

network application protocols 개념적인, 구현적인 측면.

- transport-layer service model.
- client-server paradigm.
- peer-to-peer paradigm.

대표적인 app layer protocol.

Net app 만들기.

## 2. / principles of network application.

- app architectures
- app requirements.

## Some network apps.

- e-mail
- web
- text messaging

## Creating a network app.

프로그래밍 작성할 비용.

- end-systems에서 작동.
- net 통해 통신
- web server는 browser와 통신.

작성할 필요 있는 비용

- network-core device 비용
- (기본 원칙은 같은 layer끼리 소통!)

## Application architectures.

- client-server.
- peer-to-peer (P2P)

## Client-server architecture.

• Server

- 항상 켜져!
- IP 불변.
- 데이터를 위한 데이터 센터

• Client.

- 서버와 통신.
- 간단한 응용 프로그램.
- IP 바뀔 수 있음.
- client끼리 직접적으로 통신 안함.

# Processes communicating

통신 방식 1.

동일 인스턴스에 있는 두 프로세스가 OS에서  
정의된 **inter-process communication**을 이용하여  
통신하는 것.

통신 방식 2.

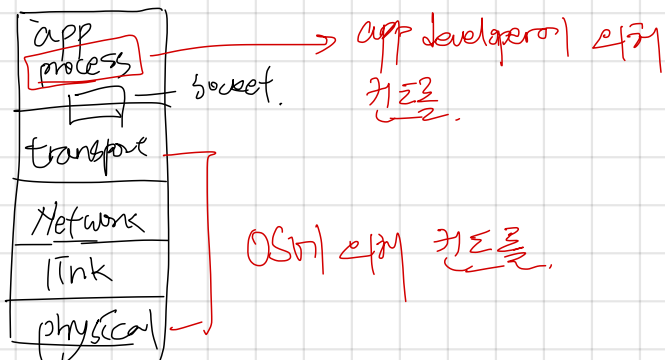
다른 인스턴스에 있는 프로세스간 **message**를 교환  
하여 통신하는 것.

client-server model에서

client가 먼저 통신을 시작하고, server는  
면접하기를 기다리는 상태가 된다.

## Socket

process는 자신의 socket을 통해 message를  
보내거나 받는다. (분리 유사)



# Addressing processes

process가 메시지 받으려면 identifier 필요  
- host device이든 외부 IP address 존재,  
IP만으로는 host내에 있는 모든 process를 구분할  
수 없을까?  
No! 동일 host에서 서로 다른 프로세스가  
동작 가능함.

따라서,  
프로세스를 구분하는 identifier인 인스턴스 연관된  
IP address와 port Number가 존재.

## App layer protocol defines.

즉, app-layer 프로토콜이 규정짓는 요소...  
해당 요소를 정의해야 한다.

① 교환되는 message 종류.  
- request  
- response

② 메시지 문법.  
- 필드의 속성과 value.

③ 메시지 의미  
- 필드 정보의 의미가 무엇인가? (문법은 표현 방법이라면 의미는 그 표현의  
뜻)

④ 언제 어떻게 프로세스가 message를 받고 보내야 하는지

⑤ open protocols (공개 프로토콜)  
ex) HTTP, SMTP...

⑥ 소유권이 있는 protocols  
ex) Skype.

# What transport service does an app need?

전송시 app들은 어떤 서비스를 요구하기을까?

## ① data integrity

- 100% reliable data.  
file transfer, web transactions. . .
- tolerate some loss.  
audio.

## ② timing

## ③ throughput

- 비디오는 연속성이 보이기 위해 일정한 전송속도 요구.
- e-mail은 탄력성이(편차) 큰 전송 속도 허용.

## ④ security

# Internet transport protocols service

TCP service. transport layer의 TCP protocol이

- reliable transport. 제공하는 기능.
- flow control
- congestion control
- does not provide.
  - timing
  - minimum throughput guarantee
  - security
- connection-oriented.

HTTP가 이러한 protocol을  
선택할지 때문에 TCP 사용 가능.

## UDP service

- Unreliable data transfer.
- TCP에서 제공해주는 상위 4개 서비스 제공 안함.

## 2.2 Web and HTTP

### Web and HTTP

· Web은 objects를 전송.

· objects는 HTML file.

- JPEG Image

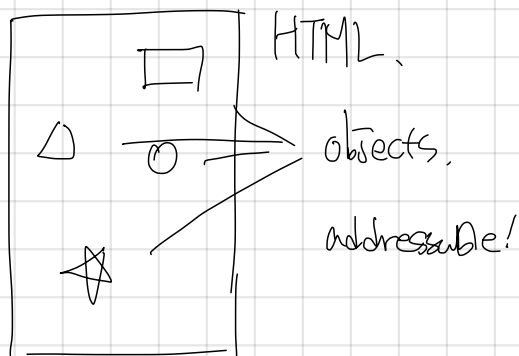
- Java applet

등이 있을 수 있다.

· 기본 HTML-file 밖대로 참조된 objects를 구성된다.

· object는 URL이 의미 있는 것 (addressable) 할 수 있다.

www-someschool.com / someDept / pic.gif.  
└── host name ─┘ └── path name ─┘



### HTTP overview.

· HTTP: Hyper-text transfer protocol.

· web application layer protocol

· client/server model

· TCP 이용.

- port 또한 tcp 이용

- client와 tcp connection 이용

→ browser(client) ↔ web server(HTTP server)

HTTP messages

- 주고 받고 TCP 연결 종료.

· HTTP is "stateless"

- client의 이전 상태를 유지하지 않음.

### HTTP connections

- non-persistent HTTP.

HTML page에 있는 object 1개 당 1번의 connection을 하여 connection 끊어

- persistent HTTP

multiple objects가 한 번의 TCP connection으로 objects를 보냄.

