

# 컴퓨터 네트워크 – 2 주차

Jong-Kyou Kim, PhD

2016-10-04

- ▶ 주 2 회
  - ▶ 화 14:00 – 15:30 (prompt 4-11)
  - ▶ 목 14:00 – 15:30 (prompt 4-11)

# 원시 사회에서 빛을 이용한 통신

컴퓨터 네트워크 -  
2 주차

Jong-Kyou Kim,  
PhD



그림: 봉수 (이상)

## 봉수를 이용한 통신

# 봉화 신호 체계

봉수(烽燧)는 밤에는 횃불, 낮에는 연기로 전하는 군사신호체계이다. 총 다섯 개의 화두를 통해 상황을 전달하는데 남쪽 첫 번째 화두부터

- 평상시에는 밤낮으로 봉수 1개
- 적이 국경근처에 나타나면 봉수 2개
- 국경선에 도달하면 봉수 3개
- 국경선을 침범하면 봉수 4개
- 적과 아군 사이에 전투가 벌어지면 봉수 5개를 올렸다.

그림: 봉수의 protocol

# 봉수 운영의 문제점

컴퓨터 네트워크 –  
2 주차

Jong-Kyou Kim,  
PhD



그림: 봉수 (실제모습?)

# 봉수 운영의 문제점



그림: 1970년대 남산?

## 개선책: 파발

컴퓨터 네트워크 -  
2 주차

Jong-Kyou Kim,  
PhD



그림: 파발

# 파발, 너무 늦은 너무 부패한



그림: 조선후기 파발망





## 참고도서

- ▶ 츠토무 토네 저, 이도희 역. 성공과 실패를 결정하는 1%의 네트워크원리. 성안당.



그림: 네트워크원리

- ▶ 프로토콜과 계층화 (layering) 의 목표
  - ▶ Interoperability: PC 와 슈퍼컴퓨터가 사이좋게
  - ▶ Reuse: PC 에서 개발한 코드를 슈퍼컴퓨터에도
  - ▶ Encapsulation (of underlying details): WiFi, 3G, ...
- ▶ 프로토콜 스택: 프로토콜 계층화의 구현체
- ▶ OSI 7 Layer: 네트워크를 이해하는 아키텍처

# 일상에서의 프로토콜

컴퓨터 네트워크 -  
2 주차

Jong-Kyou Kim,  
PhD



그림: 약속대련

## 좀더 현실적인 예

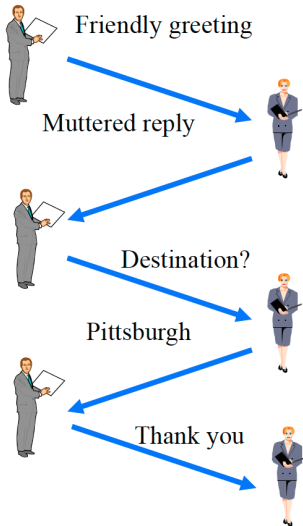


그림: 프로토콜

- ▶ 양자간에 서로 합의한 약속
- ▶ 통신을 위해서는 여러가지가 합의되어야 함
  - ▶ language, encoding, request, reply, acknowledgement, ...
  - ▶ 예외상황, 종료, 순서보장
- ▶ 컴퓨터로 구현하려면 계층화가 필요하다
  - ▶ 통신의 주체: 소프트웨어
  - ▶ 신호의 전달: 하드웨어

# 네트워크 인터페이스 카드

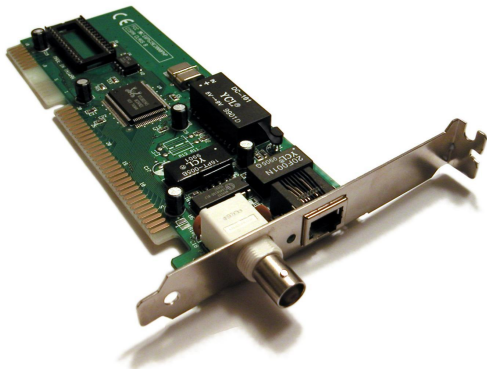


그림: Network Interface Card

# 하드웨어 인터페이스

```
$ ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
    options=3<RXCSUM,TXCSUM>
    inet6 ::1 prefixlen 128
    inet 127.0.0.1 netmask 0xff000000
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
    nd6 options=1<PERFORMNUD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
```



# 하드웨어 인터페이스

```
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether b8:e8:56:04:b5:f2
    inet6 fe80::bae8:56ff:fe04:b5f2%en0 prefixlen 64 scopeid 0x4
    inet 10.73.43.??? netmask 0xfffffe00 broadcast 10.73.43.255
    nd6 options=1<PERFORMNUD>
    media: autoselect
    status: active
```

# 잘못된 상상

```
main() {  
    int fd = open("en0");  
    char buf[BUFSIZ];  
    write(fd, "http://google.com/bigfile");  
    for (;;) {  
        read(fd, buf);  
        /* store result */  
    }  
    write(fd, "http://naver.com/bigfile");  
    for (;;) {  
        read(fd, buf);  
        /* store result */  
    }  
}
```

# 기본적인 Layer 구성

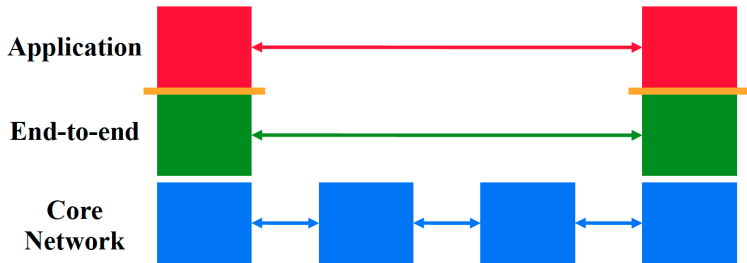


그림: Service Levels

# OSI Motivation

- ▶ 추상화 방법론: 계층화 (Layering)
  - ▶ 상위수준의 요청을 하위 계층에서 처리
  - ▶ 통신은 동일한 수준의 계층간에만 이루어짐
  - ▶ 한 계층의 변경이 다른 계층에 영향을 주지 않음
    - ▶ 유선 → 무선으로 전환: HTTP 이용에는 영향을 주지 않음
    - ▶ i.e., 무선인터넷이 바뀐다고 web browser 를 새로 띄워야 하는 것은 아님

# OSI Motivation

- ▶ 각 계층은 상위계층에는 서비스를 제공한다
- ▶ 각 계층의 구현은 바로 아래 계층에서 제공하는 서비스만을 이용한다
- ▶ 다른 시스템과 통신하는 경우 반드시 대응되는 계층과만 통신한다
  - ▶ 통신시에는 계층간에 정의된 규약만 사용한다
- ▶ 현실에서는?
  - ▶ 그저 모델일뿐.
  - ▶ 예: TCP/IP 는 4 계층으로 구성
  - ▶ 그러나 컴퓨터 네트워크를 이해하는 기본임!

# OSI functions

- ▶ Physical: 비트전달 (WiFi 전파)
- ▶ Data link: 오류탐지, 수정 (WiFi card)
- ▶ Network: 스위칭 (WiFi 장비)
- ▶ Transport: Application 간 데이터 전송
- ▶ Session: Logical connection
- ▶ Presentation: data transformation
- ▶ Application: specific use

# Protocols

- ▶ Hop by hop/link protocol (physical, data link)
  - ▶ Ethernet
- ▶ End-to-end (transport, session, presentation, application)
  - ▶ TCP/UDP
- ▶ Management (network, ... )
  - ▶ IP
  - ▶ 약간 애매모호한 영역

→ 절대적인 분류는 아님

# 프로그래머가 보는 네트워크

- ▶ Application
  - ▶ client: web browser, outlook, WoW
  - ▶ server: web server, outlook server, WoW server
- ▶ Presentation/session
  - ▶ 대부분 application 에 포함됨
  - ▶ 주된 공헌의 형태: **library**
- ▶ Transport/network
  - ▶ **대부분** OS 에 포함됨
- ▶ Data link
  - ▶ 하드웨어와 소프트웨어가 혼합되어 구현됨
- ▶ Physical: 하드웨어로 구현됨