

LAN 기술에 대한 이해 & 소켓의 이해

인하공업전문대학 컴퓨터정보과
최효현 교수

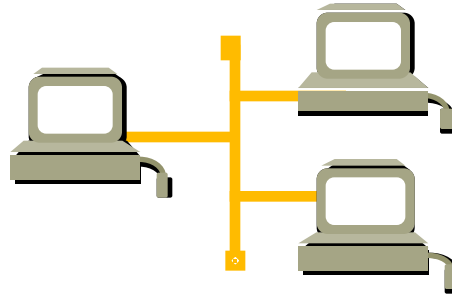
주요사항

- ❑ LAN 기술에 대한 이해
- ❑ CSMA/CD 이해
- ❑ 네트워크 백본 과 샤시형 장비

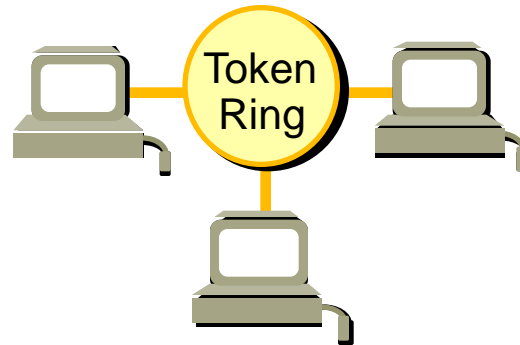
- ❑ 소켓의 개념

LAN에서 사용되는 일반적인 프로토콜

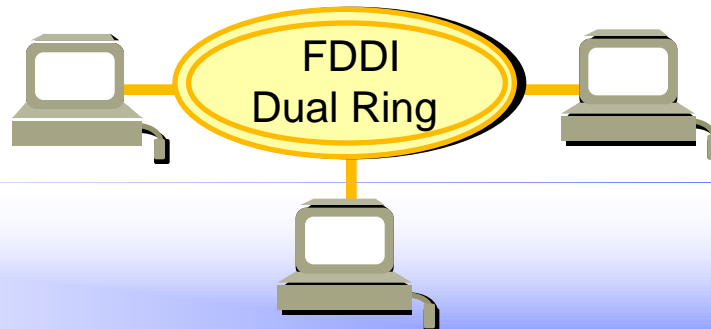
- Ethernet



- Token Ring

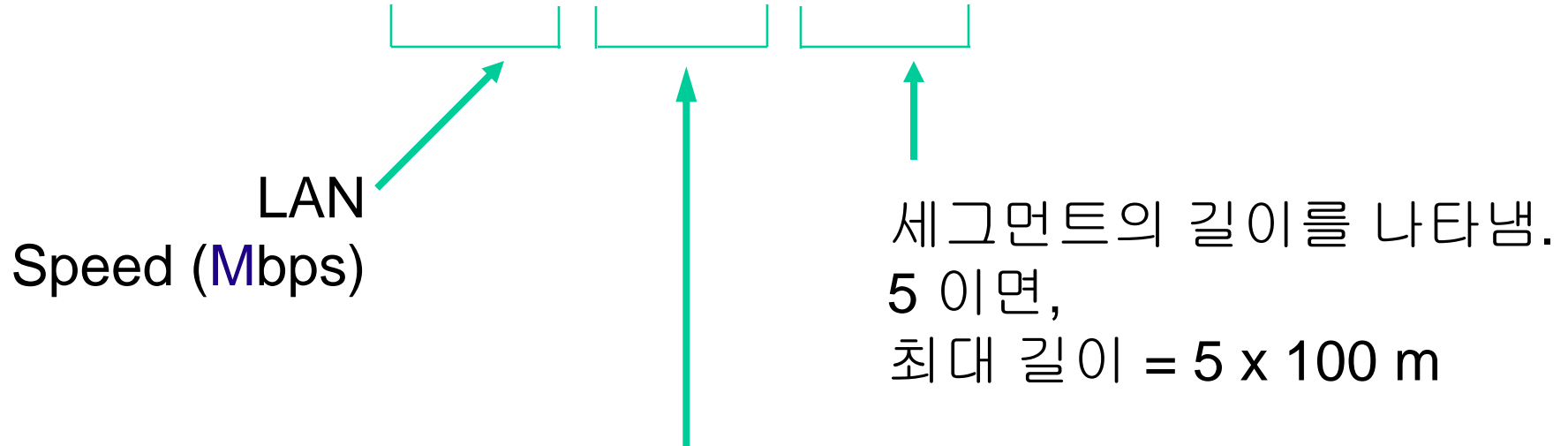


- FDDI



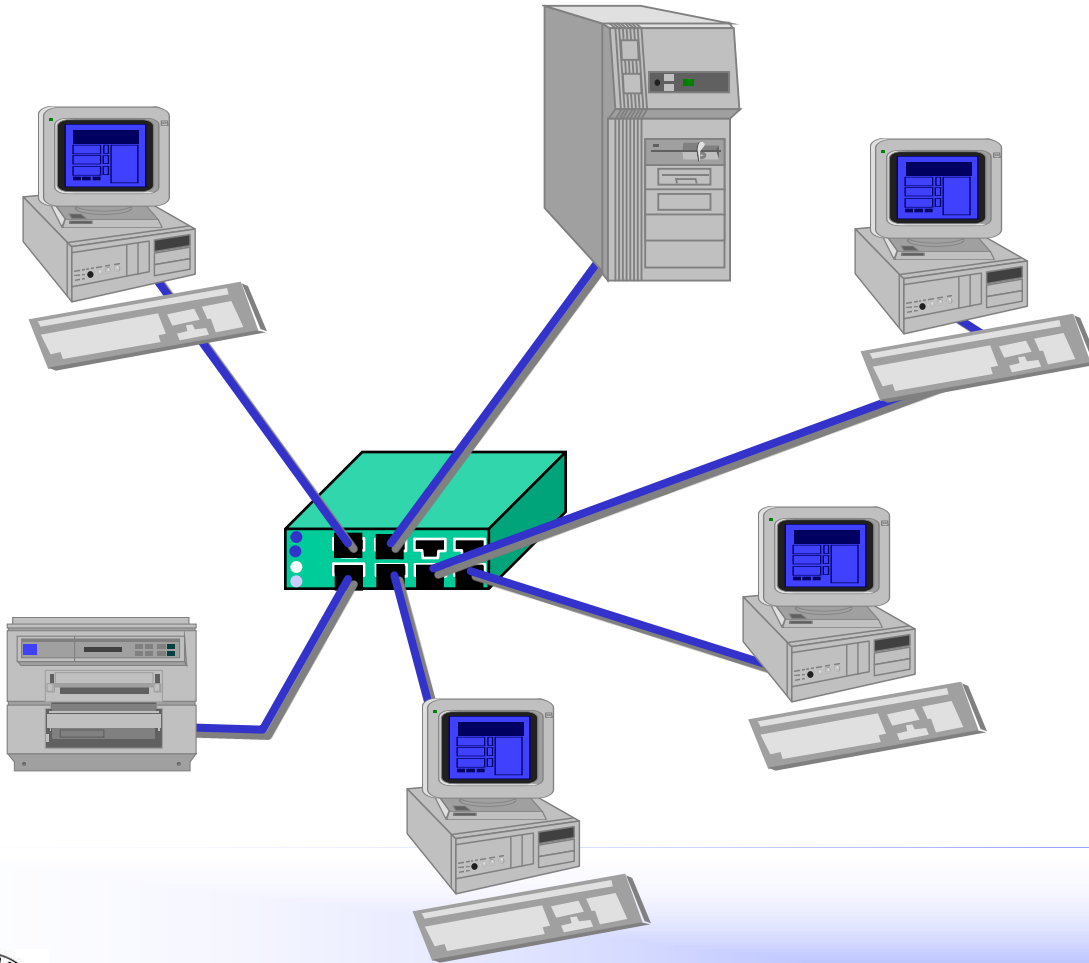
Ethernet Protocol 규격

10 base 5



“base” = baseband
“broad” = broadband

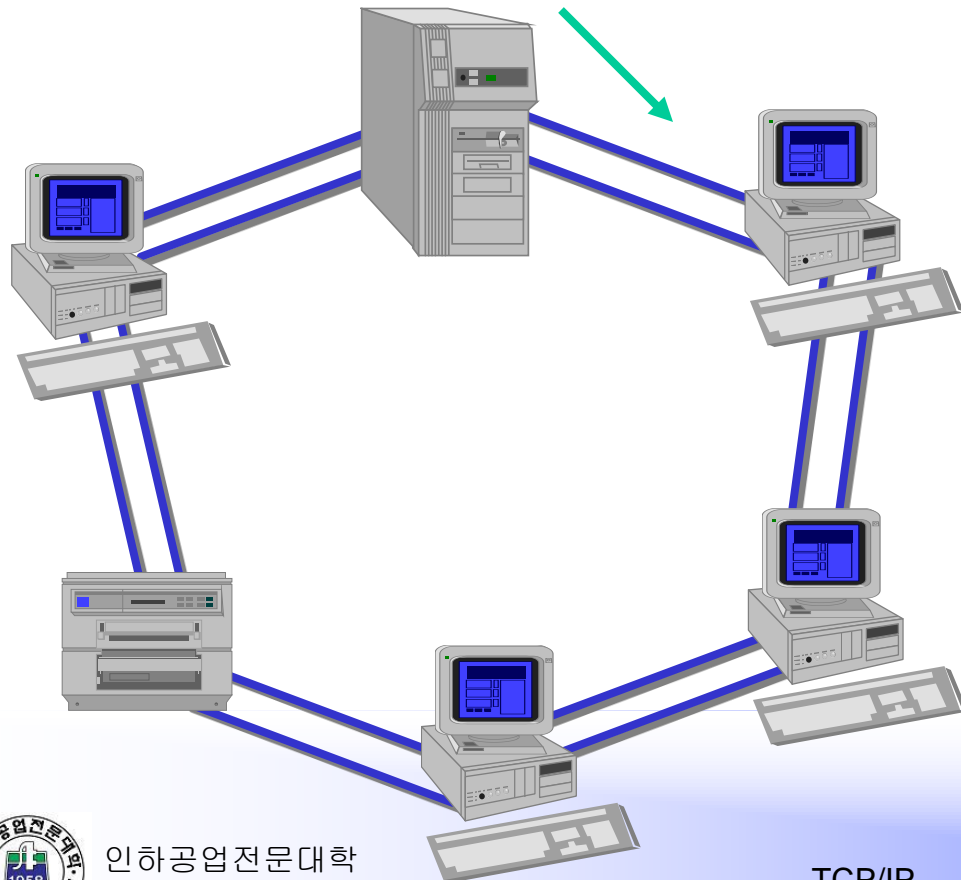
스타 토폴로지



- ✓ 중앙 : 허브, 리피터 또는 콘센트레이터
- ✓ Ethernet과 Token Ring에서 주로 사용
- ✓ 5 ~ 100 대의 장비

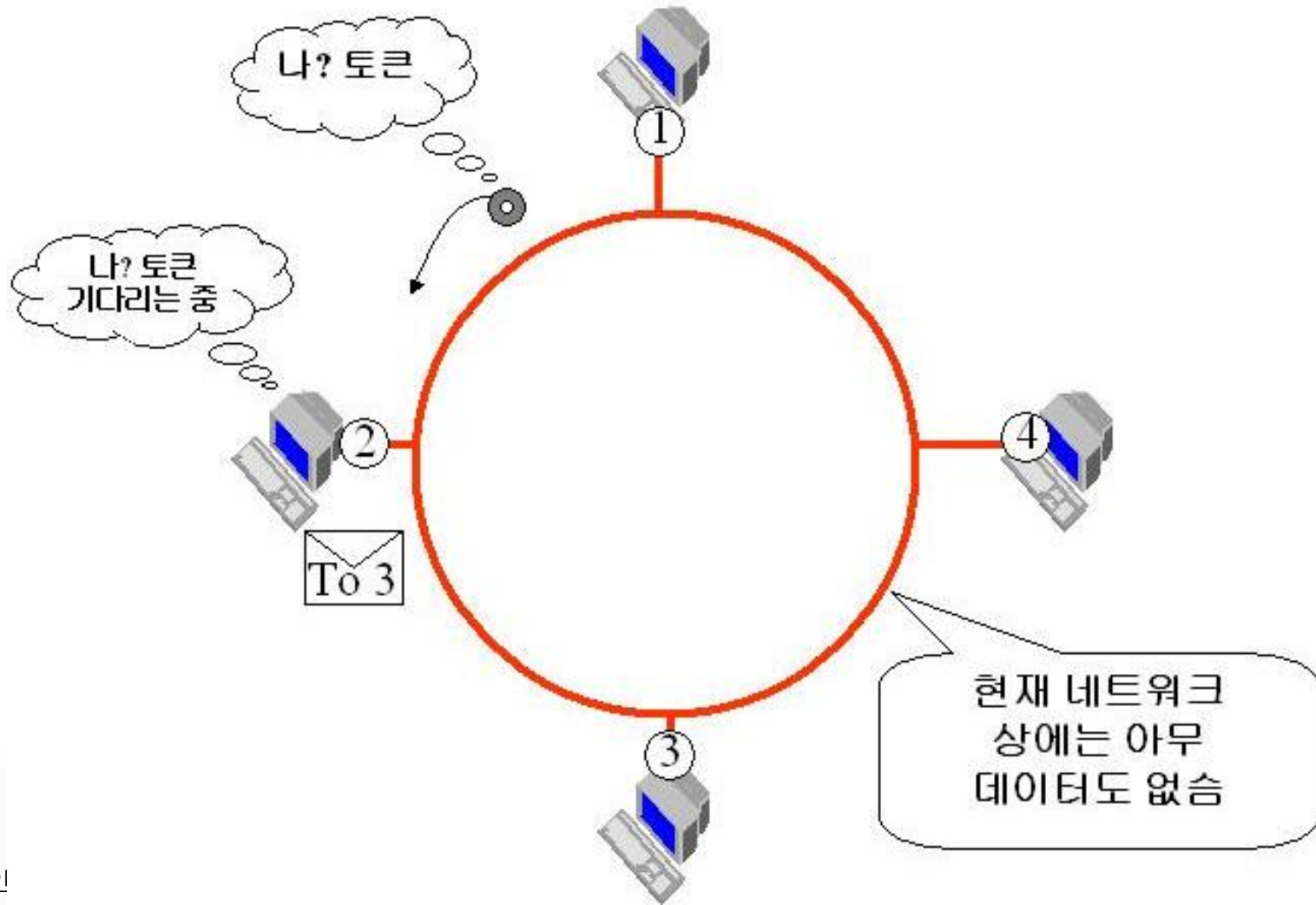
링 토폴로지

네트워크 장애를 대비한
이중링 토폴로지



- ✓ 각 부분에 리피터
- ✓ 단 방향 전송 링크
- ✓ 폐쇄된 루프 구조
- ✓ FDDI 네트워크에서
- ✓ 주로 사용됨

Token-Ring 및 FDDI



Ethernet Frame 구조

Ethernet Frame Format

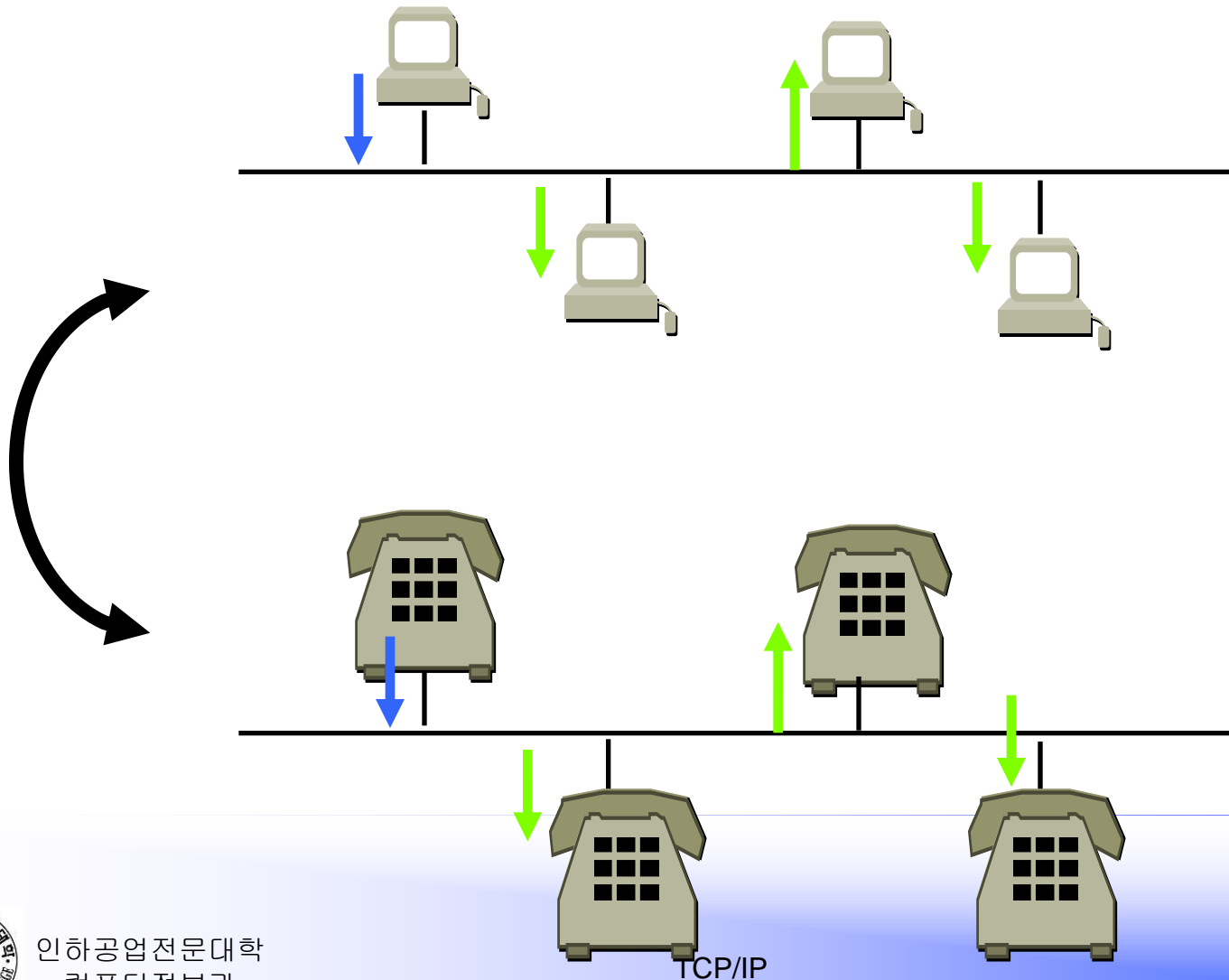
7	1	6	6	2	46 ~ 1500	7
Preamble	Frame Delimiter	Destination address	Source address	Type	data	FCS

IEEE 802.3 Frame Format

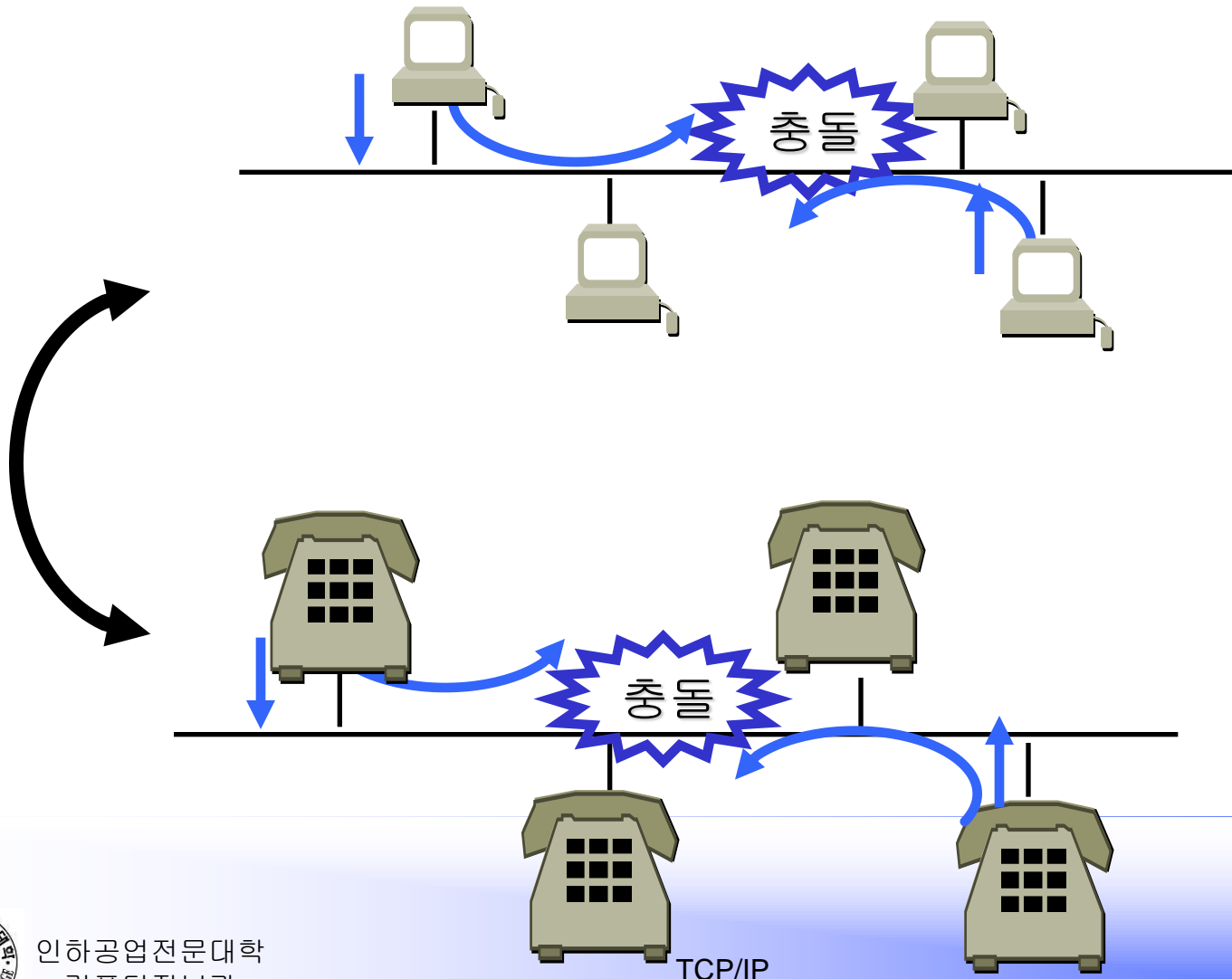
7	1	6	6	2	46 ~ 1500	7
Preamble	Frame Delimiter	Destination address	Source address	Type	data	FCS

- 프리엠블(preamble) : 비트 동기화에 사용된다. (7바이트)
- 프레임 시작(start of frame delimiter) : 프레임의 시작을 나타냄. (1바이트)
- 목적지 주소(destination address) : 수신측 주소를 나타냄. (6바이트)
- 근원지 주소(source address) : 송신측 주소를 나타냄. (6바이트)
- 종류 필드(type) : 상위 계층의 네트워크 프로토콜을 식별함. (2바이트)
- 데이터(data) : 실제 전송될 자료가 들어감. (46 ~ 1500바이트)
- 프레임 체크 시퀀스(frame check sequence) : 에러검사에 사용됨. (4바이트)

Ethernet의 CSMA/CD 동작



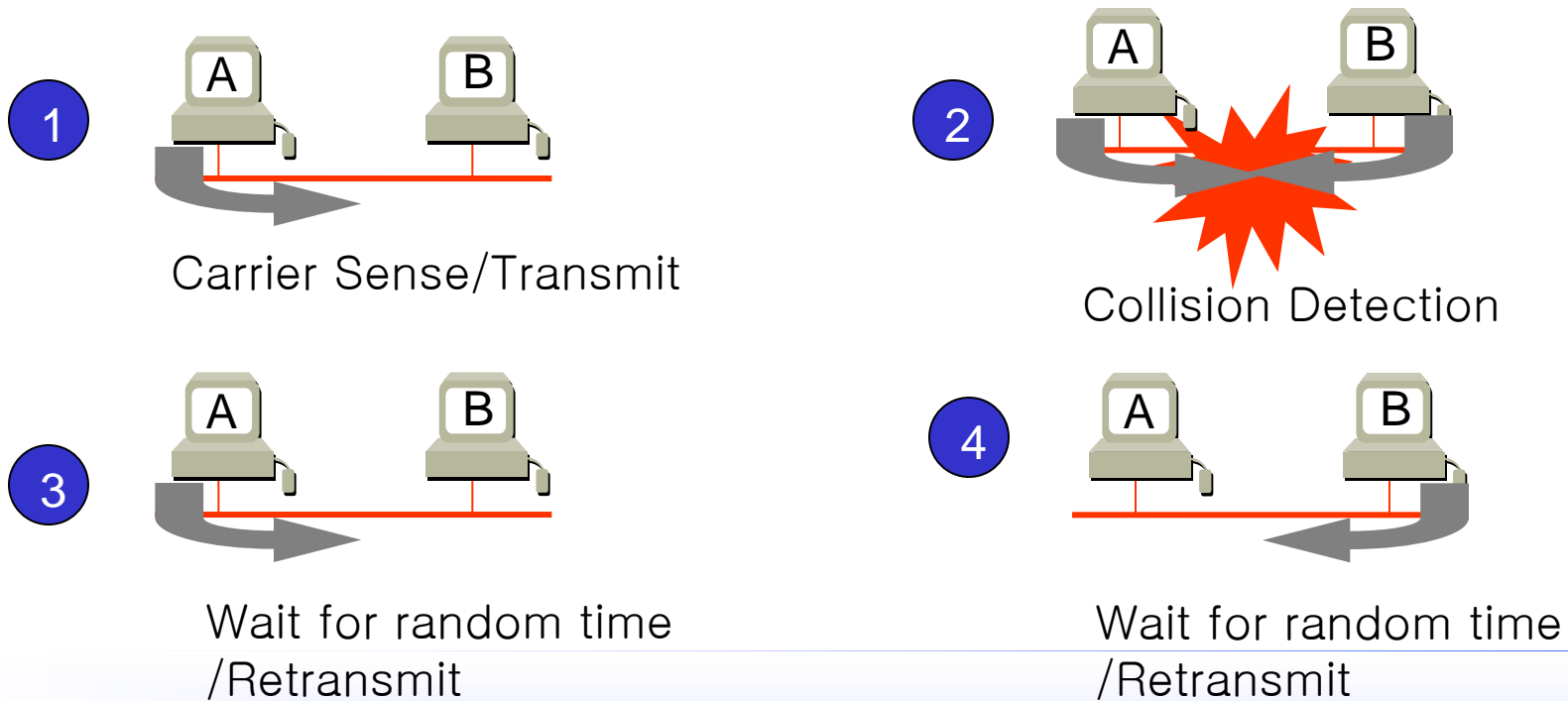
동시 사용시 충돌(Collision) 발생



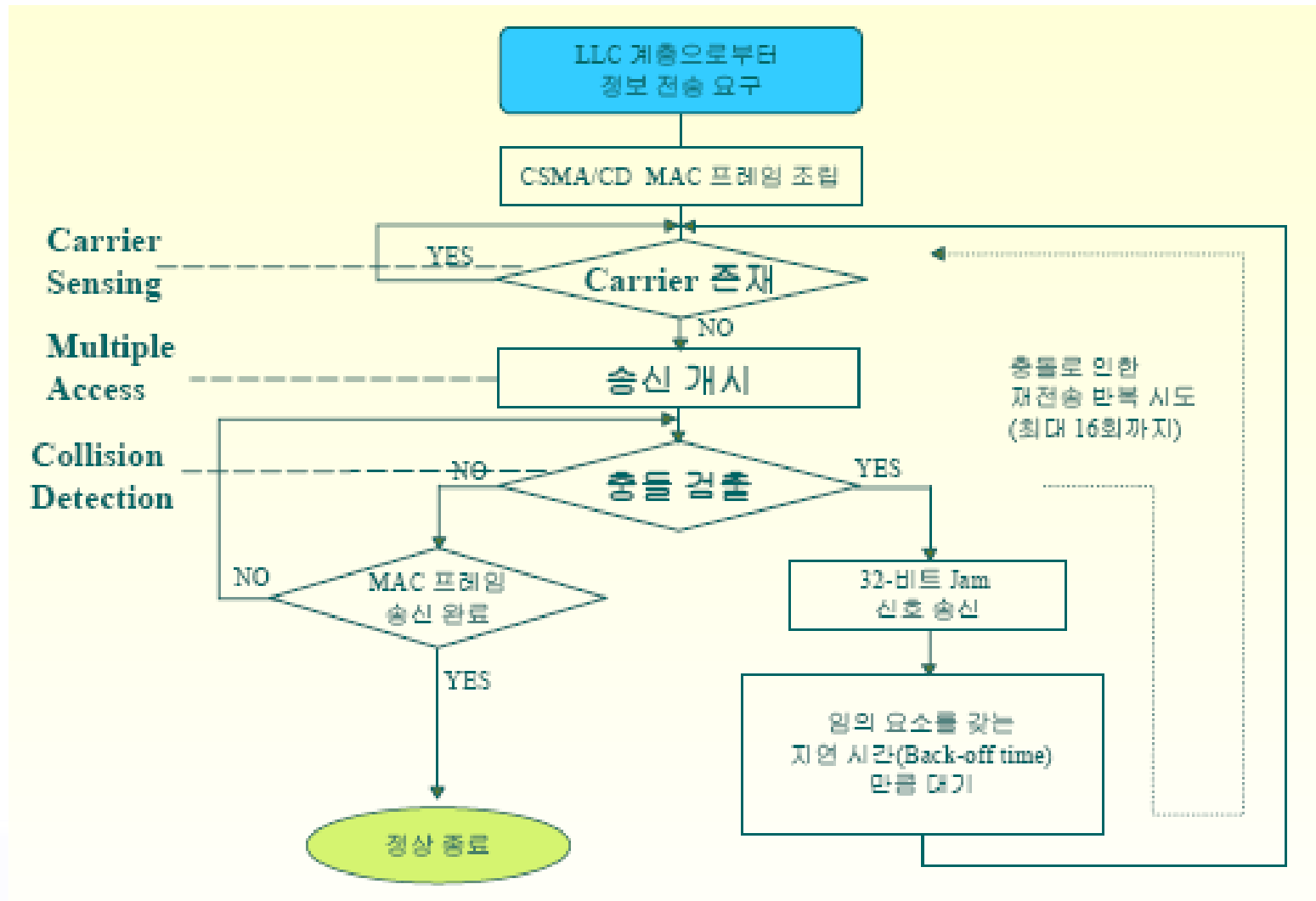
Ethernet (IEEE 802.3)의 통신방식 : CSMA/CD

❑ 매체 Access 방식

- CSMA/CD(Carrier Sense Multiple Access/Collision Detection) 방식을 사용



CSMA/CD 동작 방식

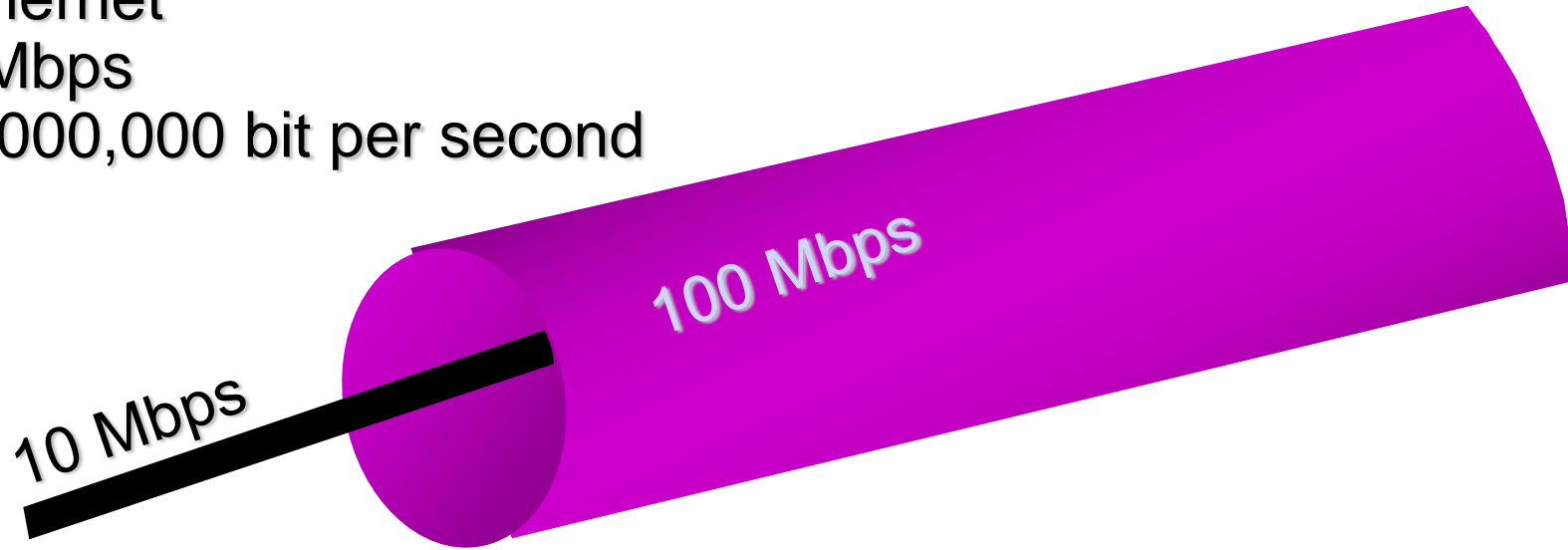


Fast Ethernet

Fast Ethernet

= 100Mbps

= 100,000,000 bit per second



- **100BaseTX** : UTP 케이블
- **100BaseFX** : Fiber 케이블

기가비트 이더넷 (Gigabit Ethernet)

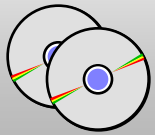
Gigabit Ethernet
= 1,000Mbps
= 1,000,000,000 bits !

10 Mbps

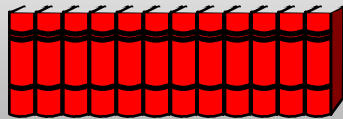
100 Mbps

1000 Mbps

Gigabit per Second:
브리태니커 백과사전 1질을
5초 이내에 전송 !



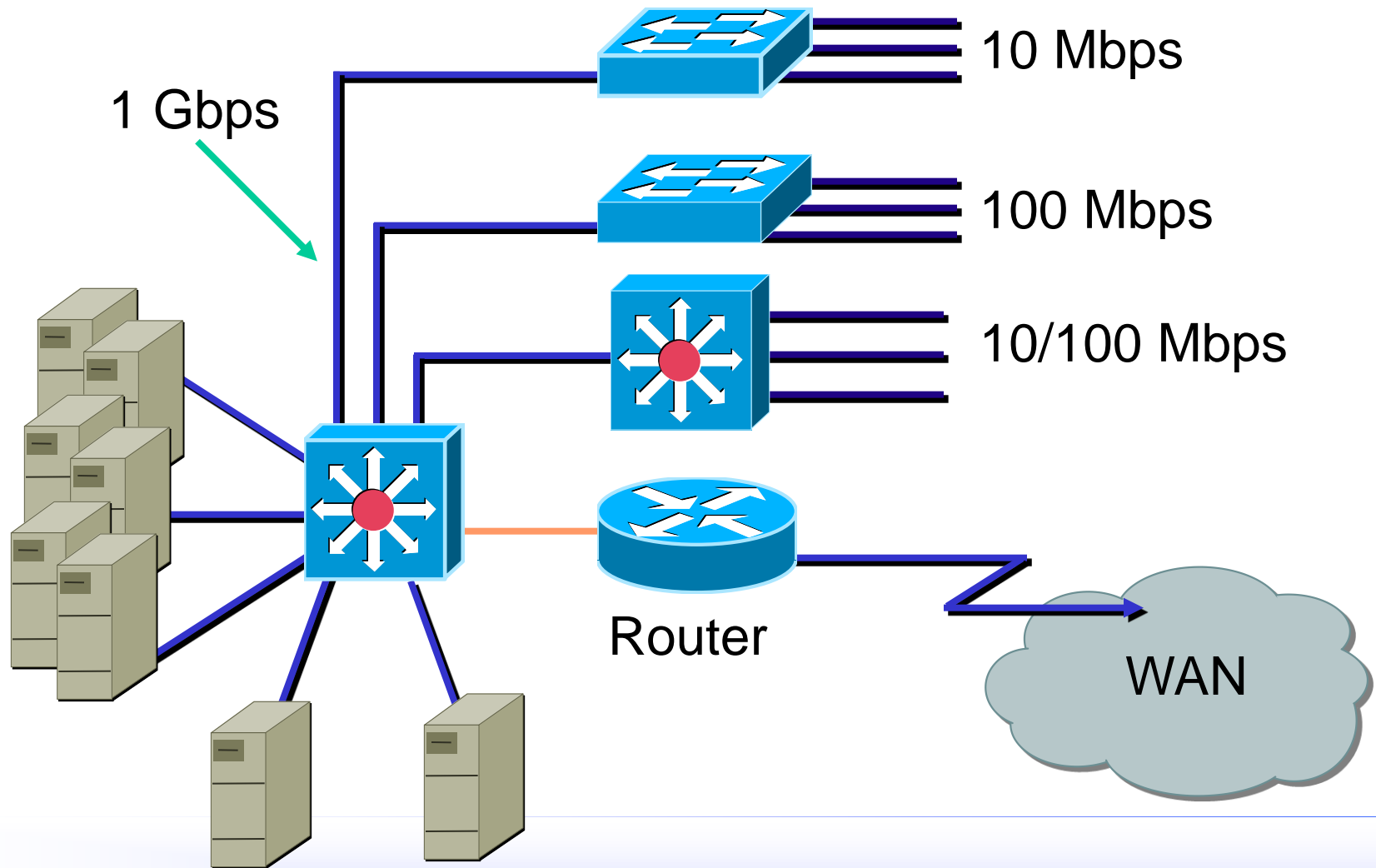
CD 2장



또는 27 권

- **1000BaseT** : UTP
- **1000BaseSX** : Fiber
- **1000BaseLX** : Fiber
- **1000BaseCX** : 동축 케이블

Gigabit Ethernet 캠퍼스 백본으로 사용

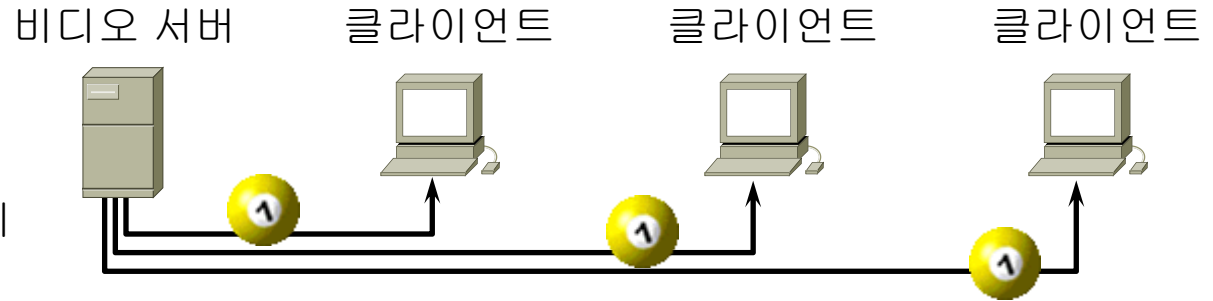


전송방식 비교

Unicast

Point-to-Point

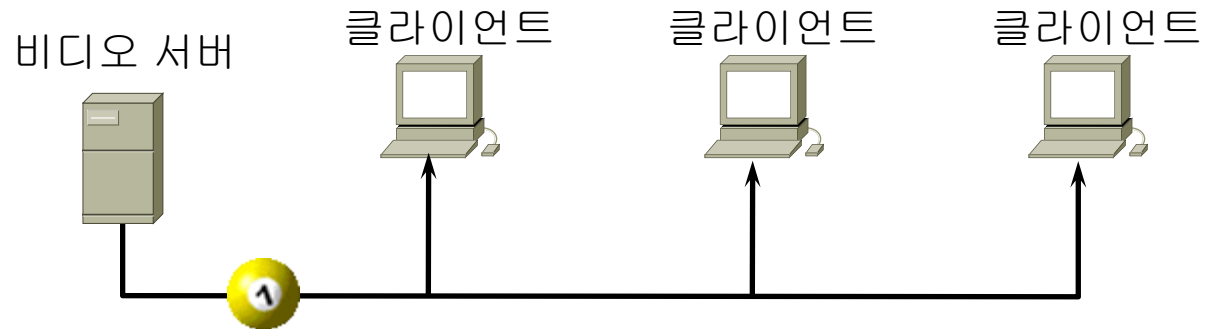
한 번에 한 개의 단말에
똑같은 패킷 3 번 보냄



Broadcast

Point-to-Multipoint

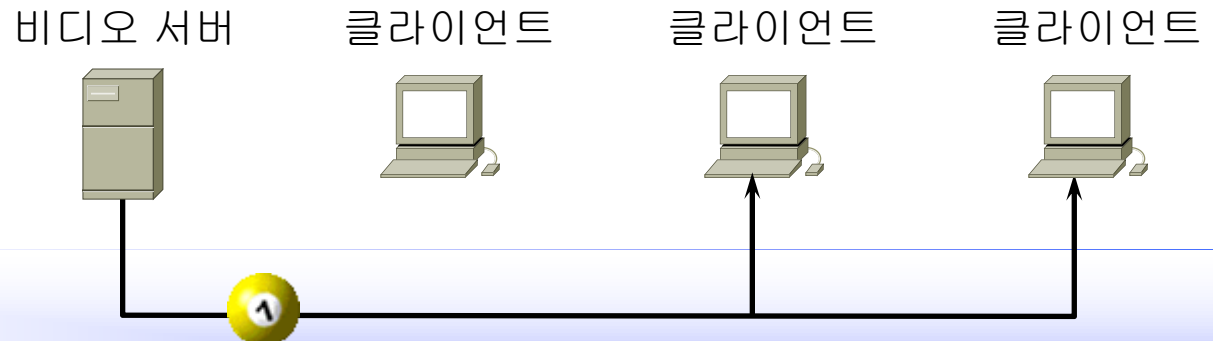
모든 단말에
패킷을 1 번 보냄



Multicast

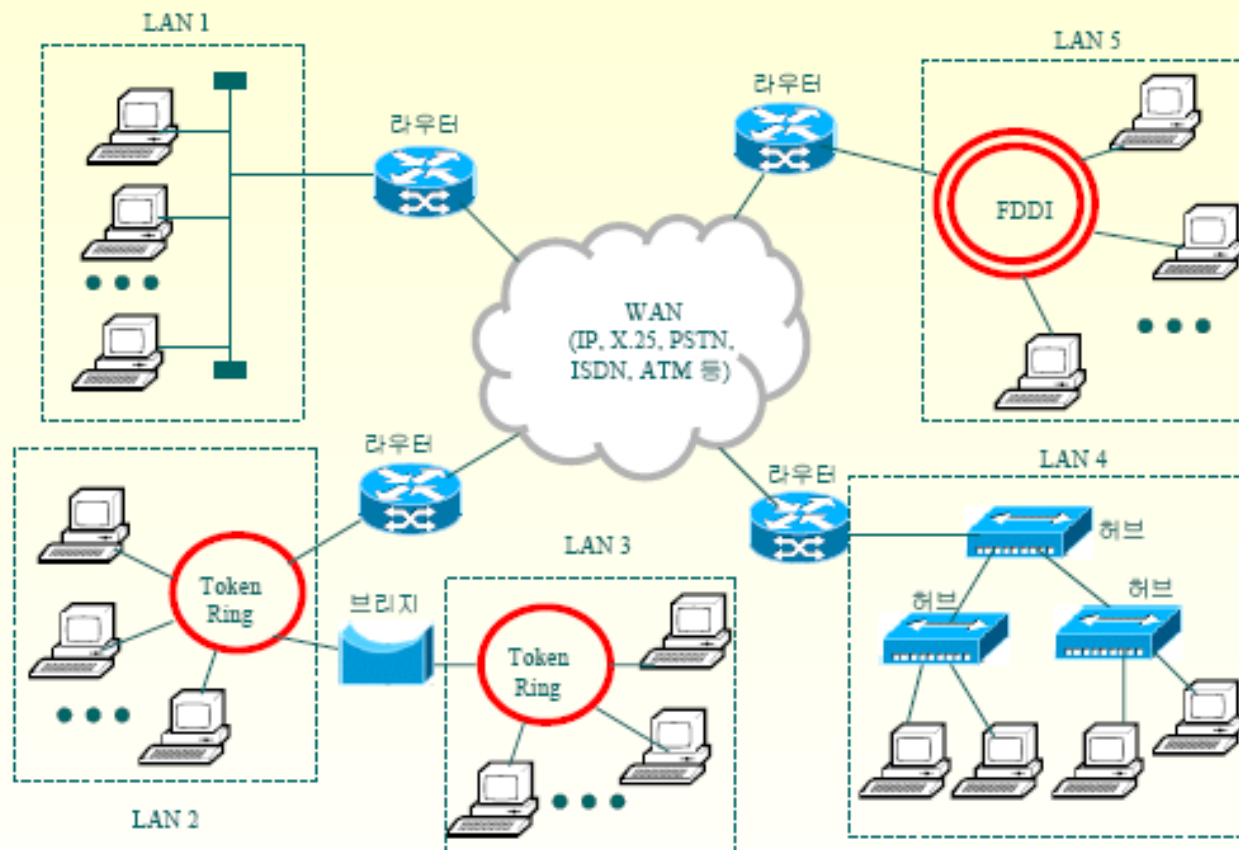
Point-to-Multipoint

그룹에 속한 단말에만
패킷을 1 번 보냄

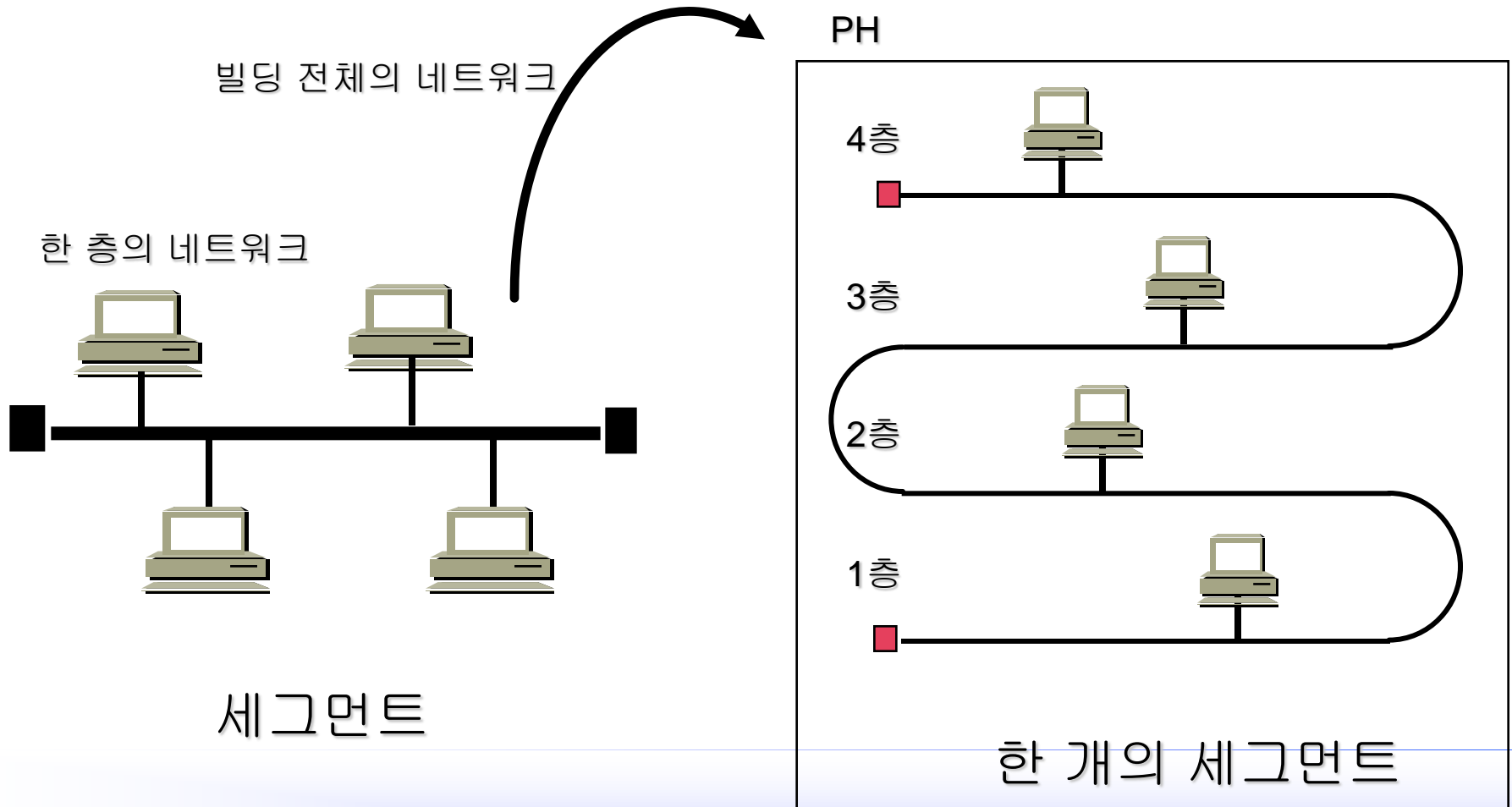


LAN과 WAN의 구성 예

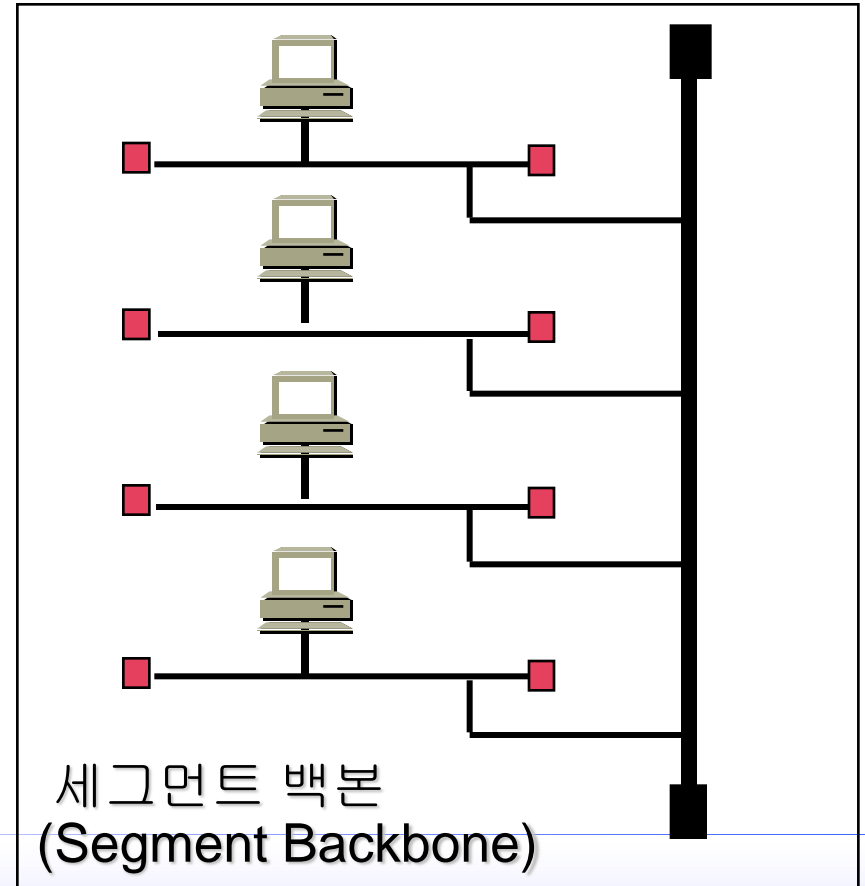
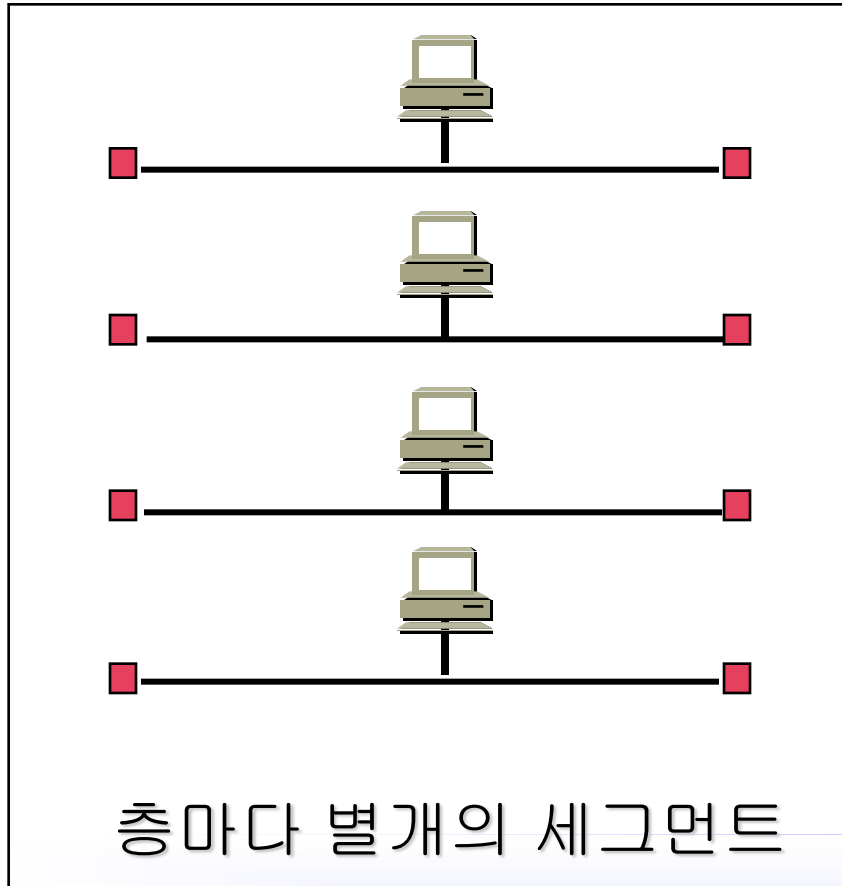
LAN과 WAN의 인터넷워킹



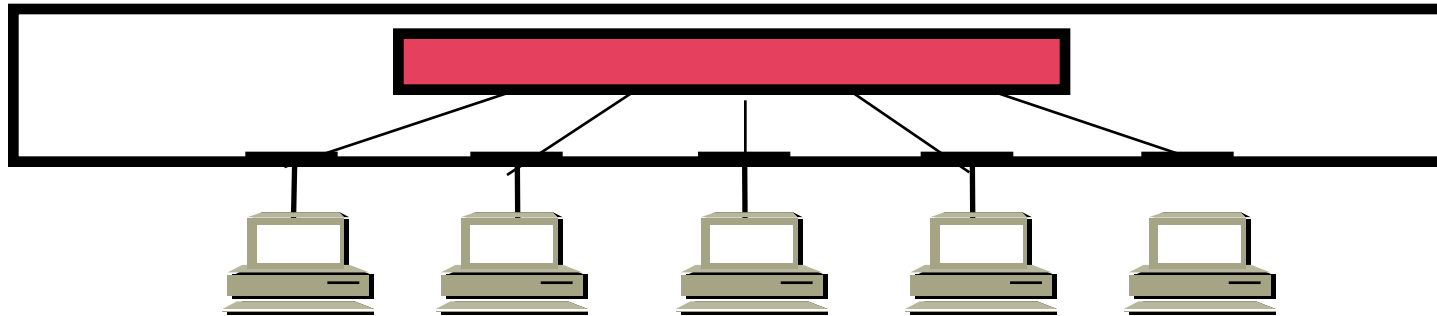
네트워크 백본



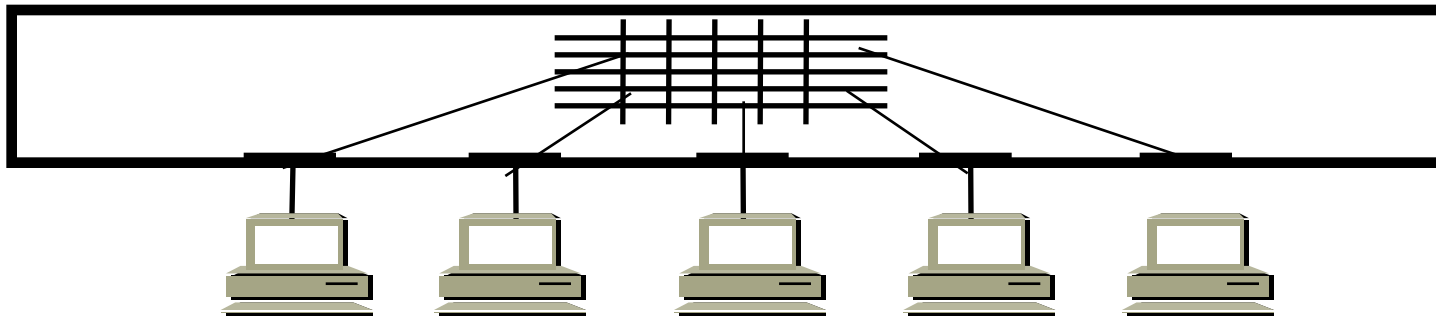
네트워크 백본



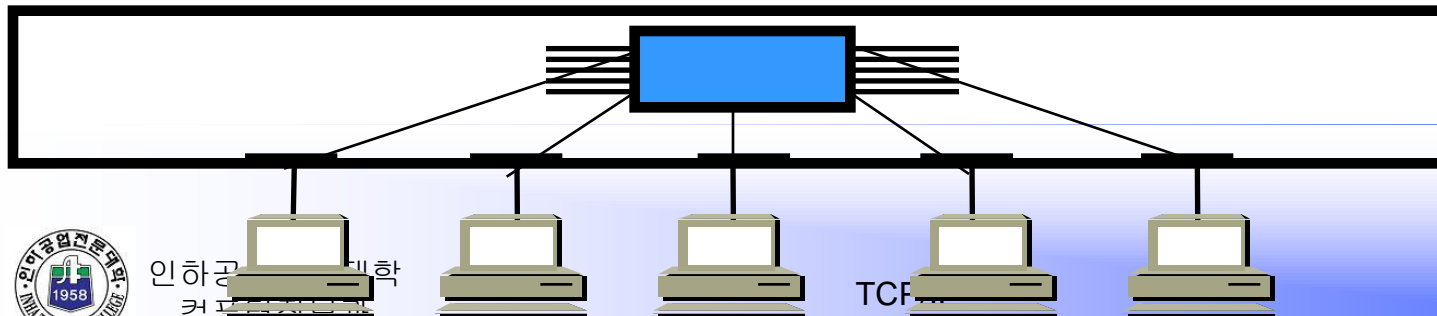
스위칭 엔진(백플레인)의 종류



BUS



Cross Bar

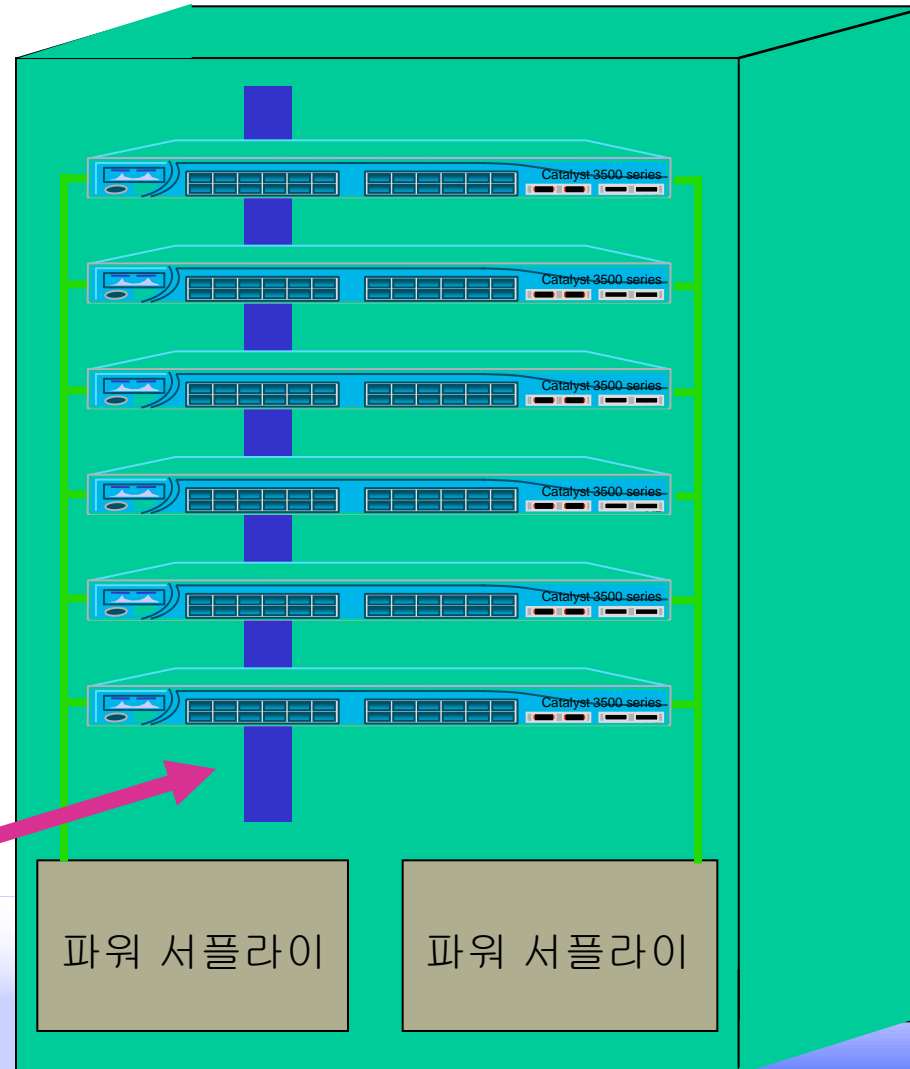


Shared
Memory

샤시형 스위칭 허브의 내부 아키텍처



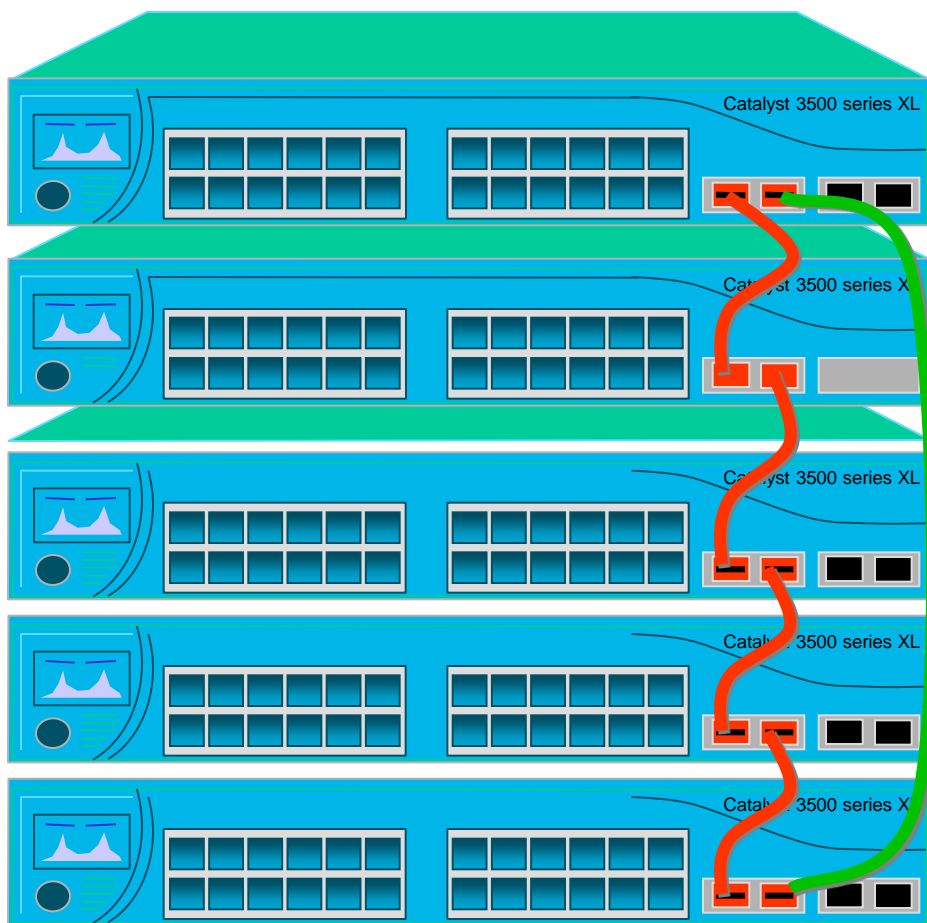
=



BackPlane 종류

- BUS
- CrossBar
- Shared Memory
- Point to Multipoint
- Point to Point

GigaStack 데이지체인 구성

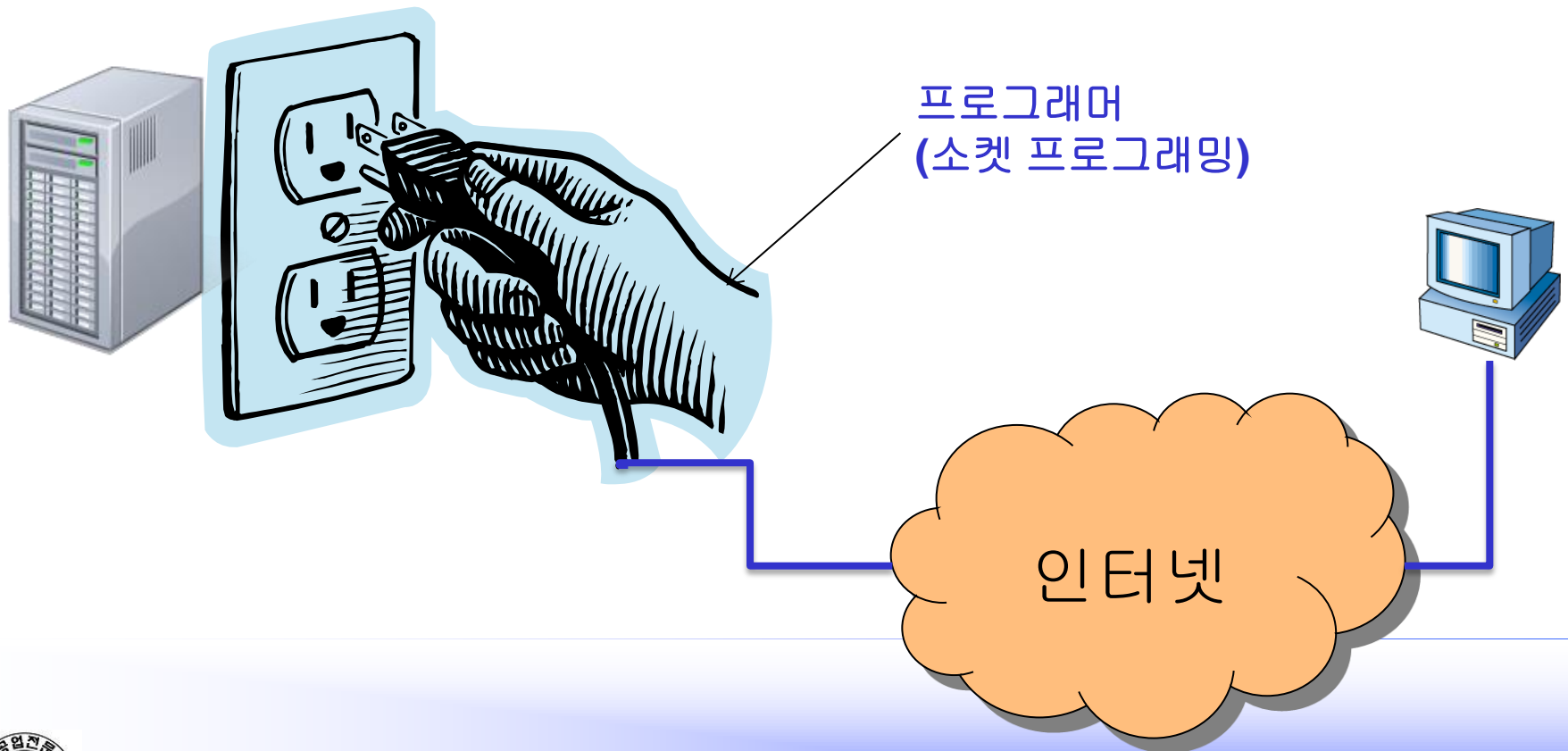


- ❑ 저렴한 Daisy chain 옵션
- ❑ 1Gbps shared bus 구현
- ❑ GBIC 포트를 이용한 redundant loopback 지원으로 높은 운용성 보장
-
- ❑ 9개까지 스택킹 가능



1. 네트워크 프로그래밍과 소켓의 이해

- 소켓 : PC (호스트) 간에 연결을 해주는 소프트웨어적인 장치



1. 네트워크 프로그래밍과 소켓의 이해

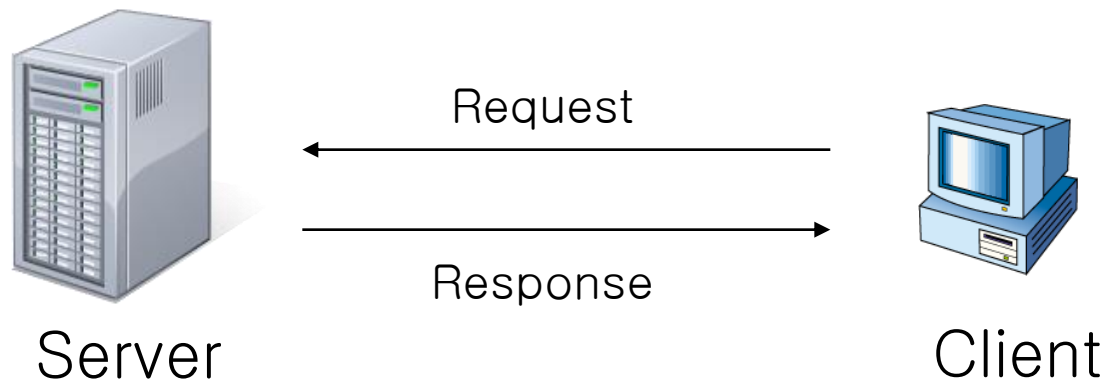
□ 네트워크 프로그래밍의 이해

- 네트워크로 연결되어 있는 두 호스트간의 데이터 송수신.
- 파일 입출력과 차이점은 데이터를 주고 받는 대상에 있다.
- 소켓(socket)이라는 장치를 사용하여 프로그래밍 한다.
 - 소켓 : 소프트웨어 차원에서 두 호스트를 연결해 주는 장치
- 소켓 프로그래밍 == 네트워크 프로그래밍

1. 네트워크 프로그래밍과 소켓의 이해

❑ Client/Server 모델

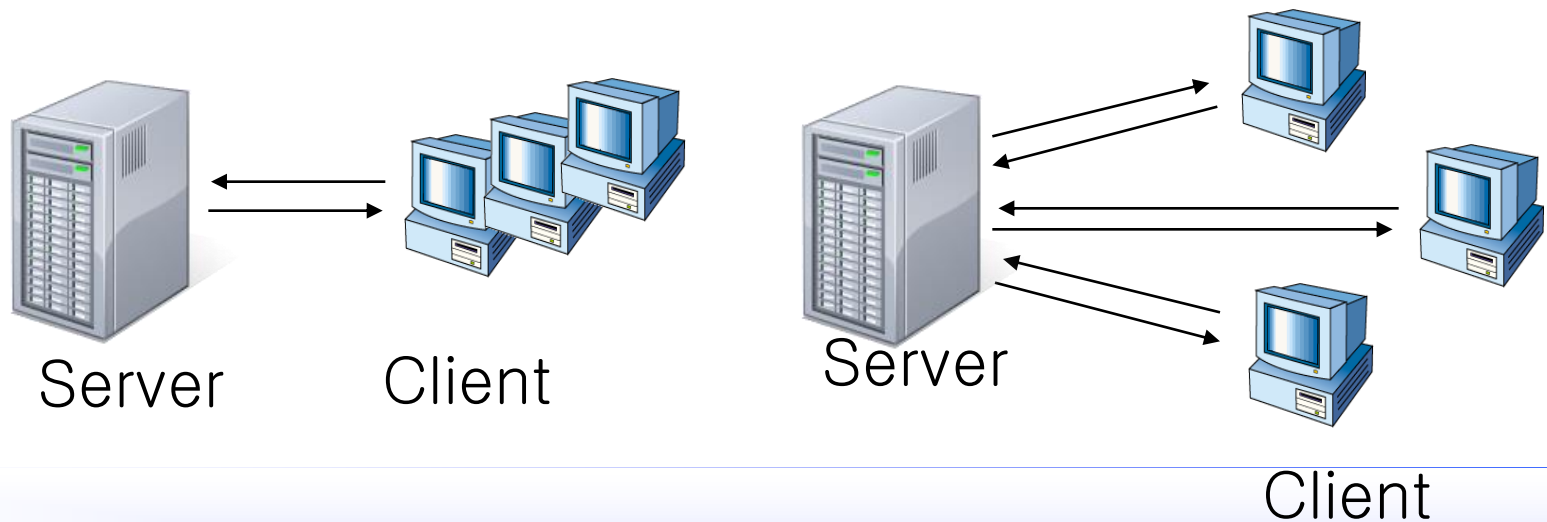
- Server와 Client는 프로그램이다. Machine이 아니다!
- Server는 Client의 연결요청을 기다린다.
- Client는 Server에 요청을 하고 응답을 기다리는 호스트를 의미한다.



1. 네트워크 프로그래밍과 소켓의 이해

□ Server의 종류

- 두 가지 종류의 서버.
 - Iterative Server : 한 순간에 하나의 클라이언트에게 응답한다.
 - Concurrent Server : 동시에 여러 클라이언트에게 응답한다.



1. 네트워크 프로그래밍과 소켓의 이해

□ 서버 소켓 구현의 이해

- 순서 1. 전화를 걸고 싶는데 무엇이 필요합니까?
 - 소켓의 생성을 요구한다.

```
#include <sys/types.h>
#include <sys/socket.h>

int socket (int domain, int type, int protocol)
```

1. 네트워크 프로그래밍과 소켓의 이해

□ 서버 소켓 구현의 이해

- 순서 2. 당신의 전화번호는 어떻게 되나요?
 - 주소의 할당을 요구한다(IP 주소, Port 주소).

```
#include <sys/socket.h>
```

```
int bind(int sockfd, struct sockaddr *myaddr, int addrlen)
```

1. 네트워크 프로그래밍과 소켓의 이해

□ 서버 소켓 구현의 이해

- 순서 3. 전화기를 연결 하시겠습니까?
 - 연결 요청 대기 상태로의 진입.

```
#include <sys/socket.h>  
  
int listen(int sockfd, int backlog)
```

1. 네트워크 프로그래밍과 소켓의 이해

□ 서버 소켓 구현의 이해

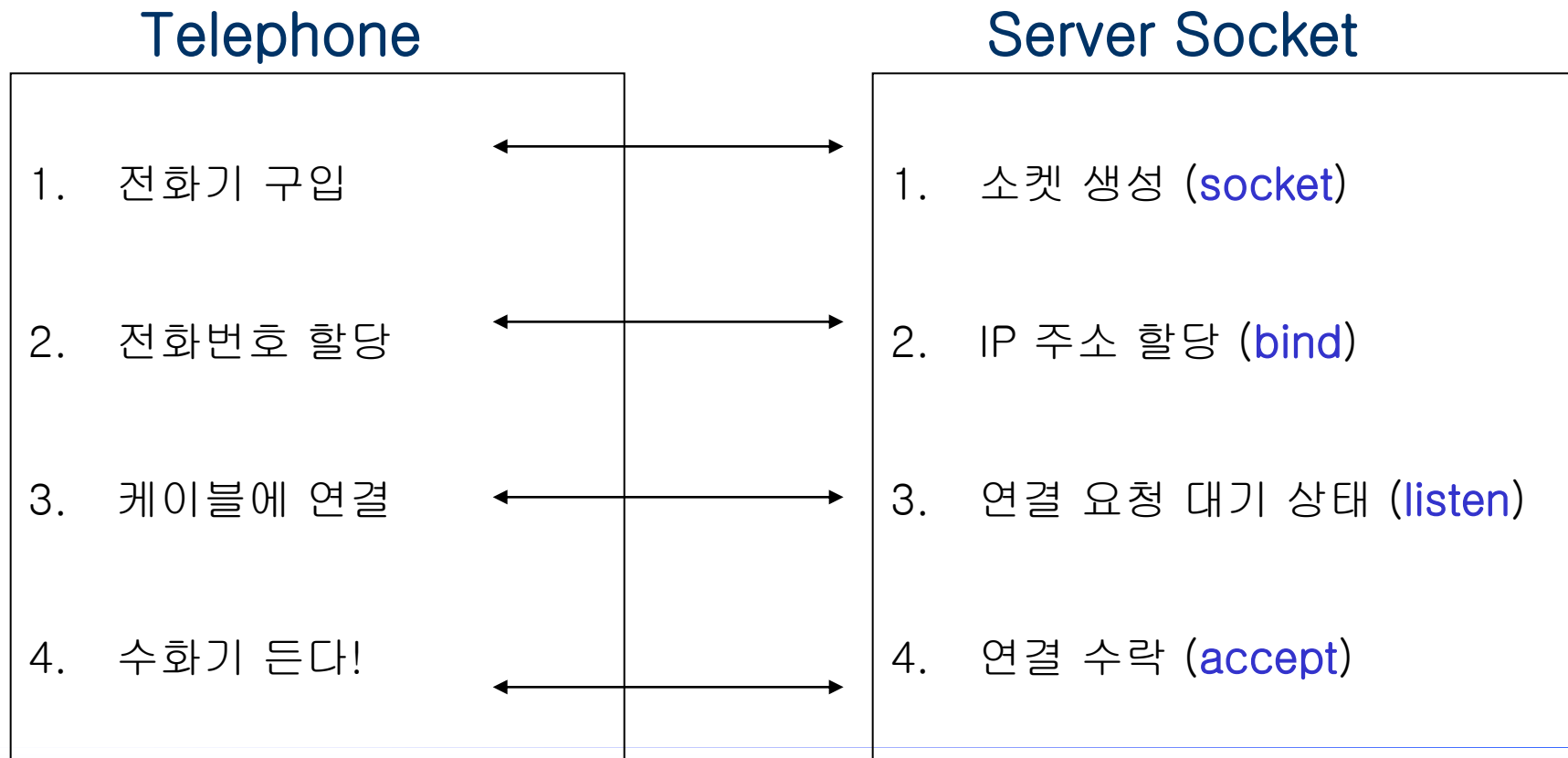
- 순서 4. 전화벨이 울립니다. 어서 전화 받으세요!!!
– 연결 요청 수락.

```
#include <sys/socket.h>
```

```
int accept(int sockfd, struct sockaddr *addr, int *addrlen)
```

1. 네트워크 프로그래밍과 소켓의 이해

□ 서버 소켓 구현의 이해



1. 네트워크 프로그래밍과 소켓의 이해

□ 클라이언트 소켓 구현의 이해

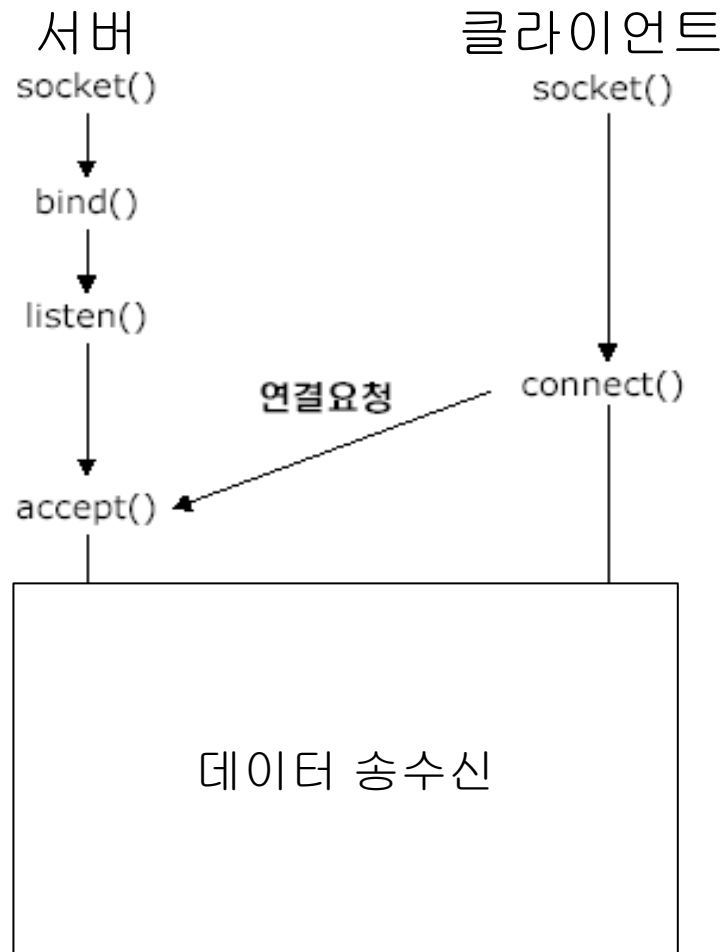
- **순서 1.** 전화를 걸고 싶는데 무엇이 필요합니까?
 - 소켓의 생성 : `socket()`;
- **순서 2.** 전화를 걸어요
 - 연결 요청

```
#include <sys/socket.h>
```

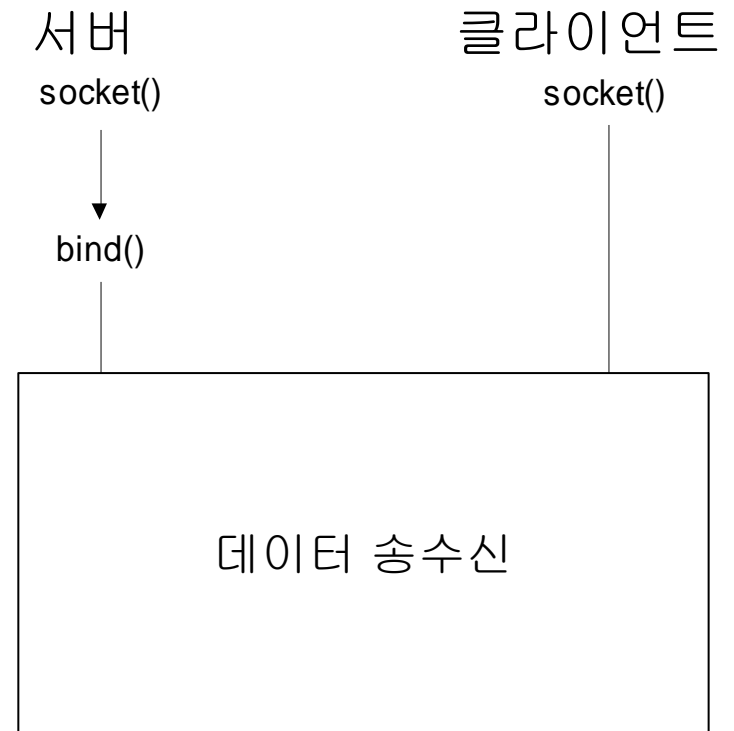
```
int connect(int sockfd, struct sockaddr *serv_addr, int addrlen)
```

TCP와 UDP 기반 서버/클라이언트

TCP기반 서버/클라이언트



UDP기반 서버/클라이언트



Q&A

