서브넷 계산 실습 &
소켓 생성 실습

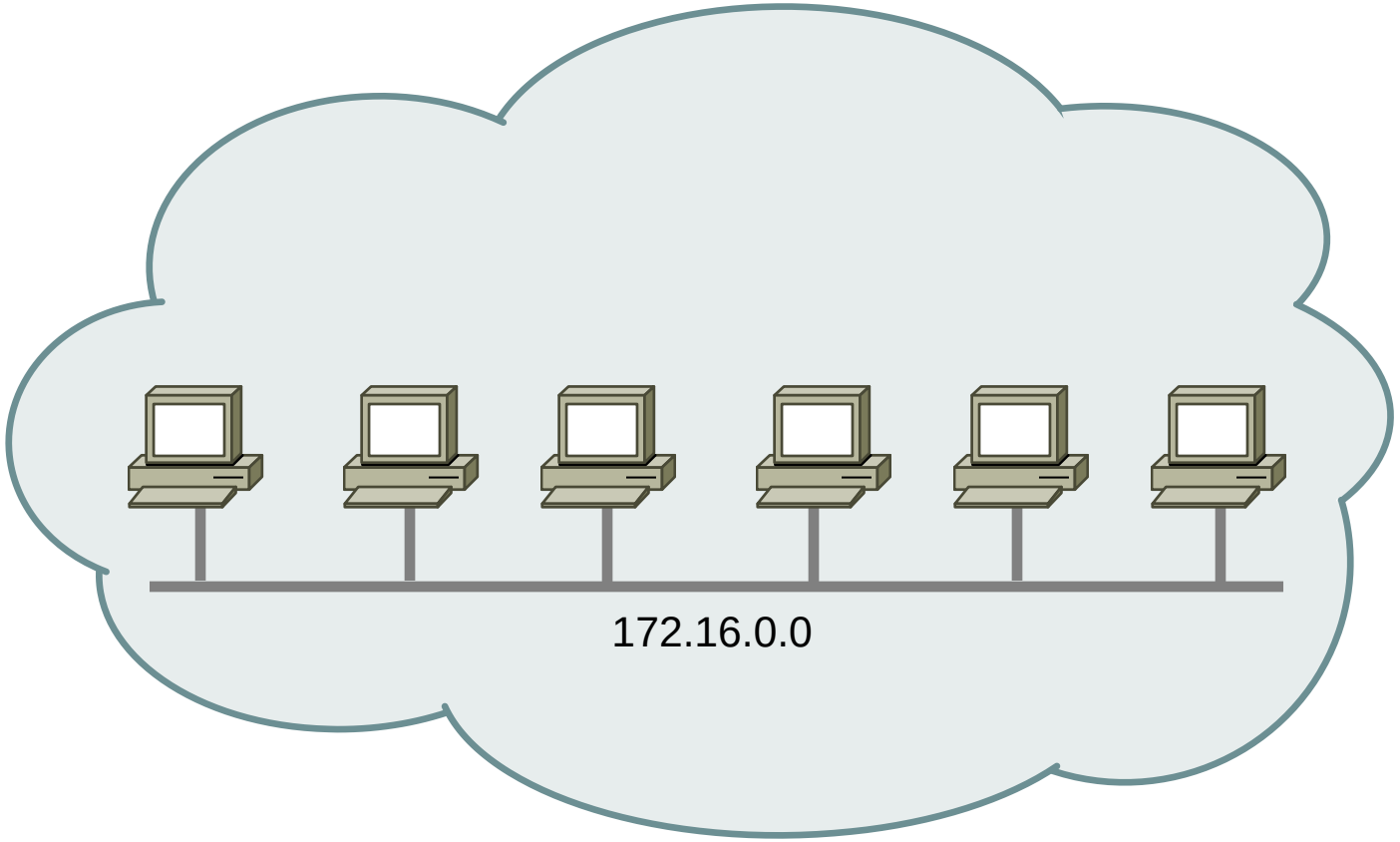
인하공업전문대학 컴퓨터정보과
최효현 교수

주요사항

- Subnet 개념
- Subnet 주소 계산 방법

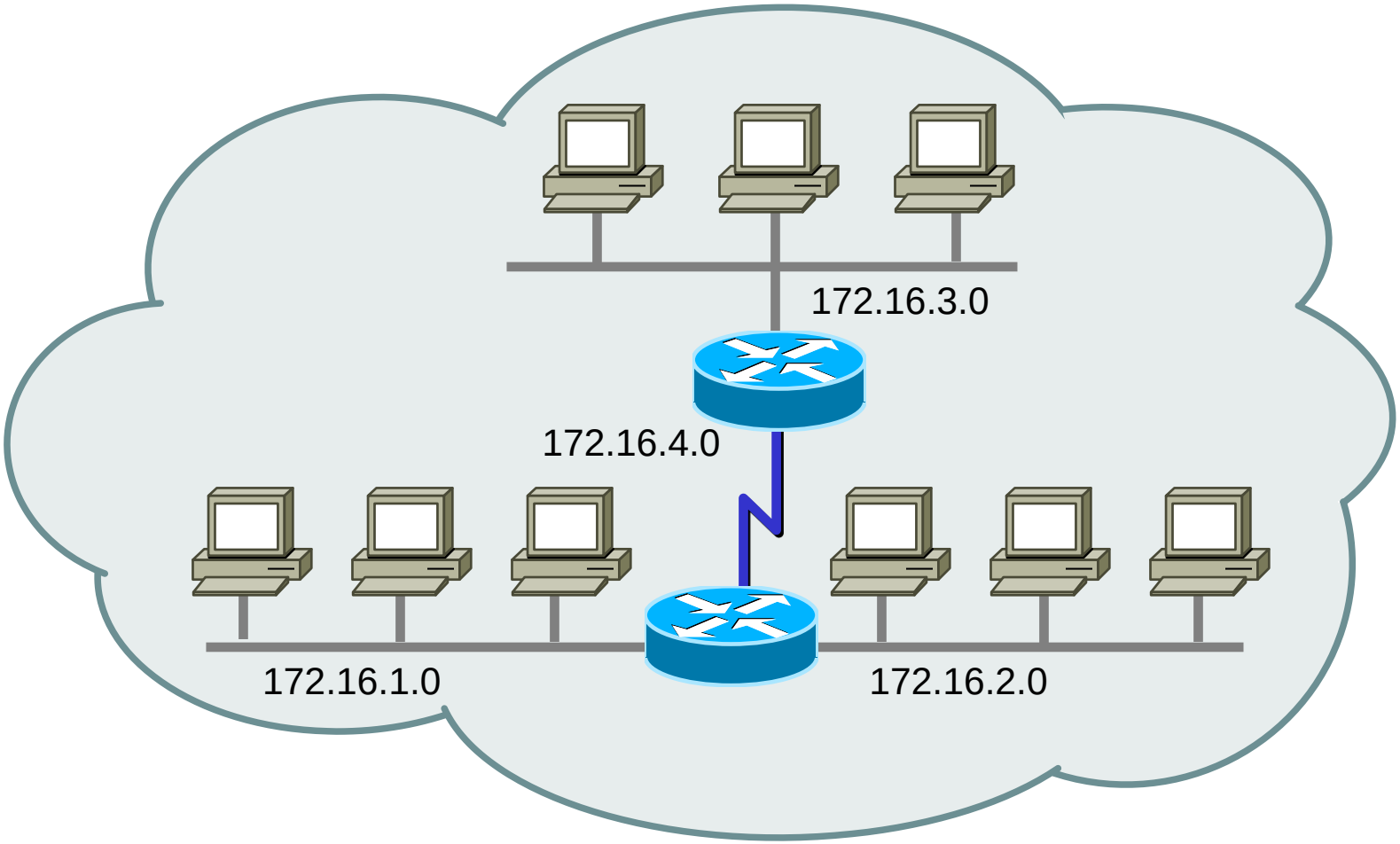
- 소켓의 생성
- 프로토콜 체계
- 소켓의 타입
- 프로토콜의 선택

서브넷이 없는 주소 지정

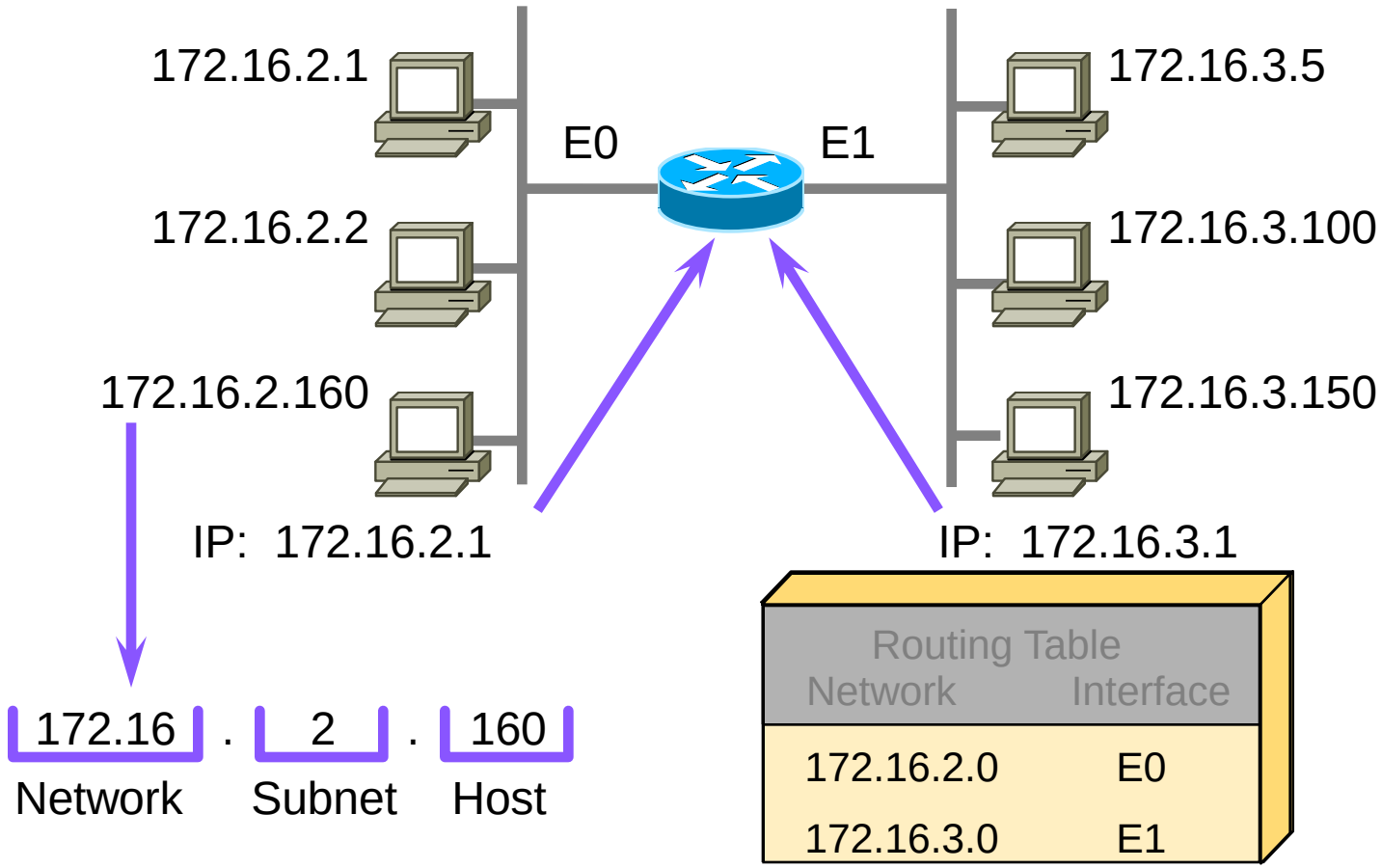


- Network 172.16.0.0

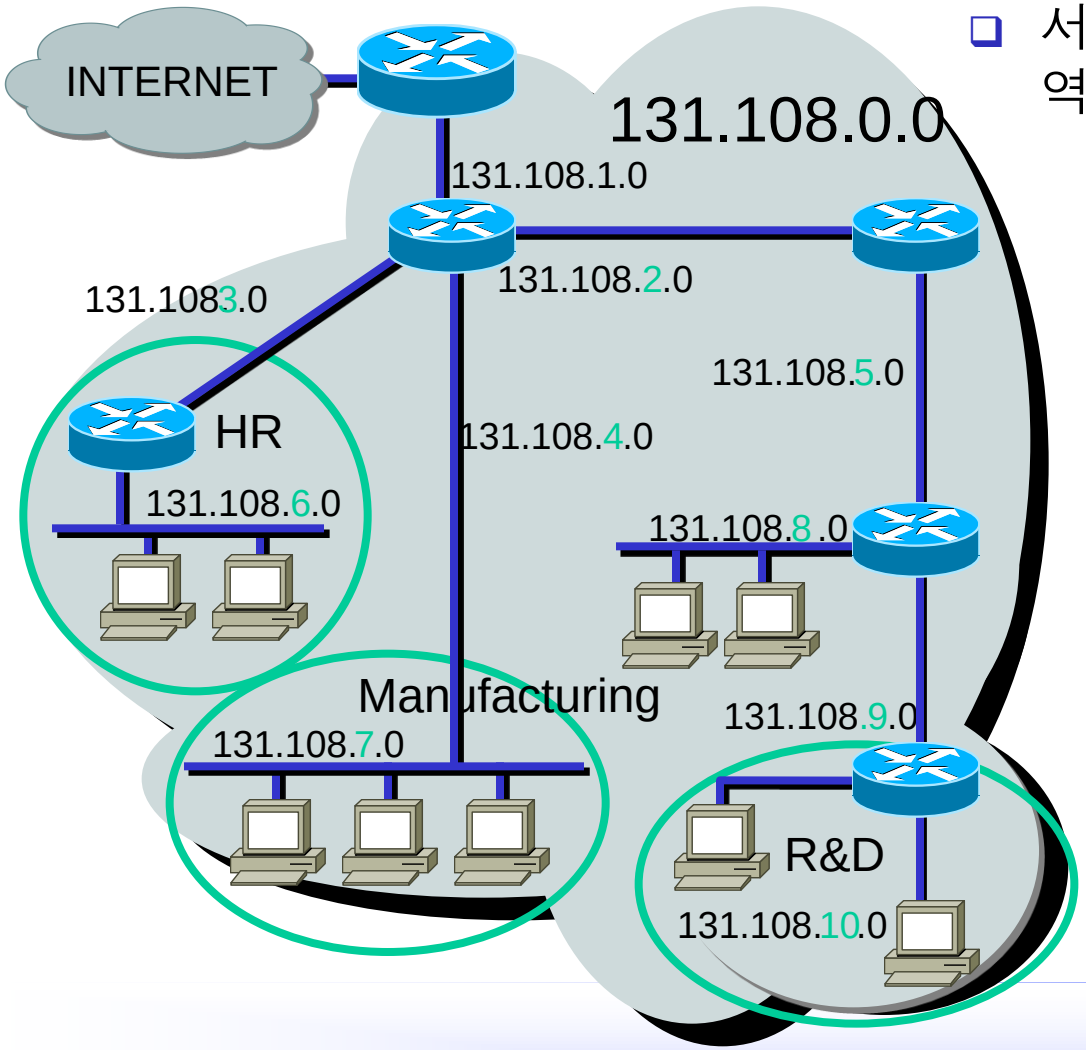
서브넷을 갖는 주소 지정



서브넷 어드레싱

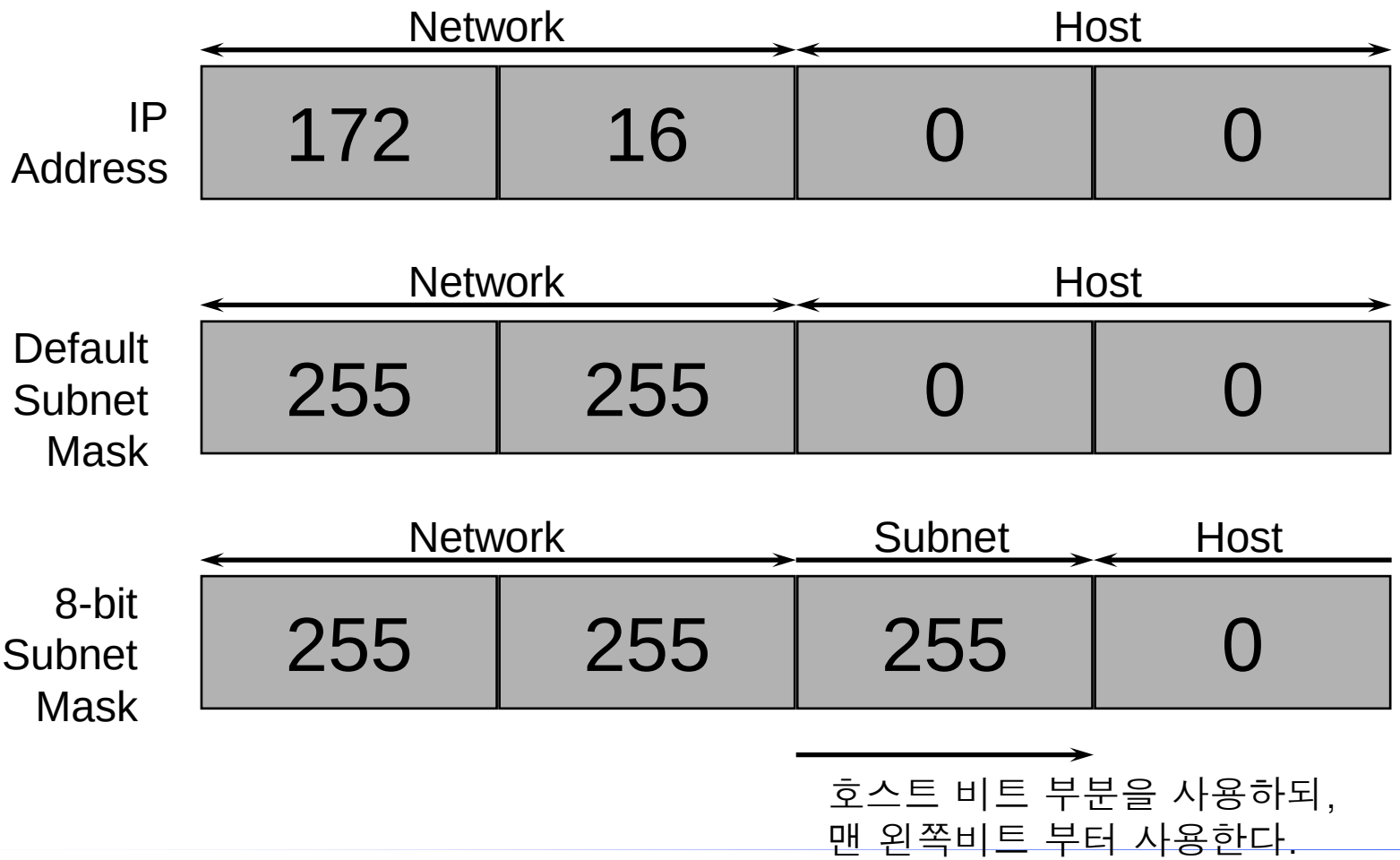

172.16.2.1
172.16.2.2
172.16.2.160
E0
E1
172.16.3.5
172.16.3.100
172.16.3.150
IP: 172.16.2.1
IP: 172.16.3.1
172.16 . 2 . 160
Network Subnet Host
Routing Table
Network Interface
172.16.2.0 E0
172.16.3.0 E1

서브넷 어드레싱

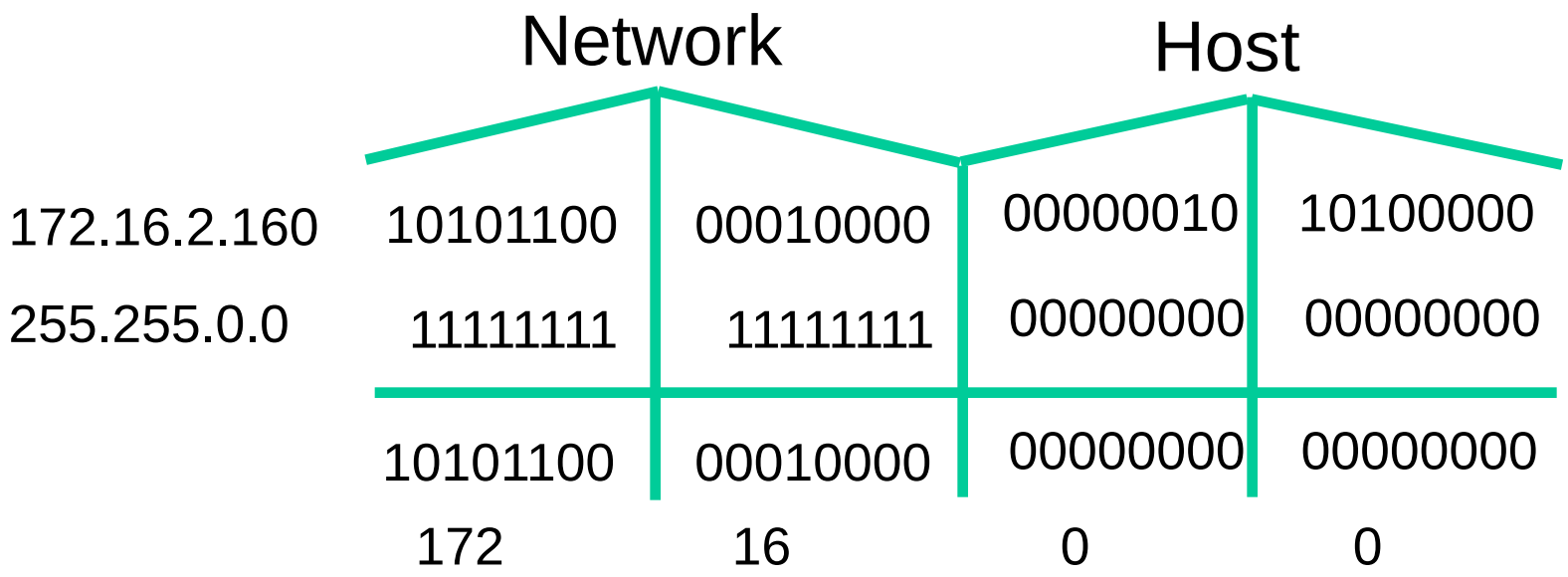


- 서브네팅은 주소를 다시 작은 영역으로 나누는걸 의미..
  - 네트워크를 체계적으로 관리 할 수 있게 한다
  - 부서별로 네트워크를 나눌 수 있으므로 보안성이 좋다.
  - 내부에서 브로드캐스트 문제를 줄일 수 있다.
  - 밖에서는 하나로 보이므로 라우팅 정보를 줄일 수 있다 .
  - (131.108.0.0)

Subnet Mask 만들기

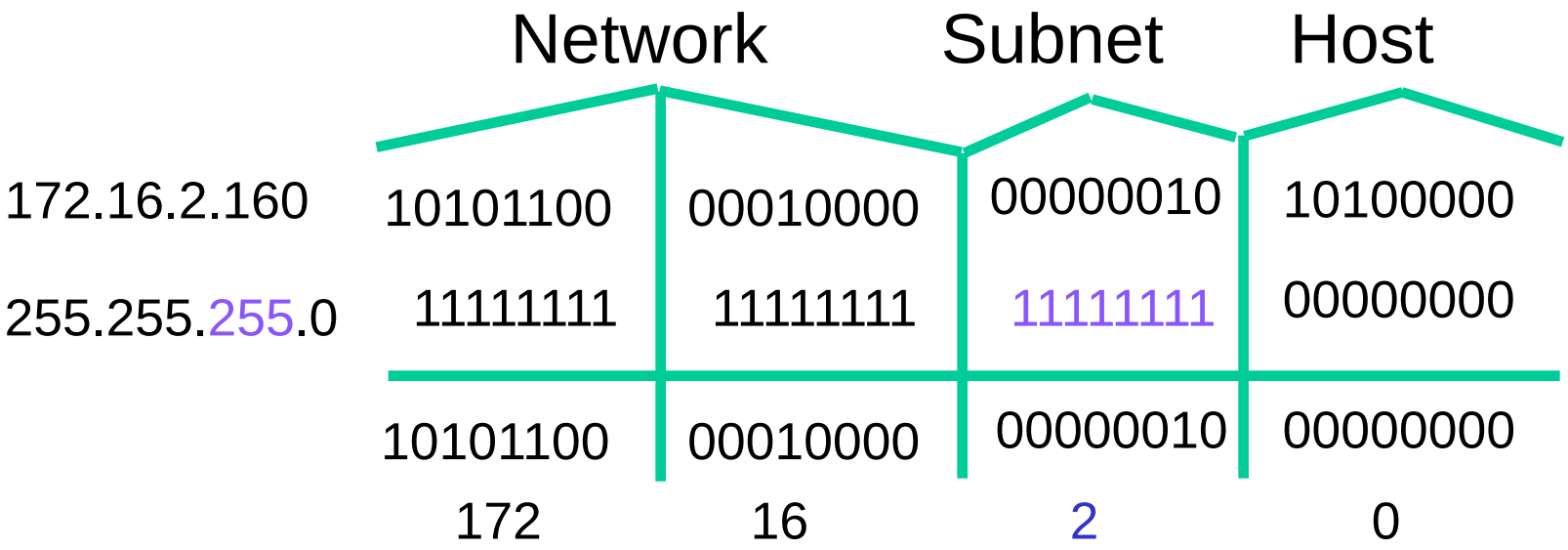


서브넷을 하지 않은 경우 (디폴트 서브넷 마스크)



- 서브넷을 하지 않으면 디폴트 서브넷 마스크를 사용한다.

서브넷을 수행한 서브넷마스크



- Network number가 8비트 더 늘어났다.

서브넷마스크

Address	Subnet Mask	Class	Subnet
172.16.2.10	255.255.255.0	B	172.16.2.0
10.6.24.20	255.255.0.0	A	10.6.0.0
172.30.36.12	255.255.255.0	B	172.30.36.0

IP 주소

- 사설 IP
  - ✓ Class Networks
    - A 10.0.0.0
    - B 172.16.0.0 through 172.31.0.0
    - C 192.168.0.0 through 192.168.255.0

IP 주소

- Class 당 예약된 IP
  - ✓ 각 Class마다 나타낼 수 있는 최대 네트워크수 및 컴퓨터 수는 해당 주소를 표기하는 비트 수(n)를 2의 n승 -2한 값과 같다.
  - ✓ Host 주소가 모두 0인 것 : local node 주소를 말한다.
  - ✓ Host 주소가 모두 1인 것: 해당 네트워크의 모든 컴퓨터를 말한다 → broadcast 주소
  - ✓ → 따라서, 총 가능한 호스트 주소 중 2개는 사용하지 못함

Subnet 나누기

- Subnet (서브넷) 나누기
  - ✓ Subnetmask에 따른 서브넷 개수와 가능한 주소 대역
  - ✓ 예 1
    - ➢ Network : 172.16.0.0
    - ➢ Subnetmask : 255.255.255.0
    - ➢ 가능한 subnet → 총 256개
      - 172.16.0.0
      - 172.16.1.0
      - 172.16.2.0
      - …
      - 172.16.254.0
      - 172.16.255.0

Subnet 나누기

- Subnet (서브넷) 나누기
  - ✓ Subnet 나누기 원리
    - Default Subnetmask 이외의 subnetmask에 해당하는 수를 2진수로 바꾸고, 1에 해당하는 자리에 모든 가능한 경우를 대입
    - Network : 172.16.0.0, Subnetmask : 255.255.255.0 의 경우,
    - Default subnetmask는 255.255.0.0
    - 따라서 255에 해당하는 것을 이진수로 변환

11111111

(1에 해당하는 자리에 모든 가능한 경우 수 적용)

00000000 → 0 → 172.16.0.0
00000001 → 1 → 172.16.1.0
00000010 → 2 → 172.16.2.0
00000011 → 3 → 172.16.3.0
…
11111101 → 253 → 172.16.253.0
11111110 → 254 → 172.16.254.0
11111111 → 255 → 172.16.255.0

총 256개

Subnet 나누기

- Subnet (서브넷) 나누기
  - ✓ 가능한 호스트 개수
    - Default Subnetmask 이외의 subnetmask에 해당하는 수를 2진수로 바꾸고, 0에 해당하는 자리에 모든 가능한 경우를 대입
    - Network : 172.16.0.0, Subnetmask : 255.255.255.0 의 경우,
    - Default subnetmask는 255.255.0.0, subnet 172.16.1.0의 경우
    - 0에 해당하는 것을 이진수로 변환 (단, 2개는 host주소로 사용 못함)

00000000

(0에 해당하는 자리에 모든 가능한 경우 수 적용)

00000000 → 0 → 172.16.1.0 → local 주소로 사용
00000001 → 1 → 172.16.1.1
00000010 → 2 → 172.16.1.2
00000011 → 3 → 172.16.1.3
…
11111101 → 253 → 172.16.1.253
11111110 → 254 → 172.16.1.254
11111111 → 255 → 172.16.1.255 → broadcast 주소로 사용

(256 – 2)
총 254개

IP 주소

- Subnetting a Class C network ID

Subnet 개수	Subnet Mask	Number of Hosts per Subnet
1-2	255.255.255.128	126
3-4	255.255.255.192	62
5-8	255.255.255.224	30
9-16	255.255.255.240	14
17-32	255.255.255.248	6
33-64	255.255.255.252	2

Subnet 나누기

- Subnet (서브넷) 나누기
  - ✓ Subnet 나눈 원리
    - Default Subnetmask 이외의 subnetmask에 해당하는 수를 2진수로 바꾸고, 1에 해당하는 자리에 모든 가능한 경우를 대입
    - Network : 221.154.90.0, Subnetmask: 255.255.255.128 의 경우,
    - Default subnetmask는 255.255.255.0
    - 128에 해당하는 것을 이진수로 변환

10000000

(1에 해당하는 자리에 모든 가능한 경우 수 적용)

00000000 → 0 → 221.154.90.0

10000000 → 128 → 221.154.90.128

} 총 2개

.

Subnet 나누기

- Subnet (서브넷) 나누기
  - ✓ 가능한 호스트 개수
    - Default Subnetmask 이외의 subnetmask에 해당하는 수를 2진수로 바꾸고, 0에 해당하는 자리에 모든 가능한 경우를 대입
    - Network : 221.154.90.0, Subnetmask : 255.255.255.128 의 경우,
    - subnet 221.154.90.128의 경우
    - 0에 해당하는 것을 이진수로 변환 (단, 2개는 host주소로 사용 못함)

10000000

(0에 해당하는 자리에 모든 가능한 경우 수 적용)

10000000 → 128 → 221.154.90.128 → local 주소로 사용
10000001 → 129 → 221.154.90.129
10000010 → 130 → 221.154.90.130
10000011 → 131 → 221.154.90.131
…
11111101 → 253 → 221.154.90.253
11111110 → 254 → 221.154.90.254
11111111 → 255 → 221.154.90.255 → broadcast 주소로 사용

(128 – 2)
총 126개

Subnet 나누기

- Network : 221.154.90.0,
  Subnetmask: 255.255.255.128 의 경우
  - ✓ 두 개의 서브네트워크
  - ✓ 221.154.90.0 서브네트워크 주소 범위:
    221.154.90.0 ~ 221.154.90.127
    (2개의 주소는 사용 못함) → 126개 사용 가능
  - ✓ 221.154.90.128 서브네트워크 주소 범위:
    221.154.90.128 ~ 221.154.90.255
    (2개의 주소는 사용 못함) → 126개 사용 가능

Subnet 나누기

- Subnet (서브넷) 나누기
  - ✓ Subnet 나눈 원리
    - ➢ Network : 221.154.90.0, Subnetmask: 255.255.255.224 의 경우,
    - ➢ Default subnetmask는 255.255.255.0
    - ➢ 224에 해당하는 것을 이진수로 변환

111**00000**

(1에 해당하는 자리에 모든 가능한 경우 수 적용)

000**00000**	→ 0	→ 221.154.90.0
001**00000**	→ 32	→ 221.154.90.32
010**00000**	→ 64	→ 221.154.90.64
011**00000**	→ 96	→ 221.154.90.96
100**00000**	→ 128	→ 221.154.90.128
101**00000**	→ 160	→ 221.154.90.160
110**00000**	→ 192	→ 221.154.90.192
111**00000**	→ 224	→ 221.154.90.224

총 8개

Subnet 나누기

- Subnet (서브넷) 나누기
  - ✓ 가능한 호스트 개수
    - Default Subnetmask 이외의 subnetmask에 해당하는 수를 2진수로 바꾸고, 0에 해당하는 자리에 모든 가능한 경우를 대입
    - Network : 221.154.90.0, Subnetmask : 255.255.255.224 의 경우,
    - subnet 221.154.90.32의 경우
    - 0에 해당하는 것을 이진수로 변환 (단, 2개는 host주소로 사용 못함)

111**00000**

(0에 해당하는 자리에 모든 가능한 경우 수 적용)

001**00000** → 32 → 221.154.90.32 → local 주소로 사용
001**00001** → 33 → 221.154.90.33
001**00010** → 34 → 221.154.90.34
001**00011** → 35 → 221.154.90.35
…
001**11101** → 61 → 221.154.90.61
001**11110** → 62 → 221.154.90.62
001**11111** → 63 → 221.154.90.63 → broadcast 주소로 사용

(32 – 2)
총 30개

소켓 생성

2. 소켓의 생성과 프로토콜의 설정

- 프로토콜의 정의

1. 컴퓨터(호스트) 상호간의 대화에 필요한 통신 규약
   - 호스트 상호간에는 프로토콜에 대한 이해가 필요.

2. 프로토콜은 잘 정의되어야 하며, 혼돈의 소지가 있으면 않 된다.

3. 이미 정의 되어 있는 프로토콜도 존재하며, 앞으로 우리도 많은 프로토콜을 설계 할 것이다.

2. 소켓의 생성과 프로토콜의 설정

- 소켓의 생성

1. 소켓은 기본적인 통신의 도구이다.

2. 파일 디스크립터를 리턴하는 함수

3. 데이터 전송 타입을 지정 해 줘야 한다.
   - 연결 지향 소켓, 비 연결 지향 소켓.

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

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

2. 소켓의 생성과 프로토콜의 설정

- **프로토콜 체계 (첫번째 인자)**

프로토콜 체계	정 의
PF_INET	IPv4 인터넷 프로토콜
PF_INET6	IPv6 인터넷 프로토콜
PF_LOCAL	Local 통신을 위한 UNIX 프로토콜 (내부프로세스간 통신)
PF_PACKET	Low level socket을 위한 인터페이스
PF_IPX	IPX 노벨 프로토콜

2. 소켓의 생성과 프로토콜의 설정

- 소켓의 타입 : 데이터전송의 타입 (두번째 인자)

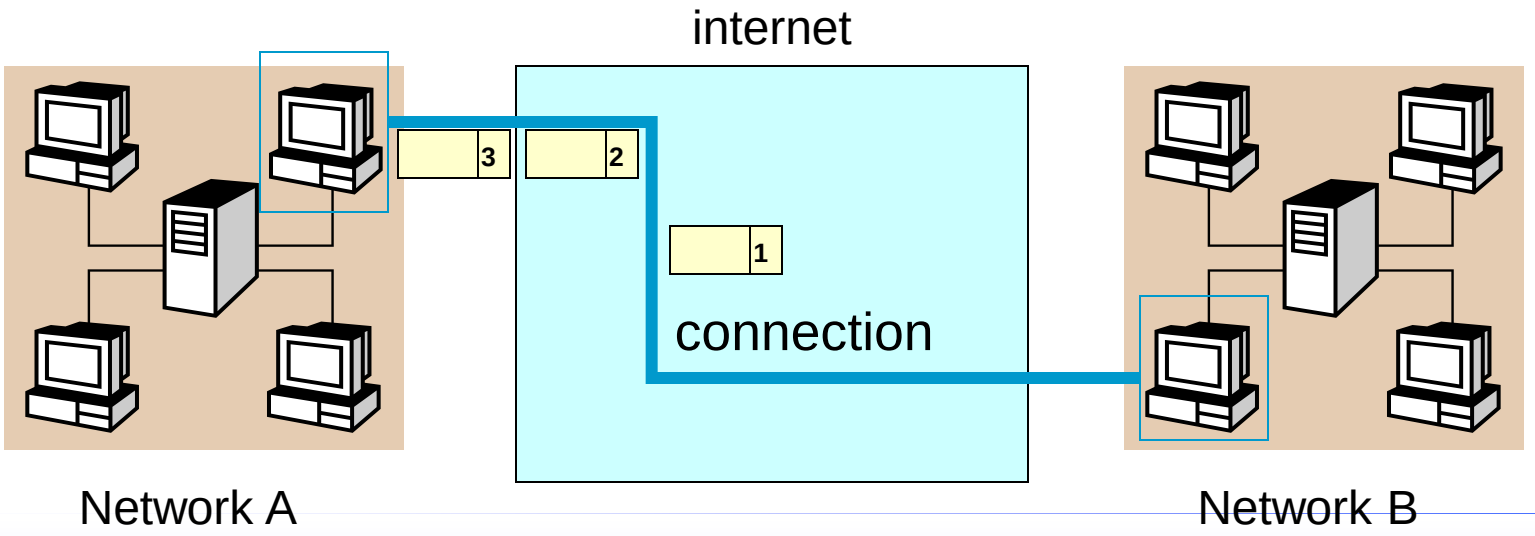
1. **SOCK_STREAM** : 연결지향형(TCP)
2. **SOCK_DGRAM** : 비연결형(UDP)
3. SOCK_RAW : TCP/UDP를 거치지 않고 바로 IP계층을 이용하는 모드
4. SOCK_PACKET : IP층도 거치지않고 바로 링크계층 인터페이스를 이용하는 모드(리눅스에서 제공)

2. 소켓의 생성과 프로토콜의 설정

- 소켓의 타입 : 데이터전송의 타입

1. 연결 지향형 소켓(Connection-Oriented : SOCK_STREAM, TCP 소켓)

- 에러나 데이터의 손실 없이 무사히 전달된다.
- 전송하는 순서대로 데이터가 전달된다.
- 전송되는 데이터의 경계가 존재하지 않는다.

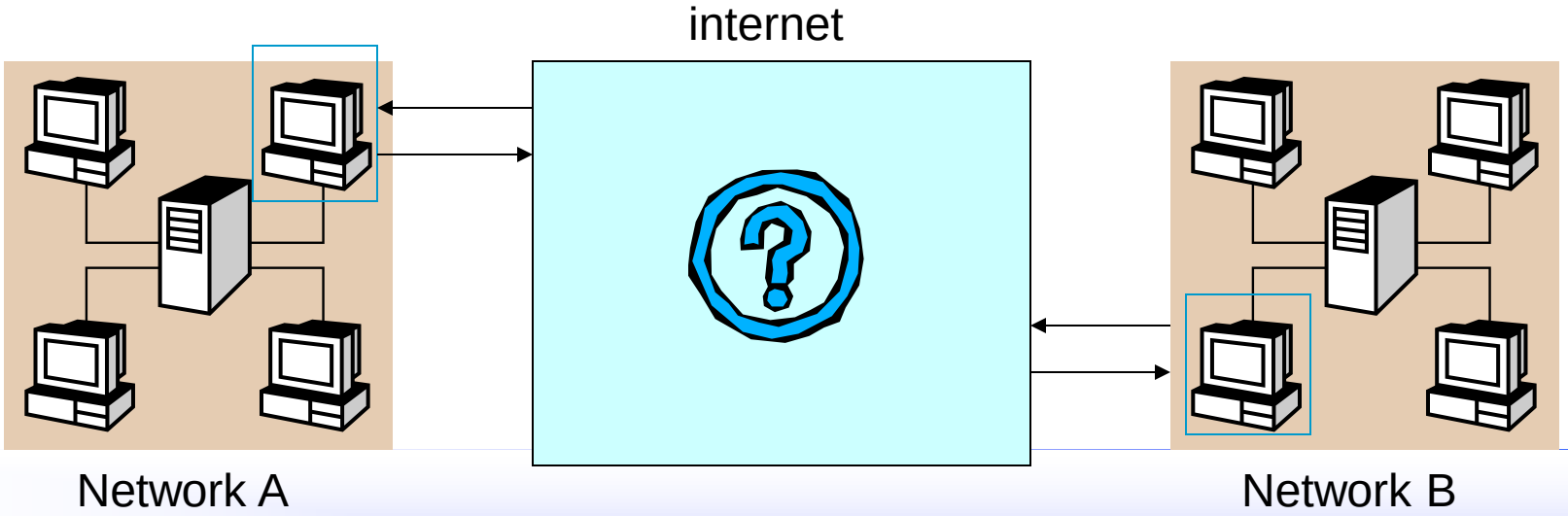


2. 소켓의 생성과 프로토콜의 설정

- 소켓의 타입

2. 비연결형 소켓(Connectionless : SOCK_DGRAM, UDP 소켓)

- 전송되는 순서에 상관없이 가장 빠른 전송을 지향한다.
- 전송되는 데이터는 손실 될 수도, 에러가 발생할 수도 있다.
- 전송되는 데이터의 경계가 존재한다.
- 한번에 전송되는 데이터의 크기는 제한된다.



2. 소켓의 생성과 프로토콜의 설정

- **프로토콜의 선택 (세번째인자)**
  - 호스트 대 호스트가 사용할 프로토콜을 선택

CASE	Domain	Type	Protocol
1	PF_INET	SOCK_STREAM	IPPROTO_TCP (0)
2	PF_INET	SOCK_DGRAM	IPPROTO_UDP (0)

세 번째 인자는 프로토콜을
더 구체화하기위해 사용
보통의 경우 0 으로 표시하기도 한다.

2. 소켓의 생성과 프로토콜의 설정

1. 소켓의 생성

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

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

2. 소켓의 종료

```
#include <unistd.h>

int close(int fildes);
```

예제

- 프로그램 예제
  - ✓ tcp_client.c
  - ✓ 특징: 서버에서는 한번의 write를 하였으나, client는 여러 번의 read를 수행하여 데이터를 수신
  - ✓ → “TCP는 데이터의 경계가 없다”를 보여줌

실습

- 프로그램 예제
  - ✓ IPPROTO_TCP 대신에 0 값을,
    IPPROTO_UDP 대신에 0 값을 입력하여
    실행해 본다.

Q&A