

KGITBANK 네트워크 초급 과정

# Network Beginner Class

Chapter 02 TCP/IP 프로토콜



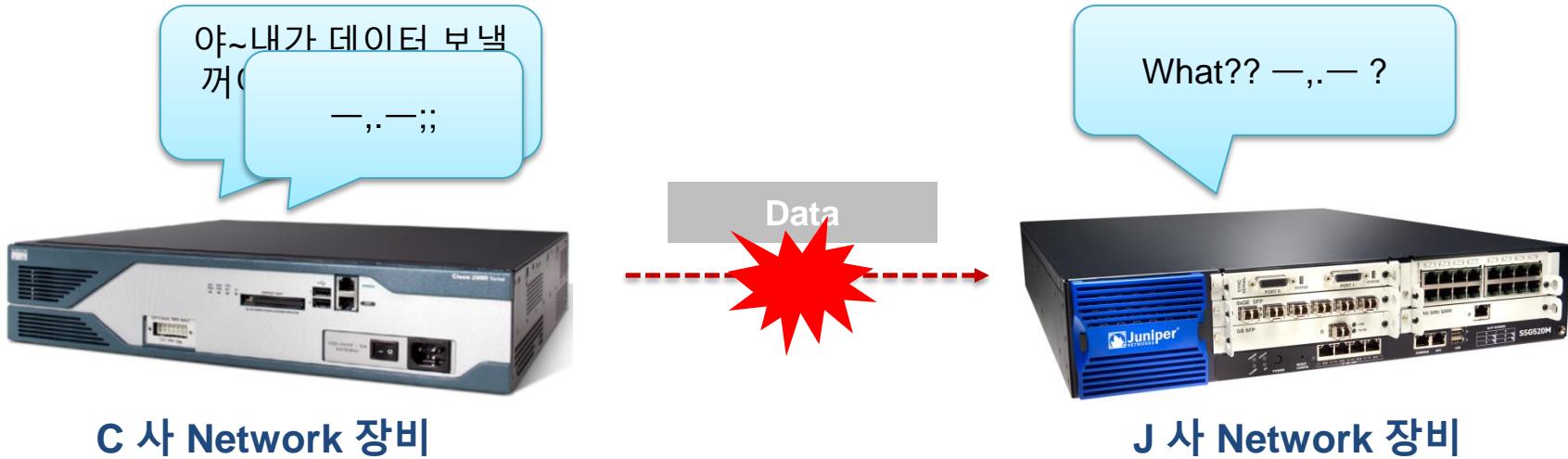
# 01

## OSI 7 Layer vs TCP/IP

### TCP/IP and OSI 7 Layer Reference Model

모델은 어려운 개념과 복잡한 시스템을 이해하는데 도움을 주기 때문에 유용하다. 네트워킹에는 다양한 기술이 수행하는 역할과 그들의 상호 작용을 설명하는 데 쓰이는 여러 모델이 존재하는데, 이 중에서 우리는 OSI 7Layers model과 TCP/IP model에 대해 살펴본다.



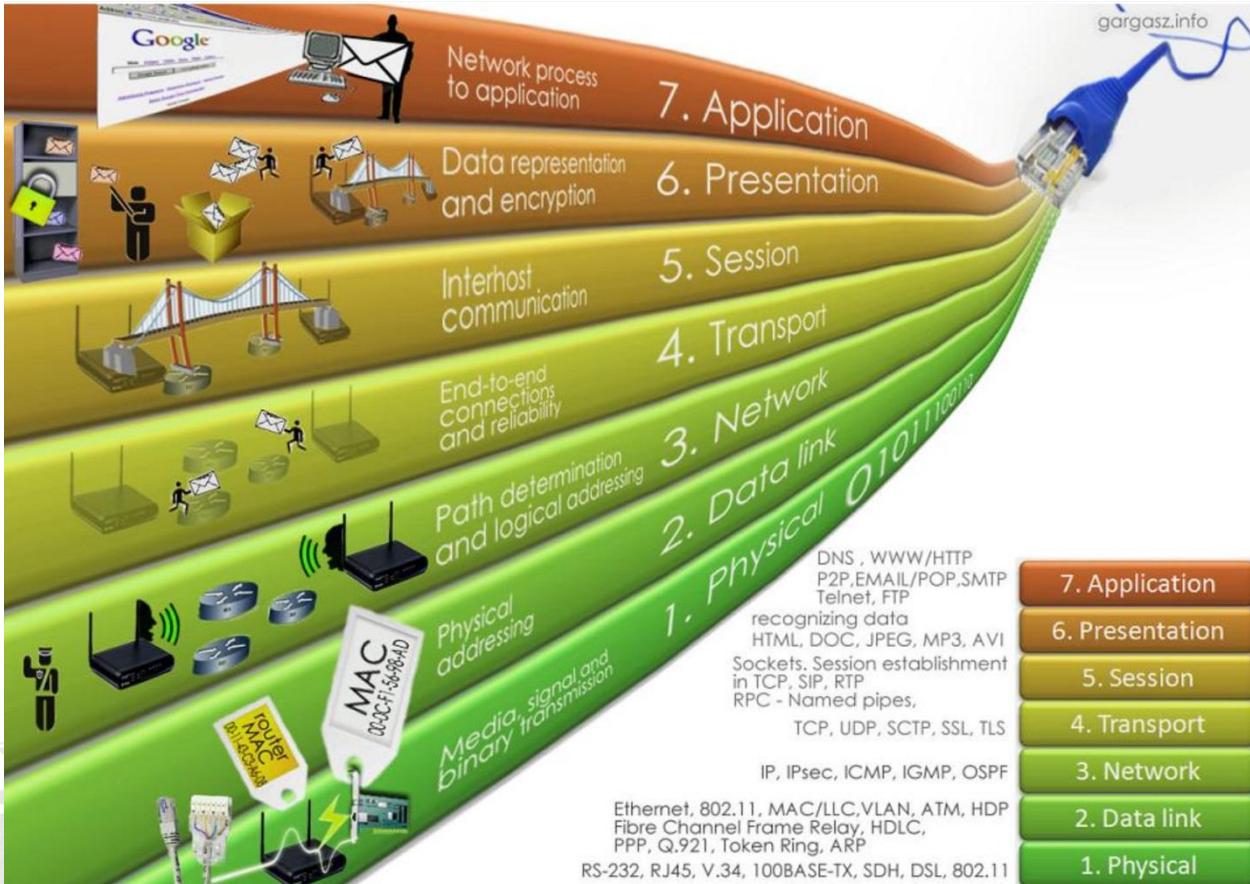


## 공통된 모델이 없는 경우

- 공통된 Model이 없던 과거의 경우 서로 다른 벤더의 장비는 호환성이 없었음.
- 서로 연결하는 널리 인정된 방법을 개발하기 위해 네트워킹 표준의 필요성이 제기됨.
- 초창기 다양한 하드웨어와 소프트웨어 회사가 여러 네트워킹 기술을 개발하기 위한 구조에 동의하도록 하기 위해 OSI 참조 모델이 시도.
- 1984년 ISO에서 ISO 7498 표준으로, CCITT(현재 ITU-T)에서 X.200 표준으로 출판됐음.

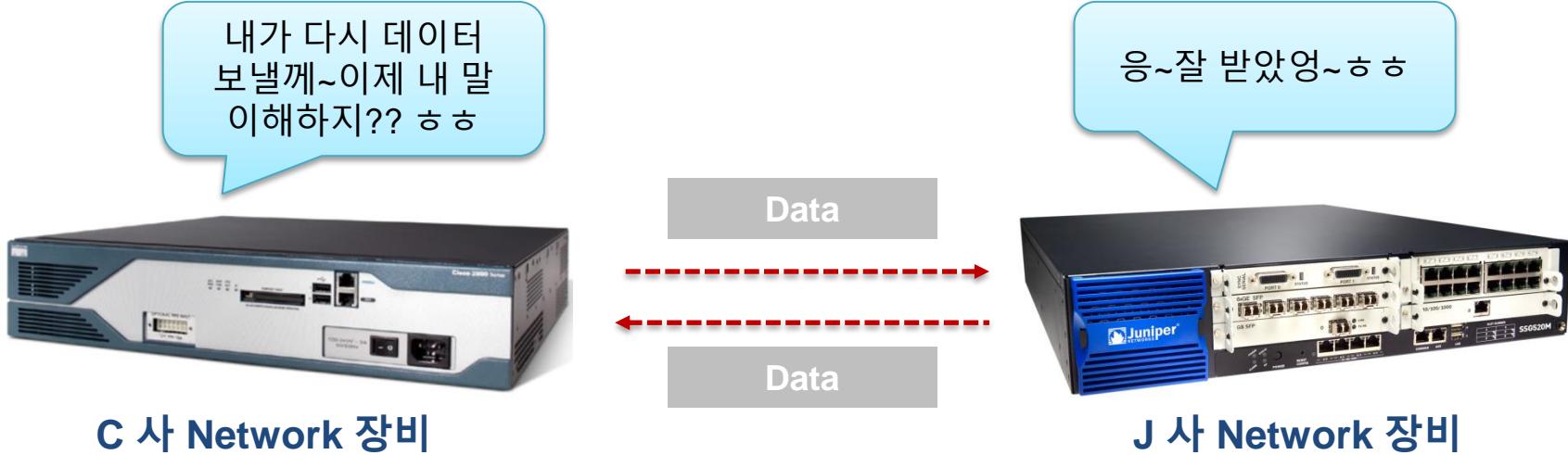
# Needs of Network model

## Chapter. 02 TCP/IP 프로토콜



# Needs of Network model

Chapter. 02 TCP/IP 프로토콜



## 공통된 모델의 필요성

- 공통된 Model이 사용된 경우 서로 다른 벤더의 장비 역시 호환성이 있기 때문에 상호간의 통신이 가능하다.
- 계층별로 구분된 Model의 경우 전체적인 네트워크 동작을 좀 더 쉽게 이해할 수 있다.
- 네트워크 장애 발생 시 이를 해결하기 위한 Trouble Shooting 접근을 체계적으로 수행할 수 있도록 도와준다.

**Network Model**

# OSI 7 Layer Model vs TCP/IP Model

Chapter. 02 TCP/IP 프로토콜



## OSI 7 Layer



- OSI 목표는 전 세계적인 internetwork에서 사용할 Protocol suite 기반을 마련하는 것이었음.
- 하지만 INTERNET과 TCP/IP Protocol이 인기를 끌면서 TCP/IP에 자리를 내줬다.
- 하지만 OSI 모델 자체는 OSI Protocol 뿐만 아니라 네트워킹 전반적인 동작을 설명하는 도구로 자리잡았음.
- 교육 및 기타 Protocol Suite 구성 요소와 하드웨어 장비 간의 상호 작용을 설명 시 도움이 된다.

## TCP/IP



- INTERNET 표준 Protocol인 TCP/IP Protocol Suite는 OSI모델이 나오기 전에 개발됐다. 그래서 TCP/IP 개발자는 구조를 설명하기 위해 OSI 모델을 사용하지 않았고, TCP/IP 고유한 모델을 만들었으며 이는 TCP/IP 모델, DoD 모델 등으로 불린다.
- TCP/IP와 OSI 모델은 네트워크 기능을 동일한 방법으로 분리하지는 않지만 매우 유사하다. OSI 모델이 널리 쓰이기 때문에 TCP/IP 구조를 그에 대응되는 OSI 계층을 이용하여 설명하는 경우가 많다.

# OSI 7Layers

What is the OSI 7 Layer?

# Introduction of OSI 7 Layer Model

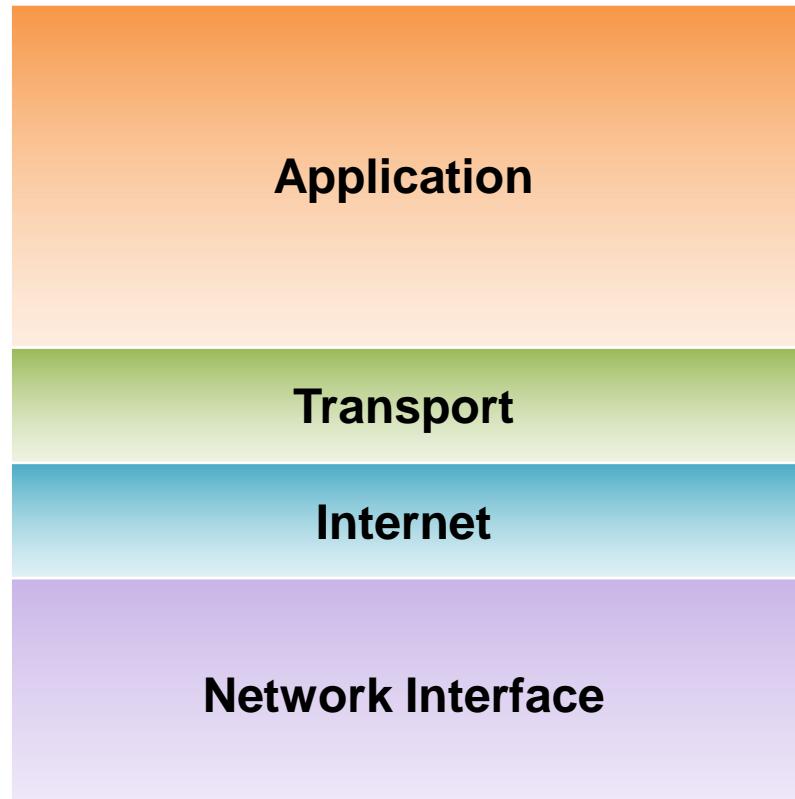
Chapter. 02 TCP/IP 프로토콜



**OSI Model**

7	<b>Application</b>
6	<b>Presentation</b>
5	<b>Session</b>
4	<b>Transport</b>
3	<b>Network</b>
2	<b>Data-Link</b>
1	<b>Physical</b>

**TCP/IP Model**





1

TCP/IP 기술이 모델의 어떤 부분에 해당하는지 이해하기 어려운 경우가 존재. 많은 프로토콜은 OSI 모델을 염두에 두고 설계되지 않았기 때문에 OSI 계층에 정확히 들어맞지 않을 수 있다. 일부 프로토콜은 두 개 이상의 계층에 걸쳐 있을 수도 있으며 두 프로토콜이 하나의 계층을 공유할 수도 있다.

2

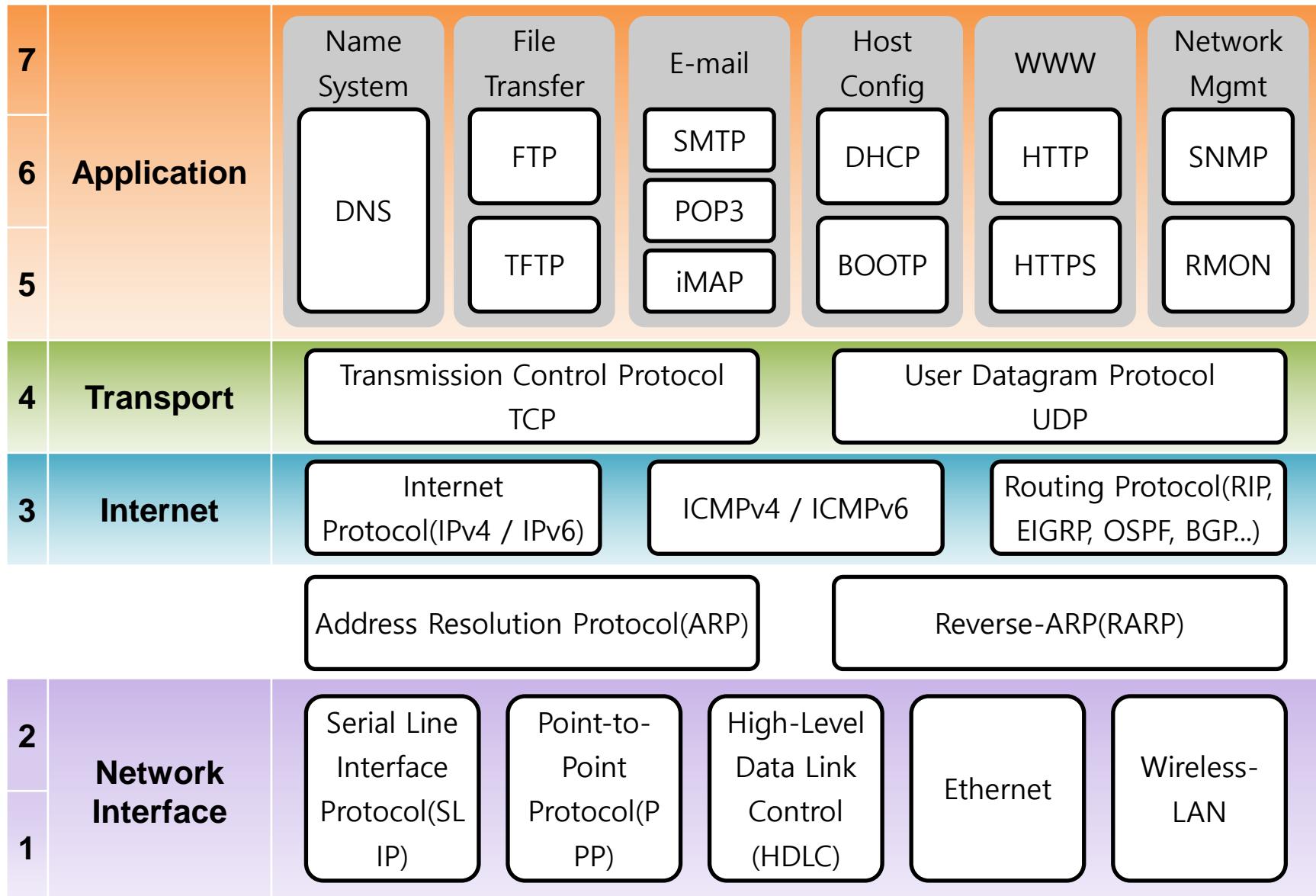
상위 계층(Session, Presentation, Application) 사이의 경계는 특히 모호하다. 일부 프로토콜은 이들 계층 중 하나에 정확히 들어맞지만 이들 계층에 중첩되는 여러 프로토콜이 존재한다. 따라서 많은 문서에서는 상위 수준 프로토콜을 계층으로 분류하지 않는다.

3

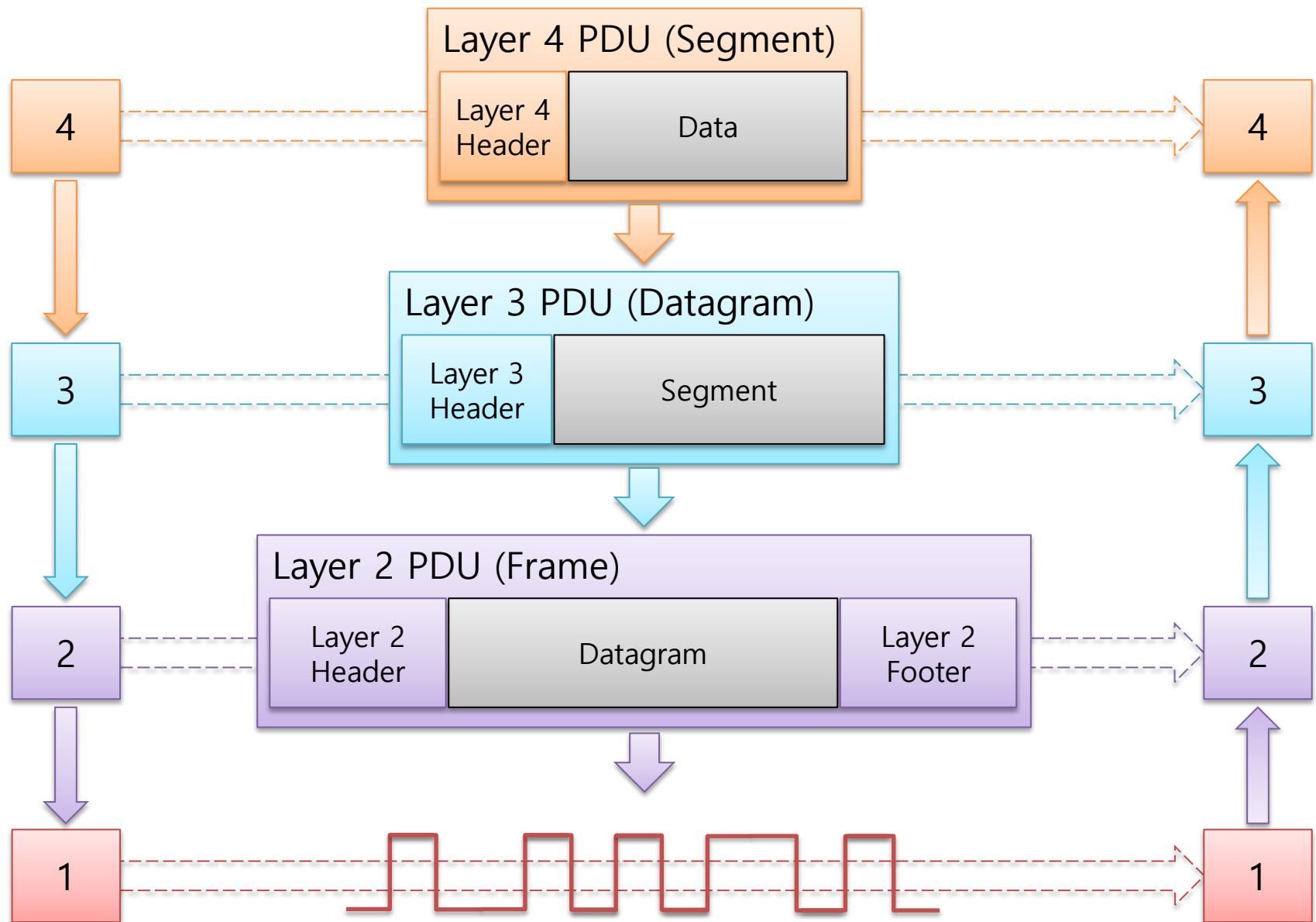
제품을 설계하는 사람들은 자신의 최신 제품이 OSI 모델의 특정 계층만을 구현했는지에 대해 신경 쓰지 않는다. 때문에 새로 나온 제품이 관습을 깨고 기존에 여러 계층의 개별 장비에서 수행하던 기능을 구현하는 경우가 있다.

# OSI 7 Layer Model vs TCP/IP Model

## Chapter. 02 TCP/IP 프로토콜

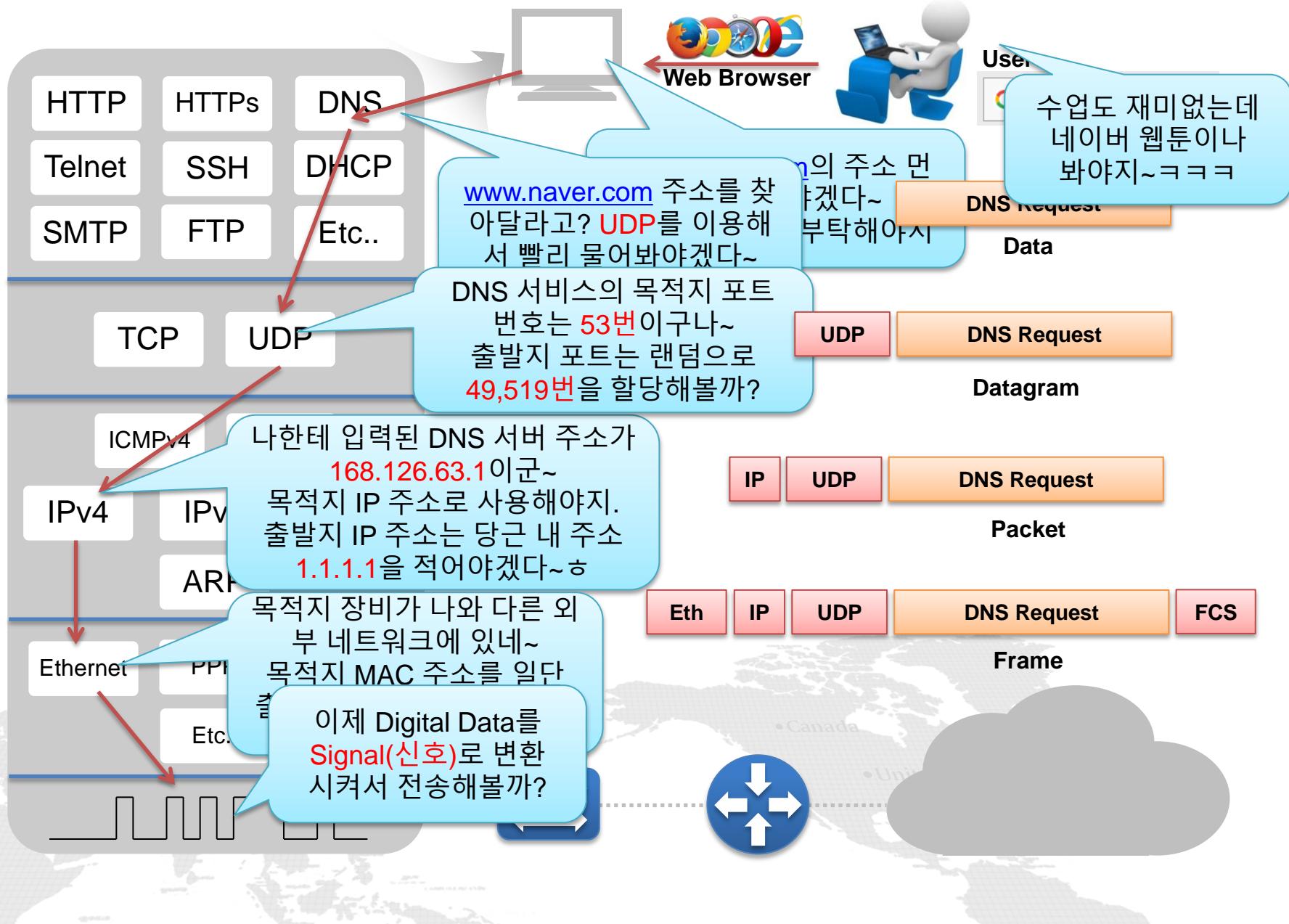


# Encapsulation & De-encapsulation



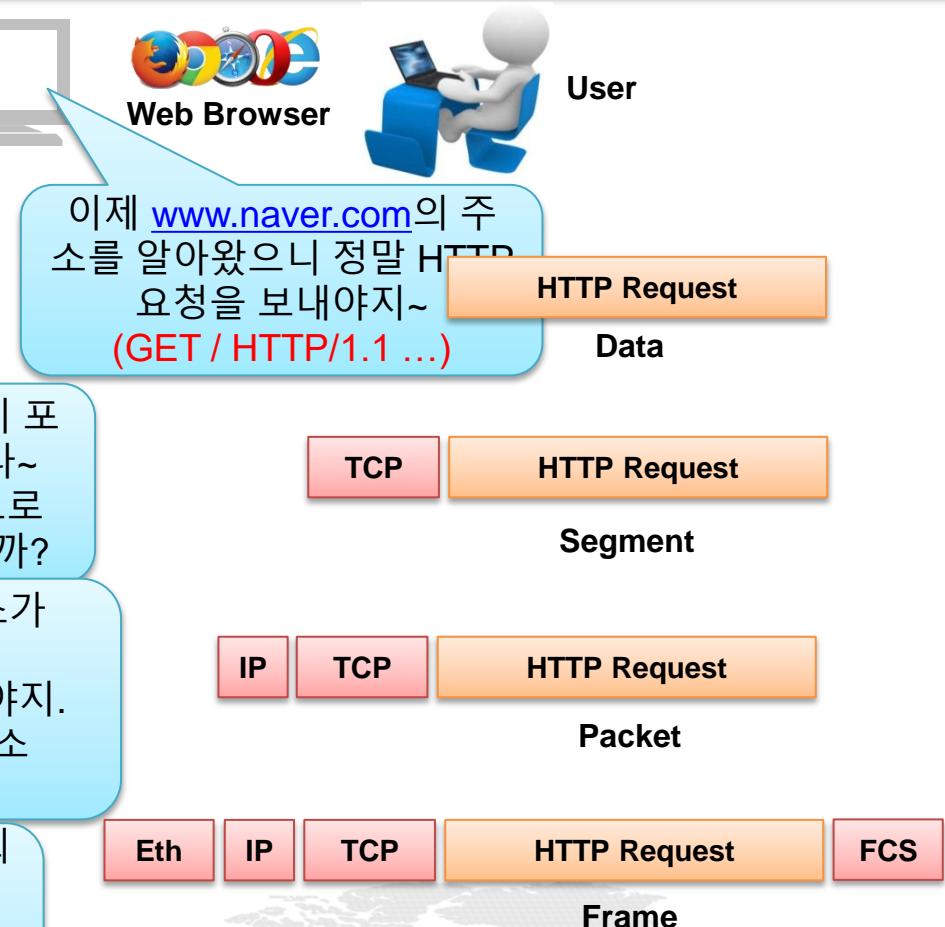
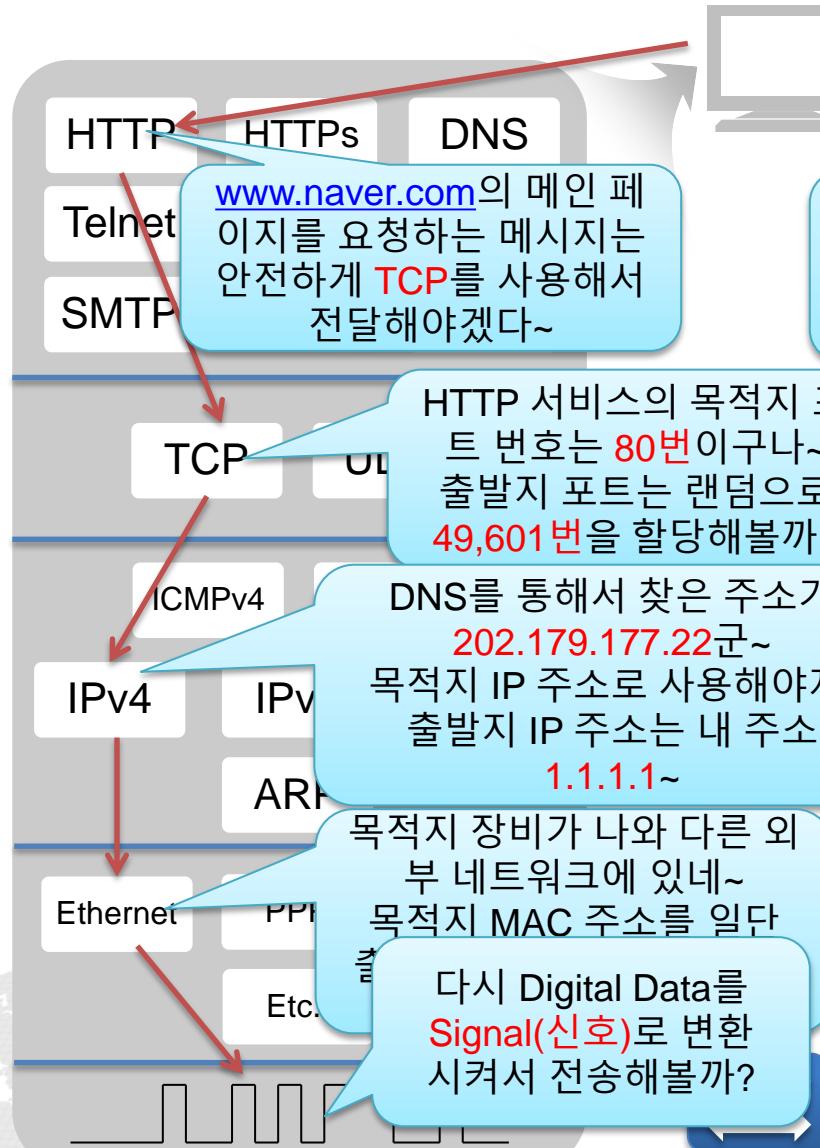
# Data Encapsulation

## Chapter. 02 TCP/IP 프로토콜



# Data Encapsulation

## Chapter. 02 TCP/IP 프로토콜



# Data De-encapsulation

Chapter. 02 TCP/IP 프로토콜



Web Server Application

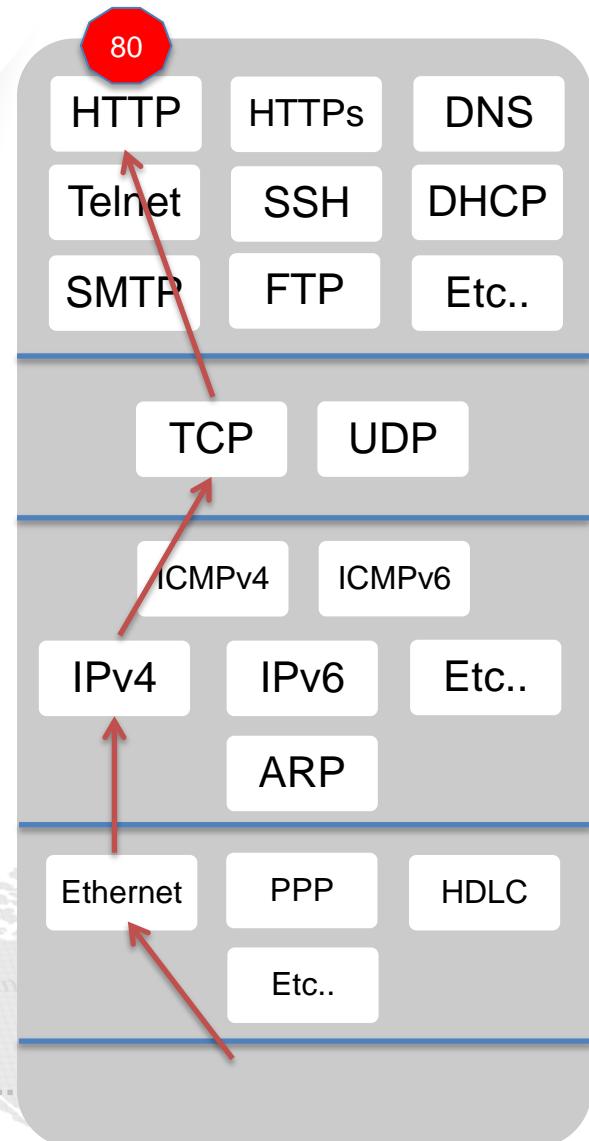
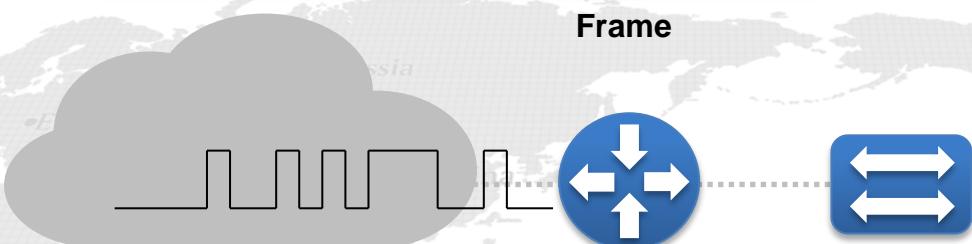


HTTP Request  
Data

TCP      HTTP Request  
Segment

IP      TCP      HTTP Request  
Packet

Eth      IP      TCP      HTTP Request      FCS  
Frame



# 02

## Transport Layer

What will they do on Transport layer?

Application Layer는 사용자가 인터넷을 편하고 효과적으로 사용할 수 있도록 인터페이스를 제공하며, 사용자는 그 인터페이스를 통해서 원하는 데이터를 생성한다. Transport Layer는 사용자가 생성한 데이터를 어떤 방법을 통해서 상대방의 Application Layer에 효과적으로 전송 할 것인지 결정하는 계층이다.



# Introduce Transport Layer

## Chapter. 02 Transport Layer



영어로 Transport라는 단어는 ‘수송하다, 운반하다’라는 의미가 있습니다. 이 뜻을 통해서 알 수 있듯이 트랜스포트계층의 역할은 ‘데이터를 상대에게 전달하는 것’입니다. ‘상대 컴퓨터에 전달하면 그것으로 끝’인 것은 아닙니다. ‘상대 애플리케이션 계층에 있는 어떤 서비스에게 전달’할지 까지 책임을지고 전달해야 합니다.

A

**Port Address** 애플리케이션 계층에는 사용자의 인터넷 사용의 편의성을 제공하기 위해서 여러가지 서비스가 존재합니다. 이런 서비스를 Transport Layer에서 구분하기 위해서 Port Address가 필요합니다



B

TCP(Transmission Control Protocol)는 대용량 데이터를 나누어서 전송할 때 ‘안전하고 확실하게 전달’하기 위해서 만들어진 프로토콜이다. 데이터가 도중에 손실되거나 뭔가 사정이 있어 상대에게 전달 되지 못했을 때 재전송하는 기능이 있습니다.



C

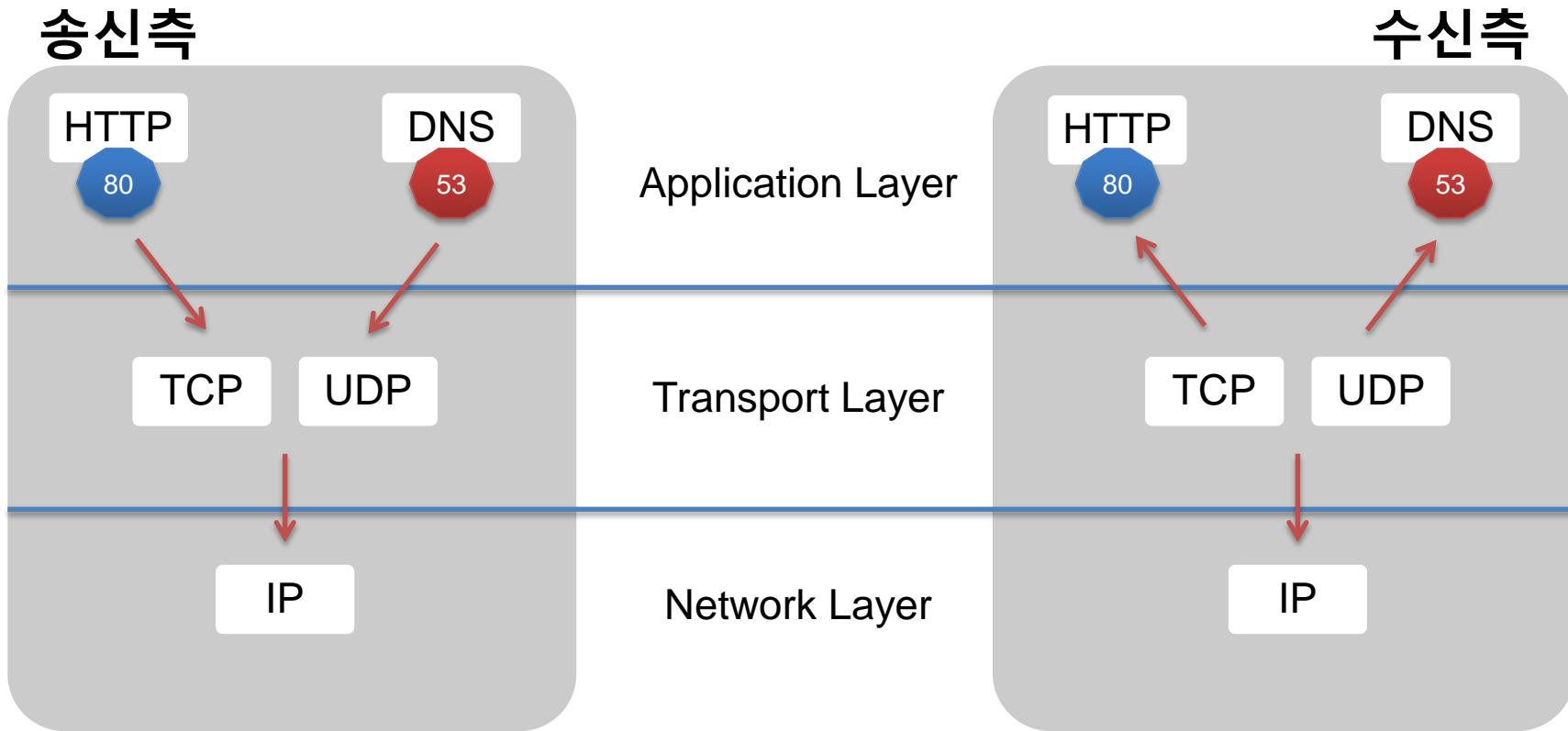
UDP



UDP(User Datagram Protocol)는 TCP 프로토콜과는 반대로 소량의 데이터를 빠르게 전송하기 위해서 만들어진 프로토콜이다. 문제가 발생한 부분에 대해서 재전송하는 기능은 없지만 실시간(IPTV/IP전화)으로 데이터를 전송하는 경우에 적합하다.

# Rule of Transport Layer

Chapter. 02 Transport Layer



## Transport Layer의 위치

- Transport Layer(트랜스포트 계층)는 애플리케이션 층과 네트워크 층의 중계 역할을 담당한다.
- Transport Layer는 데이터가 항상 확실하게 상대방에게 전달 된다고는 보장 할 수 없다. 중간에 문제가 발생하여 데이터가 손실 될 수 있으므로 뭔가 대처 방안이 필요하다. 이 때 각 서비스에 맞는 방법으로 그것을 처리하는 것이 트랜스포트 층의 역할이다.

**전송 계층의 위치**

# TCP Reliability vs UDP Speed

Chapter. 02 Transport Layer

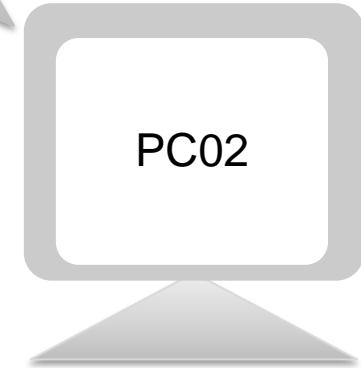


## TCP 속도가 느려



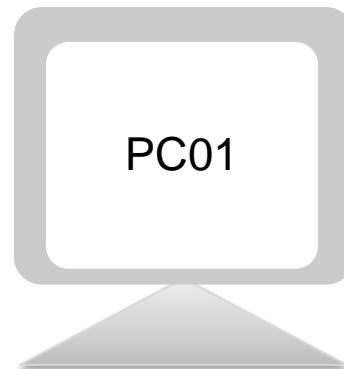
자! 이제  
Data를 전송  
해 볼까

Hi~ 준비 되셨나요? TCP



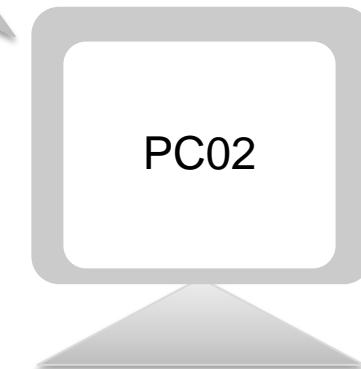
오~데이터를  
받을 준비를  
해야지

## TCP 신뢰성 제공



Data를 잘 받  
았나요?

자!~Data#1받으세요 ! TCP



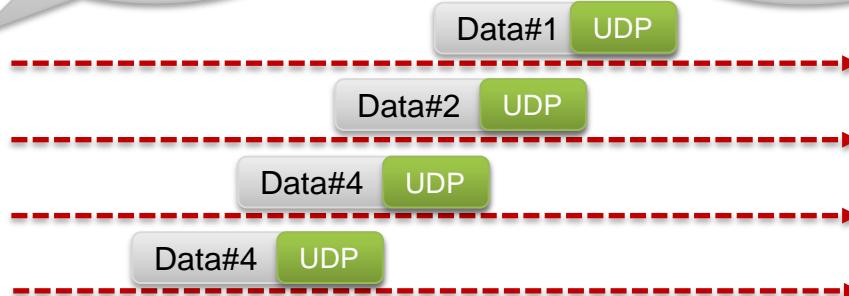
데이터 손실  
발생 했구나  
다시 요청해  
야지.

# TCP Reliability vs UDP Speed

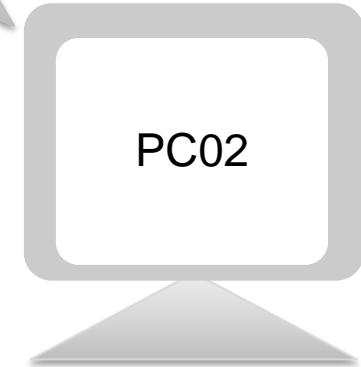
Chapter. 02 Transport Layer



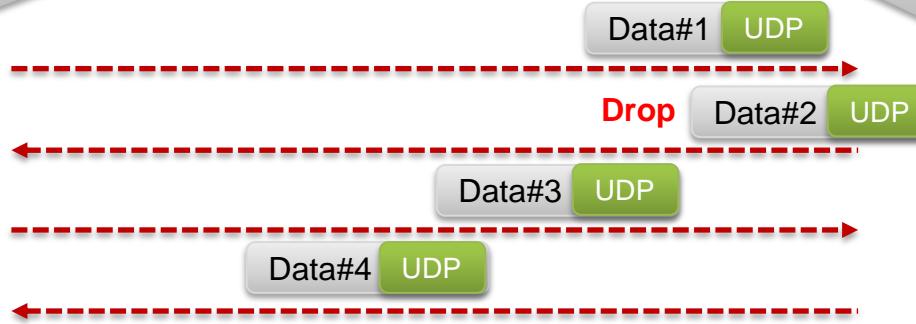
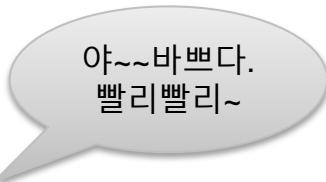
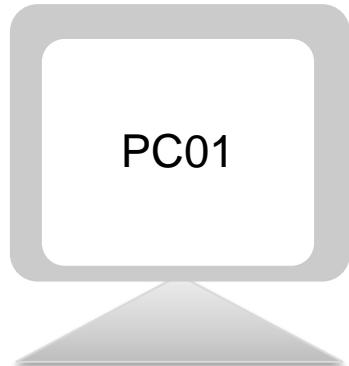
UDP 속도가 빠름



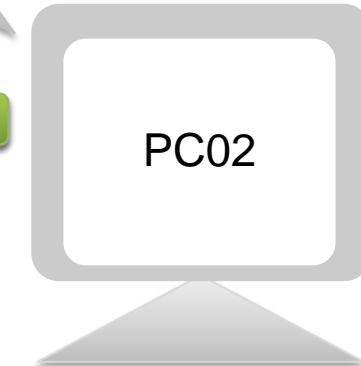
오~좋아 정  
말 빠르다



UDP 신뢰성 없음

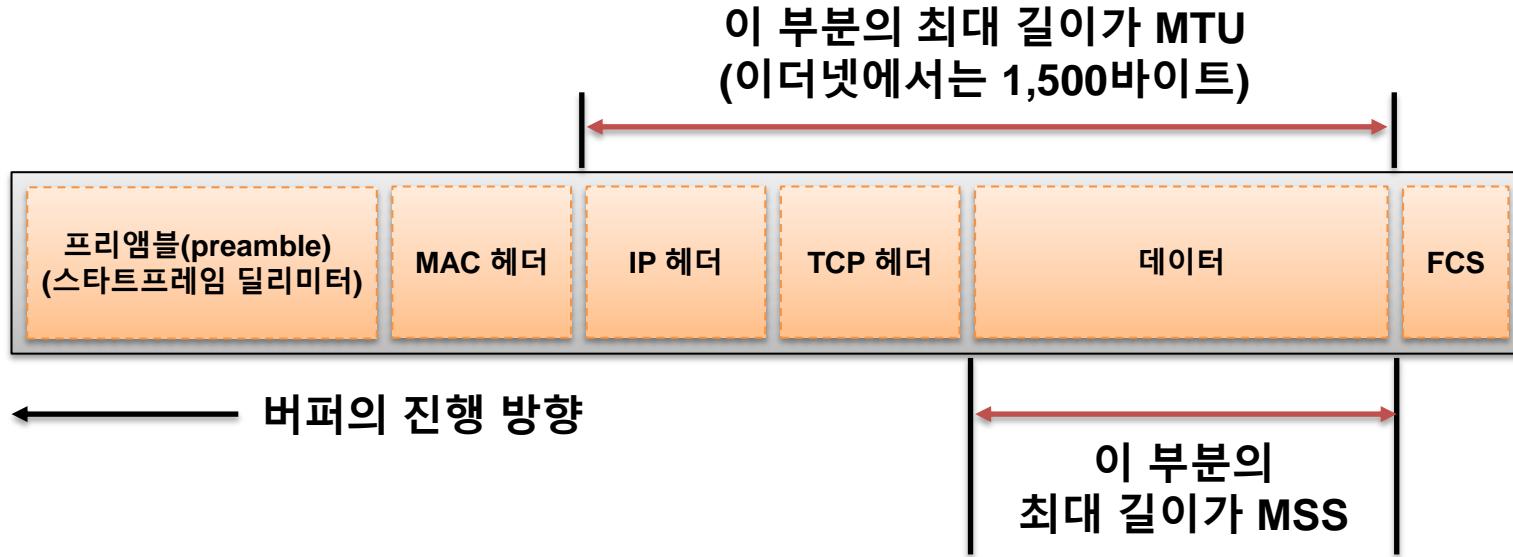


빠르긴 한데  
Data#2.....  
실패도



# Data Segmentation

## Chapter. 02 Transport Layer



### MTU / MSS

- MTU(Maximum Transmission Unit) : 패킷 한 개로 운반할 수 있는 디지털 데이터의 최대 길이, 이더넷에서는 보통 1,500Byte
- MSS(Maximum Segment Size) : L4 Header를 제외하고 한 개의 패킷으로 운반 할 수 있는 TCP의 데이터 최대 길이

최대 전송 단위

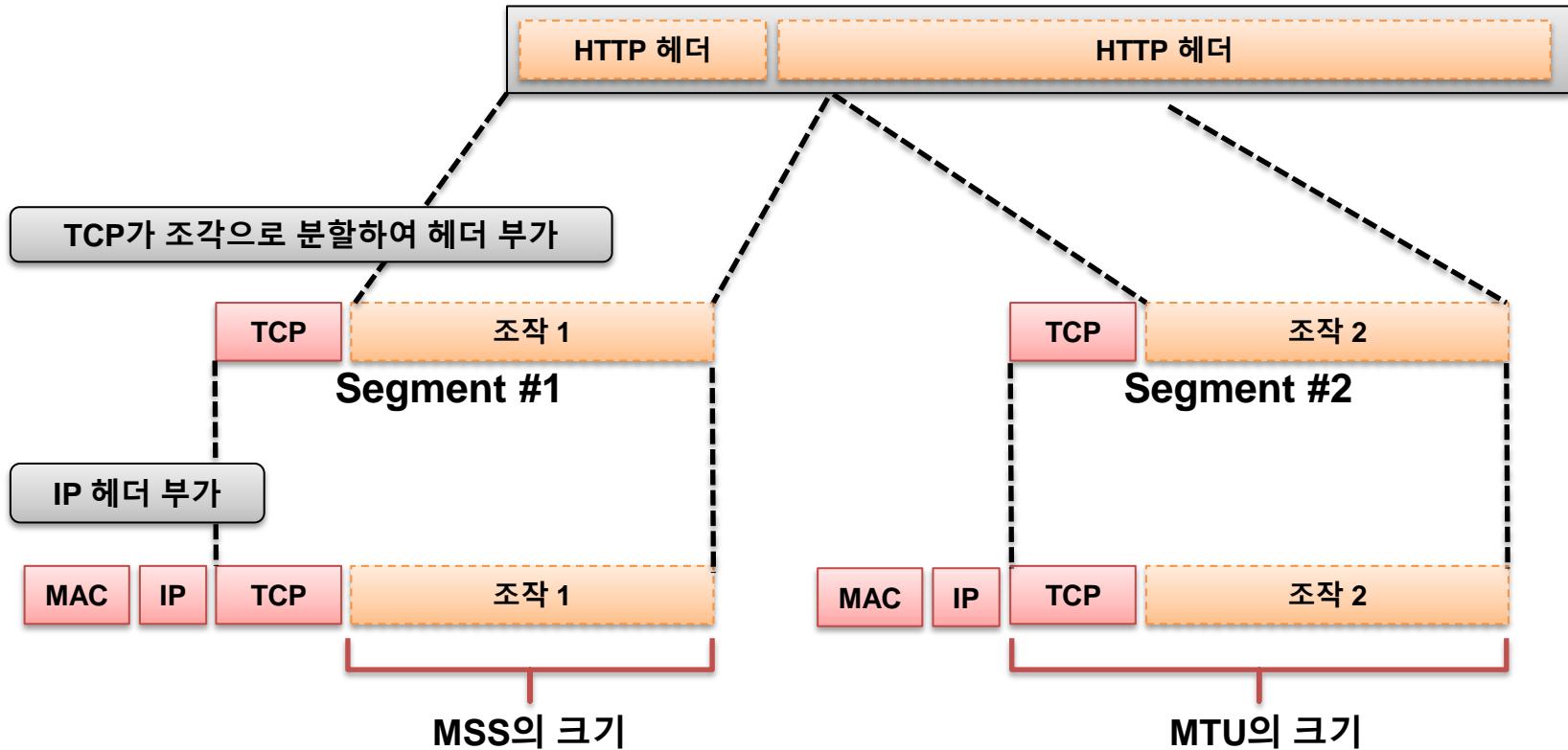
# Data Segmentation

## Chapter. 02 Transport Layer



← 패킷의 진행 방향

애플리케이션의 데이터

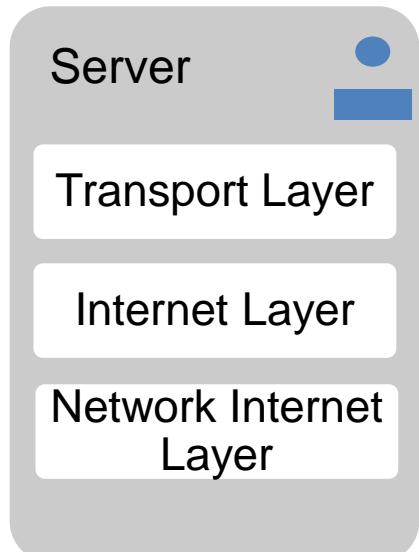
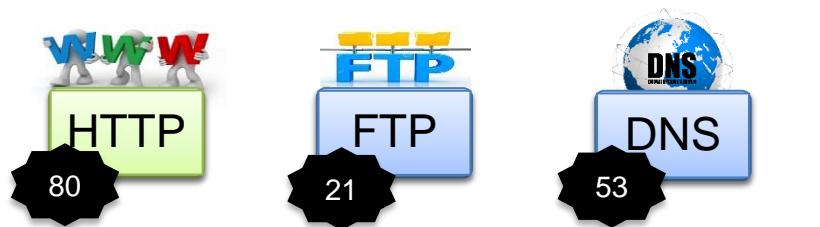


### Segmentation

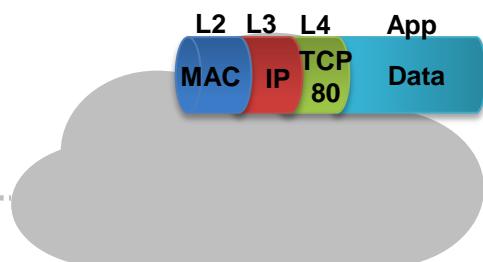
- 애플리케이션의 데이터를 분할해서 전송합니다. 대개 애플리케이션의 데이터는 너무 크므로 TCP가 IP패킷 바구니에 넣는 크기로 분할 합니다.

분할

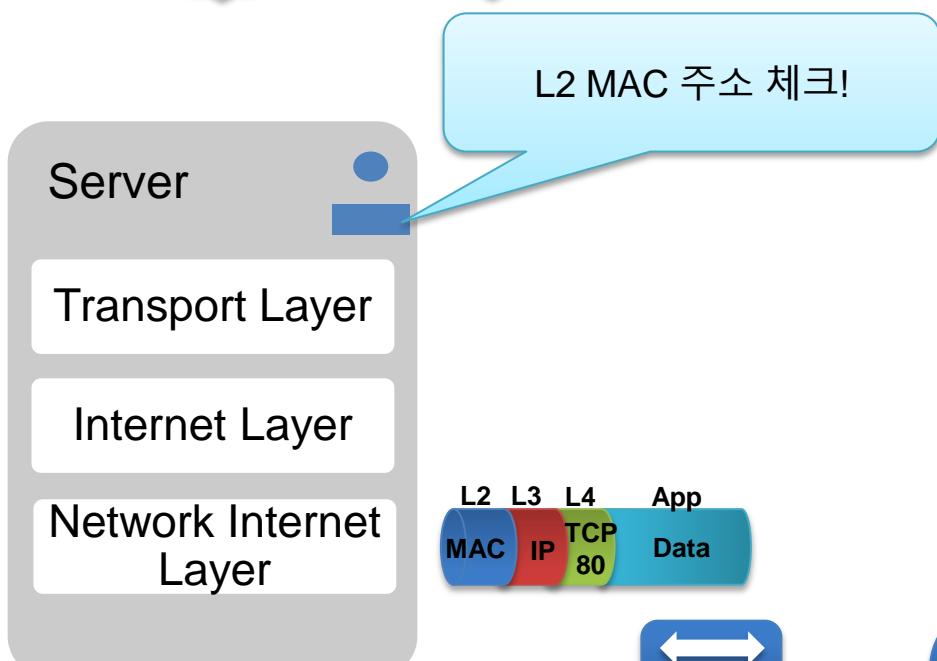
# Application Identity



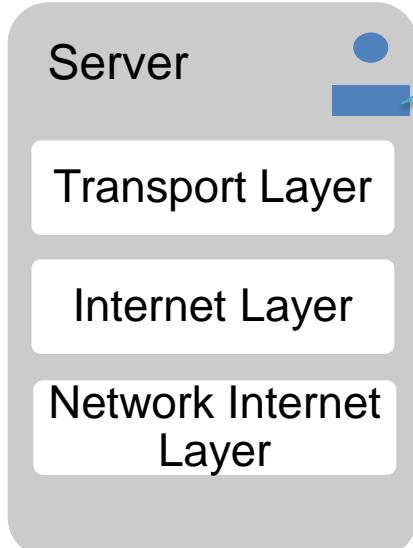
자 ~ 이제 처  
리해 볼까



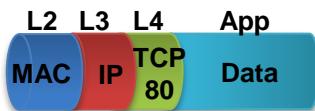
# Application Identity



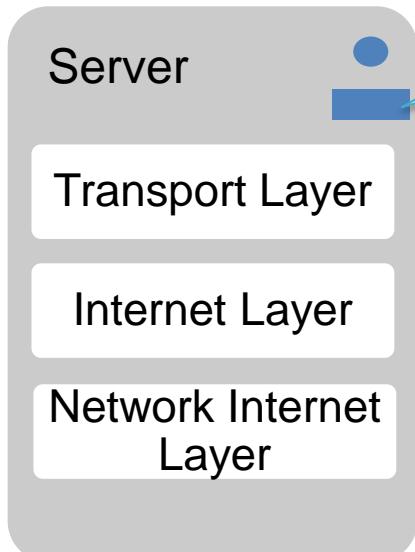
# Application Identity



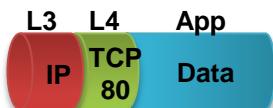
자기 MAC 주소가 맞으면  
L2 헤더 제거하고 상위계  
층으로 전달한다.



# Application Identity



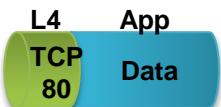
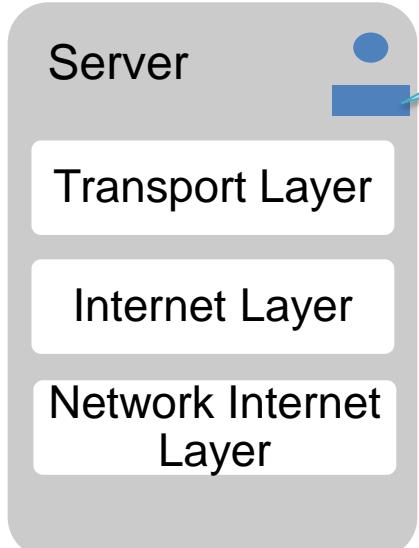
자기 IP 주소가 맞으면 L3 헤더를 제거하고 상위계층으로 전달한다.



# Application Identity



L4 헤더에 80 포트 주소를  
확인하고 L4 헤더 제거 후  
웹 서비스로 데이터 전달





인터넷 할당 번호 관리 기관(Internet Assigned Number Authority, IANA)은 포트 번호를 세 개의 범위로 나누어서 관리하고 있다.  
참조 사이트 : <http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml>

## 1 Well Known Port : 1 ~ 1,023

이미 널리 알려진(Well-Known) 포트를 Well Known Port 이라고 하며 이는 Server 측에 각 용도별로 예약되어 동작되고 있으며 이를 사용하게되는 클라이언트는 보통 임시 포트 번호를 이용하여 접속하게 된다.

## 2 Registered Port : 1,024 ~ 49,151

이 포트 범위를 가지고 IANA에 의해 할당되거나 또는 통제를 받지 않는다. 중복 방지를 위해서 단지 IANA에 등록만 되어 있다.

## 3 Dynamic and Private Port : 49,152 ~ 65,535

이 포트 범위는 통제도 안 되고 등록도 되어 있지 않다. 어떠한 프로세스에 의해 서도 사용이 가능한데, 이들을 임시 포트라고 이야기 한다.

# Well Known port number



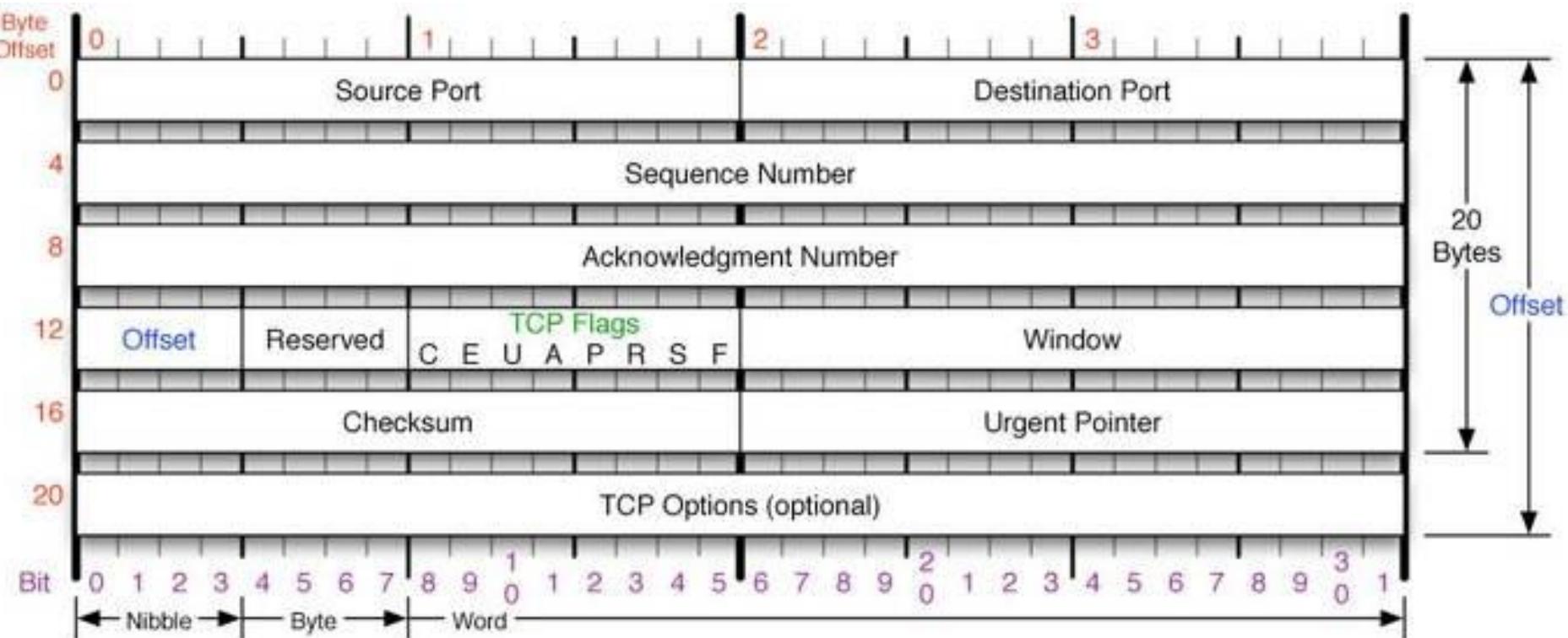
각 운영체제는 Well Known 서비스 포트에 대한 정보를 /etc/services 파일에 정의 되어 있다.

Windows : c:\windows\system32\drivers\etc\

Linux : /etc/services

서비스 명	어플리케이션	포트번호	전송 프로토콜
WWW	HTTP	80	TCP
WWW 보안접속	HTTPs	443	TCP
전자메일(송신)	SMTP	25	TCP/UDP
전자메일(수신)	POP3	110	TCP/UDP
파일전송	FTP	20-21	TCP
원격 관리	Telnet	23	TCP
원격 관리 보안 접속	SSH	22	TCP
도메인 관리 시스템	DNS	53	TCP/UDP
IP 관리 시스템	DHCP	67/68	UDP
네트워크 관리 시스템	SNMP	161/162	UDP

# TCP (Transmission Control Protocol)



# TCP (Transmission Control Protocol)

Chapter. 02 Transport Layer



## Source Port (16비트)

송신측의 포트 번호를 기록

예) 80 -> 0000 0000 0101 0000

## Destination Port (16비트)

수신측의 포트 번호를 기록

예) 80 -> 0000 0000 0101 0000

## Sequence Number(32비트)

전체 데이터 중 이 데이터가 몇 번째에 해당하는지를 기록합니다.

## Acknowledgment Number(32비트)

다음에 받을 데이터가 전체 데이터 중 몇 번째 데이터인지를 기록합니다.

### Offset

TCP 헤더 크기

### Reserved

현재는 사용하지 않습니다.

### TCP Flags(6비트)

U	A	P	R	S	F
G	C	S	S	Y	I
R	K	H	T	N	N

### Windows Size(16비트)

수신 가능한 데이터 크기를 기록합니다.

### Checksum

데이터가 무사한지 아닌지를 확인하기 위한 값을 기록합니다.

### Urgent Pointer

URG 플래그가 1인 경우 사용합니다.

### TCP Options

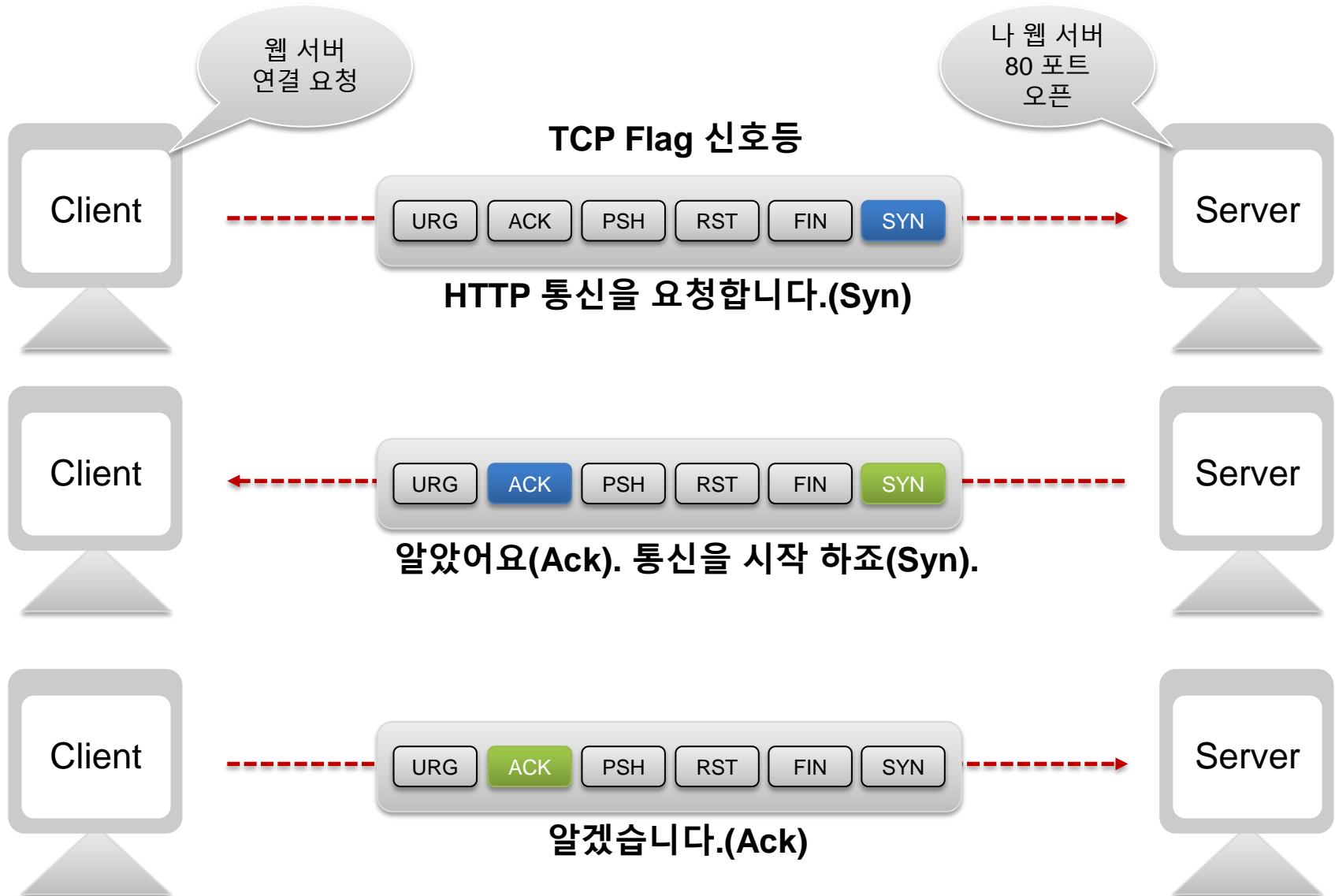
TCP의 기능을 확장할 때 사용합니다.  
(세그먼트 사이즈를 정할 때 등.)

### 패딩

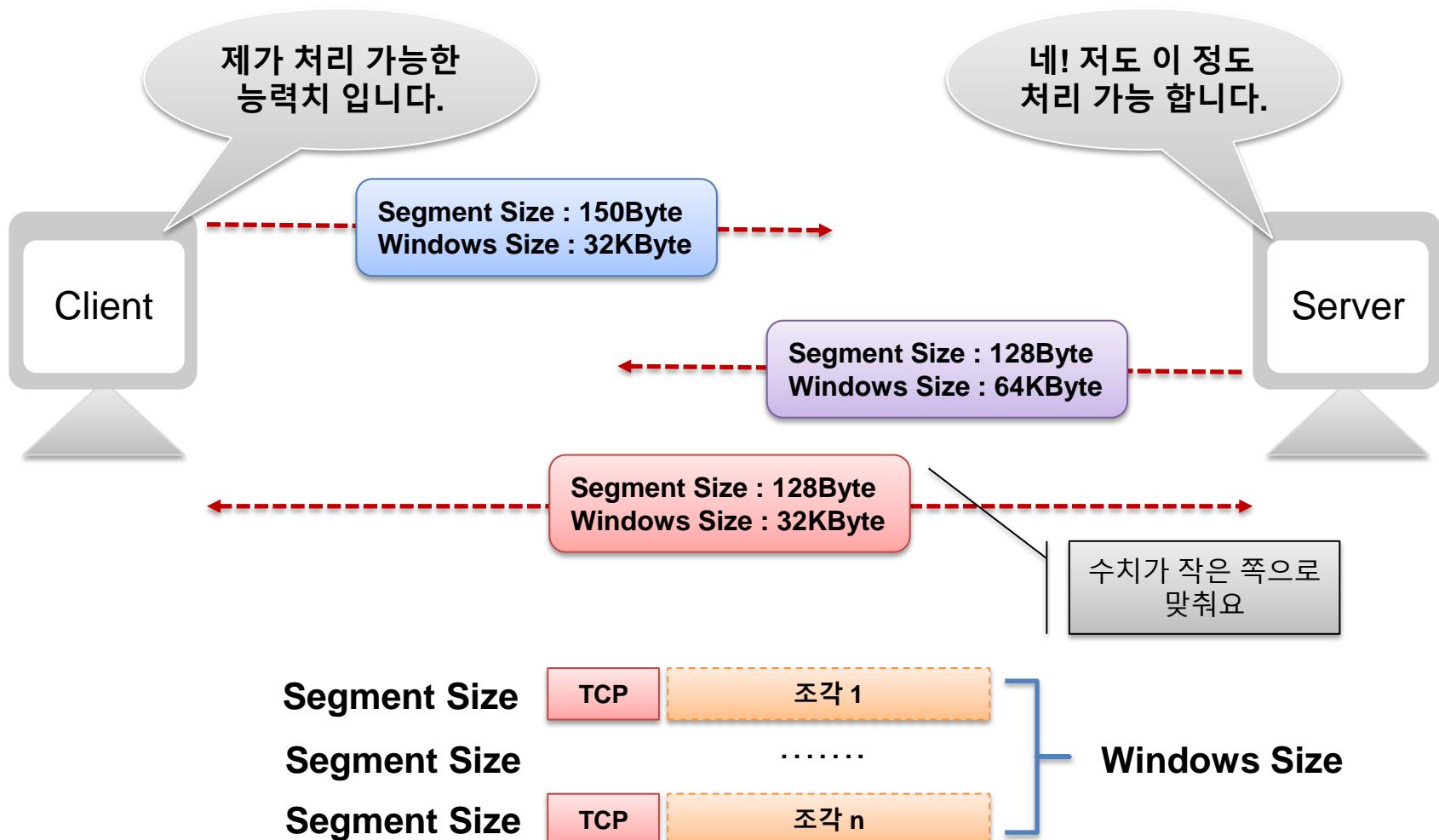
헤더가 32비트의 정수배가 되지 않을 때 0을 덧붙여서 헤더의 크기를 조정합니다.

# TCP 3Way Handshake

## Chapter. 02 Transport Layer

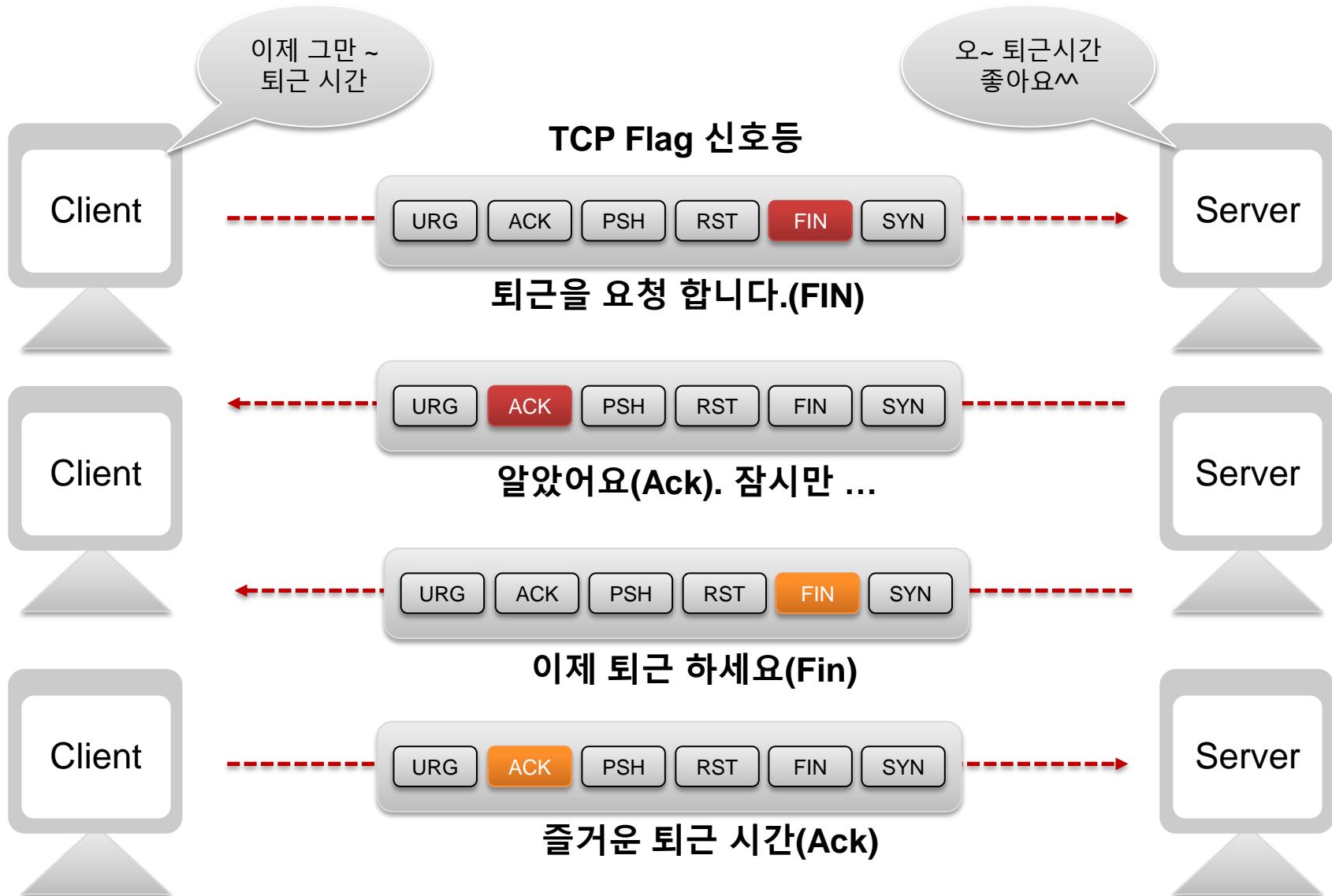


# TCP 3Way Handshake

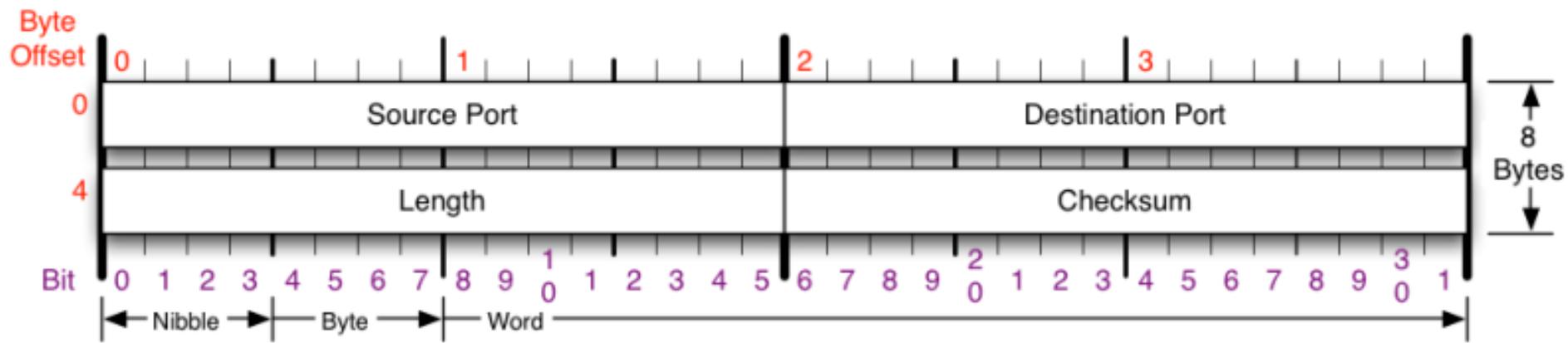


# TCP 4Way Handshake

## Chapter. 02 Transport Layer



# UDP (User Datagram Protocol)



# UDP (User Datagram Protocol)

## Chapter. 02 Transport Layer



<b>Source Port (16비트)</b> 송신측의 포트 번호를 기록 지정하지 않는 경우는 모두 0으로 채워진다.	<b>Destination Port (16비트)</b> 수신측의 포트 번호를 기록 예) 80 → 0000 0000 0101 0000
<b>Length(16비트)</b> 헤더와 데이터의 합계가 몇 바이트인지를 기록합니다.	<b>CheckSum(16비트)</b> 데이터가 무산한지 아닌지를 확인하기 위한 값을 기록합니다.



# 03

## Network Layer

Network layer can do many things

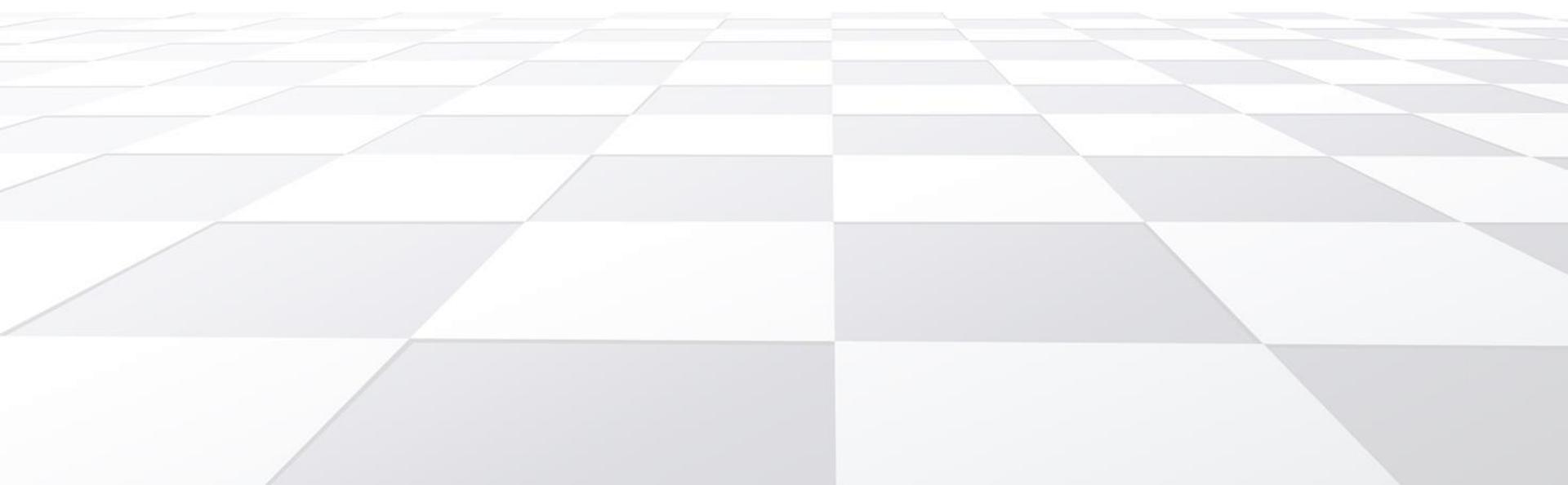
Network 계층에서는 서로 다른 네트워크를 구분하고 연결해주는 역할을 하며, 네트워크를 넘나드는 트래픽을 관리하고 필터링해주는 등 많은 역할을 수행하고 있다.

이러한 역할들을 수행하기 위해 필요한 프로토콜 몇 가지를 먼저 살펴보도록 하자.



# Internet Protocol

Internet Protocol version 4





**127.0.0.1**



**A Network**

종로구 효자동  
광일빌딩



**192.168.10.23**



**B Network**

종로구 연지동  
리도빌딩



**172.30.16.30**



물건을 전달하고자 소포를 보낼 때 사용하는 주소

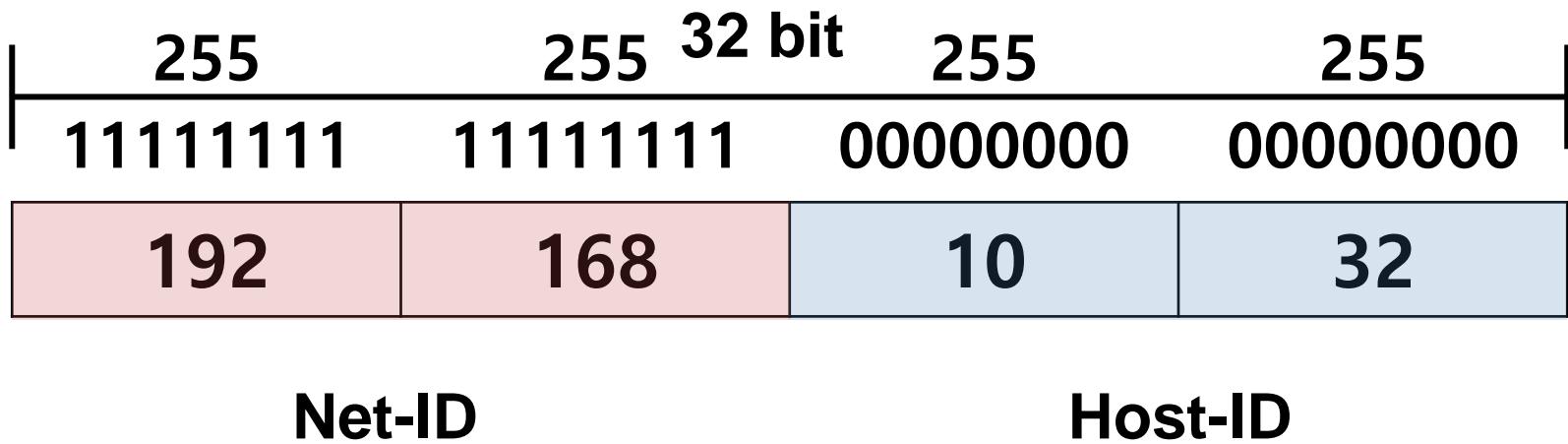
# 종로구 효제동 광일빌딩



Data를 전달하기 위한 통신에 사용하는 주소

**192.168.10 .32**

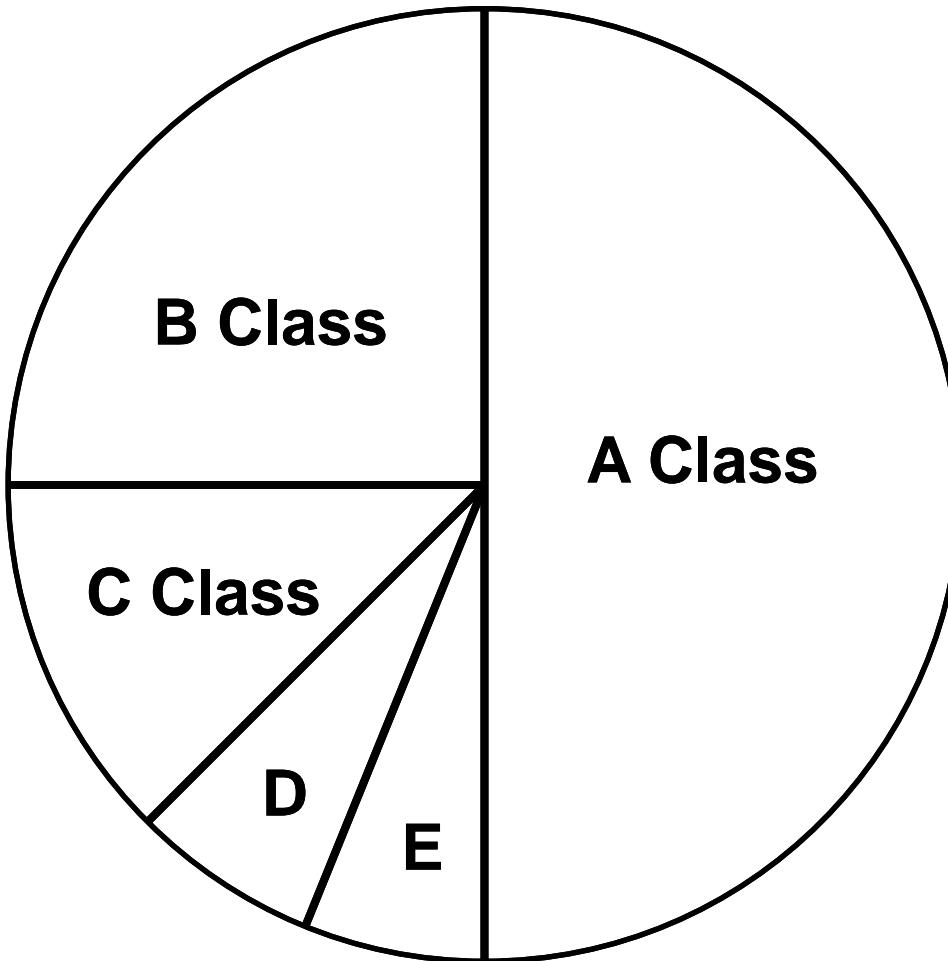


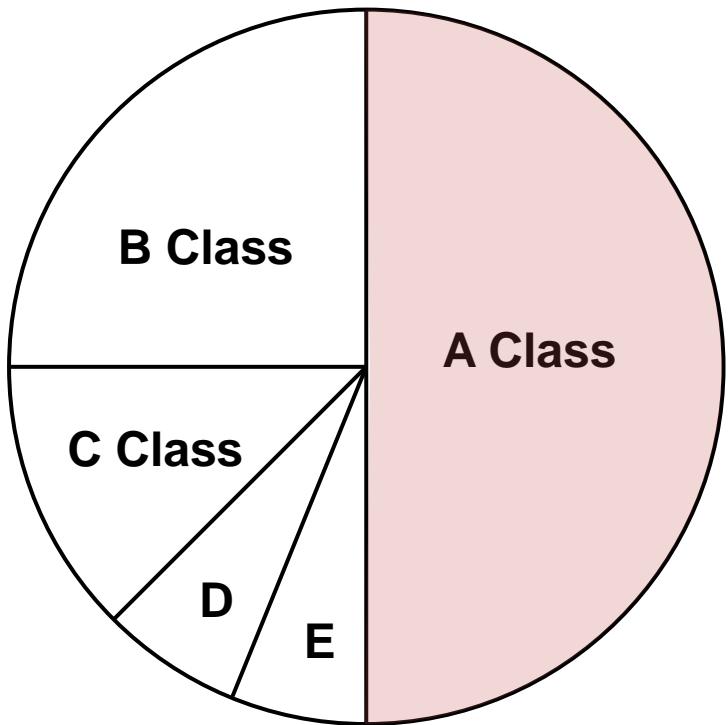


**Subnet-mask** : Net-ID와 Host-ID를 식별해주는 값

1은 Net-ID를 표시하고  
0은 Host-ID를 표시한다

# IPv4 Address Class

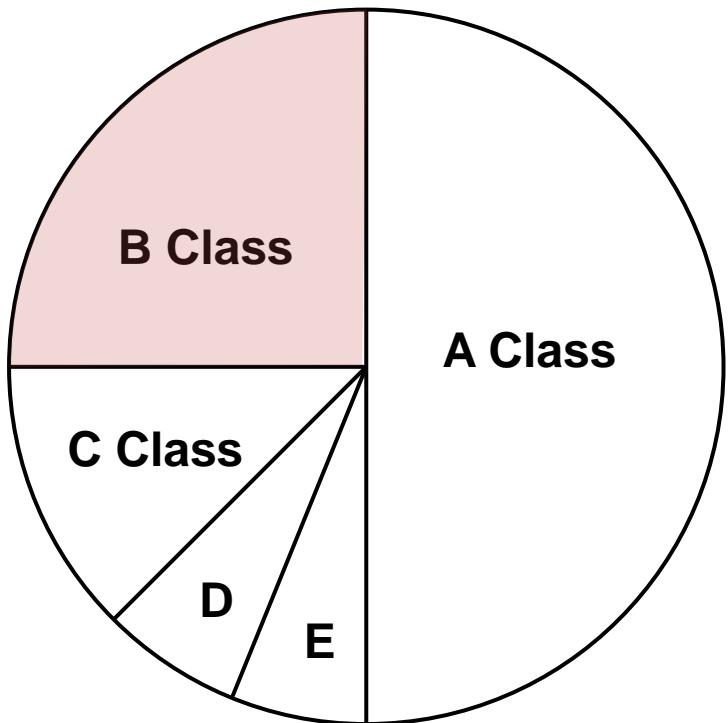




첫번째 옥텟의 bit만을 사용하여 구분하고  
Class별 사용 IP의 범위를 나눈다.

0 0 0 0 0 0 0 0

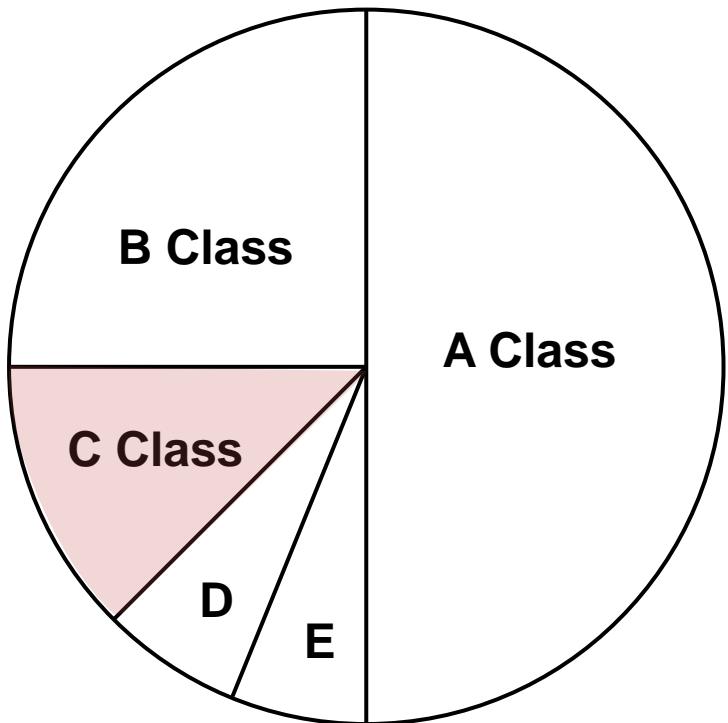
8bit( $2^8=256$ )중 7bit( $2^7=128$ )를 사용하여  
첫번째 옥텟의 숫자가 **0~127**인 IP주소를  
A class의 범위로 사용한다.



첫번째 옥텟의 bit만을 사용하여 구분하고  
Class별 사용 IP의 범위를 나눈다.

1 0 0 0 0 0 0 0

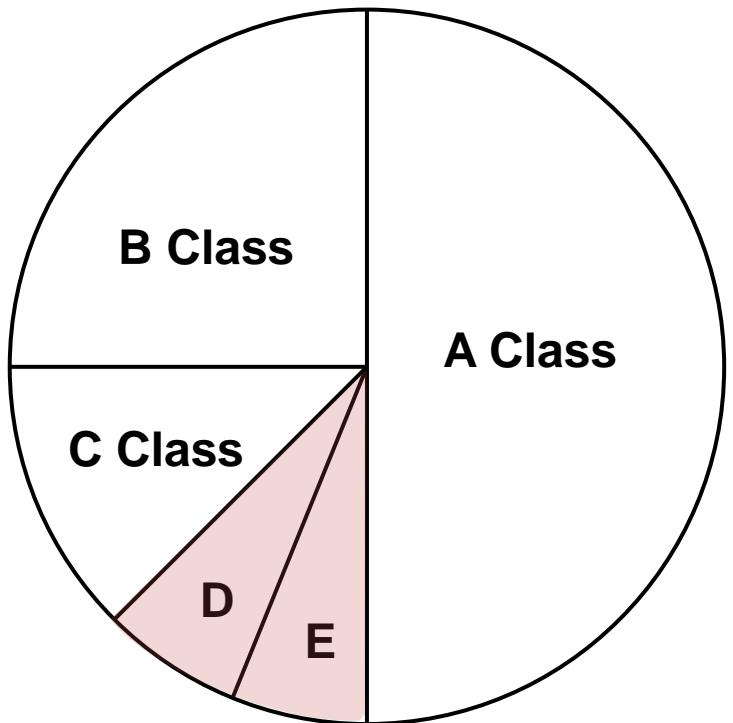
8bit( $2^8=256$ )중 6bit( $2^6=64$ )를 사용하여  
첫번째 옥텟의 숫자가 **128~191**인 IP주소를  
B class의 범위로 사용한다.



첫번째 옥텟의 bit만을 사용하여 구분하고  
Class별 사용 IP의 범위를 나눈다.

1 1 0 0 0 0 0 0

8bit( $2^8=256$ )중 5bit( $2^5=32$ )를 사용하여  
첫번째 옥텟의 숫자가 **192~223**인 IP주소를  
C class의 범위로 사용한다.



## D Class 주소 범위

: 224.0.0.0~239.255.255.255

1 1 1 0 0 0 0 0

## E Class 주소 범위

: 240.0.0.0~255.255.255.255

1 1 1 1 0 0 0 0

# IPv4 Address Class

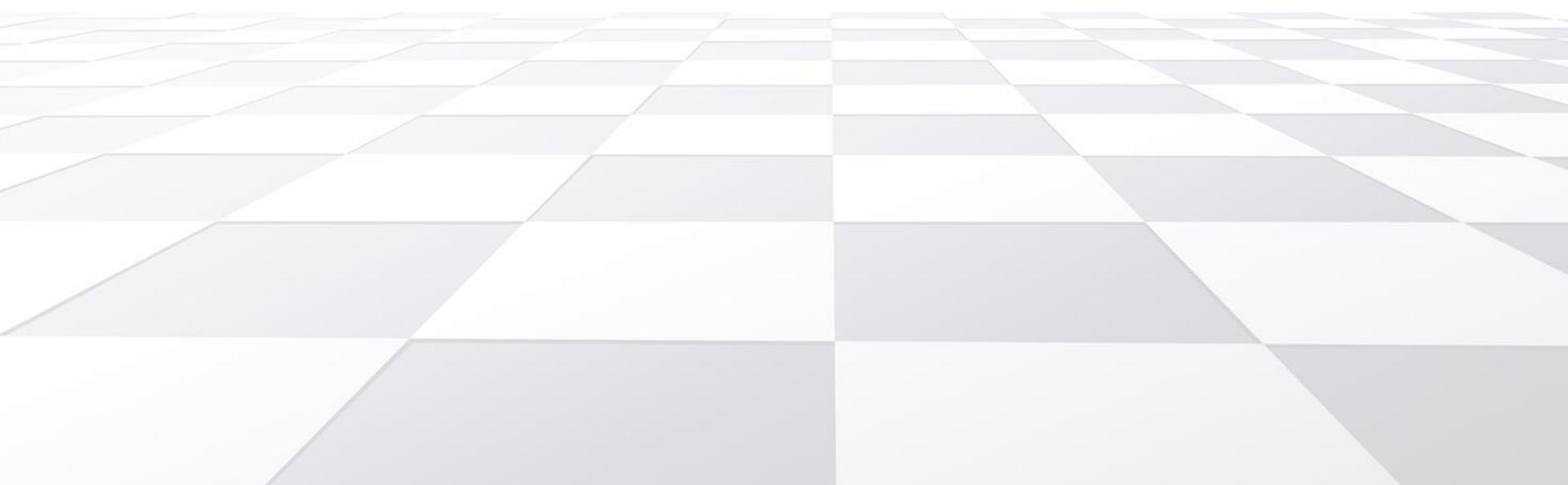


	From	To
Class A	<b>0.0.0.0</b> Net-id Host-id	<b>127.255.255.255</b> Net-id Host-id
Class B	<b>128.0.0.0</b> Net-id Host-id	<b>191.255.255.255</b> Net-id Host-id
Class C	<b>192.0.0.0</b> Net-id Host-id	<b>223.255.255.255</b> Net-id Host-id
Class D	<b>224.0.0.0</b> Multicast Address	<b>239.255.255.255</b> Multicast Address
Class E	<b>240.0.0.0</b> Reserved	<b>255.255.255.255</b> Reserved



# IP Subnetting

Classless Inter-network Domain Routing





## STEP.1 기준 구하기

네트워크를 나누고자 할 때,  
Host 기준으로 나눌 건지 Network 기준으로 나눌 건지 정한다.

## STEP.2 필요한 bit 수

정해진 기준 값에 따라 필요한 bit 수를 구한다.

## STEP.3 Net-ID 구분

잘라진 지점을 기준으로 새로운 Net-ID 범위를 구한다.

## STEP.4 Subnet의 수와 IP주소의 수 파악

나눠진 서브넷의 개수는 몇이고, 각 서브넷 당 몇 개의 IP가 할당되는지 확인한다.

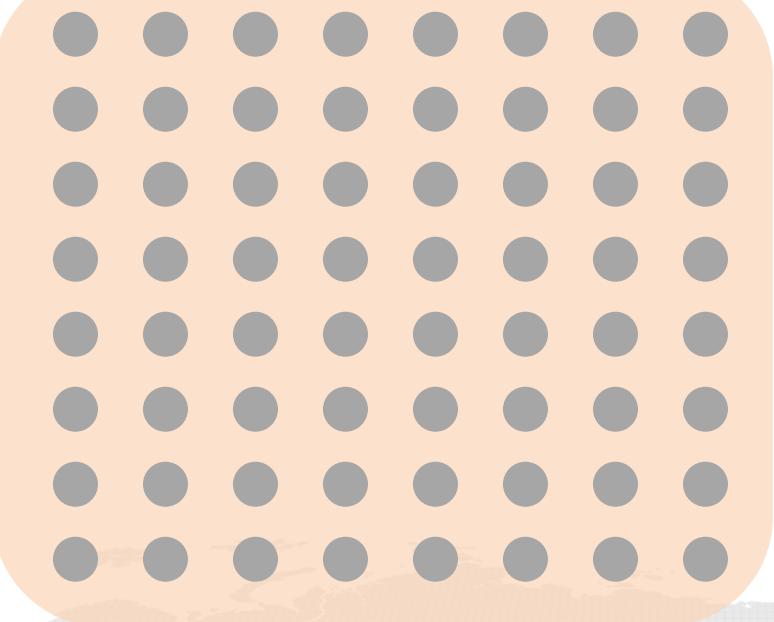
## 각 Subnet의 범위 구하기

나눠진 서브넷 당 네트워크 범위를 구하고  
네트워크 대표주소와 Subnet-mask로 표시한다.

## STEP.6 가용 IP 범위 파악

장비에 할당할 수 있는 가용 IP의 범위를  
파악한다.

# Subnetting



Subnet

Subnet

Subnet

Subnet





## Step 1

### Subnetting 할 때의 기준 정하기

예를 들면 강의실을 증축하려고 하는데 총 몇 개의 강의장을 더 만들 건지, 혹은 한 강의당 몇 명의 학생들이 강의를 듣게 할 것인지에 대해서 결정하는 것과 같다.

강의장을 기준으로 했다면 Network를 기준으로, 학생 수를 기준으로 했다면 Host 기준으로 한 것이 된다.

한 강의장에 PC가 25대가 있다.

200.1.1.0/24 주소를 PC의 수를 기준으로 Subnetting을 하려고 한다면?

① 기준은? Host 25

② 25개 이상의 IP를 표현하기 위해 필요한 bit 수는?

1 bit는 2개의 수를 표현한다.

따라서 bit로 표현할 수 있는 정보의 수는  $2^n$ 으로 구한다.

$$2^n - 2 \geq 25$$
$$n = 5$$

이때 몇 개의 bit가 필요 할까?



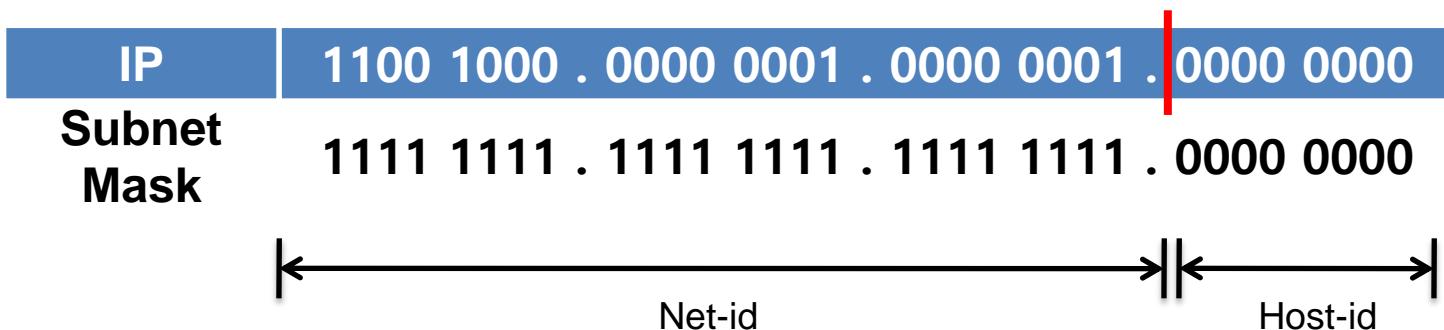
## Step 2

### 주어진 Host-ID에서 Subnetting 하기

기존에 주어진 IP 범위에서 Network가 더 증가한다는 것은 Network-ID로 사용될 bit가 더 필요하다는 뜻이 된다.

따라서 현재 Host-ID의 범위가 어디인지 확인한 후에 필요하다고 판단된 bit까지를 구분하여 잘라준다.

200.1.1.0/24





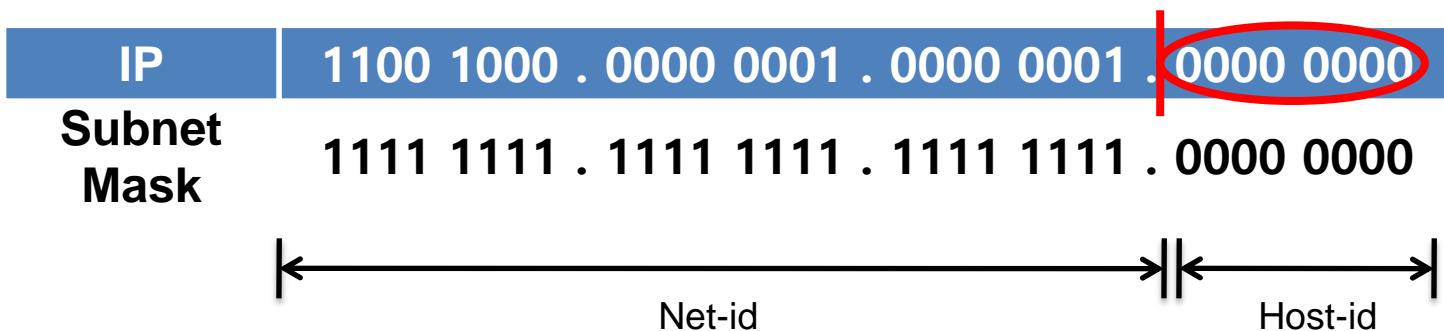
## Step 3

### 새로운 Net-ID와 Host-ID의 범위 구하기

Host-ID로 필요한 bit가 5개였으니, Subnetting할 기준의 Host-ID의 범위 뒤에서부터 5개의 bit를 세어본다.

세어진 bit를 기준으로 나눠서 새로운 Net-ID와 Host-ID의 범위를 구한다.

200.1.1.0/24



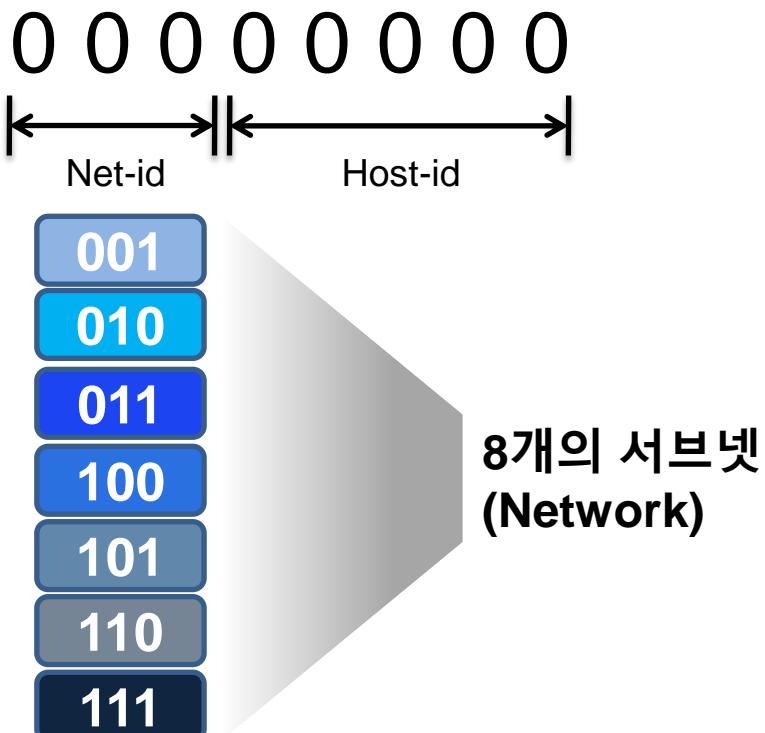


## Step 3

### 새로운 Net-ID와 Host-ID의 범위 구하기

Host-ID로 필요한 bit가 5개였으니, Subnetting할 기준의 Host-ID의 범위 뒤에서부터 5개의 bit를 세어본다.

세어진 bit를 기준으로 나눠서 새로운 Net-ID와 Host-ID의 범위를 구한다. 늘어난 Net-ID의 bit만큼 Subnet을 구분할 수 있다.



# Subnetting



# Step 4

## 잘려진 Subnet의 수와 Subnet당 IP의 개수 파악하기

나뉘어진 Net-ID와 Host-ID의 범위를 보고 증가된 네트워크의 수와 각 네트워크의 IP개수를 파악한다.

IP	1100 1000 . 0000 0001 . 0000 0001 . 0000 0000
Subnet Mask	1111 1111 . 1111 1111 . 1111 1111 . 1110 0000

- Net-ID가 3bit가 추가되었기 때문에 Network는 총 8개( $2^3$ )로 늘어난다.
  - Host-ID가 5bit로 감소되었기 때문에 각 네트워크 당 32개씩( $2^5$ ) IP를 가진다.



## Step 5

### 각 네트워크의 IP주소 범위 구하기

나누어진 Net-ID와 Host-ID의 범위를 보고 네트워크의 대표주소와 Broadcast 주소, 가용IP 등을 구해보자.

IP	1100 1000 . 0000 0001 . 0000 0001 . 000 00000
첫번째 Subnet	1100 1000 . 0000 0001 . 0000 0001 . 000 00000 1100 1000 . 0000 0001 . 0000 0001 . 000 11111
두번째 Subnet	1100 1000 . 0000 0001 . 0000 0001 . 001 00000 1100 1000 . 0000 0001 . 0000 0001 . 001 11111
세번째 Subnet	1100 1000 . 0000 0001 . 0000 0001 . 010 00000 1100 1000 . 0000 0001 . 0000 0001 . 010 11111
	• • •



## Step 5

### 각 네트워크의 IP주소 범위 구하기

나누어진 Net-ID와 Host-ID의 범위를 보고 네트워크의 대표주소와 Broadcast 주소, 가용IP 등을 구해보자.

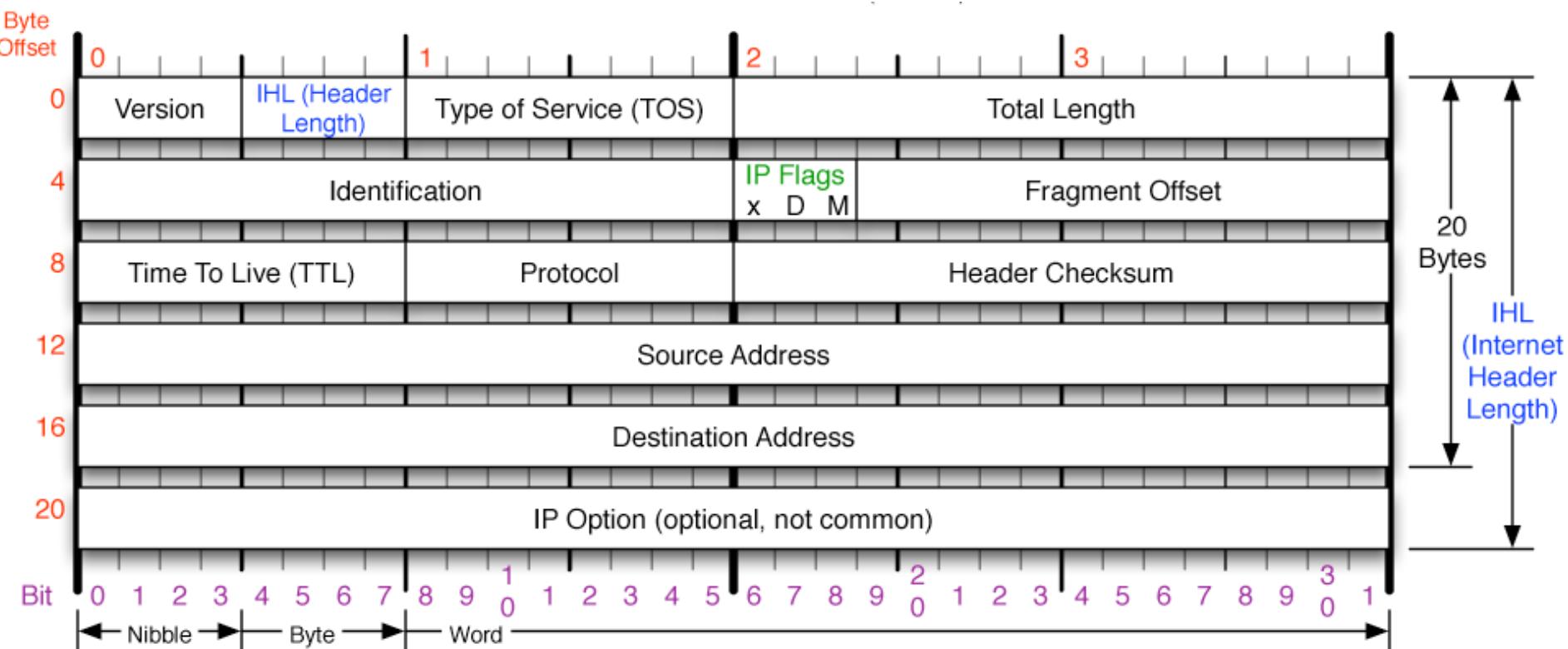
### 네트워크별 대표주소

- ① 200.1.1.0    ⑤ 200.1.1.128
- ② 200.1.1.32    ⑥ 200.1.1.160
- ③ 200.1.1.64    ⑦ 200.1.1.192
- ④ 200.1.1.96    ⑧ 200.1.1.224

**Prefix : /27**

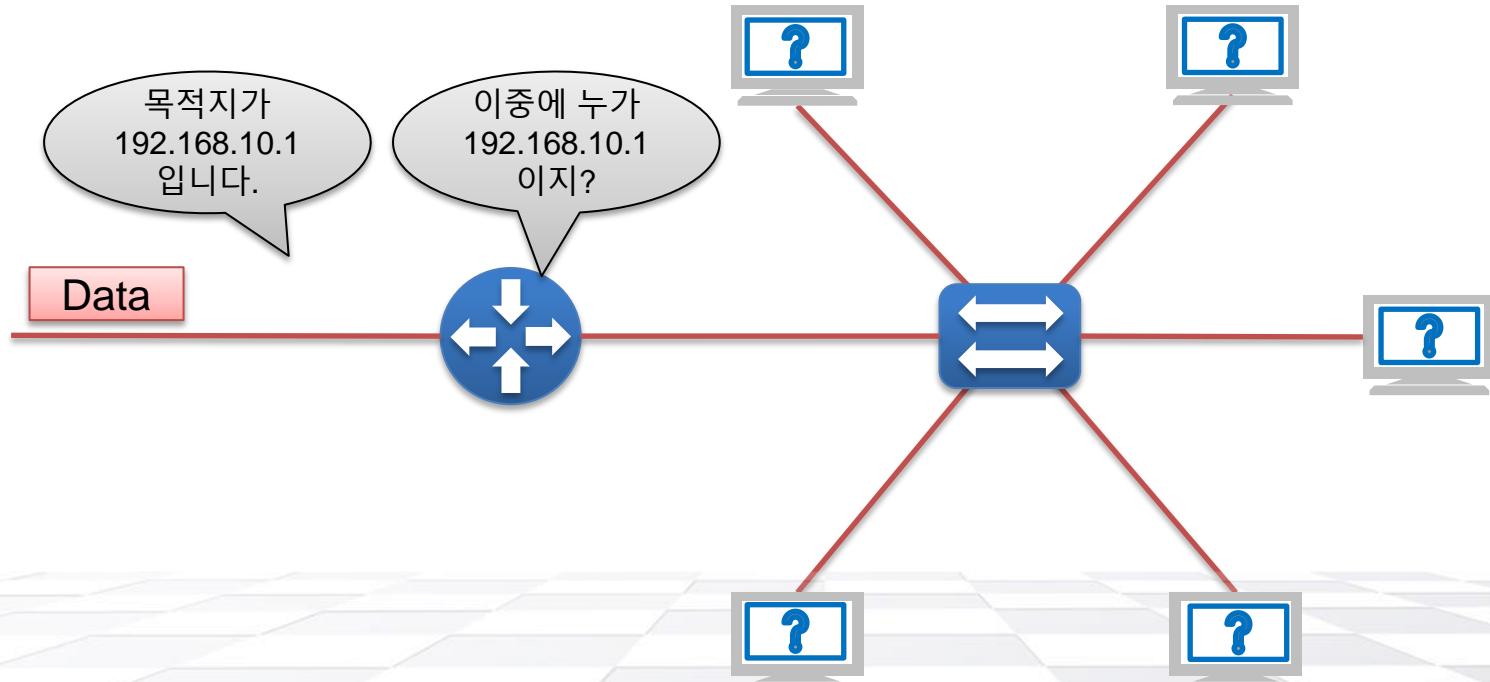
**Subnet-mask : 255.255.255.224**

# IP Header



# ARP Protocol

Address Resolution Protocol

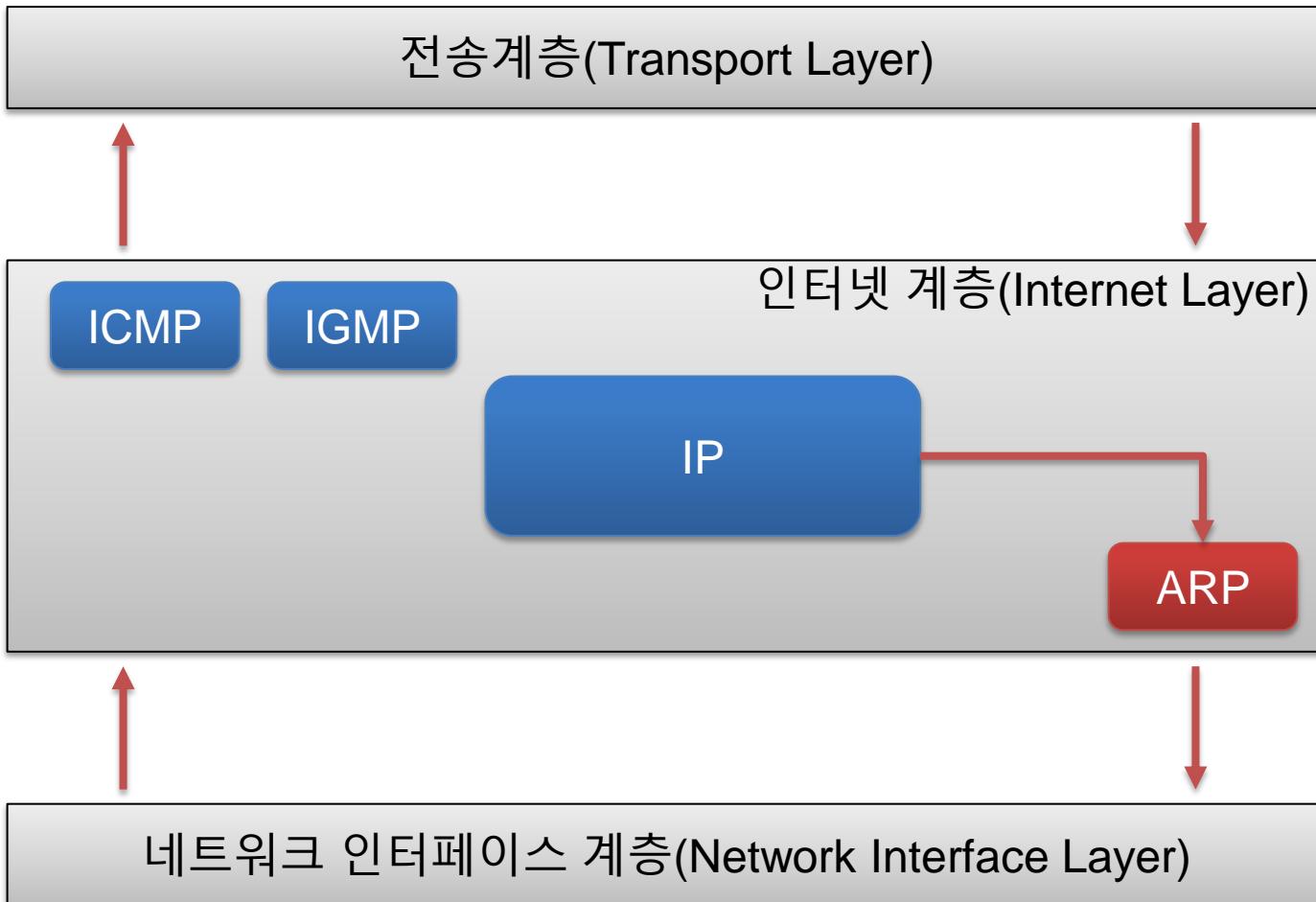


## ARP 사용 개요

- 스위치(Switch)나 허브(Hub)같은 장비를 이용하여 여러 장비가 서로 통신할 수 있도록 연결되어 있는 경우 2계층 물리적 주소를 이용하여 데이터(Data)를 전송한다. 3계층 IP 주소를 이용하여 2계층 물리적 주소를 확인하기 위해서 TCP/IP Protocol에서는 Address Resolution Protocol(ARP)를 이용한다.

**ARP**

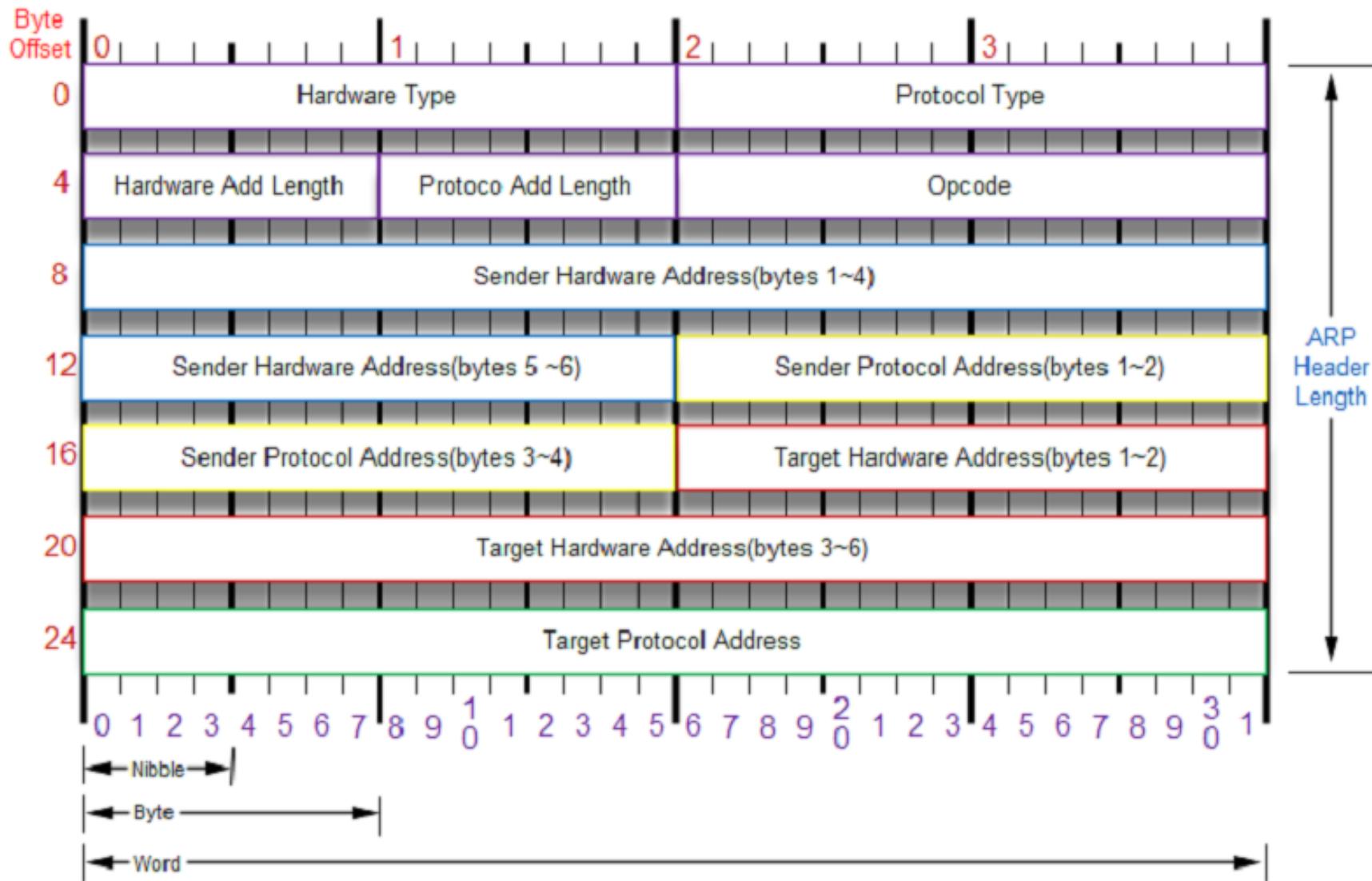
# Location of ARP on TCP/IP



# ARP Packet Format



# ARP Header





## ARP Operation Code

### ARP 요청(Request)

ARP 요청 메시지는 Broadcast로 이루어지면 요청 메시지를 받는 장비는 Send H/W 주소와 IP 주소를 학습하여 ARP Cache 저장한다.

### ARP 응답(Reply)

ARP 응답 메시지는 Unicast로 전송이 되면 전송 받은 장비는 Send H/W 주소와 IP 주소를 학습하여 ARP Cache 저장한다.

# Host-to-Host Packet Process(1/14)

Chapter. 02 Network Layer



## Application 계층

192.168.10.2 PC 에게 Data를 전송한다.



## Transport 계층

TCP를 통해서 안전하고 신뢰성을 가지고 전송.



## Transport 계층

192.168.10.2 PC 에게 먼저 Syn를 전송한다.



3계층 : 192.168.10.1

2계층 : 0800:0222:1111

3계층 : 192.168.10.2

2계층 : 0800:0222:2222

# Host-to-Host Packet Process (2/14)



192.168.10.2으로 Packet를 전송하기  
위해서 2계층으로 내려 보낸다.



# Host-to-Host Packet Process (3/14)



2계층에서는 Packet를 전송하기 위해서 192.168.10.2 번의 2계층 주소를 ARP Cache 테이블에서 확인 하지만 매칭되는 정보가 없다.

ARP Cache 테이블에 매칭되는 2계층 주소가 없기 때문에 Packet를 잠시 임시저장하고 ARP를 통해서 2계층 주소를 학습한다.

출발지 IP  
192.168.10.1      목적지 IP  
192.168.10.2      TCP  
SYN



# Host-to-Host Packet Process (4/14)



임시저장

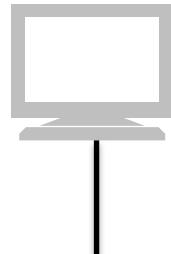
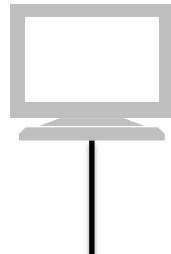
Packet

첫 번째로 ARP 요청을 Packet를 전송한다. 나는 IP 192.168.10.1 과 MAC 0800:0222:1111를 사용하고 있다. 192.168.10.2 PC 너의 MAC 주소를 알려 주세요.

ARP  
Request

ARP 요청 Packet은 출발지 MAC 주소는 자기 MAC를 입력하고, 목적지 MAC 주소는 상대방에 MAC 주소를 모르기 때문에 Broadcast 주소로 전송한다.

목적지 MAC FFFF:FFFF:FFFF	출발지 MAC 0800:0200:1111	ARP Request
---------------------------	---------------------------	----------------



3계층 : 192.168.10.1  
2계층 : 0800:0222:1111

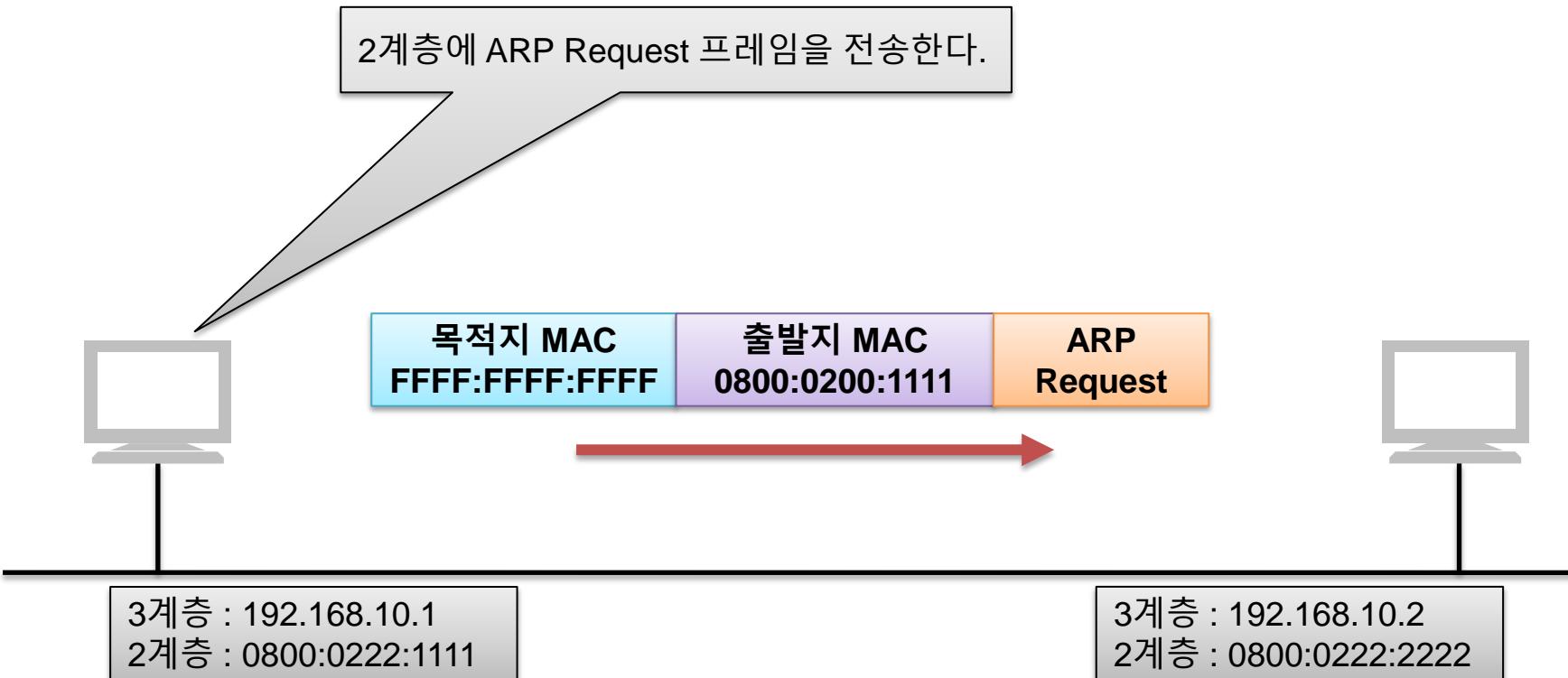
3계층 : 192.168.10.2  
2계층 : 0800:0222:2222

# Host-to-Host Packet Process (5/14)



임시저장  
Packet

2계층에 ARP Request 프레임을 전송한다.



# Host-to-Host Packet Process (6/14)



임시저장

Packet

목적지 주소가 Broadcast Mac 주소로 입력되어 있는 Frame 받아서 처리 한다.  
Frame 헤더 부분에 Type 필드를 통해서 ARP Packet을 확인한다.

목적지 MAC  
FFFF:FFFF:FFFF      출발지 MAC  
0800:0200:1111      ARP  
Request



# Host-to-Host Packet Process (7/14)

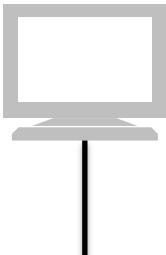


임시저장

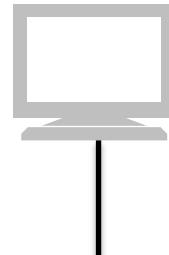
Packet

2계층 Frame 헤더를 제거하고 ARP에게 요청 메시지를 넘겨 준다.

ARP  
Request



3계층 : 192.168.10.1  
2계층 : 0800:0222:1111



3계층 : 192.168.10.2  
2계층 : 0800:0222:2222

# Host-to-Host Packet Process (8/14)

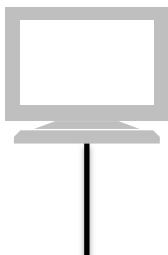


임시저장

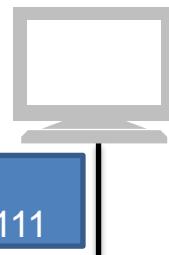
Packet

ARP는 192.18.10.1에서 전송된 요청 메시지를 받아서 처리한다. 요청 메시지 안에 요청자에 IP 주소와 MAC 주소를 ARP Cache 테이블에 저장하고 응답한다.

ARP  
Request



3계층 : 192.168.10.1  
2계층 : 0800:0222:1111



Arp cache Table  
192.168.10.1 = 0800.0222.1111

3계층 : 192.168.10.2  
2계층 : 0800:0222:2222

# Host-to-Host Packet Process (9/14)



임시저장

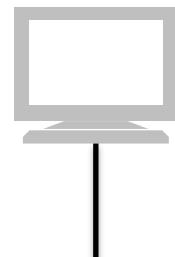
Packet

ARP는 자신의 IP 주소와 MAC 주소를 응답메시지에 입력하고 전송한다.

2계층에서는 학습된 ARP Cache 정보를 이용하여 Frame 헤더에 목적지 MAC 주소를 입력하여 Frame을 생성한다.

ARP  
Reply

목적지 MAC  
0800:0200:1111      출발지 MAC  
0800:0200:2222      ARP  
Reply



3계층 : 192.168.10.1  
2계층 : 0800:0222:1111

Arp cache Table  
192.168.10.1 = 0800.0222.1111

3계층 : 192.168.10.2  
2계층 : 0800:0222:2222

# Host-to-Host Packet Process (10/14)

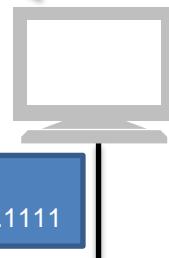
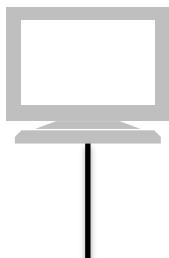
Chapter. 02 Network Layer



임시저장

Packet

2계층에서 ARP Reply 프레임을 전송한다.



Arp cache Table  
192.168.10.1 = 0800.0222.1111

3계층 : 192.168.10.1  
2계층 : 0800:0222:1111

3계층 : 192.168.10.2  
2계층 : 0800:0222:2222

# Host-to-Host Packet Process (11/14)

Chapter. 02 Network Layer



임시저장

Packet

목적지 MAC 주소가 자신의 MAC 주소와 동일하면 받아서 처리 한다. Frame 헤더 부분에 Type 필드를 통해서 ARP Packet를 확인 한다.

목적지 MAC  
0800:0200:1111

출발지 MAC  
0800:0200:2222

ARP  
Reply



3계층 : 192.168.10.1  
2계층 : 0800:0222:1111

Arp cache Table  
192.168.10.1 = 0800.0222.1111

3계층 : 192.168.10.2  
2계층 : 0800:0222:2222

# Host-to-Host Packet Process (12/14)

Chapter. 02 Network Layer



임시저장

Packet

2계층 Frame 헤더를 제거하고 ARP에게 응답 메시지를 넘겨 준다.

ARP  
Reply



3계층 : 192.168.10.1  
2계층 : 0800:0222:1111



Arp cache Table  
192.168.10.1 = 0800.0222.1111

3계층 : 192.168.10.2  
2계층 : 0800:0222:2222

# Host-to-Host Packet Process (13/14)

## Chapter. 02 Network Layer



임시저장

Packet

ARP는 192.168.10.2에서 전송된 응답 메시지를 받아서 처리한다. 응답 메시지 안에 응답자의 IP 주소와 MAC 주소를 ARP Cache 테이블에 저장하고 응답한다.

ARP  
Reply



Arp cache Table  
192.168.10.2 = 0800.0222.2222

3계층 : 192.168.10.1  
2계층 : 0800:0222:1111



Arp cache Table  
192.168.10.1 = 0800.0222.1111

3계층 : 192.168.10.2  
2계층 : 0800:0222:2222

# Host-to-Host Packet Process (14/14)



임시저장 되어 있는 Packet를 ARP를 통해서 학습한 MAC 주소를 이용하여 Frame을 전송한다.



Arp cache Table  
192.168.10.2 = 0800.0222.2222

3계층 : 192.168.10.1  
2계층 : 0800:0222:1111



Arp cache Table  
192.168.10.1 = 0800.0222.1111

3계층 : 192.168.10.2  
2계층 : 0800:0222:2222

# ARP Cache Table



```
관리자: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright <c> 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>arp -a

인터넷페이스: 121.160.70.205 --- 0xb
인터넷 주소          물리적 주소
121.160.70.41        00-26-66-81-3d-dd
121.160.70.81        e8-03-9a-63-05-6b
121.160.70.82        50-b7-c3-9e-eb-dc
121.160.70.83        e8-03-9a-64-e1-6b
121.160.70.84        e8-03-9a-64-90-d7
121.160.70.86        e8-03-9a-64-90-d3
121.160.70.88        50-b7-c3-9e-e8-36
121.160.70.89        e8-03-9a-64-90-cf
121.160.70.90        e8-03-9a-64-93-f2
```



## ARP Cache Table

- ARP를 통해서 IP 주소와 MAC 주소와의 Mapping 정보를 기록한 Table이다.
- ARP Cache Table은 임시적으로 정보를 저장하기 때문에 한동안 통신을 하지 않으면, 자동적으로 정보가 삭제 된다.

# ARP Cache Table



The image shows two separate windows of the Windows Command Prompt (cmd.exe) running under administrator privileges. Both windows have a light blue title bar with the text '관리자: C:\Windows\system32\cmd.exe'.  
The left window contains the command: `C:\> netsh interface ip delete arpcache`. Below it, the output '확인됨' (verified) is displayed.  
The right window contains the command: `C:\> arp -d *`.  
Both windows show a single command prompt line at the bottom.



## ARP Cache 삭제

- ARP Cache 정보를 지우기 위해서는 관리자 권한으로 cmd.exe 파일을 실행해야 한다.
- `netsh interface ip delete arpcache`
- `arp -d *`

# ICMP Protocol

Internet Control Message Protocol



## 1 IP는 신뢰성을 보장하지 않는다.

따라서 네트워크 장애나 중계 라우터 등의 에러에 대처 할 수 없다. 이런 경우 수신측에서 송시측으로 데이터의 사고에 대한 내용을 전달할 필요가 있다. ICMP는 이와 같은 오류 정보를 발견 송시측에 메시지를 전달하는 기능을 한다.

## 2 네트워크 유ти리트 – ping

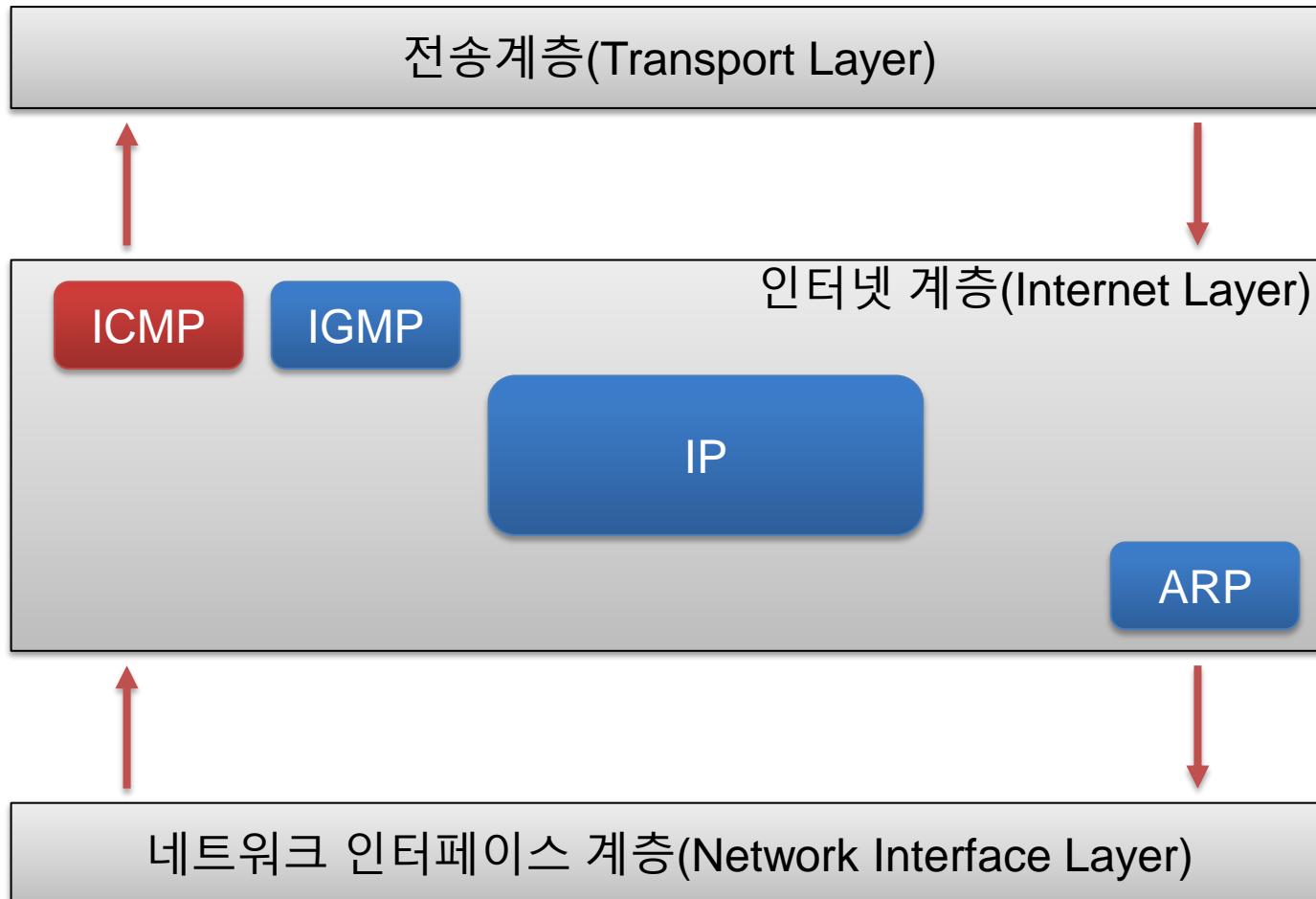
핑(ping)유ти리티는 ICMP “에코 요청(Echo request)”과 “에코 응답(Echo reply)” 메시지를 사용해 구현할 수 있다. 또한 이 방식을 이용하여 특정 컴퓨터와의 통신 상태를 체크 할 수 있다.

## 3 네트워크 유ти리트 - traceroute

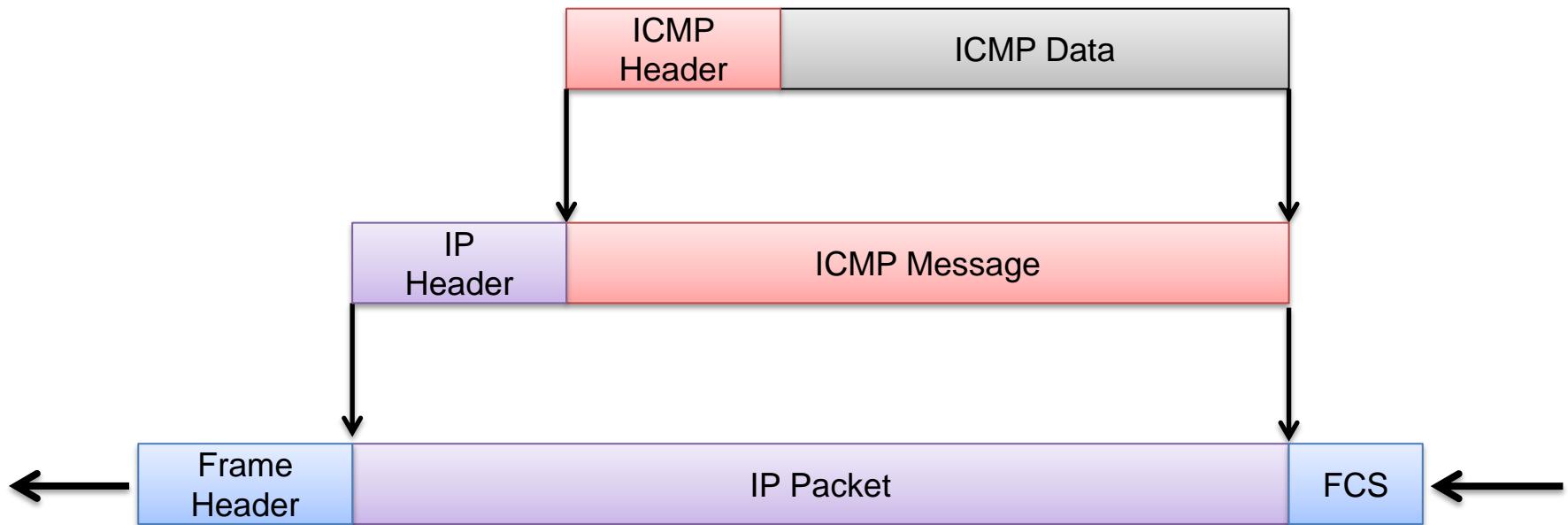
트레이스라우트(traceroute)명령어는 특별하게 만들어진 IP TTL 헤더 필드들을 가진 IP 데이터그램을 전송하고 응답에서 ICMP TTL 초과 메시지와 “목적지에 닿을 수 없음(Destination unreachable)” 메시지가 생성되었는지 찾는다.

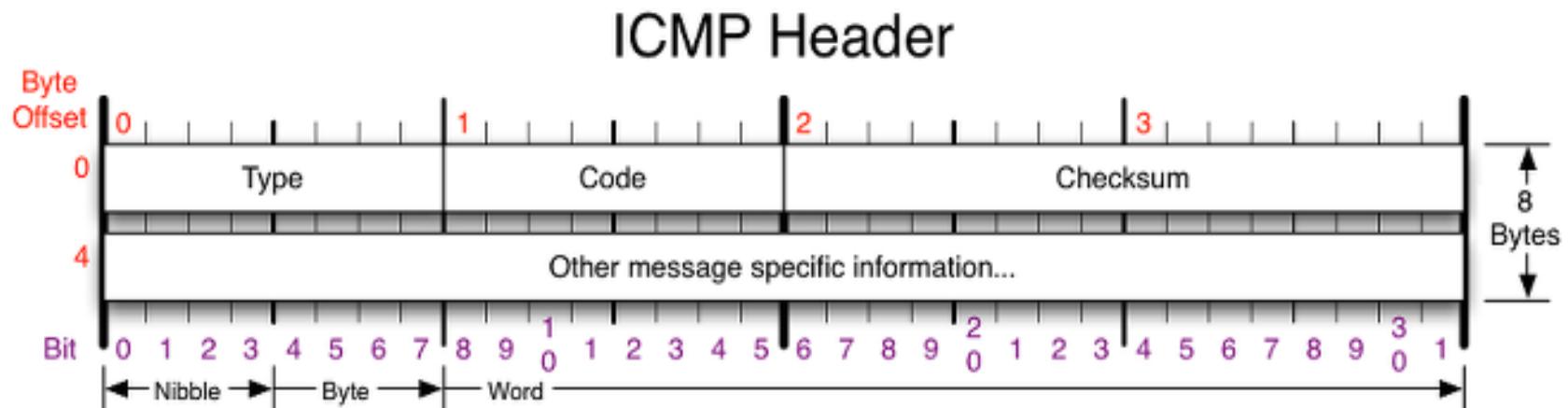
# Location of ICMP on TCP/IP

Chapter. 02 Network Layer



# ICMP Packet Format





#### ICMP Type

0 에코 응답(Echo Reply)

#### ICMP Type

8 에코 요청(Echo Request)

#### ICMP Type

3 목적지에 연결할 수 없다.(Destination Unreachable)

##### Type Code

0 네트워크 연결할 수 없다.(Network Unreachable)

1 호스트에 연결할 수 없다.(Host Unreachable)

.....

13 통신금지(Communication Prohibited)



## ICMP Message Type

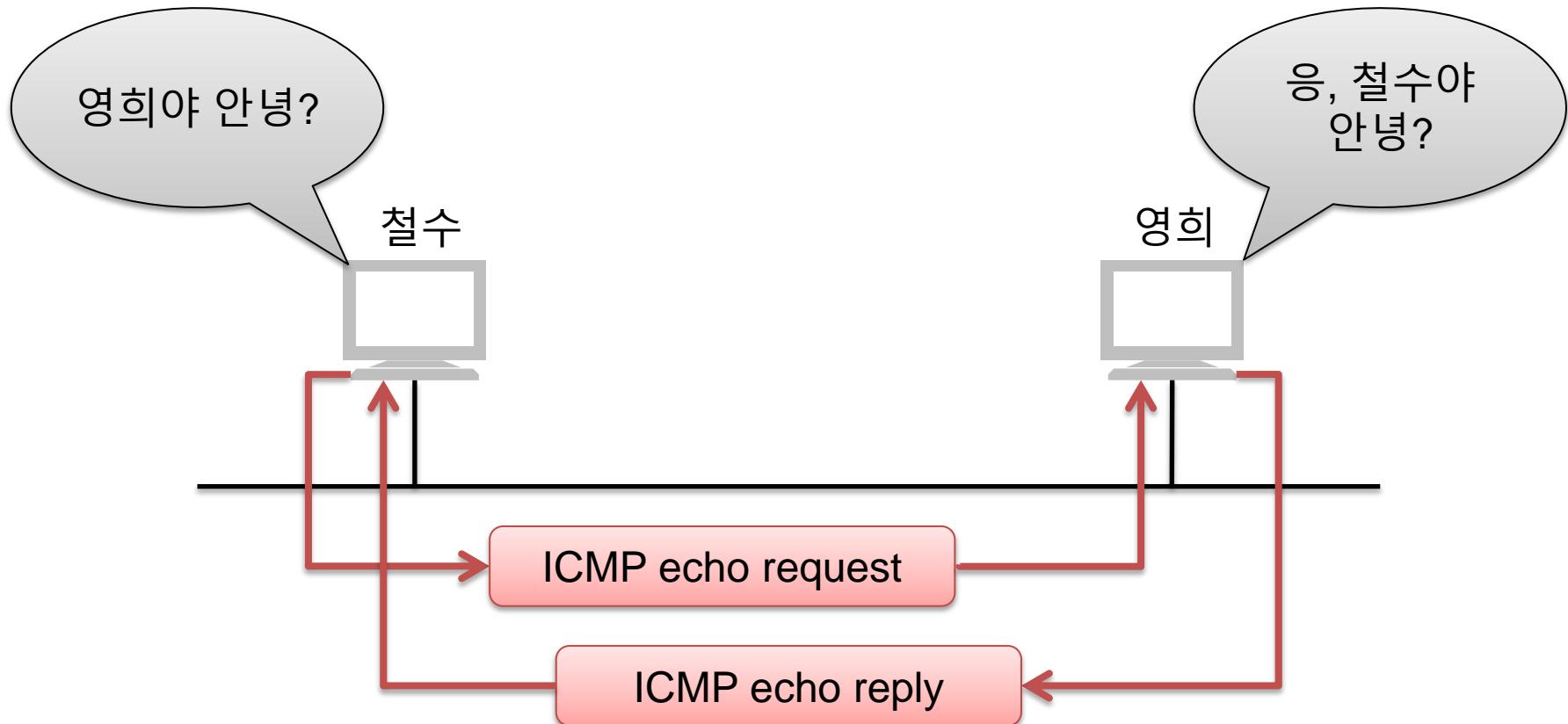
### 에러 보고 메시지 형식

Type	Message
3	Destination Unreachable
4	Source quench
5	Redirect Message
11	Time exceeded
12	Parameter Problem

### 요청 메시지 형식

Type	Message
8	Echo Request
0	Echo Reply
13	Timestamp request
11	Timestamp reply

# ICMP Echo Message



Ping 명령어를 이용하여 ICMP Echo 메세지를 생성할 수 있다.

# ICMP Echo Message(Cont.)



```
cmd C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.

C:\Users\Gene Administrator>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time=2ms TTL=64
Reply from 192.168.1.1: bytes=32 time=1ms TTL=64
Reply from 192.168.1.1: bytes=32 time=1ms TTL=64
Reply from 192.168.1.1: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 2ms, Average = 1ms

C:\Users\Gene Administrator>
```

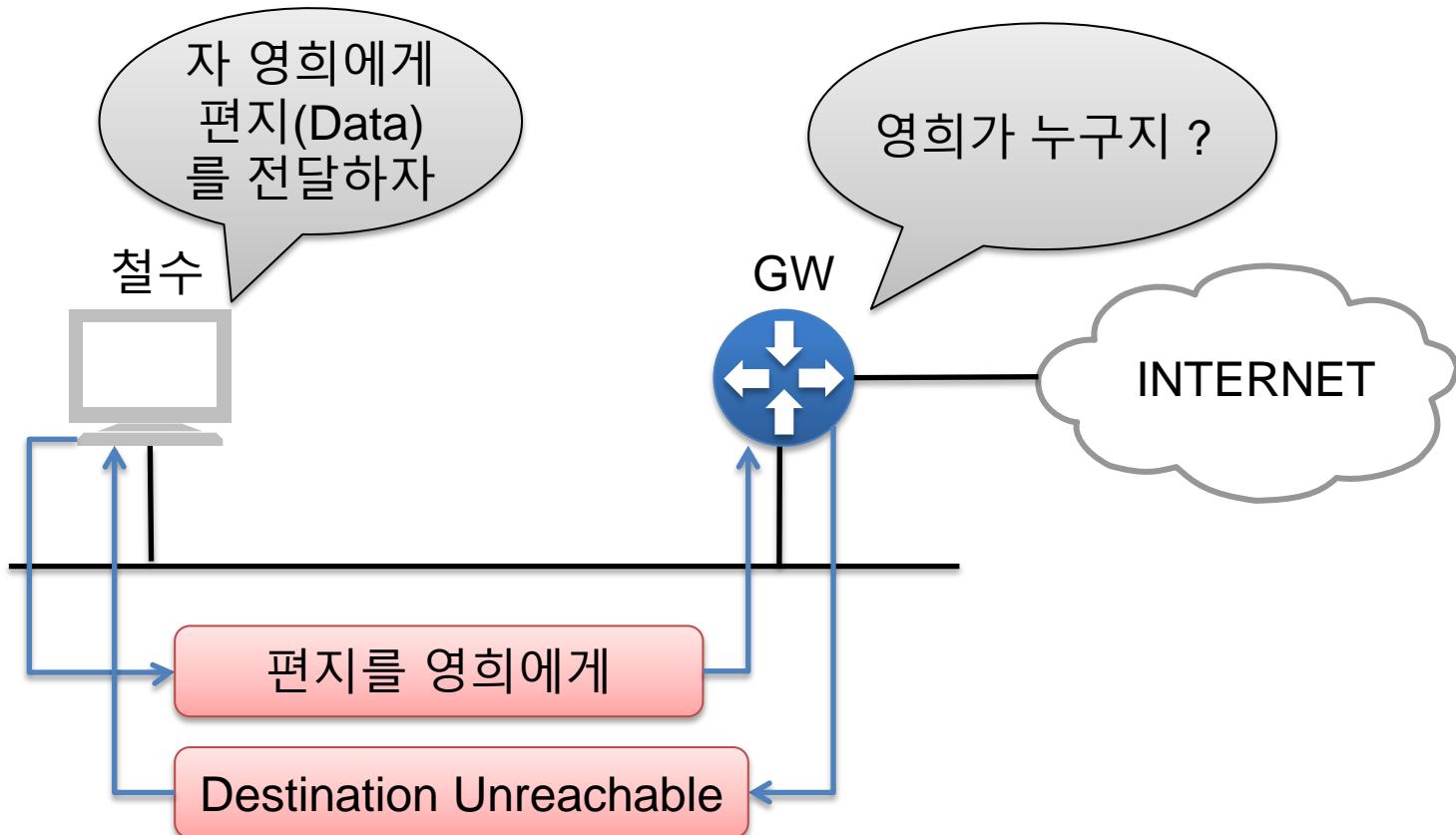


## ICMP Utility PING

- ICMP 패킷을 통해 네트워크의 연결성을 확인하기 위해서 Ping이라는 유틸리티를 사용한다.
- ICMP 패킷으로 목적지 네트워크 까지의 연결성을 확인하는 것을 ‘도달성 체크’라 한다.

**PING**

# ICMP Unreachable Message



철수가 요청한 정보에 대해서 Router장비가 모르는 경우 발생

# ICMP Unreachable Message (Cont.)



```
C:\>Command Prompt  
C:\>ping 67.12.0.1  
Pinging 67.12.0.1 with 32 bytes of data:  
Reply from 160.81.110.41: Destination host unreachable.  
Reply from 160.81.110.45: Destination host unreachable.  
Reply from 160.81.110.45: Destination host unreachable.  
Reply from 160.81.110.45: Destination host unreachable.  
  
Ping statistics for 67.12.0.1:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 0ms, Maximum = 0ms, Average = 0ms  
  
C:\>
```

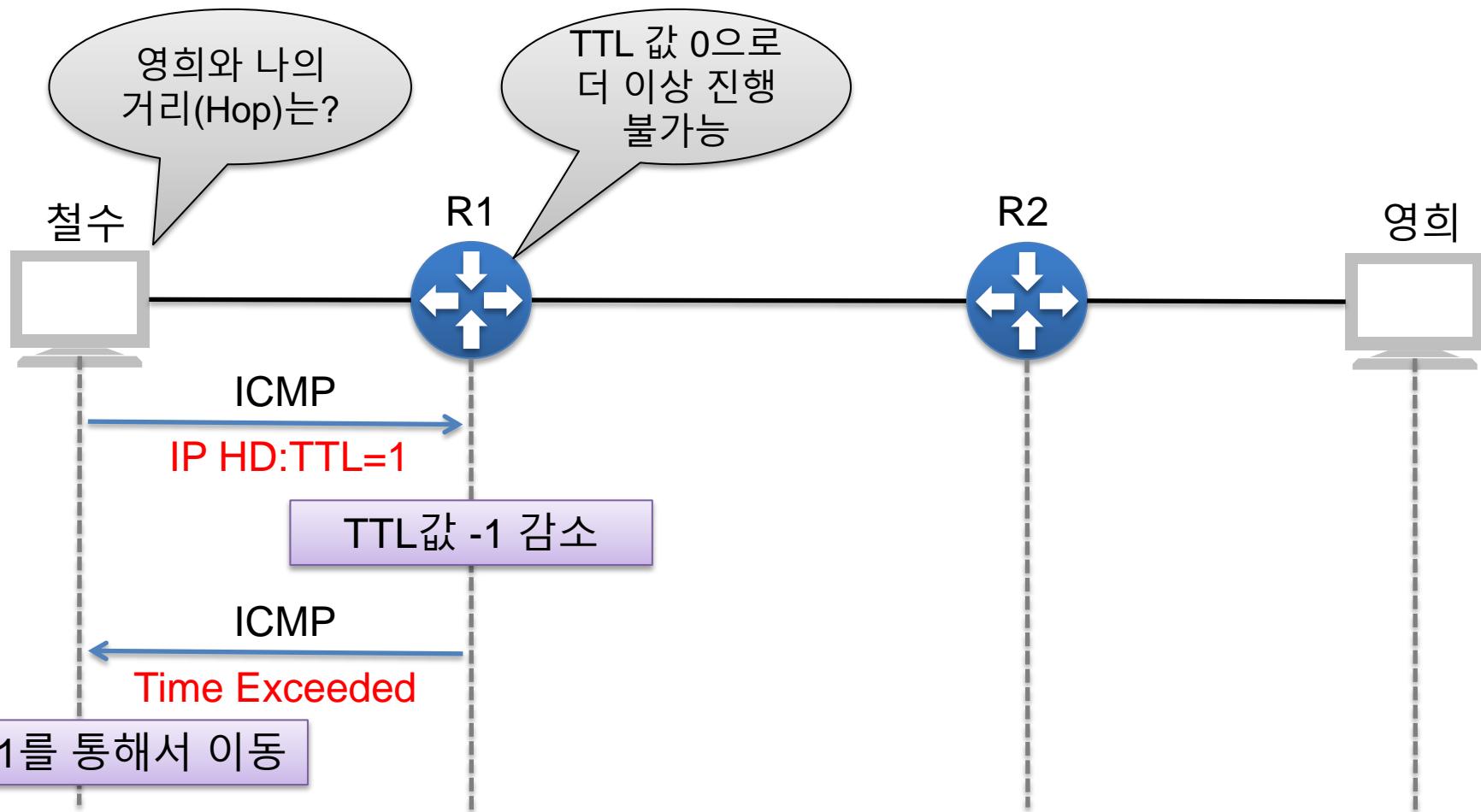


## Destination Unreachable

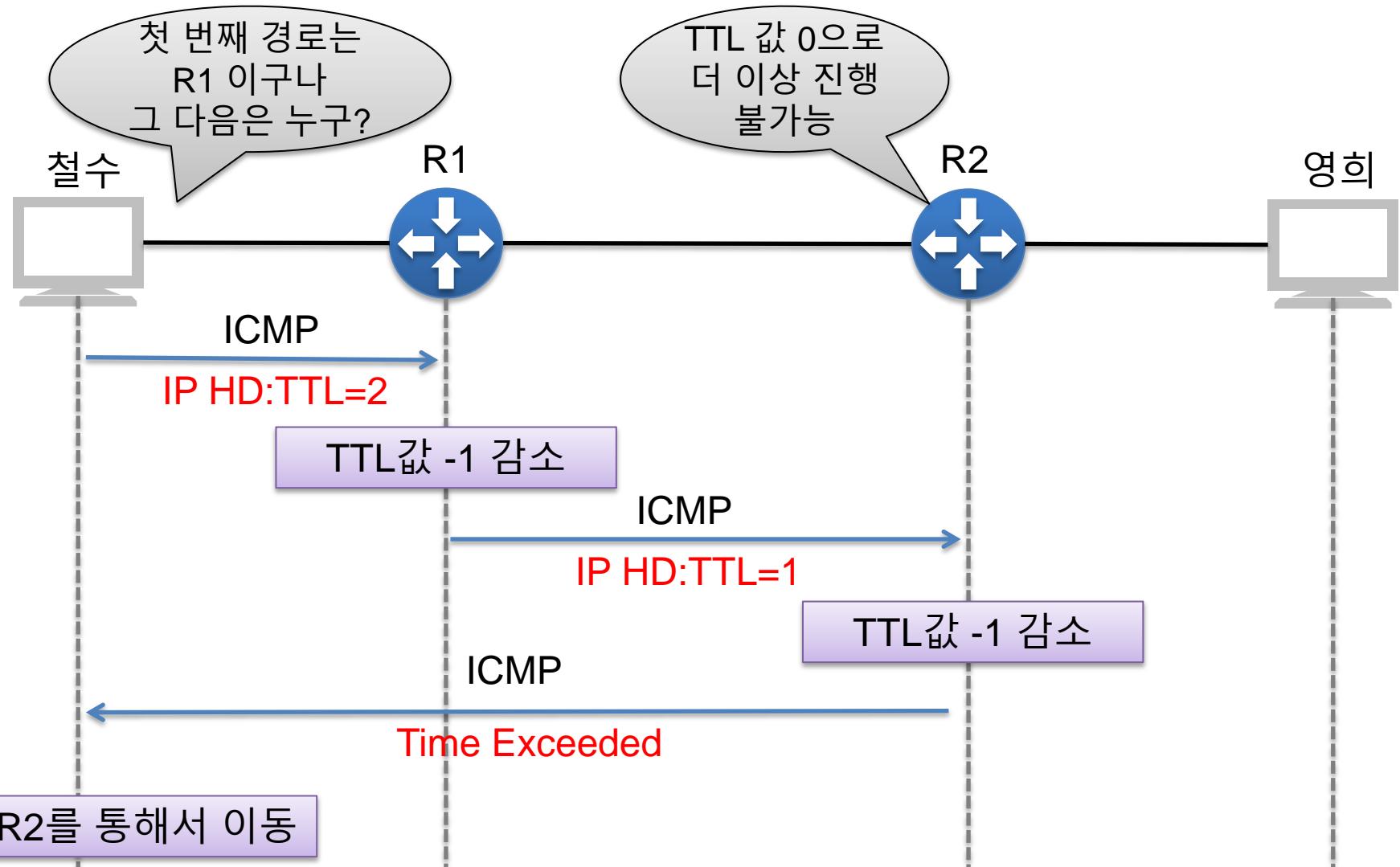
- 도달성 체크를 위해 Ping을 사용 할 때 발생되는 오류 중 하나로 'Destination host unreachable'이 있다.
- 이 오류는 Router가 목적지에 해당하는 네트워크 경로를 모를 때 발생하게 된다.

## Error Message

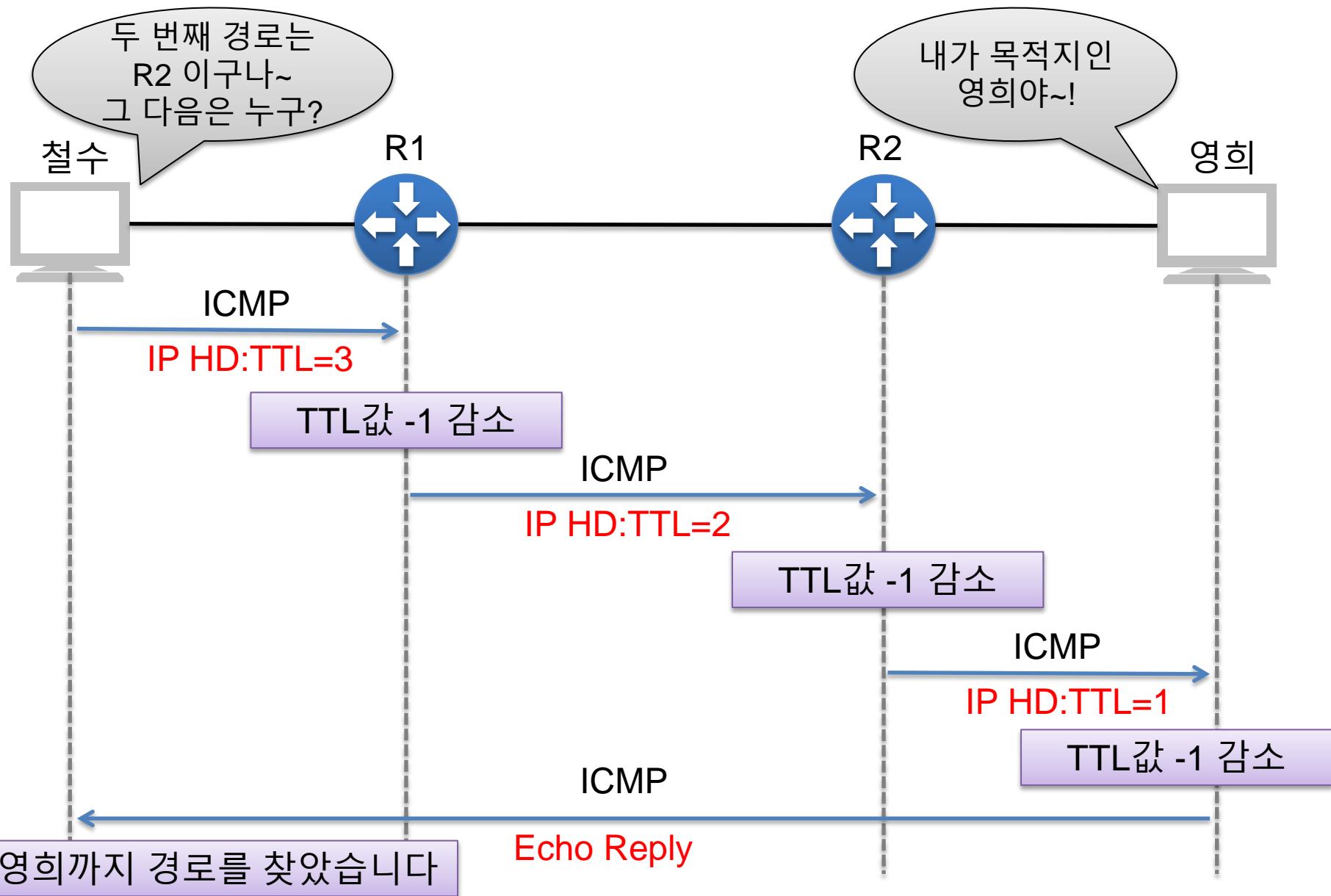
# ICMP Time Exceeded



# ICMP Time Exceeded(Cont.)



# ICMP Time Exceeded(Cont.)



# ICMP Time Exceeded (Cont.)



```
C:\Windows\system32\cmd.exe
C:\Users\Jon Bennell>tracert 8.8.8.8
Tracing route to google-public-dns-a.google.com [8.8.8.8]
over a maximum of 30 hops:
 1  12 ms    <1 ms    <1 ms  10.250.250.1
 2  1 ms      2 ms      1 ms  86.188.176.113
 3  4 ms      4 ms      4 ms  86.189.44.5
 4  4 ms      4 ms      4 ms  217.32.170.177
 5  7 ms      6 ms      6 ms  62.172.103.58
 6  8 ms      7 ms      6 ms  core4-pos0-1-5-0.telehouse.ukcore.bt.net [195.99
.125.225]
 7  6 ms      7 ms      6 ms  195.99.126.62
 8  7 ms      7 ms      7 ms  209.85.255.175
 9  13 ms     13 ms     13 ms  66.249.95.173
10  12 ms     13 ms     12 ms  209.85.251.231
11  14 ms     15 ms     17 ms  209.85.243.85
12  13 ms     13 ms     13 ms  google-public-dns-a.google.com [8.8.8.8]

Trace complete.
```



## Time Exceeded

- 도달성 체크를 위해 Ping을 사용 할 때 발생되는 오류 중 하나로 ‘Time Exceeded’ 가 있다.
- 이 오류는 IP Packet에 있는 TTL 값이 0이 되어 더 이상 Routing을 해 줄 수 없을 때 발생 된다.
- 이러한 특성을 이용하여, 목적지 네트워크 까지의 경로를 추적하는 명령어가 존재한다.
- Windows에서는 tracert를 사용하며, Linux에서는 traceroute를 사용 한다.

## Error Message

# 04

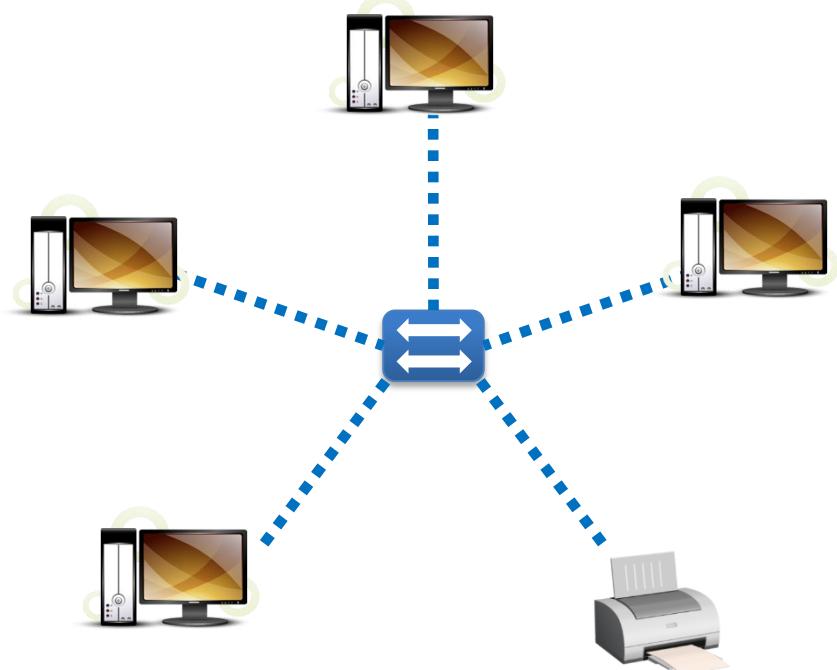
## Data-link Layer

네트워크 토플로지는 컴퓨터 네트워크에 참여하는 노드와 링크의 배치 형태, 망 구성 방식을 의미한다. 다양한 종류의 네트워크 토플로지 특징을 살펴 본다.

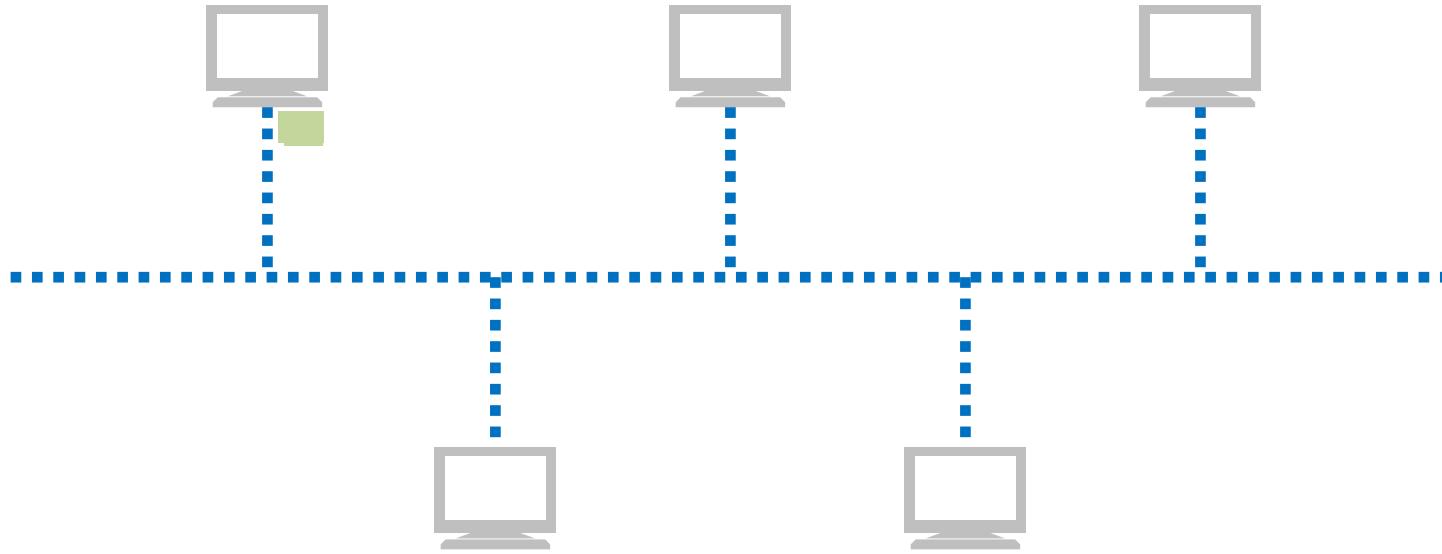


# LAN(Local Area Network)

## Chapter. 02 Data Link Layer



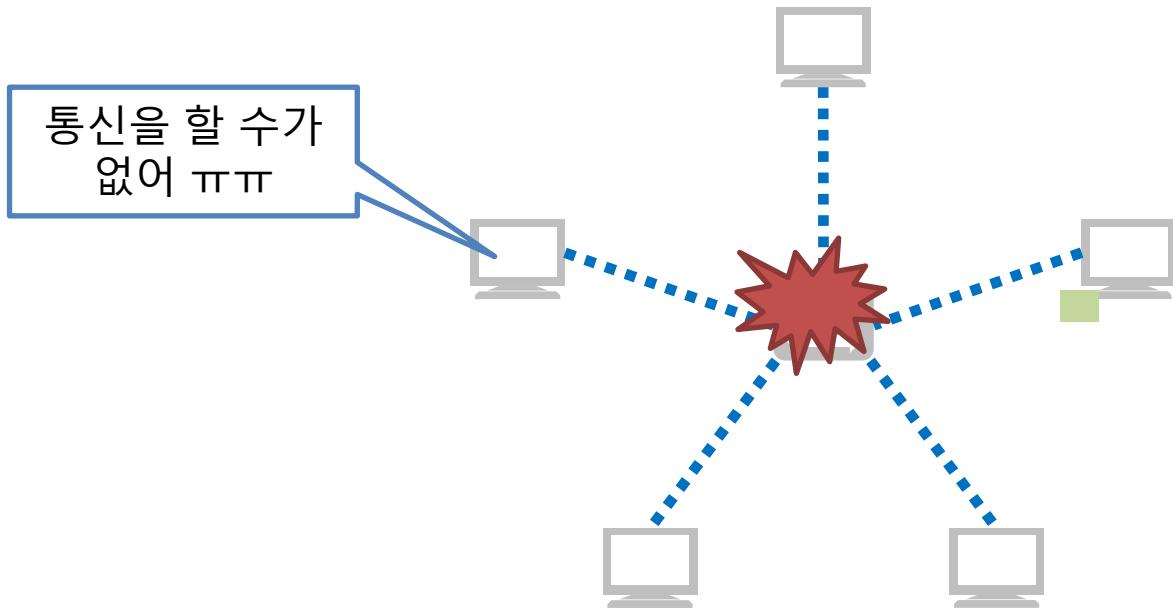
- 조직 내부나 동일 건물 등 비교적 좁은 지역을 연결하기 위한 네트워크
- 초기 투자 비용은 많이 들고 유지 비용은 적게 듈다
- 관리자가 직접 관리



## Bus Topology

- 단일 Media(케이블)에 모든 단말 장치들이 연결이 된 형태
- 각 단말의 고장이 다른 부분에 영향을 끼치지 않는다.
- 공유 Media에 고장이 발생이 되면 전체 단말에 대한 영향을 준다.
- 거리에 민감하기 때문에 거리가 길어 질 수록 추가적인 장치가 필요해 진다.
- 너무 많은 단말들이 연결될 경우 통신 속도가 떨어지며, 쟁돌이라는 문제가 발생이 된다.

**Topology**

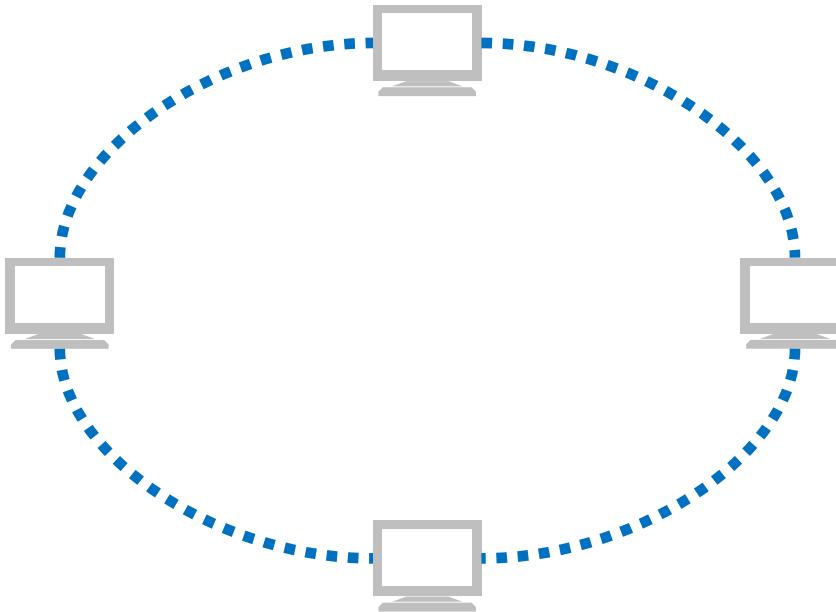


## Star Topology

- 중앙 장치와 직접 연결이 되어 별 모양으로 연결 되어 있는 특징을 가진다.
- 모든 단말 장치들은 중앙 장치를 통해서 통신을 하게 된다.
- 통신 문제의 위치를 쉽게 파악 할 수 있다.
- 영역의 확장을 쉽게 할 수 있다.
- 중앙 장치에 문제가 발생 시, 전체 단말의 통신에 문제가 발생 된다.

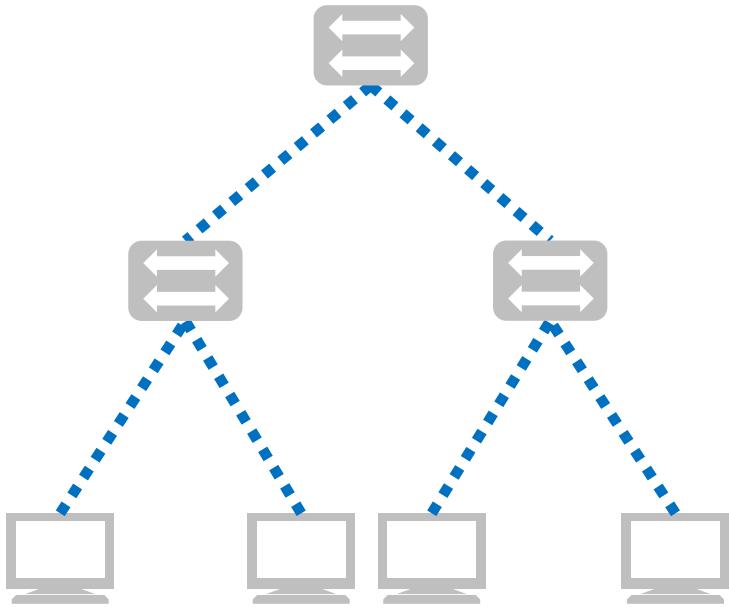


Token



## Ring Topology

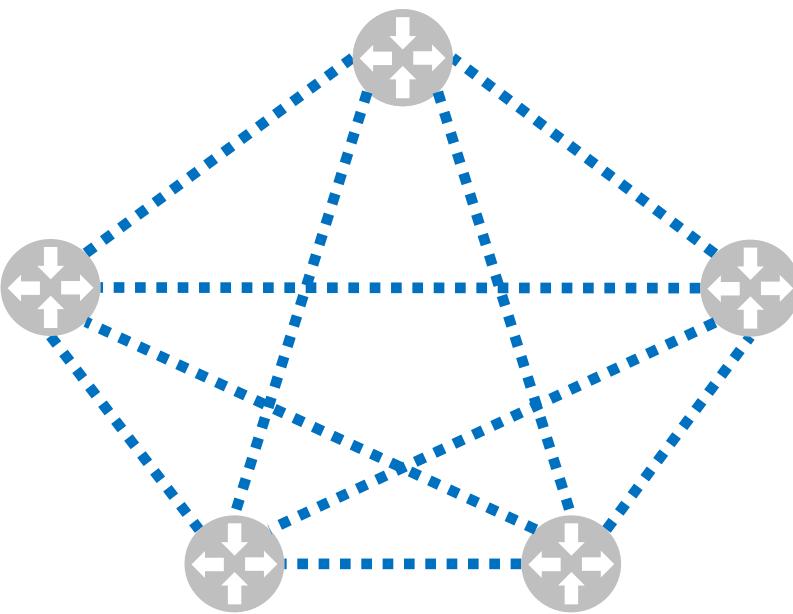
- 통신의 흐름은 항상 단방향으로 진행을 한다.
- 토큰이라고 불리는 제어 체계에 의해서 통신의 순서를 제어하게 된다.
- 통신의 순서를 정하기 때문에 충돌이라는 문제가 발생하지 않는다.
- 순환 구조로 되어 있기 때문에 이 순환이 끊어지게 되면 전체 통신에 문제가 발생된다.
- 통신 망의 변경이 어렵다.



## Tree Topology

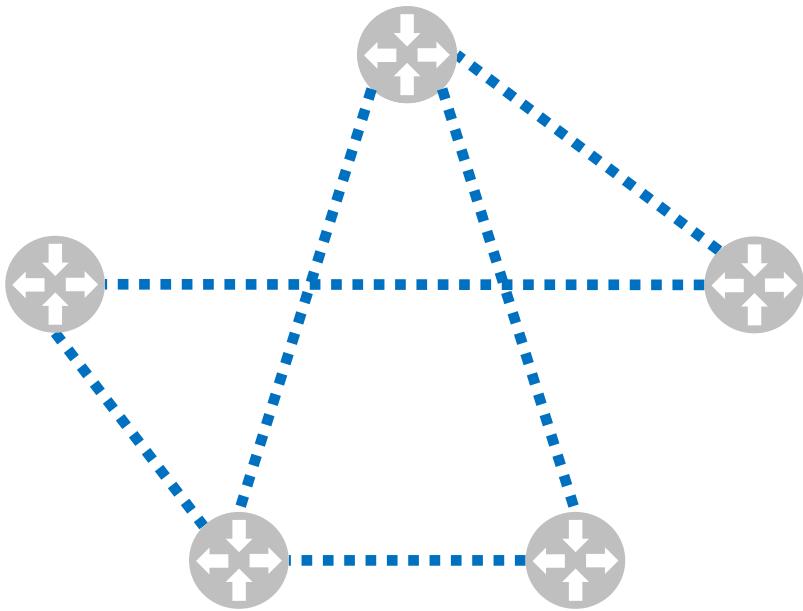
- 계층적 구조를 가지고 있으며, Loop를 발생 시키지 않는 연결이다.
- 통신 망의 확장에 용의하다.
- 최 상위 Root에 문제가 발생 되면, 전체 단말간의 통신에 영향을 끼친다.
- 통신 트래픽이 최 상위 Root에 집중이 되기 때문에 고 성능의 장치를 필요로 한다.

**Topology**



## Full-Mesh Topology

- 통신 영역에서 모든 장치들이 서로간에 연결이 되어 있다.
- 일부 장치에 문제가 발생이 되어도 전체 통신에 대한 문제가 없다.
- 모든 장치들과 연결하기 위해서 많은 비용이 소모 된다.

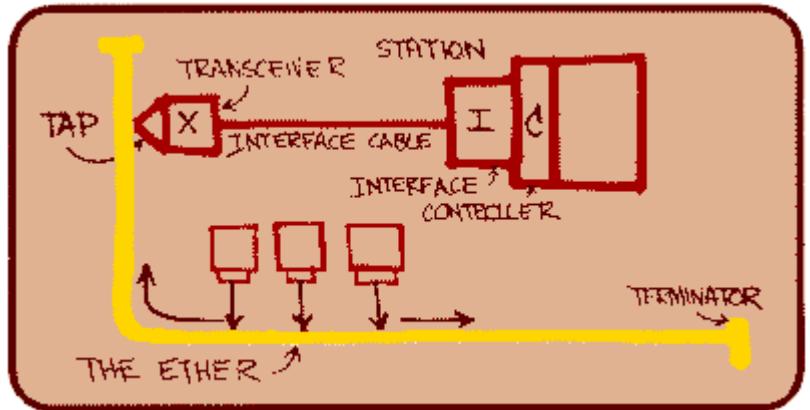


## Partial-Mesh Topology

- Full-Mesh 형의 장점과 단점을 적절히 사용하였다.
- 모든 장치들간에 2개 이상의 연결이 되어 있는 형태이다.
- Full-Mesh 형보다는 안정성이 떨어지지만, 비용적인 면에서는 저렴한 형태이다.

# Ethernet Protocol

What is the Ethernet?



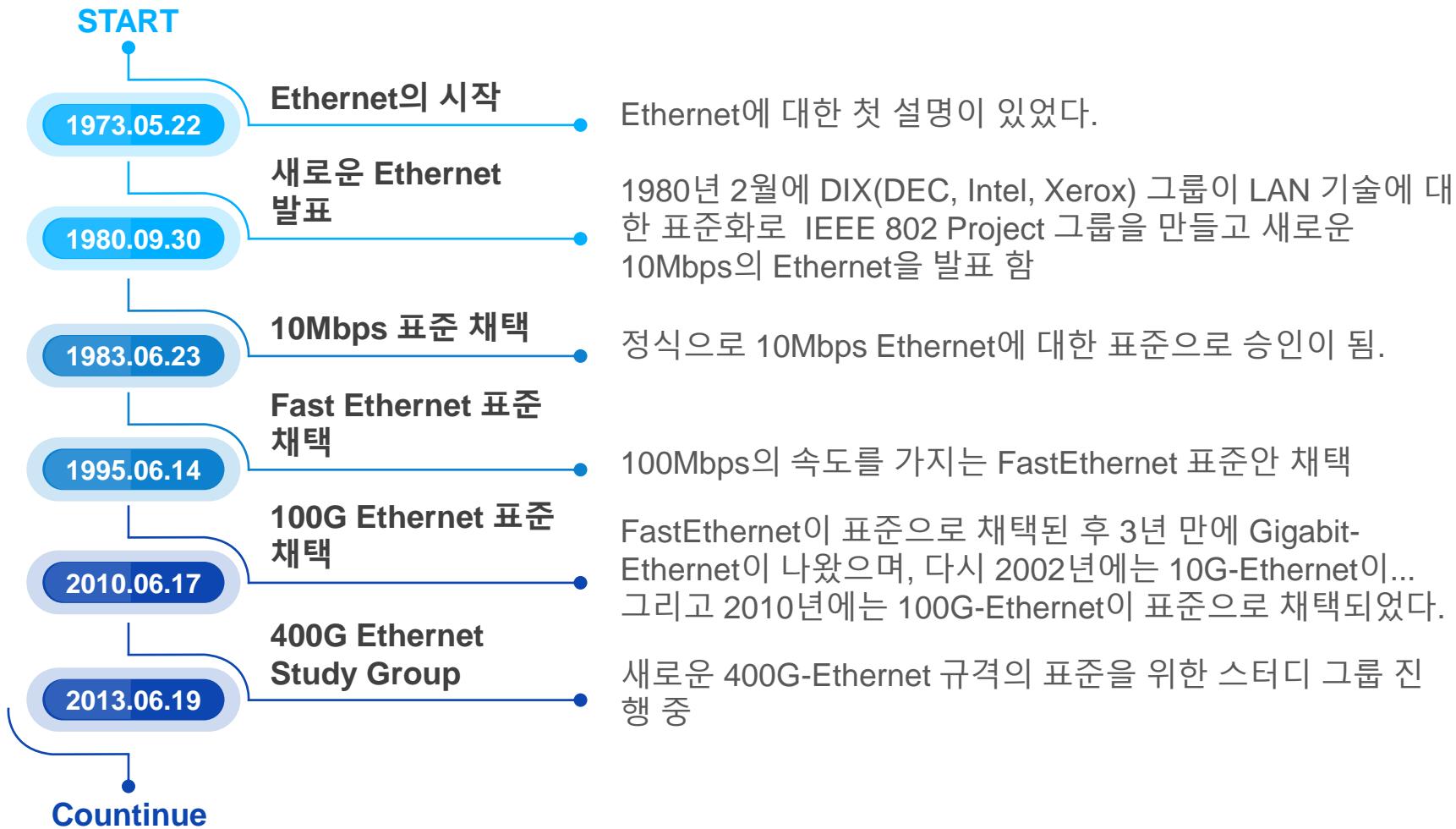
## Ethernet

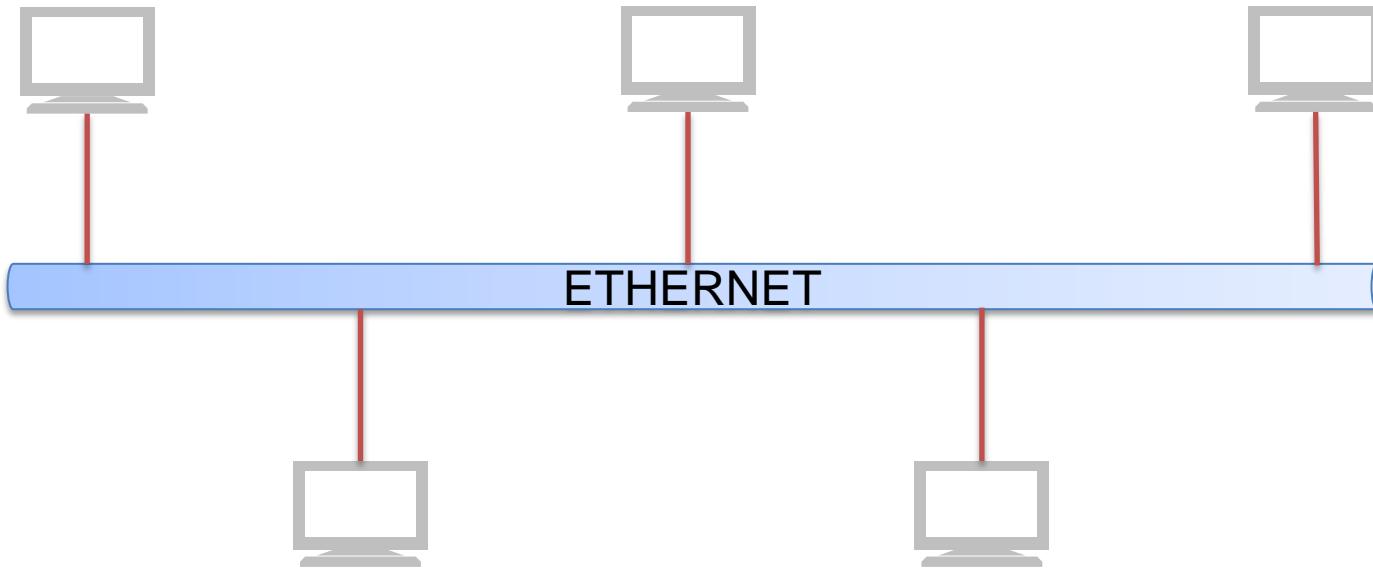
- Xerox사의 Palo Alto Research Center(PARC)에서 근무 했던 Robert Matcalfe(로버트 맷칼프)에 의해 개발 되었다.
- Ethernet은 짧은 거리에서 컴퓨터를 연결하기 위해 개발 되었으며, 당시의 맷칼프는 하와이 대학에서 하와이 섬을 연결하기 위한 무선 네트워크인 ALOHAnet에 착안하여 자신의 박사논문으로 다루게 되었다.
- 이더넷은 빛의 매질로 여겨졌던 Ether(에테르)에서 유래되었으며, 이는 데이터를 전송함에 있어서 또는 네트워크를 구성함에 있어서 어떠한 매질로도 할 수 있어 Ether-Net이라는 의미로 사용 되었다.

이더넷

# Ethernet History

## Chapter. 02 Data Link Layer

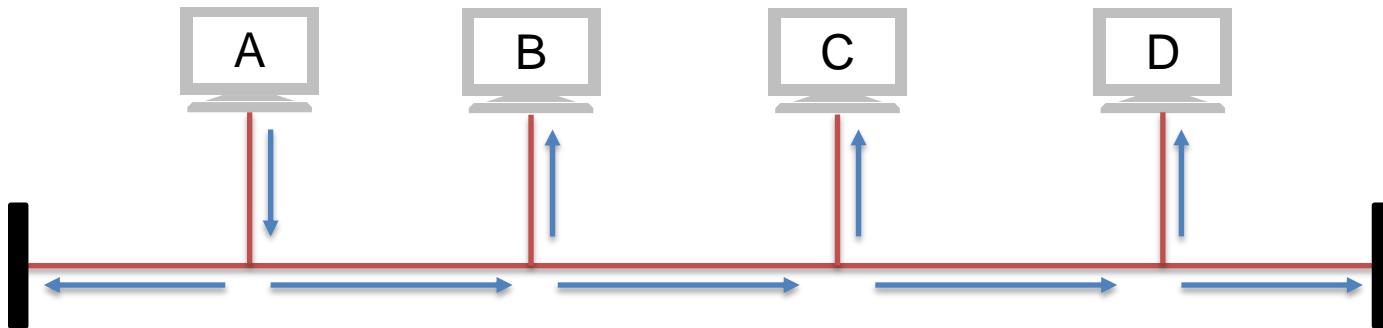




## 매체 접근 제어

- LAN에서 사용하는 가장 대표적인 네트워크 방식은 이더넷(Ethernet)이며, 전세계 LAN의 거의 대부분이 이더넷을 사용하고 있다.
- 이더넷은 하나의 물리적인 전송 매체를 여러 단말기가 공유해야 하기 때문에 Multiple Access 프로토콜이 필요했습니다.
- 여러 단말기가 공유 매체를 사용할 때 충돌이 발생하지 않도록 조절하거나 충돌 발생 시 조정을 하는 역할이 CSMA/CD(Carrier Sense Multiple Access/ Collision Detection) 프로토콜이다.
- 초기 이더넷은 두꺼운(thick) 동축 케이블을 이용한 10BASE5 방식을 이용하여 구축하였다.

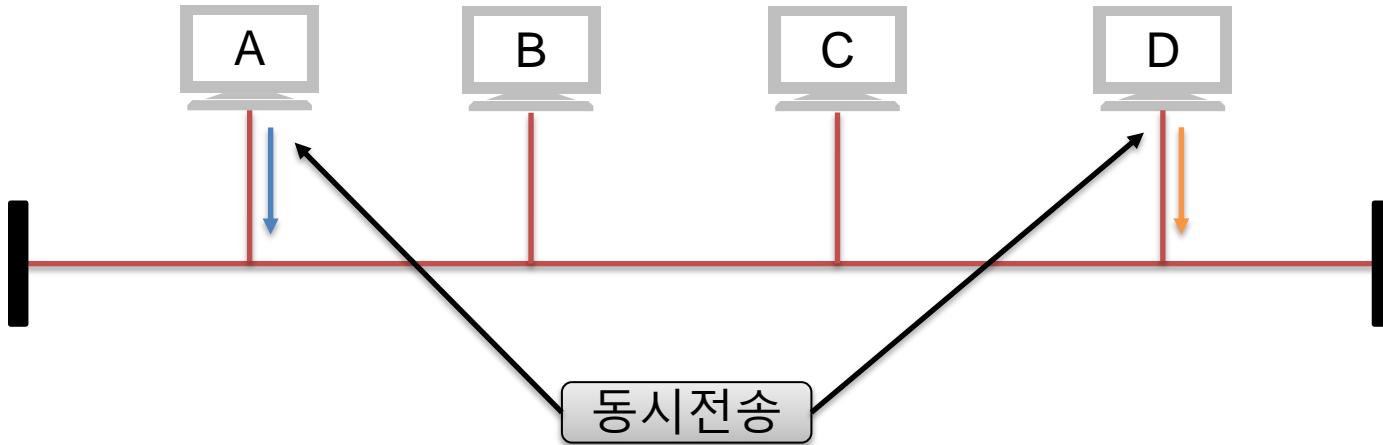
**MAC(Media Access Control)**



## 전체 매체 신호 감지

- Ethernet 환경에서 PC들은 전송하기 전에 다른 장비의 Carrier 상태를 확인 한다. 다른 장비의 Carrier를 감지 하면, 전송을 하기 전에 지정된 시간 동안 기다릴 다음 다시 시도한다.
- 다른 장비의 Carrier가 감지 되면 9.6micro 초의 IFG(Inter-Frame Gap) 시간 동안 대기한 다음에 전송 매체가 비어 있으면 프레임을 송신하고
- IFG는 Ethernet 반이중(Half-duplex)의 통신 상태에서 각 단말기 전송 가능할 때 바로 데이터를 송출하지 않고 일정 대기하는 시간 간격
- 한 노드가 프레임을 연속적으로 전송하면 채널독점 현상이 발생함
- 하나의 Frame 전송을 완료한 직후 연속하여 Frame을 전송하지 않도록 하는 규칙

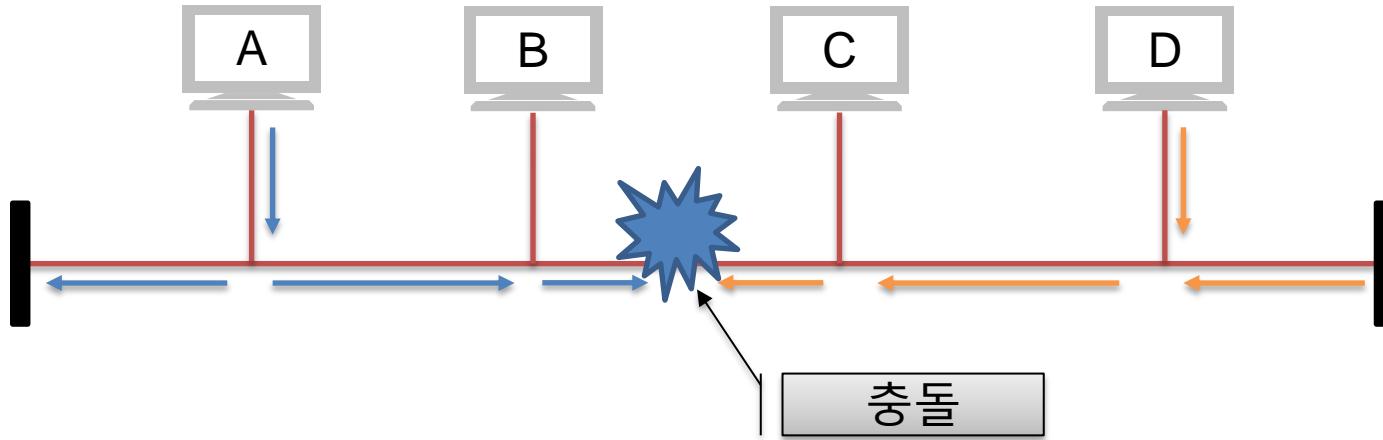
**CS(Carrier Sense)**



## 동시 전송

- A PC 와 D PC는 전송 매체의 상태를 체크한 결과 신호가 없음을 감지하고, 동신에 데이터를 전송한다.
- 하나의 전송 매체를 통해서 여러 노드에서 동시에 통신을 시도할 경우 동축케이블 및 허브를 사용하는 경우에는 충돌이 발생한다. 이 충돌에 대한 문제를 해결하기 위해서 유선 네트워크는 CSMA/CD 솔루션을 사용 한다.

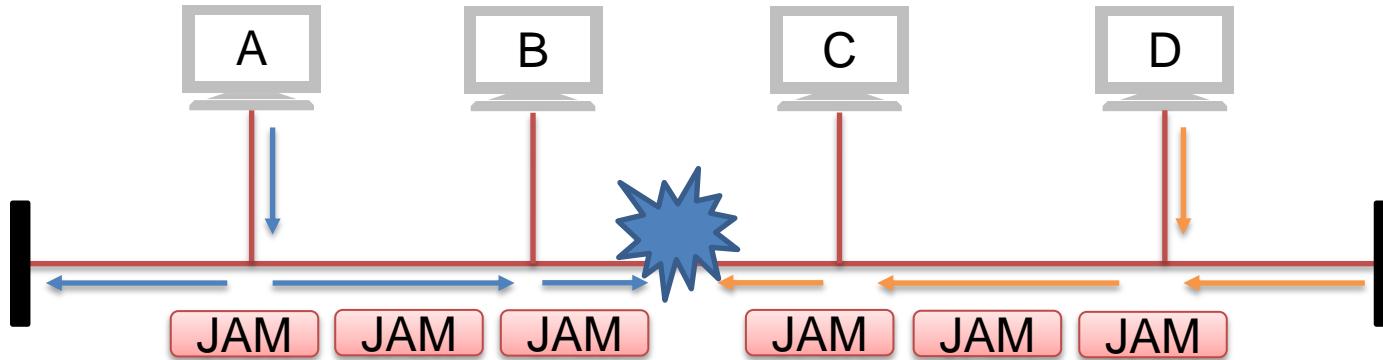
**MA(Multiple Access)**



## 전송 중 충돌 감지

- 네트워크의 디바이스의 NIC(Network Interface Card)은 보통 두 가지 이상의 신호가 충돌되면 신호 에너지가 거의 두 배 이상이 되므로 충돌 여부를 판단 할 수 있다.
- 송신 노드가 자신의 보낼 수 있는 신호 보다 높은 전압이 케이블에 나타나면 이를 충돌로 감지함.
- 너무 멀리 떨어져 있는 노드 간에는 수신 받는 신호가 약해 지기 때문에 충돌시 발생하는 에너지 보다 낮은 전압을 갖을 수 있으므로 전송 매체의 길이 제한된다.

**CD(Collision Detection)**

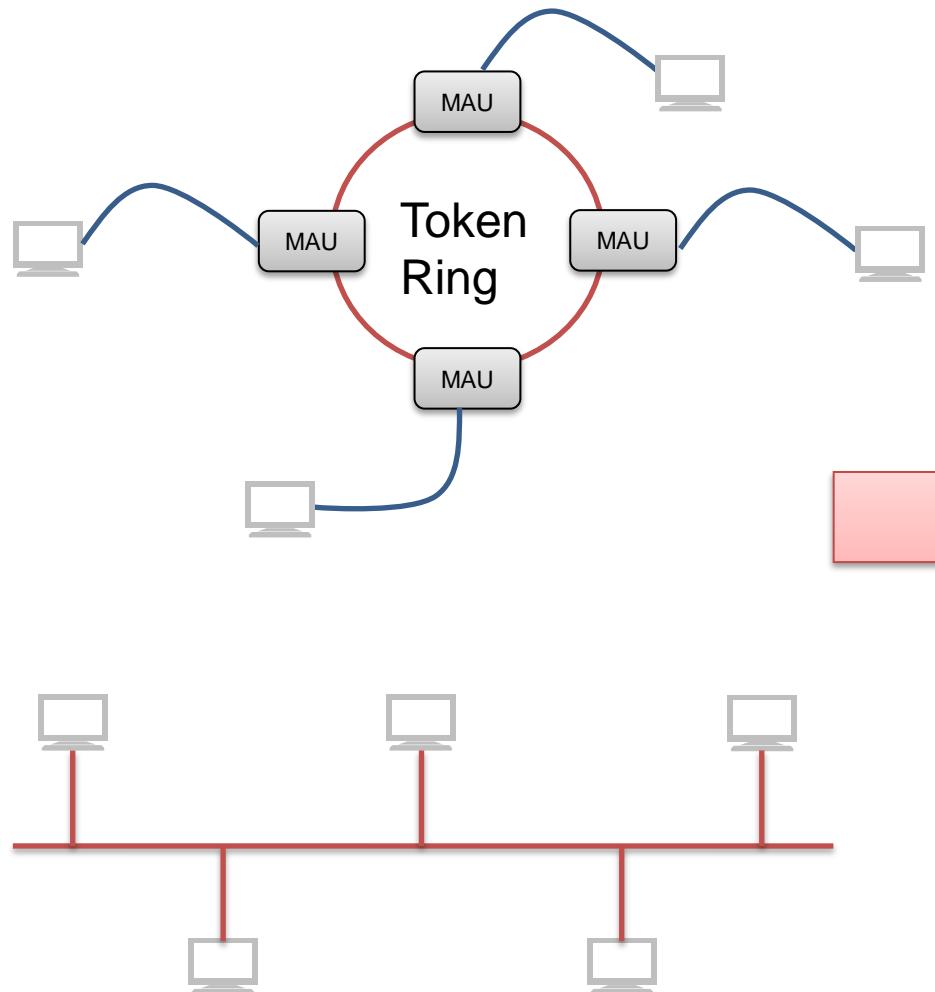


## 전송 중 충돌 감지

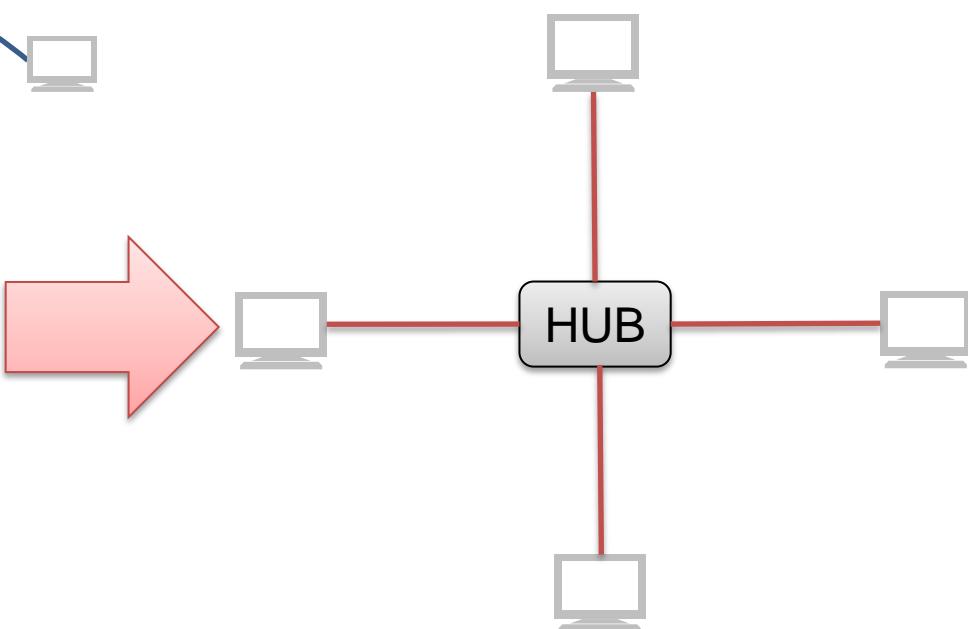
- 충돌이 발생하면 32비트 시간 동안 재밍(Jamming)신호를 모든 디바이스에 전송한다.
- 재밍 신호를 받은 디바이스는 백오프(Backoff) 지연시간 동안 통신을 중단한다.
- 백오프 시간이 만료되면 다시 신호를 감지하여 재전송을 시도한다.
- Backoff는 영어 뜻으로는 뒤로 물러서다라는 의미를 갖는 용어로 재전송하기 전에 임의 지연 대기 시간을 뜻함.

**CD(Collision Detection)**

# Ethernet Topology

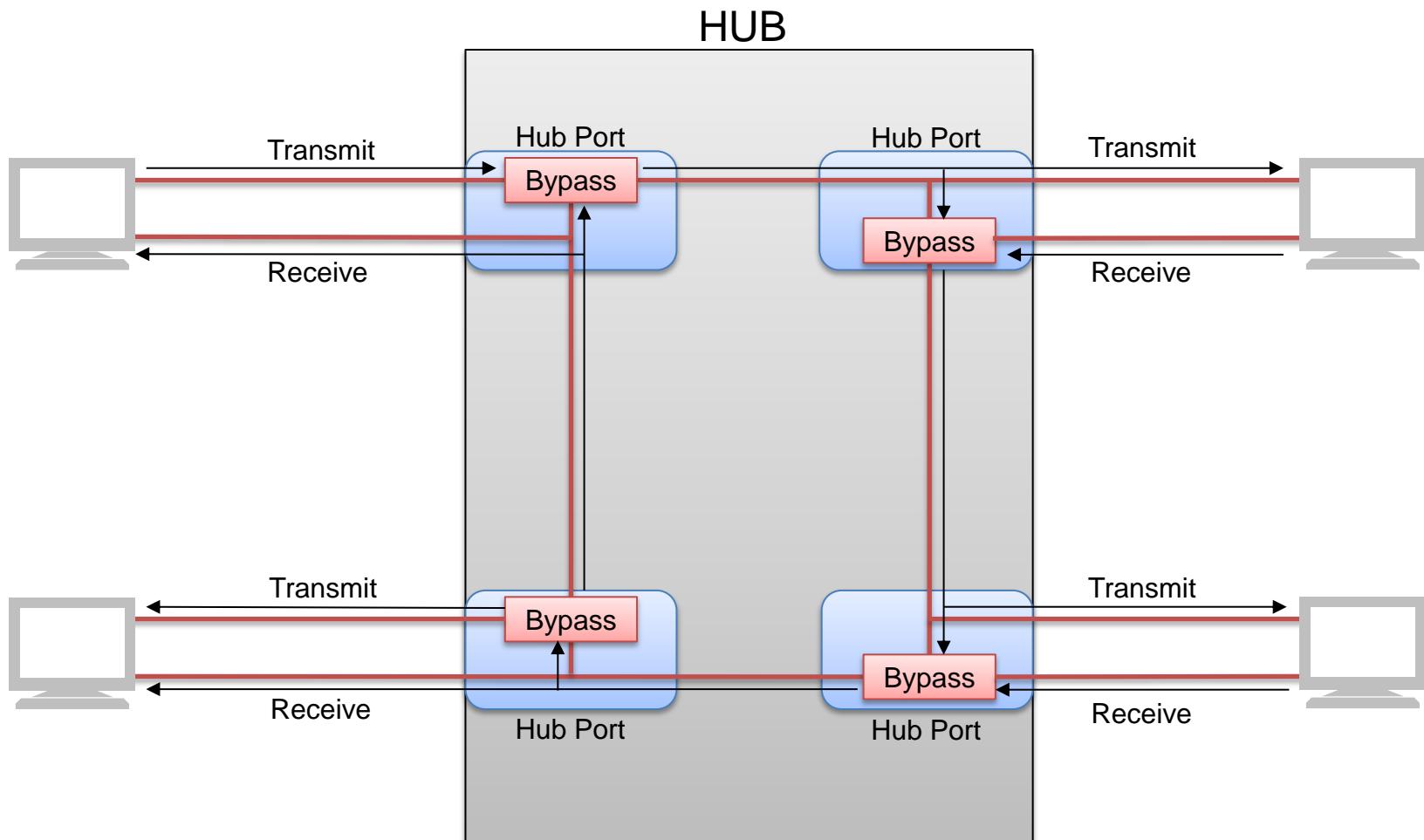


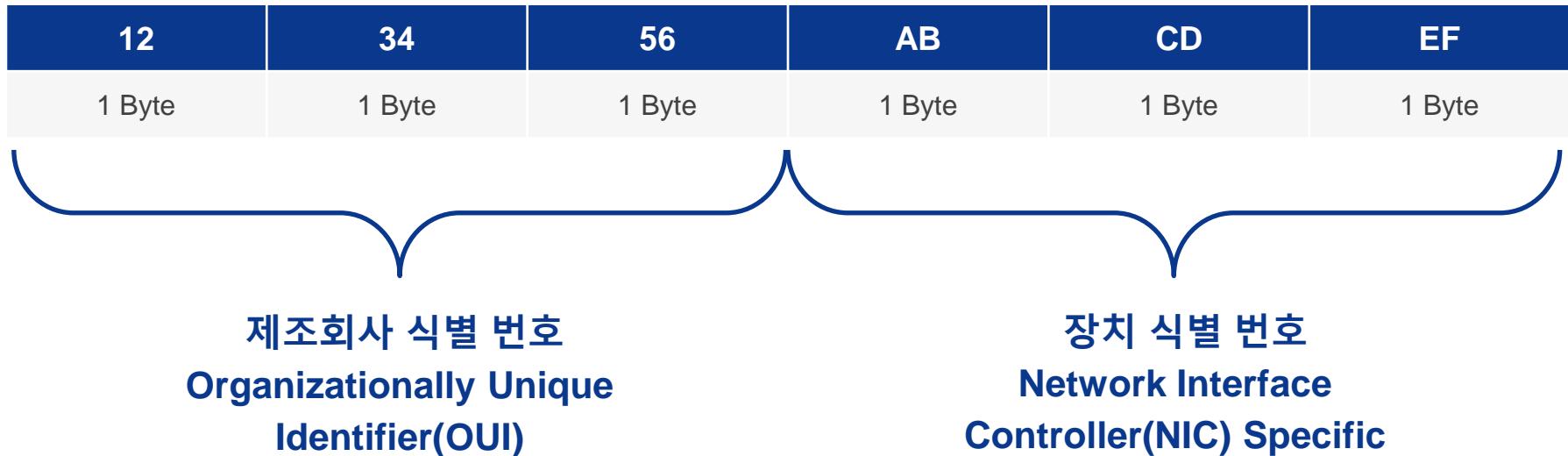
STAR Network



BUS Network

# Loop Hub Works



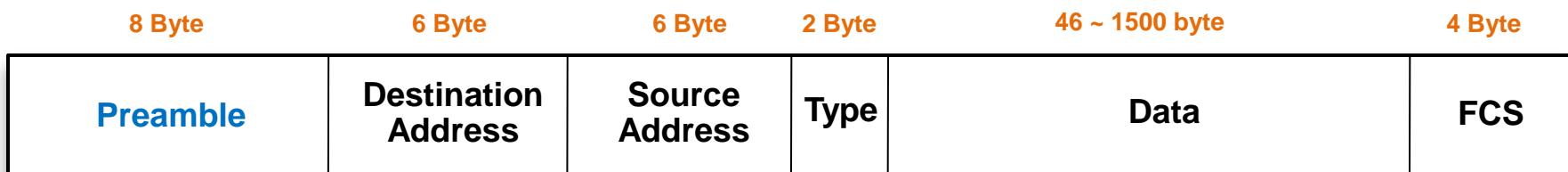


### MAC – EUI-48

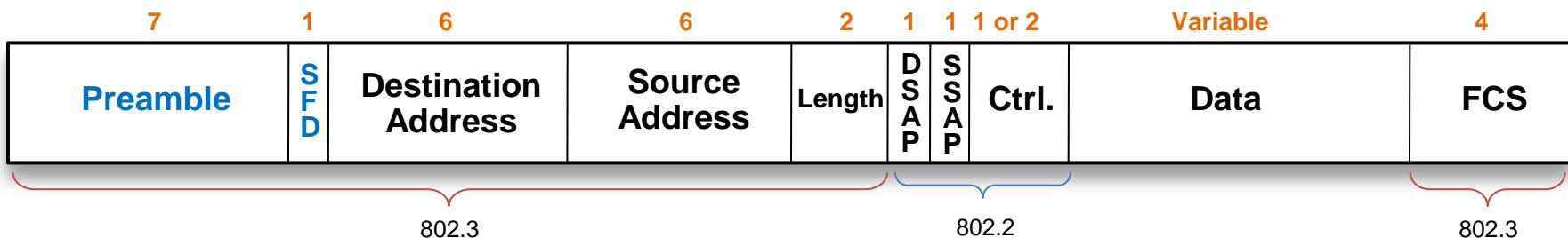
- NIC(Network Interface Card)라 불리는 장치는 별도의 식별 번호를 가지고 있으며, 이 식별 번호는 변경 할 수 없는 번호이다.
- 이 식별 번호를 통해서 모든 장치들은 자신이 처리해야 하는 Frame을 구분하게 된다.
- 만약 수신처의 주소가 FF:FF:FF:FF:FF:FF의 주소를 가지게 되면, 모든 장치들이 처리해야 하는 브로드캐스트 Frame이 된다.
- 이 식별 번호를 쉽게 구분하기 위해 16진수의 형태로 표기를 하며, 8 비트씩 ':'(콜론) 또는 '-'(하이픈) 등으로 구분 합니다.



- Ethernet II (DIX II) Frame



- IEEE 802.3 Frame



# 05

## Physical Layer

Cable / Signaling

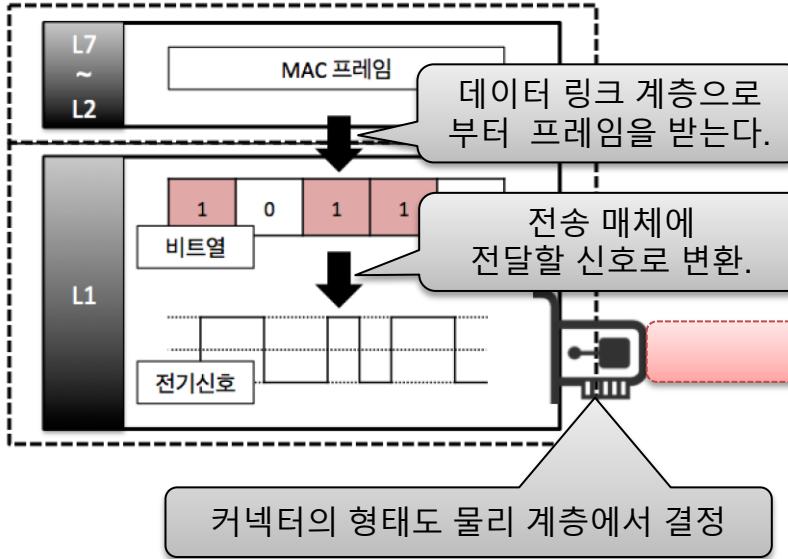
데이터를 전송하기 위한 여러 매체들이 있다.

사람들과 대화를 하기 위해서 공기라는 매체를 통해 음파를 전달 하는 것처럼 기계 장치들도 서로 대화를 하기 위한 매체가 필요하며, 음파처럼 특정 신호가 필요하게 된다.

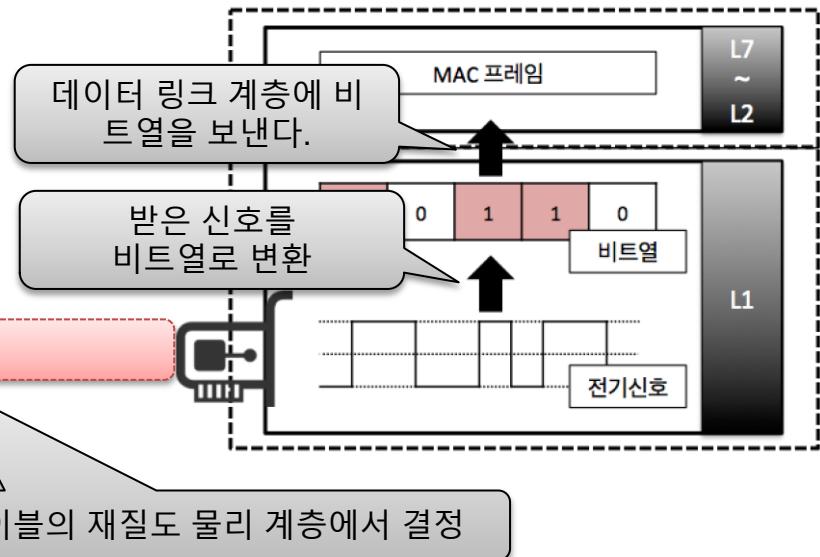




## PC#1(송신 측)



## PC#2(수신 측)



물리 계층은 '0'과 '1'로 되어 있는 데이터(프레임)를 데이터 링크 계층에서 받아 전기 신호와 광신호로 변환하여 전송매체(케이블)로 흐르게 하기 위한 규칙을 규정하고 있다.



## 실 전화기

수화

동축케이블

Twisted-Pair

광섬유

### ■ 실 전화기



어린 시절 많이 만들어 보았던 실 전화기 이다.

이 실 전화기로 친구와 서로 말을 주고 받을 수 있으며, 여기에서 실이 하나의 매체가 되며, 실의 진동은 Signal로 볼 수 있다.

- Media – \_\_\_\_\_
- Signal – \_\_\_\_\_



실전화기

## 수화

동축케이블

Twisted-Pair

광섬유

### ■ 수화



ㅅ ㅂ ㅁ ㅂ ㅋ ㅌ ㅍ



ㅎ ㅍ ㅌ ㅍ ㅊ ㅈ ㅇ



ㅏ ㅑ ㅓ ㅑ ㅗ ㅓ ㅓ



ㅐ ㅔ ㅒ ㅖ ㅣ ㅡ ㅠ

수화도 사람간에 대화를 할 수 있는 한가지의 방법이다.

수화의 경우 손의 모양을 인식 할 수 있는 눈이라는 매체가 있으며, 눈은 빛의 파장을 통해 신호를 뇌에 전달하게 된다.

• Media - -----

• Signal - -----



실 전화기

수화

## 동축케이블

Twisted-Pair

광섬유

### ■ 동축케이블



동축케이블(coaxial cable)

중앙의 절연된 구리선을 절연체가 둘러싸고 있는 전송 매체이다.  
동축이란 말은 전류가 흐르게 되는 도체의 축이 같기 때문에 온 말이다.

- Media – -----
- Signal – -----



실 전화기

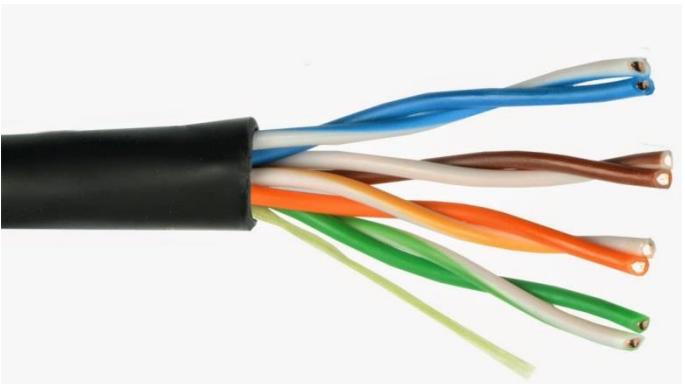
수화

동축케이블

Twisted-Pair

광섬유

## ■ Twisted-Pair



단말 장치간에 통신을 하기 위한 매체로 트위스티드-페어 케이블을 사용하게 된다.

이 케이블 안에는 구리선이 들어가 있어서 데이터를 전송 할 때 전기 신호를 통해 전송을 하게 된다.

- Media – -----
- Signal – -----



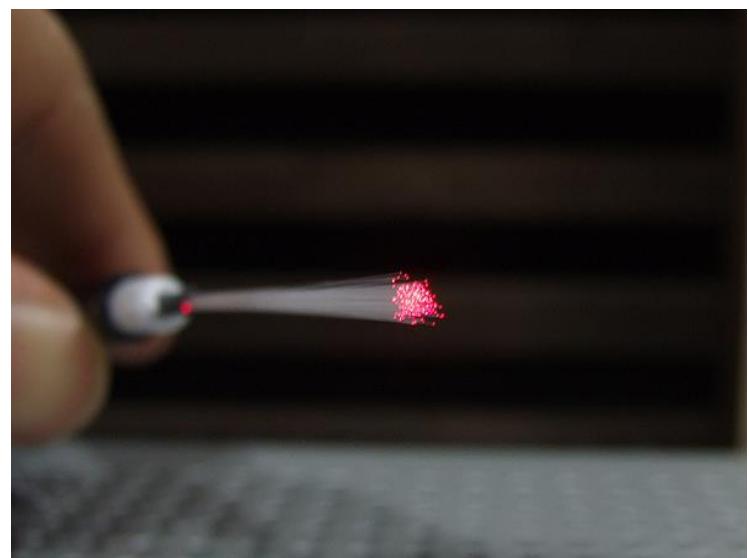
실전화기

수화

동축케이블

Twisted-Pair

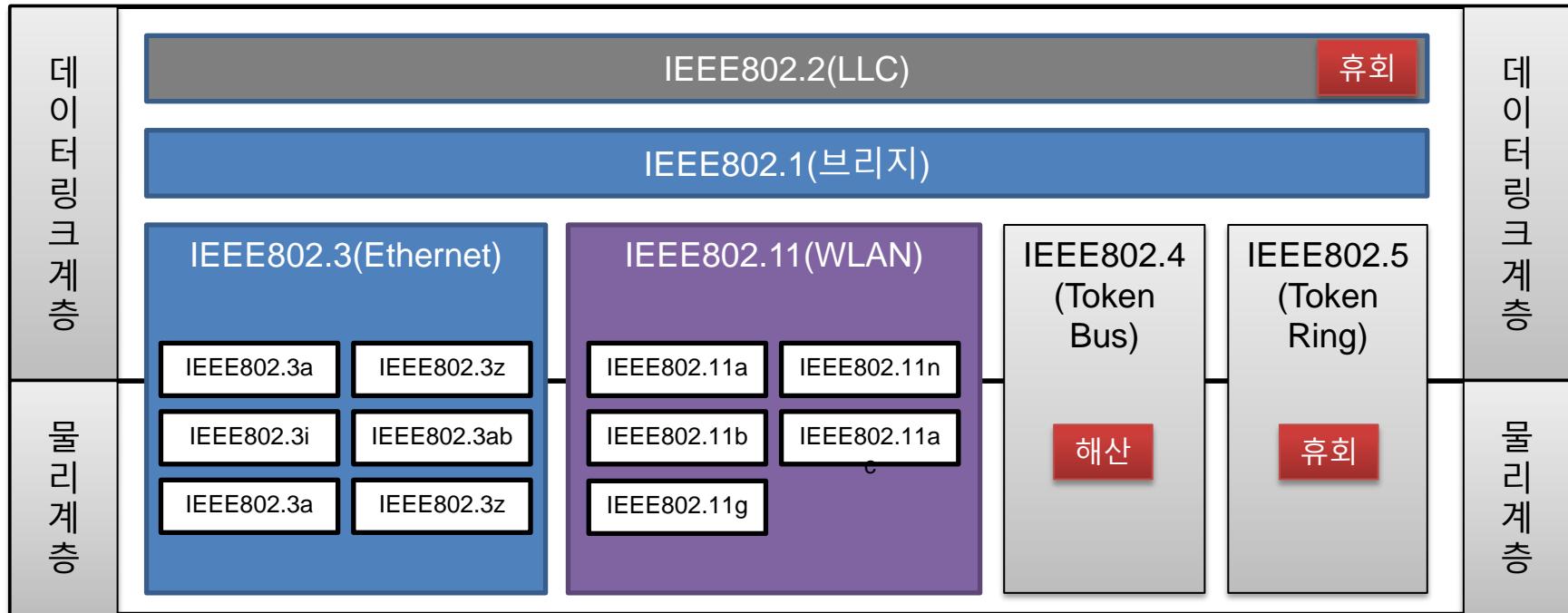
광섬유



단말 장치간에 통신을 하기 위한 또 다른 매체로 광 케이블도 있다.

이 케이블 안에는 광섬유가 들어가 있어서 데이터를 전송 할 때 빛 신호를 통해 전송을하게 된다.

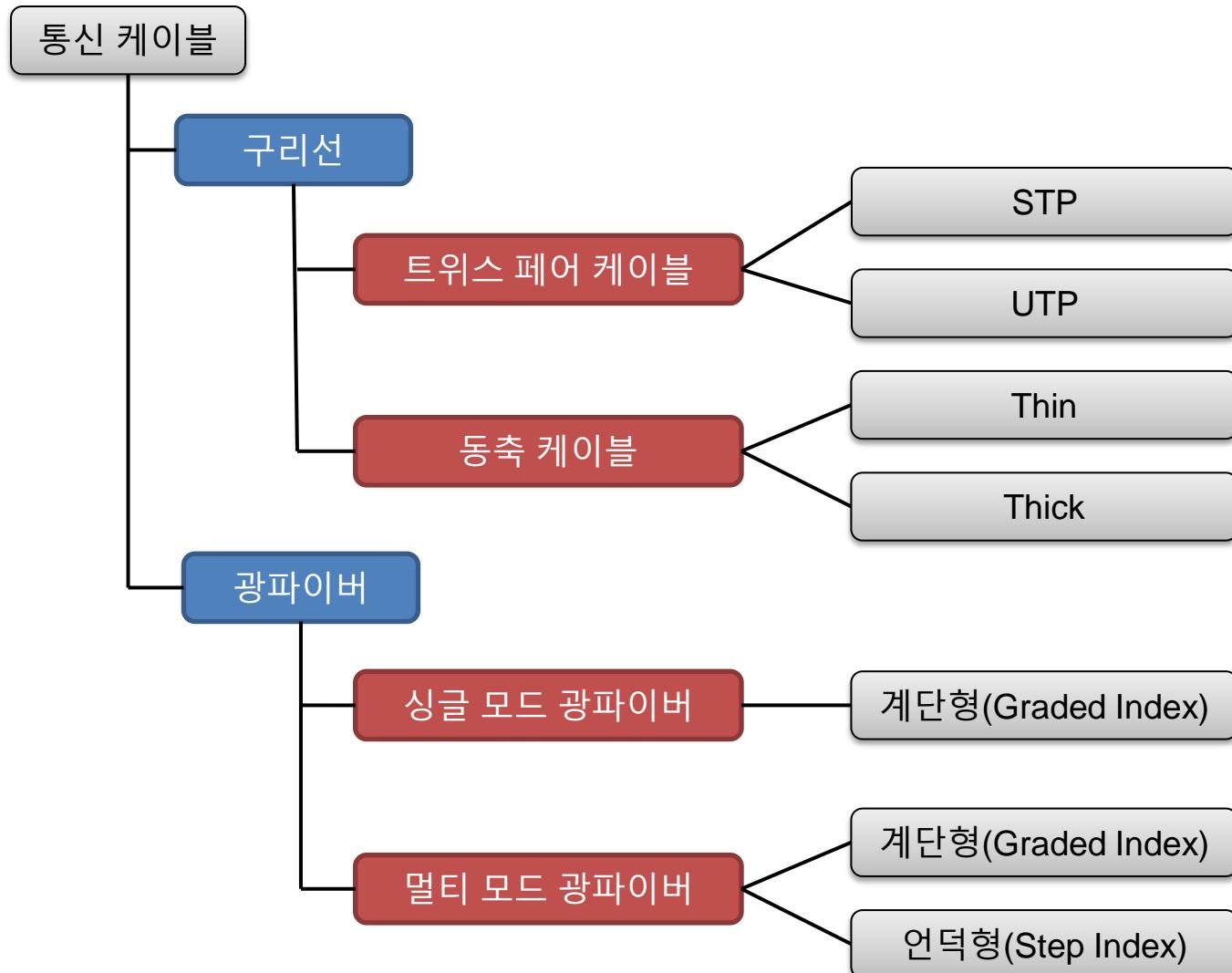
- Media – -----
- Signal – -----



IEEE802 위원회는 미국 전기전자학회(IEEE, Institute of Electrical and Electronics Engineers)에서 설립한 국제 표준화 단체다. 물리 계층과 데이터링크 계층에 관련한 기술의 표준화를 추진하고 있다.

# 이더넷(Ethernet) 케이블 규격

## Chapter. 02 Physical Layer



# 이더넷(Ethernet) 케이블 규격

## Chapter. 02 Physical Layer



		전송매체		레이저의 종류	
		동축 케이블	트위스트 페어 케이블	단파장 레이저	장파장 레이저
전송 매체	10Mbps	10BASE5 IEEE802.3	10BASE2 IEEE802.3a	10BASE-T IEEE802.3i	
	100Mbps			100BASE-TX IEEE802.3u	
	1Gbps			1000BASE-T IEEE802.3ab	1000BASE-SX IEEE802.3z
	10Gbps			10GBASE-T IEEE802.3an	10GBASE-SR IEEE802.3i



## xBASEy-z

[X]	전송속도
10	10Mbps
100	100Mbps
1000	1Gbps
10G	10Gbps

[y]	전송거리
2	10Mbps
5	100Mbps

[z]	케이블방식
T	트위스트페어
S	광(단파장)
L	광(장파장)

용어	설명
BASE	베이스밴드(Baseband)방식으로 디지털 데이터를 다른 주파수 대역으로 변조하지 않고 직류 펄스의 형태 그대로 전송하는 방식.
단파장	Short Wavelength(단파장)은 짧은 파장의 레이저를 사용하고 있으며, 멀티 모드 광파이버에만 사용할 수 있다.
장파장	Long Wavelength(장파장)은 긴 파장의 레이저를 사용하고 있으며, B 싱글 모드에서 주로 사용하며, 멀티모드에서도 사용이 가능하다.
bps	Bit per second(비피에스) 컴퓨터 모뎀과 전송매체의 데이터 속도를 나타내는 일반적인 척도이다.



## 10BASE2

### ■ Thinnet



## 10BASE5

IEEE 802.3 으로 표준화된 구내 정보 통신망(LAN) 전송로 규격의 하나로 전송 속도가 10Mbps, 신호 방식이 기저대역(baseband), 세그먼트의 최대 길이가 185m 것을 말한다.

10Bbase2는 직경 4mm 정도의 가는 동축 케이블(Thin Cable)을 사용하여 설치가 용이하고, BNC 커넥터(Connector)를 사용하여 컴퓨터를 케이블과 직접 연결한다.



10BASE2

10BASE5

## ■ Thicknet



1977년 Xerox사의 PARC(Palo Alto Research Center)에서 처음 개발된 이더넷은 직경 10mm 정도의 두꺼운 동축 케이블(Thick cable)을 사용하였다.

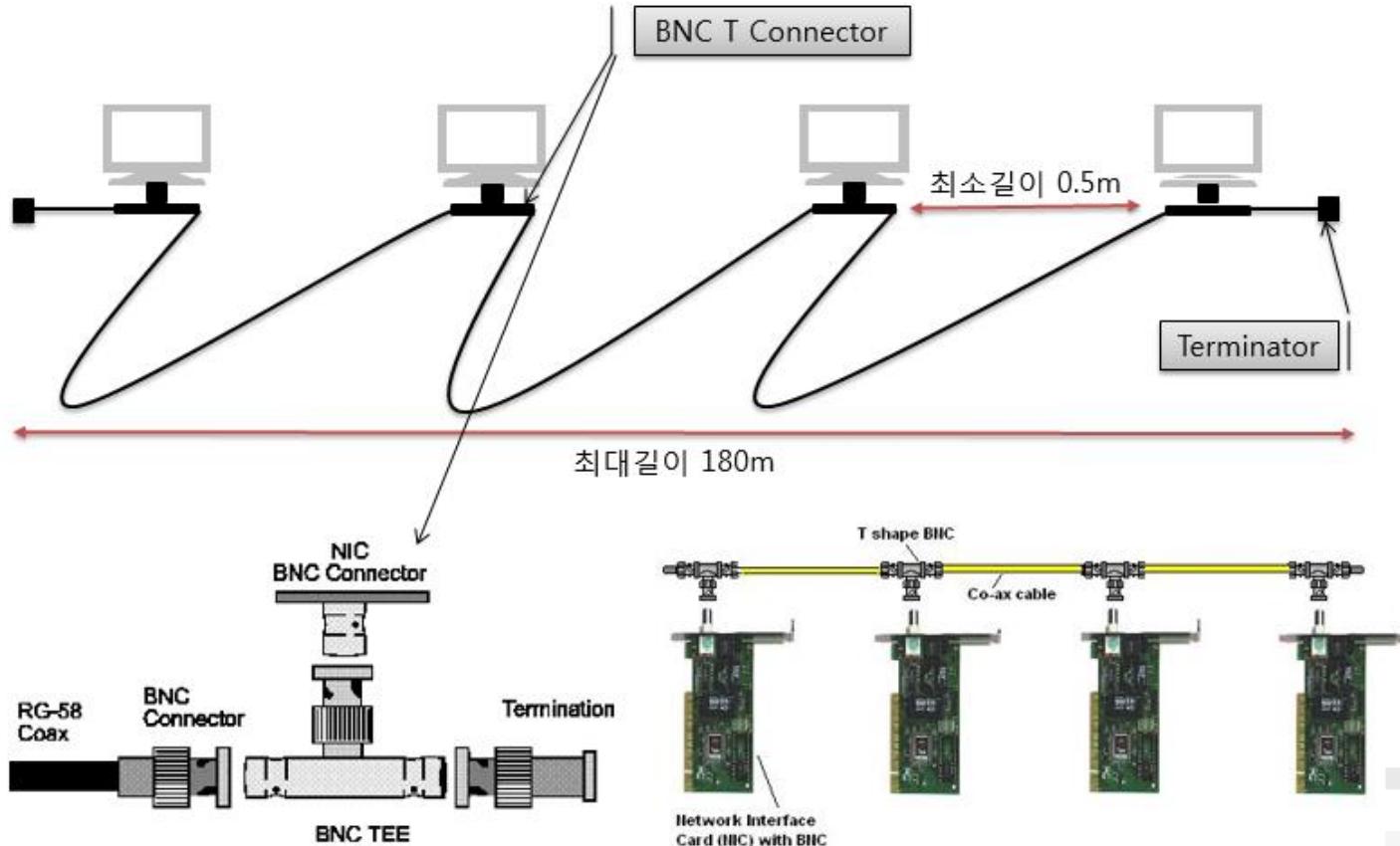
초기 이더넷은 10Base5로 불리어졌는데 이는 최대 길이 500m이 두꺼운 동축케이블을 사용하고 기저대역(baseband) 신호 전송 기법을 적용하여 10Mbps의 속도를 지원한다는 의미이다.



10BASE2

10BASE5

## ■ BNC T Connector/BNC Cable connector

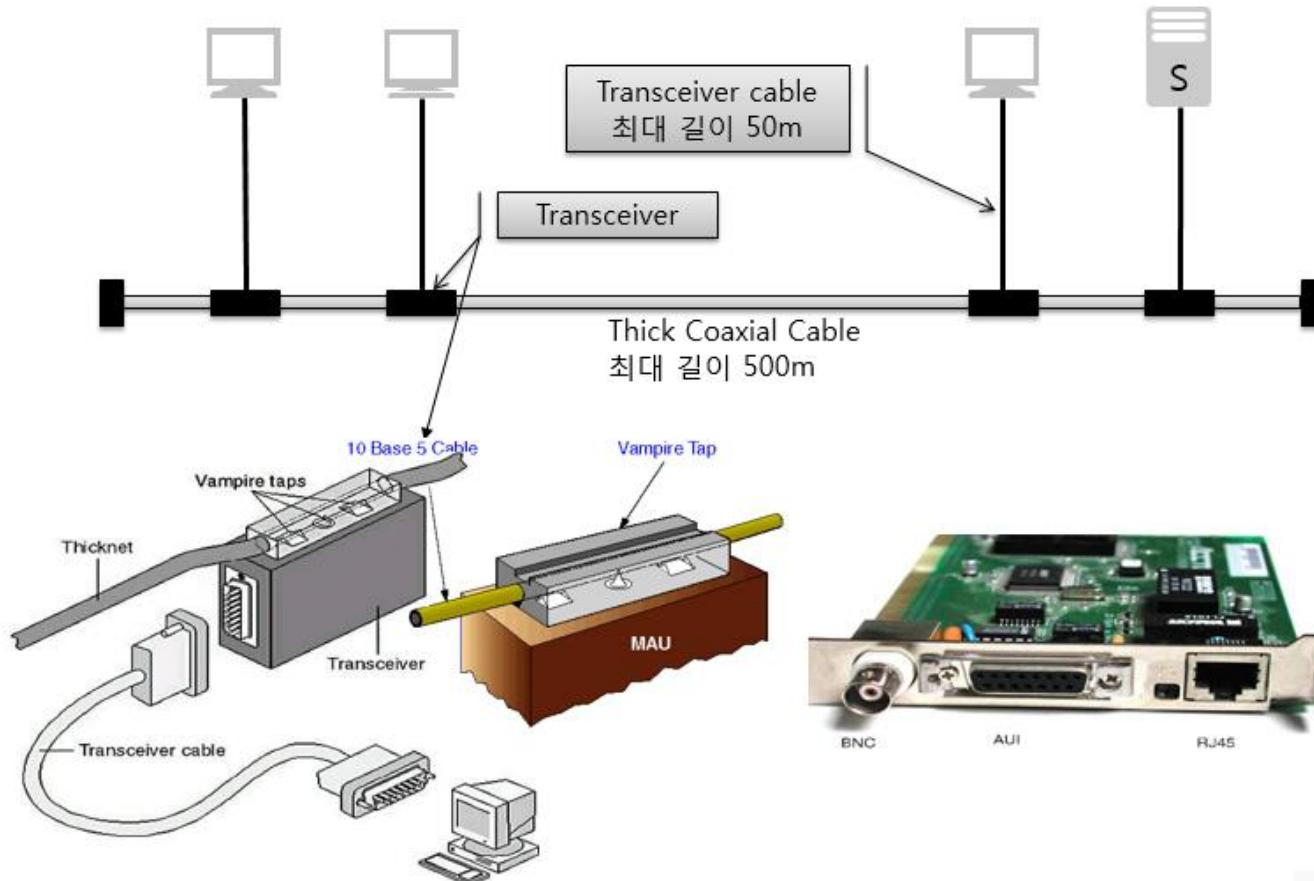




10BASE2

10BASE5

## ■ Transceiver/AUI Cable





## STP

## UDP

### ■ Shielding Twisted Pair

#### Shielded twisted pair (STP)



차폐연선은 실드막을 풀어 둔 연선이라고 불리며 STP(Shielded Twisted Pair)라고도 한다. 신호 간섭이 많은 공장이나 야외, 또 빠른 통신 속도가 필요한 곳에 쓰인다.

유럽에서는 STP가 주로 쓰이며 UTP는 거의 보급되어 있지 않다.

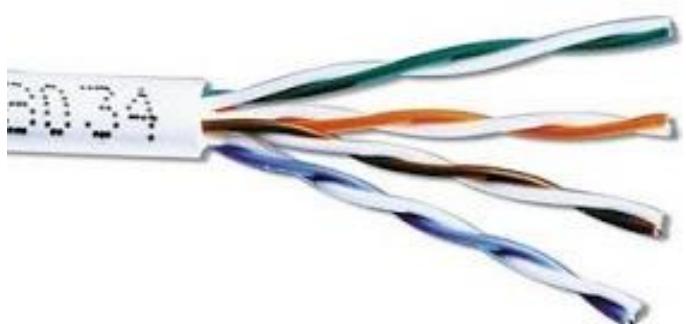


STP

UTP

## ■ Unshielded Twisted Pair

### Unshielded twisted pair (UTP)



비차폐연선은 실드 막을 풀지 않는 연선이라고 불리며 **UTP(Unshielded Twisted Pair)**라고도 한다. 이것은 전화선이나 이더넷(Ethernet) 등에 쓰인다.

처리가 간단하고 값이 싸서 빠른 전송이 필요 없는 이더넷(Ethernet)의 랜 용도에 표준으로 쓰이고 있다.

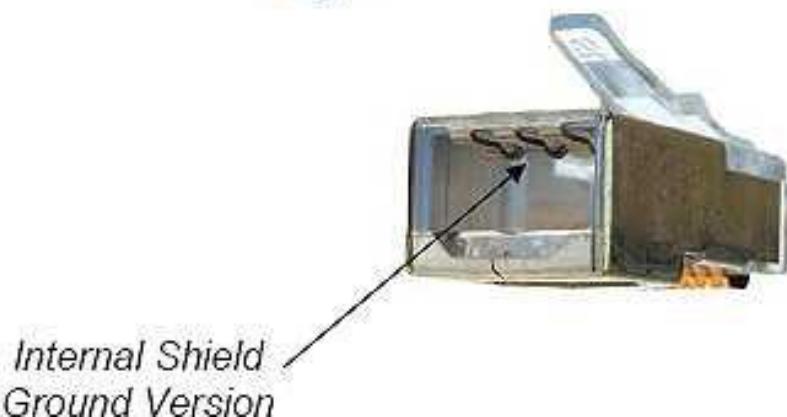


## STP RJ45 Connector

### ■ STP Connector



## UTP RJ45 Connector



*Internal Shield  
Ground Version*



STP RJ45 Connector

## UTP RJ45 Connector

### ■ UTP Connector



# UTP Cable Category

## Chapter. 02 Physical Layer



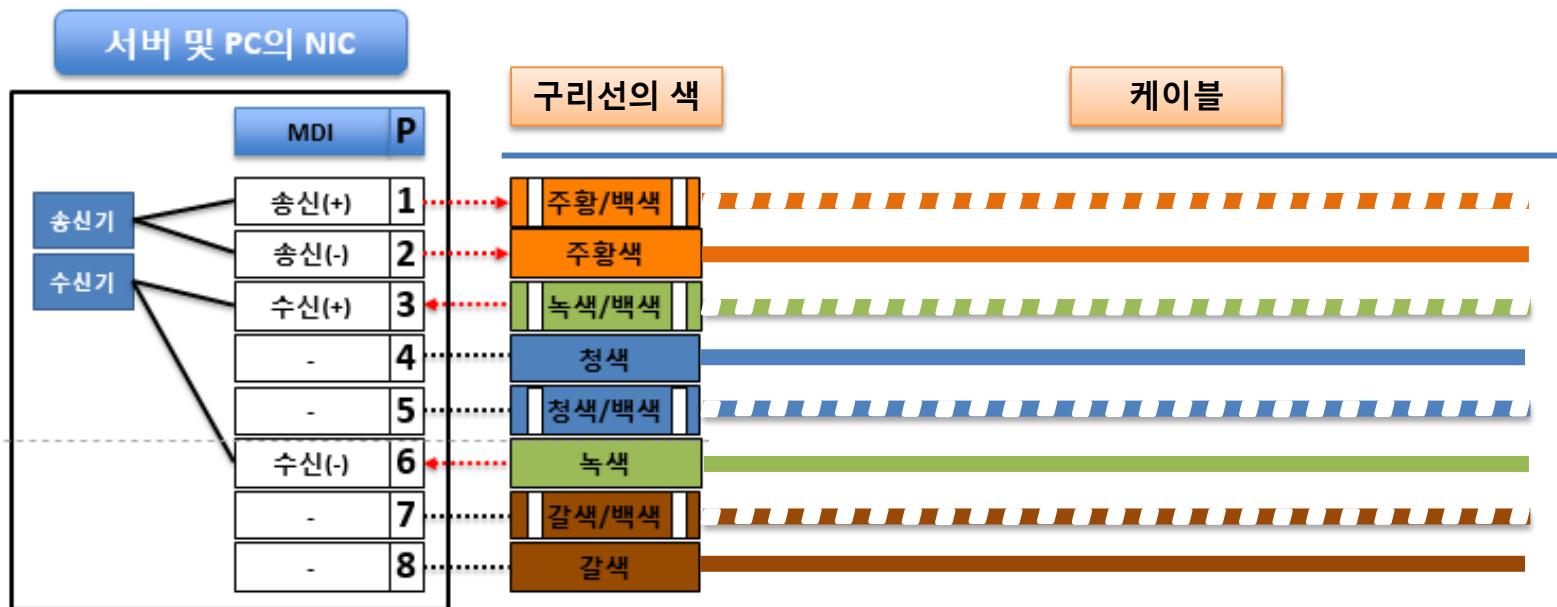
카테고리	종류	심수	대응주파수	주요대응규격	최대전송속도	최대전송거리
CAT3	UTP/STP	4심 2대	16Mbps	10BASE-T	16Mbps	100m
CAT4	UTP/STP	4심 2대	20MHz	Token Ring	20Mbps	100m
CAT5	UTP/STP	8심 4대	100MHz	100BASE-TX	100Mbps	100m
CAT5e	UTP/STP	8심 4대	100MHz	1000BASE-T	1Gbps	100m
CAT6	UTP/STP	8심 4대	250MHz	1000BASE-T 10GBASE-T	1Gbps 10bps	100m 55m(10GBASE-T)
CAT6A	UTP/STP	8심 4대	500MHz	10GBASE-T	10Gbps	100m
CAT7	STP	8심 4대	600MHz	10GBASE-T	10Gbps	100m
CAT7A	STP	8심 4대	1000MHz	10GBASE-T	10Gbps	100m



## MDI 포트

## MDI-X 포트

### ■ PC나 서버의 NIC의 MDI 포트

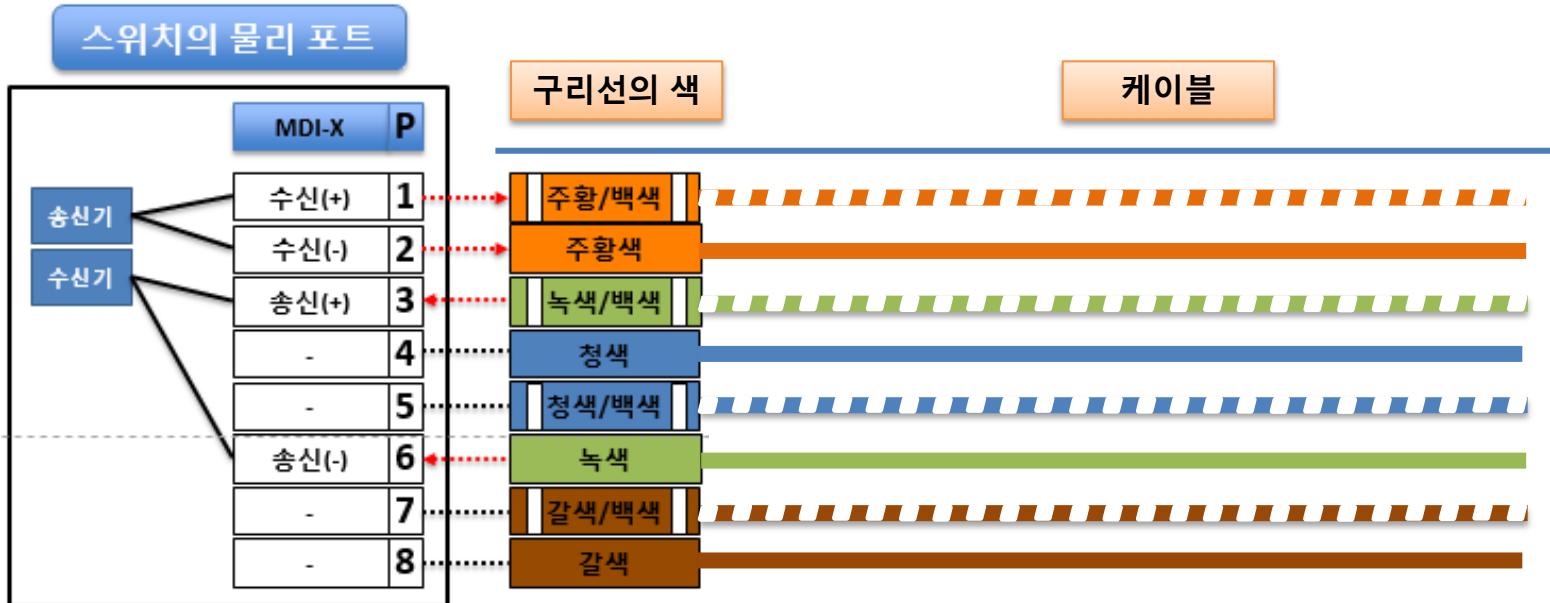




MDI 포트

MDI-X

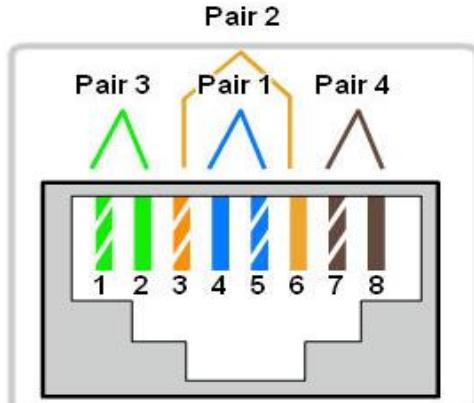
## ■ 허브나 스위치의 물리 포트



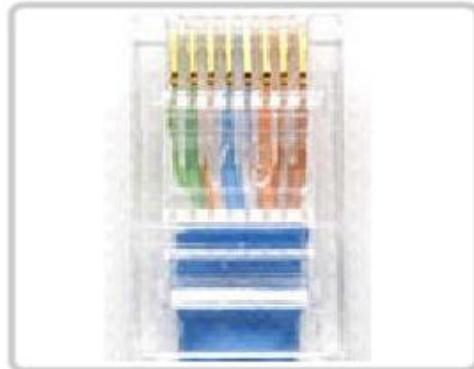
# Marking LAN Connections



## RJ45 T568A Termination

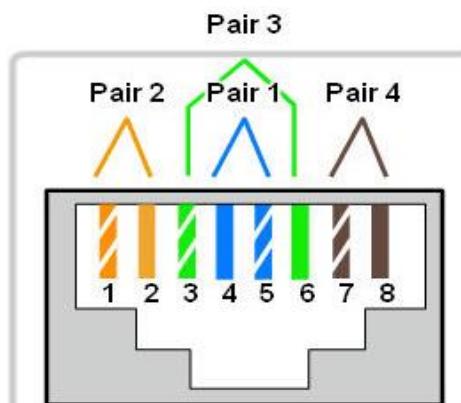


T568A

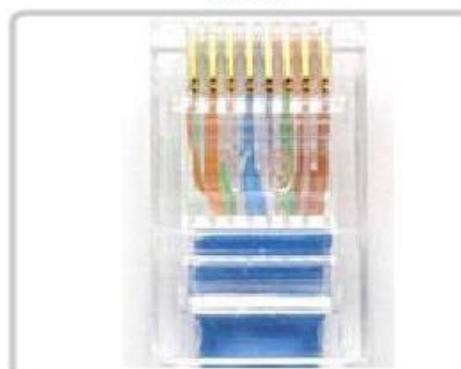


T568A  
(Top View)

## RJ45 T568B Termination



T568B



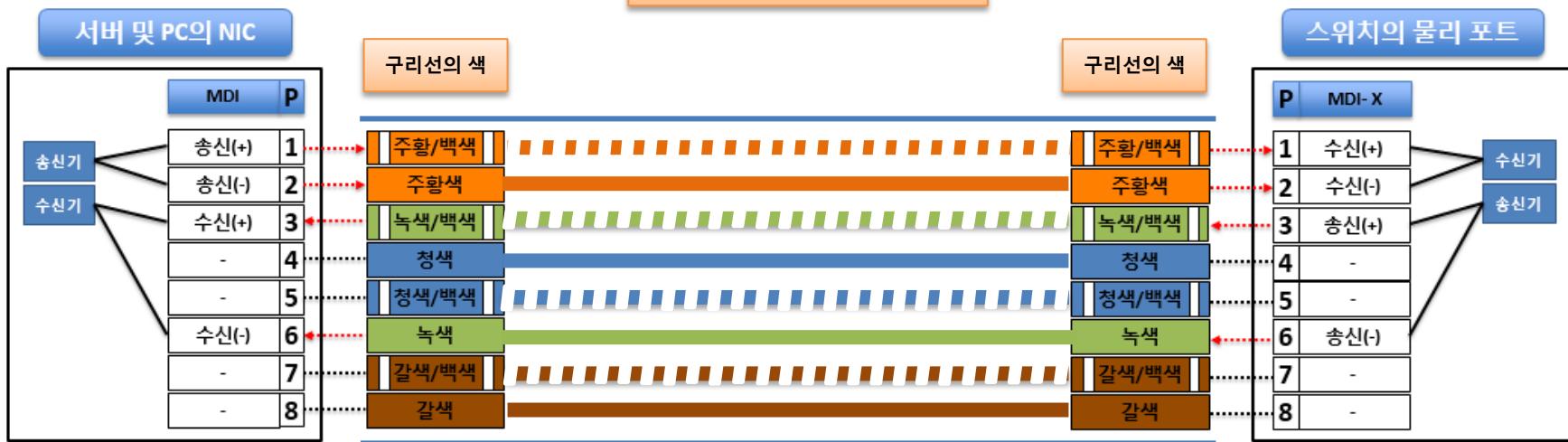
T568B  
(Top View)

# UTP Straight-Through Cable

Chapter. 02 Physical Layer



Straight-Through Cable



라우터



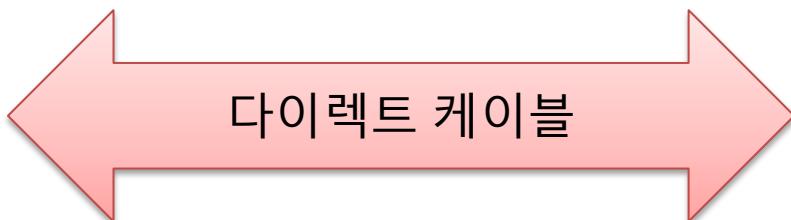
스위치

서버



허브

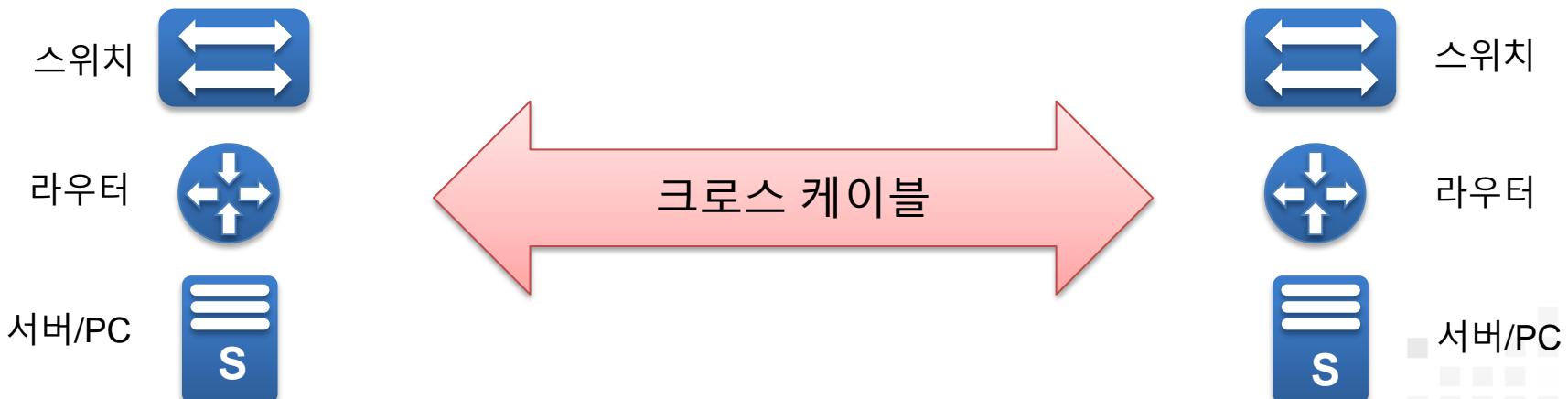
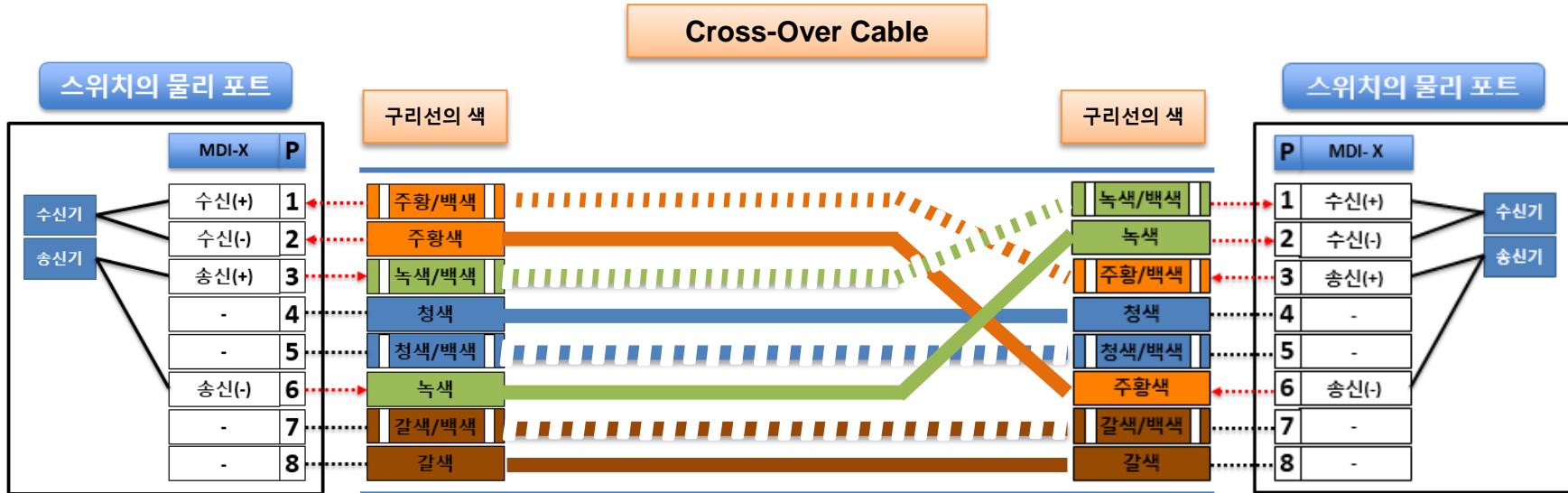
PC



다이렉트 케이블

# UTP Cross-Over Cable

Chapter. 02 Physical Layer

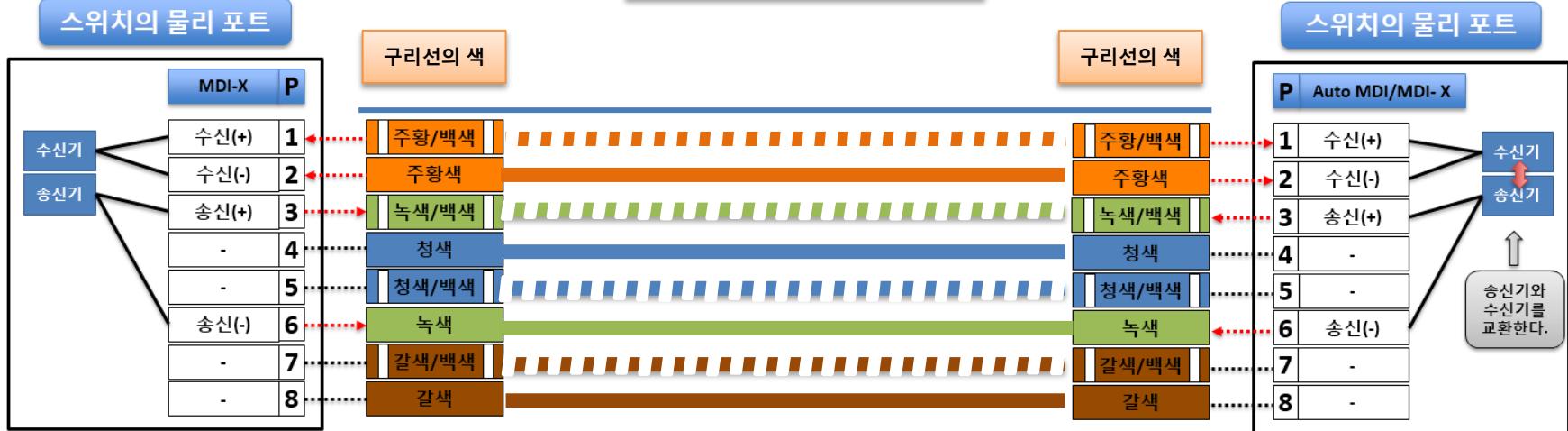


# UTP Cable Auto Detection

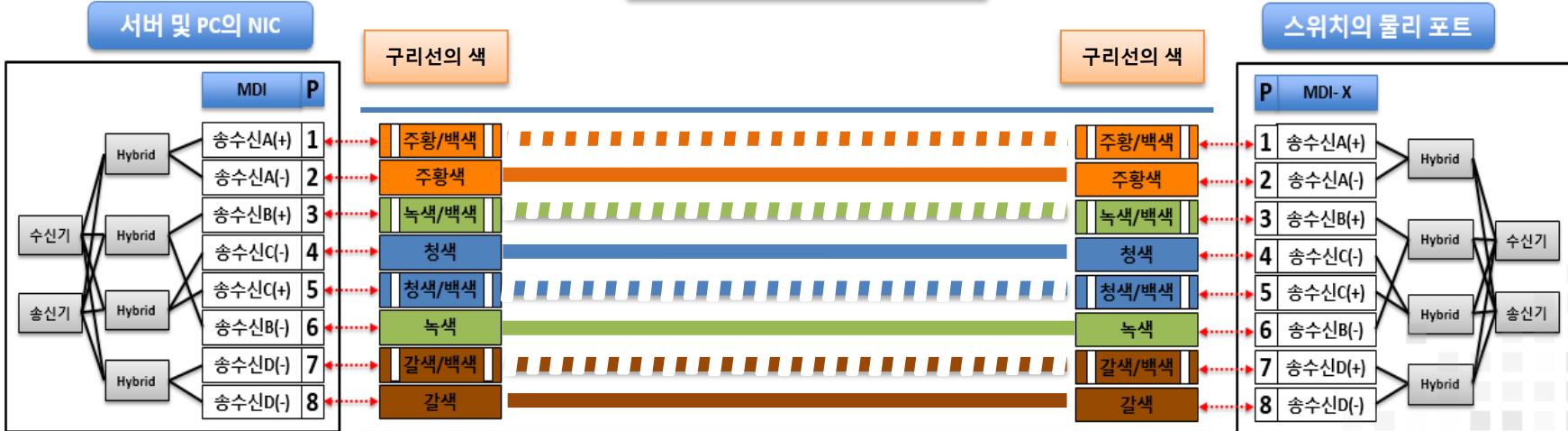
## Chapter. 02 Physical Layer



### AUTO MDI/MDI-X Port



### 1000BASE-T Port



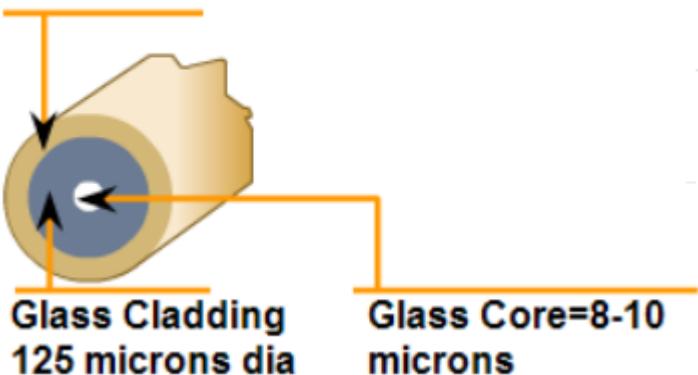


## Single Mode

## Multi Mode

### ■ Single-mode Fiber Media Mode

#### Polymeric Coating



코어 직경이 8-10 microns 정도로 광대역, 장거리 전송에 사용되며, 대략 50Km 까지 무중계 전송이 가능하다.

코어 직경이 작은 싱글모드의 경우 케이블 통로가 좁아 많은 양의 정보를 전달하기는 어렵지만 대신 먼거리까지 전송이 가능.



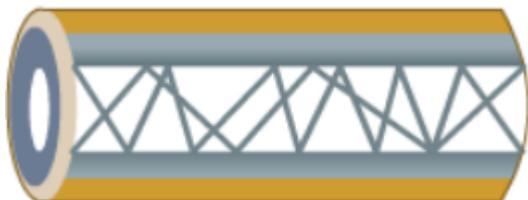
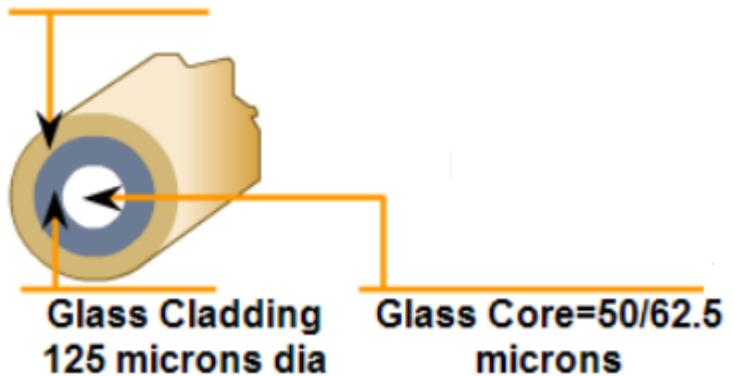


Single Mode

Multi Mode

## ■ Multi-mode Fiber Media Mode

### Polymeric Coating



코어 직경이 50-62.5 microns 으로 단거리에서 많이 쓰인다.

코어 직경이 큰 멀티모드의 경우 케이블의 통로가 넓기 때문에 많은 양의 정보를 전송 가능.



SC 형

FC 형

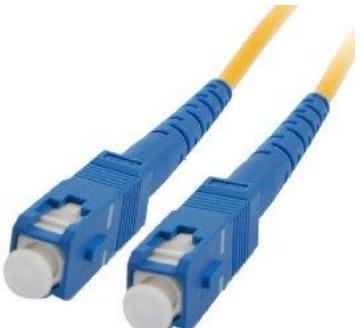
ST 형

MU 형

D4 형

LC 형

## ■ SC(Subscriber Connector)



SC 커넥터는 플러그를 밀어 넣는 것만으로 잠금 기능(lock)이 작동되고, 당기면 쉽게 분리되는 푸시 풀(push pull)구조의 커넥터다. 취급하고 쉽고, 저비용인 점이 특징이다. 다만 약간 플러그가 큰 것이 단점이다.

현재는 SC형 보다는 플러그가 작은 LC형을 더 많이 사용하고 있다.



SC 형

FC 형

ST 형

MU 형

D4 형

LC 형

## ■ FC(Face Connector)



원형으로 나사를 둘려서 고정, 스크류 나사식 탑입, 나사 캡 구조.

일본 Nippon사에서 개발 했으며, CATV 설비 연결에 주로 사용.



SC 형

FC 형

ST 형

MU 형

D4 형

LC 형

## ■ ST(straight Tip Connector)



원형으로 생겼으며, 접속시 빠지지 않도록 푸쉬후 돌려 접속하는 Push-pull 타입 및 BNC 형 타입이 있음.  
현재 잘 사용하지는 않음.  
Lucent사에서 개발



SC 형

FC 형

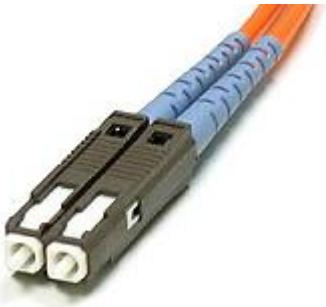
ST 형

MU 형

D4 형

LC 형

## ■ MU(Miniature Unit Connector)



각형으로 생겼으며 접속이 간편하고 크기가 작아 매우 많이 사용.  
현재는 LC형이 나온 이후론 잘 사용하지 않음



SC 형

FC 형

ST 형

MU 형

D4 형

LC 형

## D4형

광배선반 실장에 적합한 형태로 스프링이  
내장되어 있다.  
일본 NEC에 의해 개발





SC 형

FC 형

ST 형

MU 형

D4 형

LC 형

## ■ LC(Subscriber Connector)



RJ-45 SFP      Optic SFP



GBIC 모듈

LC 커넥터의 모습은 SC 커넥터와 비슷하다. 트위스트 페어 케이블의 커넥터(RF-45)와 동일하게 밀어 넣는 것만으로 잠금 기능이 작동되고, 작은 돌기(ratch)를 눌러 당기면 분리된다.

SC 커넥터보다 플러그가 작아서 많은 포트를 구현할 수 있다. SFP 모듈 및 SFP+ 모듈과 접속할 때 사용한다.

SFP(Small From Factor Pluggable)는 GBIC의 컴팩트 버전으로 Mini-GBIC으로 불리기도 한다. 또한 SFP 모듈은 RJ-45도 제공한다.

# Copper SFP Module RJ-45 Connector

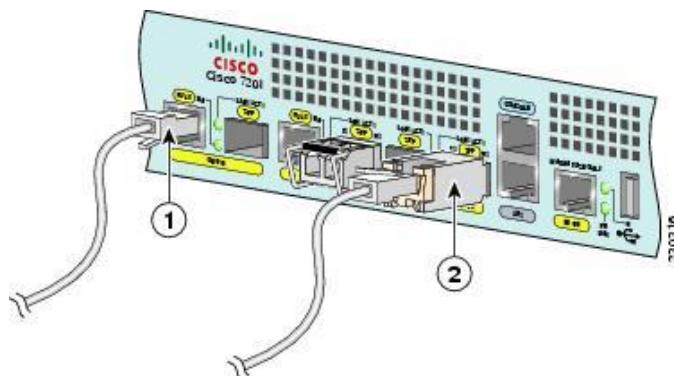
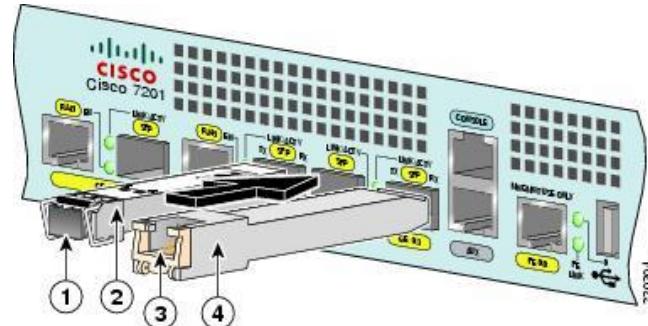
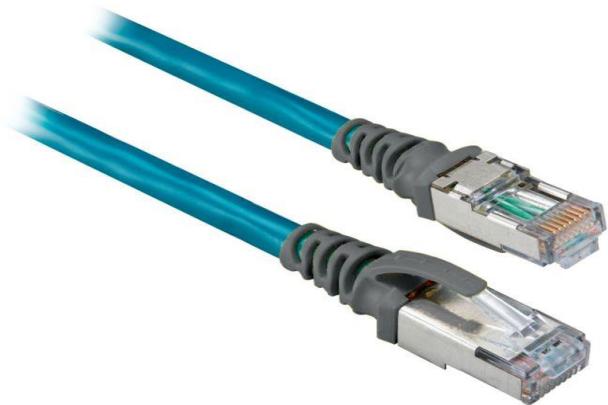
Chapter. 02 Physical Layer



RJ-45 Connector

Optical LC Connector

## ■ Copper SFP Module RJ-45 Connector



# Fiber Optical SFP Module LC Connector

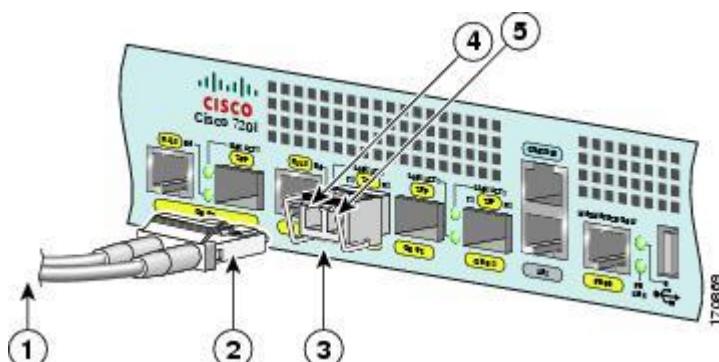
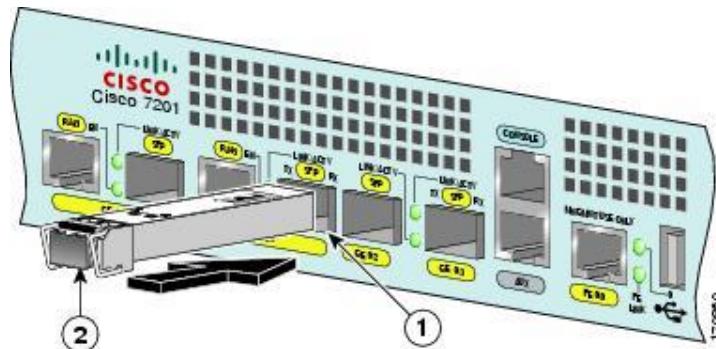


Chapter. 02 Physical Layer

RJ45 Connector

Optical LC Connector

## Fiber Optical SFP Module



# Fiber Optical Ethernet Type

## Chapter. 02 Physical Layer



이더넷 타입	케이블 타입	최대전송속도	Duplex	최대전송거리
100BASE-FX	Multi-mode	100Mbps	Half	400m
100BASE-FX	Multi-mode	200Mbps	Full	2km
1000BASE-SX	Multi-mode	1Gbps	Full	550m
1000Base-LX	Single-mode	1Gbps	Full	2km
10GBASE-LX4	Multi-mode	10Gbps	Full	300m
10GBASE-LX4	Single-Mode	10Gbps	Full	10km

# Fiber Optical Standard

## Chapter. 02 Physical Layer



케이블타입	표준	최대 속도	최대 거리	최대 대역폭	연결방식	용도
MMF (62.5/125)	OM1	1Gbps	300m	200MHz	LC, SC, ST, MPO	FDDI, Ethernet
MMF (50/125)	OM2	1Gbps	500m	500MHz	LC, SC, ST, MPO	SNAs, High Speed Ethernet
MMF Laser Optimized (50/125)	OM3	10Gbps	300m	2GHz	LC, SC, ST, MPO	SNAs, High Speed Ethernet
		40Gbps	100m	2GHz		
		100Gbps	100m	2GHz		
MMF Laser Optimized (50/125)	OM4	10Gbps	550m	4.7GHz	LC, SC, ST, MPO	SNAs, High Speed Ethernet
		40Gbps	150m	4.7GHz		
		100Gbps	150m	4.7GHz		
SMF (9/125)	OS1	10Gbps	40km	Infinite	LC, SC, ST, FC,FJ, MPO	SANs, WANs, Telco