



# 소켓 네트워크 프로그래밍 이해

뇌를 자극하는 TCP/IP 소켓 프로그래밍



# 3장. 소켓 네트워크 프로그래밍에 대한 이해

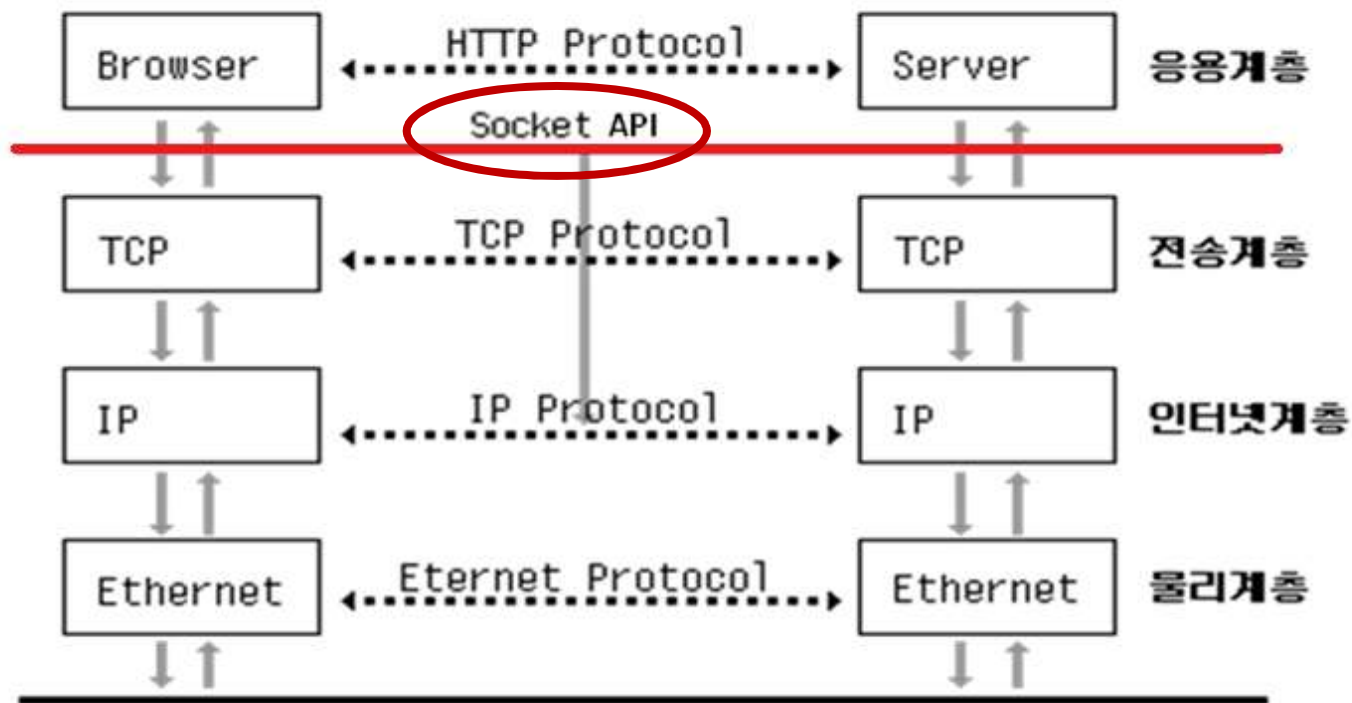


1. 소켓이란 무엇인가 ?
2. 서버/클라이언트 모델
3. 인터넷 서비스 포트



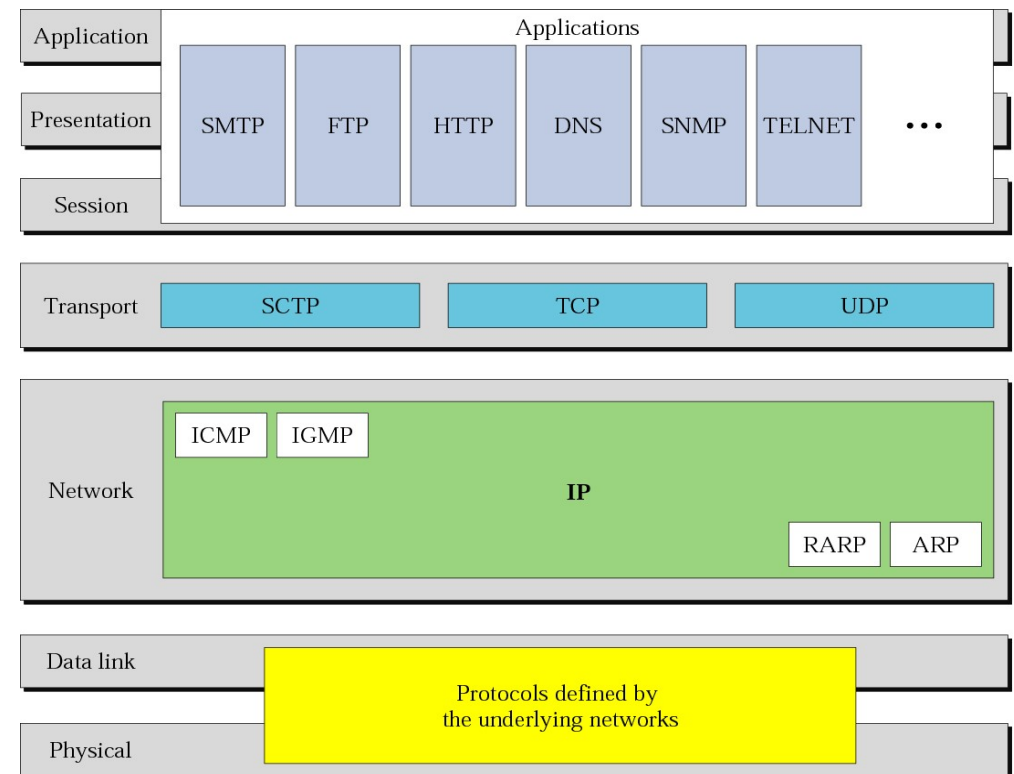
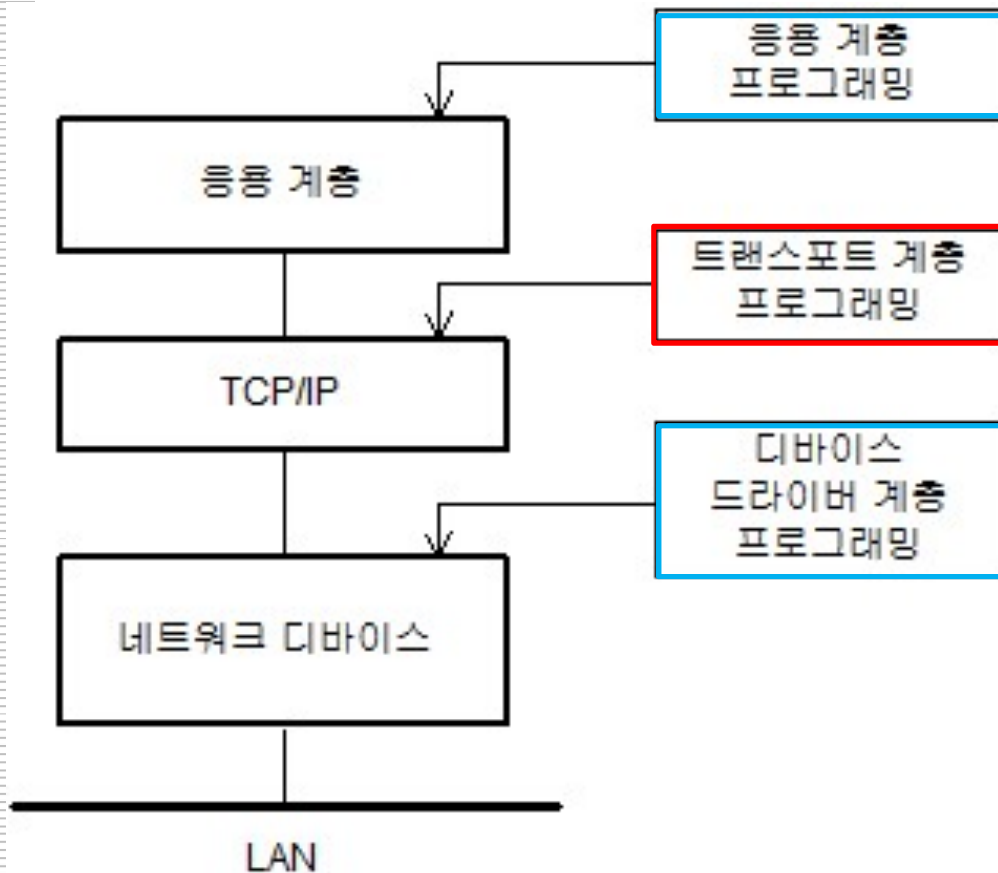
# 소켓이란 무엇인가.

- TCP/IP 4계층에서 전송계층위에 놓인다.
- 전송 계층위에서 전송계층의 프로토콜을 이용하여 데이터 통신을 제공하는 인터페이스 제공



# 소켓이란 무엇인가.

- 네트워크 프로그래밍의 계층별 분류



# 소켓이란 무엇인가.



- 전송 계층 아래를 모두 추상화 하는 **고수준 네트워크 프로그래밍 인터페이스**를 제공한다.
- **Socket Network Programming**
  - Socket API에서 제공하는 소켓 함수로 프로그램을 개발
- 필요할 경우 네트워크 계층 프로토콜(**IP 수준**)을 직접 제어하기 위한 인터페이스를 사용할 수도 있다.
  - 패킷을 캡취하거나 **ICMP**와 같은 프로토콜을 직접 제어하기 위한 목적.



# BSD 소켓(버클리 소켓)

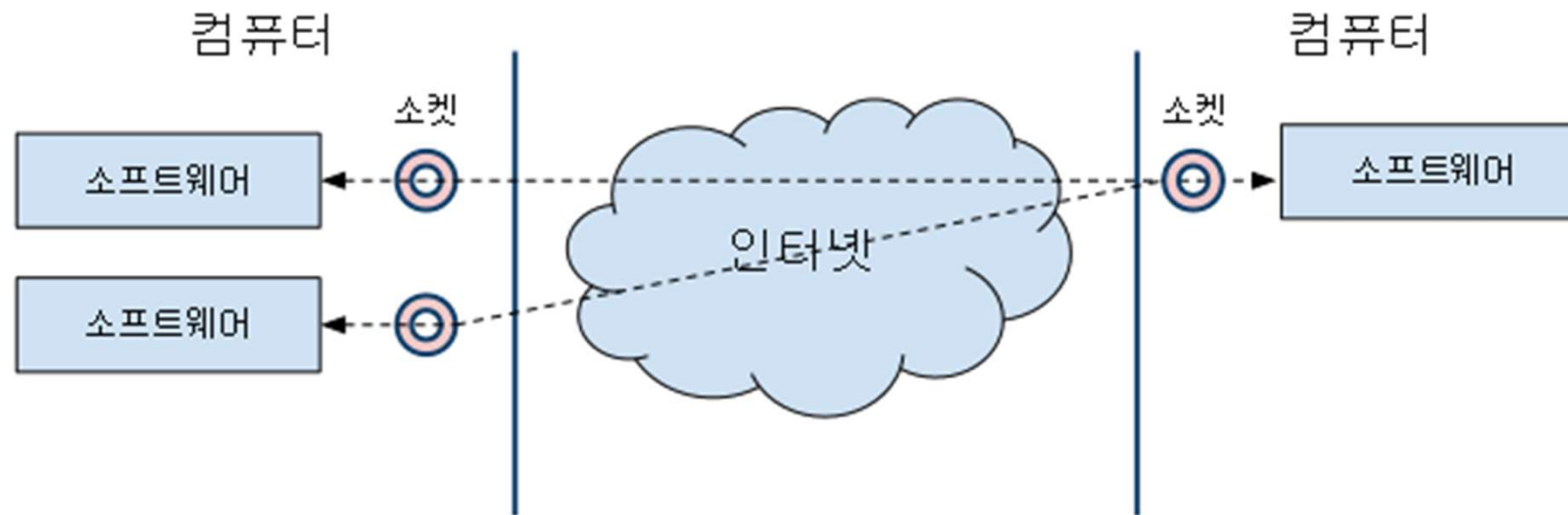


- UC Berkeley 대학에서 개발. 1989년
- C 언어로된 인터페이스를 제공
- 자유롭게 사용할 수 있도록 공개
  - AT&T에서 상업적인 용도로 사용하기 위한 독자적인 네트워크 라이브러리를 개발할 목적이었으나, **BSD Socket**으로 인하여 현재의 개방된 라이브러리가 사용됨.
- **UNIX, LINUX, Windows, Mac OS, Android를 포함한 거의 모든 OS가 소켓을 사용함**
  - 어떤 운영체제에서 만들어진 네트워크 프로그램을 (수정 없이) 다른 운영체제에서 사용할 수 있음



# 소켓이 하는 일

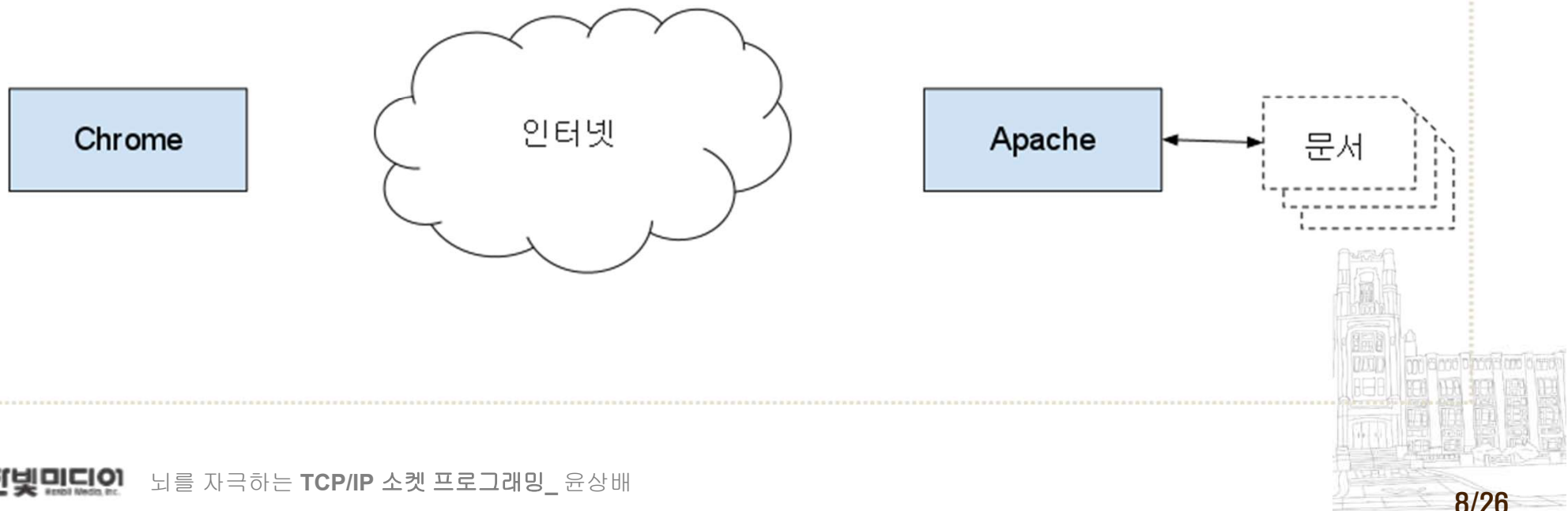
- 소프트웨어와 소프트웨어를 연결 : 인터넷 접점
- 소켓은 프로그램(**프로세스, 스레드**)과 프로그램이 연결되어서 통신할 수 있도록 통신선로를 만드는 역할





# 소켓이 하는 일 : WWW를 예로

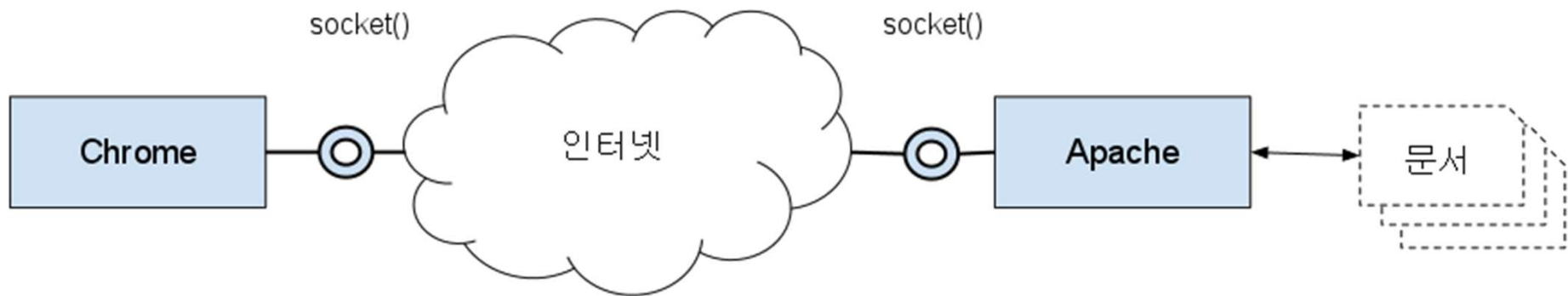
- Apache 웹 서버와 Chrome 브라우저가 인터넷을 사이에 두고 떨어져 있다.
- Apache 웹 서버 : 웹에서 HTML 문서 서비스를 위해서 사용되는 인터넷 프로그램.
- Chrome 브라우저 : 웹 브라우저로 웹 서버에 문서를 요청하고, 이를 Rendering 한다.





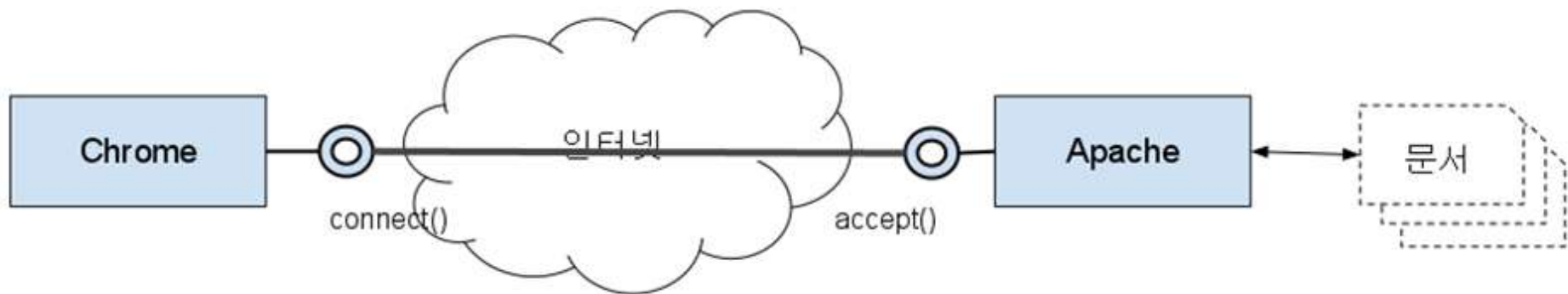
# 소켓이 하는 일

1. 인터넷에 연결하기 위해서 웹서버와 브라우저가 모두 **socket()** 함수를 이용 소켓을 만든다.



# 소켓이 하는 일

2. 기다리는 측은 **accept()** 함수를 이용해서 기다린다.
3. 연결하는 측은 **connect()** 함수를 이용해서 연결을 시도한다.
4. **connect** 함수를 호출하면, 인터넷의 경로시스템을 이용 **apache** 소켓과 연결하고 연결 통로가 만들어진다.



5. 웹서버와 브라우저는 **read(), write()** 을 이용하여 데이터를 전송한다.



# 네트워크 연결 모델

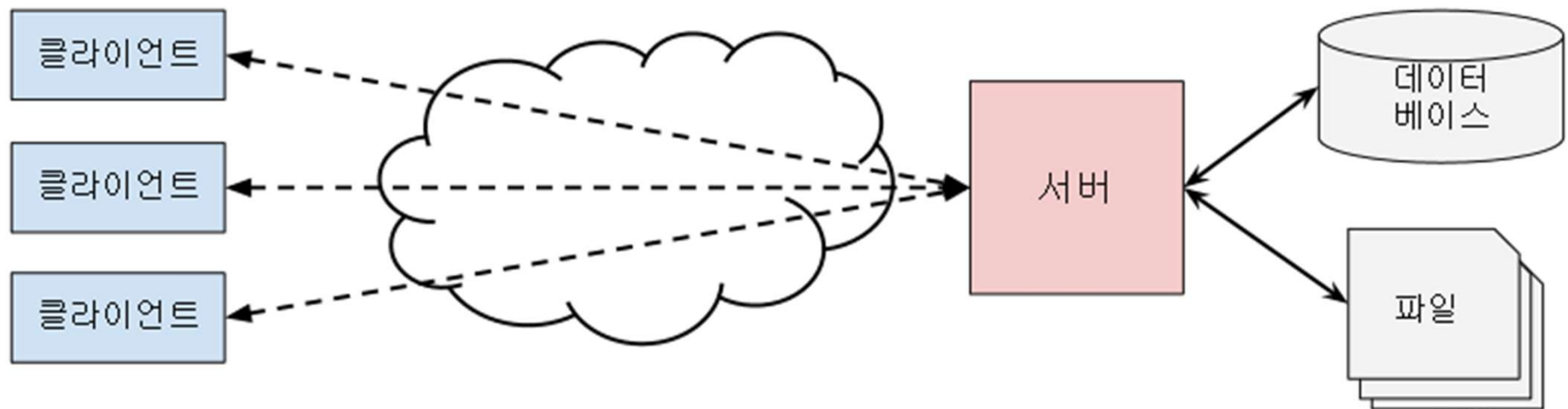


- 인터넷은 프로그램과 프로그램의 연결
- 연결에는 다양한 방식이 있다.
  - Server & Client 모델
  - Manager & Agent 모델
  - P2P 모델



# 서버 & 클라이언트 모델

- 고객 & 서비스 제공자 모델 이라고도 한다.
- **서버(server process)** : 데이터를 관리하고 서비스를 제공하는 네트워크 프로그램
- **클라이언트(client process)** : 서버에 정보를 요청하는 프로그램
- 중앙 집중형 정보 제공 방식



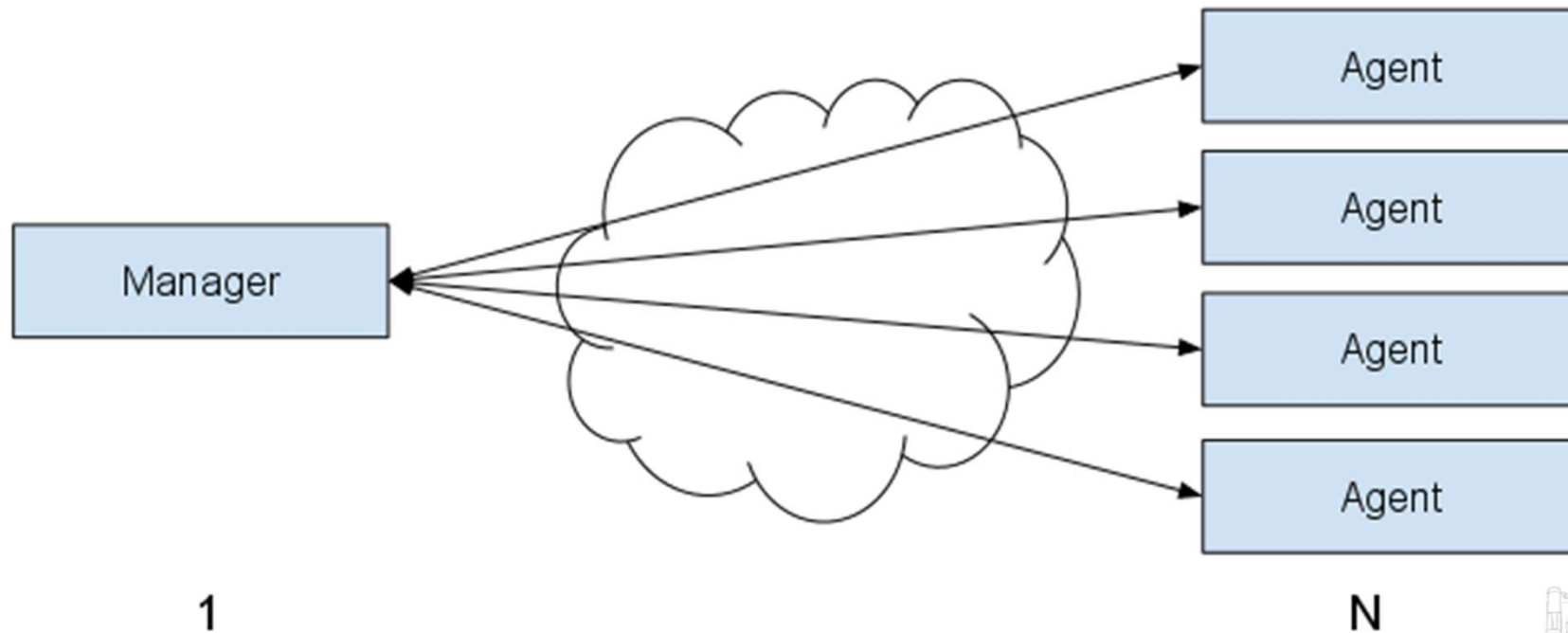
# 서버 & 클라이언트 모델

- 실 세계에서 의 매장 <-> 고객 시스템의 인터넷 확장
- 상업적으로 정보를 판매하는데 유리한 모델이기 때문에 초기 부터 지금까지 널리 사용하고 있다.
- 서버에서 정보를 집중 관리하기 때문에, 서버 성능이 중요하다.
- 정보가 집중되기 때문에 악의적인 공격과 보안에 취약한 측면이 있다.
- 서버 성능을 높이는데 한계가 있기 때문에, 대량의 데이터를 처리해야 할 경우 서버 성능 확장에 많은 비용이 소비된다.



# Manager & Agent 모델

- Manager 프로그램과 Agent 프로그램으로 구성
- Manager 프로그램이 Agent 프로그램으로 정보 요청



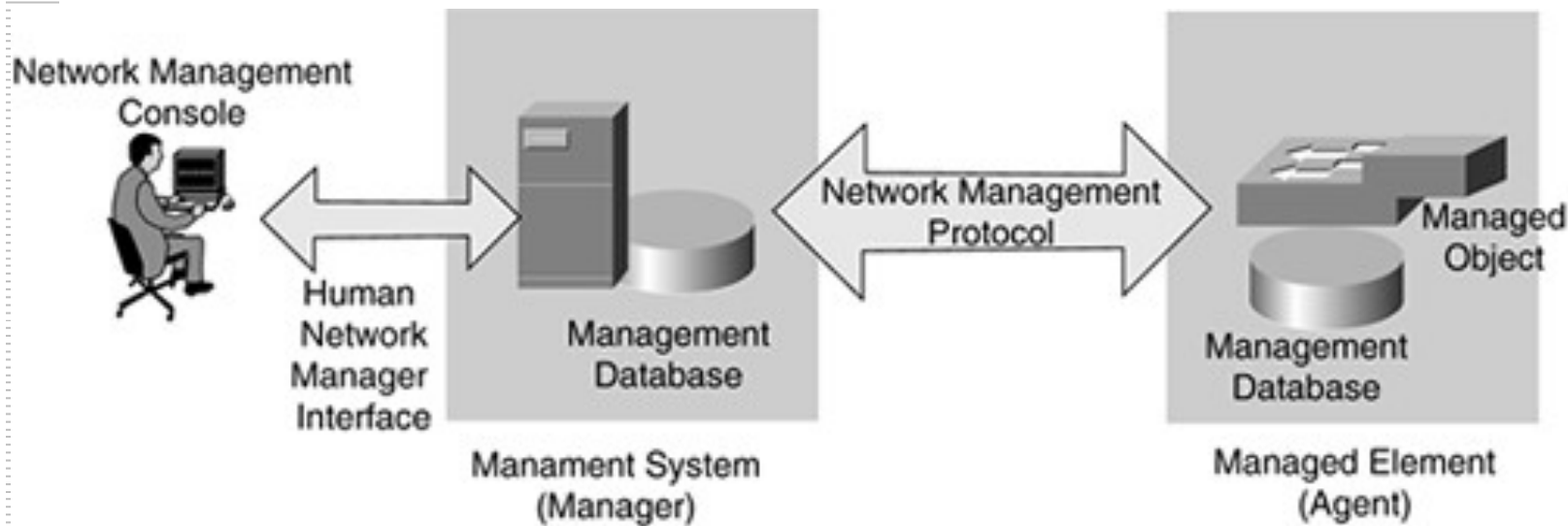
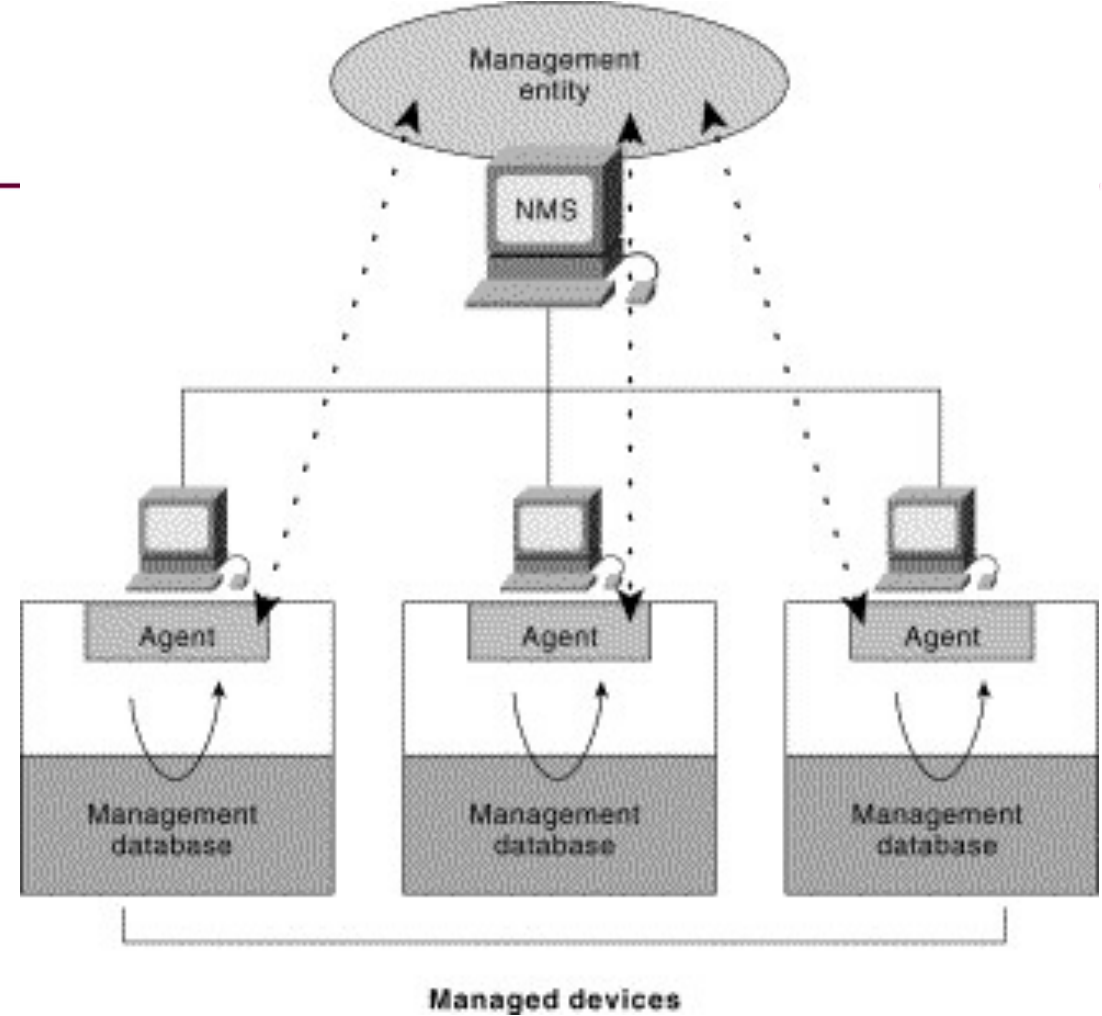
# Manager & Agent 모델

- 서버 & 클라이언트 모델 - 1(server) : N(client)
- Manager & Agent 모델 - 1(manager) : N(agent)
- **Manager**가 다수의 **노드(agents residing on nodes)**로부터 주기적으로 정보를 수집하고 명령을 내리는 일에 적합
  - **SNMP(Simple Network Management Protocol)** - 기업규모의 네트워크에서 네트워크 장비와 호스트를 관리하기 위하여 사용되는 프로토콜





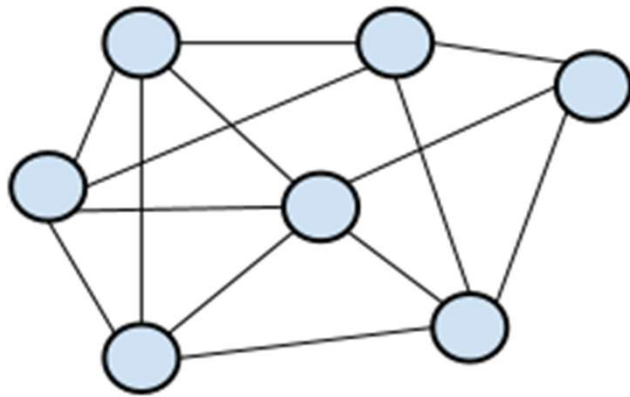
# Manager & Agent 모델



# Peer To Peer



- 동등 계층 모델
- P2P로 알려져 있다.
- 모든 노드가 동등한 자격으로 네트워크에 참여



# Peer To Peer



- 유저간 동등한 정보 교환으로 인터넷의 철학을 구현한 모델로 평가
- 정보 산업화와 맞지 않는 측면으로 사장됨.
  - 정보의 불균질에서 이윤이 발생
  - 정보가 균질해지는 **P2P**에서는 이윤만들기가 어려움.





- 가장 많이 사용되는 모델 : 서버 & 클라이언트 모델
  - 상업적인 목적에 적당함.
  - 잘 알려진 모델이라서 구현이 수월함.
- 분산 시스템 관리 : **Agent & Manager** 모델
  - 시스템 관리의 대부분은 이 모델을 따름
- P2P
  - 일부 서비스를 위해서 사용하고 있기는 하지만
  - 주로 부하를 클라이언트에 분산하기 위해서 사용
  - 예) 클라이언트 프로그램 다운로드를 P2P로
  - 모바일기기가 인터넷에 연결되면서 중요성이 재부각



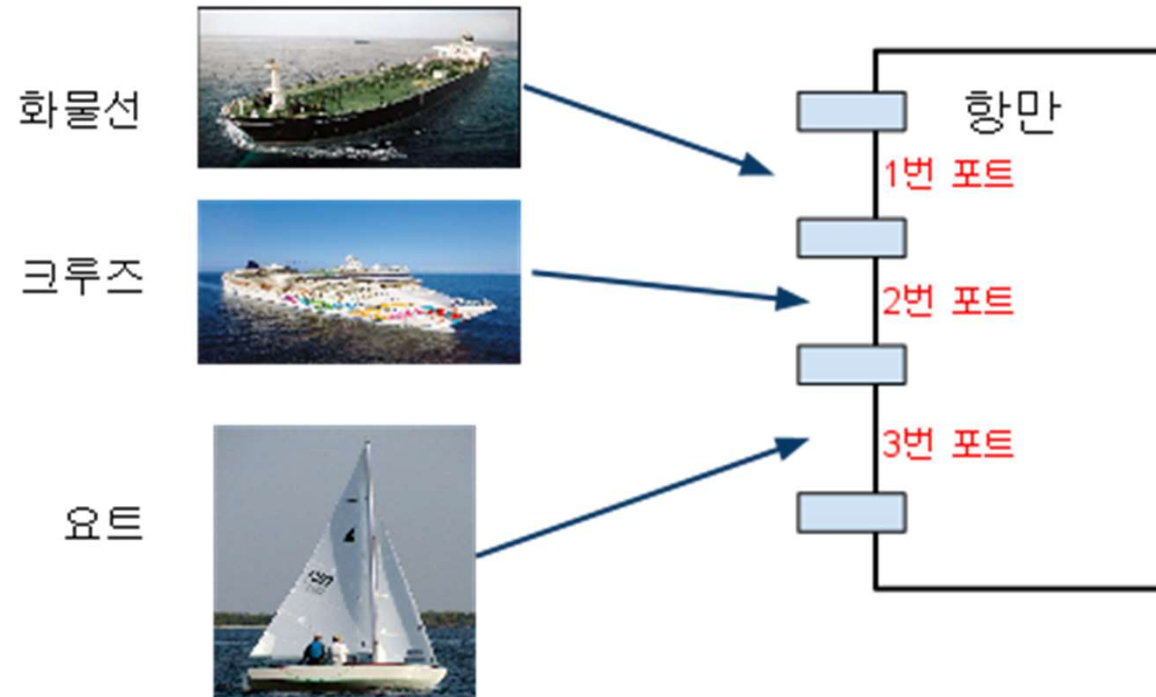


- 인터넷은 프로그램(not computer - process) 과 프로그램의 연결
- 어떻게 프로그램(**client process**)이 프로그램(**server process**)을 찾을 수 있을까?
  - 컴퓨터(server)의 위치 ? **IP 주소**
  - 컴퓨터(server)내에서 프로그램(server process)의 위치 ? **port(포트) 번호**



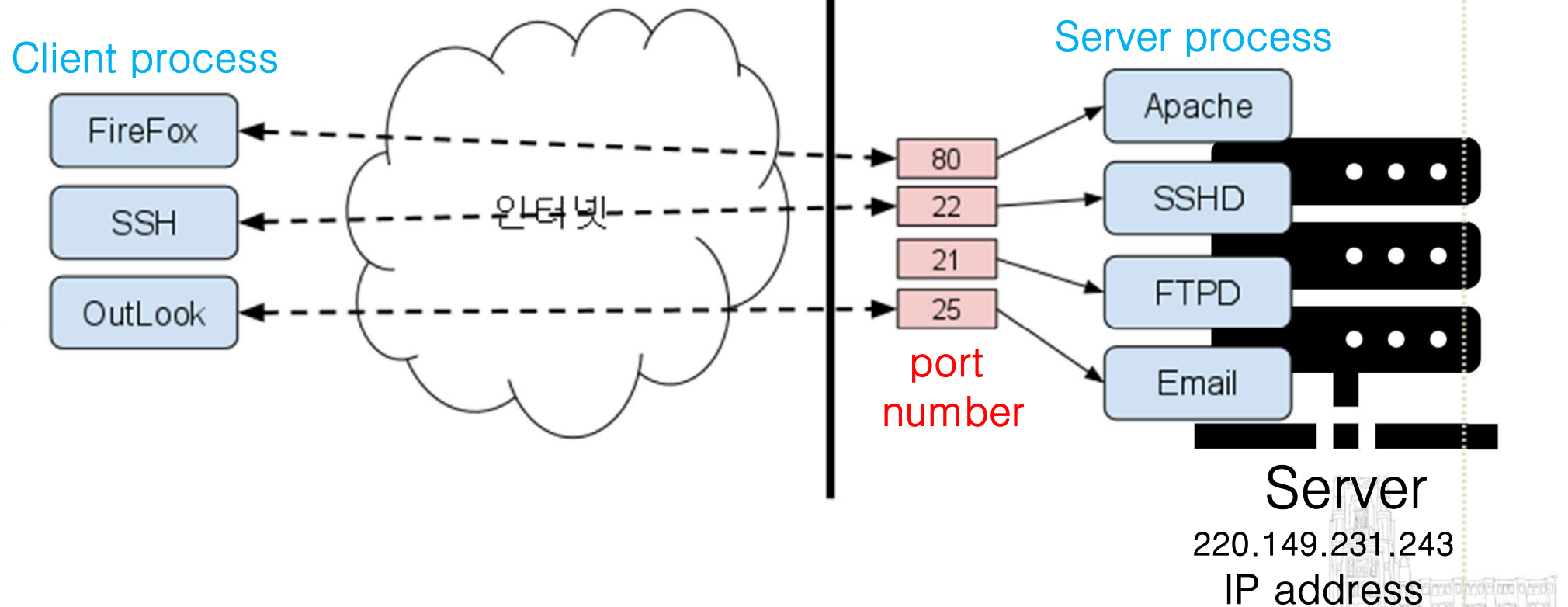
# 서비스 포트에 대한 일반 개념

포트 : 단일 플랫폼에서 성격이 다른 정보를 서비스 하기 위한 구조



# 서비스 포트

- 포트는 일련의 번호로 이루어지며, **각 서비스는 고유의 포트 번호**를 가진다
- 원격 프로그램은 **IP주소를 이용하여 컴퓨터를 찾고 포트번호로 원하는 프로그램을 찾는다.**





# 표준 서비스 포트

- 운영체제는 표준적인 웹 서비스 포트를 정의한다.
- 리눅스 : /etc/services
- 윈도우 : C:\WINDOWS\system32\drivers\etc\services

netstat	15/tcp
ftp-data	20/tcp
ftp	21/tcp
fsp	21/udp
ssh	22/tcp
telnet	23/tcp
smtp	25/tcp
time	37/tcp
time	37/udp
nameserver	42/tcp
whois	43/tcp
bootps	67/tcp
bootps	67/udp
bootpc	68/tcp
bootpc	68/udp
tftp	69/udp
gopher	70/tcp
finger	79/tcp
http	80/tcp

fspd  
# SSH Remote Login Protocol

mail  
timserver  
timserver  
name # IEN 116  
nickname  
# BOOTP server

# BOOTP client

# Internet Gopher

www # WorldWideWeb HTTP



# 포트 번호(port number)

- 포트 번호 범위 : 16-bit unsigned integer(1-65535)
  - for TCP, port number 0 is reserved and can't be used. for UDP the source port is optional and a value of zero means no port
- 1. **Well-Known Port(잘 알려진 포트) : 1-1023**
  - ftp(21번), telnet(23번), mail(25번), http(80번)
  - Controlled and assigned by IANA(Internet Assigned Numbers Authority)
- 2. **Registered Port(등록된 포트) : 1024-49151**
  - Not controlled by IANA, but IANA registers and lists the use of these ports as a convenience to the community
- 3. **Dynamic or Private(동적 혹은 사적인 포트) : 49152-65535**
  - IANA says nothing



# 포트 번호의 할당



- 서버 포트 : 명시적으로 할당
  - 포트 번호에 서버 프로그램이 bind 되므로 클라이언트도 사전에 알고 있어야 한다.
- 클라이언트 포트 : 임의 할당
  - 클라이언트 포트는 1024 보다 큰 포트 번호 중 남은 포트 번호가 임의로 할당.





# Thank You !

뇌를 자극하는 TCP/IP 소켓 프로그래밍