Introduction to Internet #3

Introduction to Internet and Web

부산대학교 정보·의생명 공학대학
정보컴퓨터공학부

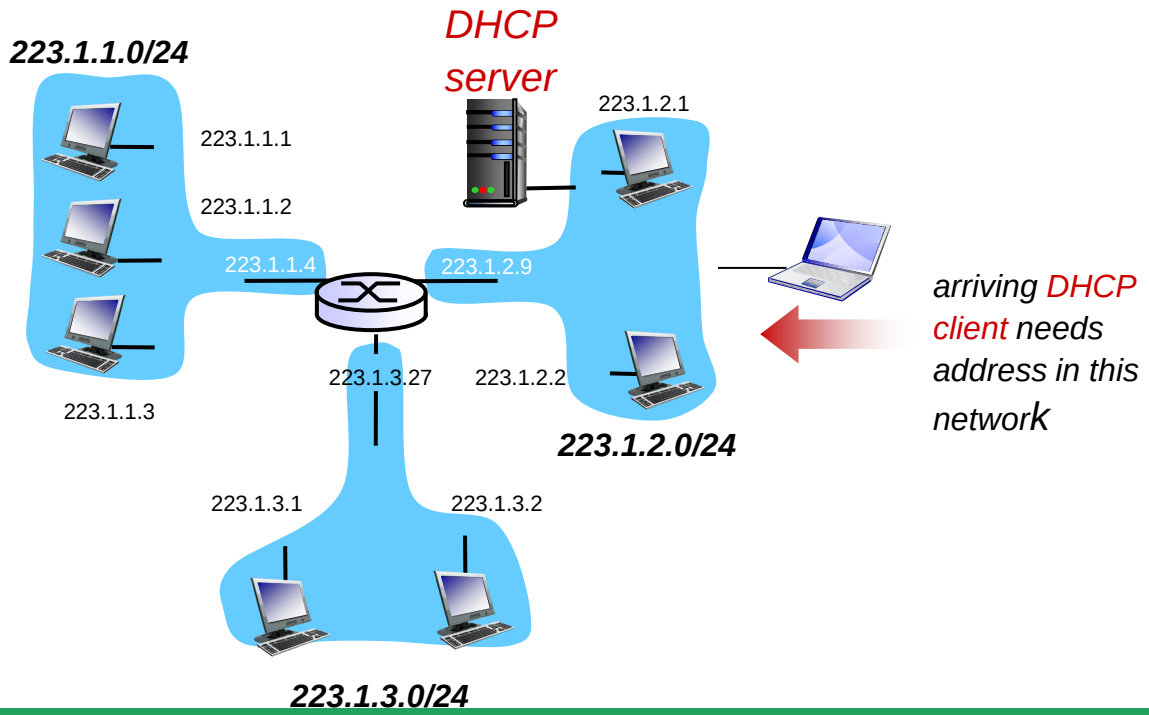
Table of Contents

DYNAMIC HOST CONFIG. PROTOCOL

DHCP (Dynamic Host Configuration Protocol)

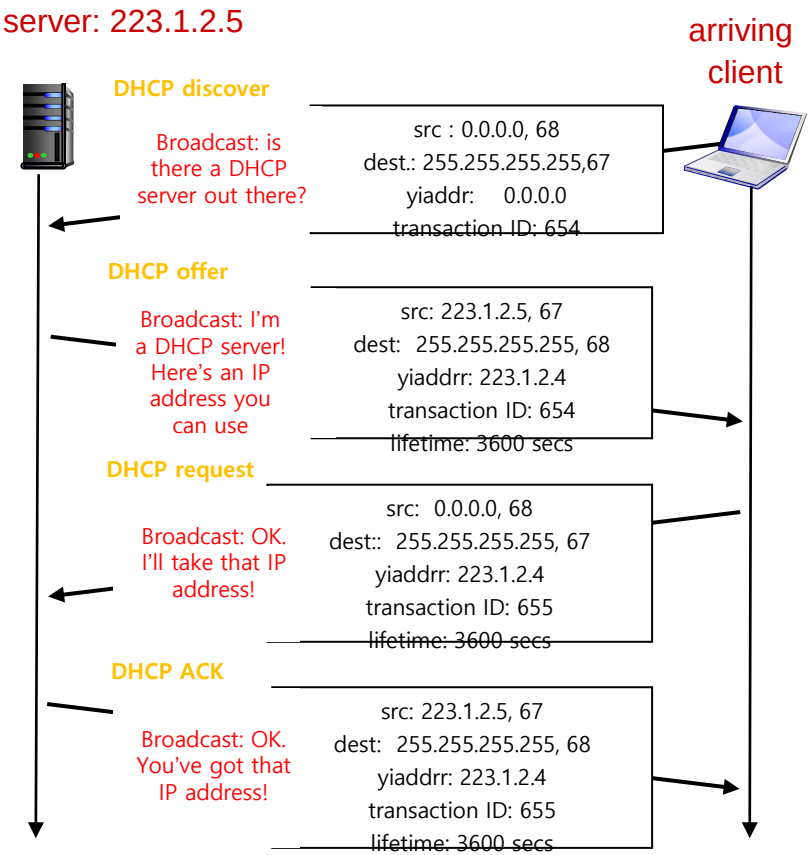
❖ **Host dynamically obtains its IP address from network server when it joins network**

- can renew its lease on address in use
- allows reuse of addresses (only hold address while connected/"on")
- support for mobile users who want to join network (more shortly)



DHCP Overview

1) host broadcasts “DHCP discover”
2) DHCP server responds with “DHCP offer”
3) host requests IP address: “DHCP request”
4) DHCP server sends address: “DHCP ack”

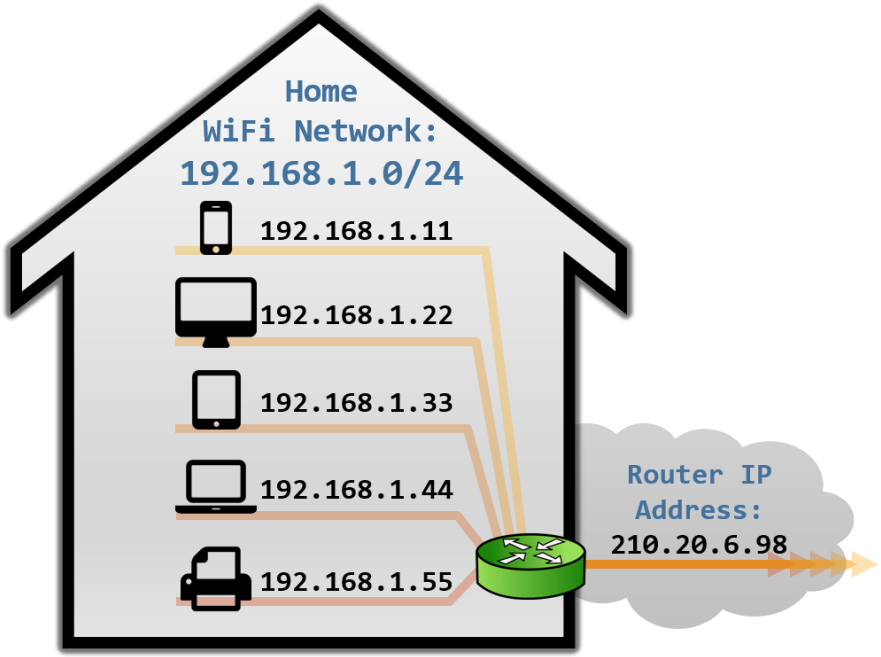


NETWORK ADDRESS TRANSLATION

NAT (Network Address Translation)

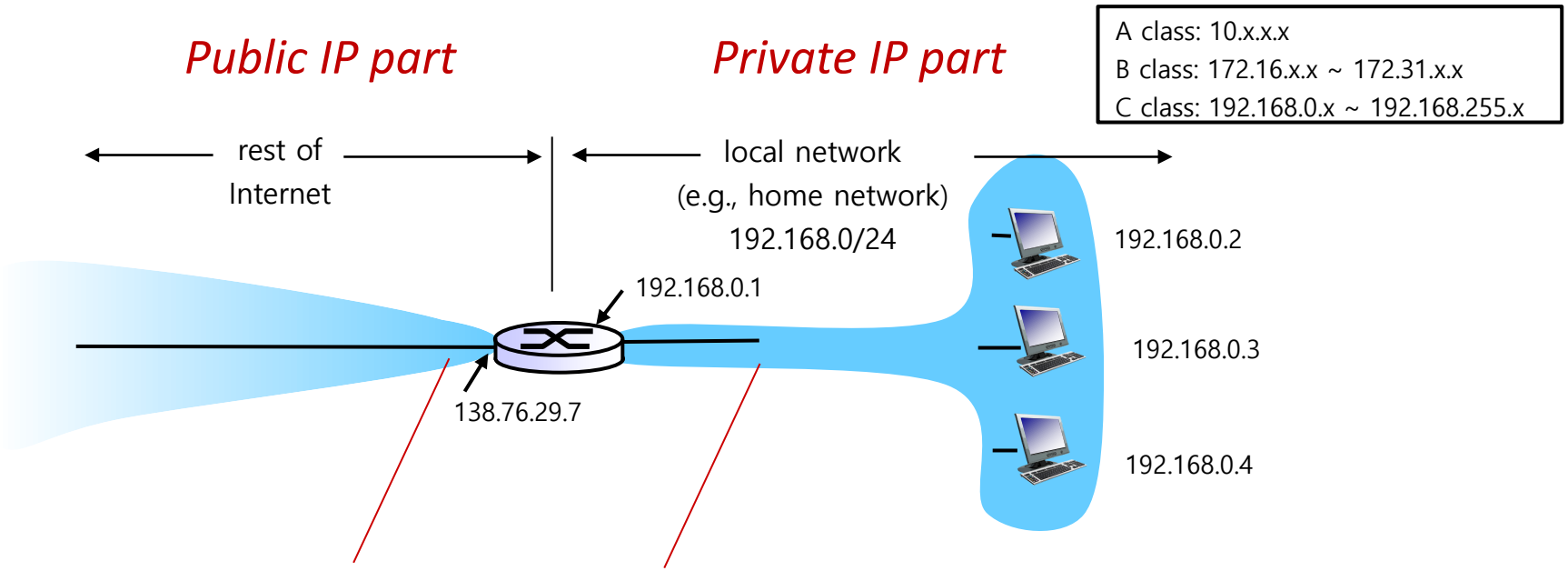
❖ **Motivation: local network uses just one IP address as far as outside world is concerned:**

- range of addresses not needed from ISP: just one IP address for all devices
- can change addresses of devices in local network without notifying outside world
- can change ISP without changing addresses of devices in local network
- devices inside local net not explicitly addressable, visible by outside world (a security plus)



출처 –
https://www.google.co.kr/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwii9YPw74TcAhWUZt4KHYuxABUQjRx6BAgBEAU&url=http%3A%2F%2Fwww.practicalnetworking.net%2Fseries%2Fnat%2Fwhy-nat%2F&psig=AOvVaw3Rck0oh7KTX3cP0a6Pv5w9&ust=1530773925619362

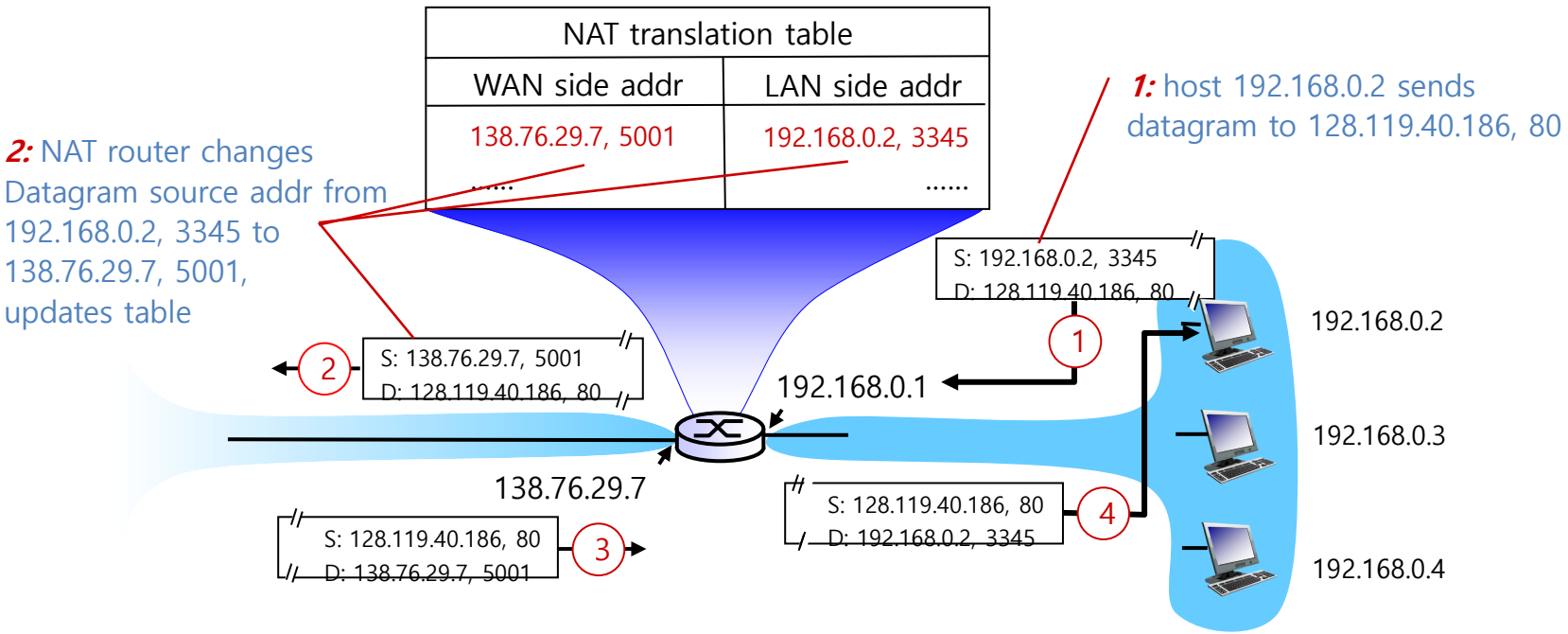
NAT Example



all datagrams *leaving* local network have the *same* single source NAT IP address: 138.76.29.7,different source port numbers

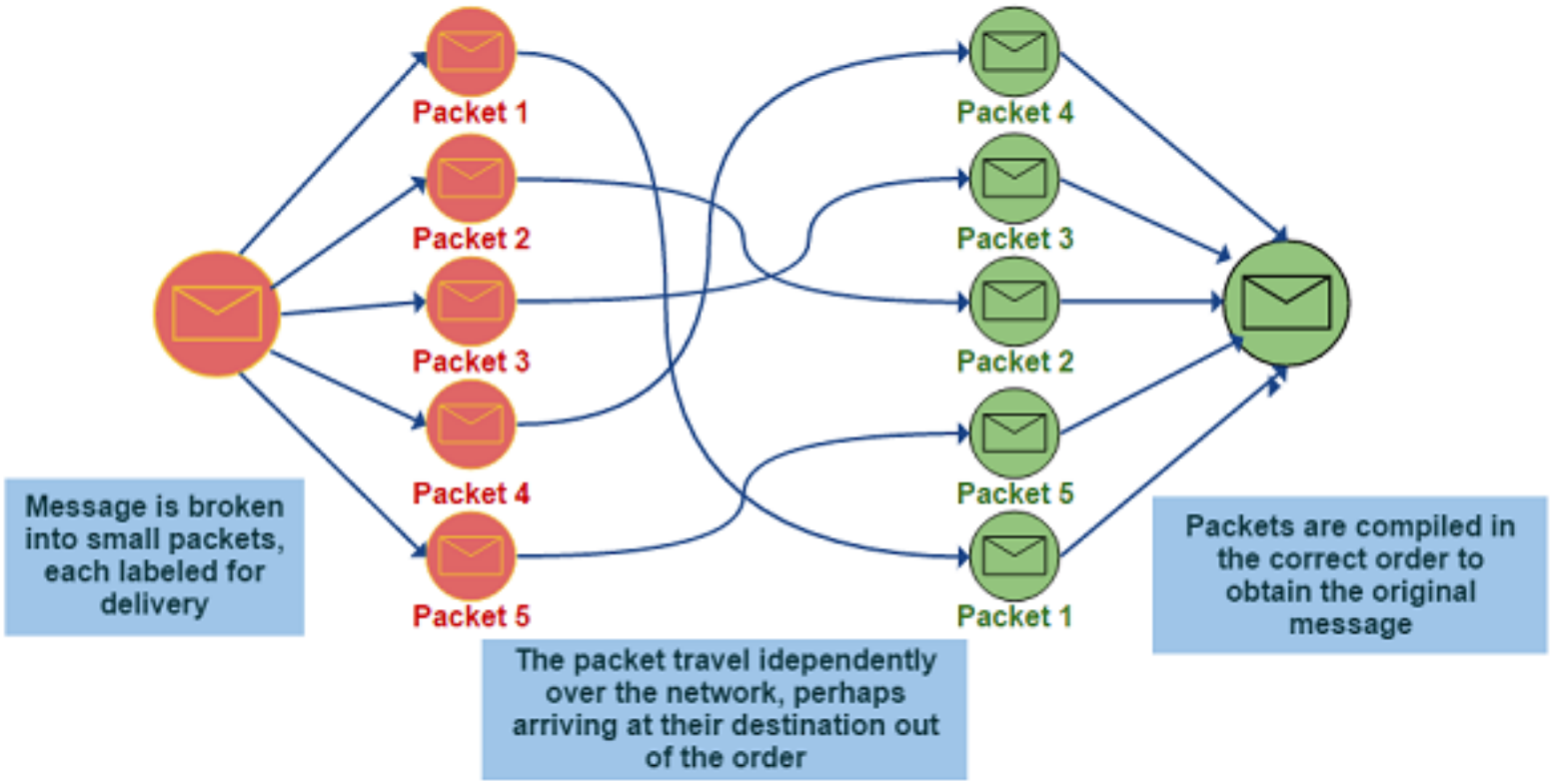
datagrams with source or destination in this network have 192.168.0/24 address for source, destination (as usual)

NAT Example (cont'd)



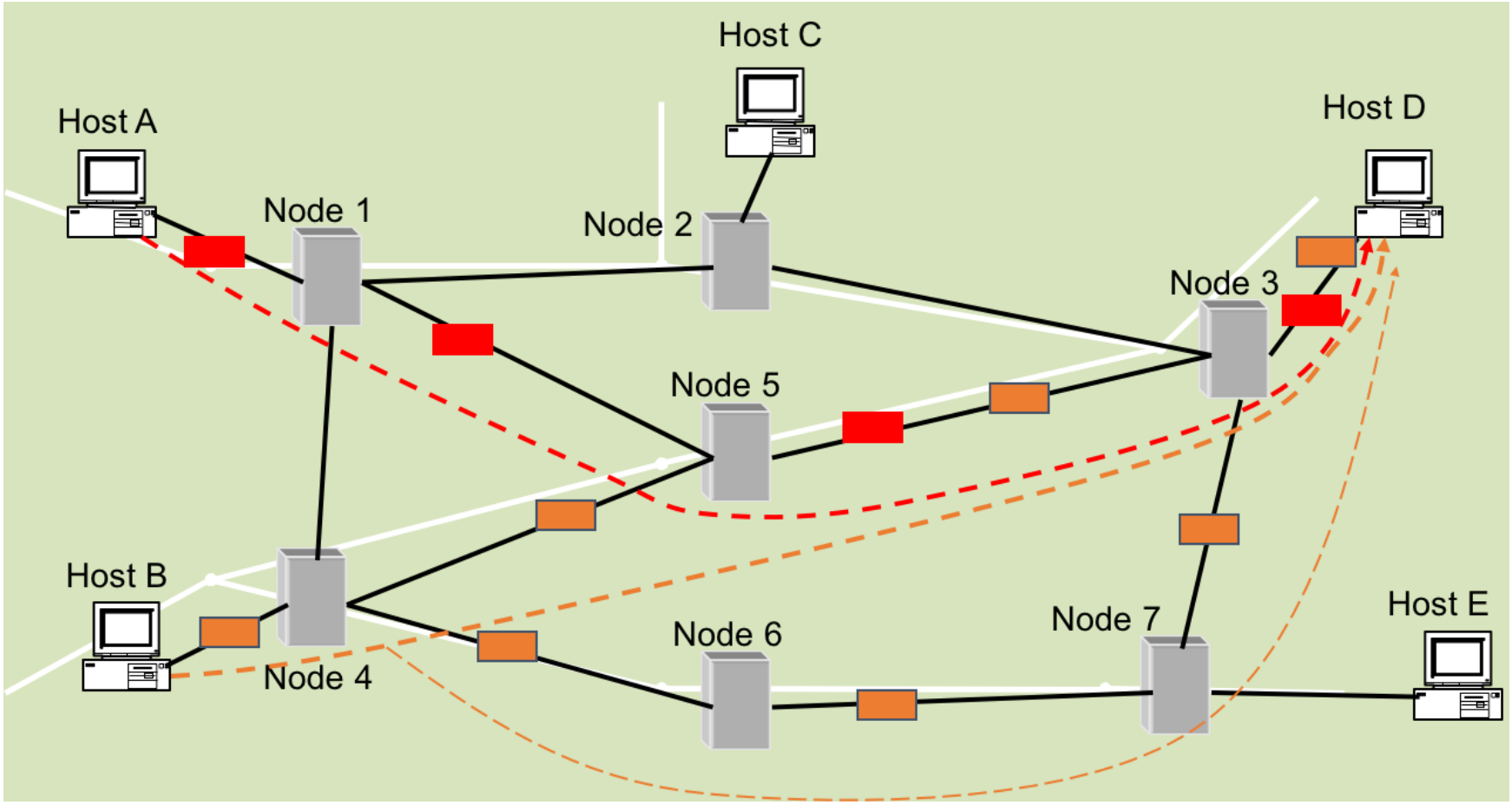
DATA DELIVERY IN INTERNET

Message into Packets



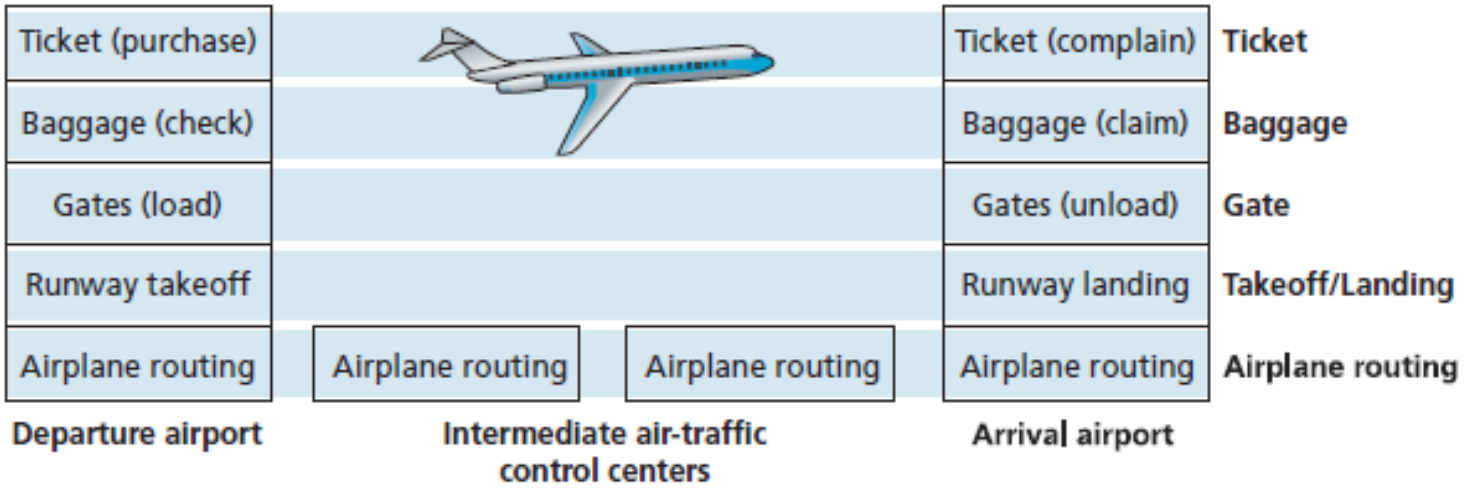
Internet Packets Transmission

Packet Delivery



PROTOCOL STACK

Layering of Airline Travel



❖ **layers: each layer implements a service**

- via its own internal-layer actions
- relying on services provided by layer below

Why Layering?

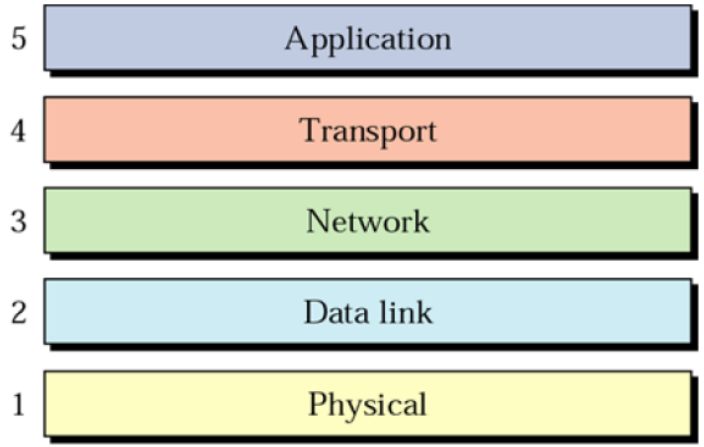
- **Explicit structure allows identification, relationship of complex system's pieces**
  - layered *reference model* for discussion

- **Modularization eases development, maintenance, and 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

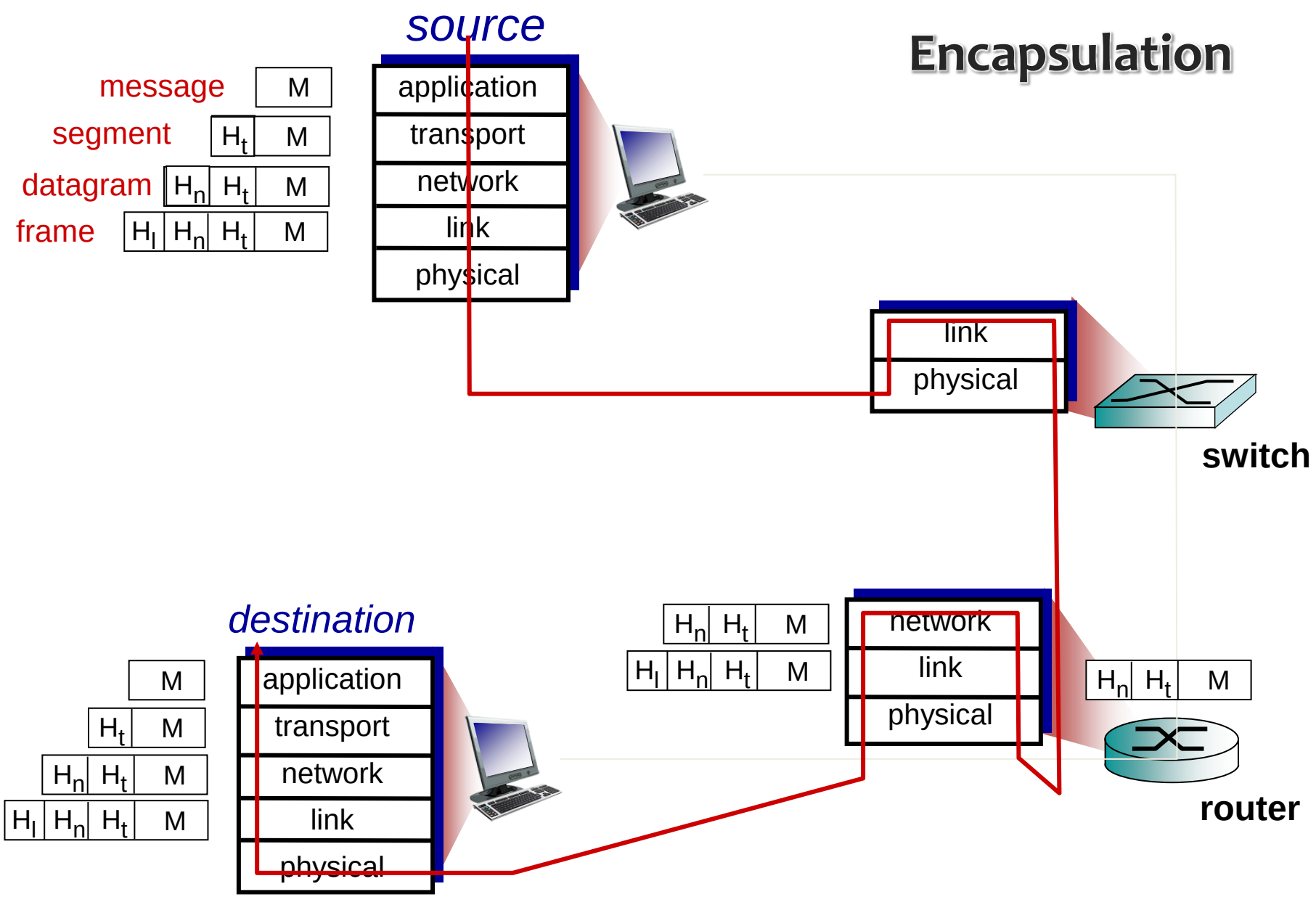
- **Layering considered harmful?**

Internet Protocol Stack

- **application: supporting network applications**
  - FTP, SMTP, HTTP
- **transport: process-process data transfer**
  - TCP, UDP
- **network: routing of datagrams from source to destination**
  - IP, routing protocols
- **link: data transfer between neighboring network elements**
  - Ethernet, 802.11 (WiFi), PPP
- **physical: bits "on the wire"**



Encapsulation



TRANSPORT-LAYER SERVICES

Terminologies: Program, Process, and Thread

❖ Program

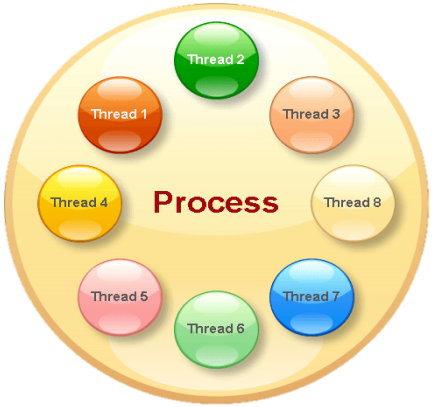
- an executable file containing the set of instructions written to perform a specific job
- stored on a disk

❖ Process

- an executing instance of a program
- resides on the primary memory
- several processes related to same program at the same time

❖ Thread

- the smallest executable unit of a process



출처 - https://www.quora.com/What-is-difference-between-thread-process-and-program

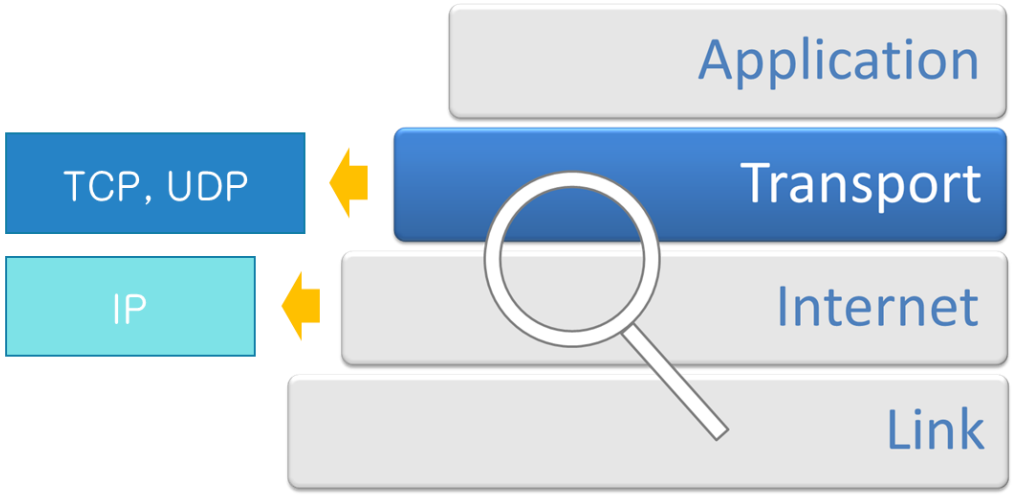
Transport Layer Function (1/2)

Transport layer

- logical communication between processes
- relies on, enhances, network layer services

Network layer

- logical communication between hosts



TCP vs. UDP

TCP

- Transmission Control Protocol
- Reliable, in-order delivery
- Connection-oriented service
  - connection setup
  - error control
  - flow control
  - congestion control

UDP

- User Datagram Protocol
- Unreliable, unordered delivery
- Connectionless service
  - faster than TCP

USER DATAGRAM PROTOCOL

User Datagram Protocol [RFC 768]

❖ **"No frills," "bare bones" Internet transport protocol**

❖ **Connectionless service:**

- each UDP segment handled independently of others
- Unreliable: UDP segments may be lost or delivered out-of-order to app

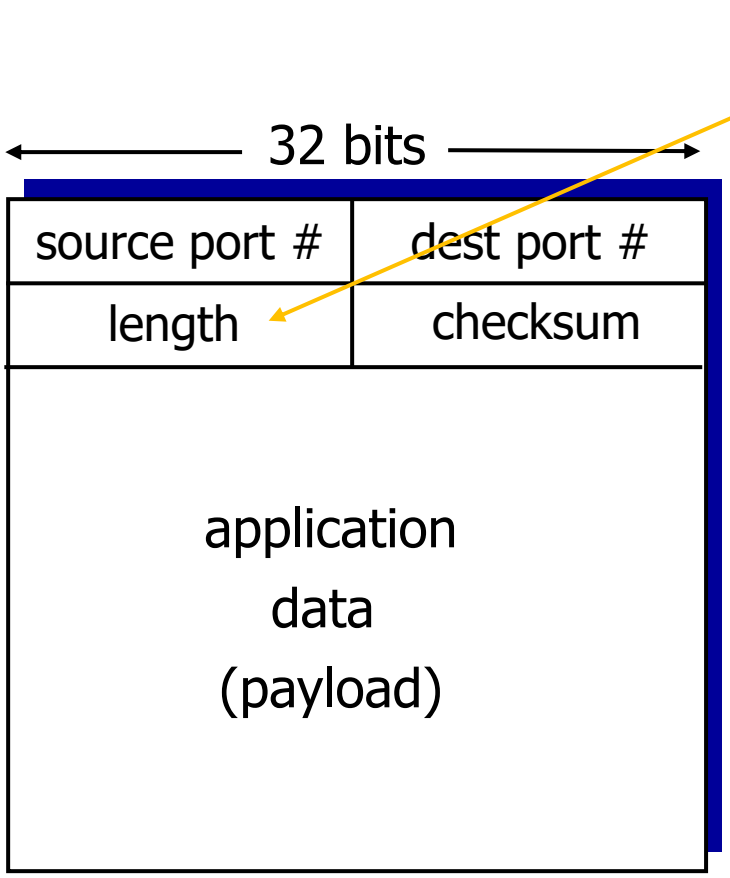
❖ **UDP use:**

- streaming multimedia apps (loss tolerant, rate sensitive)
- DNS
- SNMP

❖ **Reliable transfer over UDP:**

- add reliability at application layer
- application-specific error recovery!

UDP Segment Header



length, in bytes of UDP segment, including header (header plus data)

UDP segment format

❖ Advantages of UDP

- no connection establishment (which can add delay)
- simple: no connection state at sender, receiver
- small header size
- no congestion control: UDP can blast away as fast as desired

TRANSMISSION CONTROL PROTOCOL

TCP Overview

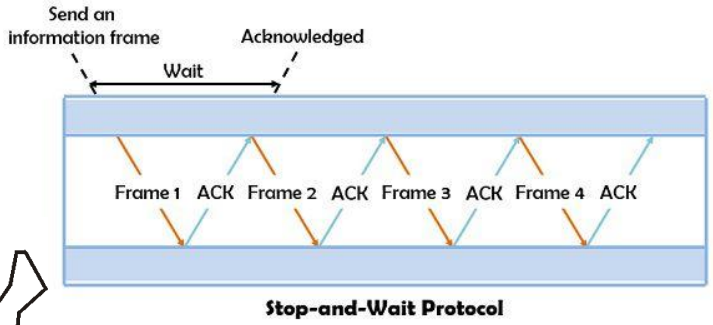
- **Point-to-point: one sender, one receiver**
- **Connection-oriented service**
  - reliable transfer, in-order delivery
  - handshaking initializes sender and receiver state before data exchange

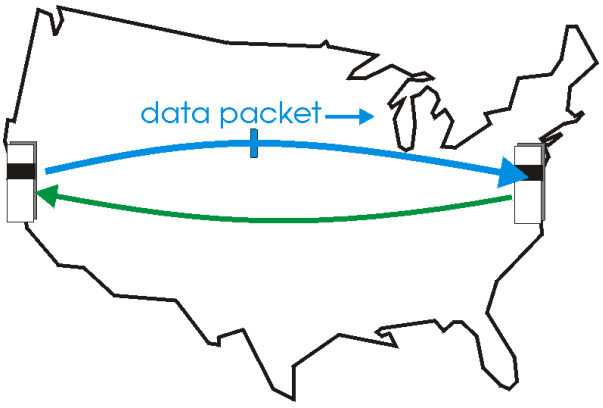
Reliable Transfer

❖ **Simple method: Stop-and-wait**

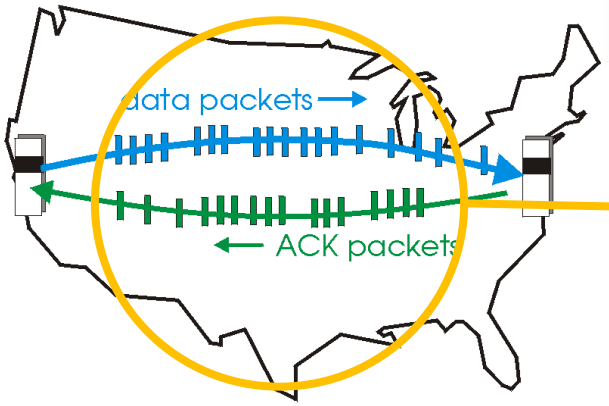
- sender sends one packet, then waits for receiver response
- after receiving ACK, sender resumes transmission
- if timer expires without receiving ACK, sender retransmits the previous packet
- low bandwidth utilization

❖ **Efficient method: Pipelined transmission**





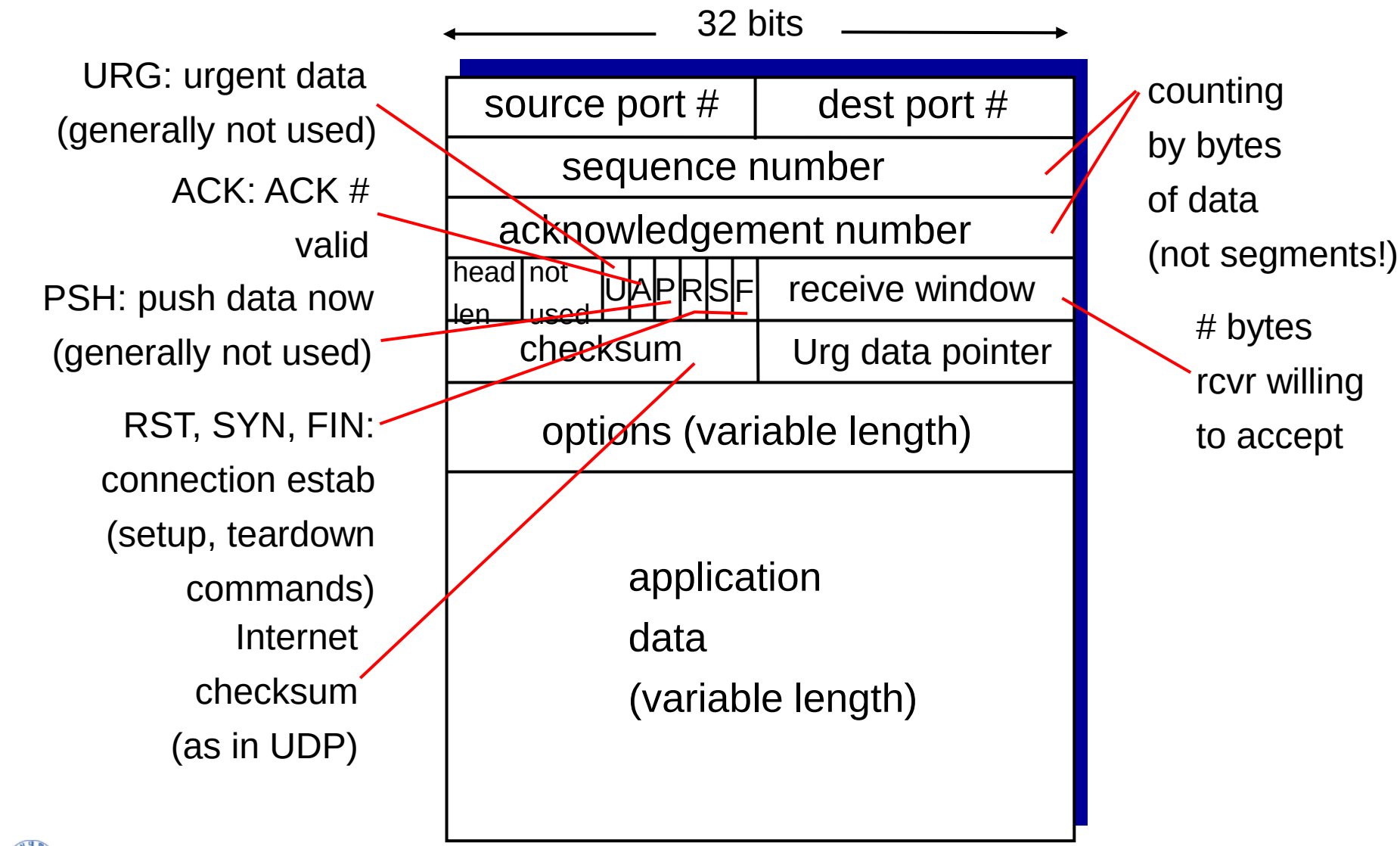
(a) a stop-and-wait protocol in operation



(b) a pipelined protocol in operation

This is the way TCP works!!!

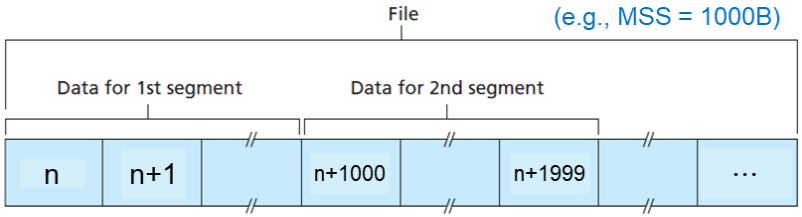
TCP Segment Structure



Sequence and Acknowledgment Number

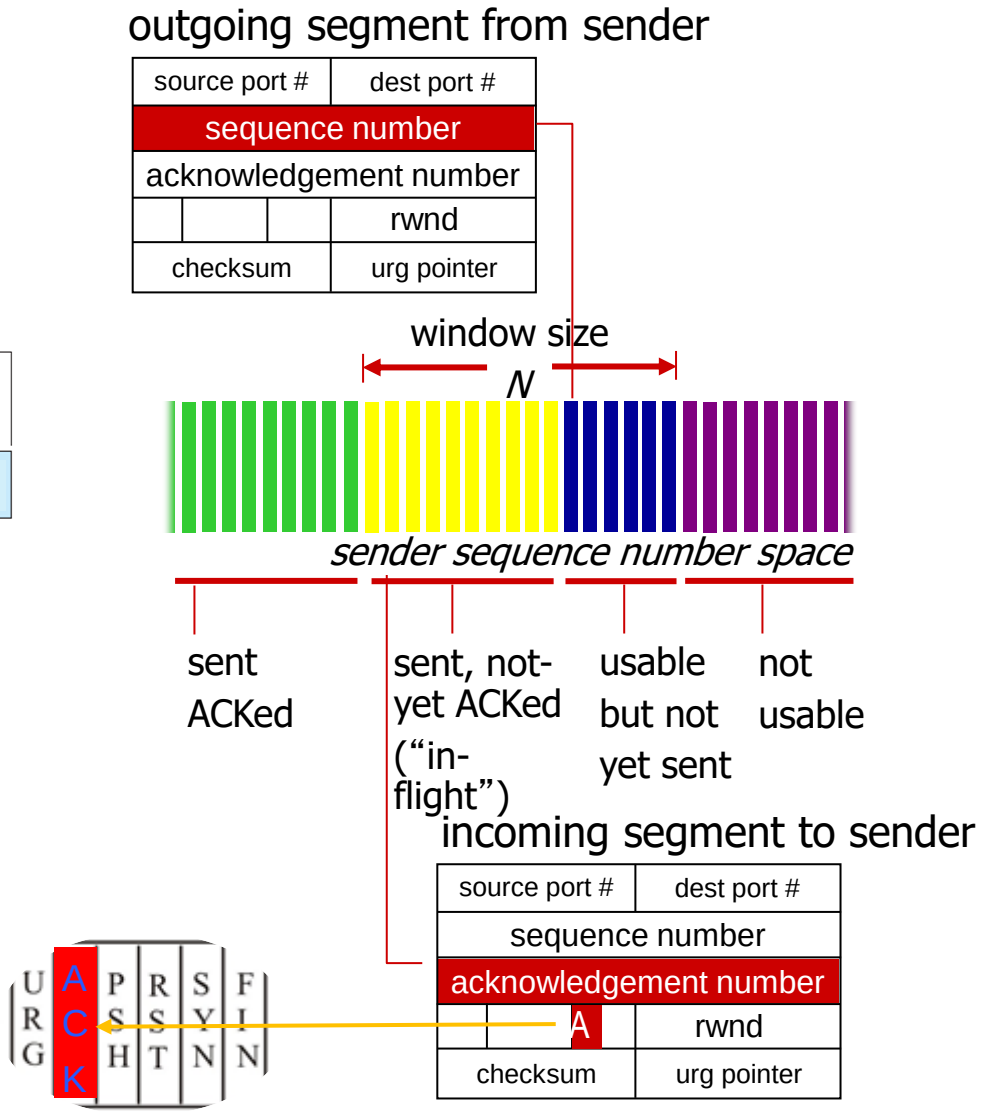
❖ Sequence number

- byte stream "number" of first byte in segment's data

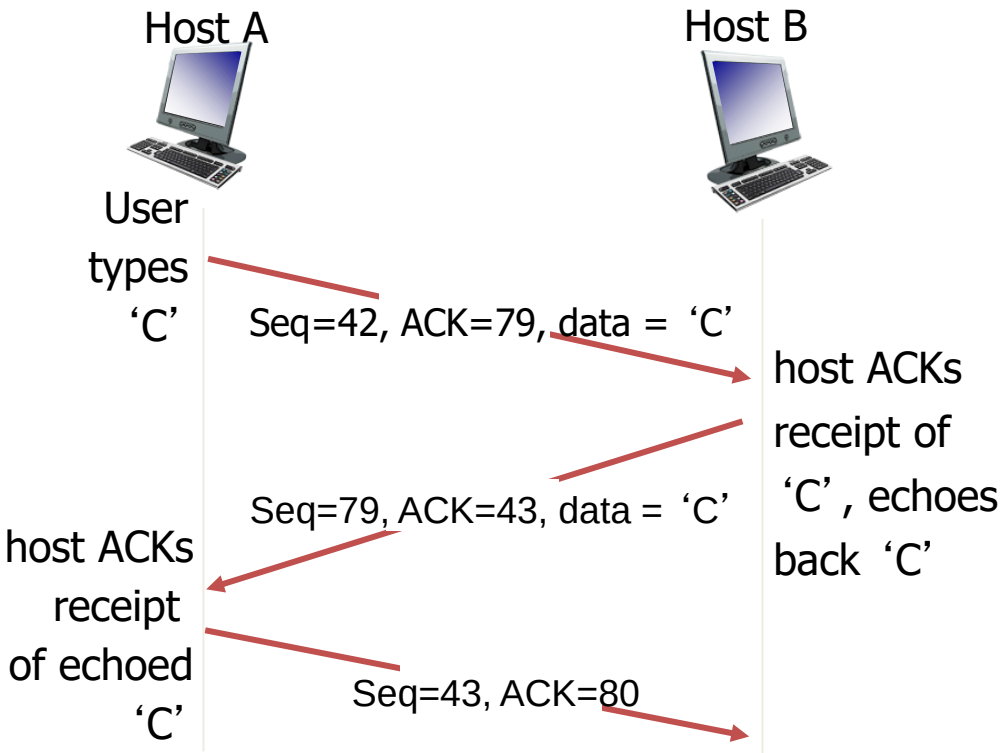


❖ Acknowledgment number

- sequence number of the next segment expected by receiver
- cumulative ACK

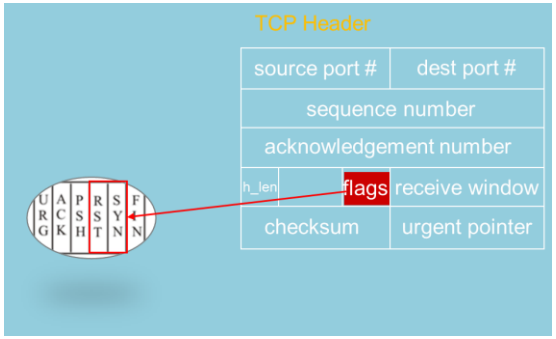


Usage of Seq. & ACK Numbers



simple telnet scenario

Establishing Connection



❖ Three-way handshake

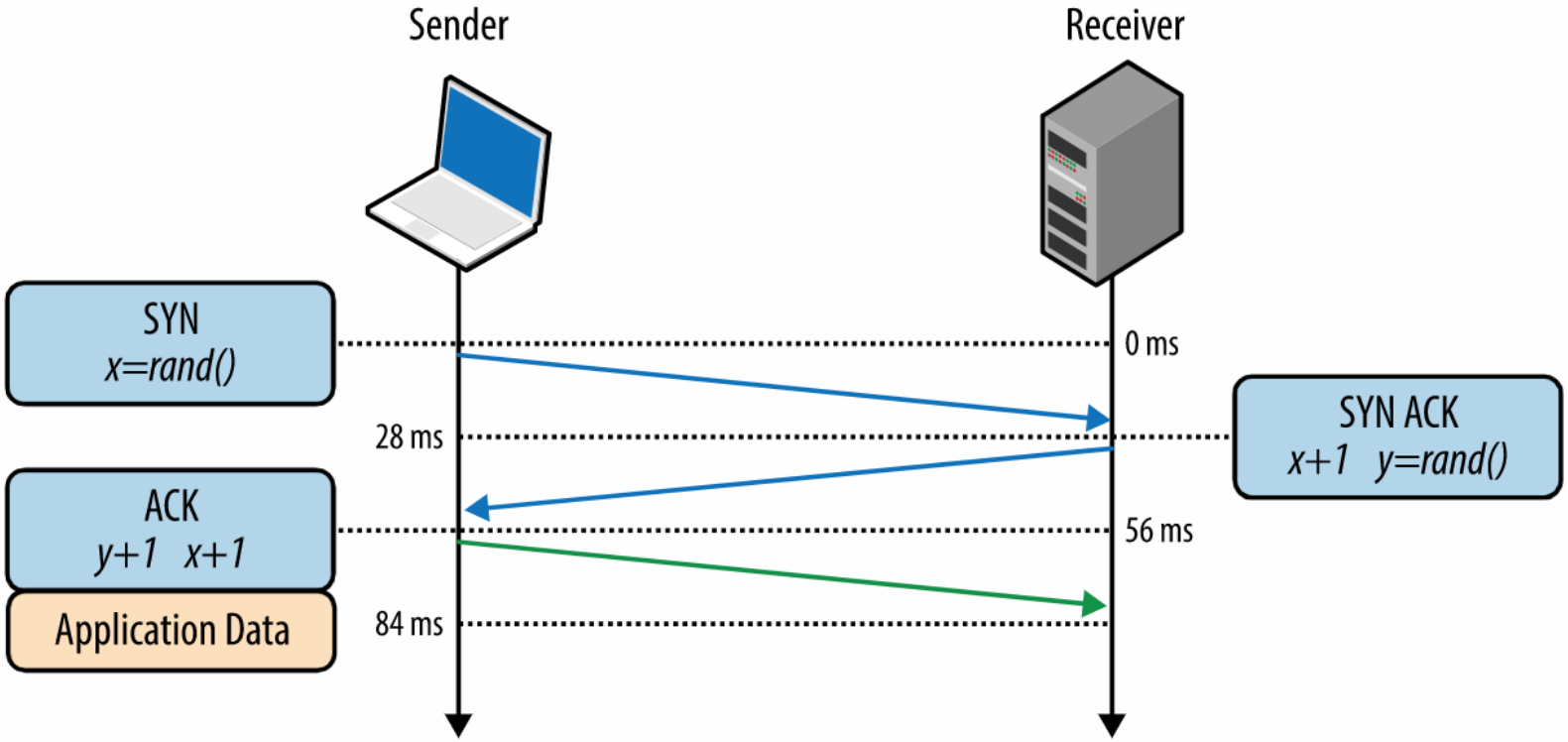
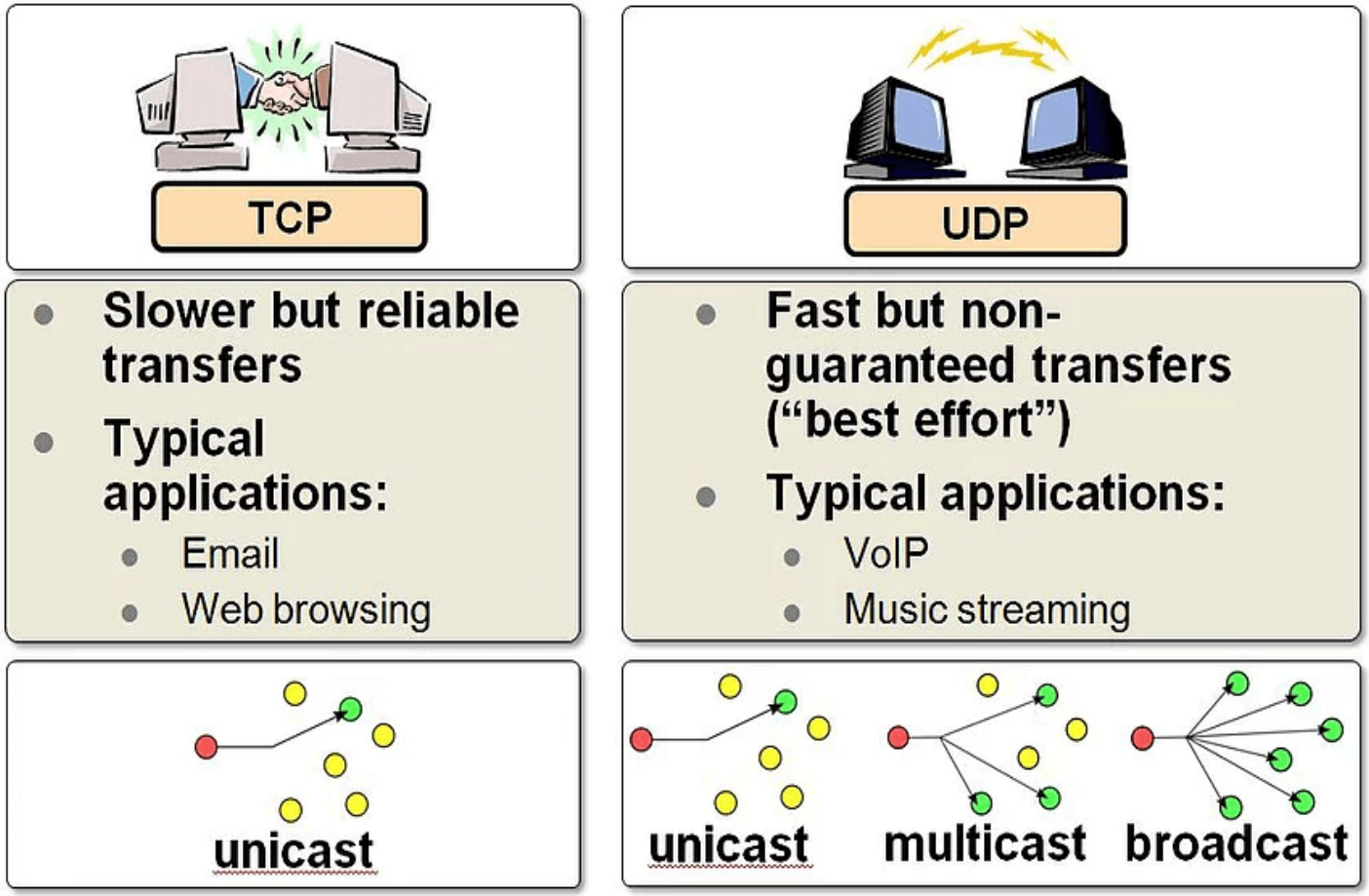


Figure 2-1. Three-way handshake

TCP vs. UDP



출처 -
https://www.google.co.kr/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwju_8aR0PXbAhVGf7wKHfEmAYUQjRx6BAgBEAU&url=https%3A%2F%2Fknowledgeofthings.com%2Ftcpip-vs-udp-internet-protocol-suite%2F&psig=AOvVaw2QofBIqkfTFxG8_J4eyPGI&ust=1530250010414409/