3계층(Network)

네트워크 계층

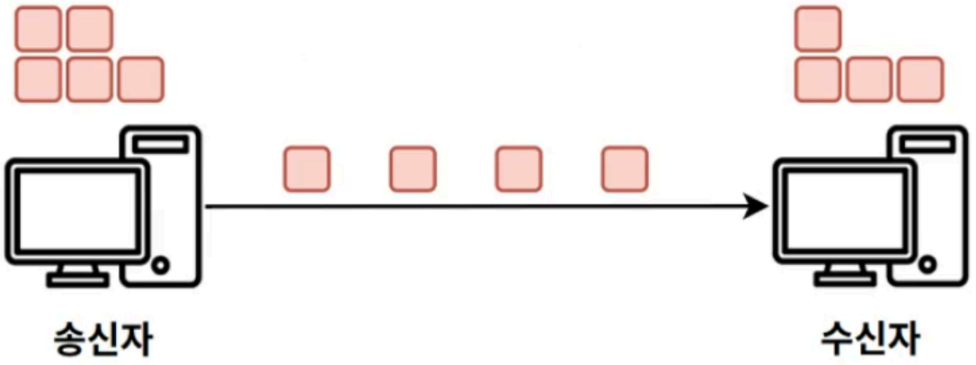
- 데이터를 출발지에서 목적지까지 전달할 수 있도록 최적의 경로를 설정한다.
- 서로 다른 네트워크를 연결하여 통신 가능토록 한다
- Routing(경로결정), Forwarding(전달), Fragmetation(단편화) 등의 기능을 수행한다.

데이터 단위 : Packet

장비 : 라우터

Fragmentation (단편화)

- 패킷의 크기가 네트워크의 최대 전송 단위(MTU)를 넘어설 경우 패킷을 작게 나누는 작업
- MTU보다 큰 패킷은 네트워크 계층에서 여러 개의 조각으로 분할되어 전송된다
- 단편화된 패킷들은 수신지에서 재조립되어서 원래 데이터로 복원된다.



IP 주소 (Internet Protocol Address)

- 네트워크 상에서 장치를 식별 하기 위한 주소
- 논리적으로 설정된 주소이므로 사용자 임의로 변경 가능하다
- 패킷의 출발지와 목적지를 지정한다
- 32비트로 구성되며 1옥텟(8비트)마다 점을 찍어 구분한다.

0000 0000 =0 / 1111 1111 =255

1100 0000. 1010 1000. 0000 0001. 0000 1010 =192.168.1.10

1111 1111. 1111 1111. 1111. 1111. 1111 1111 =255.255.255.255
43억개 존재

IP 주소 버전

1. IPv4

- 32비트의 주소 체계로 10진수를 사용해 표기한다
- 약 43억개의 주소를 제공하며 현재 가장 널리 이용된다
- IP 주소 부족 문제로 인해 NAT와 함께 사용한다

2. IPv6

- 128비트의 주소 체계로 무제한에 가까운 주소를 제공한다.
- IPv4 환경이 아직 안정적으로 운영되어 아직 많이 이용하진 않는다.
- 16진수로 표현한다.

IP 주소 분류

1. 사설 IP 주소

- 내부 네트워크(LAN)에서 사용되는 IP 주소
- 서로 다른 네트워크에 존재한다면 중복해서 사용할 수 있다.
- 인터넷에서 직접적으로 사용하지 못한다.

2. 공인 IP 주소

- 인터넷에서 사용하는 전 세계 유일한 IP 주소
- 임의로 사용하는 것이 불가능하고 ISP를 통해서만 할당된다
- 웹 서버와 같은 인터넷에 공개된 서비스, 사설IP 들의 NAT에 사용된다

==========

NAT(Network Address Translation)

- 사설 IP 주소를 공인 IP 주소로 변환하는 기술
- IP 주소를 절약하기 위해서 사용한다

IP 주소 클래스

- 초기 네트워크에서 IP를 효율적으로 사용하기 위해서 등장하였다
- IP 주소를 네트워크 규모와 용도에 따라 분류한 체계

- IP 주소의 첫 옥텟에 따라 클래스가 자동으로 결정된다.

클래스	첫번째 옥텟 2 진수	첫번째 옥텟 10 진수	호스트 개수	Default 서브넷 마스크
A Class	0xxx xxxx	0 ~ 126	16,777,214개	255.0.0.0
B Class	10xx xxxx	128 ~ 191	65,534개	255.255.0.0
C Class	110x xxxx	192 ~ 223	254개	255.255.255.0
D Class	1110 xxxx	224 ~ 239	-	-
E Class	1111 xxxx	240 ~ 255	-	-

==========

서브넷 마스크(subnet mask)

- IP 주소에서 네트워크 부분과 호스트 부분을 구분하기 위하여 사용하는 값

클래스리스가 최근 동향

127.0.0.1 루프백(자기 자신)

==========

클래스별 사설 IP 주소 대역

-각 클래스마다 사설 IP 주소로 사용할 수 있는 대역이 정해져있다.

-사설 IP와 공인 IP의 혼용, 충돌을 방지하기 위한 영역을 구분한다.

A 클래스 사설 IP 대역

10.0.0.0 ~ 10.255.255.255

B 클래스 사설 IP 대역

172.16.0.0. ~ 172.31.255.255

C 클래스 사설 IP 대역

192.168.0.0 ~ 192.168.255.255

1. A클래스 사설
2. A클래스공인
3. B공인
4. B클래스 사설
5. C공인
6. B공인
7. C공인
8. C클래스 사설
9. A공인
10. C공인

서브넷 마스크(Subnet Mask)

- IP 주소의 네트워크 부분과 화스트 부분을 구분하는데 사용되는 값
- 서브넷 마스크를 통해 네트워크 대역, 호스트 IP 개수를 정의한다
- 32비트로 구성되며 1은 네트워크 비트 0은 호스트 비트를 나타낸다
- 네트워크 비트와 호스트 비트가 절대 섞이지 않는다.

서브넷 마스크 표기법

1. 이진수 표기법

- 서브넷 마스크를 이진수를 표기하는 방법
- ex) 1111 1111. 1111 1111. 1111 1111. 0000 0000

2. 십진수 표기법

- 서브넷 마스크를 진수로 표기 ex) 255.255.255.0

3. Prefix 표기법

- ip 주소의 뒤에 /를 붙이고 네트워크의 비트 갯수를 적는방법

십진수: 0~9, 숫자 10개를 사용 수를 표현

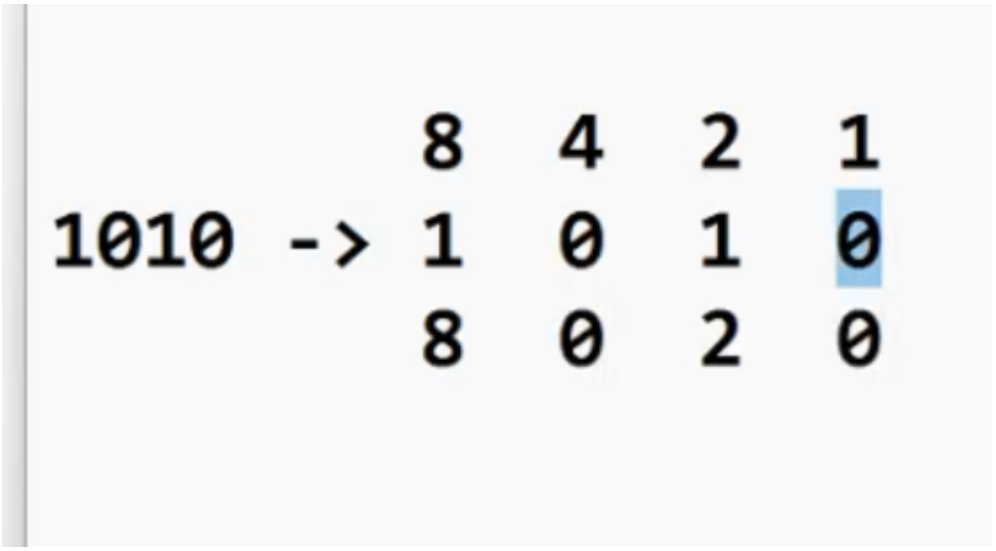
이진수: 0,1 두 개의 숫자를 사용하여 수를 표현

0=0

1=1

2=10

3=11



1100 = 12

1110 = 14

1111 =15

1010 1110 =174

1100 0010 = 128 + 64 + 2

1번 2번 4번

CIDR (Classless Inter-Domain Routing)

- 기존의 클래스 기반 네트워크에서 클래스 개념을 없애 네트워크를 더 유연하게 사용하기 위한 방법

- 네트워크를 효율적으로 분할하거나 합칠 수 있다
- 낭비되는 IP의 갯수의 최소한으로 조절할 수 있다

192.168.108.0/24 192.168.108.0/25

==========

prefix 퀴즈

/25 : 255.255.255.128 호스트 ip 개수 128개

/26: 255.255.255.192 호스트 ip 개수 64개

/27: 255.255.255.224 호스트 ip 개수 32개

/28: 255.255.255.240 호스트 ip 개수 16개

/29: 255.255.255.248 호스트 ip 개수 8개

/30: 255.255.255.252 호스트 ip 개수 4개

==========

- 네트워크 ID (네트워크 자체를 표현하기 위한 이름)
- 네트워크를 식별하기 위해 사용하는 IP
- 네트워크의 첫 번째 IP

- 브로드캐스트 IP
- 네트워크 내의 모든 장치에 데이터를 전송하기 위해서 사용하는 주소
- 네트워크의 마지막 IP

10.0.0.0 /24

10.0.0.0~10.0.0.255

Network ID: 10.0.0.0

BroadCast IP: 10.0.0.255

2

네트워크 ID : 172.16.10.64 [172.16.10.64 ~ 172.16.10.127]

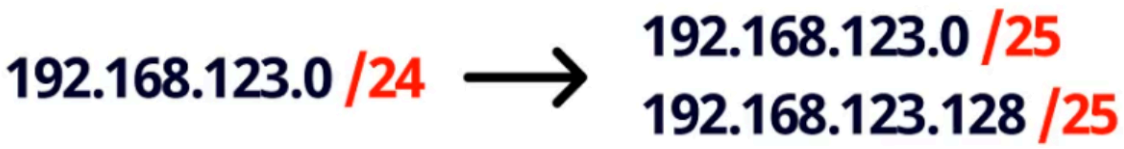
Prefix: /26 → 64개

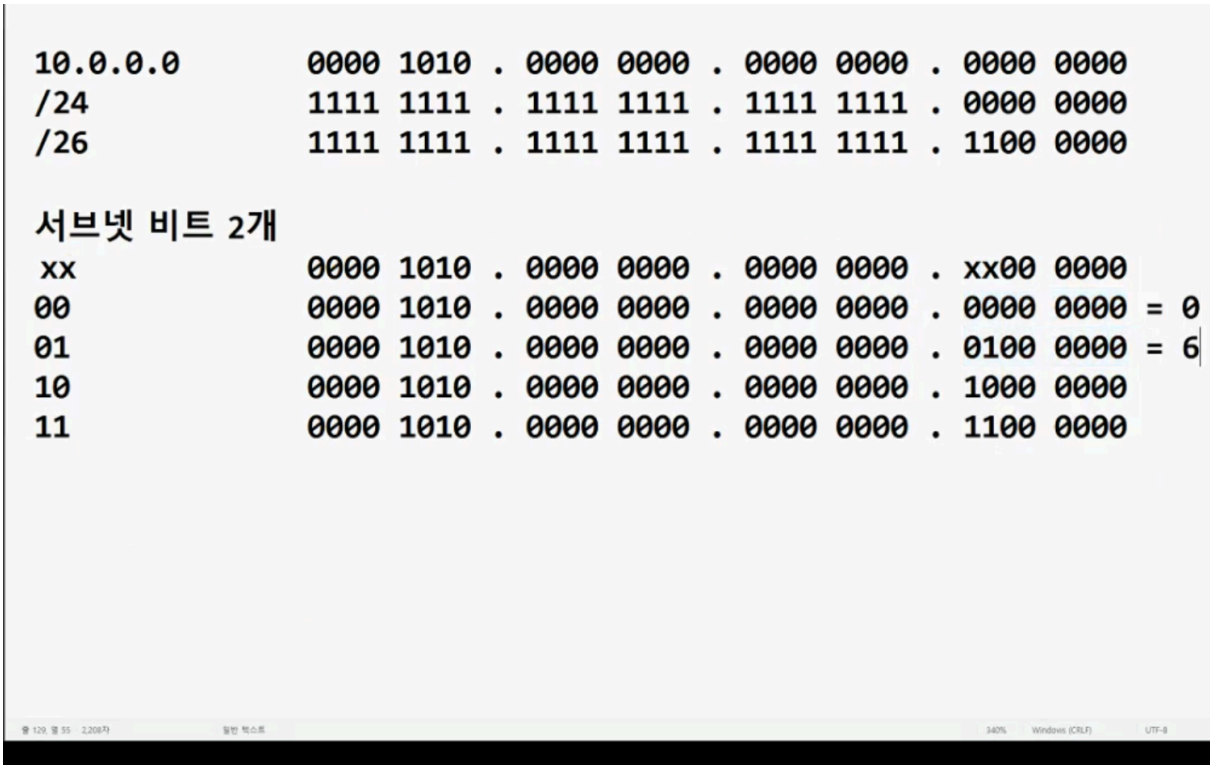
BroadCast IP : 172.16.10.127

실 사용할 수 있는 IP 대역: 172.16.10.65 ~ 172.16.10.126

서브네팅(Subnetting)

- 하나의 큰 네트워크를 여러 개의 작은 서브넷으로 분할하는 작업
- 서브넷 마스크를 조정하여 네트워크를 나눌 수 있다..





1. 서브넷의 호스트 개수를 알면 쉽게 서브네팅이 가능하다

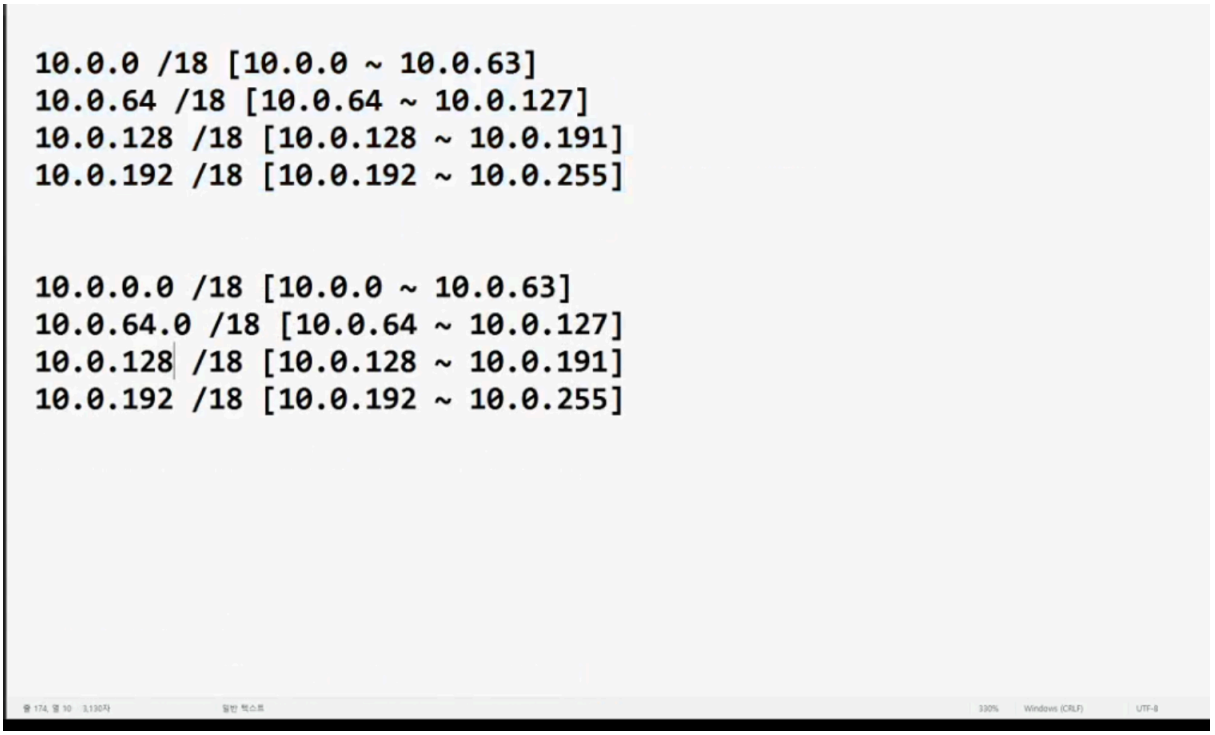
(호스트 갯수 기반 서브네팅 ~~~)

2. Prefix 값을 이용하여 서브네팅 될 네트워크의 개수를 알 수 있다

/24 → /26 → 비트 갯수 차이? 2개 xx → 서브넷 개수 4개

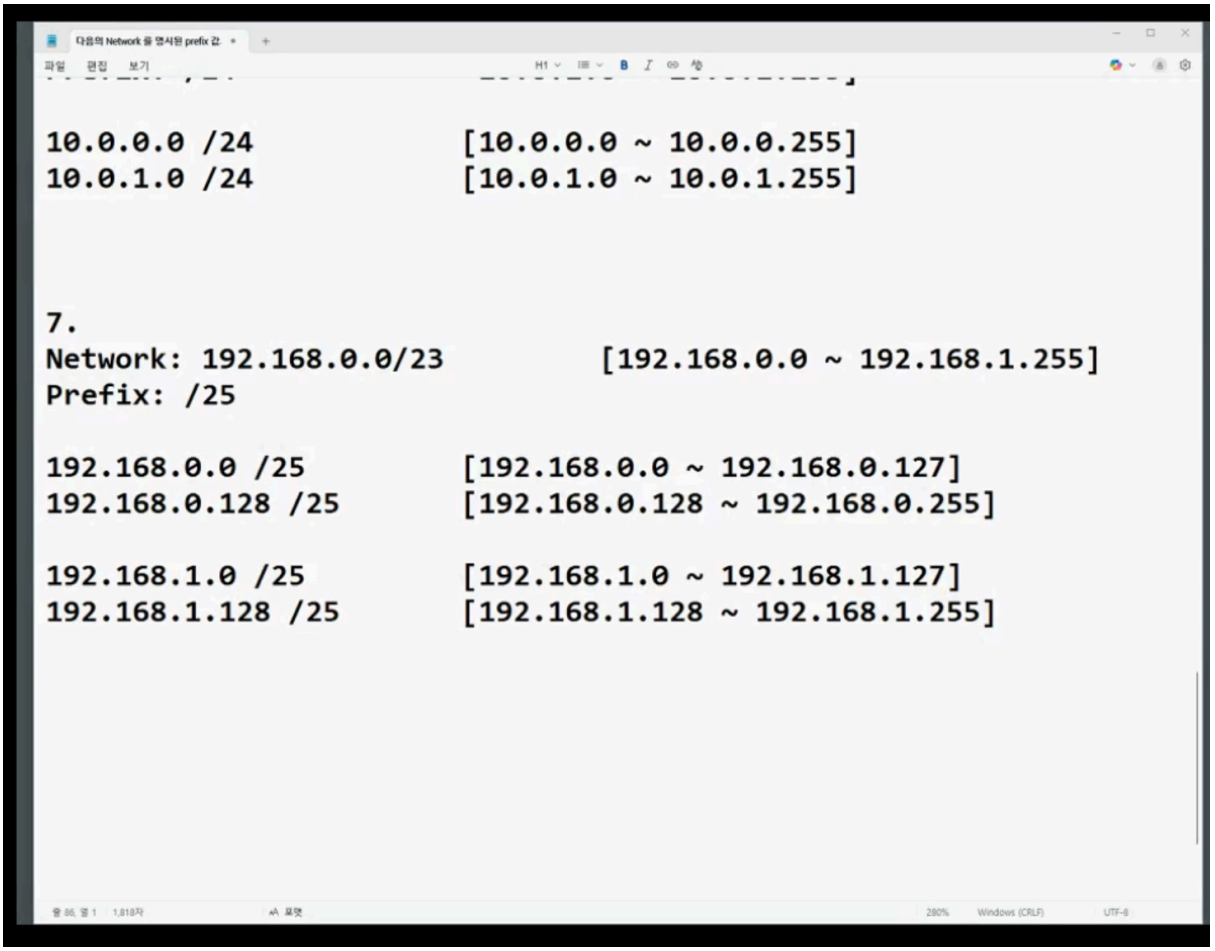
/24 → /27 q비트 갯수 차이? 3개 xxx → 서브넷 개수 8개

3. 앞쪽 옥텟에서 서브네팅이 이루어질 경우, 뒤쪽 옥텟은 머릿속에서 지워버린다.



1111 1111 .1111 1111. 1111 1111 .1(00)0 0000

1111 1111. 1111 1111. 1111 111(0) . 0000 0000



1. A 회사에서 60 대의 서버를 운영하고자 한다. 192.168.123.160 을 포함하여, IP의 낭비를 최소화하는 네트워크 / 서브넷을 찾으시오.

   /26 192.168.123.0/26

   192.168.123.64/26

   192.168.123.128/26

- 슈퍼네팅(Supernetting)
- 여러 개의 작은 연속된 네트워크를 하나의 큰 네트워크로 합치는 기술!
- 라우터가 관리해야할 경로 정보를 요약하여 라우터의 리소스를 절약하는 것을 목표로 한다.

10.0.0.0/24 → A 경로

10.0.1.0/24 → A 경로

⇒ 요거 두개 합치면 사실상 10.0.0.0/23 아닌가용? 맞습니다

10.0.0.0/23

10.0.2.0/23

10.0.4.0/23

10.0.6.0/23 ⇒10.0.0.0/21

- 라우터 인터페이스 설정

  >enbale

  #configure terminal

  (config)# int fa 0/0

  (config-if)# ip address 10.0.0.1 255.255.255.0

  (config-if)# no shutdown

  (config-if)# exit

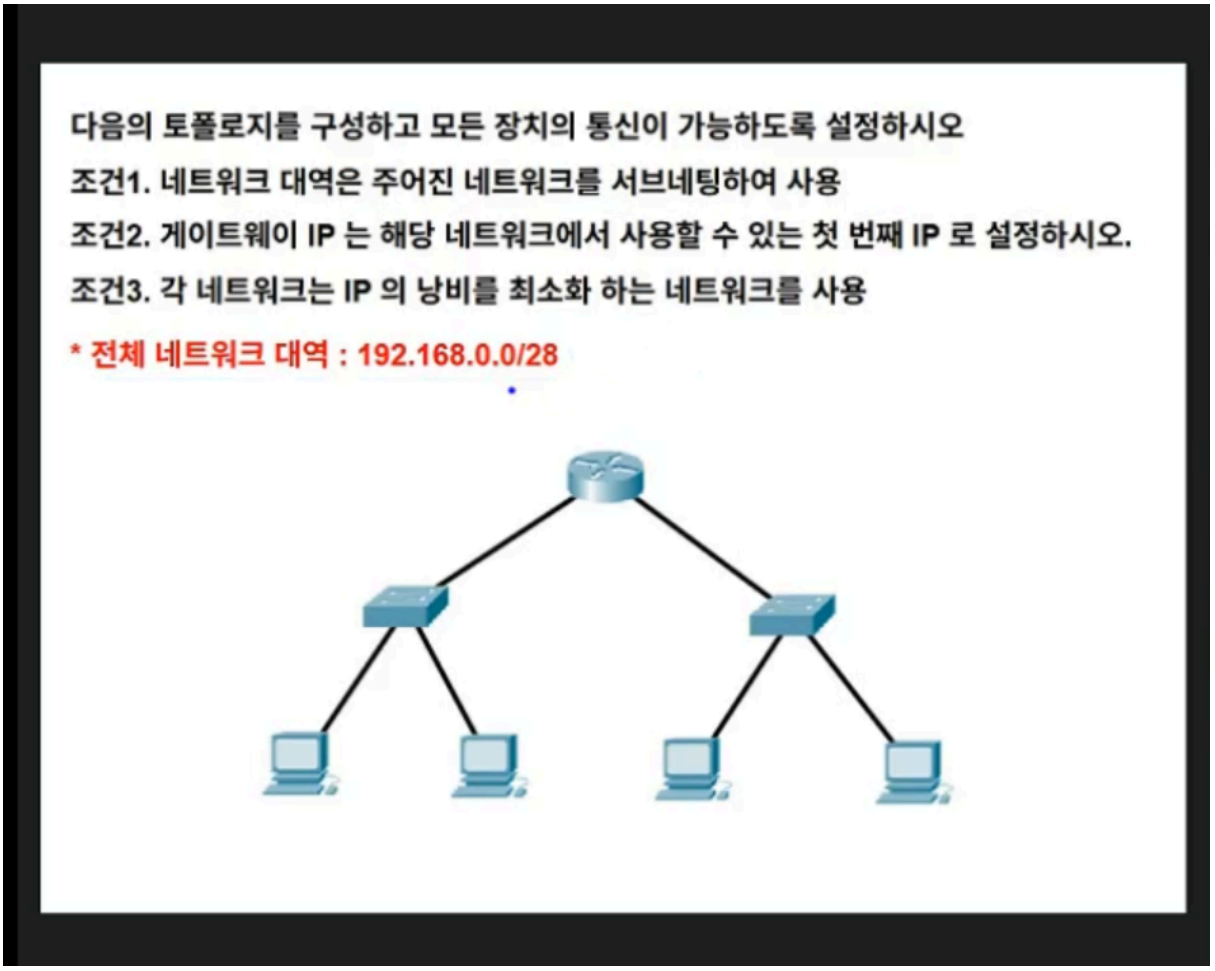
  (config-if)# int fa 0/1

  (config-if)# ip address 20.0.0.1 255.255.255.0

  (config-if)# no shutdown

  IP 취소

  (config-if)# no ip address



1111 1111 1111 1111 1111 1111 1111 0000 ⇒ 16개

240

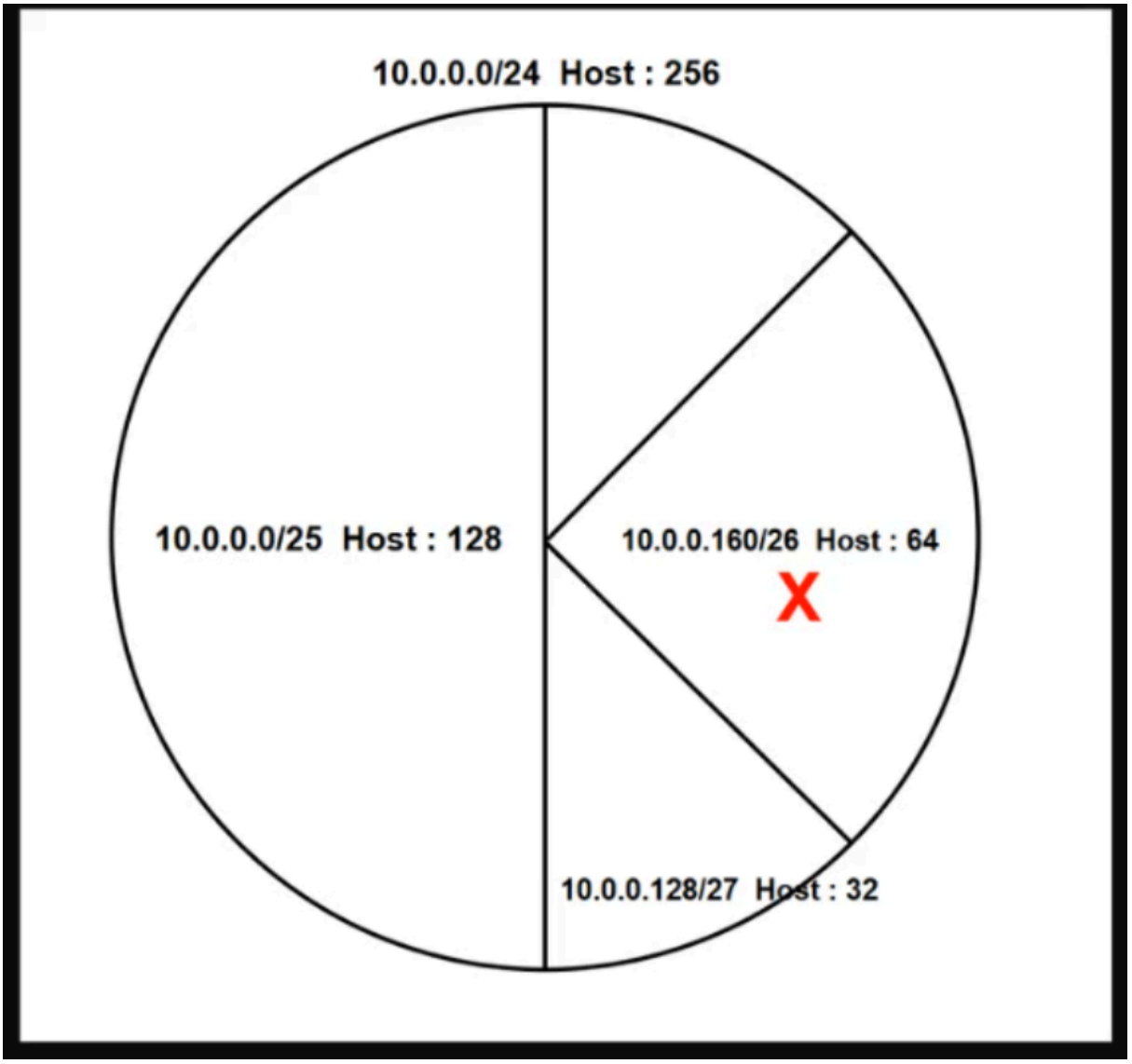
FLSM (Fixed Length Subnet Mask)

- 서브네팅의 결과가 모두 같은 크기의 서브넷으로 나온다
- 10.0.0.0 /24 → /26

VLSM (variable Length Subnet Mask)

- 네트워크마다 다른 서브넷 마스크 값을 가지는 서브네팅 방식
- 서브네팅의 확장 개념으로 서브네팅보다 유연하다
- 낭비되는 ip최소화

192.168.123.0/24 → 192.168.123.0/25, 192.168.123.128/26, 192.168.123.192/27



주어진 네트워크 대역을 사용하여 최적의 네트워크를 계산하세요

1. 네트워크 대역 :10.0.0.0/24

A. 50개 호스트 10.0.0.0/26 [10.0.0.0 ~ 10.0.0.65]

B. 20개 호스트 10.0.0.64 [10.0.0.64~10.0.0.95]

C. 10개 호스트 10.0.0.96/28 [10.0.0.96~10.0.0.111]

D. 5개 호스트 /29 : 10.0.0.112/29 [10.0.0.112 ~ 10.0.0.119]

2. 네트워크 대역 : 192.168.10.0/24

A. 50개 호스트 192.168.10.0. /26 [192.168.10.0 ~ 63]

B. 20개 호스트 192.168.10.64 /27 [192.168.10.64 ~ 95]

C. 14개 호스트 192.168.10.128 /28 [192.168.10.128 ~ 143]

D. 20개 호스트 192.168.10.96 /27 [192.168.10.96 ~ 127]

3. 네트워크 대역 :172.16.0.0/16

A. 300개 172.16.0.0 /23 [172.16.0.0~172.16.1.255]

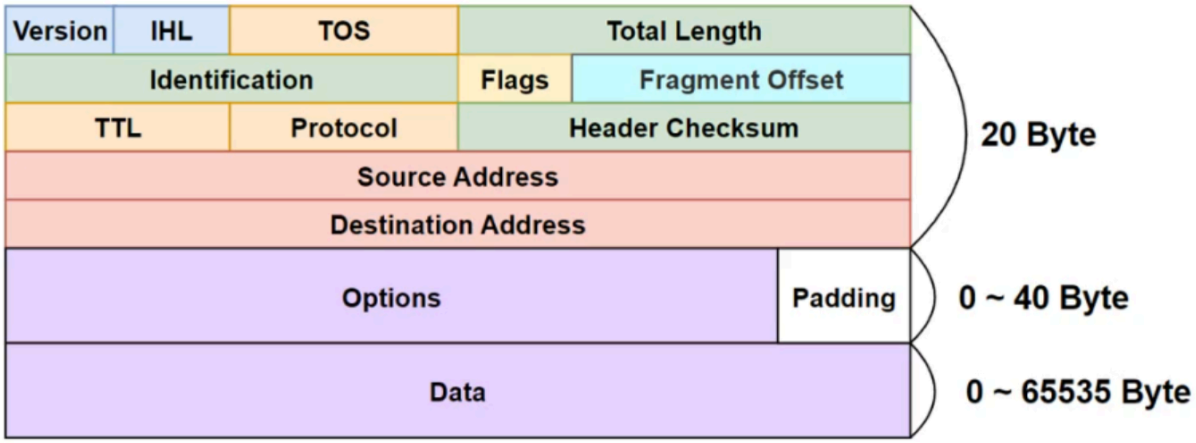
B. 100개 172.16.2.0 /25 [172.16.2.0 ~ 172.16.2.127]

C. 50개 172.16.2.128 /26 [172.16.2.128 ~ 172.16.2.191]

D. 10개 172.16.2.192 /28 [172.16.2.192 ~172.16.2.207]

IP 헤더 (IP 헤더)

- IP 를 사용하는 패킷의 헤더 부분



1. Version - 4bit

- ip 프로토콜의 버전을 나타낸다.

2. IHL (INternet Header Length, 헤더 길이) - 4 bit

- IP 헤더의 길이를 나타내는 필드
- 전체 필드에서 헤더 부분, 데이터 부분을 구분하기 위하여 사용

3. TOS (Type Of Service) - 8bit

- 패킷이 어떻게 처리되어야 하는 지 명시하는 필드
- 0~63 까지의 코드 조합으로 패킷의 처리 방식을 명시

4. Total Length (전체 길이) - 16bit

- Ip 헤더, 데이터를 포함한 패킷의 전체 길이
- 수신 측에서 패킷의 길이를 파악하여 데이터를 수월하게 처리할 수 있도록 한다.

5. Identification (식별자) - 16bit

- 조각화된 패킷을 재조립하기 위한 식별자
- 원래 어떤 데이터에서 조각화 된 것인지를 나타낸다.

6. Flags - 3bit

- 데이터의 조각화 여부, 추가 데이터 여부 등을 표시한다

X X X (R DF MF) (0 0 1)

R (Reserved) : 예약 비트. 항상 0으로 표시

DF (Don't Flag) : 데이터의 조각화 여부 표시(1: 조각화 금지, 0: 조각화 허용)

MF (More Flag) : 추가 데이터 여부 표시 (1: 추가 데이터 있음, 0 : 마지막 데이터)

7. Fragment Offset - 13bit

- 조각화된 데이터의 원래 위치를 나타낸다
- 조각화된 데이터에서 고유

8. TTL (Time to Live) - 8bit

- 패킷의 생존 시간(생존할 수 있는 홉 수)
- 네트워크에서 한 노드를 지날 때마다 1씩 감소되며 0이 되면 해당 패킷을 폐기한다
- 패킷의 루프를 방지하기 위하여 사용한다

윈도우 :128

리눅스 :64

9. Protocol - 8bit

- 상위 계층의 프로토콜
- 전달될 데이터가 상위계층에서 어떻게 사용되는지 명시하는 필드

10. Header Checksum - 16bit
    - 헤더의 오류를 검사하기 위한 필드
    - 헤더 부분만 검사한다

11. Source Address - 32 bit
    - 출발지 IP 주소

12. Destination Address - 32 bit
    - 목적지 IP 주소

13. Option - 0 Byte ~ 20 Byte
    - IP 헤더에서의 추가 제어를 위한 필드
    - 4Byte 크기로 사용된다

0~4~8~12

10Byte + 2Byte(Padding)

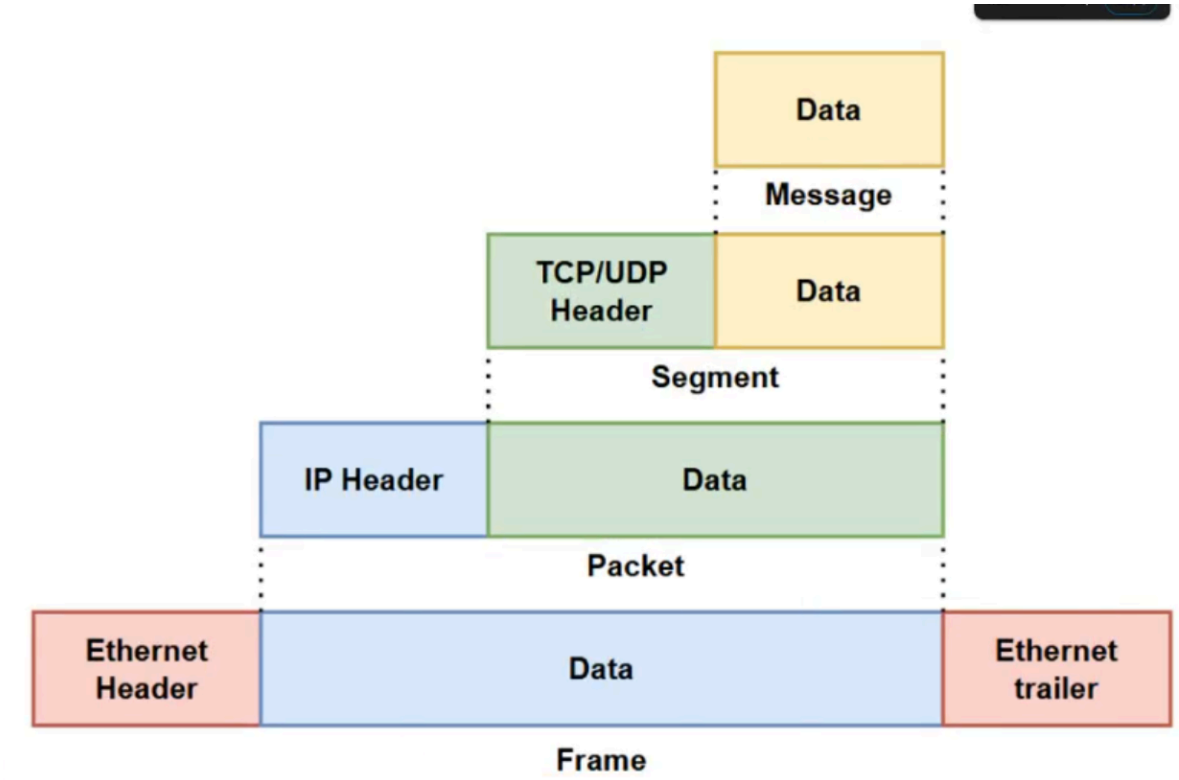
==========

- 인캡슐레이션 (캡슐화 , Encapsulation)

  데이터의 송수신을 위해 각 계층의 헤더, 트레일러가 추가되는 과정

  상위 계층에서 하위 계층으로 이동할 때에 발생한다

  각 계층들의 여러 제어 정보가 추가된다



1계층 비트

2계층 프레임

3계층 패킷

4계층 세그먼트

7계층 올라가서 메시지 된다.

로켓 분리되는 거마냥 이더넷 헤더 때지고 아이피 헤더 떼지고 티시피 헤더 떼지고 데이터만 올라간다

- 디캡슐레이션 (역캡슐화, Decapsulation)

수신받은 데이터의 헤더, 트레일러를 제거하며 상위계층으로 전달하는 과정

하위 계층에서 상위 계층으로 이동할때 발생

여러 제어정보가 사용되고 제거된다

==========

- 3계층 대표적 프로토콜

1. IP (Internet Protocol)

- Packet이 알맞는 목적지로 도착할 수 있도록, 주소를 지정해주는 역할

- IP는 Packet 순서, 분실 여부를 책임지지 않는다. (신뢰성 x 신뢰성을 책임지기 위하여 TCP를 함께 쓴다)

2. ICMP (Internet Control Message Protocol)

- IP를 보조하는 역할로 사용
- Packet이 목적지까지 도달하지 못했을 때 원인을 보고하거나 네트워크의 상태를 진단하기 위하여 사용한다.
- 라우터에서 ICMP를 만들어서 사용한다.

ping : 특정 IP 주소로 ICMP 메시지를 보내, 응답 여부로 장치의 동작 여부를 확인한다.

tracert (traceroute) : 데이터 목적지까지 어떤 경로(라우터)를 거쳐가는지 추적한다.

3. ARP (Address Resolution Protocol)

- IP 주소를 MAC 주소로 변환하는 프로토콜
- 최종적으로 데이터를 목적지까지 보낼 때에는 MAC 주소를 사용한다
- 목적지의 MAC 주소를 모를 때에는 ARP를 통하여 MAC 주소를 알아낼 수 있다.

4. 라우팅 프로토콜 (Routing Protocol)

- Router들끼리 네트워크 정보를 교환하여 데이터를 전송하기 위한 최적 경로를 찾는 프로토콜

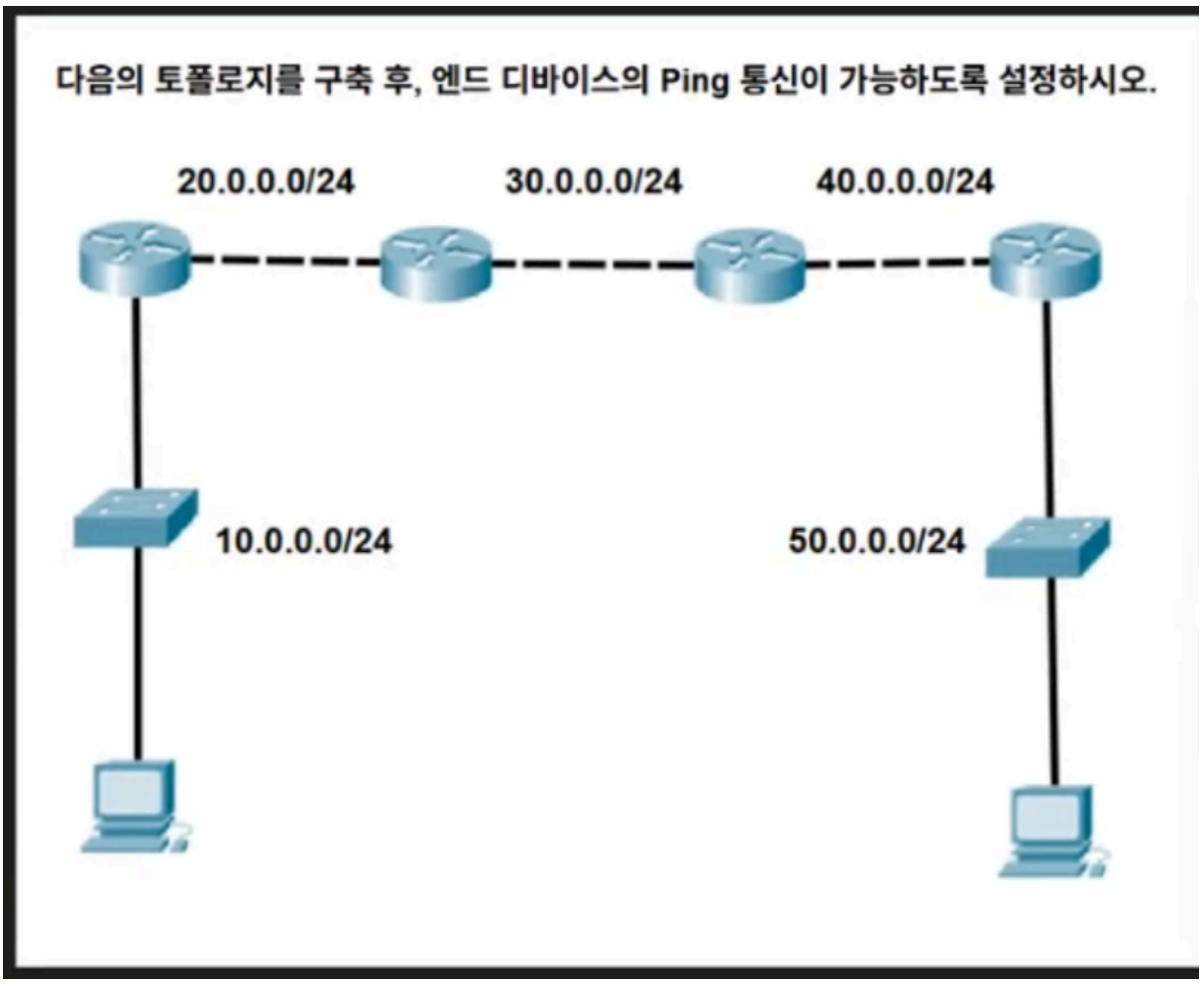
==========

- 라우팅, Routing
- 데이터를 출발지에서 목적지까지 전달하는 최적 경로를 결정하는 과정

- 스태틱 라우팅 , Static Routing

→ 관리자가 데이터가 전달된 경로를 직접 라우터에 설정하는 방식

- 라우팅 명령어
  (Config)# ip route
  [Network ID] [Subnet MAsk] [Next Hop]


다음의 토폴로지를 구축 후, 엔드 디바이스의 Ping 통신이 가능하도록 설정하시오.
20.0.0.0/24
30.0.0.0/24
40.0.0.0/24
10.0.0.0/24
50.0.0.0/24