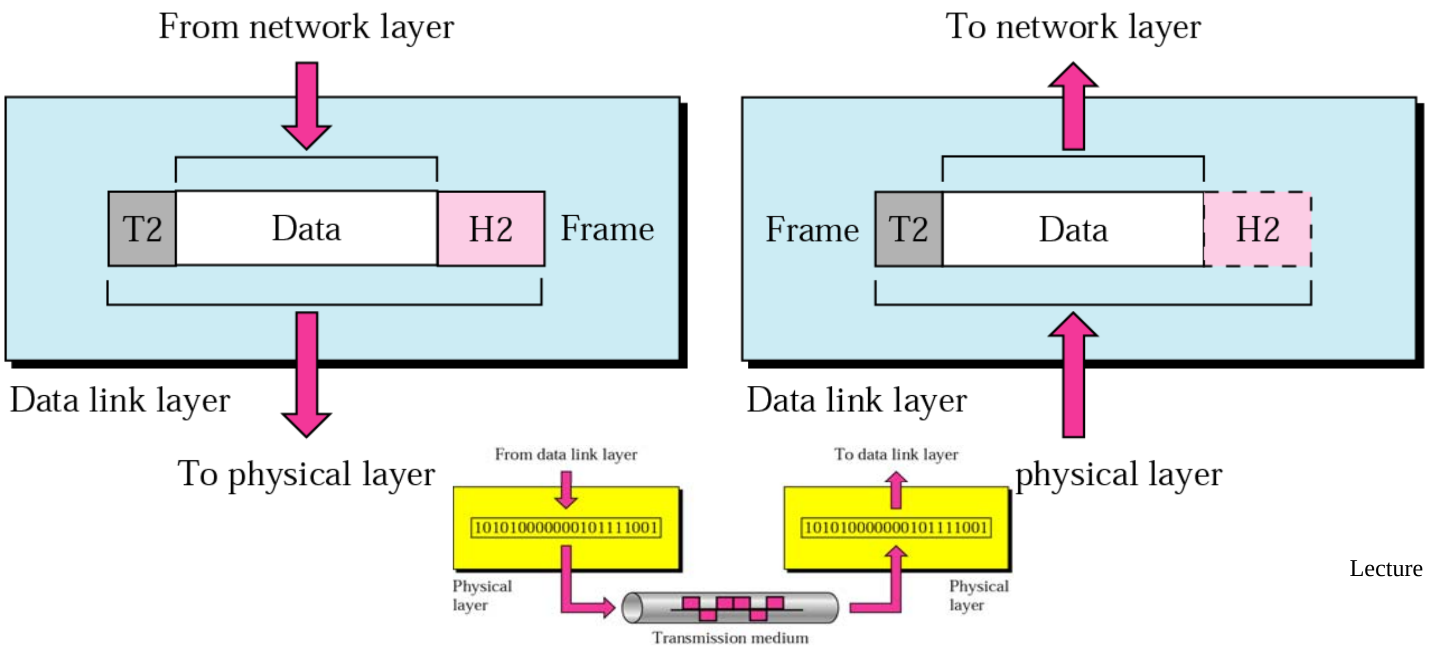
# Function of PHYSICAL LAYER

- Responsible for transmitting individual bits from one node to the next
- Function
  - Physical characteristics of interfaces and media (hardware and circuit)
    - ➢ Representation of bits (signal),
    - ➢ Data rate
    - ➢ Synchronization of bits
    - ➢ multiplexing,..



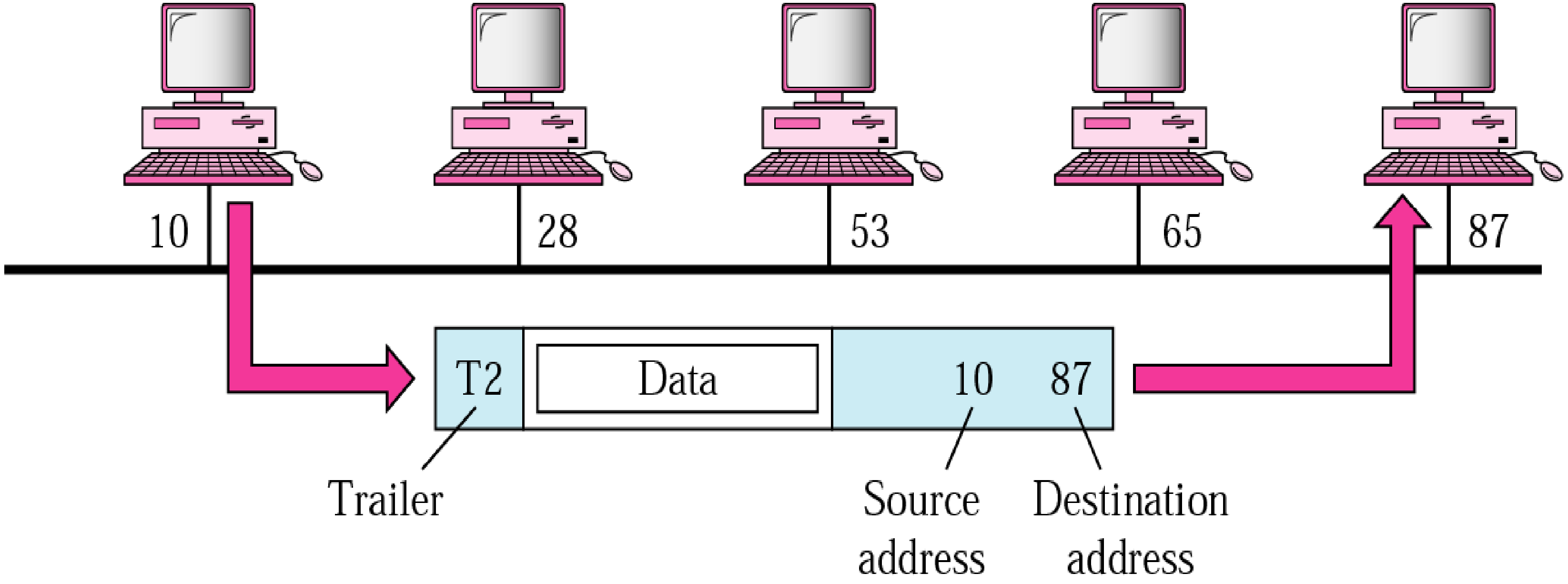
# DATA LINK LAYER

- Responsible for transmitting frames from one node to the next
- Function
  - Framing:
  - Physical addressing
  - Flow control
  - Error control
  - Access control

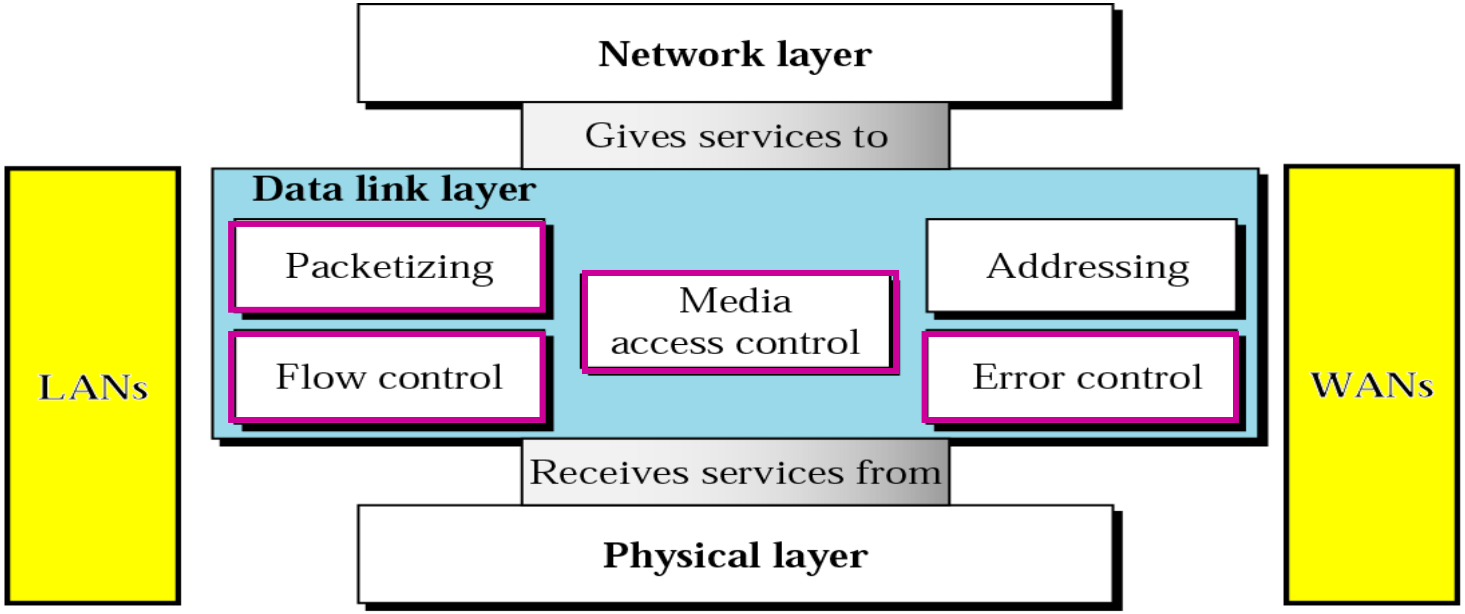


# *Example 1*

**In following Figure a node with physical address 10 sends a frame to a node with physical address 87. The two nodes are connected by a link. At the data link level this frame contains physical addresses in the header. These are the only addresses needed. The rest of the header contains other information needed at this level. The trailer usually contains extra bits needed for error detection**



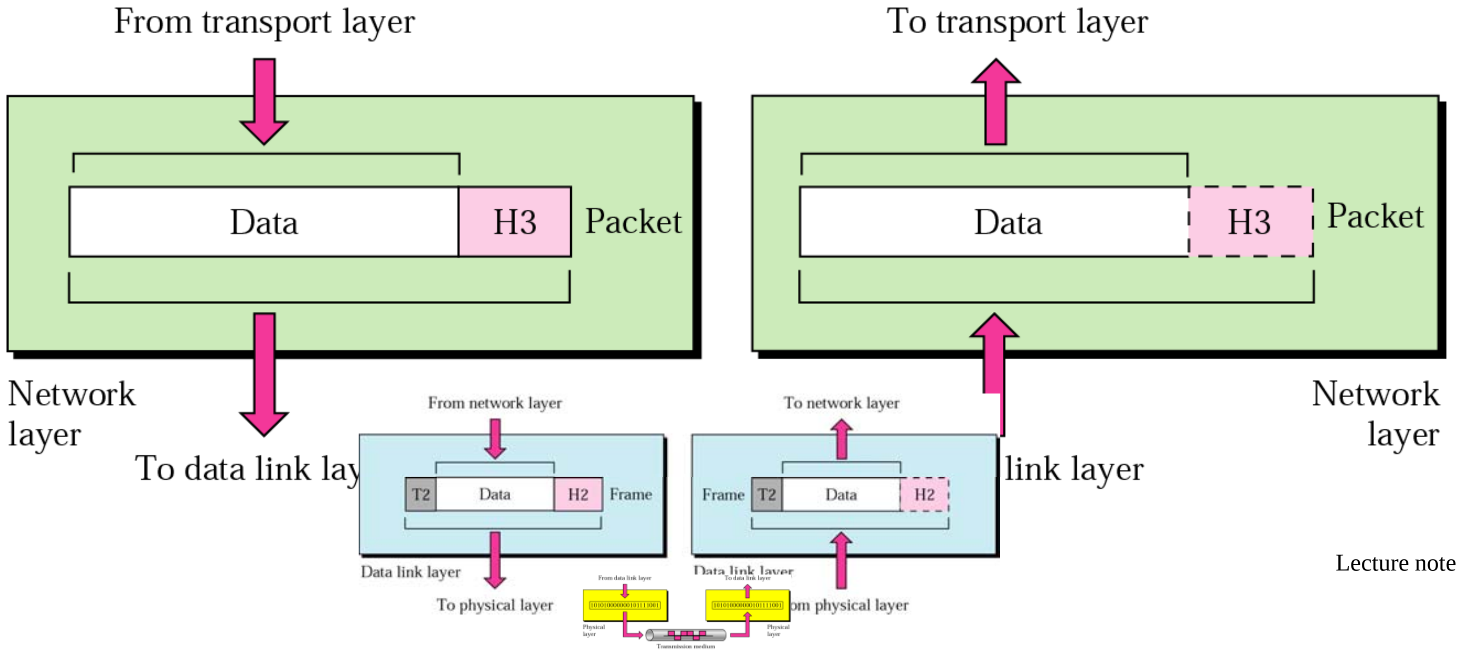
# Function of Data Link layer



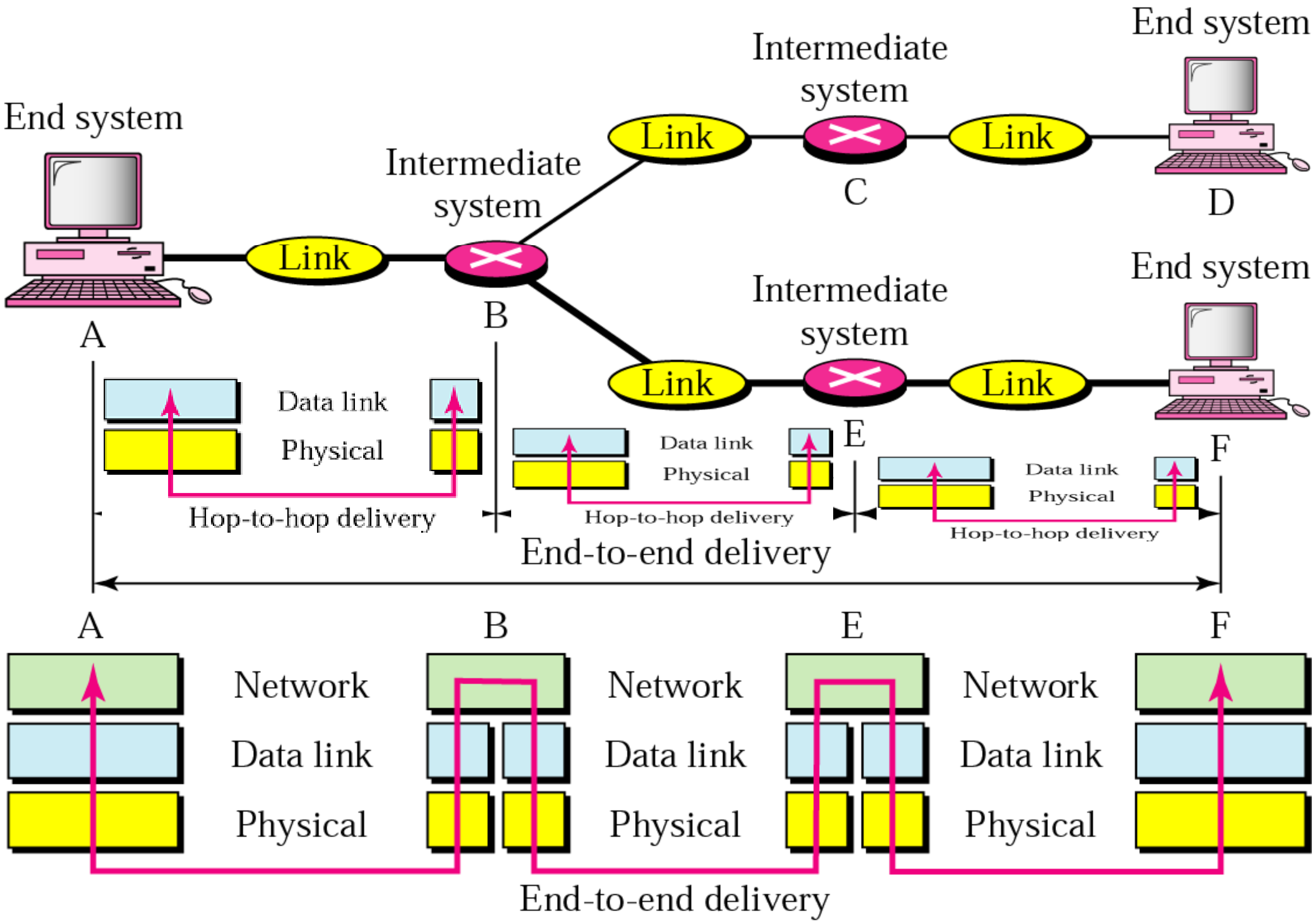
- **Data link protocols have three functions:**
  - Error Control**: Detecting and correcting transmission errors. (Error & flow)**
  - Media Access Control**: Controlling when computers transmit. Who should send now(Access control)**
  - Message Delineation**: Identifying the beginning and end of a message. (Packetizing & Addressing)**

# NETWORK LAYER

- Source-to-destination (node-to-node) delivery,
- Responsible from the delivery of packets from the original source to the final destination
- Functions
  - Logical addressing
  - routing

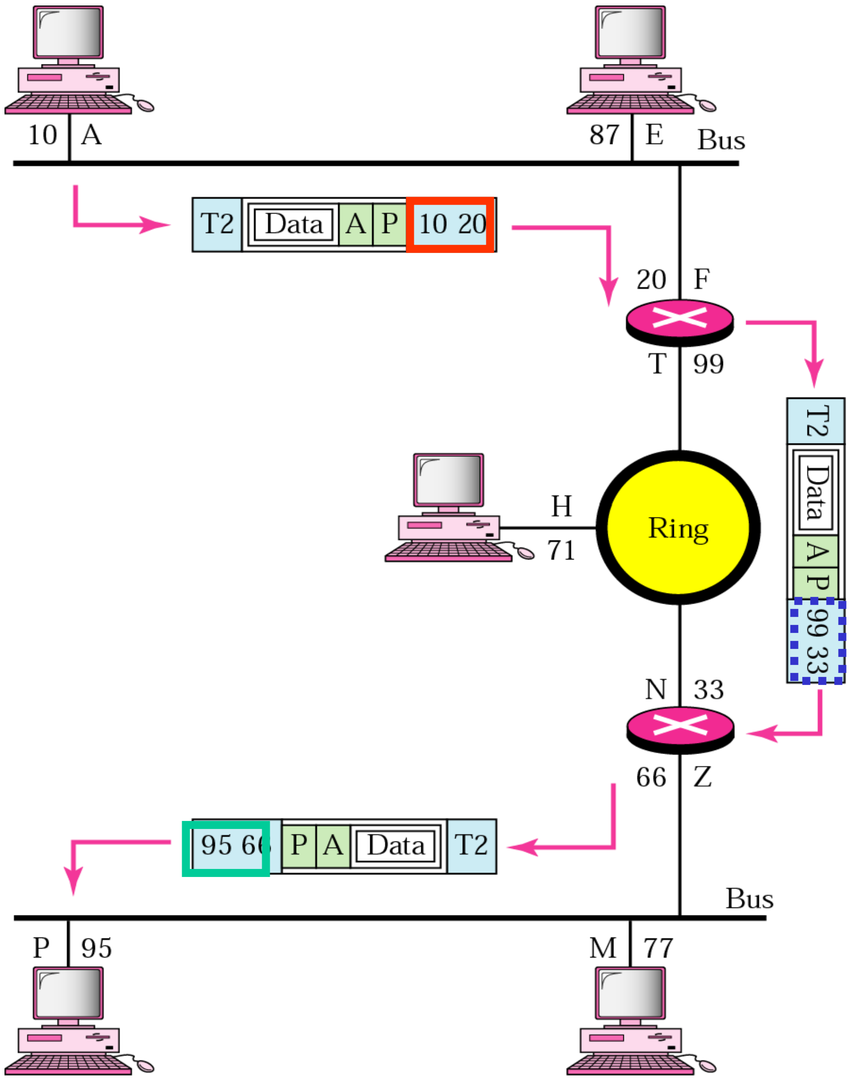


# SOURCE – TO – DESTINATION DELIVERY

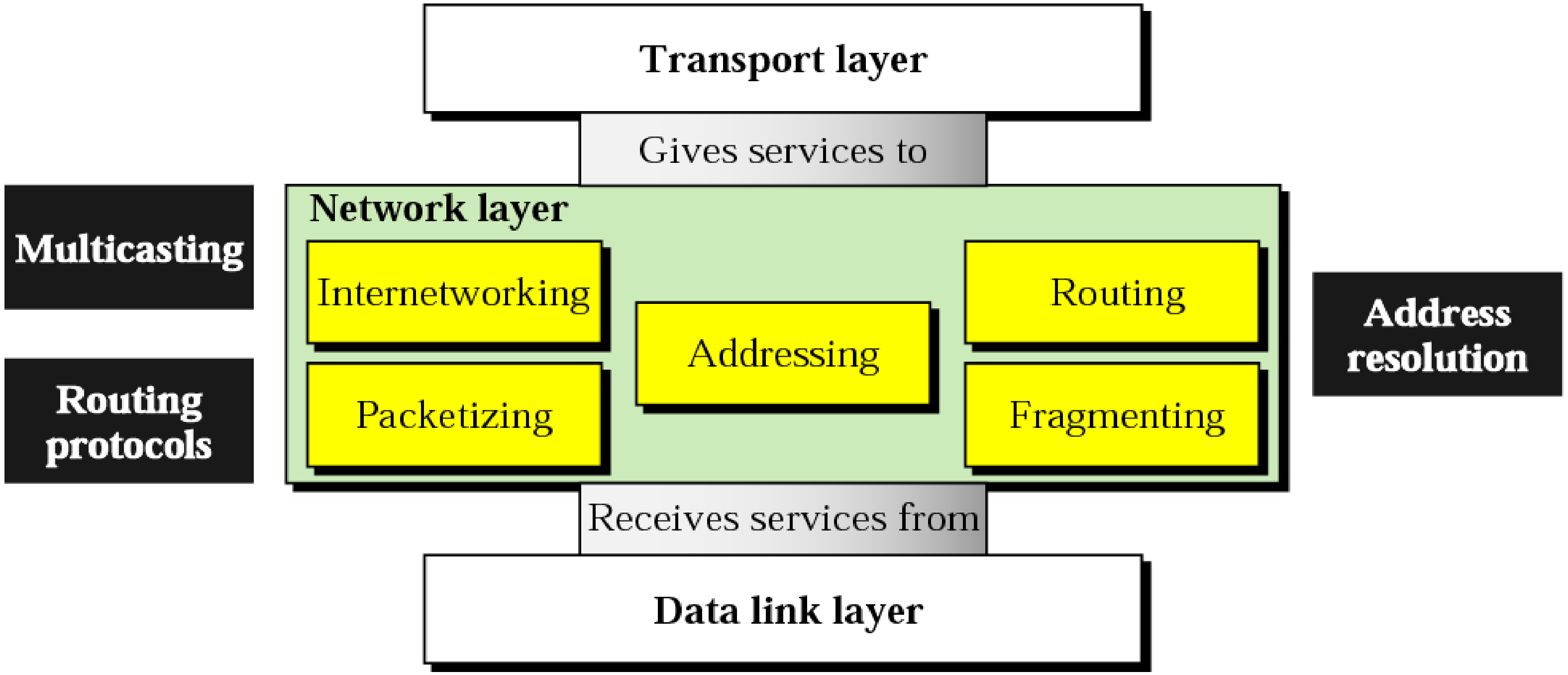


# *Example 2*

**In the Figure we want to send data from a node with network address A and physical address 10, located on one LAN, to a node with a network address P and physical address 95, located on another LAN. Because the two devices are located on different networks, we cannot use physical addresses only; the physical addresses only have local jurisdiction. What we need here are universal addresses that can pass through the LAN boundaries. The network (logical) addresses have this characteristic.**



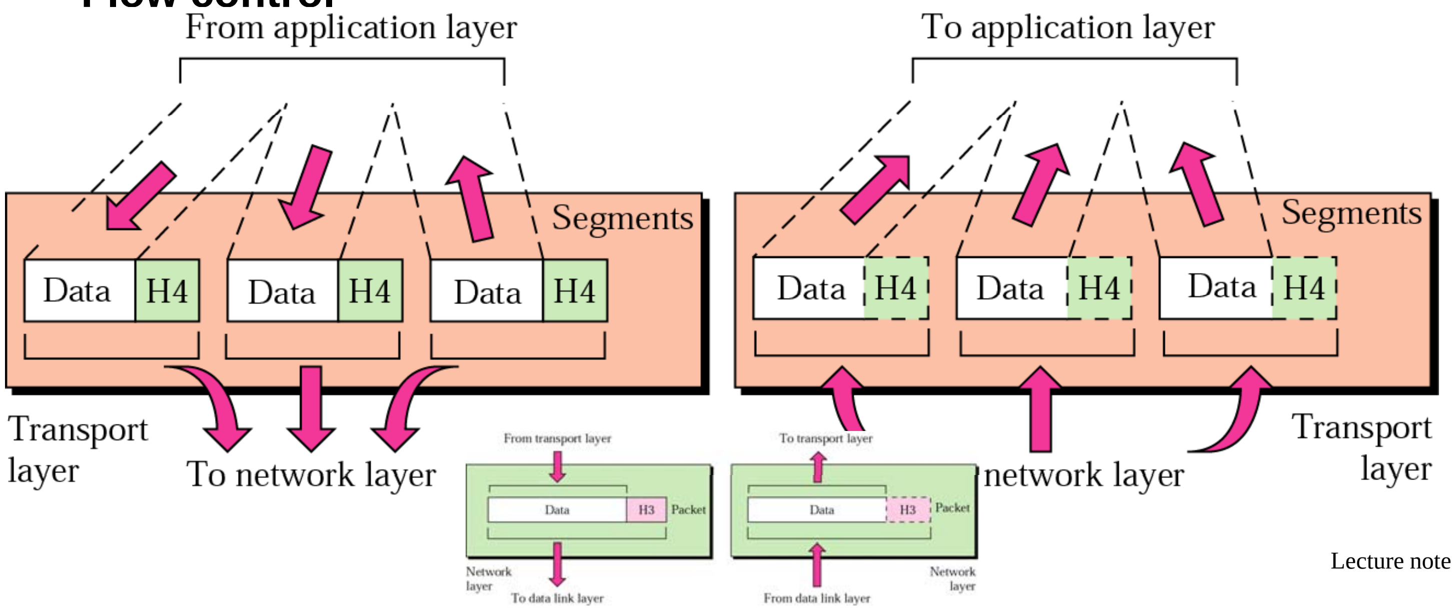
# Function of Internet layer



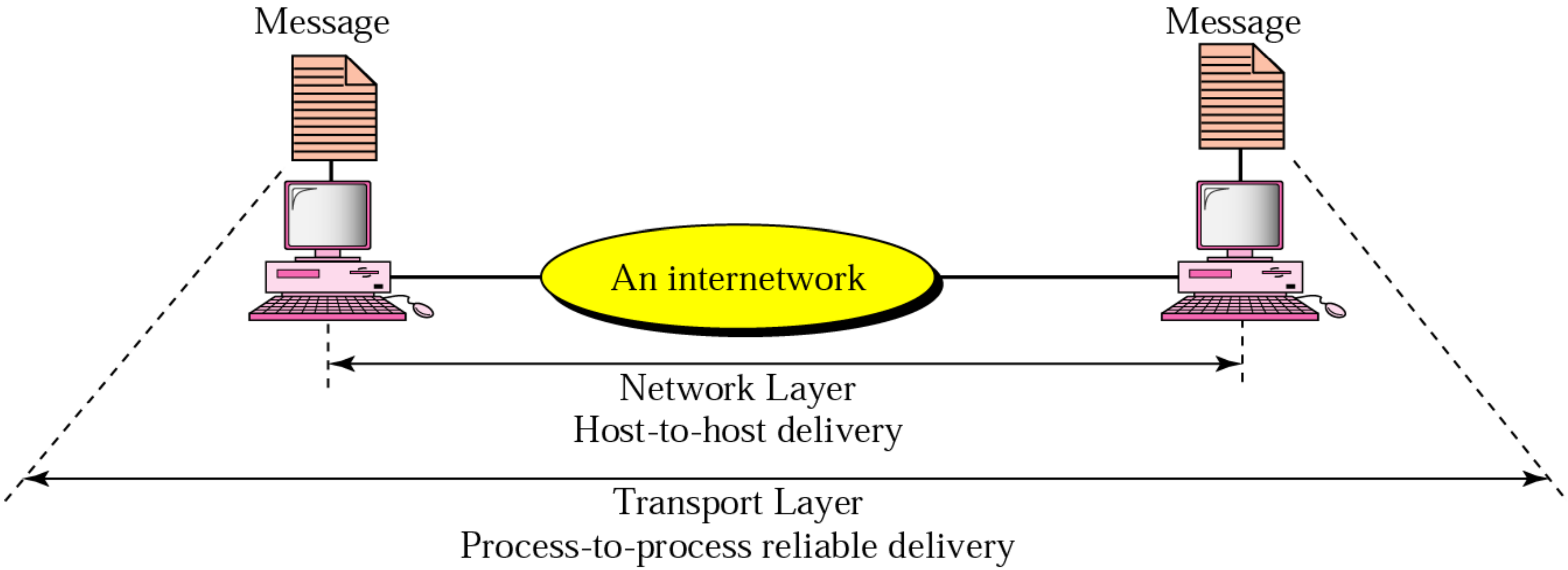
- *IP Protocol* (host-to-host delivery).
  - global addresses
  - best-effort datagrams
  - stateless gateways
  - inter-network

# TRANSPORT LAYER

- Process-to- process delivery
- Responsible for delivery of a message from one process to another
- Functions
  - Port addressing
  - Segmentation and reassembly
  - Connection control
  - Flow control

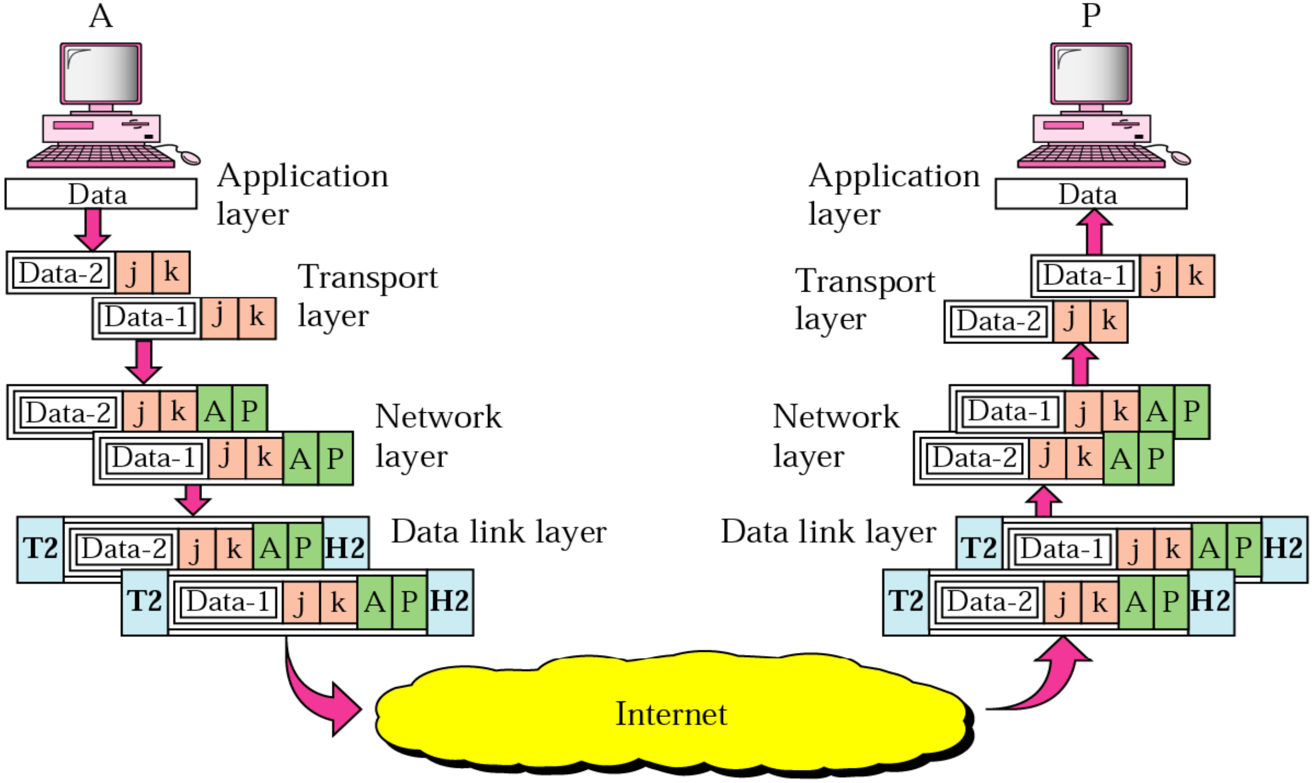


# RELIABLE PROCESS-TO-PROCESS DELIVERY OF A MESSAGE

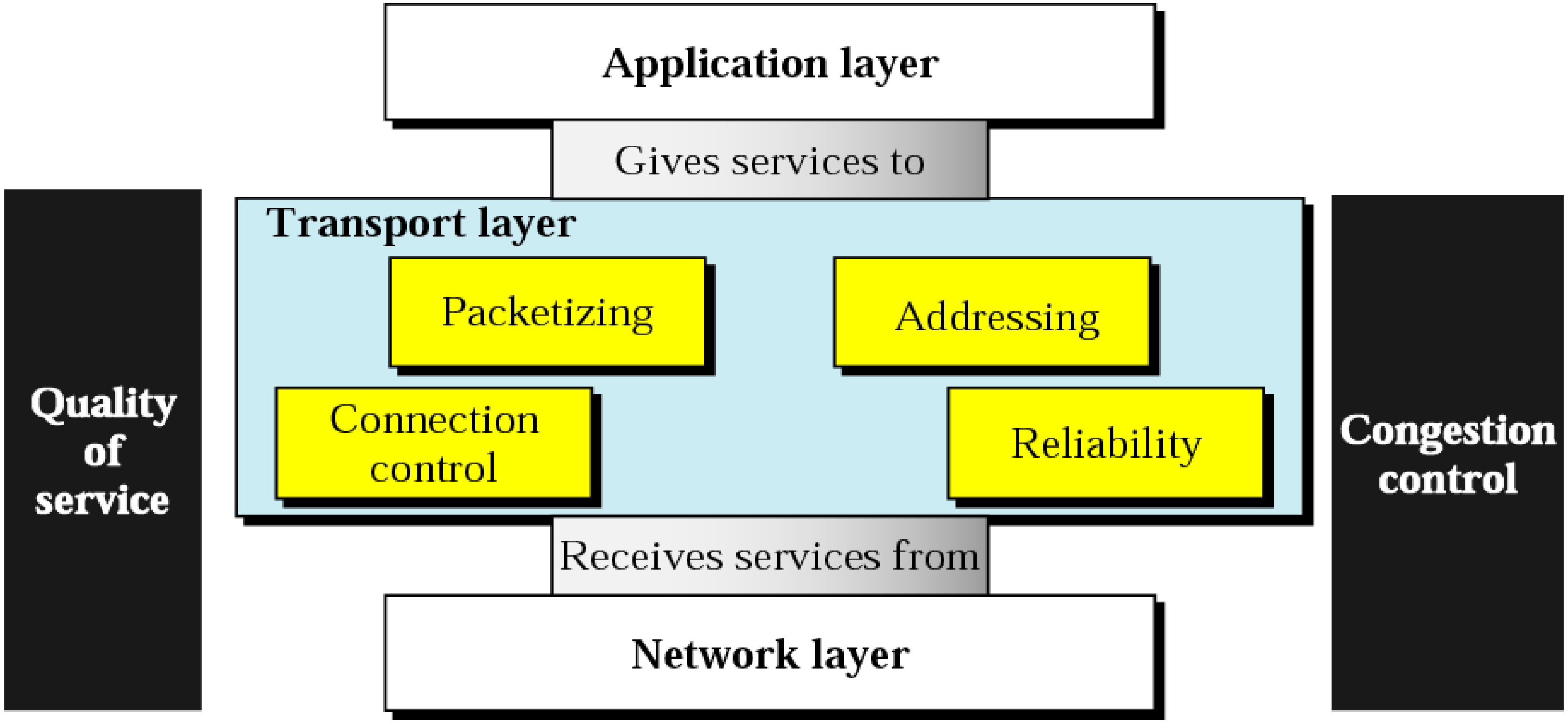


# *Example 3*

**Following Figure shows an example of transport layer communication. Data coming from the upper layers have port addresses j and k (j is the address of the sending process, and k is the address of the receiving process). Since the data size is larger than the network layer can handle, the data are split into two packets, each packet retaining the port addresses (j and k). Then in the network layer, network addresses (A and P) are added to each packet.**



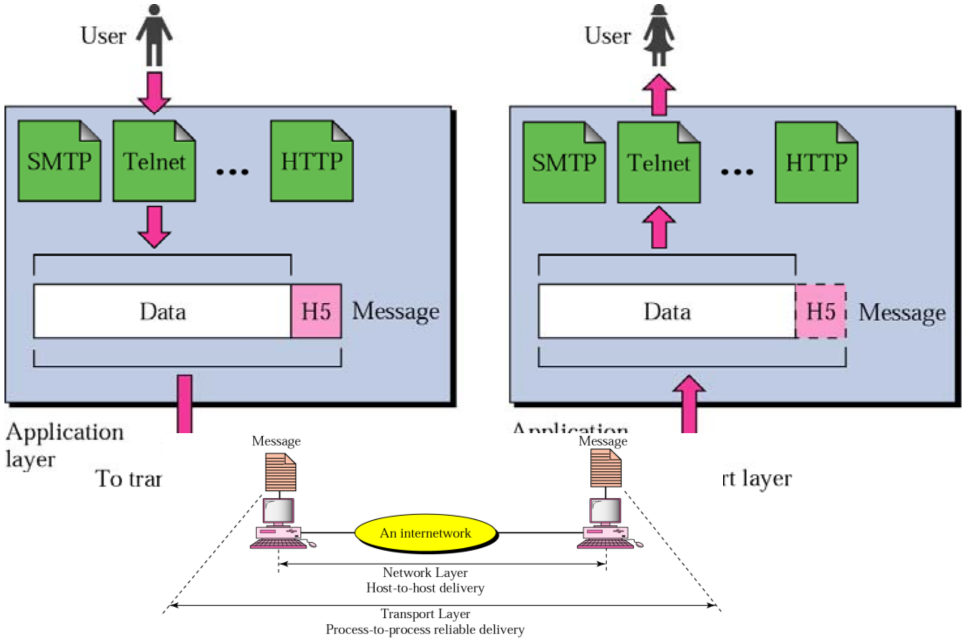
# Function of Transport layer



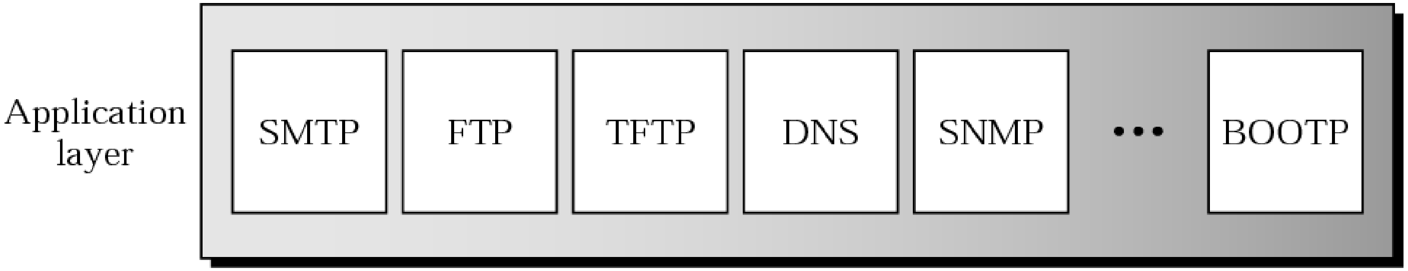
- Packetizing: encapsulating application message with transport header
- Connection control: connection-oriented vs. connectionless delivery
- Addressing: unique identifier for multiplexing
- Reliability: flow control & error control

# APPLICATION LAYER

- Responsible for providing services to the user
- Functions
  - Mail services
  - File transfer and access
  - Remote log-in
  - Accessing the World Wide Web

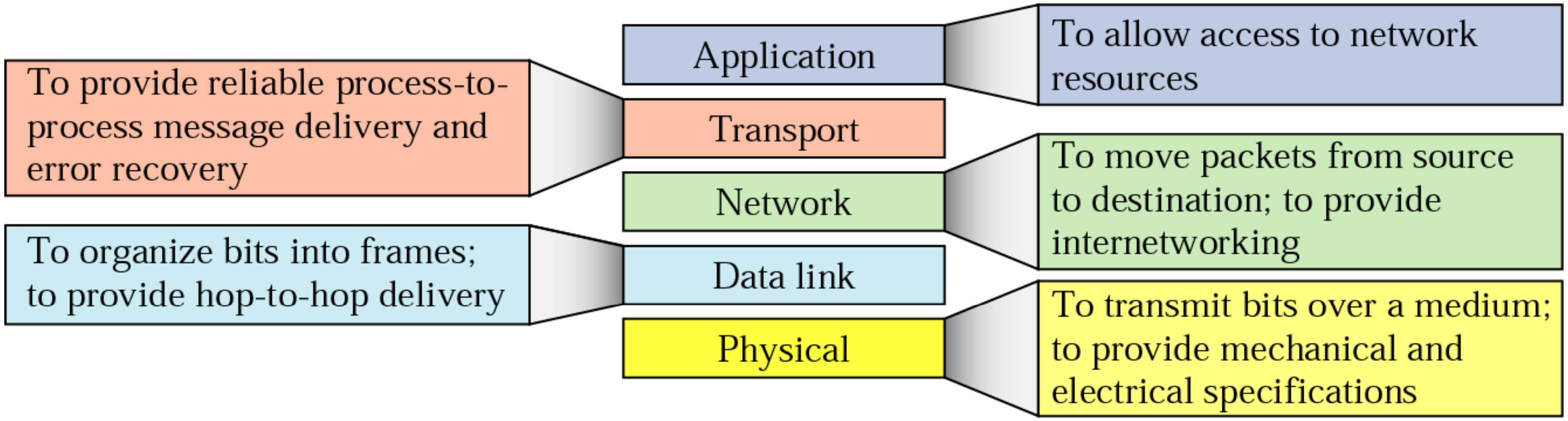


# Function of Application layer



1. 웹 브라우저가 웹 서버와의 동작
    1. 웹 브라우저가 웹 서버와의 사이에 프로세스와 프로토콜
2. 이메일이 인터넷에서 어떻게 다루어지는가?
    1. 이메일 주소의 구성
    2. 이메일 프로토콜의 동작 (클라이언트와 서버와의 사이에 프로세서와 프로토콜)
3. FTP 클라이언트와 서버와의 동작
    1. FTP 클라이언트와 서버와의 사이에 프로세서와 프로토콜
    2. 사용자 인터페이스와 응용계층프로토콜

# SUMMARY OF DUTIES



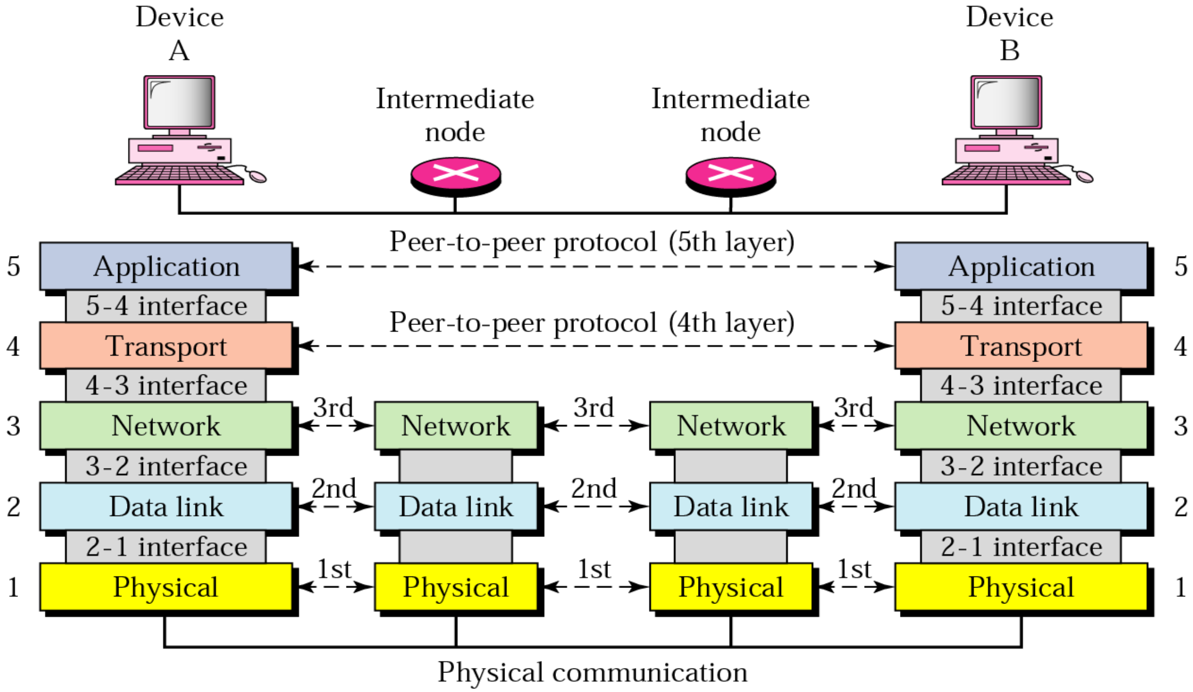
Application: To allow access to network resources

Transport: To provide reliable process-to-process message delivery and error recovery

Network: To move packets from source to destination; to provide internetworking

Data link: To organize bits into frames; to provide hop-to-hop delivery

Physical: To transmit bits over a medium; to provide mechanical and electrical specifications

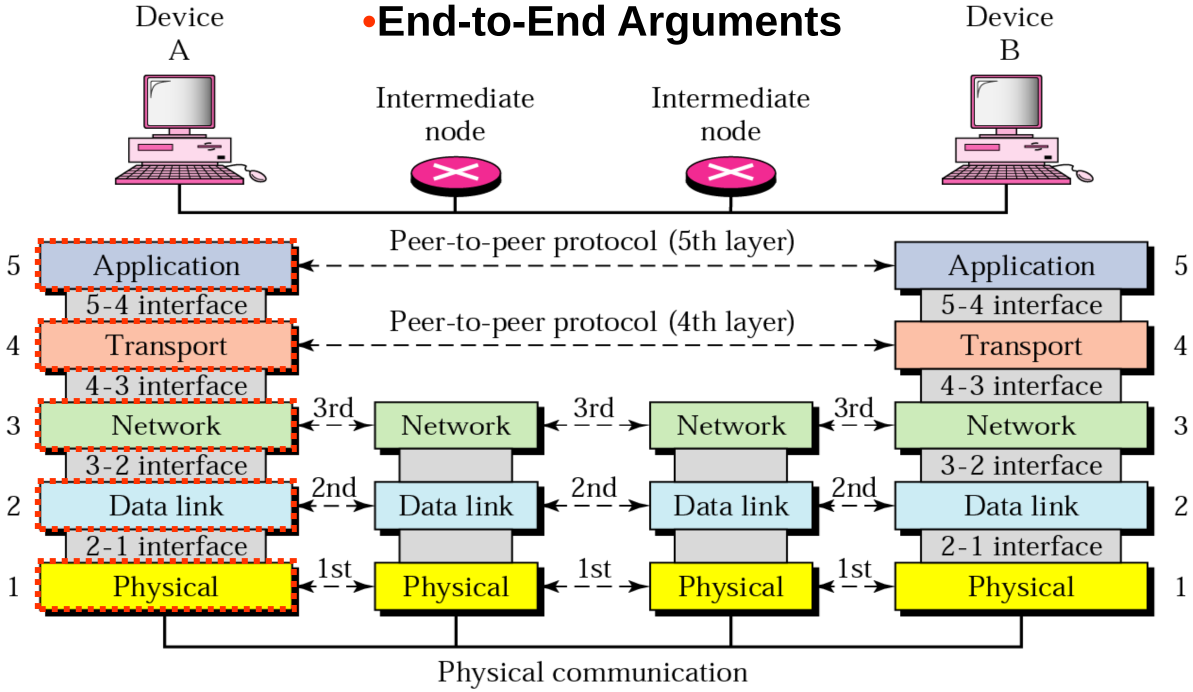


# 2-4 An example of Internet Protocol

# Internet Protocol

- A Case Study: Protocols –TCP/IP
- Layering
- End-to-End Arguments

•End-to-End Arguments



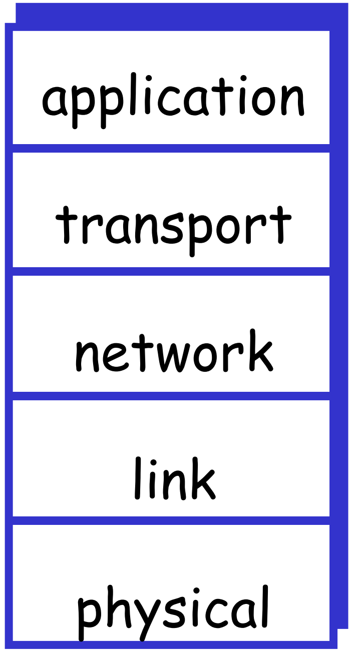
# Internet protocol stack- TCP/IP Suit

**Protocol Entity -> Peer-to-Peer Protocol**

- **"entities" implement layer functions at each node**
- **entities perform actions, exchange messages with peers**

**Each layer takes data from above**

- **adds header information to create new data unit**
- **passes new data unit to layer below**



application

transport

network

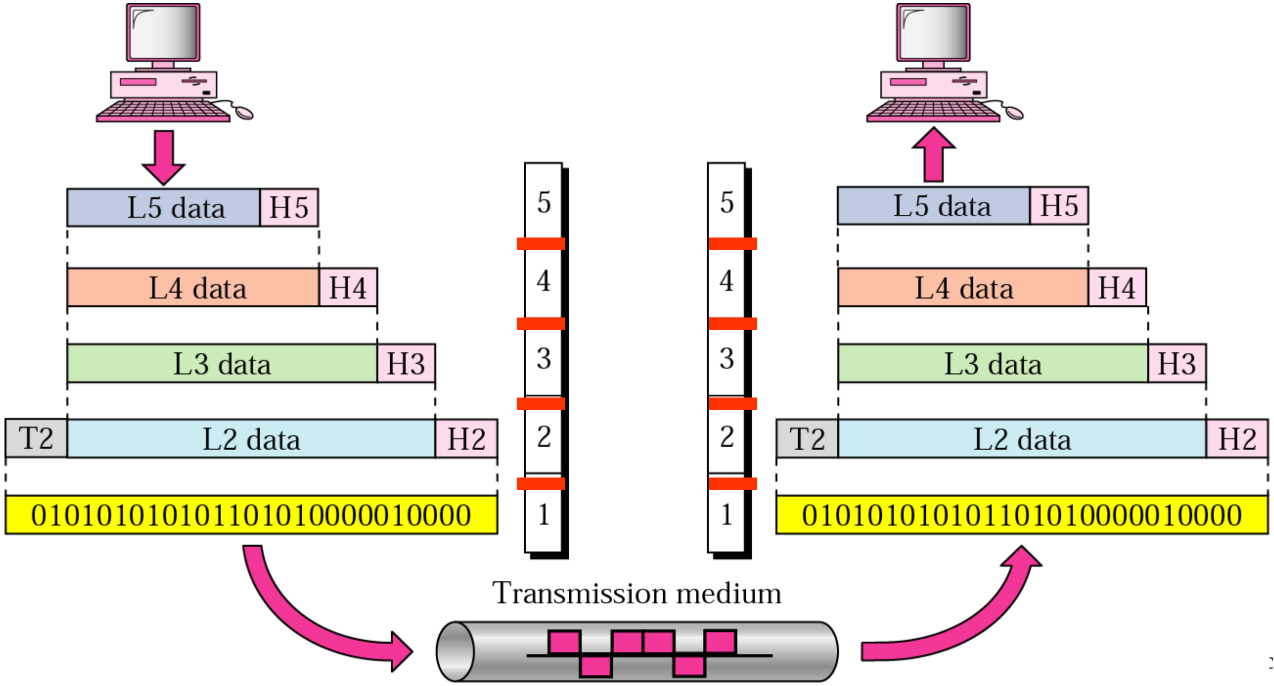
link

physical

- **application: supporting network applications (ftp, smtp, http)**
- **transport: host-host data transfer (TCP, UDP)**
- **network: routing of datagrams from source to destination (IP, routing protocols)**
- **link: data transfer between neighboring network elements (PPP, ethernet)**
- **physical: bits "on the wire"**

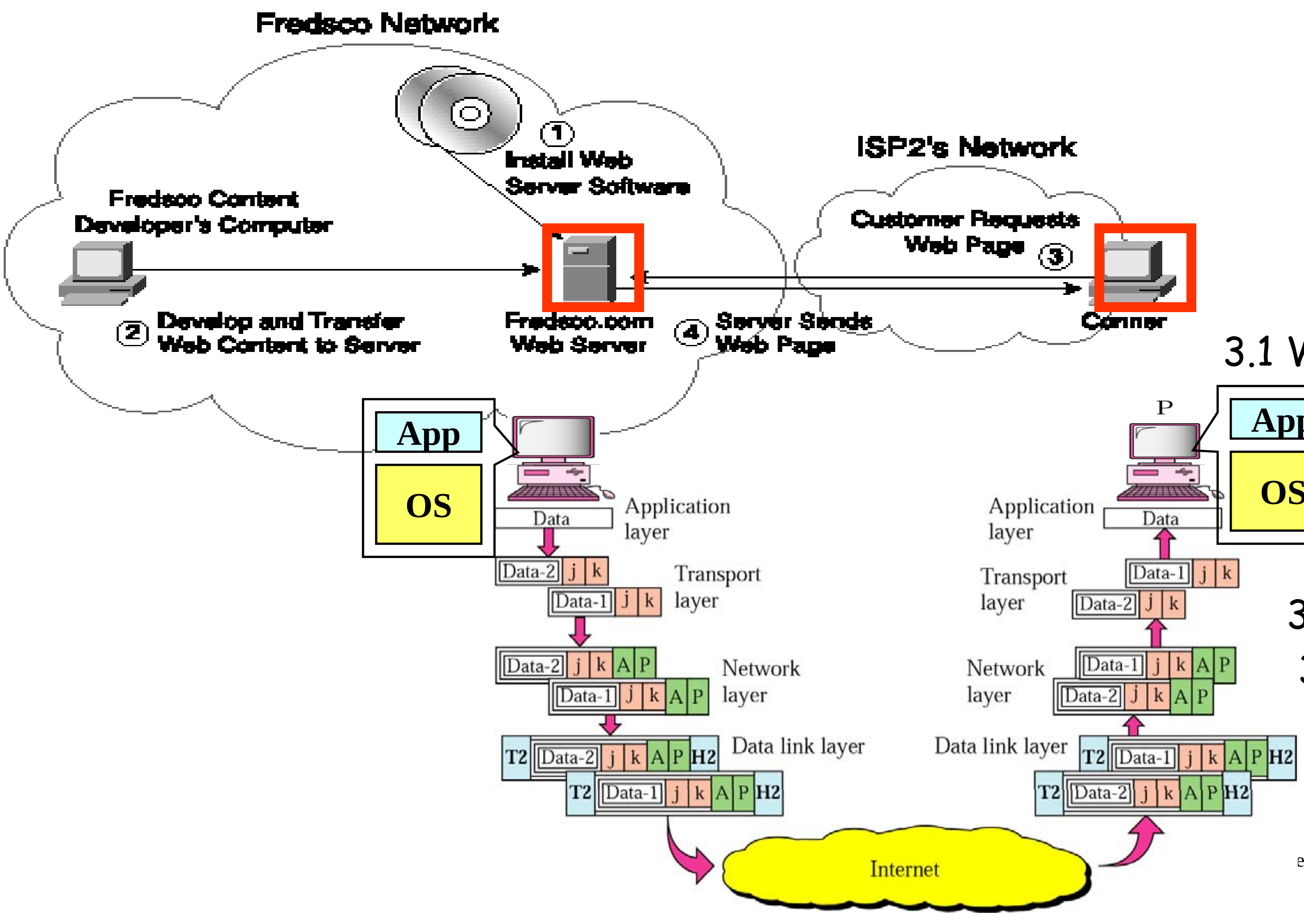
# AN EXCHANGE USING INTERNET MODEL

- The passing of the data and network information down through the layers of the sending device and backup through the layers of the receiving device is made possible by ***interface*** between each pair of adjacent layers
- Interface defines what information and services a layer must provide for the layer above it.



# HTTP Protocol over The Internet Protocol

**Figure 8-13** Building and Stocking the New Retail Store, AKA New Website



3.1 Web Browser & HTTP

3.2 TCP

3.3 IP

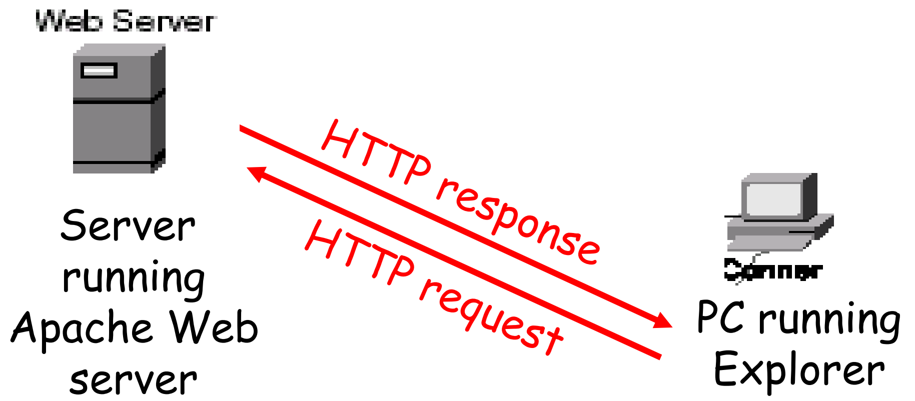
3.4 Ethernet

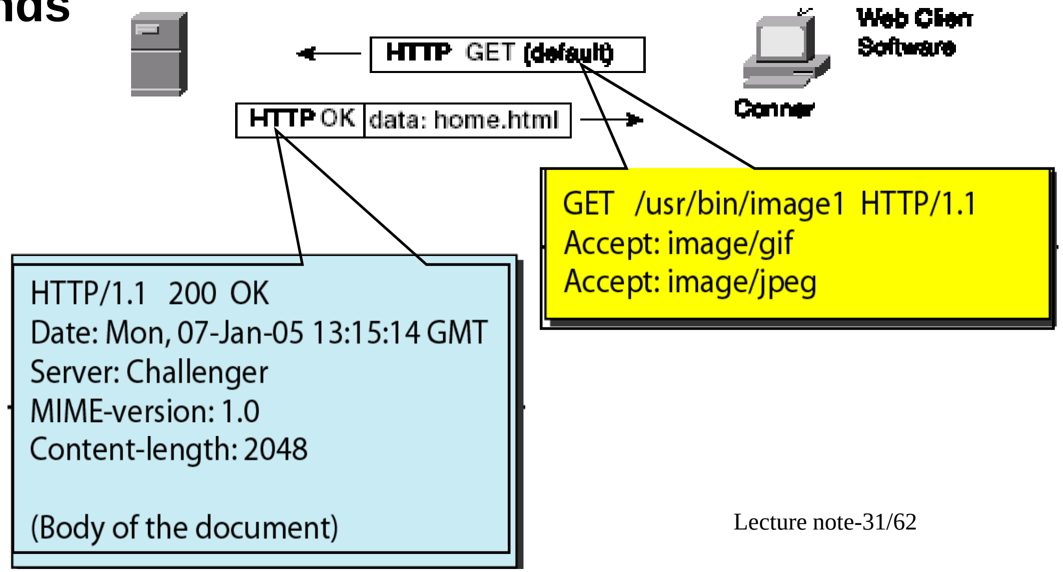
3.5 Physical

# HTTP Protocol Specification

## HTTP: hypertext transfer protocol

- Web's application layer protocol
- client/server model
  - *client:* browser that requests, receives, "displays" Web objects
  - *server:* Web server sends objects in response to requests
- HTTP 1.0: RFC 1945
- HTTP 1.1: RFC 2068
- HTML : RFC 2616





```
GET   /usr/bin/image1  HTTP/1.1
Accept: image/gif
Accept: image/jpeg
```

```
HTTP/1.1   200  OK
Date: Mon, 07-Jan-05 13:15:14 GMT
Server: Challenger
MIME-version: 1.0
Content-length: 2048

(Body of the document)
```

# TCP Protocol Specification

## TCP: Transmission Control protocol

- System Protocol in OS
- **Process-to- process delivery**
  - ***Reliable transmission***
  - ***Connection-oriented***
  - ***Port based Multiplexing***
- TCP: RFC793

**Figure 9-1** Application Program and Applicatior Transport Layer for Help

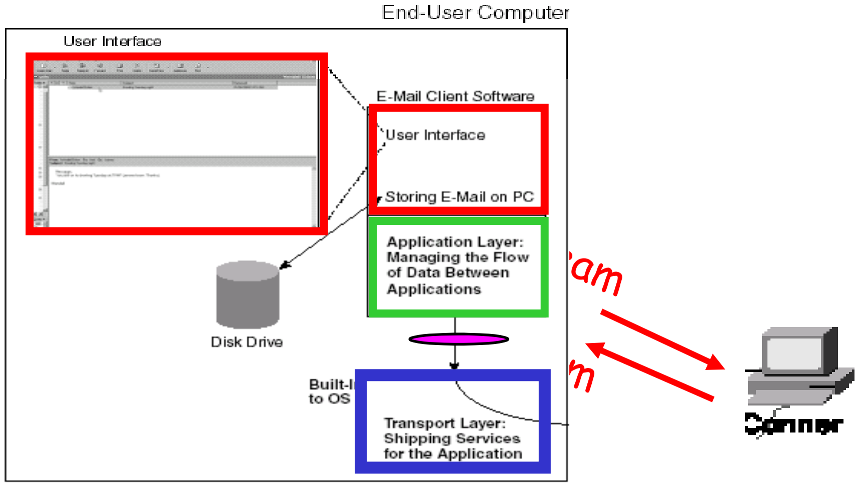
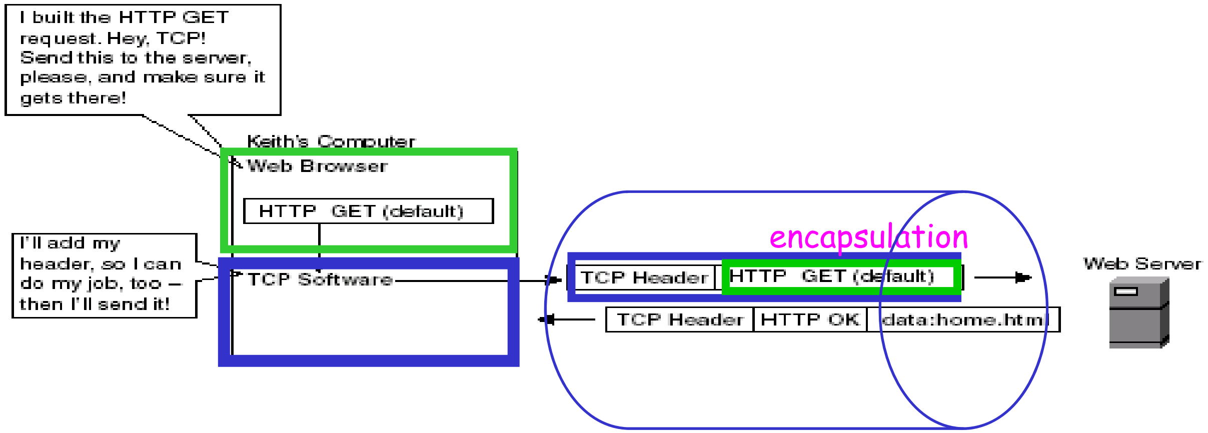
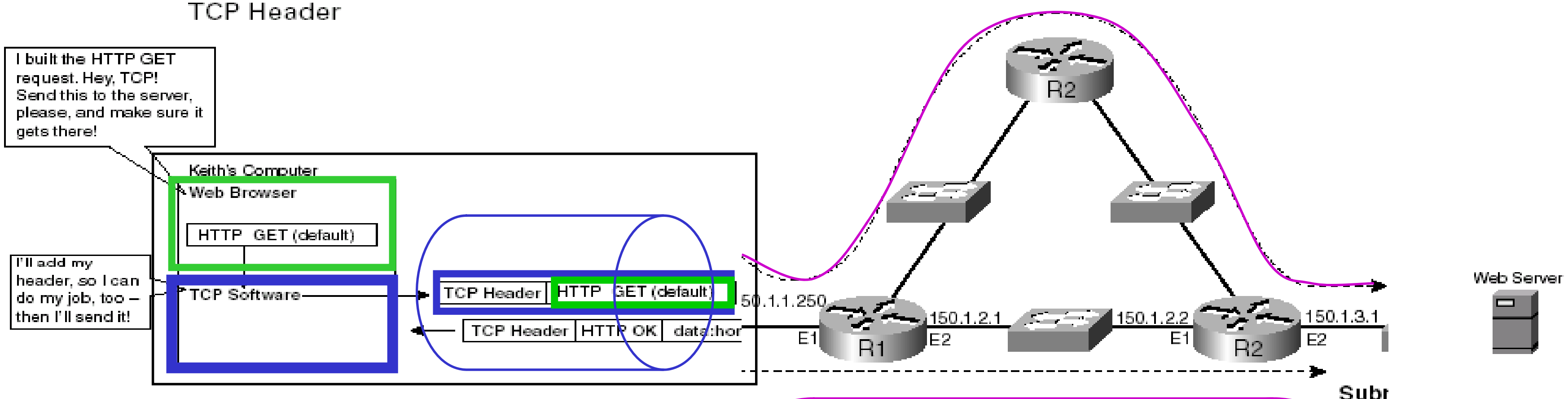

End-User Computer
User Interface
E-Mail Client Software
User Interface
Storing E-Mail on PC
Application Layer: Managing the Flow of Data Between Applications
Disk Drive
Built-I to OS
Transport Layer: Shipping Services for the Application
am
m

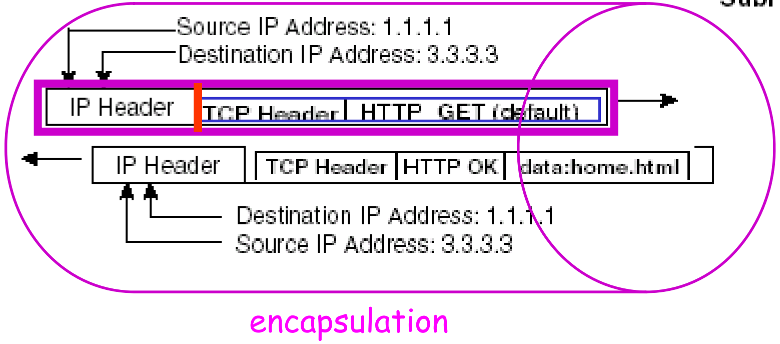
**Figure 9-2** Adding Shipping Information to Data by Encapsulating in a TCP Header


I built the HTTP GET request. Hey, TCP! Send this to the server, please, and make sure it gets there!
Keith's Computer
Web Browser
HTTP GET (default)
I'll add my header, so I can do my job, too – then I'll send it!
TCP Software
encapsulation
TCP Header
HTTP GET (default)
TCP Header
HTTP OK
data:home.htm
Web Server

# IP Protocol Specification

**Figure 9-2** Adding Shipping Information to Data by Encapsulating in a TCP Header





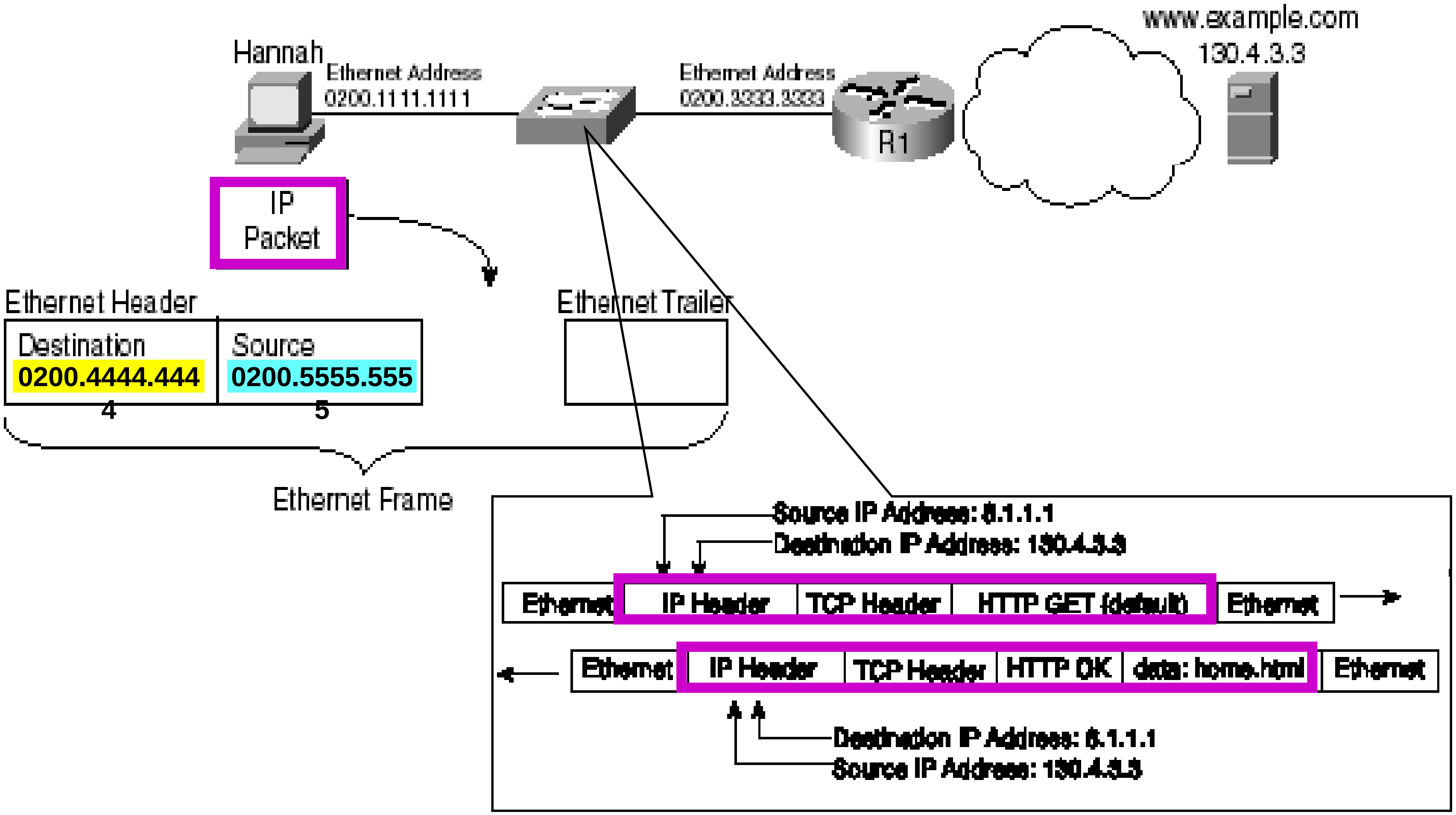
Source IP Address: 1.1.1.1
Destination IP Address: 3.3.3.3
IP Header
TCP Header
HTTP GET (default)
IP Header
TCP Header
HTTP OK
data:home.html
Destination IP Address: 1.1.1.1
Source IP Address: 3.3.3.3
encapsulation

## IP: Internet protocol

- **System Protocol in OS**
- **(node-to-node) delivery**
  - **Global Addressing: Source-to-destination**
  - **routing**
- IP: RFC791

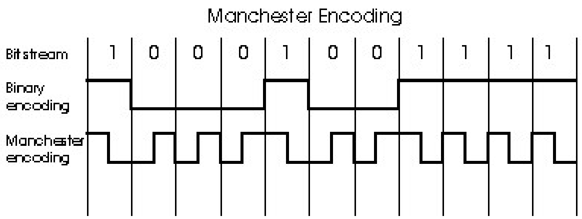
# Ethernet Specification

- 이더넷 프레임으로 캡슐링하기

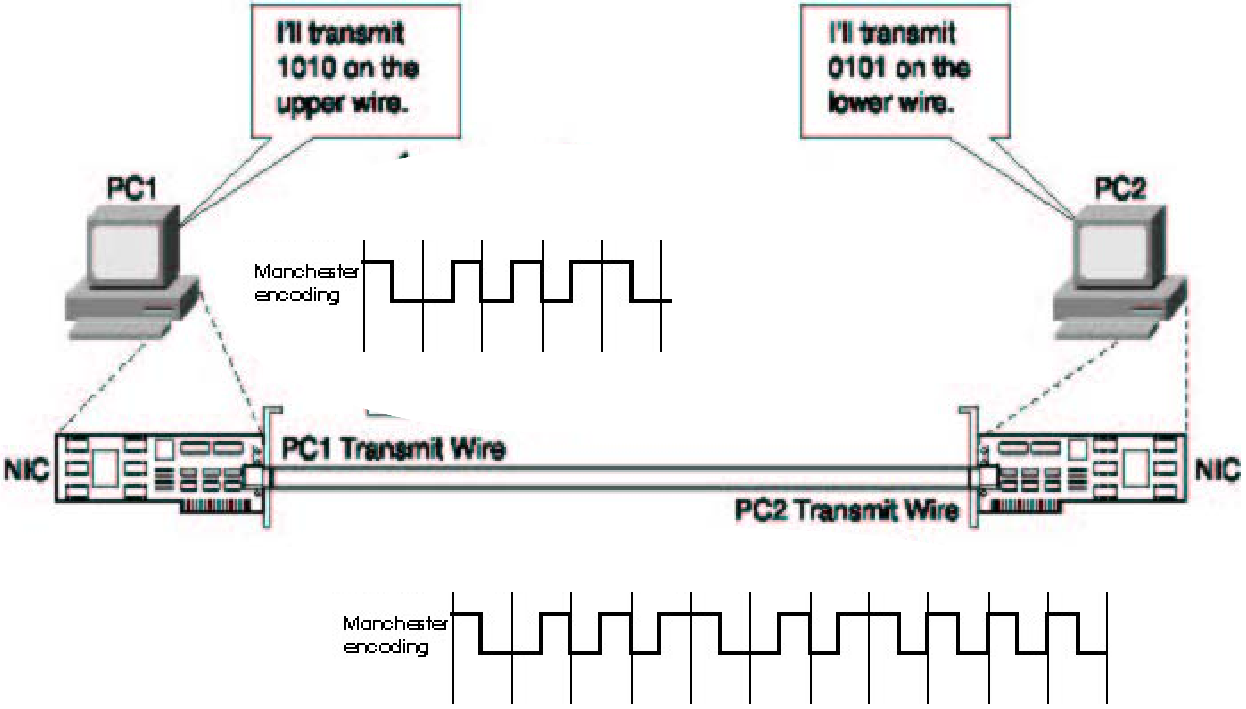


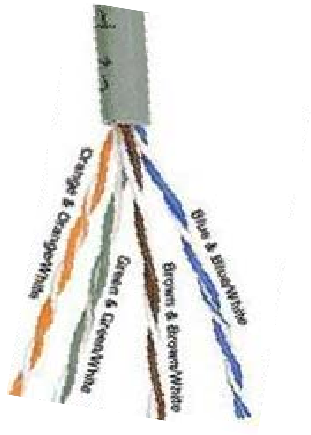
| Ethernet | IP Header | TCP Header | HTTP GET (default) | Ethernet |

# Physical Ethernet Specification



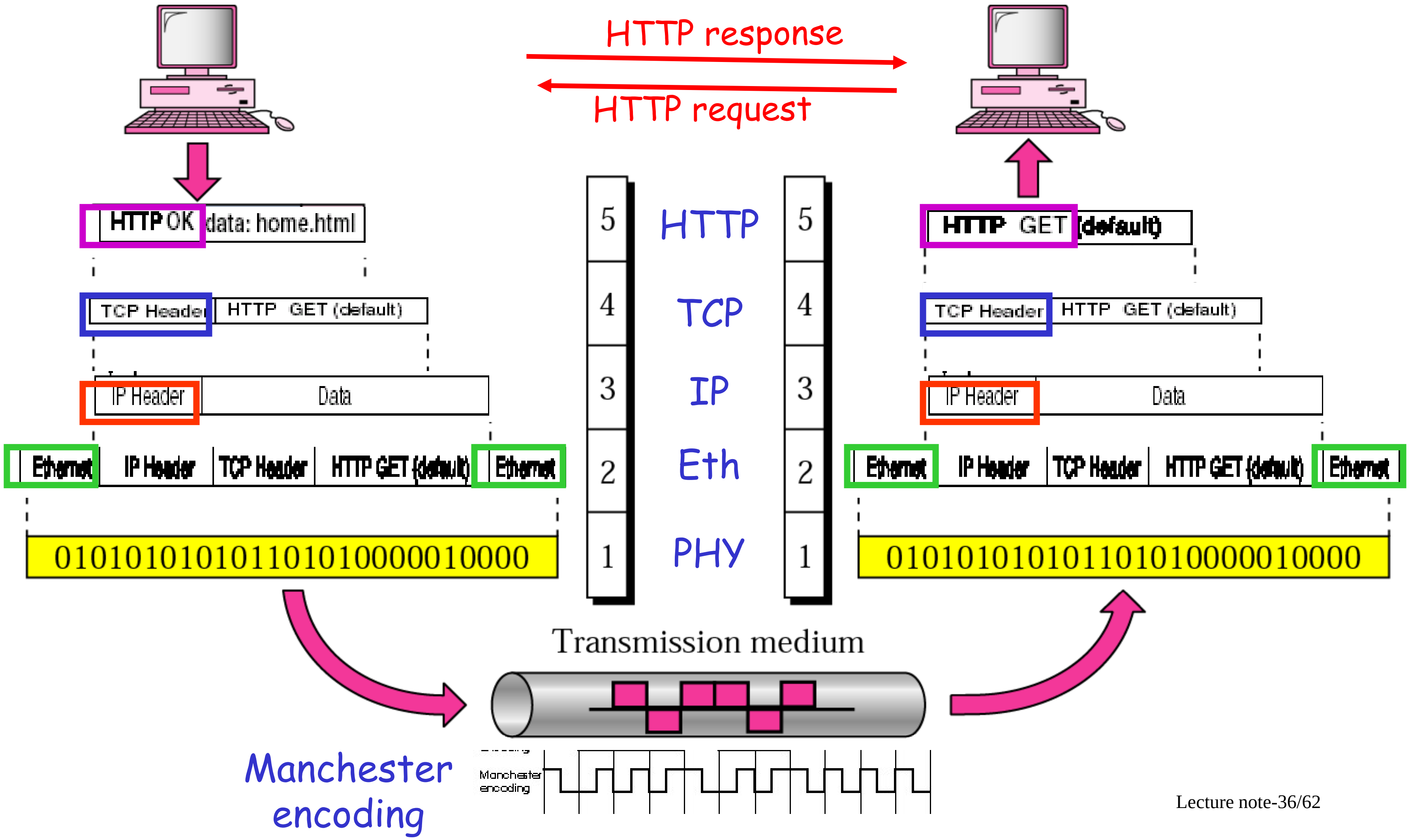
- 전선을 통해 비트 보내기
  - **NIC (**논리적 **1,0**을 전기적 **1,0**으로 변환**) –** 신호를 변환하는 기술
  - 전송 **(1,0**를 의미하는 전기 신호를 전선을 통해 전달**)**
    **-**전기신호를 표시하는 규칙 존재



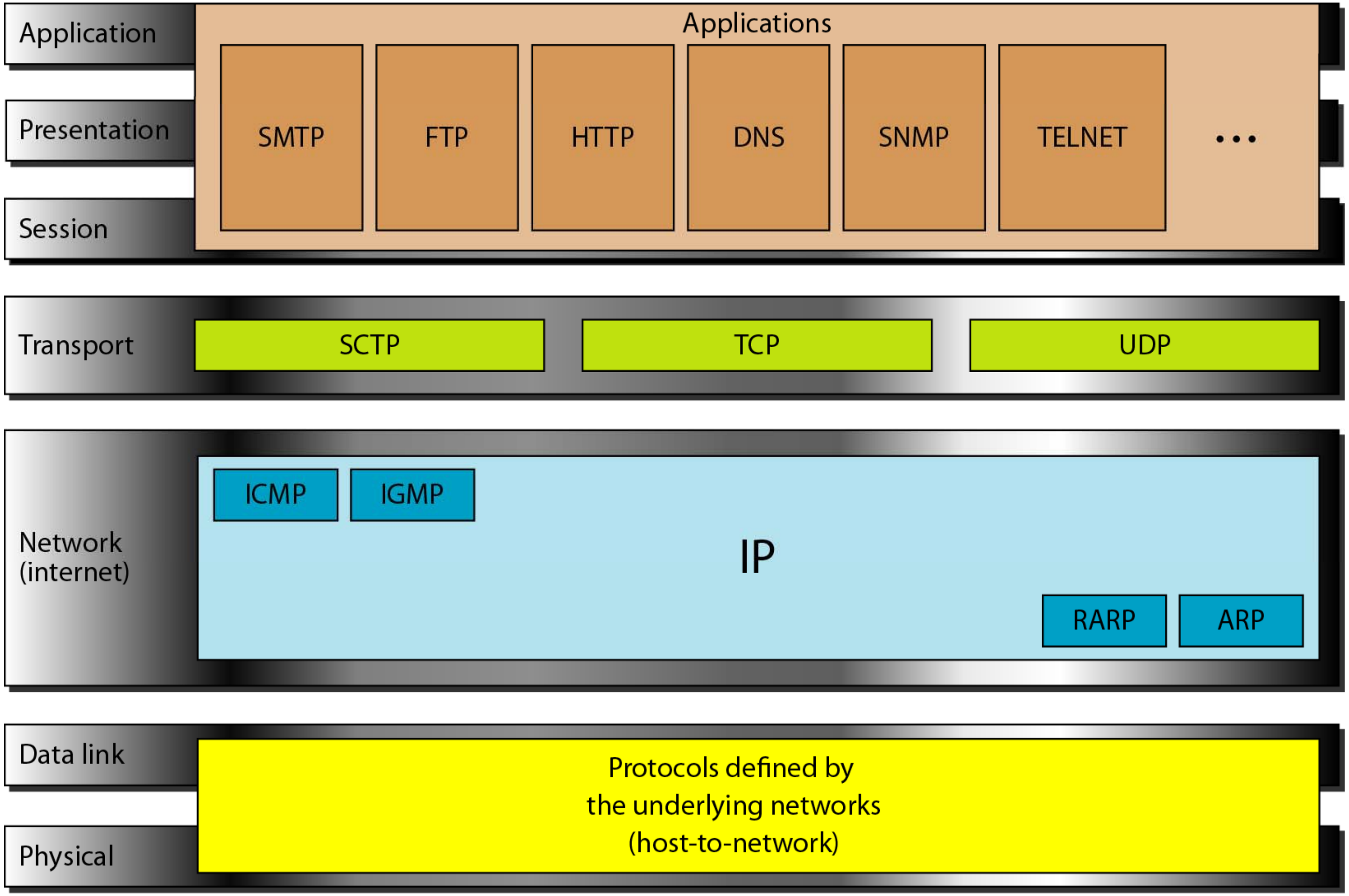


**Bps(bit per second):** 네트워킹의 연결 속도

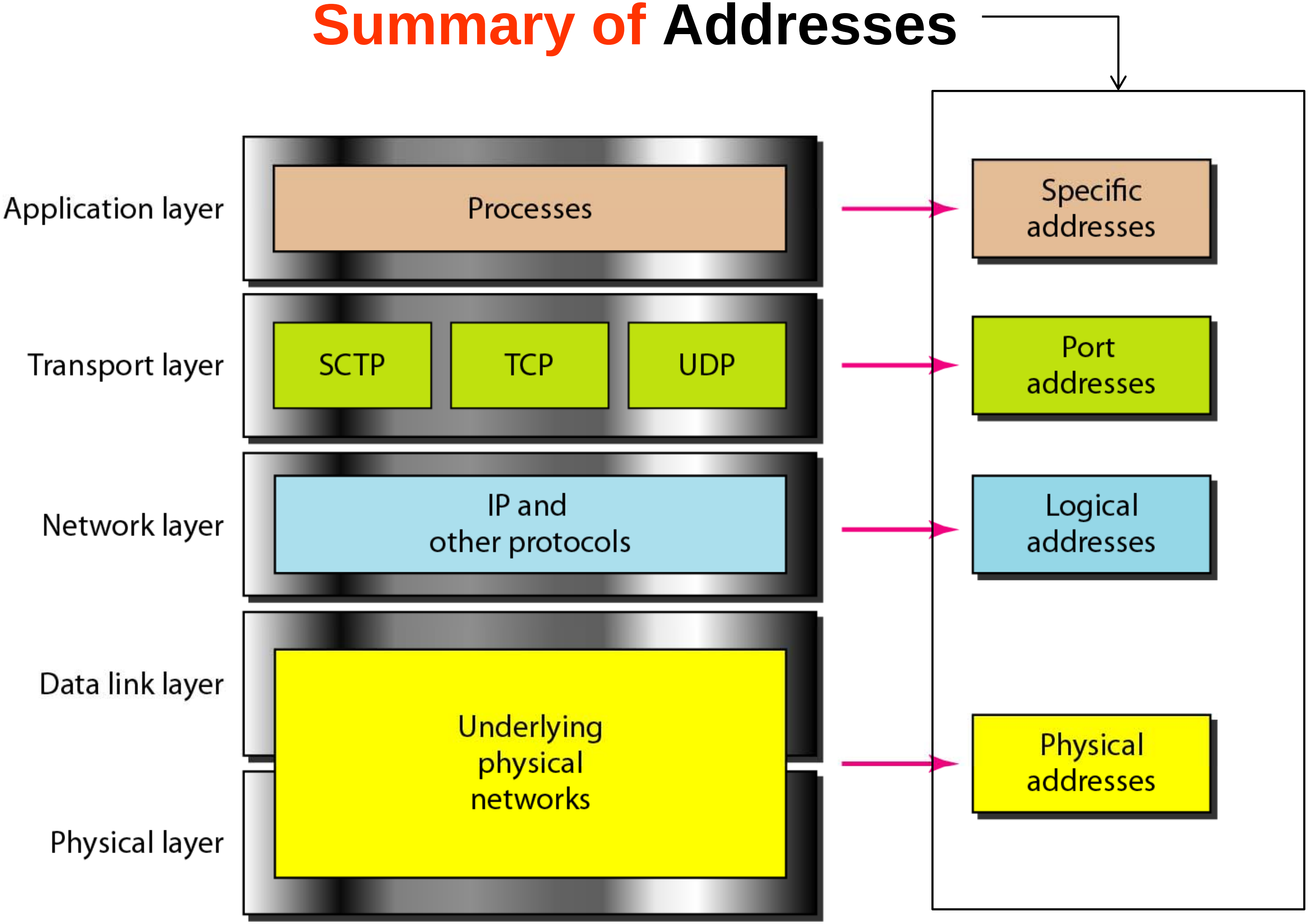
# Summary of Internet Protocol



# Summary of of Internet Protocol



# Summary of Addresses



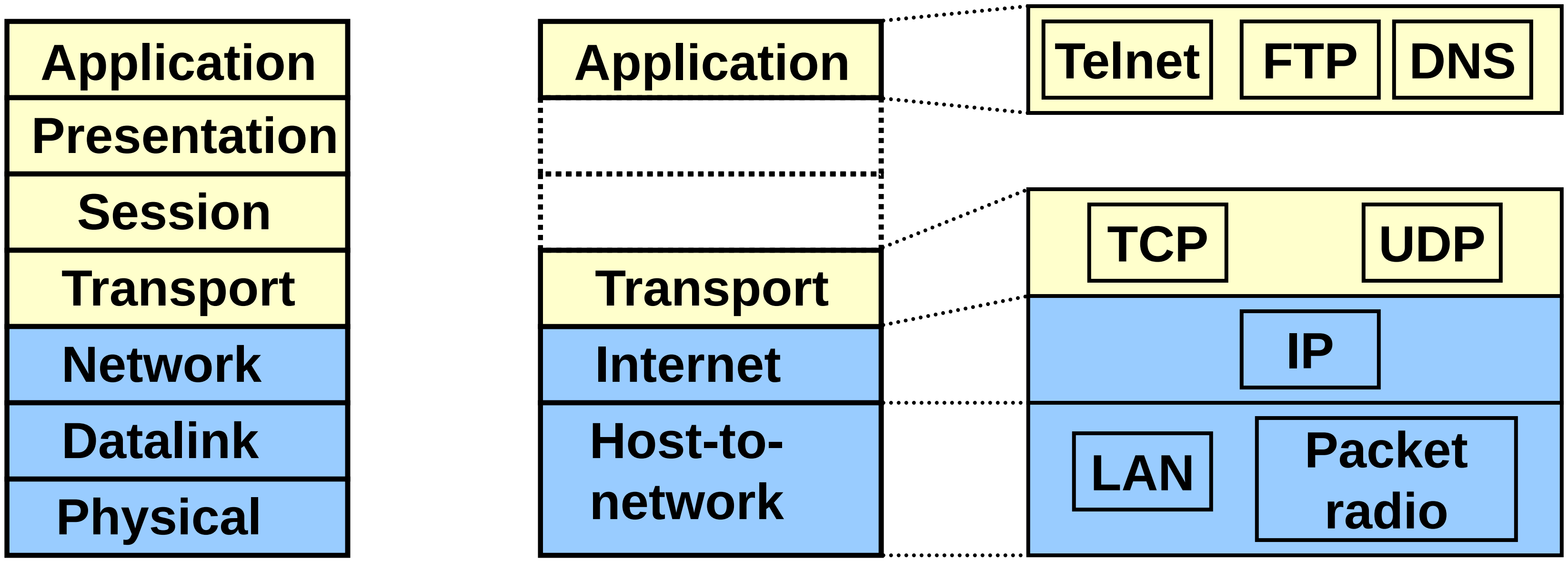
# 평가문제-Protocol

- 1. List the layers of the Internet Protocol Model?
  Answer: Physical, Link, Network, Transport, Application
- 2. What is a peer-to-peer process?
- Answer: implement functions at each layer, each function exchanges messages with peer
- 3. How does information get passed from one layer to the next in the Internet protocol model?
- Answer: Encapsulation
- 4. What are header and trailers?
- Answer: The term refers to bits that are added to end user data, for the purpose of allowing a protocol to have a place to keep information important to how the protocol does its function.
- 5. How do headers and trailers get added and removed?
- Answer:

# Review of HomeWork

# OSI vs. TCP/IP

- OSI: conceptually define: service, interface, protocol
- Internet: provide a successful implementation

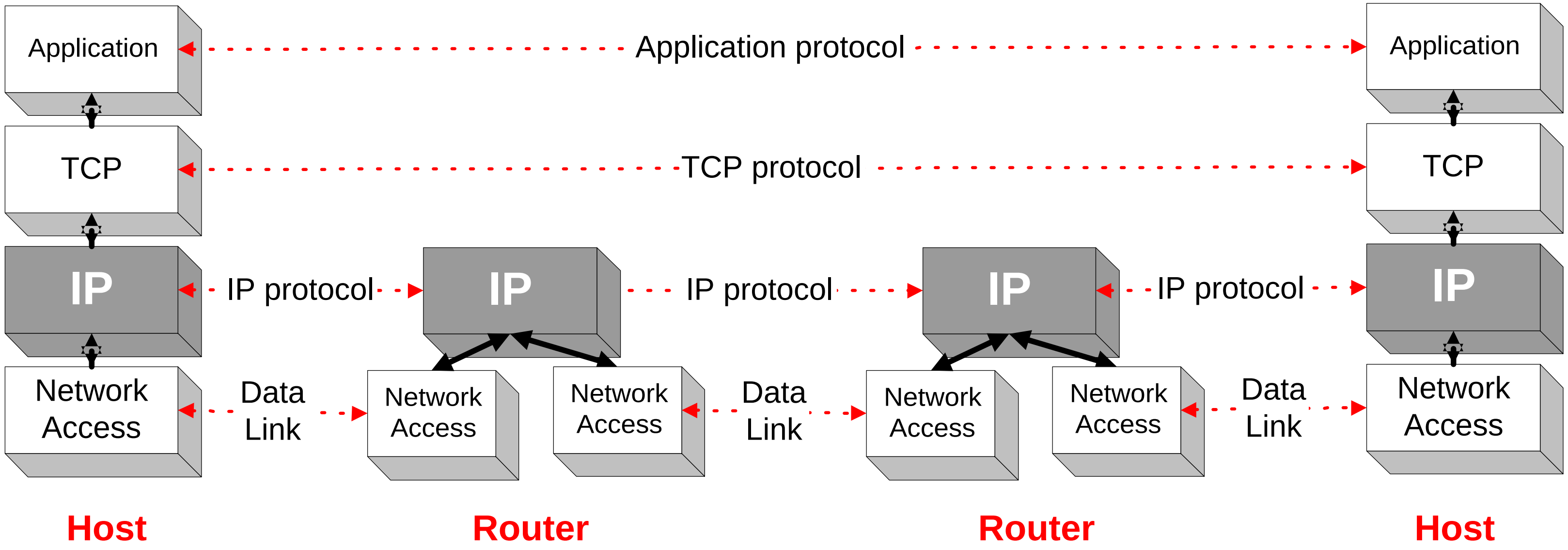


# Design Goals of Internet Protocol

## Layering

## End-to-End Arguments

# The design goals of The Internet Protocol



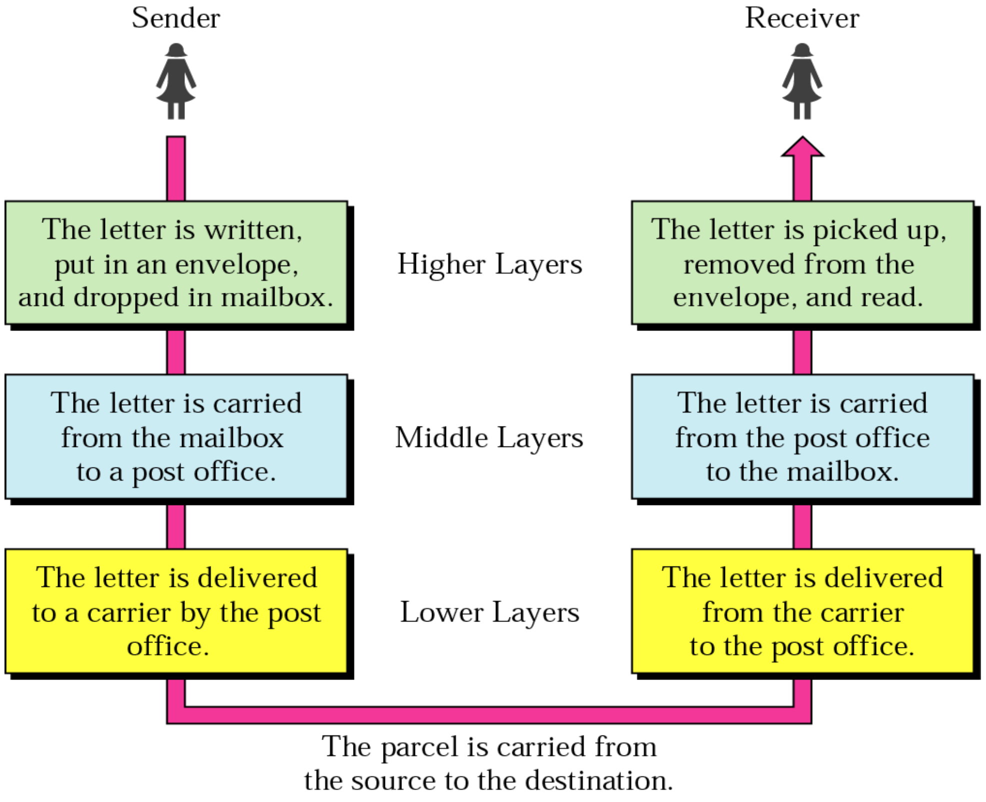
- Design goals
  - Layering
  - End-to-End Arguments

# Second level Goals

- **The Internet architecture must be cost effective.**
- **The Internet architecture must accommodate a variety of networks.**
- **The Internet must support multiple types of communications service.**
- **The Internet architecture must permit distributed management of its resources.**
- **Internet communication must continue despite loss of networks or gateways. -> robustness: failure protection, survivability**
- **The Internet architecture must permit host attachment with a low level of effort.**
- **The resources used in the Internet architecture must be accountable.**

# What is Layering?

- **A technique to organize a network system into a succession of logically distinct entities, such that the service provided by one entity is solely based on the service provided by the previous (lower level) entity**

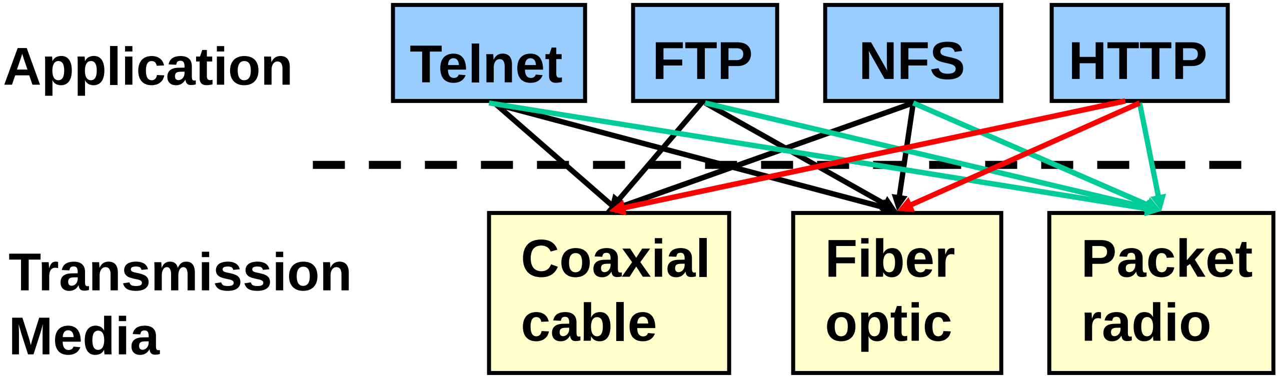

Sender
Receiver
The letter is written, put in an envelope, and dropped in mailbox.
Higher Layers
The letter is picked up, removed from the envelope, and read.
The letter is carried from the mailbox to a post office.
Middle Layers
The letter is carried from the post office to the mailbox.
The letter is delivered to a carrier by the post office.
Lower Layers
The letter is delivered from the carrier to the post office.
The parcel is carried from the source to the destination.

# Why layering?

**Dealing with complex systems:**

- **explicit structure allows identification, relationship of complex system's pieces**
  - **layered reference model for discussion**
- **modularization eases maintenance, updating of system**
  - **change of implementation of layer's service transparent to rest of system**
  - **e.g., change in gate procedure doesn't affect rest of system**
- **layering considered harmful?**

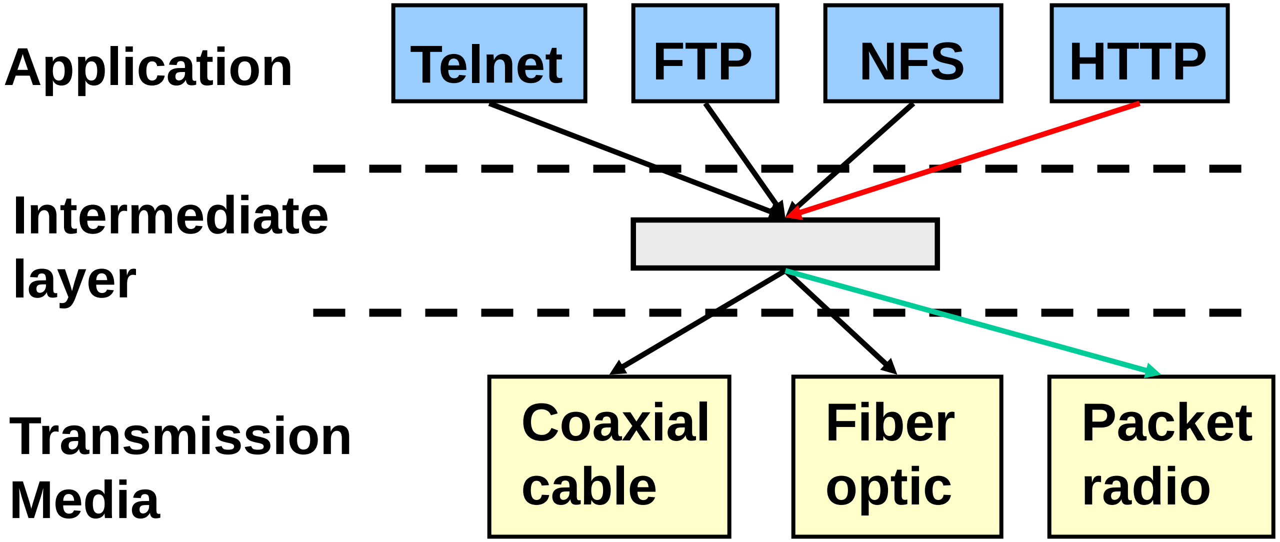
# Why Layering? –Example



- No layering: each new application has to be re-implemented for every network technology!

# Why Layering?-solution

- Solution: introduce an intermediate layer that provides a unique abstraction for various network technologies

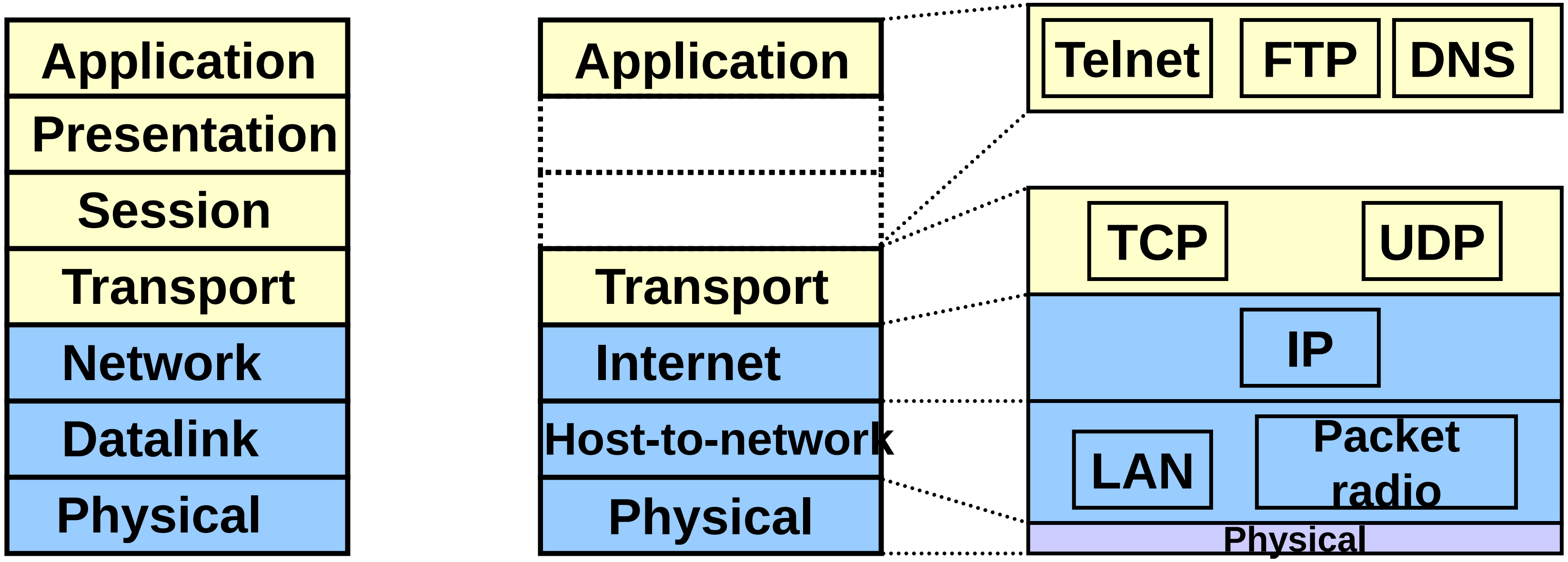


# Adv. & Dis. Adv. Of Layering

- Advantages
  - Modularity – protocols easier to manage and maintain
  - Abstract functionality –lower layers can be changed without affecting the upper layers
  - Reuse – upper layers can reuse the functionality provided by lower layers
- Disadvantages
  - Information hiding – inefficient implementations

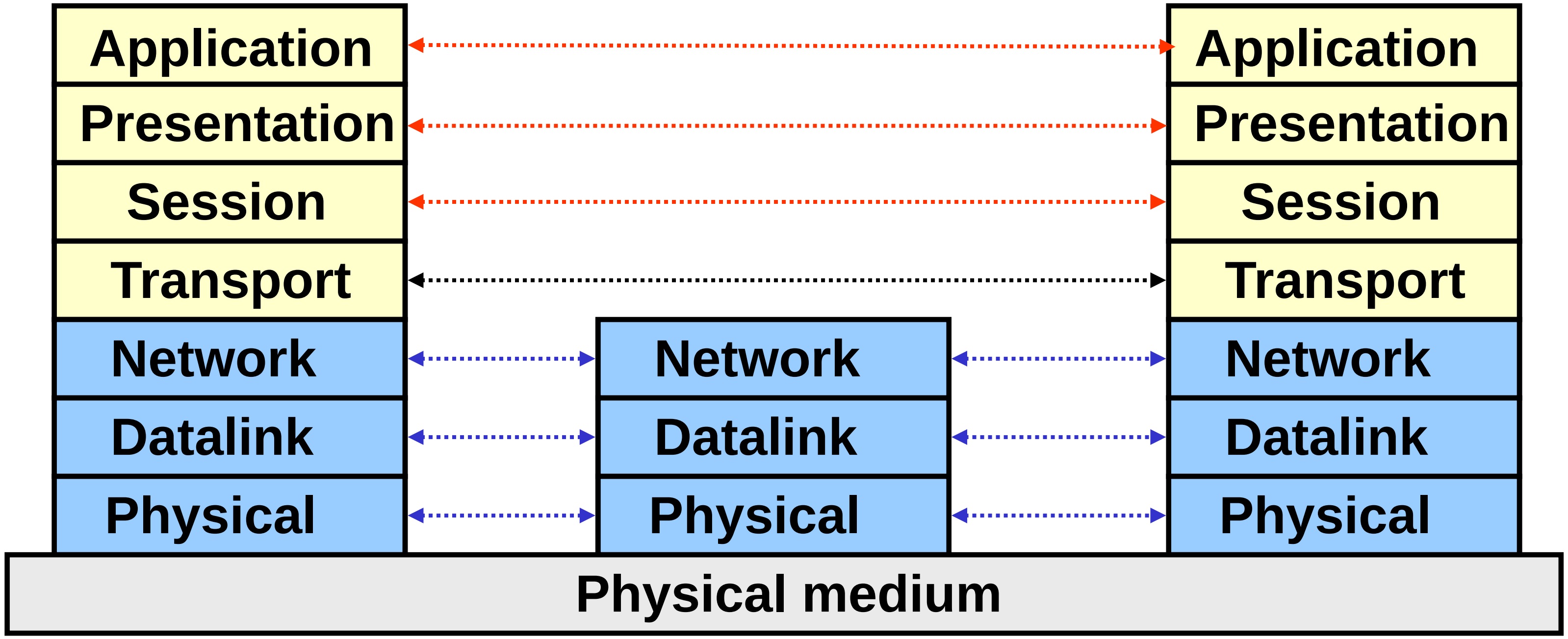
# Internet Layering - TCP/IP

- OSI: conceptually define: service, interface, protocol
- Internet: provide a successful implementation

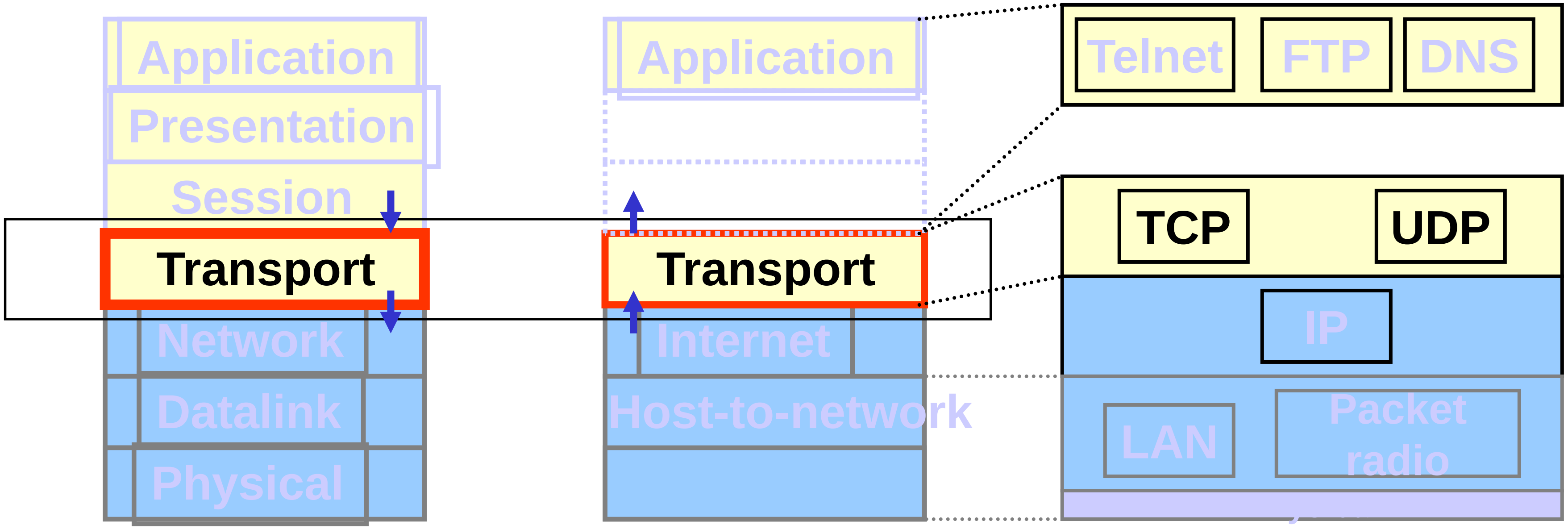


# Internet Layering - ISO OSI 7-layer Model

- Seven layers
  - Lower three layers are peer-to-peer
  - Next four layers are end-to-end



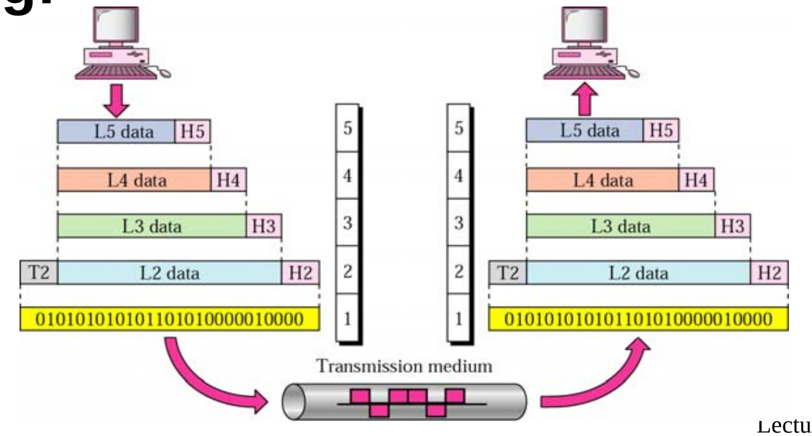
# Layering Concepts



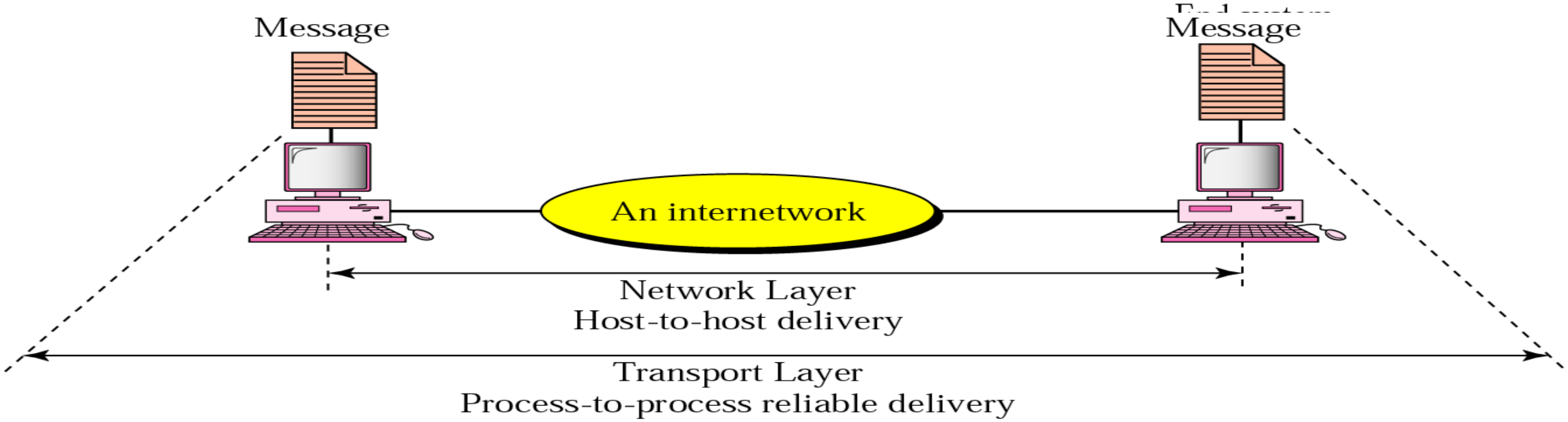
- Service – says **what** a layer does
- Interface – says **how** to **access** the service
- Protocol – says **how** is the service **implemented**
  - a set of rules and formats that govern the communication between two peers

# Summary: Layering

- Key technique to implement communication protocols; provides
  - Modularity
  - Abstraction
  - Reuse
- Key design decision:
  what functionality to put in each layer?
- The result of Layering:
  Encapsulation

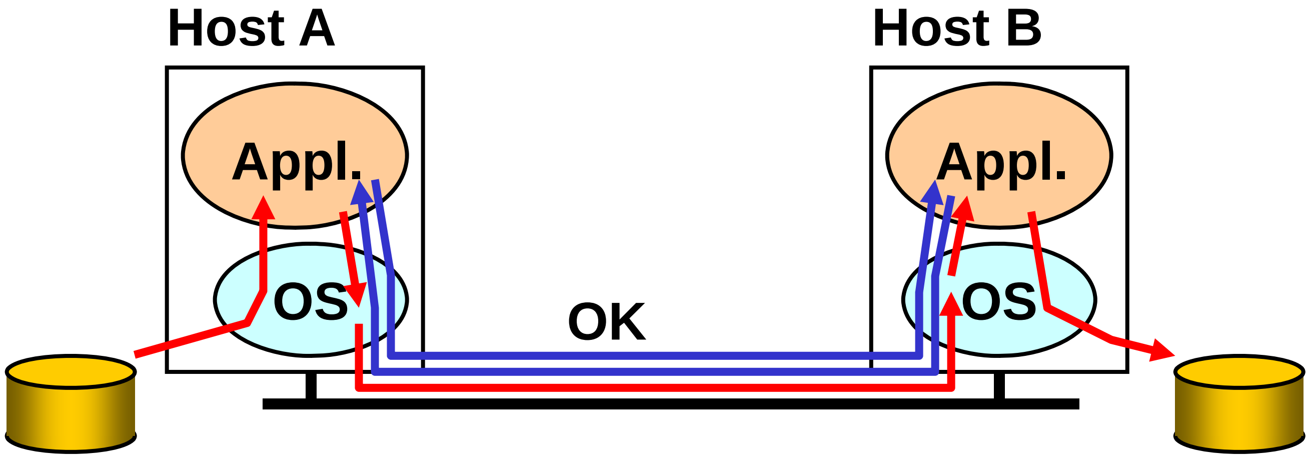


# End-to-End Argument



- At network layer provides one simple service: best effort datagram (packet) delivery
- Only one higher level service implemented at transport layer: reliable data delivery (TCP)
  - performance enhancement; used by a large variety of applications (Telnet, FTP, HTTP)
  - does not impact other applications (can use UDP)
- Everything else implemented at application level

# Example: End-to-End Protocol



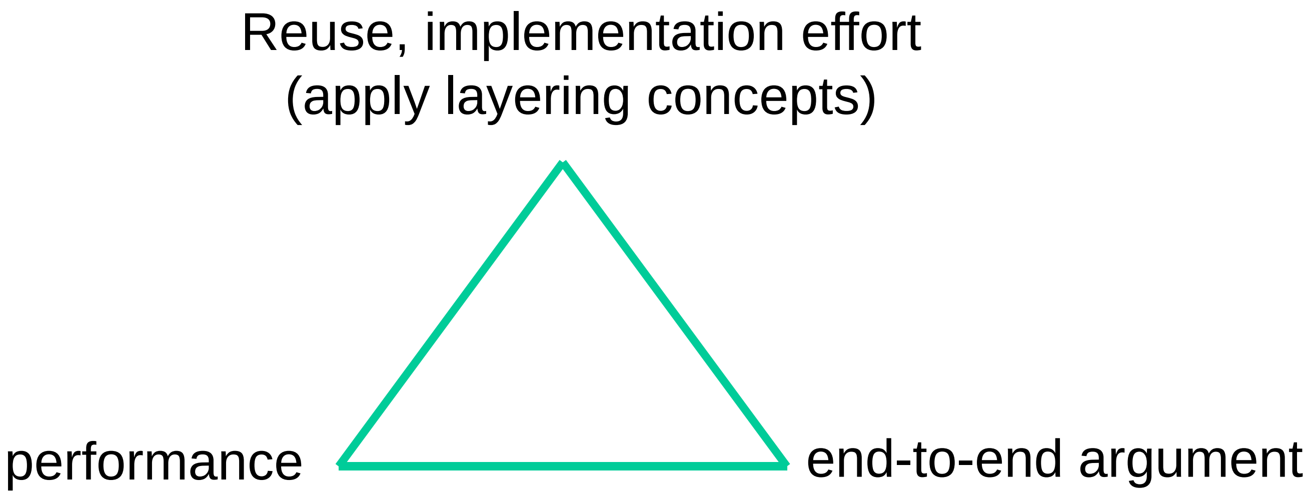
- **Solution 1(no E-t-E):** make each step reliable, and then concatenate them
- **Solution 2 (E-t-E):** end-to-end check and retry

# Summary: End-to-End Arguments

- If the application can do it, don't do it at a lower layer -- anyway the application knows the best what it needs
  - add functionality in lower layers iff it is (1) used and improves performances of a large number of applications, and (2) does not hurt other applications
- Key design decision: peer-to-peer protocol
  what is the role of End-to-End argument in each layer?
- Success story: Internet

# Summary of Design Goals

- **Challenge of building a good (network) system: find the right balance between:**



Reuse, implementation effort
(apply layering concepts)

performance

end-to-end argument

- No universal answer: the answer depends on the goals and assumptions!

# Discussion- Reliable File Transfer Protocol

- **Solution 1 not complete**
  - **What happens if the sender or/and receiver misbehave?**
- **The receiver has to do the check anyway!**
- **Thus, full functionality can be entirely implemented at application layer; no need for reliability from lower layers**
- **Is there any need to implement reliability at lower layers?**

# Key Advantages

- The service can be implemented by a large variety of network technologies
- Does not require routers to maintain any fined grained state about traffic. Thus, network architecture is
  - Robust
  - Scalable

# Trade-offs

- **Application has more information about the data and the semantic of the service it requires (e.g., can check only at the end of each data unit)**
- **A lower layer has more information about constraints in data transmission (e.g., packet size, error rate)**

- **Note: these trade-offs are a direct result of layering!**

# Rule of Thumb

- Implementing a functionality at a lower level should have minimum performance impact on the application that do not use the functionality