

네트워크 구성 요소 (CCNA)

1. 네트워크

- 정보 공유를 목적으로 시스템과 시스템들을 연결하여 구성한 망을 의미한다.
- 목적: 정보 공유
- 구성: 시스템과 시스템들을 연결
- 장점: 시간 단축, 비용 절감, 통합 운영 관리
- 단점: 보안성 취약 (정보유출/탈취, 서버공격, 악성코드 유포)

보안성 취약 사례:

- SQL injection
 - ID: 'or 1=1 --
 - PW: 1234
- 사회 공학적 기법을 이용한 'webchat.apk' 유포
 - 스팸문자(주소)
 - QR코드

2. 프로토콜(Protocol)

- 네트워크 환경에서 데이터를 전송할 때 전송 방법을 정의한 규약 및 도구를 의미한다.

No.	Time	Source	Destination	Protocol	Length	Info	Expression...	Clear	Apply	Save
94	2.265436	1/2.16.5.254	114.111.46.227	TCP	62	1980→80 [SYN] Seq=0 Win=65535 Len=0 MSS=1260 S				
95	2.271040	114.111.46.227	172.16.5.254	TCP	60	80→1980 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0				
96	2.271104	172.16.5.254	114.111.46.227	TCP	54	1980→80 [ACK] Seq=1 Ack=1 Win=65535 Len=0				
97	2.271361	172.16.5.254	114.111.46.227	TCP	1314	[TCP segment of a reassembled PDU]				
98	2.271704	172.16.5.254	114.111.46.227	HTTP	983	GET /addAndList.nhn?r=linkedMember&cafeKey=12166211&ncmc4=6452b6988bc5a91b41afe8fff2017904f8ab56b5e0c629				
99	2.277605	114.111.46.227	172.16.5.254	TCP	60	80→1980 [ACK] Seq=1 Ack=1261 Win=7560 Len=0				
100	2.277634	114.111.46.227	172.16.5.254	TCP	60	80→1980 [ACK] Seq=1 Ack=2190 Win=10080 Len=0				
101	2.279901	114.111.46.227	172.16.5.254	HTTP	902	HTTP/1.1 200 OK (text/plain)				
102	2.279947	114.111.46.227	172.16.5.254	TCP	60	80→1980 [FIN, ACK] Seq=840 Ack=2190 Win=10080				
▣ Frame 98: 983 bytes on wire (7864 bits), 983 bytes captured (7864 bits)										
▣ Ethernet II, Src: RealtekS_14:62:ba (00:e0:4c:14:62:ba), Dst: Cisco_31:81:b1 (00:13:60:31:81:b1)										
▣ Internet Protocol Version 4, Src: 172.16.5.254 (172.16.5.254), Dst: 114.111.46.227 (114.111.46.227)										
▣ Transmission Control Protocol, Src Port: 1980 (1980), Dst Port: 80 (80), Seq: 1261, Ack: 1, Len: 929										
▣ [2 Reassembled TCP Segments (2189 bytes): #97(1260), #98(929)]										
▣ Hypertext Transfer Protocol										
▣ GET /addAndList.nhn?r=linkedMember&cafeKey=12166211&ncmc4=6452b6988bc5a91b41afe8fff2017904f8ab56b5e0c629										
Accept: */*\r\n										
Accept-Language: ko-KR\r\n										
Referer: http://cafe.naver.com/common/flash/ajax.swf\r\n										
x-flash-version: 11,7,700,224\r\n										
Accept-Encoding: gzip, deflate\r\n										
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLF										
Host: lm3.cafe.naver.com\r\n										
Connection: Keep-Alive\r\n										
▣ [truncated]Cookie: npic=+uGiqtLaxTt6qfx4Ti bo1QhEKJmb3OGiaj5TiI1Akq3G6mE3Tns3dxGVUekI8xh4CA==; NNB=Z3BK\r\n										
▣ Full request URI [truncated]: http://lm3.cafe.naver.com/addAndList.nhn?r=linkedMember&cafeKey=12166211										
[HTTP request 1/1]										
[Response in frame: 101]										

글자 정보를 가지고 있는 HTTP(HyperText Transfer Protocol)를 전송하기 위해서 TCP, TP, Ethernet 프로토콜 3개를 사용한다.

- 글자 정보를 가지고 있는 물건을 배송하기 위해 박스3개를 사용한다.

Filter: arp

No.	Time	Source	Destination	Protocol	Length	Info
230	3.052446	Ubiquoss_a1:3c:02	Broadcast	ARP	60	Who has 61.42.166.8/?
236	3.113310	Ubiquoss_a1:3c:02	Broadcast	ARP	60	who has 61.42.150.127?
294	3.166989	AsrockIn_22:a0:15	Broadcast	ARP	60	who has 172.16.2.51?
307	3.312722	Ubiquoss_a1:3c:02	Broadcast	ARP	60	who has 61.42.166.43?
351	3.687822	RealtekS_13:dd:7b	Broadcast	ARP	60	who has 172.16.4.246?
352	3.688930	AsustekC_11:63:27	Broadcast	ARP	60	who has 61.42.150.20?
357	3.722495	Ubiquoss_a1:3c:02	Broadcast	ARP	60	who has 61.42.150.87?
384	4.027719	AsrockIn_22:a0:15	Broadcast	ARP	60	who has 172.16.2.51?
705	4.028074	Ubiquoss_a1:3c:02	Broadcast	ARP	60	who has f1:47:150.127?

Frame 357: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)

Ethernet II, Src: Ubiquoss_a1:3c:02 (00:07:70:a1:3c:02), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

Address Resolution Protocol (request)

Hardware type	Ethernet (1)	key : Val
Protocol type	IP (0x0800)	
Hardware size	6	
Protocol size	4	

Opcode: request (1)

Sender MAC address: Ubiquoss_a1:3c:02 (00:07:70:a1:3c:02)

Sender IP address: 61.42.150.1 (61.42.150.1)

Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)

Target IP address: 61.42.150.87 (61.42.150.87)

) Data

arp라는 데이터는 Ethernet만을 사용해서 보낸다.

***전송하는 데이터마다 사용하는 프로토콜들이 정해져있다.

ETH | IP | TCP | HTTP

- HTTP 데이터를 전송할때는 TCP -> IP -> ETH 순서로 보내진다.

3. 인캡슐레이션(Encapsulation) -데이터 발신:

- 데이터를 전송하기 위해서 프로토콜 정보를 추가하는 패키지 과정을 의미한다.
- 프로토콜 정보를 더 많이 추가함으로서 보내는 장소가 정확해진다.
- 물건을 포장하기 위해 박스포장하는 과정
- Ex) ETH | IP | TCP | HTTP
 헤더 헤더 헤더
- Ex) HTTP 포장과정:
 - TCP | HTTP
 - IP | TCP | HTTP
 - ETH | IP | TCP | HTTP

4. 디캡슐레이션(Decapsulation) -데이터 수신:

- 데이터를 받을때 포장을 푸는 과정
- HTTP데이터를 받을때
 - ETH | IP | TCP | HTTP
 - IP | TCP | HTTP (ETH의 주소가 맞을경우 ETH 프로토콜 삭제)
 - TCP | HTTP (IP의 주소가 맞을경우 IP 프로토콜 삭제)
 - HTTP (TCP의 주소가 맞을경우 TCP 프로토콜 삭제)

**여기서 주소는 다음을 의미한다:

lo.	Time	Source	Destination	Protocol	Length	Info	Expression...	Clear	Apply	Save
394	4.143901	1/2.16.5.254	1.226.51.70	TCP	54	1994->80 [ACK] Seq=1 Ack=1 Win=65535 Len=0				
395	4.144164	172.16.5.254	1.226.51.70	HTTP	411	GET /201307/ed56487d-31be-4c77-baa2-b00b18a32f				
396	4.144249	172.16.5.254	1.226.51.70	HTTP	411	GET /201307/79636edb-460e-4b59-a698-bbd4a93a7f				
397	4.147798	1.226.51.70	172.16.5.254	TCP	60	80->1993 [ACK] Seq=1 Ack=358 Win=6432 Len=0				
398	4.147837	1.226.51.70	172.16.5.254	TCP	60	80->1994 [ACK] Seq=1 Ack=358 Win=6432 Len=0				
399	4.147848	1.226.51.70	172.16.5.254	TCP	1314	[TCP segment of a reassembled PDU]				
400	4.147866	1.226.51.70	172.16.5.254	TCP	1314	[TCP segment of a reassembled PDU]				
401	4.147891	172.16.5.254	1.226.51.70	TCP	54	1993->80 [ACK] Seq=358 Ack=2521 Win=65535 Len=0				
402	4.147951	1.226.51.70	172.16.5.254	TCP	1314	[TCP segment of a reassembled PDU]				
Frame 399: 1314 bytes on wire (10512 bits), 1314 bytes captured (10512 bits)										
Ethernet II, Src: Cisco_31:81:b1 (00:13:60:31:81:b1), Dst: RealtekS 14:62:ba (00:e0:4c:14:62:ba)										
Internet Protocol Version 4 Src: 1.226.51.70 (1.226.51.70), Dst: 172.16.5.254 (172.16.5.254)										
Transmission Control Protocol, Src Port: 80 (80), Dst Port: 1993 (1993), Seq: 1, Ack: 358, Len: 1260										
Source Port: 80 (80) Destination Port: 1993 (1993) [Stream index: 33] [TCP Segment Len: 1260] Sequence number: 1 (relative sequence number) [Next sequence number: 1261 (relative sequence number)] Acknowledgment number: 358 (relative ack number) Header Length: 20 bytes										

data

Ex) '1-1.프로토콜.pcap'에서 http, telnet, ssl, dns, icmp 의 헤더 구조와 헤더 크기를 확인한다.

No	Protocol
98	http
	ETH IP TCP http
	14 20 20
2081	telnet
363	ssl
55	dns
1359	icmp
9	arp

헤더 크기 보는법:

Filter: tcp							Expression...
No.	Time	Source	Destination	Protocol	Length	In	
394	4.143901	1/2.16.5.254	1.226.51.70	TCP	54	1	
395	4.144164	172.16.5.254	1.226.51.70	HTTP	411	G	
396	4.144249	172.16.5.254	1.226.51.70	HTTP	411	G	
397	4.147798	1.226.51.70	172.16.5.254	TCP	60	8	
398	4.147837	1.226.51.70	172.16.5.254	TCP	60	8	
399	4.147848	1.226.51.70	172.16.5.254	TCP	1314	[
400	4.147866	1.226.51.70	172.16.5.254	TCP	1314]	
401	4.147881	172.16.5.254	1.226.51.70	TCP	64	1	

```

Frame 399: 1314 bytes on wire (10512 bits), 1314 bytes captured
Ethernet II, Src: Cisco_31:81:b1 (00:13:60:31:81:b1), Dst: Router (00:0c:29:1d:00:00)
Internet Protocol Version 4, Src: 1.226.51.70 (1.226.51.70),
Transmission Control Protocol, Src Port: 80 (80), Dst Port: 1993 (1993)
Source Port: 80 (80)
Destination Port: 1993 (1993)
[Stream index: 33]
[TCP Segment Len: 1260]
Sequence number: 1 (relative sequence number)
Next sequence number: 1261 (relative sequence number)

```

0020	05 fe 00 50 07 c9 1f c6 b2 5d 35 a6 b3 cb 50 10 ..P.
0030	19 20 ab a3 00 00 48 54 54 50 2f 31 2e 31 20 32
0040	30 30 20 4f 4b 0d 0a 44 61 74 65 3a 20 54 75 65 00 OK.
0050	2c 20 30 39 20 4a 75 6c 20 32 30 31 33 20 30 35 , 09 .
0060	3a 31 37 3a 32 34 20 47 4d 54 0d 0a 53 65 72 76 :17:24

Transmission Control Protocol (tcp) [20 bytes] Packets: 2670 · Displayed: 613 (... | P)

2081 telnet:

ETH | IP | TCP | TELNET
14 20 20

363 ssl:

ETH | IP | TCP | SSL
14 20 20

Byte > Bit 변경법

Byte * 8 = Bit

십진수	이진수	8진수	16진수
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

4. 네트워크 유형

1) LAN(local Area Network)

- 내부 네트워크 (건물 안)
- 장비: 스위치, PC, 랜카드, UTP 케이블, 무선 AP
- 프로토콜: Ethernet
- 구축 방식: 버스 토플로지, 스타 토플로지
- 권장 연결: 스타 토플로지 + 이중화 구성
- 설계 핵심: 확장성, 이중성, 가용성
- 관리: 사내 관리자 및 업체 관리자

2) WAN(Wide Area Network)

- a) LAN과 LAN 을 연결하는 외부 네트워크 (멀리 떨어져있는 공간 연결, 서울 - 대전)
- b) 장비: 라우터
- c) 프로토콜: [HDLC, PPP, Frame-Relay 잘 안씀], Ethernet**
- d) 연결 방법: 기업 입장에서는 ISP 업체로부터 회선(네트워크망/인터넷망)을 임대한다.
- e) 관리: ISP 업체 관리자 및 SI/NI 업체 관리자
- f) WAN 구간에서도 Ethernet 프로토콜을 주고 사용하고 있으며 이유는 다음과 같이 Ethernet 프로토콜을 지원하는 장치들의 대역폭이 크기 때문이다.

LAN, WAN은 처음에 군사 목적으로 나왔다.

장치명	대역폭
Ethernet 인터페이스	10M
FastEthernet 인터페이스	100M
GigabitEthernet 인터페이스	1000M
10GigabitEthernet 인터페이스	10000M

ISP(internet service provider)

ex) SKT, KT, LGU+

- 기업 고객 및 사용자에게 네트워크 망을 임대해주는 기업

SI/NI(시스템 네트워크 구조를 만들어주는 업체)

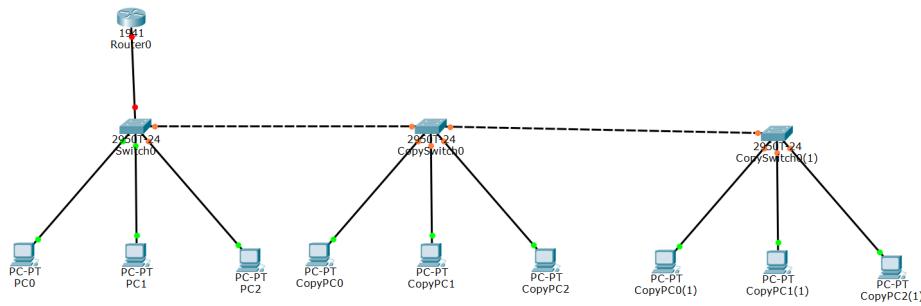
- 기업 고객 및 사용자에게 시스템 및 네트워크 환경을 구축해주는 기업

밴더

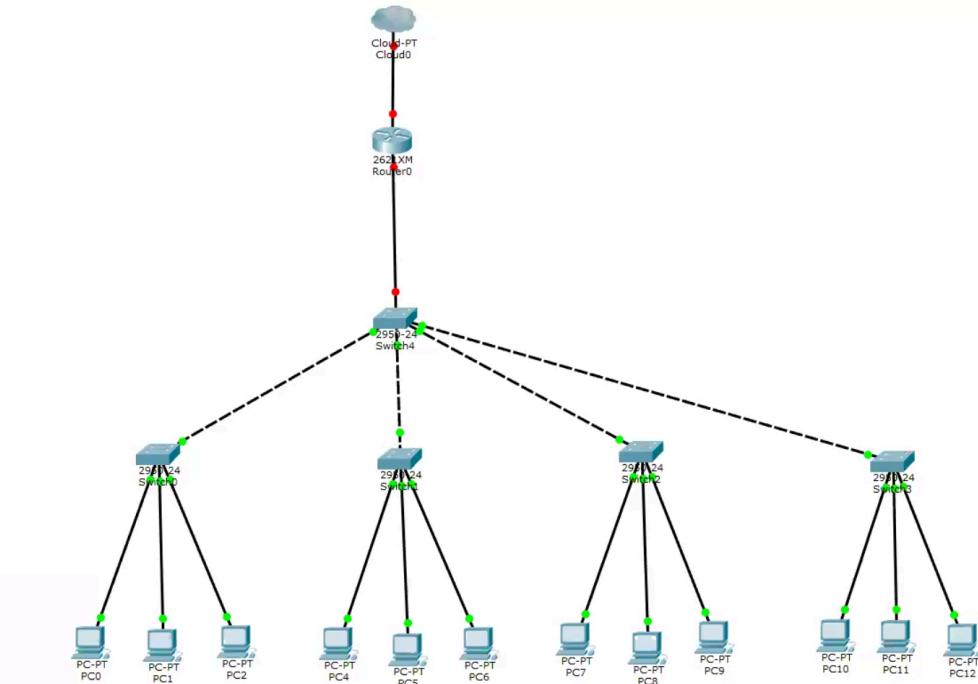
- 자사 기술력으로 제품을 연구, 개발, 생산, 판매, 기술 지원을 제공하는 기업

KT 2026년 IPv6 망 구축 사업(3조)

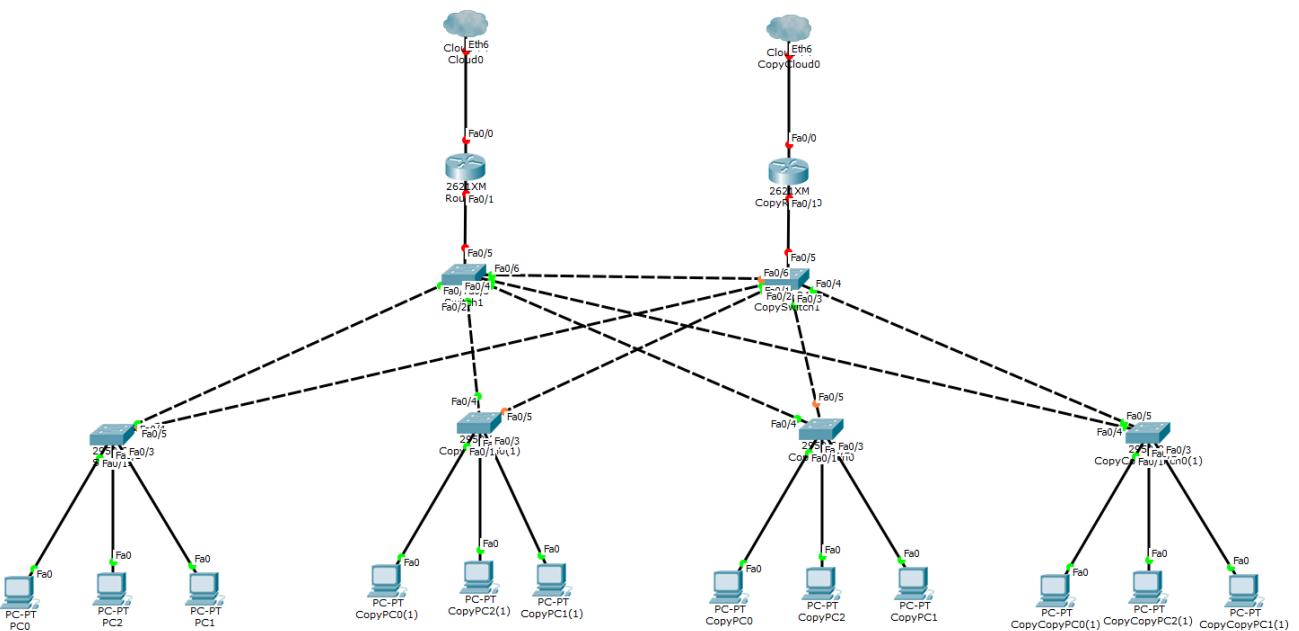
버스 토플로지(군대 훈련식. 한군데가 나가면 다나감):



스타 토플로지(군대식. 스위치를 연결해 안정성 강화):



스타 토플로지 + 이중화 구성 예시:



3) Internet(International network)

- 전 세계적으로 연결된 네트워크 망
- 프로토콜: TCP/IP, UDP
- 해저 케이블 웹사이트: <https://www.submarinecablemap.com/>

4) Intranet

- 기업 내부에서 사용하는 네트워크 망
- 용도: 회사 게시판, 공지사항, 기록 열람 기타 등등
- 현재는 대부분 웹 서비스로 제공하고 있기 때문에 일반 사용자들도 사용하기 간편하다.
 - 웹에서 접속하는 인트라넷은 진짜 인트라넷이 아니다.
 - 외부의 공격에 취약하다.
- 인트라넷은 보안상의 이유 때문에 외부에서 접속하는 것은 추천하지 않는다.
- 구글 검색: intitle:(“인트라넷”|”Intranet”)

5) 데이터 전송 관계

- 요청에 의한 응답 관계
 - 네트워크를 통해서 데이터를 전송
- **요청자:** 클라이언트(CLient)
- **응답자:** 서버(Server)

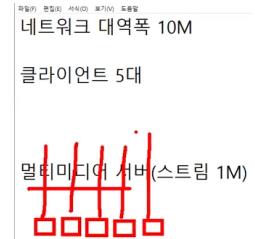
Ex) PC 브라우저에서 'www.naver.com' 접속했을 때, 서버와 클라이언트는 각각 어떻게 되는가?

- PC: Client
- Naver server: Server

6) 데이터 전송 방식

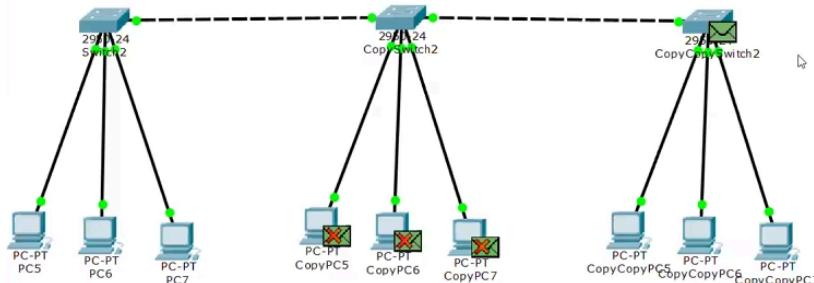
a) 유니캐스트 (Unicast)

- 1:1 데이터 전송
- Ex) 인터넷, 네이버 접속



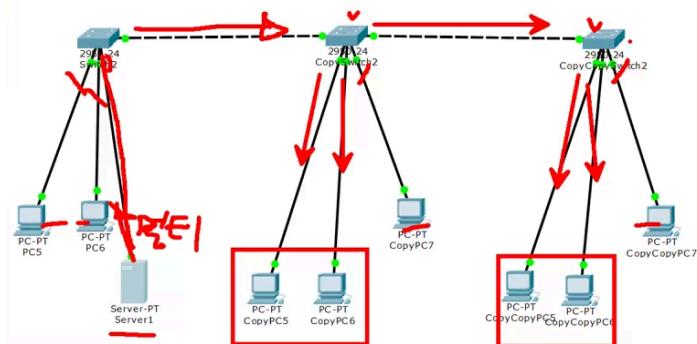
b) 브로드캐스트(Broadcast)

- 1:전체 데이터 전송 (각각의 스위치에서 복제해서 연결된 컴퓨터에 데이터를 보낸다)
- Ex) ARP 요청, DHCP 요청, 내부 네트워크에서만 사용함 (절대 다른 네트워크로 나가지 않음.)
 - 멀티미디어 서버에서 5개를 만들어서 각각 보낸다



c) 멀티캐스트(Multicast)

- 1:특정 그룹 데이터 전송 (각각의 스위치에서 복제해서 정해진 그룹한테 데이터를 보낸다)
- Ex) IPTV



Ex1) 여러 사용자에게 실시간으로 영상 서비스를 하기 위한 효율적인 방식은 무엇인가?

- Broadcast, multicast

Ex2) 과금을 실시한 다수의 사용자에게만 실시간으로 영상 서비스를 하기 위한 방식

- Multicast

Ex3) VOD와 같은 과금을 실시하여 특정 사용자에게만 영상서비스를 하기 위한 효율적인 방식은 무엇인가?

- Unicast

제2장 네트워크 주소 체계

1. 포트 번호

- '2-1.네트워크 주소 체계.pcap' 파일을 와이어샤크로 실행한다.
- TCP, UDP 헤더 안에 포함된 주소
- 주소 크기: 16bit($2^{16} = 0\sim65535$)
- 클라이언트 입장: 서비스 요청 및 실행
- 서버 입장: 서비스 구분 및 제공
- 참고 사이트:
<https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>

User Ports(1024-49151)

- PC에서 사용하는 포트
- mysql 3306

Dynamic/PrivatePort (49152-65535)

- PC에서 사용하는 포트, 동적/사설 포트

System Ports (0-1023)

- 서비스 예약용
- 예전에 개발되어 있던 서비스
- http 80

EX) [IANA Port Number](#) 사이트에서 프로토콜(서비스)를 검색할수 있다. ****기억해야할것****

TCP		UDP	
<u>http</u>	<u>80</u>	<u>domain(dns)</u>	<u>53</u>
<u>https(ssl)</u>	<u>443</u>	bootps(dhcp server)	67
telnet	23	bootpc(dhcp client)	68
<u>ssh</u>	<u>22</u>	syslog	514
ftp	21	ntp	123
ftp-data	20	snmp	161
smtp	25	tftp	69
pop3	110		
mysql	3306		

ex) 클라이언트가 서버에게 요청을 했을 때 다음은 무엇을 요청한 것인가?



- SA = source address
- DA = destination address
- DA 53은 domain을 요구한다
- 포트 번호를 보고 클라이언트가 뭐를 요구하는지 파악할 수 있다.

***SA포트번호는 테이블 번호, DA포트번호는 메뉴판이라고 생각하면 편하다.

Window에서 포트 번호 확인법:

cmd창에서 'netstat' 명령어를 통해 네트워크 연결 상태를 확인할 수 있다:

활성 연결			
프로토콜	로컬 주소	외부 주소	상태
TCP	192.168.11.181:49733	153:https	ESTABLISHED
TCP	192.168.11.181:49755	134:https	TIME_WAIT
TCP	192.168.11.181:49802	tp-in-f188:5228	ESTABLISHED
TCP	192.168.11.181:49827	208.103.161.1:https	ESTABLISHED
TCP	192.168.11.181:49832	27:https	ESTABLISHED
TCP	192.168.11.181:49879	3:https	ESTABLISHED
TCP	192.168.11.181:49893	110.93.158.140:https	ESTABLISHED
TCP	192.168.11.181:49897	110.93.158.140:https	ESTABLISHED
TCP	192.168.11.181:49935	218:https	TIME_WAIT
TCP	192.168.11.181:49943	218:https	TIME_WAIT
TCP	192.168.11.181:49946	211:https	TIME_WAIT

- 여기서 상태가 ESTABLISHED 면 서버와 연결된 상태이다. 나머지는 연결이 안된 거다.

Ex1) 'netstat' 정보 확인을 이용한 네트워크 연결 상태 확인

프로토콜	로컬 주소	외부 주소	상태
TCP	192.168.10.27:49469	211.241.228.21:80	ESTABLISHED
TCP	192.168.10.27:49473	211.241.228.25:443	ESTABLISHED
TCP	192.168.10.27:49476	121.160.34.231:21	FIN_WAIT_2
TCP	192.168.10.27:49479	61.42.100.13:22	CLOSE_WAIT
TCP	192.168.10.27:49481	61.42.100.13:23	ESTABLISHED

- 클라이언트의 IP 주소: 192.168.10.27
- 클라이언트가 제공받고 있는 서비스 또는 연결된 서비스는 무엇인가?
 - 클라이언트와 연결된 외부주소의 포트 번호를 볼 때
 - 80 : http
 - 443 : https
 - 23 : telnet

Ex2) '192.168.10.27'은 서버인가? 클라이언트인가?

프로토콜	로컬 주소	외부 주소	상태
TCP	192.168.10.27:80	211.241.228.21:43511	ESTABLISHED
TCP	192.168.10.27:443	211.241.228.25:53122	ESTABLISHED

- system ports를 사용하고, 외부주소의 포트가 User Ports이므로 서버이다.

2. IP 주소

- IP 헤더 안에 포함된 주소
- 주소 크기: $32\text{bit}(2^{32}\text{개}) = 4,294,967,296$
 - 2012년 2월에 ipv4주소 고갈
 - 우리나라에는 1억 몇천개 정도 가지고 있다. '
 - IPv6 128bit = 2^{128} (간 단위)
- **로컬 환경에서 리모트 환경으로 데이터 전송 담당**
- 변경이 가능한 논리적인 주소(만들어서/설정해서 사용하는 주소)
- LAN이 아닌 다른 네트워크(WAN)으로 보낼때는 IP가 필수이다.
- IP 주소검색: <https://xn--c79as89aj0e29b77z.xn--3e0b707e/>
 - 나라, 할당한 회사 등을 알 수 있다.

3. MAC 주소

- ETH 헤더 안에 포함된 주소
- 주소 크기: $48\text{bit}(2^{48}\text{개})$
- ETH 로컬 환경 내에서 데이터 전송 담당
- **00-e0-4c-14-62-ba**
[-----]
OUI 24bit: 랜카드 업체(이더넷 장치 업체)가 IEEE 기관으로부터 임대받은 주소
 - ex)
 - cisco (00-00-0C)
- 변경 불가능한 물리적인 주소
- MAC 주소 조회 사이트: <https://aruljohn.com/>
 - 나라, 제조업체 등을 알 수 있다.

Ex) 시스템 주소 관련 예제

C:\>Administrator>**ipconfig /all**

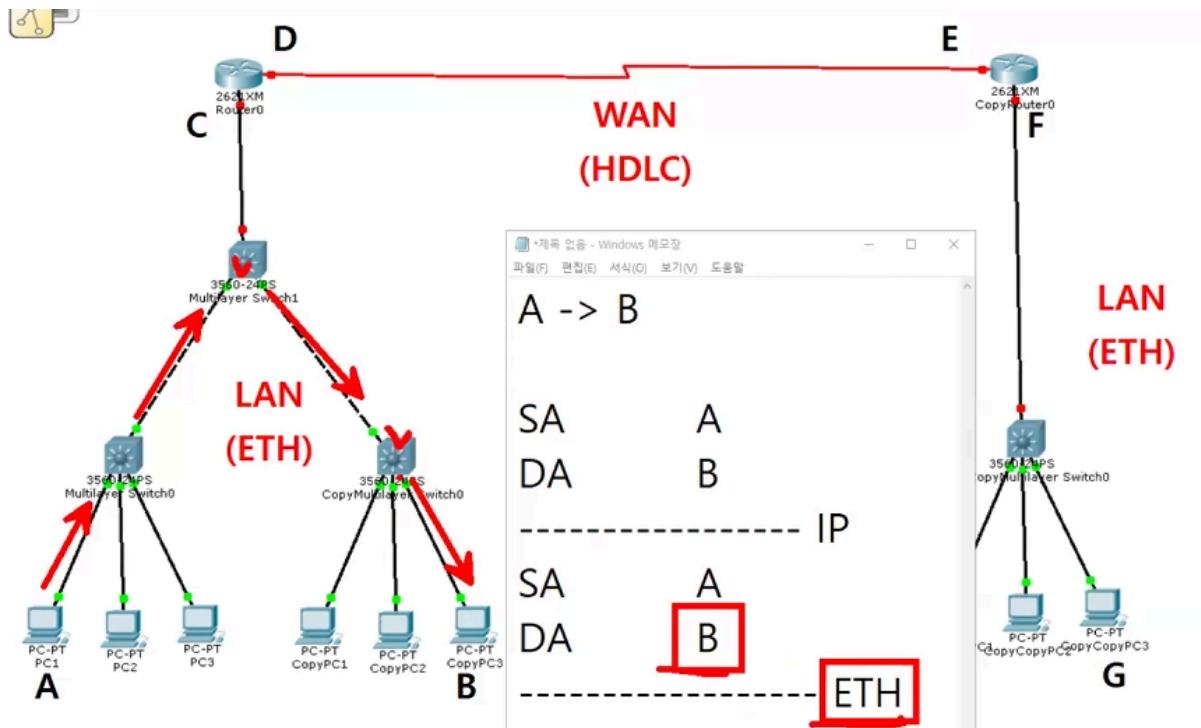
이더넷 어댑터 이더넷:

```
연결별 DNS 접미사....:  
설명.....: Realtek PCIe GbE Family Controller  
물리적 주소 .....: 00-D8-61-6E-30-B4  
DHCP 사용 .....: 예  
자동 구성 사용.....: 예  
링크-로컬 IPv6 주소 ....: fe80::b069:5b58:6879:a6ac%14(기본 설정)  
IPv4 주소 .....: 192.168.10.27(기본 설정)  
서브넷 마스크 .....: 255.255.255.0  
임대 시작 날짜.....: 2022년 4월 13일 수요일 오전 9:18:08  
임대 만료 날짜.....: 2022년 4월 13일 수요일 오전 11:18:06  
기본 게이트웨이 .....: 192.168.10.1  
DHCP 서버 .....: 192.168.10.1  
DHCPv6 IAID .....: 100718689  
DHCPv6 클라이언트 DUID...: 00-01-00-01-27-BE-97-9A-00-D8-61-6E-30-B4  
DNS 서버.....: 168.126.63.1  
168.126.63.2  
Tcpip 를 통한 NetBIOS....: 사용
```

- 현재 PC는 수동 IP 설정 방식인가? 아님 DHCP 서비스 방식인가?
 - DHCP 사용
- PC IP 주소와 서브넷 마스크는 어떻게 되는가?
 - 192.168.10.27
 - 255.255.255.0
- 게이트웨이 주소?
 - 192.168.10.1
- DNS 서버 주소?
 - 168.126.63.1
- MAC 주소?
 - 00-D8-61-6E-30-B4
- MAC 주소 검색 후 어느 회사 제품파악:
 - Micro-Star
- PC(시스템/호스트)를 구분할때 IP 주소, MAC 주소 중에 어떤 주소가 구분이 정확한가?
 - MAC주소

데이터 전송 체계 (그림 예제)

ex) 같은 네트워크 일 경우우



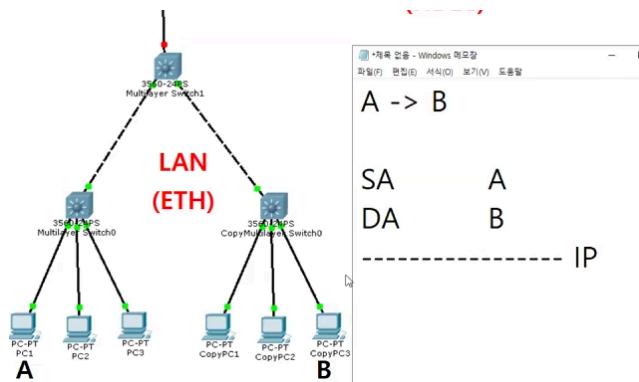
A \Rightarrow B (A에서 B로 데이터를 전송. 같은 네트워크이다.)

SA	A
DA	B
-----IP	
SA	A
DA	B
-----ETH (Mac Address)	

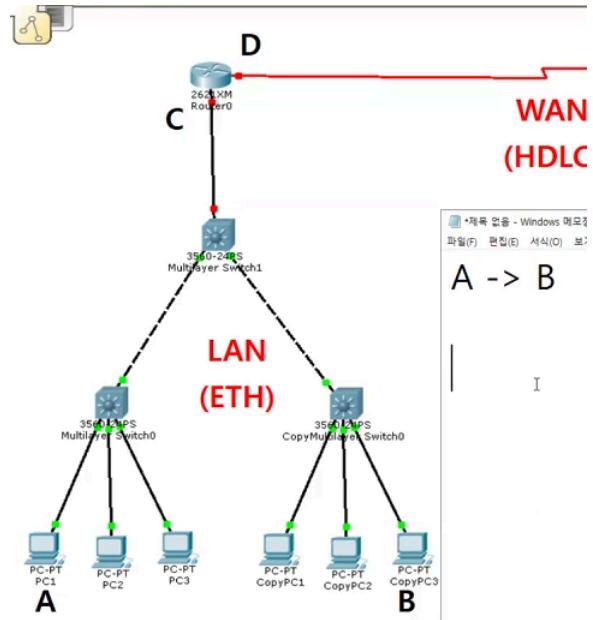
*** 스위치라는 장비는 기본적으로 ETH 헤더만 본다.

- ETH 헤더의 목적지를 확인해서 맞는 방향으로 전기 신호를 보낸다.

A PC에서 출발 \rightarrow 스위치에서 목적지로 이동(ETH 헤더 이용) \rightarrow 목적지 PC 도착
목적지에 도달했을 때 B PC는 ETH 헤더를 확인 한 뒤 ETH를 뜯어낸다 (decapsulation)

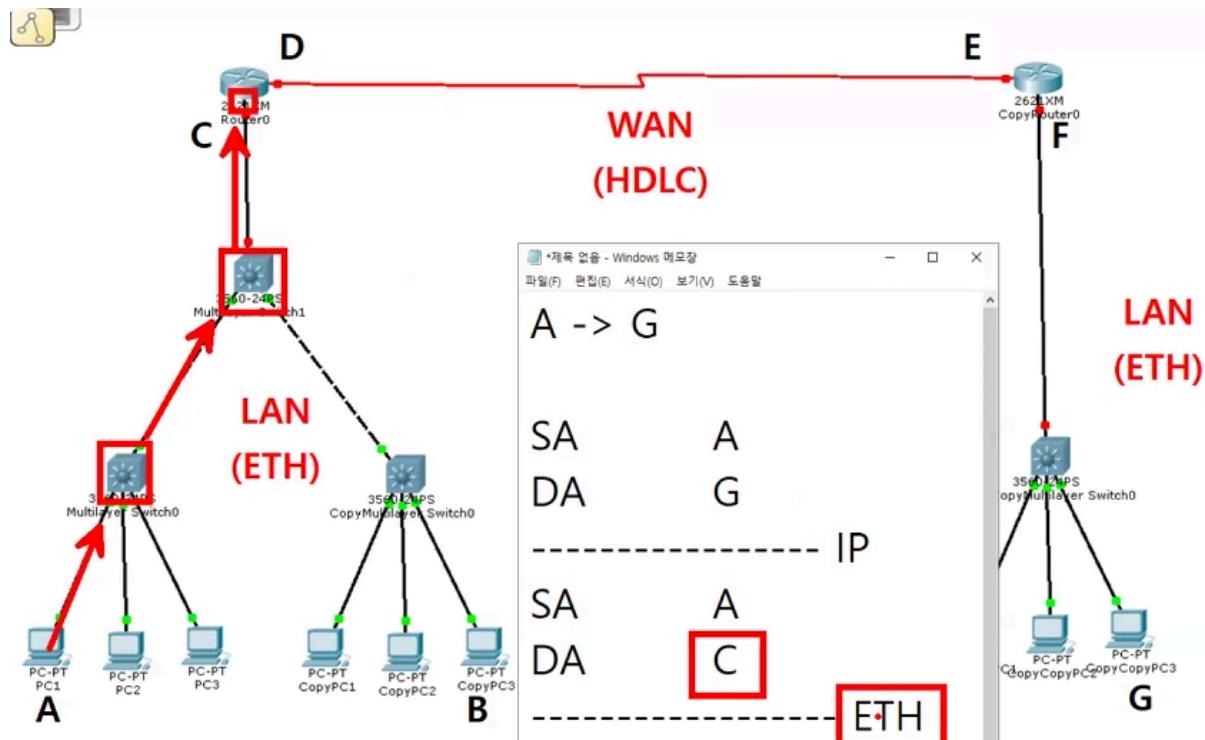


마지막으로 PC가 IP 헤더를 확인 한 뒤, IP를 뜯어낸다 (decapsulation)



ex) 다른 네트워크 일 경우 (A → G)

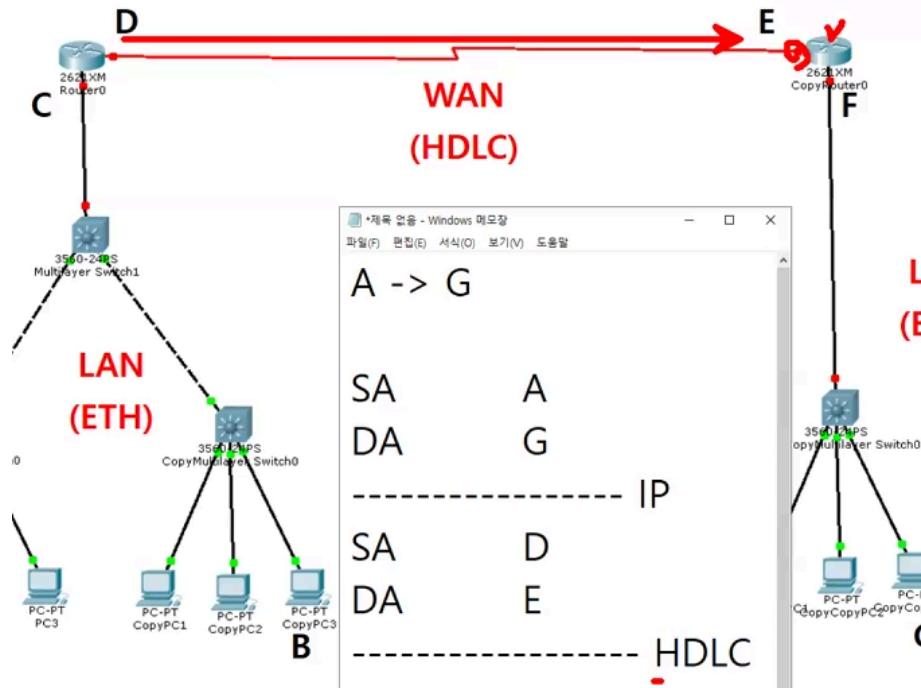
ETH은 같은 네트워크에서 데이터를 전송하려고 만들어진거기 때문에 다른네트워크로 엑세스를 못한다. 그러므로, 현재 네트워크의 게이트웨이 까지만 설정한다.



데이터가 게이트웨이까지 이동하면 ETH의 목적지가 자기한테 온걸 확인 한 후 뜯어낸다 (decapsulation)

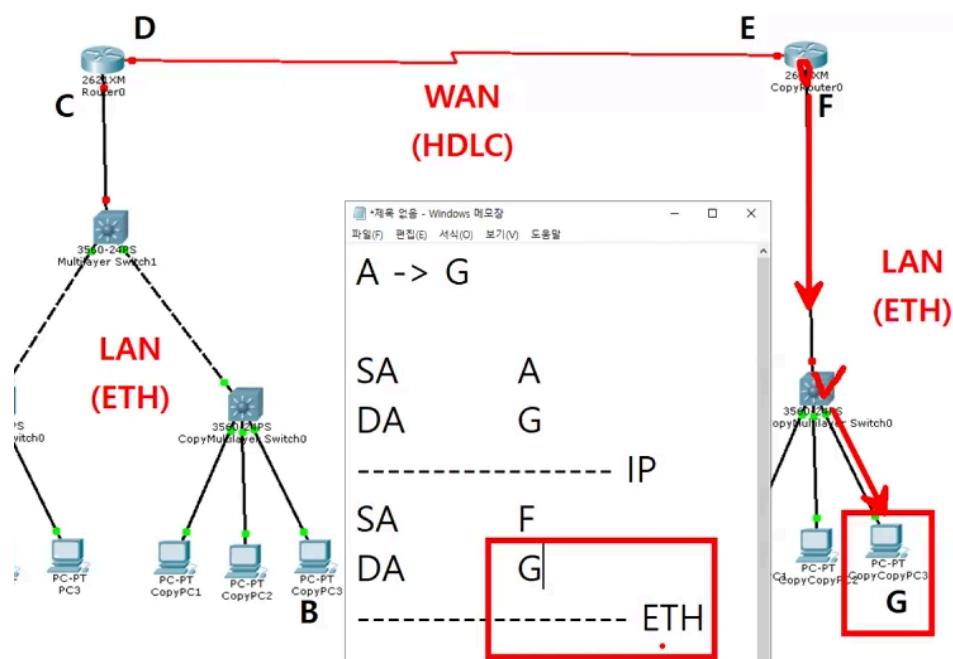
다른 네트워크끼리 데이터 전송을 하려면 WAN을 사용한다. 여기서는 HDLC 프로토콜을 사용하므로 새로 HDLC 프로토콜을 덮어씌워준다.

여기서 출발지는 현재 있는 게이트웨이, 목적지는 보내는 다른 네트워크의 게이트웨이이다. 이 예제에서는 D, E이다.



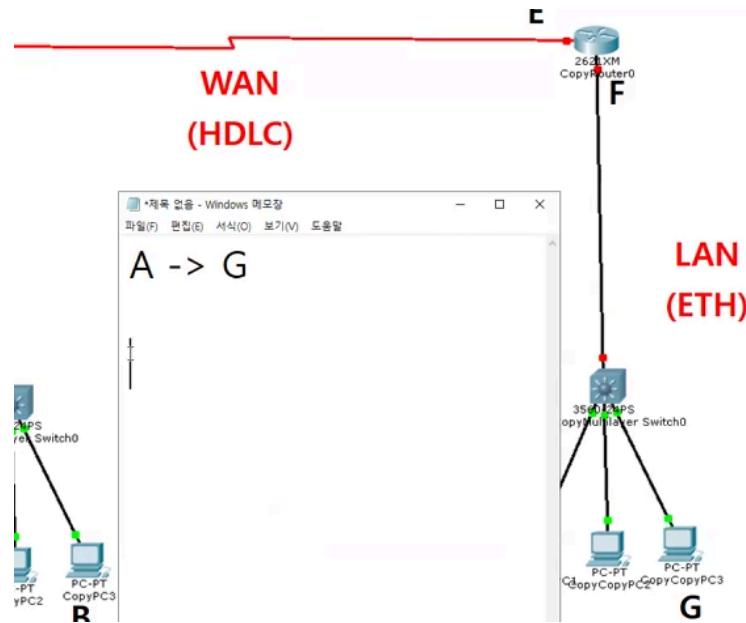
다른 네트워크의 게이트웨이에 도착했을 때, 자기 자신한테 도착한게 맞으므로 HDLC 헤더를 decapsulation 한다. 그러나 IP는 자기한테 온 것이 아니므로 목적지를 다시 정해줘야 한다.

이제 목적지의 PC가 같은 네트워크에 있기 때문에 다시 ETH 헤더를 encapsulation 한다. 출발지는 F(현 네트워크의 게이트웨이), 목적지는 G(PC)이다.

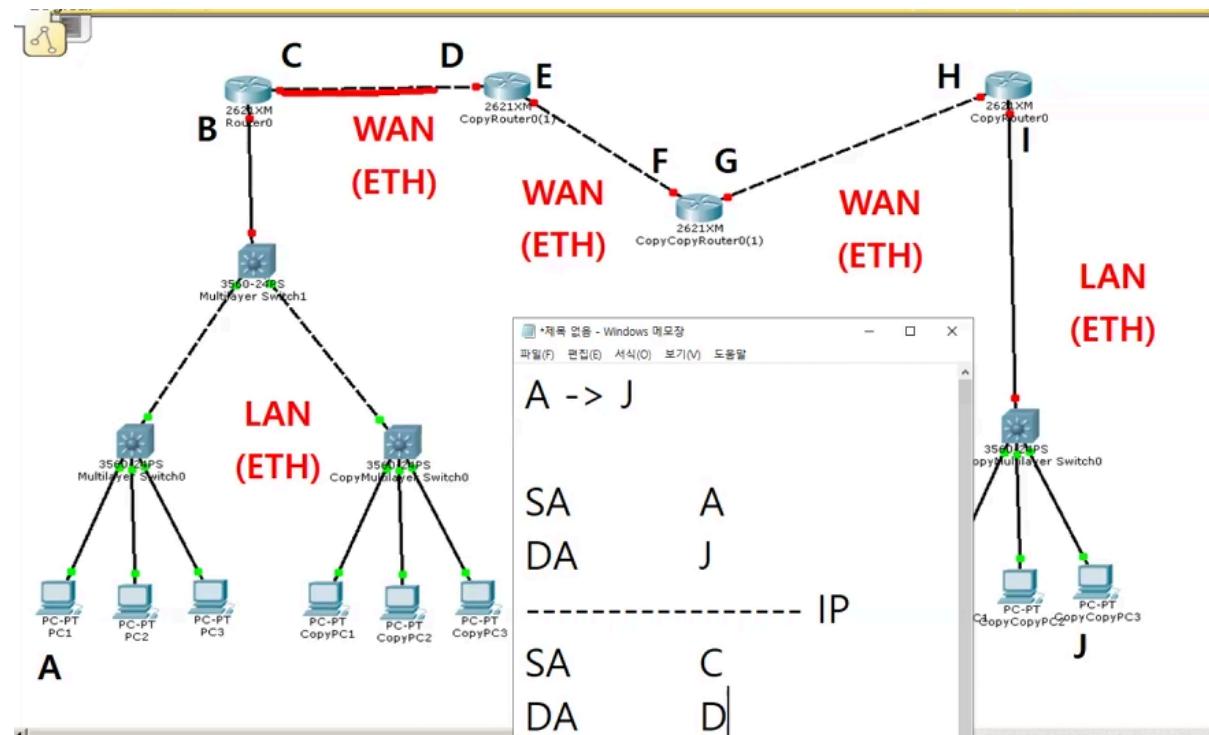


데이터가 스위치를 통해 G에 도달했을 때 PC는 ETH 헤더를 확인한 후 자기한테 온 것이 맞으면 Decapsulation 한다.

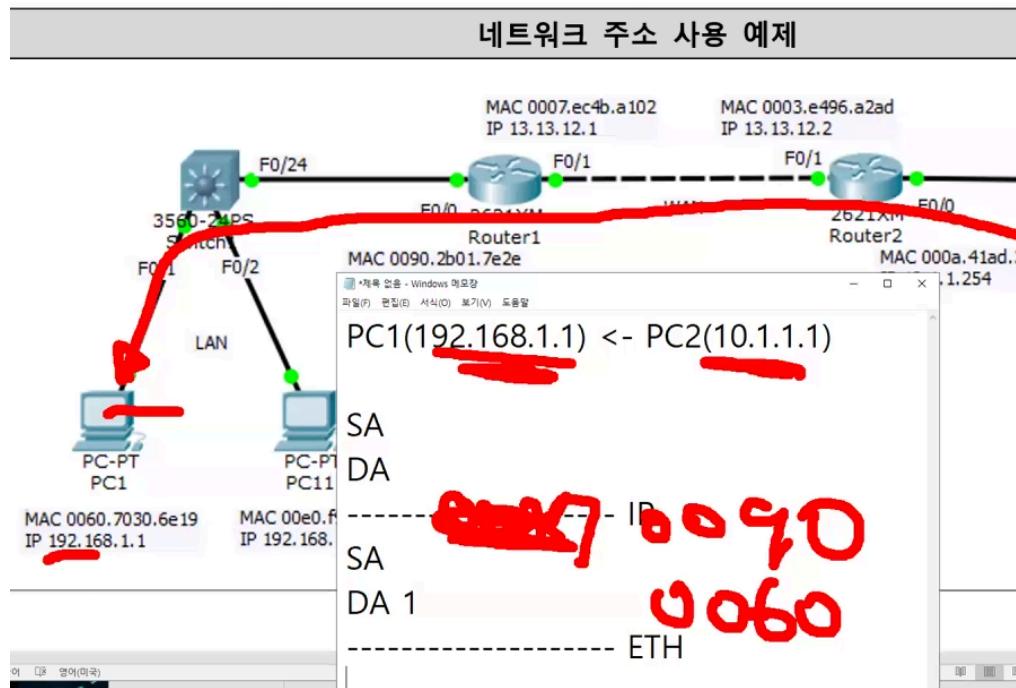
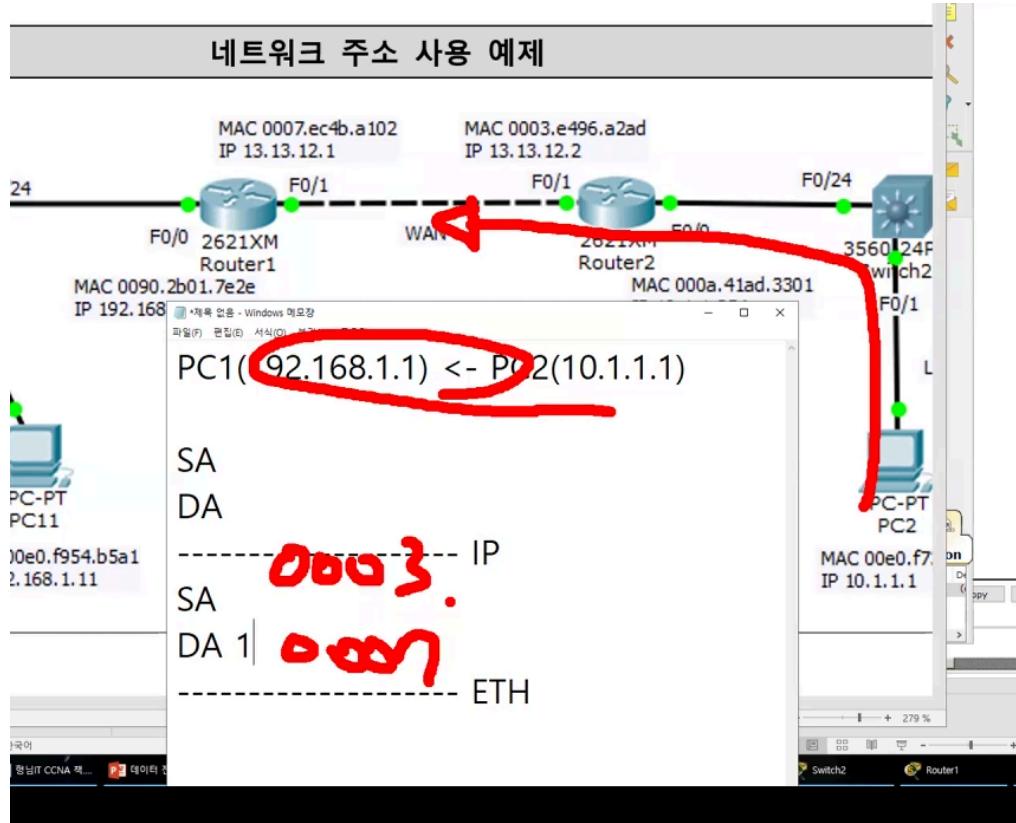
마지막으로 IP 헤더를 확인 한 후 맞으면 Decapsulation 한다.



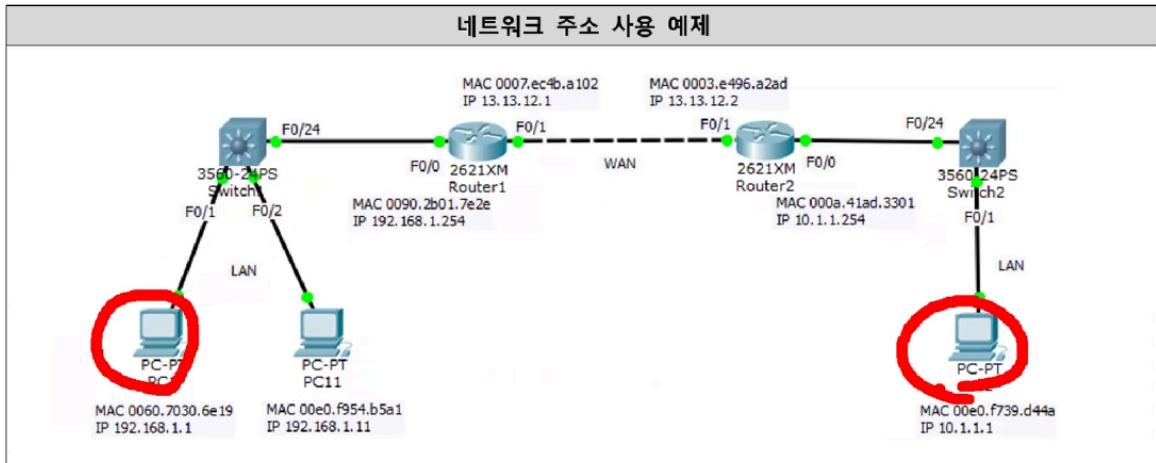
ex) 이런 경우에는 WAN끼리 이동할 때마다 decap, encap을 해 줘야한다.
WAN, LAN 둘다 같은 ETH를 사용하지만 다른 정보를 가지고 있다.



ex) 시험 문제중에 현 주소를 이야기해라:



'2-2.네트워크 주소 사용 예제.pkt' 파일을 실행한다.



PC1 → PC2로 데이터를 전송할 때 출발지, 목적지 MAC은?

- SA 0060
- DA 0090

스위치 장비는 ETH 헤드까지만 본다

ETH 프로토콜은 같은 네트워크 안에서만 데이터를 전송할 수 있다.

다른 네트워크로 데이터를 전송할 때는 ETH(LAN), IP, HDLC(WAN) 프로토콜을 함께 사용한다.

IP는 외부 통신 네트워크로 보내기 위해서 만들어졌다.

다른 네트워크로 나가기 위해서는 라우터를 거쳐야 하는데, 다른 네트워크로 연결하는 통로를 게이트웨이(Gateway)라고 한다.

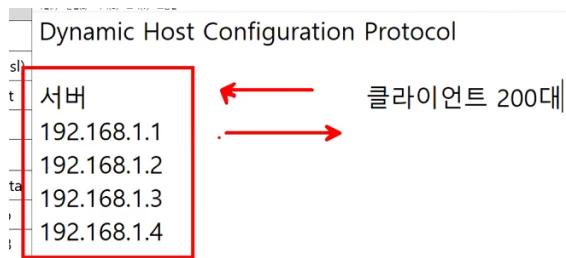
제3장 TCP & UDP 서비스

4-1.데이터 전송 프로토콜.pcap

4-2.데이터 전송 프로토콜(pkt)

DHCP(Dynamic Host Configuration Protocol)

- IP 자동 할당 프로토콜
- IP를 서버에서 자동으로 할당해주게 만들어주는 프로토콜



http

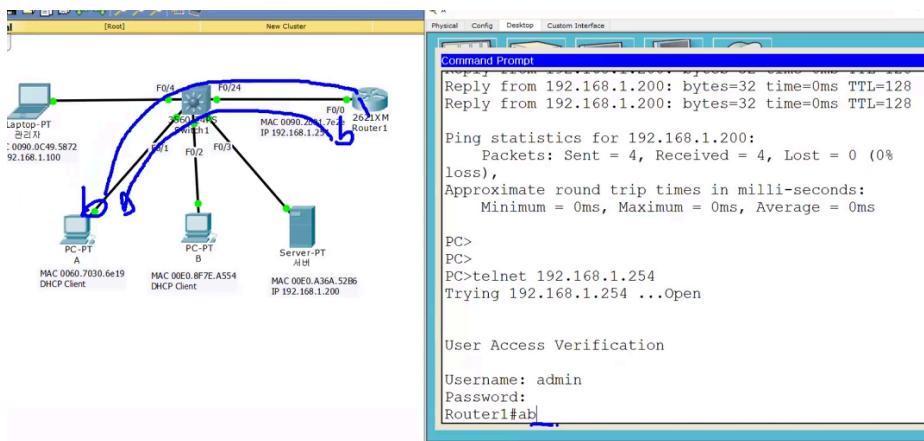
- 대량의 문자를 보내주는, 들고오는 프로토콜, 보안에 취약하다 (보든 데이터를 평문으로 보낸다)
- 미리 준비되어있는 데이터를 가져온다.

https(ssl)

- http와 동일하나 데이터 요청, 응답을 암호화시켜서 전송이 된다.
- 미리 준비되어있는 데이터를 가져온다.

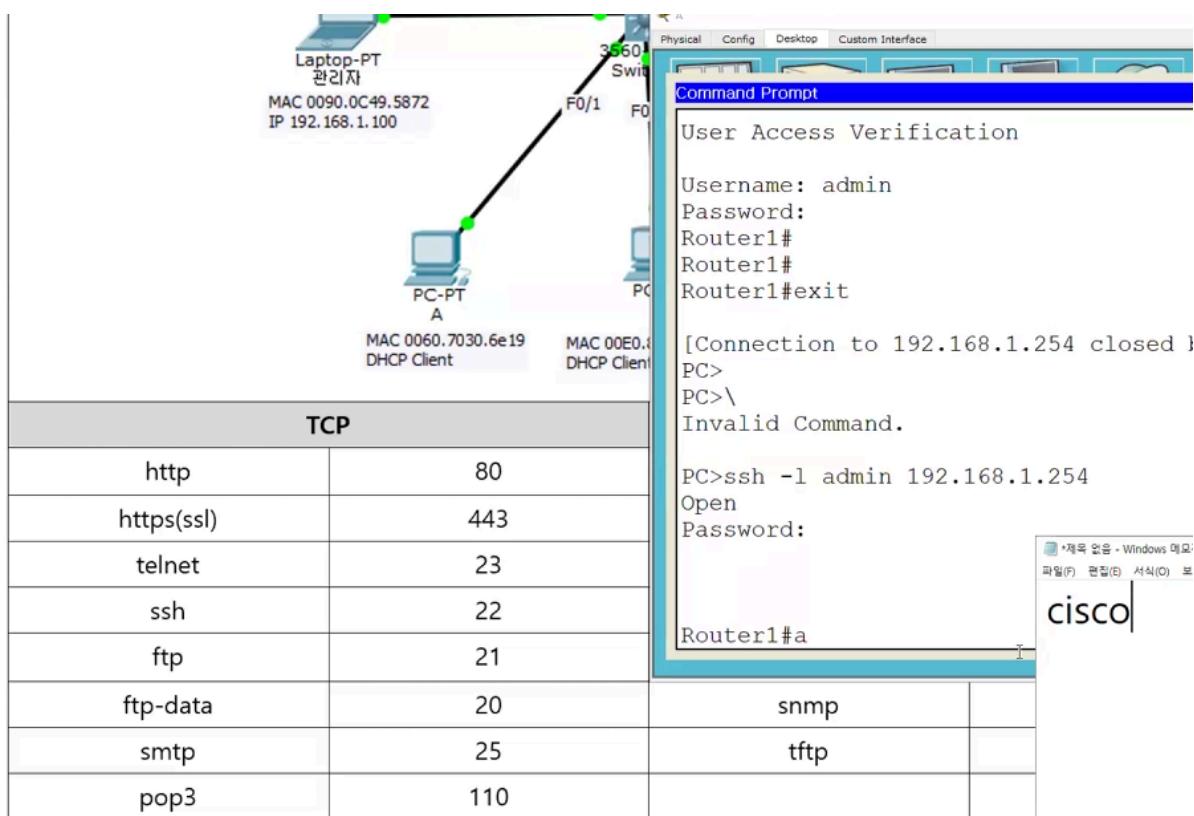
telnet (port: 23)

- 원격으로 네트워크를 통해서 접속할수 있는 서비스 (PC에서 라우터)
- 미리 준비되어있는게 아니라 실시간으로 입력가능
- Password는 라우터에서 응답을 안해줘서 안보인다.
- 문제점:
 - 데이터를 요청, 응답할때 평문으로 보낸다 (보안취약)
- 사용권장 안함 (보안취약)
- telnet [접속 IP]
 - id: admin, pw: cisco
- exit 하면 나가기



ssh (port: 22)

- telnet과 같은 서비스이다
- 실시간 문자들이 전송될때 암호화를 시켜서 전송한다.
- 사용 권장
- ssh -l admin [접속 IP]
 - id: admin, pw: cisco



ftp(file transfer protocol) 21

- 파일을 연결할때 쓴다 (PC >> Server)

```
PC>ftp 192.168.1.200
Trying to connect...192.168.1.200
Connected to 192.168.1.200
220- Welcome to PT Ftp server
Username:admin
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>
```

ftp-data(file transfer protocol) 22

- 파일을 전송할때 쓴다

▷온로드:

```
ftp>
ftp>dir

Listing /ftp directory from 192.168.1.200:
0   : asa842-k8.bin                               5571584
1   : c1841-advipsericesk9-mz.124-15.T1.bin      33591768
2   : c1841-ipbase-mz.123-14.T7.bin              13832032
3   : c1841-ipbasek9-mz.124-12.bin              16599160
4   : c2600-advipsericesk9-mz.124-15.T1.bin      33591768
5   : c2600-i-mz.122-28.bin                      5571584
6   : c2600-ipbasek9-mz.124-8.bin                13169700
7   : c2800nm-advipsericesk9-mz.124-15.T1.bin    50938004
8   : c2800nm-advipsericesk9-mz.151-4.M4.bin     33591768
9   : c2800nm-ipbase-mz.123-14.T7.bin          5571584
10  : c2800nm-ipbasek9-mz.124-8.bin            15522644
11  : c2950-i6q412-mz.121-22.EA4.bin           3058048
12  : c2950-i6q412-mz.121-22.EA8.bin           3117390
13  : c2960-lanbase-mz.122-25.FX.bin          4414921
14  : c2960-lanbase-mz.122-25.SE1.bin          4670455
15  : c2960-lanbasek9-mz.150-2.SE4.bin         4670455
16  : c3560-advipsericesk9-mz.122-37.SE1.bin    8662192
17  : pt1000-i-mz.122-28.bin                  5571584
18  : pt3000-i6q412-mz.121-22.EA4.bin          3117390
ftp>
ftp>get pt3000-i6q412-mz.121-22.EA4.bin

Reading file pt3000-i6q412-mz.121-22.EA4.bin from 192.168.1.200:
File transfer in progress...
```

파일전송 완료:

```
[Transfer complete - 3117390 bytes]

3117390 bytes copied in 19.463 secs (160170 bytes/sec)
ftp>
```

나갈때는 quit을 쓴다.

업로드:

1) 로그인

```
PC>ftp 192.168.1.200
Trying to connect...192.168.1.200
Connected to 192.168.1.200
220- Welcome to PT Ftp server
Username:admin
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>
```

2) put 매서드 사용

```
ftp>put sampleFile.txt

Writing file sampleFile.txt to 192.168.1.200:
File transfer in progress...

[Transfer complete - 26 bytes]

26 bytes copied in 0.048 secs (541 bytes/sec)
ftp>
```

3) 업로드 확인

```
26 bytes copied in 0.048 secs (541 bytes/sec)
ftp>dir

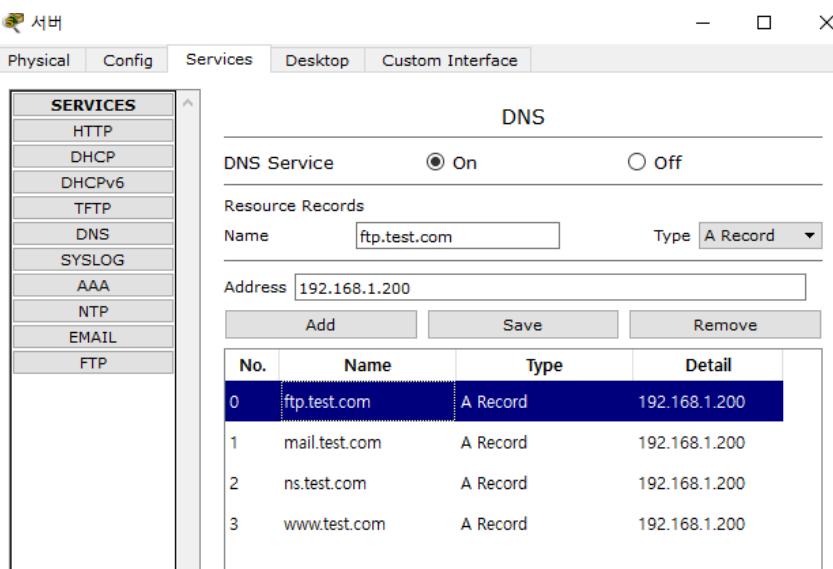
Listing /ftp directory from 192.168.1.200:
0 : asa842-k8.bin                                5571584
1 : c1841-advipservicesk9-mz.124-15.T1.bin      33591768
2 : c1841-ipbase-mz.123-14.T7.bin                13832032
3 : c1841-ipbasek9-mz.124-12.bin                16599160
4 : c2600-advipservicesk9-mz.124-15.T1.bin      33591768
5 : c2600-i-mz.122-28.bin                         5571584
6 : c2600-ipbasek9-mz.124-8.bin                  13169700
7 : c2800nm-advipservicesk9-mz.124-15.T1.bin    50938004
8 : c2800nm-advipservicesk9-mz.151-4.M4.bin     33591768
9 : c2800nm-ipbase-mz.123-14.T7.bin              5571584
10 : c2800nm-ipbasek9-mz.124-8.bin               15522644
11 : c2950-i6q4l2-mz.121-22.EA4.bin             3058048
12 : c2950-i6q4l2-mz.121-22.EA8.bin             3117390
13 : c2960-lanbase-mz.122-25.FX.bin            4414921
14 : c2960-lanbase-mz.122-25.SEE1.bin           4670455
15 : c2960-lanbasek9-mz.150-2.SE4.bin           4670455
16 : c3560-advipservicesk9-mz.122-37.SE1.bin   8662192
17 : pt1000-i-mz.122-28.bin                     5571584
18 : pt3000-i6q4l2-mz.121-22.EA4.bin           3117390
19 : sampleFile.txt                             26
ftp>
```

sampleFile 업로드 완료

DNS 서버 (53)

- domain에 대한 IP를 요청 응답해주는 서비스
- 장점: 유지/보수가 쉬움, IP만 계속 바뀌고 domain은 바뀌지 않음.
- ex) www.naver.com
 - DNS server (storage)
 - www.naver.com - 200.1.1.1 (네이버를 검색했을때 네이버의 IP로 변환해준다)

DNS 서버 설정:



PC A 접속 후 확인

```
PC>nslookup mail.test.com

Server: [192.168.1.200]
Address: 192.168.1.200

Non-authoritative answer:
Name:   mail.test.com
Address: 192.168.1.200
```

web 확인:



SMTP(simple mail transfer protocol) 25

1. 메일 전송 프로토콜
2. Gmail사용자 >> GMAIL서버 >> 네이버메일서버 >> 네이버메일사용자
3. 평문 (보안에 취약함)

Incoming Mail Server

POP3(post office protocol) 110

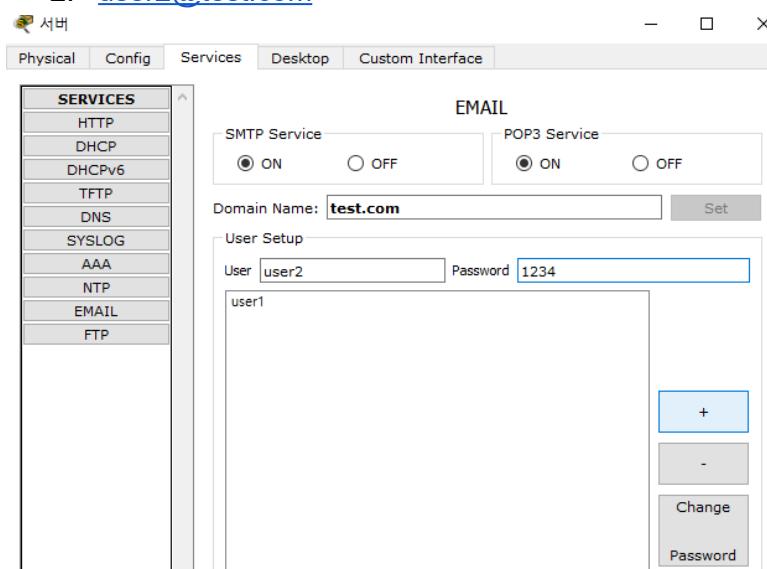
1. 메일을 메일 서버로부터 받을때 쓰는 프로토콜
2. 평문 (보안에 취약함)

Outgoing Mail Server

이메일 서버 세팅 (계정 만들기):

2개의 계정:

1. user1@test.com
2. user2@test.com



메일 세팅:

Configure Mail

User Information

Your Name:

Email Address:

Server Information

Incoming Mail Server

Outgoing Mail Server

Logon Information

User Name:

Password:

제4장 데이터 전송 프로토콜

데이터를 전송할 때 사용하는 필수 프로토콜

1. TCP, UDP, IP, Ethernet
2. ICMP, ARP

1. TCP(Transmission Control Protocol) 20byte + option

- Transmission (전송)
- Control (제어)
- 데이터를 전송하는 프로토콜인데 제어하는 기능이 들어가 있다.
- 20byte
 - 더 크게 나오는 경우가 있다. (옵션추가)
- Layer 4 계층 프로토콜
- 다른 시스템과 통신 수립 연결을 실시한 이후, 데이터 요청 및 응답을 실시하는 연결 지향성 특징을 가지고 있다.

해더 크기는 20byte이며, 옵션이 있을 경우 더 크게 나오는 경우도 있다. Layer 4 계층 프로토콜이며, 다른 시스템과 통신 수립 연결을 실시한 이후, 데이터 요청 및 응답을 실시하는 연결 지향성 특징을 갖고 있다.

29	0.58799	{62.248.74.55	61.42.166.26	TCP	62	1942→445	[SYN]	Seq=0
39	0.82500	:120.50.139.44	172.16.6.13	TCP	60	80→55235	[RST, ACK]	
88	2.19335	:172.16.5.254	121.78.58.15	TCP	70	1320→30060	[PSH, ACK]	
89	2.23565	:121.78.58.15	172.16.5.254	TCP	60	30060→1320	[ACK]	Seq=1
92	2.25173	:121.78.58.15	172.16.5.254	TCP	70	30060→1320	[PSH, ACK]	
94	2.26543	(172.16.5.254	114.111.46.227TCP		62	1980→80	[SYN]	Seq=0
95	2.27104	(114.111.46.227	172.16.5.254	TCP	60	80→1980	[SYN, ACK]	
96	2.27110	:172.16.5.254	114.111.46.227TCP		54	1980→80	[ACK]	Seq=1
97	2.27136	:172.16.5.254	114.111.46.227TCP		1314	[TCP segment of a re		
98	2.27170	:172.16.5.254	114.111.46.227HTTP		983	GET /addAndList.nhn		
99	2.27201	111.111.111.111	111.111.111.111	TCP	60	80.1000	ACK	Seq=1

Frame 94: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)
Ethernet II, Src: Realtek_14:62:ba (00:e0:4c:14:62:ba), Dst: Cisco_ Internet Protocol Version 4, Src: 172.16.5.254 (172.16.5.254), Dst: Transmission Control Protocol, src Port: 1980 (1980), Dst Port: 80 (Source Port: 1980 (1980)
Destination Port: 80 (80)
[Stream index: 3]
[TCP Segment Len: 0]
Sequence number: 0 (relative sequence number)
Acknowledgment number: 0
Header Length: 28 bytes
⊕ 0000 0000 0010 = Flags: 0x002 (SYN)
Window size value: 65535
[Calculated window size: 65535]
⊕ Checksum: 0xcc2a [validation disabled]
Urgent pointer: 0
⊕ Options: (8 bytes), Maximum segment size, No-Operation (NOP), No-Op

1) 3-Way 핸드 쉐이킹 동작 (3번 데이터가 옮겨진다)

1) '3-Way 핸드 쉐이킹' 동작 실시

클라이언트와 서버는 다음과 같이 '3-Way 핸드 쉐이킹' 동작을 실시하여 TCP 연결을 성립하고 데이터 요청 및 응답을 실시한다.

94	2.265	172.16.5.254	114.111.46.227	TCP	62 1980->80 [SYN] Seq=0 Win=65535
95	2.271	114.111.46.227	172.16.5.254	TCP	60 80->1980 [SYN, ACK] Seq=0 Ack=1
96	2.271	172.16.5.254	114.111.46.227	TCP	54 1980->80 [ACK] Seq=1 Ack=1 Win=1
97	2.271	172.16.5.254	114.111.46.227	TCP	1314 [TCP segment of a reassembled
98	2.271	172.16.5.254	114.111.46.227	HTTP	983 GET /addAndList.nhn?r=linkedMe
99	2.277	114.111.46.227	172.16.5.254	TCP	60 80->1980 [ACK] Seq=1 Ack=1261 W
100	2.277	114.111.46.227	172.16.5.254	TCP	60 80->1980 [ACK] Seq=1 Ack=2190 W
101	2.279	114.111.46.227	172.16.5.254	HTTP	902 HTTP/1.1 200 OK (text/plain)

클라이언트	서버
172.16.5.254:1980	114.111.46.227:80


```

94      ① Syn(Seq=0) ->
95                      <- Syn, Ack(Seq=0, Ack=1) ②
96      ③ Ack(Seq=1, Ack=1) ->
97          ----- 통신 수립 완료(ESTABLISHED) -----
98          데이터/서비스 요청 ->
99      ④ HTTP GET
100
101             <- 데이터/서비스 응답
102                 HTTP/1.1 200 OK      ⑤

```

[클라이언트]

[서비]

TCP를 통해서 연결을 하려면 TCP 성립을 해야한다.

1. Sync 라는 데이터를 보낸다 (서버한테 통신한다고 알려줌) ->
 2. 서버가 Sync를 받았다면 서버는 Sync + Ack 를 보낸다 <-
 3. Sync + Ack를 Client 가 받은 후 서버가 잘 연결된걸 확인한다
 4. Client는 Ack를 다시 서버로 보낸다. ->

이 과정을 거치면 TCP 연결 성립이 된다.

데이터 요청 ->

◀-데이터 응답

ex) 3-way hand shaking

92	2.251.5.121.7.8.58.15	1/2.16.5.254	TCP	/0 50000→1520 [PSH, ACK] Seq=1 ACK=1
94	2.26543(172.16.5.254	114.111.46.227	TCP	62 1980→80 [SYN] Seq=0 Win=65535 Len=0
95	2.27104(114.111.46.227	172.16.5.254	TCP	60 80→1980 [SYN, ACK] Seq=0 Ack=1 Win=
96	2.27110(172.16.5.254	114.111.46.227	TCP	54 1980→80 [ACK] Seq=1 Ack=1 win=65535
97	2.27136(172.16.5.254	114.111.46.227	TCP	1314 [TCP segment of a reassembled PDU]

```

Frame 94: 62 bytes on wire (490 bits), 62 bytes captured (490 bits)
Ethernet II, Src: Realtek_14:62:ba (00:e0:4c:14:62:ba), Dst: Cisco_31:81:b1 (00:13
Internet Protocol Version 4, Src: 172.16.5.254 (172.16.5.254), Dst: 114.111.46.227
Transmission Control Protocol, Src Port: 1980 (1980), Dst Port: 80 (80), Seq: 0, Len
Source Port: 1980 (1980)
Destination Port: 80 (80)
[Stream index: 3]
[TCP Segment Len: 0]
Sequence number: 0 (relative sequence number)
Acknowledgment number: 0
Header Length: 28 bytes
.... 0000 0000 0010 = Flags: 0x002 (SYN)
000. .... .... = Reserved: Not set
...0 .... .... = Nonce: Not set
.... 0.... .... = Congestion Window Reduced (CWR): Not set
.... .0. .... = ECN-Echo: Not set
.... ..0. .... = Urgent: Not set
.... .0 .... = Acknowledgment: Not set
.... 0.... = Push: Not set
.... .... 0.. = Reset: Not set
+ .... .... .1. = Syn: Set
.... .... .0 = Fin: Not set
Window size value: 65535
[Calculated window size: 65535]

```

Sync 설정됨 (연결됨)

```

.... 0000 0001 0010 = Flags: 0x012 (SYN, ACK)
000. .... .... = Reserved: Not set
...0 .... .... = Nonce: Not set
.... 0.... .... = Congestion Window Reduced (CWR): Not set
.... .0. .... = ECN-Echo: Not set
.... ..0. .... = Urgent: Not set
.... .1 .... = Acknowledgment: Set
.... .... 0... = Push: Not set
.... .... .0.. = Reset: Not set
+ .... .... .1. = Syn: Set
.... .... .0 = Fin: Not set

```

Sync, Ack (잘 받았다는 의미로 보낸다) 설정

```

.... 0000 0001 0000 = Flags: 0x010 (ACK)
000. .... .... = Reserved: Not set
...0 .... .... = Nonce: Not set
.... 0.... .... = Congestion Window Reduced (CWR): Not set
.... .0. .... = ECN-Echo: Not set
.... ..0. .... = Urgent: Not set
.... .1 .... = Acknowledgment: Set
.... .... 0... = Push: Not set
.... .... .0.. = Reset: Not set
.... .... ..0. = Syn: Not set
.... .... ...0 = Fin: Not set

```

다시 Ack만 설정

2) TCP Control Flag(6bit)

다음은 TCP 플래그 상태에 따른 TCP 주요 동작이다.

.... 0000 0000 0010 = Flags: 0x002 (SYN)	urg (1....) : 긴급한 데이터 표기
000. = Reserved: Not set	ack (1...) : 확인 응답/승인
...0 = Nonce: Not set	psh (...1...) : 상위 프로세스 처리
.... 0.... = Congestion Window Reduced (CWR): Not set	rst (...1.) : 강제 종료
.... .0.... = ECN-Echo: Not set	syn (...1.) : 통신 개시
.... .0.... = Urgent: Not set	fin (...1) : 정상적인 종료
.... .0.... = Acknowledgment: Not set	
.... .0.... = Push: Not set	
.... .0.... = Reset: Not set	
.... .0.... 1. = Syn: Set	
.... .0.... 0 = Fin: Not set	

Fin, Reset은 종료이다.

Fin: 상대방의 상태를 확인 후 종료 (정상종료)

Reset: 상대방의 상태를 확인 안하고 강제종료

- TCP 연결 된 상태에서, 연결이 없는 경우에도 가능
- ex)

클라이언트

SA 50001

DA 22

Syn ----->

웹-서버(TCP 80포트 오픈)

SA 22

DA 50000001

<----- RST(Reset) + Ack

리셋을 받은 경우 포트번호 22번이 닫혀 있기 때문이다.

공격자 입장에서는 22번이 닫혀있다는 걸 알 수 있다.

클라이언트

SA 50001

DA 80

Syn ----->

웹-서버(TCP 80포트 오픈)

SA 80

DA 50000001

<----- Syn + Ack

Syn를 받은 경우 80 번 포트가 열여있다는 걸 알 수 있다.

Push

- 데이터를 모아놨다가 출력한다
- 모아놨다가 출력하는 걸 버퍼링이라고 한다.
- 1로 설정되어있으면 버퍼링하지 말고 바로 출력시켜라 라는 뜻이다.
- 바로바로 처리해라 라는 뜻으로 해석

```
000. .... .... = Reserved: Not set
....0 .... .... = Nonce: Not set
.... 0.... .... = Congestion Window Reduced (CWR): Not set
.... .0.... .... = ECN-Echo: Not set
.... ..0.... .... = Urgent: Not set
.... ...1.... .... = Acknowledgment: Set
.... .... 0.... .... = Push: Not set
.... .... .0.. .... = Reset: Not set
.... .... ..0.. .... = Syn: Not set
.... .... ...0 = Fin: Not set
```

Ack

- 뭔가를 잘 받았다는 의미로 보낸다.

```
.... 0000 0001 0000 = Flags: 0x010 (ACK)
000. .... .... = Reserved: Not set
....0 .... .... = Nonce: Not set
.... 0.... .... = Congestion Window Reduced (CWR): Not set
.... .0.... .... = ECN-Echo: Not set
.... ..0.... .... = Urgent: Not set
.... ...1.... .... = Acknowledgment: Set
.... .... 0.... .... = Push: Not set
.... .... .0.. .... = Reset: Not set
.... .... ..0.. .... = Syn: Not set
.... .... ...0 = Fin: Not set
```

Urgent

- 서비스를 먼저 출력해주는 서비스
- 긴급한 데이터
- 하지만 설정하려면 소프트웨어 상으로 설정해야한다 하지만 그냥 놔두면 아무것도 안한다.

```
000. .... .... = Reserved: Not set
....0 .... .... = Nonce: Not set
.... 0.... .... = Congestion Window Reduced (CWR): Not set
.... .0.... .... = ECN-Echo: Not set
.... ..0.... .... = Urgent: Not set
.... ...1.... .... = Acknowledgment: Set
.... .... 0.... .... = Push: Not set
.... .... .0.. .... = Reset: Not set
.... .... ..0.. .... = Syn: Not set
.... .... ...0 = Fin: Not set
```

다음은 '3-Way 핸드 쉐이킹' 과정에서 사용하는 TCP 플래그의 10 진수, 16 진수 값이다.

URG	ACK	PSH	RST	SYN	FIN	10 진수	16 진수	tcp.flag
2^5	2^4	2^3	2^2	2^1	2^0			
32	16	8	4	2	1			
0	0	0	0	1	0	2	0x02	syn
0	1	0	0	1	0	18	0x12	syn+ack
0	1	0	0	0	0	16	0x10	ack

계산기 >> 프로그래머 >> 10진수 변환

3) Fin 플래그를 이용한 TCP 정상 종료



- ① 클라이언트가 TCP 종료를 하기 위해서 Fin 세그먼트를 전송한다..
- ② Fin 세그먼트를 전송한 클라이언트의 TCP 연결 상태는 'FIN_WAIT_1'이 된다.
- ③ Fin 세그먼트를 수신한 서버는 TCP 연결 상태를 'CLOSE_WAIT'로 전환한다.
- ④ 그리고 서버는 클라이언트로 Ack 세그먼트를 전송한다.
- ⑤ Ack 세그먼트를 수신한 클라이언트는 TCP 연결 상태를 'FIN_WAIT_2'로 전환한다.
- ⑥ 서버는 클라이언트로 Fin 세그먼트를 전송한다.
- ⑦ Fin 세그먼트를 전송한 서버의 TCP 연결 상태는 'LAST_ACK'가 된다.
- ⑧ Fin 세그먼트를 수신한 클라이언트는 TCP 연결 상태를 'TIME_WAIT'로 전환한다.
- ⑨ 그리고 클라이언트는 서버로 Ack 세그먼트를 전송한다.
- ⑩ Ack 세그먼트를 수신한 서버는 TCP 연결 상태를 'CLOSED'로 전환한다.
- ⑪ 클라이언트가 전송한 Ack 를 서버가 수신할 수 있도록 일정 시간 대기한다.
(만약, 대기 시간이 없다면, 서버가 Ack 를 못받을 경우 FIN 를 재접속하기 때문이다.)
- ⑫ 대기 시간이 경과되면 클라이언트의 TCP 연결 상태가 'CLOSED'로 전환되면서 TCP 연결이 종료된다

- 서버는 Client가 Finish를 받을때까지 그리고 서버가 Ack를 받을때까지 Fin을 보낸다.
- 그리고 서버가 Ack를 받으면 바로 닫힌다.
- Client가 서버에서 또 정보를 받을수도 있기 때문에 240초라는 대시시간을 받는다.
- 240초가 지난 후에 컴퓨터를 종료한다.

4) 데이터 스트림 서비스

데이터를 세그먼트 단위로 생성하여 전송 및 수신 처리하는 기능이다. 이를 통해서 전송률과 처리률을 효율적으로 운영할 수 있다. 다음과 같이 순서 번호와 확인 번호를 이용한 'Stop & Wait', 'Sliding Window'라는 흐름 제어 기능이 필요하며, 현재 TCP에서는 'Sliding Window' 기법을 사용하고 있다.

- 큰 데이터를 통째로 들고오는 것보다 분해해서 들고오면 더 효율적이다.
- 분할된 데이터 단위를 Segment 라고 부른다.
- 수신측에서는 분할된 데이터를 받으면 다시 조립해야 한다.
 - 다시 원래대로 조립하려면 번호가 필요하다
 - Sequence Number (순서 번호)
 - 데이터를 보낼 때 쓰는 번호
 - Ark Number (아까랑 다른 에크)
 - Segment 몇 번을 잘 받았음을 알리는 번호이다. (옆면 윗면 등)
 - 데이터를 받을 때 쓰는 번호
- 흐름 제어 기능
 1. Stop & Wait
 - a. 2 번 segment를 받으려면 1 번 segment를 받아야 한다.
 - b. 순서대로 받아야 한다.
 - c. 수신하는 양이 많아지는 게 단점이다.

유형	내용
Stop & Wait	<ul style="list-style-type: none">- 송신한 세그먼트에 대한 Ack 를 수신해야지만, 그 다음 세그먼트를 전송한다.- 다음 세그먼트 송신 처리에 대한 지연이 발생하고 수신하는 Ack 양이 많다. <p>① 송신측에서 1 번 세그먼트 전송</p> <p>[1] →</p> <p>② 1 번 세그먼트에 대한 Ack 를 수신해야지만, 송신측에서 2 번 세그먼트 전송</p> <p>← ack</p> <p>[2] →</p> <p>③ 2 번 세그먼트에 대한 Ack 를 수신해야지만, 송신측에서 3 번 세그먼트 전송</p> <p>← ack</p> <p>[3] →</p>

ex)

나는홍길동입니다

나
는
홍
길
동
입
니
다

2. Sliding Window

a. 데이터 수 통신을 하려면 window 크기를 협의한다.

b. 윈도우를 계속 옆으로 밀어서 슬라이딩이다.

i. 내가 받을 수 있는 segment 크기이다.

ii. ex)

A: 20 B: 100

B가 A한테 보낼려면 20으로 보내야한다.

수신측 기준으로 맞춘다.

- 수신측 윈도우 크기에 맞게 송신측에서 세그먼트 양 조정하여 전송한다.

- 세그먼트 송신 지연 발생과 Ack 양을 최소화한다.

① 송신측 윈도우 크기가 '5'라면, 1~5 번 세그먼트 전송 가능

[1][2][3][4][5] 6 7 8 9

② 그러나 수신측 윈도우 크기가 '2'라면, 송신측은 1~2 번 세그먼트만 전송 실시

1 2 [3][4][5] ->

③ 수신측으로부터 1~2 번 세그먼트에 대한 Ack 를 받으면, 송신측은 자신의 윈도우 크기만큼 슬라이딩 윈도우 실시

1 2 [3][4][5][6][7] <- ack

c. segment의 크기를 조절하여 한번에 여러개를 보내고 Ack를 받을 수 있다.

i. 수신측의 window를 고려하여 보낸다.

ii. 조금 더 효율적이다.

iii. ex)

나는 흥길동입니다

나는
흥길동

입니다

ex) Sliding Window method:

- 다음은 힙의된 윈도우 크기를 이용하여 세그먼트를 전송한 예제이다.

[1~1000][1001~2000][2001~3000][3001~4000][4001~5000]

(세그먼트: 1000byte)

클라이언트

서버

윈도우 크기(5000) ->

<- 윈도우 크기(10000)

<- seq=1 (~1000)

<- seq=1001 (~2000)

<- seq=2001 (~3000)

<- seq=3001 (~4000)

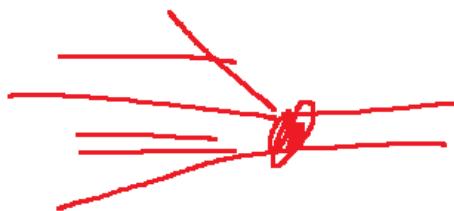
<- seq=4001 (~5000)

ack=5001 ->

5) 혼잡 제어 기능

혼잡 발생시 전송률을 최소화하여 혼잡을 줄이는 기능이다. TCP에만 있는 필드이기 때문에 잘 사용하지 않으며, 라우터에서 QoS 정책을 별도로 구성하여 적용하고 있다.

- 들어오는 속도보다 나가는 속도가 느린 경우 (트래픽 혼잡)



- TCP에만 있는 필드이다. (잘 사용하지 않는다)
- 라우터에서 QoS 정책을 별도로 구성 (모든 패킷들을 대상)

6) 오류 검사

- 수신한 세그먼트에 대한 손상 여부 판단하여, 세그먼트를 드랍하는 기능이다.

7) 재전송 기능 (Ack를 못받으면 재전송한다)

송신한 세그먼트에 대한 Ack를 재전송 시간 초과 타이머(RTO) 안에 수신하지 못하면, 해당 세그먼트가 손상되었거나, 손실된 것으로 간주하여 세그먼트를 재전송한다. RTO 타이머는 가변적인 시간을 갖고 있다.

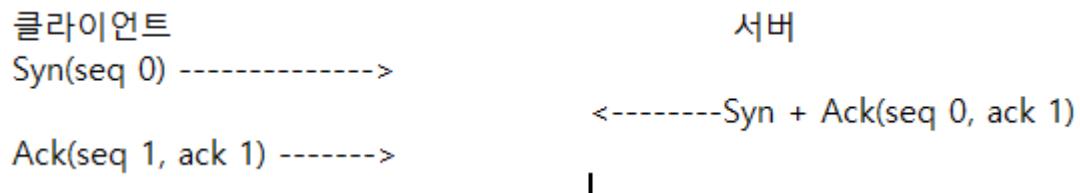
8) Window Size

처리할 수 있는 세그먼트양을 표기하는 필드이다. 송신측에게 자신의 윈도우 사이즈를 알려주면, 송신측에서 그만큼의 세그먼트를 한번에 전송하고 수신측이 다 처리했는지 확인 후에 다음 세그먼트를 전송한다.
윈도우 사이즈는 가변적이기 때문에 상황에 따라서 증가되거나 감소된다.

- 크기는 가변적이다 (상황에 따라 바뀐다)

9) TCP를 사용하는 서비스

- HTTP(80), HTTPs/SSL(443), Telnet(23), SSH(22), FTP(21), FTP-Data(20), SMTP(25), POP3(110), MySQL(3306)
- Tomcat(8080), NFS(2049), RPC(111), SMB(139,445)



2. UDP (User Datagram Protocol)

헤더 크기는 8byte 이다. Layer 4 계층 프로토콜이며, 상대방과 연결 과정 없이 데이터 요청 및 응답을 바로 실시하는 비연결 지향성 특징을 갖고 있다.

- TCP와 마찬가지로 Layer 4 계층 프로토콜이다.
- 바로 데이터 요청과 응답을 실시한다. (비연결 지향성 특징)

1) 비연결 지향성 프로토콜

UDP는 연결성이 없기 때문에 데이터 요청 및 응답을 바로 실시한다.

55 1.374	172.16.5.254	168.126.63.1	DNS	73 Standard query 0x3968	클라이언트	서버
60 1.415	168.126.63.1	172.16.5.254	DNS	221 Standard query response		
	172.16.5.254:53258			168.126.63.1:53		
		데이터/서비스 요청 ->				
55		① DNS(query)			<- 데이터/서비스 응답	
60					DNS(response) ②	

TCP에서 제공하는 다음과 같은 기능은 지원하지 않는다. 단, 오류 검사 기능은 지원한다.

- '3-Way 핸드 쉐이킹' 동작
- 데이터 스트림 서비스
- 흐름 제어 기능
- 혼잡 제어 기능
- 재전송 기능
- Window Size

2) UDP를 사용하는 서비스

- DNS(53), TFTP(69), DHCP Server(67), DHCP Client(68), SNMP(161), NTP(123), NMB(137,138), Syslog(514)

3) UDP 헤더 내용 (8byte)

```
User Datagram Protocol, Src Port: 60669 (60669), Dst Port: 53 (53)
Source Port: 60669 (60669)
Destination Port: 53 (53)
Length: 39
Checksum: 0x6760 [validation disabled]
[Stream index: 22]
```

TCP vs UDP

TCP	UDP
신뢰성	
>	
신속성	
<	

3. IP (Internet Protocol)

헤더 크기는 20byte 이다. Layer 3 계층 프로토콜이며, 비연결 지향성 특징을 갖고 있다. IP 프로토콜은 로컬 환경에서 리모트 환경으로 데이터 전송을 하기 위해서 사용한다. IP 헤더 내용은 다음과 같다.

- 20byte
- Layer 3 계층 프로토콜
- 비연결성
- **로컬 환경에서 리모트 환경으로 데이터 전송을 하기 위해서 사용한다.**

```
Internet Protocol Version 4, Src: 172.16.5.254 (172.16.5.254), Dst: 168.126.63.1 (168.126.63.1)
Version: 4
Header Length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
    0000 00.. = Differentiated Services Codepoint: Default (0x00)
    .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)
Total Length: 59
Identification: 0xc9a5 (51621)
Flags: 0x00
    0... .... = Reserved bit: Not set
    .0... .... = Don't fragment: Not set
    ..0. .... = More fragments: Not set
Fragment offset: 0
Time to live: 128
Protocol: UDP (17)
Header checksum: 0xd77e [validation disabled]
Source: 172.16.5.254 (172.16.5.254)
Destination: 168.126.63.1 (168.126.63.1)
```

항목	내용
Version	IP 버전 표기(IPv4, IPv6)
Header Length	IP 헤더 크기
Differentiated Services Field	QoS 정책 구현시 사용하는 필드
Identification	패킷 식별자
Flags	IP Fragments 가 실시된 패킷을 알리는 필드(More fragments 가 '1'로 설정됨)
Fragment offset	IP Fragments 가 실시된 누적된 패킷 크기
Time to live(0~255)	패킷이 네트워크상에 전송될 수 있는 시간(시간 단위는 라우터이다.)
Protocol	상위 프로토콜 정보
Header checksum	헤더 오류 검사(불필요한 필드이기 때문에 IPv6 프로토콜에서는 삭제함)
Source	출발지 IP 주소
Destination	목적지 IP 주소

3bit

IP Precendence

000	0	
001	1	
010	2	
011	3	<-A
100	4	
101	5	<-B
110	6	
111	7	

6bit (DSCP)

```
Frame 55: 73 bytes on wire (584 bits), 73 bytes captured (584 bits)
Ethernet II, Src: Realtek_14:62:ba (00:e0:4c:14:62:ba), Dst: Cisco_31:81:b1 (00:13:60:31:81:b1)
  Destination: Cisco_31:81:b1 (00:13:60:31:81:b1)
  Source: Realtek_14:62:ba (00:e0:4c:14:62:ba)
  Type: IP (0x0800)
Internet Protocol Version 4, Src: 172.16.5.254 (172.16.5.254), Dst: 168.126.63.1 (168.126.63.1)
  Version: 4
  Header Length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECT))
    0000 00.. = Differentiated Services Codepoint: Default (0x00)
    .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport)
```

MTU 크기 (기본값 1500byte)

4000 byte

		id	more fragment	fragment offset
1500byte		17	1 (처음분할)	0
1500byte		17	1	1500byte (누적)
1000byte		1y	0 (분할x)	3000byte (누적)

Flags: 0x00

```
0... .... = Reserved bit: Not set
.0... .... = Don't fragment: Not set
..0. .... = More fragments: Not set
```

IP의 Time to live (TTL) (0~255) 기능

■ Flags: 0x00

0.... = Reserved bit: Not set
.0... = Don't fragment: Not set
.0. = More fragments: Not set

Fragment offset: 0

Time to live: 128

- 네트워크가 바뀔때마다 차감된다.
- Windows(162) → R1(161) → R2(160) → R3(159) → R4 → Linux
- 패킷이 네트워크상에 전송될 수 있는 시간(시간 단위는 라우터이다)
- 원래 목적은 네트워크 상에서 루프가 발생했을 때 방지하기 위해서이다.

```
Packet Tracer PC Command Line 1.0
PC>ping 10.1.1.1

Pinging 10.1.1.1 with 32 bytes of data:

Request timed out.
Reply from 10.1.1.1: bytes=32 time=8ms TTL=126
Reply from 10.1.1.1: bytes=32 time=1ms TTL=126
Reply from 10.1.1.1: bytes=32 time=6ms TTL=126

Ping statistics for 10.1.1.1:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 8ms, Average = 5ms

PC>
```

TTL = 126

Apc to Bpc에 라우터가 2개 있다.

```
PC>ping 13.13.12.2

Pinging 13.13.12.2 with 32 bytes of data:

Reply from 13.13.12.2: bytes=32 time=1ms TTL=254
Reply from 13.13.12.2: bytes=32 time=3ms TTL=254
Reply from 13.13.12.2: bytes=32 time=3ms TTL=254
Reply from 13.13.12.2: bytes=32 time=4ms TTL=254

Ping statistics for 13.13.12.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 4ms, Average = 2ms

PC>
```

TTL = 254

중간에 라우터가 1대 있다.

```
PC>ping 192.168.1.254

Pinging 192.168.1.254 with 32 bytes of data:

Reply from 192.168.1.254: bytes=32 time=0ms TTL=255
Reply from 192.168.1.254: bytes=32 time=1ms TTL=255
Reply from 192.168.1.254: bytes=32 time=0ms TTL=255
Reply from 192.168.1.254: bytes=32 time=0ms TTL=255

Ping statistics for 192.168.1.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

중간에 라우터가 없다 (TTL이 줄어들지 않았다)

같은 네트워크라는 의미다. 255

```
PC>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time=0ms TTL=128

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

PC>

줄어든게 없다 128

```
C:\Users\Administrator>ping 168.126.63.1

Ping 168.126.63.1 32바이트 데이터 사용:
168.126.63.1의 응답: 바이트=32 시간=3ms TTL=53

168.126.63.1에 대한 Ping 통계:
    패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),
왕복 시간(밀리초):
    최소 = 3ms, 최대 = 3ms, 평균 = 3ms

C:\Users\Administrator>
```

63 -> 53

중간에 라우터가 10대가 있다.

```
C:\#Users\#Administrator>ping www.google.com

Ping www.google.com [142.250.196.132] 32바이트 데이터 사용:
142.250.196.132의 응답: 바이트=32 시간=85ms TTL=109

142.250.196.132에 대한 Ping 통계:
    패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),
    왕복 시간(밀리초):
        최소 = 85ms, 최대 = 85ms, 평균 = 85ms
```

라우터 10대

```
C:\#Users\#Administrator>ping 192.168.11.30

Ping 192.168.11.30 32바이트 데이터 사용:
192.168.11.30의 응답: 바이트=32 시간<1ms TTL=128

192.168.11.30에 대한 Ping 통계:
    패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),
    왕복 시간(밀리초):
        최소 = 0ms, 최대 = 0ms, 평균 = 0ms
```

학원 PC 같은 네트워크 이기때문에 TTL이 줄어들지 않는다.

4. 네트워크 계층 모델

1) TCP/IP 5 Layer

계층	프로토콜		주소 유형
상위계층 (Application)	http, https(ssl), telnet ssh, ftp, ftp-data, smtp, pop3	dns, tftp, snmp, ntp dhcp server/client, syslog	-
Layer 4 (Transport)	TCP	UDP	포트 번호
	통신 연결	O	
	'3-Way 핸드 쉐이킹' 동작	O	
	데이터 스트림 서비스	O	
	홀잡 제어 기능	O	
	재전송 기능	O	
	Window Size 오류 검사	O	
Layer 3 (Internet)	IP		IP 주소
	로컬 환경에서 리모트 환경으로 데이터 전송 TTL를 이용하여 패킷 루프 방지 기능 및 거리 측정		
Layer 2 (Network Interface)	Ethernet		MAC 주소
Layer 1 (Physical)	전기 신호 변환 및 출력력		bit(0,1)

ex)

Ethernet | IP | UDP | DNS 요청

2) OSI 7 Layer

데이터 생성과 전송 과정을 7개 계층으로 제시한 모델이다. OSI 7 Layer를 이해하고 있다면, 네트워크 작업 및 장애 처리 접근을 손쉽게 할 수 있다.

Layer 7 애플리케이션	서비스가 구현되는 계층	
Layer 6 프레젠테이션	서비스를 어떤 방식으로 표현할 것인지를 결정	
Layer 5 세션	OS/서비스 간에 논리적인 연결	
----- 상위 계층(서비스 구현, OS/응용 프로그램 담당)		
Layer 4 트랜스포트	TCP	UDP
Layer 3 네트워크	IP	
Layer 2 데이터 링크	Ethernet	
Layer 1 물리 계층	전기 신호 변환, 출력력	
----- 하위 계층(전송 담당, 네트워크 장비/전송 프로토콜 담당)		

- Application 계층
- 사용자 입장에서 서비스가 구현되는 계층
- 응용 프로그램

5. ICMP (Internet Control Message Protocol)

- IP 프로토콜을 이용하여 데이터 전송이 가능한지 확인하기 위해서 메시지를 생성하여 요청 및 응답을 실시하는 프로토콜이다.
- 부록 (결제부록)
 - 있어도 되고 없어도 된다.
- 서버를 연결한뒤 테스트를 할 때 쓰는 프로토콜
 - 데이터가 전송되는지 안되는지를 테스트할때 사용
- 요청 메시지와 응답 메시지를 만들어주고 전송해주는 프로토콜

1) 기본적인 ICMP 메세지 유형

유형	타입	내용
echo	8	ICMP 요청 메세지
echo-reply	0	ICMP 응답 메세지
Destination Unreachable	3	ICMP 목적지 도달 불가능 응답 메세지
TTL Exceeded	11	TTL 이 만료된 응답 메세지

8번은 응답, 0번은 무응답

Internet Control Message Protocol
Type: 0 (Echo (ping) reply)
Code: 0
Checksum: 0x505c [correct]
Identifier (BE): 512 (0x0200)
Identifier (LE): 2 (0x0002)
Sequence number (BE): 768 (0x0300)
Sequence number (LE): 3 (0x0003)
[Request frame: 1359]
[Response time: 0.423 ms]
Data (32 bytes)

2) 기본적인 'Destination Unreachable' 메세지 유형

유형	코드	내용
Network Unreachable	0	목적지에 대한 경로가 없는 경우
Host Unreachable	1	최종 목적지 호스트에 도달할 수 없는 경우
Protocol Unreachable	2	목적지에서 특정 프로토콜을 사용할 수 없는 경우
Port Unreachable	3	목적지 호스트에서 특정 포트가 닫혀 있는 경우

1. 목적지 경로가 없음
2. 목적지 네트워크까지 왔는데 시스템이 없는 경우
3. 목적지에서 특정 프로토콜 못사용하는 경우
4. 목적지 시스템까지 도착은 했는데 포트가 안 열려있는 경우 (UDP)
 - a. 많이 사용함

3) ICMP 헤더 내용

ICMP Echo	ICMP Echo-Reply	Destination Unreachable
Internet Control Message Protocol Type: 8 (Echo (ping) request) Code: 0 Checksum: 0x485c [correct] Identifier (BE): 512 (0x0200) Identifier (LE): 2 (0x0002) Sequence number (BE): 768 (0x0300) Sequence number (LE): 3 (0x0003) [Response frame: 1362]	Internet Control Message Protocol Type: 0 (Echo (ping) reply) Code: 0 Checksum: 0x505c [correct] Identifier (BE): 512 (0x0200) Identifier (LE): 2 (0x0002) Sequence number (BE): 768 (0x0300) Sequence number (LE): 3 (0x0003) [Request frame: 1359] [Response time: 0.423 ms]	Internet Control Message Protocol Type: 3 (Destination unreachable) Code: 3 (Port unreachable) Checksum: 0xa76d [correct]

4) Ping/tracer 명령어

Ping 명령어:

'ping' 명령어를 실행하면 ICMP 메세지를 생성하고 전송하여 IP 패킷이 전송이되는지 확인할 수 있다.



ex) 4-2 데이터 전송 프로토콜 pkt 파일을 실행하여 ping test를 실시한다.

```
Packet Tracer PC Command Line 1.0
PC>
PC>ping 10.1.1.1

Pinging 10.1.1.1 with 32 bytes of data:

Request timed out.
Reply from 10.1.1.1: bytes=32 time=2ms TTL=126
Reply from 10.1.1.1: bytes=32 time=2ms TTL=126
Reply from 10.1.1.1: bytes=32 time=2ms TTL=126

Ping statistics for 10.1.1.1:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 2ms, Average = 2ms
```

요청보내고 응답까지 2ms가 걸린다.

ping -t [ip]

- 계속 응답을 보낸다

Ctrl + C

- 정지시킨다.

tracert 명령어:

```
PC>tracert 10.1.1.1

Tracing route to 10.1.1.1 over a maximum of 30 hops:

 1  0 ms      0 ms      0 ms      192.168.1.254
 2  7 ms      0 ms      1 ms      13.13.12.2
 3  0 ms      4 ms      5 ms      10.1.1.1

Trace complete.

PC>
```

- 어디 라우터를 통해서 보내는지 알 수 있다.
- 출발지에서 목적지까지 경로를 추적할 수 있다.
- 어디까지 가는지 알 수 있다.
- 라우터의 IP주소를 알 수 있다.
- ICMP 메세지를 생성하고 전송한다.

어떻게 작동되느냐:

ICMP Echo

```
-----|ICMP
SA 192.168.1.100
DA 10.1.1.1
-----|IP(TTL = 1)
```

- 처음에는 ttl을 0으로 함으로서 처음 닿는 라우터로밖에 안가게 한다.

라우터에 닿는다. 하지만 목적지에 안닿는다

ttl을 2로 설정한다 +1

라우터에 닿는다. 같은 네트워크지만 목적지가 아니다.

ttl을 3으로 설정한다. +1 >> 목적지에 닿는다

다시 값을 관리자 PC로 보낸다.

```
C:\Users\Administrator>tracert 8.8.8.8

최대 30홉 이상의
dns.google [8.8.8.8](으)로 가는 경로 추적:

 1  <1 ms    <1 ms    <1 ms  192.168.11.1
 2  *          *          *          요청 시간이 만료되었습니다.
 3  2 ms      2 ms      2 ms  100.80.28.13
 4  2 ms      1 ms      1 ms  100.80.28.13
 5  *          3 ms      2 ms  1.213.145.13
 6  *          3 ms      2 ms  61.42.201.237
 7  10 ms     3 ms      4 ms  1.213.114.9
 8  37 ms     38 ms     37 ms  100.67.30.2
 9  38 ms     39 ms     38 ms  100.67.30.50
10  38 ms     39 ms     38 ms  142.250.168.244
11  39 ms     39 ms     39 ms  64.233.175.91
12  61 ms     61 ms     62 ms  74.125.253.93
13  38 ms     38 ms     38 ms  dns.google [8.8.8.8]

추적을 완료했습니다.
```

'tracert' 명령어를 실행하면 ICMP 메세지를 생성하고 전송하여 목적지까지 라우팅을 실시한 라우터의 IP 주소를 확인할 수 있다. 이를 통해서 출발지에서 목적지까지 경로를 추적할 수 있다.

관리자_PC>tracert 10.1.1.1

```
Tracing route to 10.1.1.1 over a maximum of 30 hops:
```

```
① 1 0 ms      0 ms      0 ms      192.168.1.254  
② 2 2 ms      0 ms      2 ms      13.13.12.2  
③ 3 6 ms      1 ms      0 ms      10.1.1.1
```

```
④ Trace complete.
```

- ① TTL=1로 'Echo'를 전송하고 'TTL Exceeded' 응답을 수신한 내용이다.
- ② TTL=2로 'Echo'를 전송하고 'TTL Exceeded' 응답을 수신한 내용이다.
- ③ TTL=3으로 'Echo'를 전송하고 'Echo-Reply'를 수신한 내용이다.
- ④ 목적지로부터 'Echo-Reply'를 수신했기 때문에 경로 추적 테스트 완료되었다.

5) TTL이 만료된 경우

```
C:\Users\Administrator>ping -i 3 168.126.63.1
```

```
Ping 168.126.63.1 32 바이트 데이터 사용:  
61.78.42.165 의 응답: 전송하는 동안 TTL 이 만료되었습니다.  
61.78.42.165 의 응답: 전송하는 동안 TTL 이 만료되었습니다.  
61.78.42.165 의 응답: 전송하는 동안 TTL 이 만료되었습니다.  
61.78.42.165 의 응답: 전송하는 동안 TTL 이 만료되었습니다.
```

```
Internet Control Message Protocol  
Type: 11 (Time-to-live exceeded)  
Code: 0 (Time to live exceeded in transit)  
Checksum: 0x9fa3 [correct]
```

```
168.126.63.1에 대한 Ping 통계:
```

```
패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),
```

6) ICMP 공격을 방지하기 위해서 방화벽에서 차단한 경우

```
C:\Users\Administrator>ping www.naver.com
```

```
Ping www.naver.com.nheos.com [223.130.200.107] 32 바이트 데이터 사용:
```

```
요청 시간이 만료되었습니다.  
요청 시간이 만료되었습니다.  
요청 시간이 만료되었습니다.  
요청 시간이 만료되었습니다.
```

```
223.130.200.107에 대한 Ping 통계:
```

```
패킷: 보냄 = 4, 받음 = 0, 손실 = 4 (100% 손실),
```

6. ARP(Address Resolution Protocol)

목적지 IP 주소에 대한 MAC 주소를 설정하는 프로토콜이다. 만약, 목적지 IP 주소에 대한 MAC 주소가 ARP 테이블에 없는 경우, 다음과 같이 ARP 요청 및 응답 실시한다.

1) ARP 동작 과정

'4-2.데이터 전송 프로토콜.pkt' 파일 환경에서 ARP 동작 과정을 알아보도록 한다.

```
C_PC>arp -d          // arp 테이블 정보 삭제  
C_PC>arp -a          // arp 테이블 확인
```

```
No ARP Entries Found
```

① C에서 D로 'ICMP Echo'를 전송할 때, D에 대한 MAC 주소 정보가 ARP 테이블에 없는 경우

```
C(10.1.1.1)      ->      D(10.1.1.2)  
00e0.f739.d44a          MAC...?  
  
ICMP Echo  
----- ICMP  
SA 10.1.1.1  
DA 10.1.1.2  
----- IP  
SA 00E0.F739.D44A  
DA X      // ARP 테이블에 D(10.1.1.2)에 대한 MAC 주소 정보가 없기 때문에 목적지 MAC 주소 설정 못함  
----- ETH X      // 그렇기 때문에 Ethernet 프로토콜 인캡슐레이션 실패된다.
```

② C는 D(10.1.1.2)에 대한 MAC 주소를 학습하기 위해서 ARP 요청 메세지를 브로드캐스트로 전송한다.

```
C(10.1.1.1)      D(10.1.1.2)  
00e0.f739.d44a          000A.4196.458A  
  
"10.1.1.2, MAC...?"  
----- ARP 요청 메세지  
SA 00E0.F739.D44A  
DA FFFF.FFFF.FFFF      // 브로드캐스트  
----- ETH
```

③ D는 자신의 IP 주소와 MAC 주소 정보를 ARP 응답 메세지를 생성하여 유니캐스트로 C에게 전송한다.

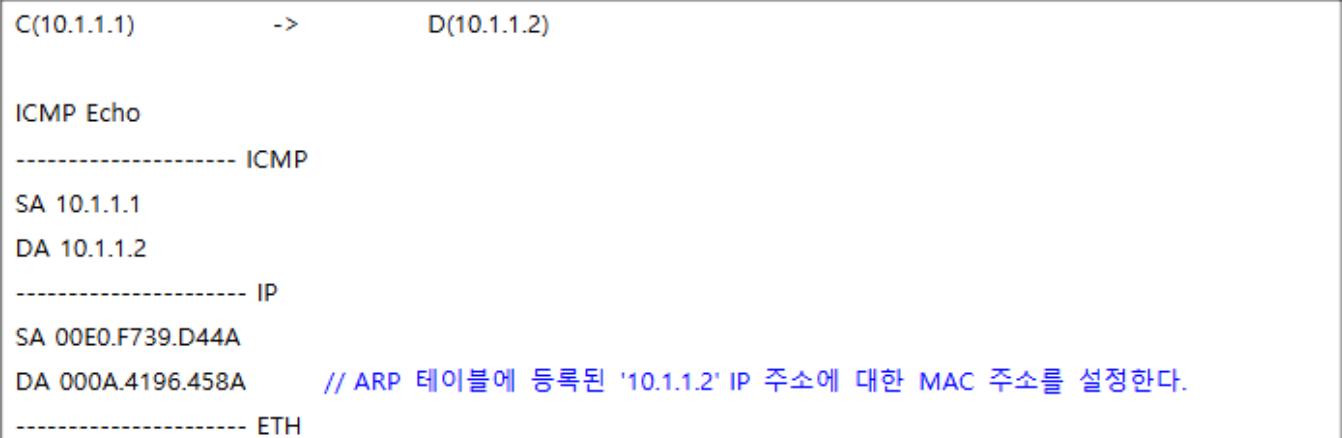


④ ARP 응답 메세지를 수신한 C는 ARP 테이블에 D의 IP 주소와 MAC 주소 정보를 등록한다.

C_PC>arp -a

Internet Address	Physical Address	Type
10.1.1.2	000A.4196.458A	dynamic

⑤ C는 ARP 테이블에 D에 대한 IP 주소와 MAC 주소 정보가 있기 때문에 'ICMP Echo'를 전송할 수 있다.



⑥ C에서 D(10.1.1.2), E(10.1.1.3)로 Ping 테스트를 실시한 이후, ARP 테이블 정보를 확인한다.

C_PC>**ping 10.1.1.2**

```
Pinging 10.1.1.2 with 32 bytes of data:
```

```
Reply from 10.1.1.2: bytes=32 time=0ms TTL=128
Reply from 10.1.1.2: bytes=32 time=0ms TTL=128
Reply from 10.1.1.2: bytes=32 time=0ms TTL=128
Reply from 10.1.1.2: bytes=32 time=1ms TTL=128
```

```
Ping statistics for 10.1.1.2:
```

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

C_PC>**ping 10.1.1.3**

```
Pinging 10.1.1.3 with 32 bytes of data:
```

```
Reply from 10.1.1.3: bytes=32 time=1ms TTL=128
Reply from 10.1.1.3: bytes=32 time=1ms TTL=128
Reply from 10.1.1.3: bytes=32 time=1ms TTL=128
Reply from 10.1.1.3: bytes=32 time=0ms TTL=128
```

```
Ping statistics for 10.1.1.3:
```

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

C_PC>**arp -a**

Internet Address	Physical Address	Type
10.1.1.2	000a.4196.458a	dynamic
10.1.1.3	000c.8562.d6c2	dynamic

2) 다른 네트워크로 ARP 요청 불가

라우터는 브로드캐스트를 다른 네트워크로 전송하지 않는다. 그렇기 때문에 다른 네트워크 환경에 있는 시스템에 대한 ARP 요청은 불가능하다. 그래서 로컬 PC는 Gateway IP 주소에 대한 MAC 주소를 학습하기 위한 ARP 요청을 실시하여 Gateway MAC 주소를 학습한다.

C에서 서버(192.168.1.100)로 Ping 테스트를 실시한 이후, ARP 테이블 정보를 확인한다.

C_PC>**ipconfig**

FastEthernet0 Connection:(default port)

Link-local IPv6 Address..... ::
IP Address.....: 10.1.1.1
Subnet Mask.....: 255.255.255.0
Default Gateway.....: 10.1.1.254

C_PC>**ping 192.168.1.100**

Pinging 192.168.1.100 with 32 bytes of data:

Reply from 192.168.1.100: bytes=32 time=7ms TTL=126
Reply from 192.168.1.100: bytes=32 time=3ms TTL=126
Reply from 192.168.1.100: bytes=32 time=1ms TTL=126
Reply from 192.168.1.100: bytes=32 time=1ms TTL=126

Ping statistics for 192.168.1.100:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 1ms, Maximum = 7ms, Average = 3ms

PC>**arp -a**

Internet Address	Physical Address	Type
10.1.1.2	000a.4196.458a	dynamic
10.1.1.3	000c.8562.d6c2	dynamic
10.1.1.254	000a.41ad.3301	dynamic

C(10.1.1.1) -> 관리자(192.168.1.100)

ICMP Echo

----- ICMP

SA 10.1.1.1

DA 192.168.1.100

----- IP

SA 00E0.F739.D44A

DA 000A.41AD.3301 // ARP 테이블에 등록된 게이트웨이 '10.1.1.254' IP 주소에 대한 MAC 주소를 설정한다.

----- ETH

ICMP

- IP 프로토콜을 도와주는 역할

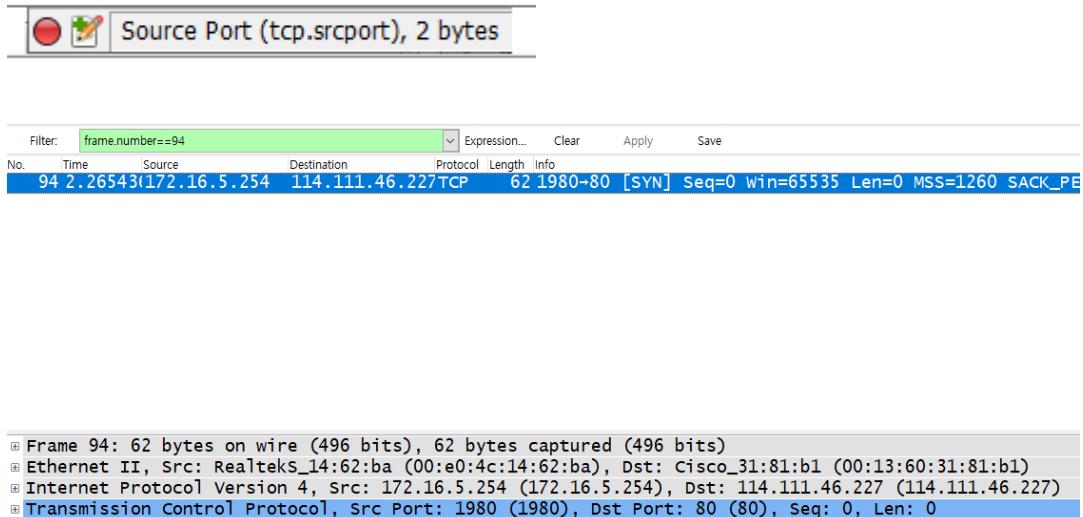
ARP

- ETH 도와주는 역할 (MAC 세팅)

제5장 와이어샤크 필터

1. 와이어샤크 필터 명령어

- '5-1.와이어샤크 필터.pcap'파일 실행
- Filter : frame.number == 94
- TCP 프로토콜 클릭 → 'Source Port' 클릭 → 좌측 하단에 필터 명령어가 () 안에 출력됨.



1) 필터 명령어

- 각각의 헤더를 클릭하여 다음 필터 명령어들을 확인한다.

항목	필터 명령어
출발지 TCP 포트	tcp.srcport
목적지 TCP 포트	tcp.dstport
출발지 UDP 포트	udp.srcport
목적지 UDP 포트	udp.dstport
출발지 IP 주소	ip.src
목적지 IP 주소	ip.dst
출발지 MAC 주소	eth.src
목적지 MAC 주소	eth.dst
TCP Syn	tcp.flags.syn
TCP Ack	tcp.flags.ack

2) 필터 관련 기호

항목	내용
&&	and
	or
==	eq
!	not
()	여러 개의 필터를 클래스로 구성

3) 필터 예제

TCP / UDP 관련 필터

```
tcp.srcport == 80 or tcp.dstport == 80  
tcp.port == 80  
  
tcp.flags.syn == 1  
tcp.flags == 0x02      // tcp.flags == 2  
  
tcp.flags.syn == 1 and tcp.flags.ack == 1  
tcp.flags == 0x12      // tcp.flags == 18  
  
tcp.flags.ack==1  
tcp.flags == 0x10      // tcp.flags == 16  
  
tcp.seq == 1  
tcp.ack == 1  
tcp.seq == 1 and tcp.ack == 1  
  
udp.srcport == 53 or udp.dstport == 53  
udp.port == 53
```

IP 관련 필터

```
ip.src == 172.16.5.254  
ip.dst == 172.16.5.254  
  
ip.src == 172.16.5.254 or ip.dst == 172.16.5.254  
ip.addr == 172.16.5.254  
  
ip.src == 172.16.5.254 && tcp  
ip.src == 172.16.5.254 && tcp.srcport == 1980  
ip.src == 172.16.5.254 && tcp.dstport == 80  
  
ip.src == 172.16.5.254 && udp  
ip.src == 172.16.5.254 && udp.dstport eq 53  
  
ip.src == 172.16.5.254 && ip.dst == 114.111.46.227 && tcp.flags == 0x02  
ip.src == 114.111.46.227 && ip.dst == 172.16.5.254 && tcp.flags == 0x12
```

1. 첫번째 쌍크
2. 두번째 싱크

```
ip.src == 172.16.5.254 && ip.dst == 114.111.46.227 && tcp.flags == 0x10  
ip.src == 172.16.5.254 && ip.dst == 114.111.46.227 && tcp.flags == 0x10 && tcp.seq == 1 && tcp.ack == 1
```

클라이언트

서버

syn ->
seq 0, ack 0

<-Syn+Ack
Seq 0, ack 1

Ack ->
seq1, ack1|

ip.src == 172.16.5.254 && ip.dst == 114.111.46.227 && tcp.flags == 0x10\
- flag중에 ack가 다 나오게

ip.src == 172.16.5.254 && ip.dst == 114.111.46.227 && tcp.flags == 0x10 && tcp.seq == 1
&& tcp.ack == 1
- 3-way handshaking 중 3번째 ack만 나오게

3) ! 관련 필터

```
!arp // not arp  
ip.addr == 172.16.5.254 && !tcp && !udp  
!ip && !ipv6
```

- 2번째 : 출발지 IP가 5.254 이고 TCP, UDP 를 뺀 아이피를 뽑아라

4) ICMP 관련 필터

```
icmp  
icmp.type == 8  
icmp.type eq 0
```

5) Ethernet 관련 필터

```
eth  
eth.src == 00:00:0c:92:ab:2c  
eth.dst == 00:e0:4c:14:62:ba  
eth.src eq 00:00:0c:92:ab:2c or eth.dst eq 00:13:60:31:81:b1  
eth.src eq 00:00:0c:92:ab:2c and eth.dst eq 00:e0:4c:14:62:ba  
eth.dst eq ff:ff:ff:ff:ff:ff
```

- eth : 이더넷 프로토콜 쓰는애들 다 나옴
- 맨아래: 브로드캐스트 (목적지)

6) ARP 관련 필터

```
arp
arp.opcode == 1          // arp.opcode == 2
arp.src.proto_ipv4 == 172.16.5.254
arp.src.hw_mac == 00:e0:4c:14:62:ba
arp.src.proto_ipv4 == 172.16.5.254 && arp.src.hw_mac == 00:e0:4c:14:62:ba
```

- opcode가 2번이면 응답이다.
- Target MacAddress 가 없으면 요청

7) HTTP 관련 필터

http http.host eq lm3.cafe.naver.com	http.request.method == GET http.request.method matches "(?i)get"
---	---

- 대소문자를 가림
- 구분없이 하려면 matches "(?i)"

와이어샤크 필터 예제:

2) '192.168.1.201' TCP 3-Way 핸드 쉐이킹 & HTTP(TCP 80) 내용 필터

PC	서버
192.168.1.254	192.168.1.201
70:85:C2:0A:E8:F6	
Syn(0x02) ->	
seq 0	<- Syn+Ack(0x12)
	seq 0, ack 1
Ack(0x10) ->	
seq 1, ack 1	

TCP 3-way handshaking:

1. ip.src == 192.168.1.254 and ip.dst == 192.168.1.201 && tcp.dstport == 80 && tcp.flags == 0x02 && tcp.seq == 0
2. ip.src == 192.168.1.201 and ip.dst == 192.168.1.254 && tcp.srcport == 80 && tcp.flags == 0x12 && tcp.seq == 0 && tcp.ack == 1
3. ip.src == 192.168.1.254 and ip.dst == 192.168.1.201 && tcp.dstport == 80 && tcp.flags == 0x10 && tcp.seq == 1 && tcp.ack == 1

HTTP(TCP 80):

1. (ip.src == 192.168.1.254 and ip.dst == 192.168.1.201 && http) or (ip.src == 192.168.1.201 and ip.dst == 192.168.1.254 && http)

Answer:

```
(ip.src == 192.168.1.254 and ip.dst == 192.168.1.201 && tcp.dstport == 80 && tcp.flags == 0x02 && tcp.seq == 0) or (ip.src == 192.168.1.201 and ip.dst == 192.168.1.254 && tcp.srcport == 80 && tcp.flags == 0x12 && tcp.seq == 0 && tcp.ack == 1) or (ip.src == 192.168.1.254 and ip.dst == 192.168.1.201 && tcp.dstport == 80 && tcp.flags == 0x10 && tcp.seq == 1 && tcp.ack == 1) or (ip.src == 192.168.1.254 and ip.dst == 192.168.1.201 && http) or (ip.src == 192.168.1.201 and ip.dst == 192.168.1.254 && http)
```

3) '192.168.1.201' TCP 3-Way 핸드 쉐이킹 및 FTP(TCP 21) 내용 필터

PC	서버
192.168.1.254	192.168.1.201
70:85:C2:0A:E8:F6	
Syn(0x02) ->	
seq 0	<- Syn+Ack(0x12)
	seq 0, ack 1
Ack(0x10) ->	
seq 1, ack 1	

Answer: (포트번호 80 → 21, http → FTP)

```
(ip.src == 192.168.1.254 and ip.dst == 192.168.1.201 && tcp.dstport == 21 && tcp.flags == 0x02 && tcp.seq == 0) or (ip.src == 192.168.1.201 and ip.dst == 192.168.1.254 && tcp.srcport == 21 && tcp.flags == 0x12 && tcp.seq == 0 && tcp.ack == 1) or (ip.src == 192.168.1.254 and ip.dst == 192.168.1.201 && tcp.dstport == 21 && tcp.flags == 0x10 && tcp.seq == 1 && tcp.ack == 1) or (ip.src == 192.168.1.254 and ip.dst == 192.168.1.201 && ftp) or (ip.src == 192.168.1.201 and ip.dst == 192.168.1.254 && ftp)
```

2. 와이어샤크 필터 예제

- '5-2.와이어샤크 필터 예제.pcap' 파일을 와이어샤크로 오픈한다.

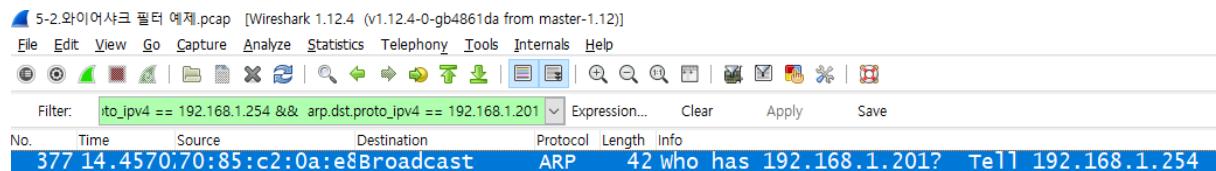
PC	서버
192.168.1.254	192.168.1.201
70:85:C2:0A:E8:F6	

1) '192.168.1.201' ARP 요청 및 응답 내용 필터

```
request(1) or response(2) : arp.opcode == N  
sender mac : arp.src.hw_mac  
sender IP : arp.src.proto_ipv4  
target mac : arp.dst.hw_mac  
target IP : arp.dst.proto_ipv4
```

- ARP 요청 메세지

```
arp.opcode == 1 && arp.src.hw_mac == 70:85:c2:0a:e8:f6 && arp.src.proto_ipv4 ==  
192.168.1.254 && arp.dst.proto_ipv4 == 192.168.1.201
```



- ARP 응답 메세지

```
arp.opcode == 2 && arp.src.proto_ipv4 == 192.168.1.201 && arp.dst.proto_ipv4 ==  
192.168.1.201
```

정답:

```
(arp.opcode == 1 && arp.src.hw_mac == 70:85:c2:0a:e8:f6 && arp.src.proto_ipv4 ==  
192.168.1.254 && arp.dst.proto_ipv4 == 192.168.1.201) or (arp.opcode == 2 &&  
arp.src.proto_ipv4 == 192.168.1.201 && arp.dst.proto_ipv4 ==  
192.168.1.201)
```

2) '192.168.1.201' TCP 3-Way 핸드 쉐이킹 & HTTP(TCP 80) 내용 필터

PC	서버
192.168.1.254	192.168.1.201
70:85:C2:0A:E8:F6	
Syn(0x02) ->	
seq 0	
	<- Syn+Ack(0x12)
	seq 0, ack 1
Ack(0x10) ->	
seq 1, ack 1	

```
ip.src == 192.168.1.254 && ip.dst == 192.168.1.201 && tcp.dstport == 80 && tcp.flags == 0x02 && tcp.seq == 0  
ip.src == 192.168.1.201 && ip.dst == 192.168.1.254 && tcp.srcport == 80 && tcp.flags == 0x12 && tcp.seq == 0 && tcp.ack == 1  
|  
ip.src == 192.168.1.254 && ip.dst == 192.168.1.201 && tcp.dstport == 80 && tcp.flags == 0x10 && tcp.seq == 1 && tcp.ack == 1
```

답:

```
(ip.src == 192.168.1.254 && ip.dst == 192.168.1.201 && tcp.dstport == 80 && tcp.flags == 0x02 && tcp.seq == 0) or (ip.src == 192.168.1.201 && ip.dst == 192.168.1.254 && tcp.srcport == 80 && tcp.flags == 0x12 && tcp.seq == 0 && tcp.ack == 1) or (ip.src == 192.168.1.254 && ip.dst == 192.168.1.201 && tcp.dstport == 80 && tcp.flags == 0x10 && tcp.seq == 1 && tcp.ack == 1)
```

```
== eq  
&& and  
|| or  
! not
```

5) '192.168.1.201' ICMP Echo-Request, ICMP Echo-Reply 내용 필터



요청 8

응답 0

목적지 못간다 3

CODE 3: 목적지까지 도착했는데 포트가 닫혀있다.

■ Internet Control Message Protocol

Type: 8 (Echo (ping) request)

Code: 0

Checksum: 0x4bf5 [correct]

Identifier (BE): 1 (0x0001)

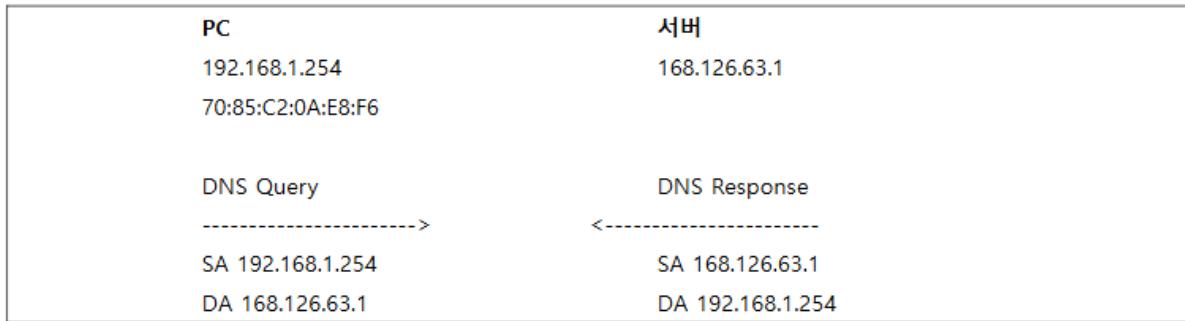
Identifier (LE): 256 (0x0100)

Sequence number (BE): 358 (0x0166)

Sequence number (LE): 26113 (0x6601)

[Response frame: 425]

6) PC 에서 '168.126.63.1'으로 'www.naver.com'에 대한 DNS 요청 및 응답 내용 필터



요청:

```
dnsqry.name  
dnsresp.name
```

```
ip.src == 192.168.1.254 && ip.dst == 168.126.63.1 && dnsqry.name == www.naver.com
```

응답:

```
ip.src == 168.126.63.1 && ip.dst == 192.168.1.254 && dnsresp.name == www.naver.com
```

정답:

```
(ip.src == 192.168.1.254 && ip.dst == 168.126.63.1 && dnsqry.name == www.naver.com) or (ip.src ==  
168.126.63.1 && ip.dst == 192.168.1.254 && dnsresp.name == www.naver.com)
```

제6장 IP 주소 특징

1. IP 주소

- Layer 3 계층 주소
- IP 헤더 안에 포함된 주소
- 주소 크기: 2^{32} 개 = 4,294,967,296 개)
- IP 주소 현황: 2011년 2월 고갈 발표
- IP 주소 고갈 문제 대책: 서브넷 마스크, 서브넷팅, VLSM, 사설 IP 주소 & NAT, IPv6 주소 전환
- IP 주소는 임대 서비스이다. (임대 과정: IANA -> APNIC -> KRNIC(KISA) -> ISP -> 사용자)
- 참고 사이트: <http://www.iana.com>

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

11111111 = 255
11111110 = 254
11111100 = 252
11111000 = 248
11110000 = 240
11100000 = 224
11000000 = 192
10000000 = 128
00000000 = 0
11101010 =

2. 서브넷 마스크(Subnet Mask)

- 목적: IP 주소 고갈 방지 대책, 네트워크 구분 및 IP 주소 개수 계산
- 특징: 맨 앞에 비트부터 '1'이 연속되어야 한다. 만약, '1'이 연속되지 않으면 서브넷 마스크로 인식하지 않는다.
- IP에다가 씌우는 마스크 (32bit, IP와 같음)

32bit	
11111111 11111111 11111111	00000000
공통 비트	비공통 비트
'1' 표기	'0' 표기
네트워크 아이디(네트워크 식별자)	호스트 아이디(호스트 식별자)

서브넷 마스크 예시:

네트워크 아이디 호스트 아이디

홍길동 1 0 0	홍	길동	홍 □ □
홍길동 1 1 0	홍길	동	홍길□
홍길동 1 1 1	홍길동	0	홍길동
□ □ □ 0 0 0	0	3자리	이름 전체

네트워크 아이디 호스트 아이디

121.160.42.2 255.255.255.0	121.160.42	.2	121.160.42.x	2^8
121.160.42.2 255.255.0.0	121.160	.42.2	121.160.x.x	2^16
121.160.42.2 255.0.0.0	121	.160.42.2	121.x.x.x	2^24개
121.160.42.2 255.255.255.255	121.160.42.2	0bit	121.160.42.2	2^0 = 1개
0.0.0.0 0.0.0.0	0bit	32bit	IP 주소 전체	2^32개

ex) 서브넷 마스크를 확인하여 같은 네트워크인지 다른 네트워크인지 구분하여라.

A 와 B 는 같은 네트워크인가?	C 는 A,B 와 같은 네트워크인가?	C 는 A,B 와 같은 네트워크인가?
A 121.160.13.45 255.255.255.0 B 121.160.13.143 255.255.255.0	A 121.160.13.45 255.255.255.0 B 121.160.13.143 255.255.255.0 C 121.160.14.45 255.255.255.0	A 121.160.13.45 255.255.0.0 B 121.160.13.143 255.255.0.0 C 121.160.14.45 255.255.0.0

같다

다르다

같다.

네트워크 아이디는 같아야한다.

그러나 호스트 아이디는 같으면 안된다.

3. IP 주소 클래스

- IP 주소 범위: 0.0.0.0 ~ 255.255.255.255
- 클래스: A, B, C, D, E

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

- 1) 유니캐스트 주소(패킷을 유니캐스트로 전송할 때 사용하는 주소이며, 장치에 설정이 가능하다.)

A Class(0~127) : 맨 앞에 비트가 '0'인 공통 비트 클래스

0.0.0.0 ~ 127.255.255.255 기본 서브넷 마스크 : 255.0.0.0
00000000. 01111111. 네트워크 아이디당 IP 주소 개수 : 2^{24} 개(16,777,216 개)

B Class(128~191) : 맨 앞에 비트가 '10'인 공통 비트 클래스

128.0.0.0 ~ 191.255.255.255 기본 서브넷 마스크 : 255.255.0.0
10000000. 10111111. 네트워크 아이디당 IP 주소 개수 : 2^{16} 개(65,536 개)

C Class(192~223) : 맨 앞에 비트가 '110'인 공통 비트 클래스

192.0.0.0 ~ 223.255.255.255 기본 서브넷 마스크 : 255.255.255.0
11000000. 11011111. 네트워크 아이디당 IP 주소 개수 : 2^8 개(256 개)

- 2) 멀티캐스트 주소(패킷을 멀티캐스트로 전송할 때 사용하는 주소이며, 장치에 설정이 불가능하다.)

D Class(224~239) : 맨 앞에 비트가 '1110'인 공통 비트 클래스

224.0.0.0 ~ 239.255.255.255
11100000 11101111.

- 3) IANA 예비용 예약 주소(IANA에서 예비용으로 예약한 주소이며, 사용 및 장치에 설정이 불가능하다.) - 리눅스는 예외

E Class(240~255)

240.0.0.0 ~ 255.255.255.255
11110000 11111111.

ex) 몇 클래스에 해당?

제목 없음 - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말

A 121.160.43.21 255.255.255.0

A class (0~127)

4. 네트워크 이름 & 서브넷 브로드캐스트 주소

- 네트워크 이름은 IP 주소가 설정된 네트워크의 이름으로 사용하는 주소이다.
- 서브넷 브로드캐스트 주소(Directed Broadcast)는 네트워크 안에서 브로드캐스트 할 때 사용하는 주소이다.
- 장치에 사용할 때 예약되어 있는 주소라 설정이 불가능하다.

```
121.160.41.0    <- 네트워크 이름 : 네트워크 아이디의 호스트 아이디가 전체 '0'인 주소
~  
121.160.41.252  255.255.255.0  
~  
121.160.41.255  <- 서브넷 브로드캐스트 주소 : 네트워크 아이디의 호스트 아이디가 전체 '1'인 주소
```

ex) 178.150.32.52 / 255.255.0.0 의 네트워크 이름과 서브넷 브로드캐스트 주소를 찾아라

178.150.0.0

178.150.32.52 255.255.0.0

178.150.255.255



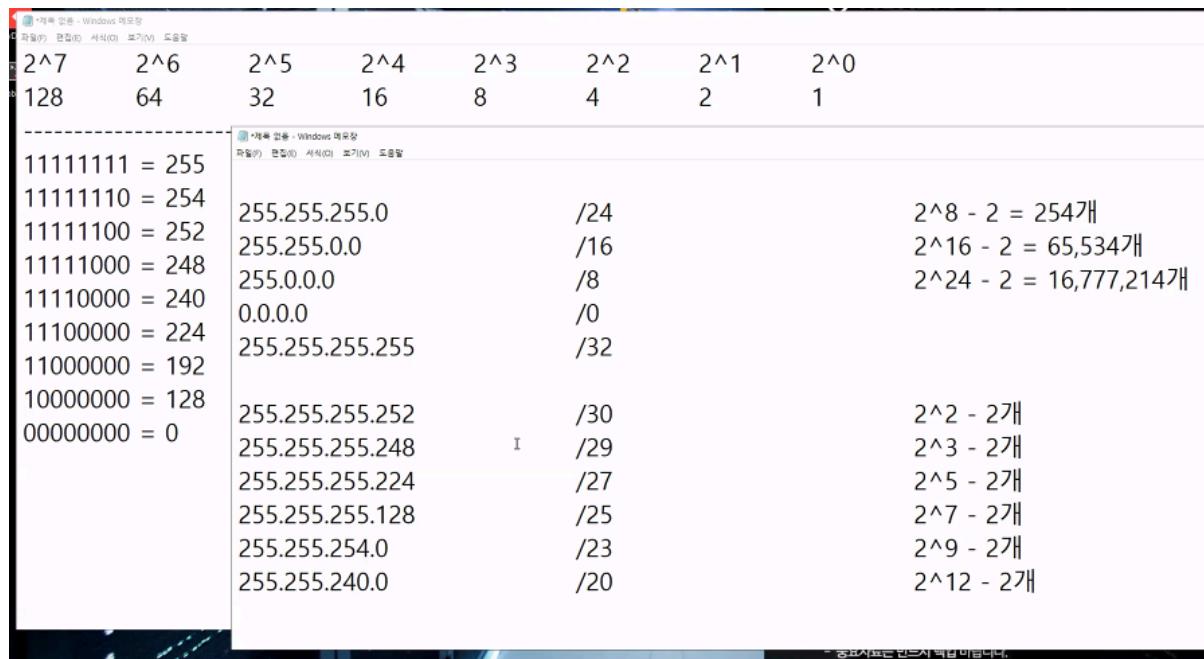
5. 설정이 불가능한 IP 주소

- D 클래스 224.0.0.0 ~ 239.255.255.255 (설정 x 사용은 가능)
- E 클래스 240.0.0.0 ~ 255.255.255.255 (설정 x 사용 x)
- 0.0.0.0 ~ 0.255.255.255 (설정 x 사용은 가능)
- 127.0.0.0 ~ 127.255.255.255 (localhost/Loopback 예약 주소)
- 네트워크 이름과 서브넷 브로드캐스트 주소(Directed Broadcast 주소)

6. 프리픽스 마스크 & 설정 가능한 IP 주소 개수 계산

서브넷 마스크	프리픽스(Prefix) 마스크	설정 가능한 IP 주소 개수 ($2^{\text{host-id}} - 2$ 개)
255.255.255.255	/32	2^0 개
255.255.255.0	/24	$2^8 - 2$ 개
255.255.0.0	/16	$2^{16} - 2$ 개
255.0.0.0	/8	$2^{24} - 2$ 개
0.0.0.0	/0	2^{32} 개
255.255.255.252	/30	$2^2 - 2$ 개
255.255.255.248	/29	$2^3 - 2$ 개
255.255.255.224	/27	$2^5 - 2$ 개
255.255.255.128	/25	$2^7 - 2$ 개
255.255.254.0	/23	$2^9 - 2$ 개
255.255.240.0	/20	$2^{12} - 2$ 개

****보고 하면 쉽다:**



7. 공인 IP 주소 & 사설 IP 주소

1) 공인 IP 주소(Public IP 주소)

- a) ISP 업체에서 할당한 인터넷이 가능한 주소이다.
- b) 돈내고 쓰는거다 (기본요금)

2) 사설 IP 주소(Private IP 주소)

- a) ISP 업체 임대와 관계 없이 내부용으로 사용하는 주소이다.
- b) 사설 IP 네트워크 정보는 ISP 업체 라우터 장비에 경로를 구성하지 않기 때문에 인터넷이 불가능하다.
- c) 사설 IP 주소 범위는 다음과 같다

A Class 10.0.0.0 ~ 10.255.255.255

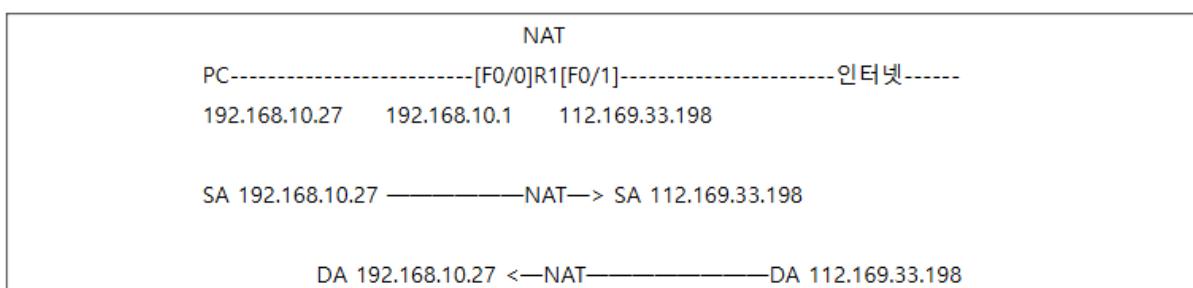
B Class 172.16.0.0 ~ 172.31.255.255

C Class 192.168.0.0 ~ 192.168.255.255

- d) 돈 안내도 쓸 수 있는, 내부적으로 사용할 수 있는 주소다

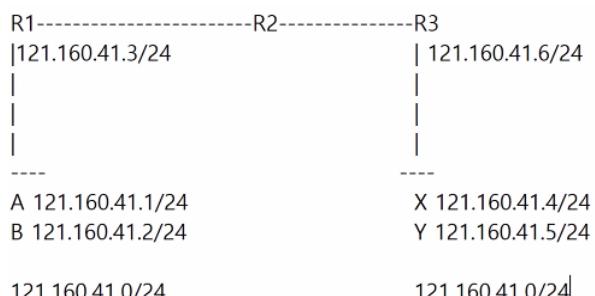
3) NAT 기능

- a) 사설 IP 주소를 사용하는 내부 네트워크 환경에서 인터넷을 하려면 라우터에 NAT 기능을 활용해야 한다
- b) 인터넷으로 리퀘스트를 보내는 것까지는 되는데 돌아오는 건 불가능하다 (ISP에서는 사설아이피에 대한 정보가 없다)
 - i) 따라서 NAT 시스템이 필요하다.

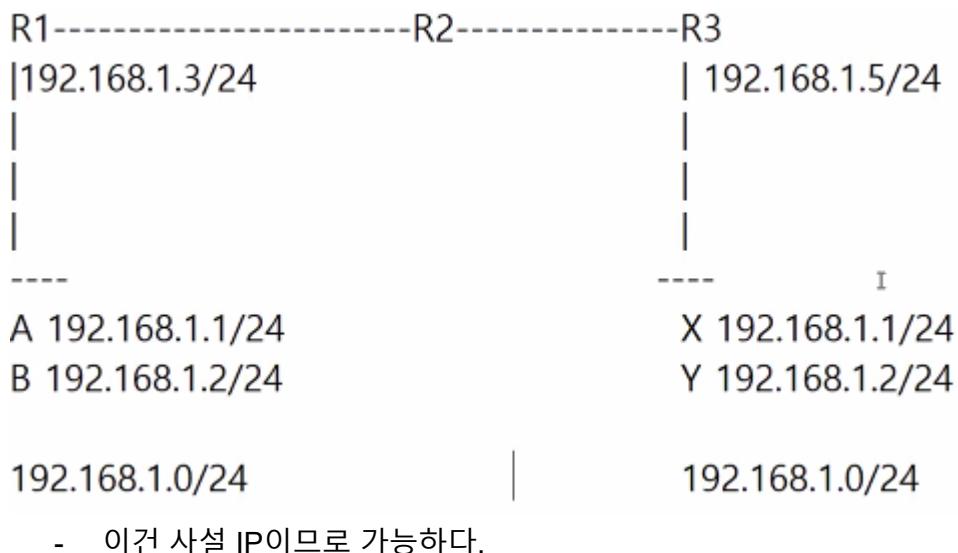


8. IP 주소 사용 주의 사항

- 같은 네트워크 환경에서는 IP 주소가 중복되면 안된다. (호스트 아이디 중복)
- 미할당 IP 주소는 다른 네트워크에 할당하면 안된다. (네트워크 이름 중복)



네트워크 이름이 같아짐으로 내부 네트워크로 인식한다. (응답을 받을 수 없다)



9. IP 주소 예제

Ex1) 19.188.27.251 255.0.0.0

- 1) 무슨 클래스?
- 2) 네트워크 이름과 서브넷 브로드캐스트 주소?
- 3) 설정 가능한 IP 주소 범위와 개수?
- 4) 서브넷 마스크를 Prefix로 표기하면 어떻게 되는가?
- 5) 사설/공인?
- 6) 후이즈 검색 결과 :

- 1) A class
- 2) 19.0.0.0, 19.255.255.255
- 3) 19.0.0.1 ~ 19.255.255.254 = $2^{24} - 2$
- 4) /8
- 5) 공인
- 6) US, 포드자동차

Ex2) 123.255.181.17 255.255.0.0

- 1) A class
- 2) 123.255.0.0, 123.255.255.255
- 3) 123.255.0.1 ~ 123.255.255.254 = $2^{16} - 2$
- 4) /16
- 5) 공인
- 6) Japan

Ex3) 172.16.255.254 255.255.0.0

- B class
- 172.16.0.0, 172.16.255.255
- $172.16.0.1 \sim 172.16.255.254 = 2^{16} - 2$ 개
- /16
- 사설
- 사설 아이피 범위

Ex4) 172.30.1.4 255.255.255.0

- B class
- 172.30.1.0, 172.30.1.255
- $172.30.1.1 \sim 172.30.1.254 = 2^8 - 2$ 개
- /24
- 사설
- 사설 아이피 범위

Ex5) 192.168.133.87 255.255.255.0

- C class
- 192.168.133.0, 192.168.133.255
- $192.168.133.1 \sim 192.168.133.254 = 2^8 - 2$ 개
- /24
- 사설
- 사설 아이피 범위

Ex6) 172.16.1.100 255.255.255.0

- B class
- 172.16.1.0, 172.16.1.255
- $172.16.1.1 \sim 172.16.1.254 = 2^8 - 2$ 개
- /24
- 사설
- 사설 아이피 범위

Ex7) 211.241.228.14 255.255.255.0

- C class
- 211.241.228.0, 211.241.228.255
- $211.241.228.1 \sim 211.241.228.254 = 2^8 - 2$ 개
- /24
- 공인
- 하이라인닷넷, 서울, 강남

Ex8) 10.211.10.7 255.255.255.0

- A class
- 10.211.10.0, 10.211.10.255
- $10.211.10.1 \sim 10.211.10.254 = 2^8 - 2$ 개
- /24
- 사설
- 사설 아이피 범위

Ex9) 다음과 같은 경우, 어떤 클래스 및 어떤 서브넷 마스크를 사용하는 것이 효율적(IP 주소 낭비 방지)인가?

1) Host 230개 :	A, B, C	/24	$2^8 - 2 = 254\text{개}$
2) Host 50,000개 :	A, B	/16	$2^{16} - 2 = 65,544\text{개}$
3) Host 10,000,000개 :	A	/8	$2^{24} - 2 = 16,777,214\text{개}$
4) Host 25개 :	A, B, C	/27	$2^5 - 2 = 30\text{개}$
5) Host 1000개 :	A, B	/22	$2^{10} - 2 = 1022\text{개}$

Ex10) 서브넷 마스크가 아닌 것은?

- ① **255.255.241.0** # 1의 연속이 안됨
- ② 255.255.248.0
- ③ 255.255.255.252
- ④ 0.0.0.0

11111111 = 255
11111110 = 254
11111100 = 252
11111000 = 248
11110000 = 240
11100000 = 224
11000000 = 192
10000000 = 128
00000000 = 0

IP 전체를 의미:

0.0.0.0/0

제7장 서브넷팅 & 주소 요약

1. 서브넷팅(Subnetting)

- 서브넷팅 목적: IP 주소 낭비 방지
- 서브넷팅 방법: 원본 네트워크를 여러 개의 네트워크로 분리하는 계산 작업

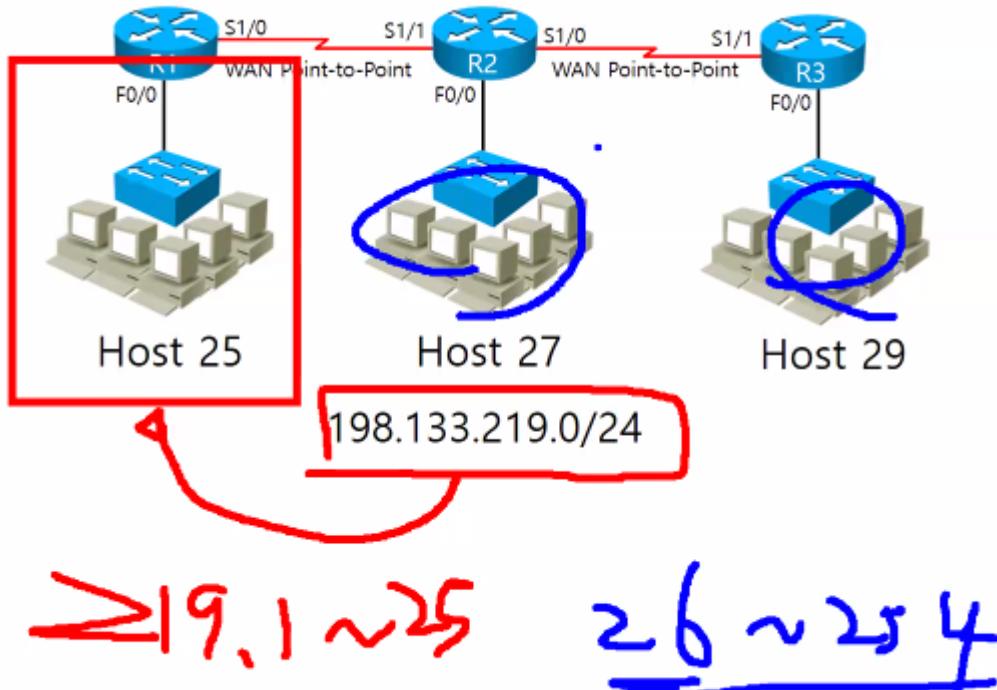
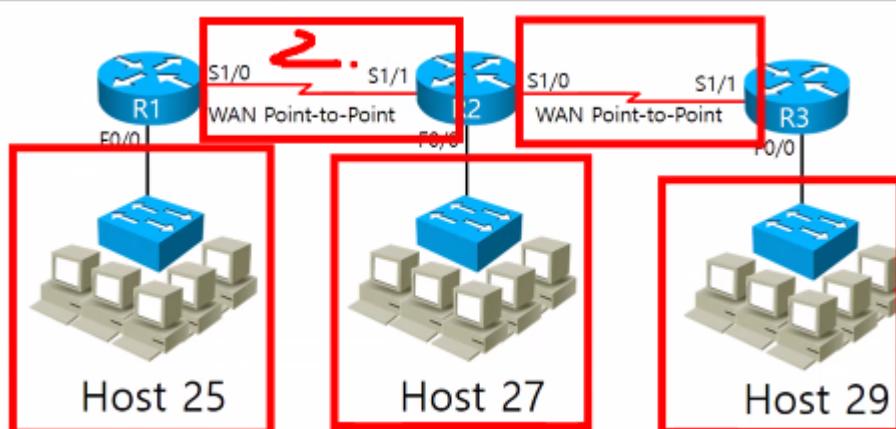


그림 8-1

1~25까지의 아이피를 host25에 할당하면 26~254는 다른데다가 할당을 못함으로 버려진다.
따라서 서브넷팅(Subnetting)이 필요하다.



이 경우 5군데로 네트워크를 나눠줘야 한다 (WAN 2개씩 포함)

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

@ 서브넷팅

- 원본 네트워크: 198.133.219.0 / 24 $\leftarrow 2^8 - 2 = 254\text{개}$
- 서브넷 개수: 5개 이상
- 최대 필요한 IP 개수: 29개

$2^{\text{host id}} - 2 \geq 29$

host id = 5

$2^5 - 2 = 30$

----- 네트워크 이름	서브넷 브로드캐스트 주소
198.133.219.000 00000 <- 198.133.219.0 ~ 198.133.219.31	
198.133.219.001 00000 <- 198.133.219.32 ~ 198.133.219.63	
198.133.219.010 00000 <- 198.133.219.64 ~ 198.133.219.95	
198.133.219.011 00000 <- 198.133.219.96 ~ 198.133.219.127	
198.133.219.100 00000 <- 198.133.219.128 ~ 198.133.219.159	
198.133.219.101 00000 <- 198.133.219.160 ~ 198.133.219.191	
198.133.219.110 00000 <- 198.133.219.192 ~ 198.133.219.223	
198.133.219.111 00000 <- 198.133.219.224 ~ 198.133.219.255	
<hr/>	
198.133.219.1 ~ 198.133.219.30 <- 30개 <- 198.133.219.0/27	
198.133.219.33 ~ 198.133.219.62 <- 30개 <- 198.133.219.32/27	
198.133.219.65 ~ 198.133.219.94 <- 30개 <- 198.133.219.64/27	
198.133.219.97 ~ 198.133.219.126 <- 30개 <- 198.133.219.96/27	
198.133.219.129 ~ 198.133.219.158 <- 30개 <- 198.133.219.128/27	
198.133.219.161 ~ 198.133.219.190 <- 30개 <- 198.133.219.160/27	
198.133.219.193 ~ 198.133.219.222 <- 30개 <- 198.133.219.192/27	
198.133.219.225 ~ 198.133.219.254 <- 30개 <- 198.133.219.224/27	

서브넷 계산기:

<https://www.site24x7.com/tools/ipv4-subnetcalculator.html>

<https://www.solarwinds.com/free-tools/advanced-subnet-calculator>

Ex3) 181.160.85.225/28

- 네트워크 이름? 181.160.85.224/28
- 서브넷 브로드캐스트 주소? 181.160.85.[†]239|

255.255.255.11110000

224 ~ 239

240

Ex4) 192.168.1.133/30

- 네트워크 이름? 192.168.1.132/30
- 서브넷 브로드캐스트 주소? 192.168.1.135

255.255.255.11111100

Ex5) 121.160.30.17/30

- 네트워크 이름? 121.160.30.16/30
- 서브넷 브로드캐스트 주소? 121.160.30.19|

Ex6) 211.240.56.188/26

- 네트워크 이름? 211.240.56.128/26
- 서브넷 브로드캐스트 주소? 211.240.56.191

255.255.255.11000000

0

I

64

128 ~ 191

192

Ex7) 각각의 네트워크 이름에 포함되는 설정 가능한 IP 주소 범위를 구하여라.

- 182.167.211.0/27 1~30

255.255.255.11100000

2)

- 121.160.32.128/27

128 ~ 159 (129~158)

160

3)

- 17.160.32.64/30 65~66

255.255.255.11111100

64 ~ 67 (65~66)

68

4)

255.255.255.11110000

64 ~ ^I79(65~78)

80

5)

- 61.42.100.0/25

255.255.255.10000000

0 ~ 127 (1~126)

128

6)

- 61.42.100.128/25 129~254

7)

- 132.21.128.0/23

255.255.254.0

255.255.11111110.0

132.21.128.0 ~ 132.21.129.255

(132.21.128.1 ~ 132.21.129.254)

132.21.130.0

132.21.132.0

132.21.134.0

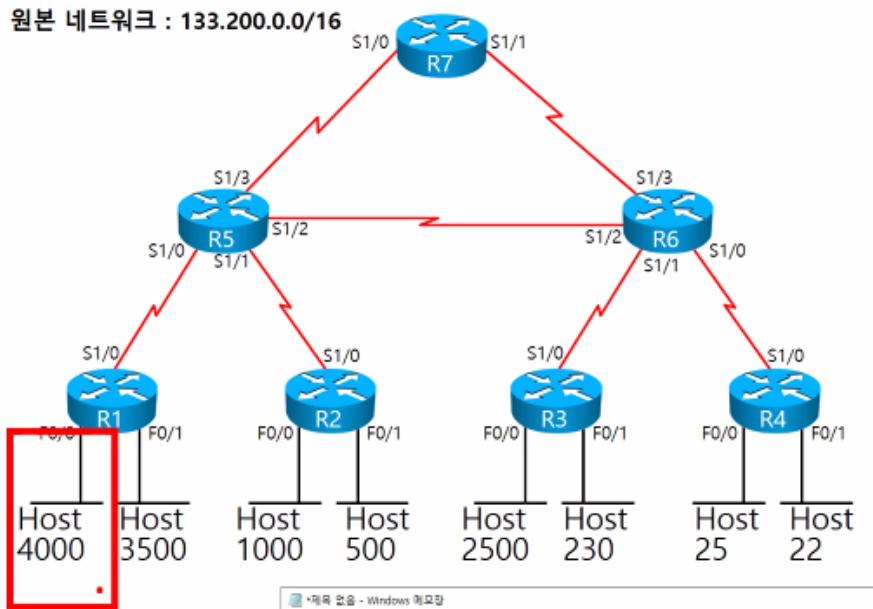
2. VLSM(Variable Length Subnet Mask)

- 서브넷팅을 실시한 서브넷의 서브넷 마스크를 더 조정하여 IP 주소 낭비를 최소화하는 기능이다.
- 원본 네트워크: $133.200.0.0/16 \leftarrow 2^{16} - 2 = 65534$ 개
- 서브넷 개수: 15 개 이상
- 최대 필요한 IP 주소 개수: 4000 개

$$2^x - 2 \geq 4000$$

$$x = 12$$

$$2^{12} - 2 = 4094\text{개}$$



최대 4000개가 필요하다.

$$2^x - 2 \geq 4000$$

$$x = 12$$

$$2^{12} - 2 = 4094\text{개}$$

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

133.200.0000 0000.00000000

255.255.1111 0000.00000000 ← 255.255.240.0 ← /20

133.200.0000 0000.00000000 ← 133.200.0.0 ~ 133.200.15.255 ← Host 4000개 구간 할당

133.200.0001 0000.00000000 ← 133.200.16.0 ~ 133.200.31.255 ← Host 3500개 구간 할당

133.200.0010 0000.00000000 ← 133.200.32.0 ~ 133.200.47.255 ← Host 2500개 구간 할당

133.200.0011 0000.00000000 ← 133.200.48.0 ~ 133.200.63.255 ← VLSM (서브넷마스크의 길이를 더 조정할 수 있다는 기능)

133.200.0100 0000.00000000 ← 133.200.64.0 ~ 133.200.79.255

~

133.200.1111 0000.00000000 ← 133.200.240.0 ~ 133.200.255.255

1. Host 1000개 구간 VLSM

133.200.0011 0000.00000000 ← 133.200.48.0 ~ 133.200.63.255 ← VLSM

원본 네트워크 : 133.200.48.0/20 ← $2^{12} - 2$ 개 4094개

$$2^x - 2 \geq 1000$$

$$x = 10$$

$$2^{10} - 2 = 1022\text{개}$$

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

133.200.0011 00 00.00000000

255.255.1111 11 00.00000000 ← 255.255.252.0 ← /22

----- 네트워크 id 호스트 주소

133.200.0011 00 00.00000000 ← 133.200.48.0 ~ 133.200.51.255 ← Host 1000개 구간 할당

133.200.0011 01 00.00000000 ← 133.200.52.0 ~ 133.200.55.255 ← VLSM

133.200.0011 10 00.00000000 ← 133.200.56.0 ~ 133.200.59.255

133.200.0011 11 00.00000000 ← 133.200.60.0 ~ 133.200.63.255

2. Host 500개 구간 VLSM

133.200.0011 01 00.00000000 ← 133.200.52.0 ~ 133.200.55.25 ← VLSM

원본 네트워크: 133.200.52.0/22 ← $2^{10} - 2 = 1022$ 개

$2^x - 2 \geq 500$

$x = 9$

$2^9 - 2 = 510$ 개

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

133.200.001101 0 0.00000000

255.215.111111 1 0.00000000 ← 255.255.254.0 ← /23

133.200.001101 0 0.00000000 ← 133.200.52.0 ~ 133.200.53.255 ← Host 500개 구간 할당

133.200.001101 1 0.00000000 ← 133.200.54.0 ~ 133.500.55.255 ← VLSM

3. Host 250개 구간 VLSM

133.200.001101 1 0.00000000 ← 133.200.54.0 ~ 133.500.55.255 ← VLSM

원본 네트워크: 133.200.54.0/23 ← $2^9 - 2 = 510$ 개

$2^x - 2 \geq 2300$

$x = 8$

$2^8 - 2 = 254$ 개

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

133.200.0011011 0. 00000000

133.200.1111111 1. 00000000 ← 255.255.255.0 ← /24

133.200.0011011 0. 00000000 ← 133.200.54.0 ~ 133.200.54.255 ← Host 230개 구간 할당

133.200.0011011 1. 00000000 ← 133.200.55.0 ~ 133.200.55.255 ← VLSM

4. Host 25, 22개 구간 VLSM

133.200.0011011 1. 00000000 ← 133.200.55.0 ~ 133.200.55.255 ← VLSM
원본 네트워크 : 133.200.55.0 / 24 ← $2^8 - 2 = 254$ 개

$$2^x - 2 \geq 25$$

$$x = 5$$

$$2^{5-2} = 304$$
 개

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

0133.200.55.000 00000

255.255.255.111 00000 ← 255.255.255.224 ← /27

0133.200.55.000 00000 <- 133.200.55.0 ~ 133.200.55.33 <- Host 25개 구간 할당

0133.200.55.001 00000 <- 133.200.55.32 ~ 133.200.55.63 <- Host 22개 구간 할당

0133.200.55.010 00000 <- 133.200.55.64 ~ 133.200.55.97 <- VLSM

0133.200.55.011 00000 <- 133.200.55.96 ~ 133.200.55.129

0133.200.55.100 00000 <- 133.200.55.128 ~ 133.200.55.161

0133.200.55.101 00000 <- 133.200.55.160 ~ 133.200.55.191

0133.200.55.110 00000 <- 133.200.55.192 ~ 133.200.55.225

0133.200.55.111 00000 <- 133.200.55.224 ~ 133.200.55.255

5. WAN P2P 구간 VLSM

$$2^x - 2 \geq 2$$

$$x = 2$$

$$2^2 - 2 = 2\text{개}$$

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

0133.200.55.010 000 00

255.255.255.111 111 00 <- 255.255.255.252 <- /30

0133.200.55.010 000 00 <- 133.200.55.64 ~ 133.200.55.67

0133.200.55.010 001 00 <- 133.200.55.68 ~ 133.200.55.71

0133.200.55.010 010 00 <- 133.200.55.72 ~ 133.200.55.75

0133.200.55.010 011 00 <- 133.200.55.76 ~ 133.200.55.79

0133.200.55.010 100 00 <- 133.200.55.80 ~ 133.200.55.83

0133.200.55.010 101 00 <- 133.200.55.84 ~ 133.200.55.87

0133.200.55.010 110 00 <- 133.200.55.88 ~ 133.200.55.91

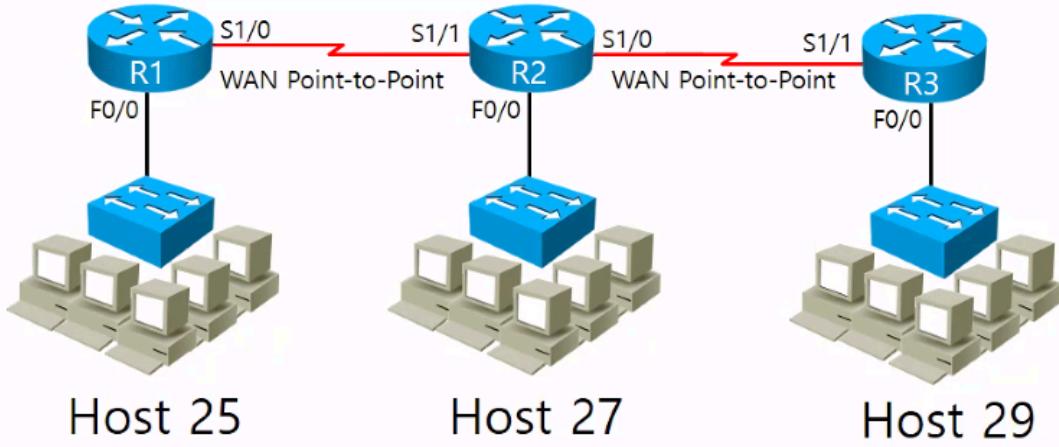
0133.200.55.010 111 00 <- 133.200.55.92 ~ 133.200.55.95 |

(WAN P2P 구간 할당 8개)

4094개 서브넷	/20	12개
1022개 서브넷	/22	2개
510개 서브넷	/23	0개
254개 서브넷	/24	0개
30개 서브넷	/27	6개
2개 서브넷	/30	1개

서브넷팅 / VLSM / 주소 요약

서브넷팅, VLSM 추가 예제 1)



198.133.219.0/24

- 원본 네트워크 : 198.133.219.0/24 $\leftarrow 2^8 - 2 = 254\text{개}$
- 서브넷 개수 5개 이상
- 최대 필요한 IP 주소 개수: 29개

네트워크 이름

192.168.1.0/24

$2^x - 2 \geq 29$

$x = 5$

subnet

$2^5 - 2 = 30$

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

198.133.219.000 00000

255.255.255.111 00000 $\leftarrow 255.255.255.224 \leftarrow - /27$

----- [네트워크 이름] [서브넷 브로드캐스트 주소]

198.133.219.000 00000 \leftarrow 198.133.219.0 ~ 198.133.219.31 \leftarrow host 25개 구간 할당

198.133.219.001 00000 \leftarrow 198.133.219.32 ~ 198.133.219.63 \leftarrow host 27개 구간 할당

198.133.219.010 00000 \leftarrow 198.133.219.64 ~ 198.133.219.95 \leftarrow host 29개 구간 할당

198.133.219.011 00000 \leftarrow 198.133.219.96 ~ 198.133.219.127 \leftarrow VLSM 실시

198.133.219.100 00000 \leftarrow 198.133.219.128 ~ 198.133.219.159

198.133.219.101 00000 \leftarrow 198.133.219.160 ~ 198.133.219.191

198.133.219.110 00000 \leftarrow 198.133.219.192 ~ 198.133.219.223

198.133.219.111 00000 ← 198.133.219.224 ~ 198.133.219.255

-WAN P2P 구간 VLSM

198.133.219.011 00000 ← 198.133.219.96 ~ 198.133.219.127 ← VLSM 실시

원본 네트워크 : 198.133.219.96/27 ← $2^5 - 2 = 30$ 개

$$2^x - 2 \geq 2$$

$$x = 2$$

$$2^2 - 2 = 2 \text{ 개}$$

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

198.133.219.011 000 00

255.255.255.111 111 00 <- 255.255.255.252 <- /30

----- [네트워크 이름] [서브넷 브로드캐스트 주소]

198.133.219.011 000 00 <- 198.133.219.96 ~ 198.133.219.99 <- WAN P2P 구간 할당

198.133.219.011 001 00 <- 198.133.219.100 ~ 198.133.219.103 <- WAN P2P 구간 할당

198.133.219.011 010 00 <- 198.133.219.104 ~ 198.133.219.107

198.133.219.011 011 00 <- 198.133.219.108 ~ 198.133.219.111

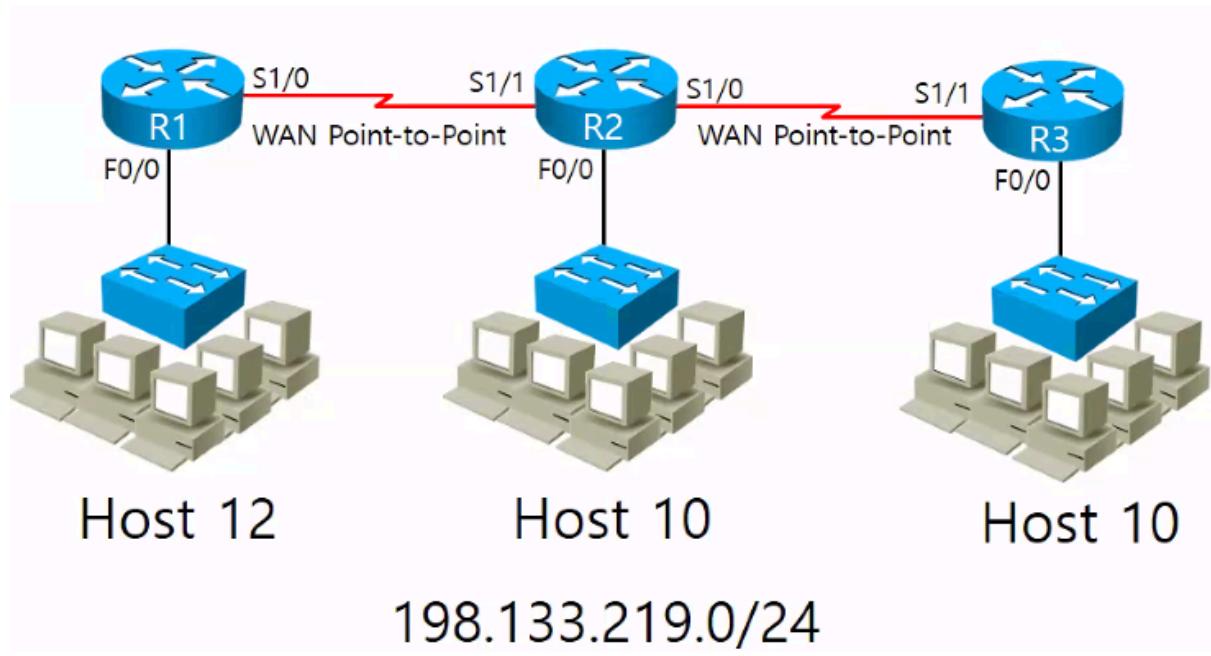
198.133.219.011 100 00 <- 198.133.219.112 ~ 198.133.219.115

198.133.219.011 101 00 <- 198.133.219.116 ~ 198.133.219.119

198.133.219.011 110 00 <- 198.133.219.120 ~ 198.133.219.123

198.133.219.011 111 00 <- 198.133.219.124 ~ 198.133.219.127

서브넷팅, VLSM 추가 예제 2)



- 원본 네트워크 : $198.133.219.0/24 \leftarrow 2^8 - 2 = 254\text{개}$
- 서브넷 개수 5개 이상
- 최대 필요한 IP 주소 개수: 12개

네트워크 이름

192.168.1.0/24

$$2^x - 2 \geq 12$$

$$x = 4$$

$$2^4 - 2 = 14$$

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

198.133.219.0000 0000

255.255.255.1111 0000 \leftarrow 255.255.255.240 \leftarrow /28

----- [네트워크 이름] [서브넷 브로드캐스트 주소]

198.133.219.0000 0000 \leftarrow 198.133.219.0 ~ 198.133.219.15 \leftarrow host 12개 구간 할당

198.133.219.0001 0000 \leftarrow 198.133.219.16 ~ 198.133.219.31 \leftarrow host 10개 구간 할당

198.133.219.0010 0000 \leftarrow 198.133.219.32 ~ 198.133.219.47 \leftarrow host 10개 구간 할당

198.133.219.0011 0000 \leftarrow 198.133.219.48 ~ 198.133.219.63 \leftarrow VLSM 실시

198.133.219.0100 0000 \leftarrow 198.133.219.64 ~ 198.133.219.79

~

198.133.219.1111 0000 \leftarrow 198.133.219.240 ~ 198.133.219.255

-WAN P2P 구간 VLSM

198.133.219.011 0000 ← 198.133.219.48 ~ 198.133.219.63 ← VLSM 실시
원본 네트워크 : 198.133.219.48/28 ← $2^4 - 2 = 14$ 개

$$2^x - 2 \geq 2$$

$$x = 2$$

$$2^2 - 2 = 2$$
 개

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

198.133.219.0011 00 00

198.133.219.0011 01 00 ← 255.255.255.252 ← /30

198.133.219.0011 00 00 ← 198.133.219.48 ~ 198.133.219.51 ← 할당

198.133.219.0011 01 00 ← 198.133.219.52 ~ 198.133.219.55 ← 할당

198.133.219.0011 10 00 ← 198.133.219.56 ~ 198.133.219.59

198.133.219.0011 00 00 ← 198.133.219.60 ~ 198.133.219.63

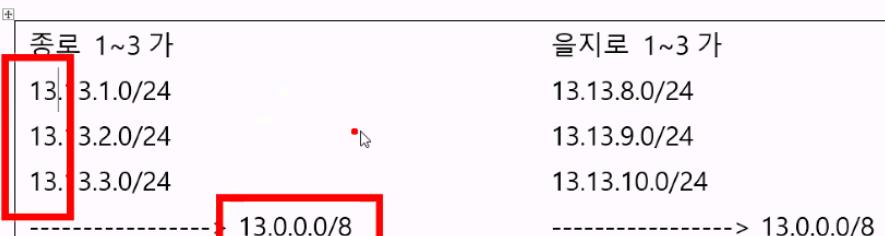
3. 주소 요약 (CIDR)

- 서브넷팅 및 VLSM 을 실시한 IP 대역들을 효율적으로 관리하기 위해서 주소 요약이 필요하다. 또한, 라우터와 라우터 간에 라우팅 업데이트를 진행 할 때 경로 정보를 최소화하기 위해서 주소 요약을 필요하다.

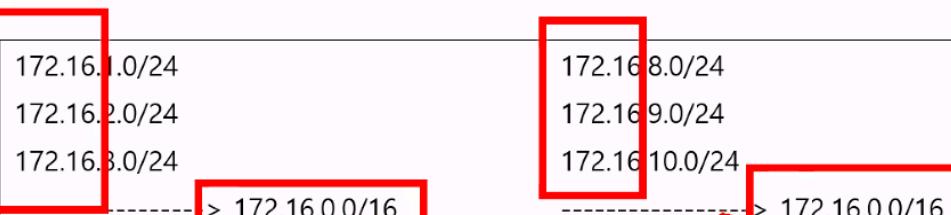
1) 클래스풀 요약

- a) 클래시 기본 서브넷 마스크를 기준으로 요약하는 방법이며 권장하지 않는다.

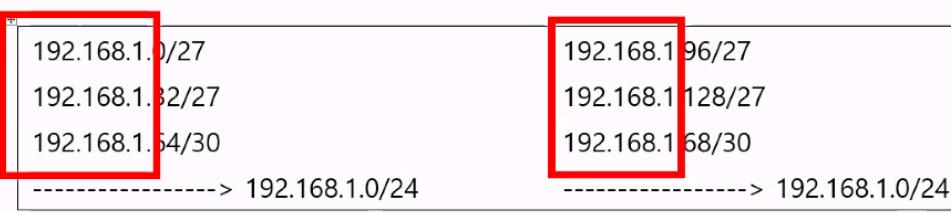
Ex1) A 클래스(255.0.0.0 <- /8)



Ex2) B 클래스(255.255.0.0 <- /16)



Ex3) C 클래스(255.255.255.0 <- /24)



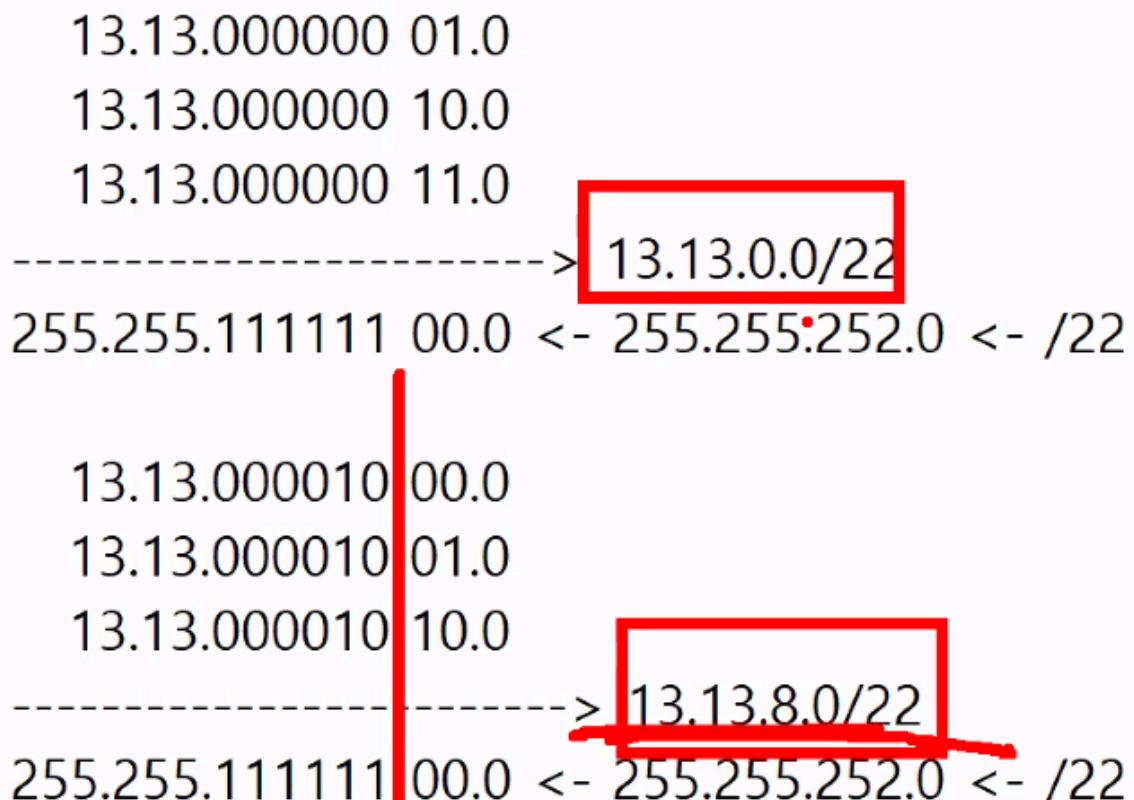
2) 상세 요약

a) 상세 요약을 해도 아직 좋지 않다. (네트워크 이름 중복)

ex1)

13.13.1.0/24	13.13.8.0/24
13.13.2.0/24	13.13.9.0/24
13.13.3.0/24	13.13.10.0/24
-----> 13.0.0.0/8	-----> 13.0.0.0/8
-----I-----> 13.13.0.0/16	-----> 13.13.0.0/16

b) 이정도로 상세요약을 할 수 있다.:



ex2)

128.28.32.0/24 ~ 128.28.63.0/24

128.28.001 00000.0

128.28.001 00001.0

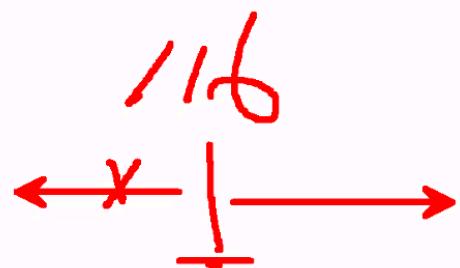
128.28.001 00010.0

~

128.28.001 11111.0

-----> 128.28.32.0/19

255.255.111 00000.0 <- 255.255.224.0 <- /19



ex3)

123.140.0.0 ~ 123.143.255.255

123.100011 00.0.0

123.100011 01.0.0

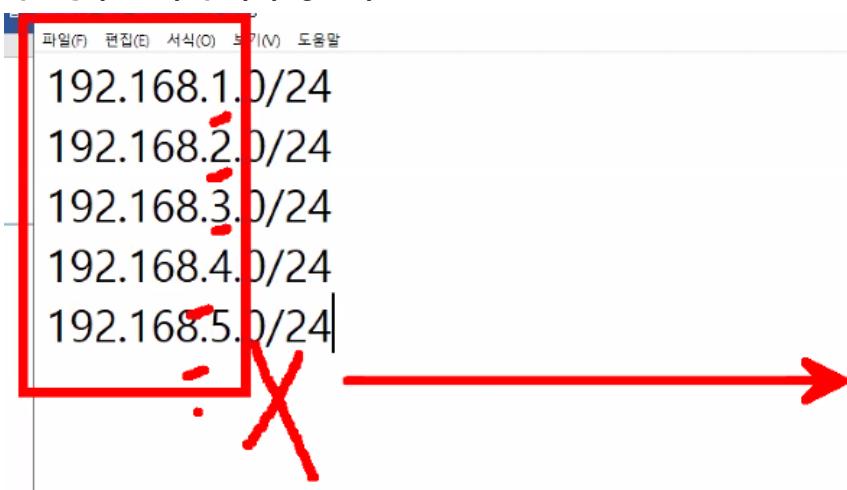
123.100011 10.0.0

123.100011 11.0.0

-----> 123.140.0.0/14

255.111111 00.0.0 <- 255.252.0.0 <- /14

이런경우 요약이 되지 않는다:



ex4)

앞으로 요약은 가능하다:

192.168.1.0/24
192.168.2.0/24
192.168.3.0/24
192.168.4.0/24
192.168.5.0/24

192.168.00000 001.0
192.168.00000 010.0
192.168.00000 011.0
192.168.00000 100.0
192.168.00000 101.0

-----> 192.168.0.0/21

255.255.11111 000.0 <- 255.255.248.0 <- /21



/24

←
/21. x

192.168.160.0/24
192.168.161.0/24
192.168.162.0/24
192.168.163.0/24

192.168.101000 00.0
192.168.101000 01.0
192.168.101000 10.0
192.168.101000 11.0

-----> 192.168.160.0/22

255.255.111111 00.0 <- 255.255.252.0 <- /22

주소 요약 예제:

Ex1) 61.42.0.0 ~ 61.42.63.0 를 상세 요약하여라.

61.42.00 000001.0

~

61.42.00 111111.0

-----> 61.42.0.0/18

255.255.11 000000.0 <- 255.255.192.0 <- /18

Ex2) 121.160.32.0 ~ 121.160.63.0 를 상세 요약하여라.

121.160.001 00000.0

121.160.001 00001.0

~

121.160.001 11111.0

-----> 121.160.32.0/19

255.255.111 00000.0 <- 255.255.224.0 <- /19

I

Ex3) B 클래스 사설 IP 주소(172.16.0.0 ~ 172.31.255.255)를 상세 요약하여라.

172.0001 0000.0.0

172.0001 0001.0.0

~

172.0001 1111.0.0

-----> 172.16.0.0/12

255.1111 0000.0.0 <- 255.240.0.0 <- /12

Ex4) 13.13.0.0 ~ 13.13.63.0 를 상세 요약하여라.

13 / 2	6	1
6 / 2	3	0
3 / 2	1	1
1 / 2	0	1

13.13.00 00000.0

~

13.13.00 111111.0

-----> 13.13.0.0/18
255.255.11 000000.0 ← 255.255.192.0 ← /18

Ex5) 13.13.64.0 ~ 13.13.127.0 를 상세 요약하여라.

13.13.01 000000.0

13.13.01 000001.0

~

13.13.01 111111.0

-----> 13.13.64.0/18
255.255.11 000000.0 <- 255.255.192.0 <- /18

Ex6) 13.13.128.0 ~ 13.13.191.0 를 상세 요약하여라.

13.13.10 000000.0

13.13.10 000001.0

~

13.13.10 111111.0

-----> 13.13.128.0/18
255.255.11 000000.0 <- 255.255.192.0 <- /18

Ex7) 13.13.192.0 ~ 13.13.255.0 를 상세 요약하여라.

13.13.11 000000.0

13.13.11 000001.0

~

13.13.11 1111111.0

-----> 13.13.192.0/18
255.255.11 000000.0 <- 255.255.192.0 <- /18

Ex8) A 클래스 IP 주소를 상세 요약하여라.

0.0.0.0 ~ 127.255.255.255
0 0000000. 0 1111111.

0.0.0.0/1

Ex9) B 클래스 IP 주소를 상세 요약하여라.

128.0.0.0 ~ 191.255.255.255
10 000000. 10 111111.

128.0.0.0/2

Ex10) C 클래스 IP 주소를 상세 요약하여라.

192.0.0.0 ~ 223.255.255.255
110 00000. 110 11111.

192.0.0.0/3

Ex11) 61.40.0.0 ~ 61.43.255.255 를 상세 요약 하여라.

61.001010 00.0.0
61.001010 01.0.0
61.001010 10.0.0
61.001010 11.0.0
-----> 61.40.0.0/14
255.111111 00.0.0 <- 255.252.0.0 <- /14

Ex12) 요약 IP 정보인 '121.160.0.0/13'에 포함되는 IP 주소 범위는 어떻게 되는가?

121.10100 000.0.0	121.160.0.0
~	
121.10100 111.255.255	121.167.255.255

Ex13) 요약 IP 정보인 '211.241.128.0/17' 에 포함되는 IP 주소 범위는 어떻게 되는가?

211.241.1 0000000.0	211.241.128.0
~	
211.241.1 1111111.255	211.241.255.255

Ex14) 요약 IP 정보인 '10.233.0.0/18'에 포함되는 IP 주소 범위는 어떻게 되는가?

10.233.00 000000.0	10.233.0.0
~	
10.233.00 111111.255	10.233.63.255

Ex15) 요약 IP 정보인 '10.233.64.0/18'에 포함되는 IP 주소 범위는 어떻게 되는가?

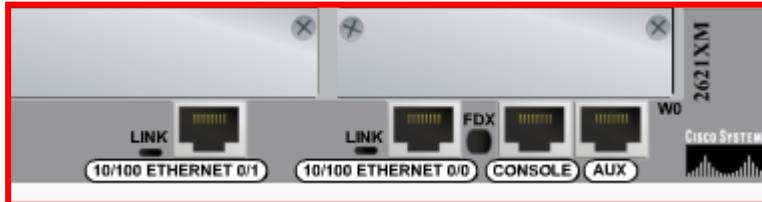
10.233.01 000000.0	10.233.64.0
~	
10.233.01 111111.255	10.233.127.255

제 2부 IP 라우팅

제1장 Cisco IOS 명령어

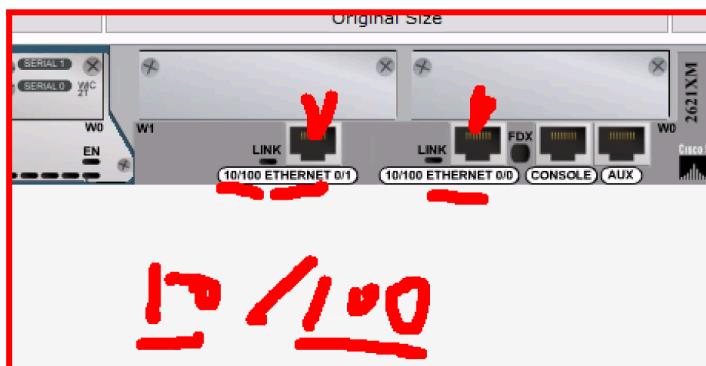
console port:

- 라우터를 다이렉트로 접속할 수 있는 포트
- PC, 노트북 등을 연결해 설정, 관리를 할 수 있는 포트



Ethernet Port

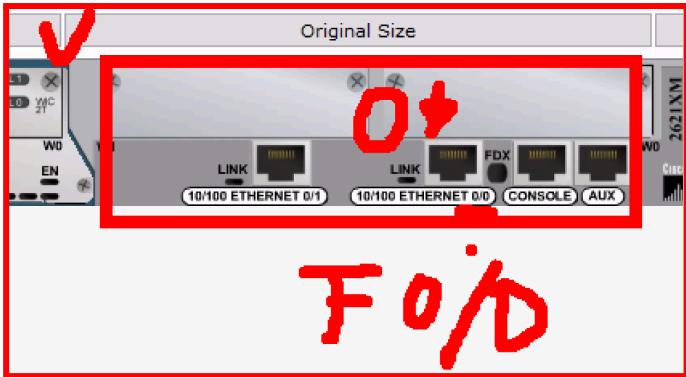
- 이더넷용 포트
- 100메가까지 지원되는 포트 (fastethernet)



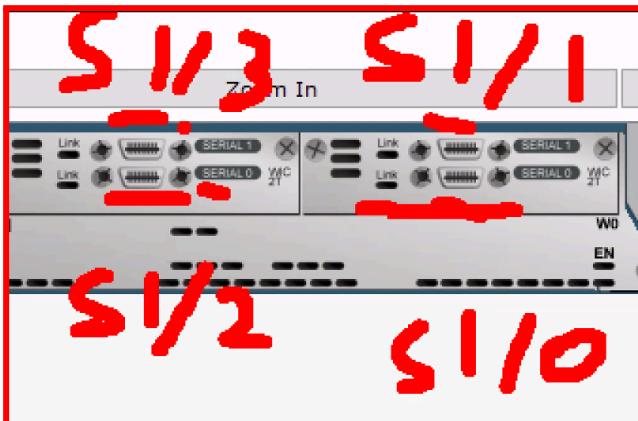
- 대역폭 차이만나고 다 이더넷용 포트이다.

파일(F) 편집(E) 서식(O) 보기(V) 도움말	
etheren	10M
fastethernet	100M
Gigabitethernet	1000M
10Gigabitethernet	10000M

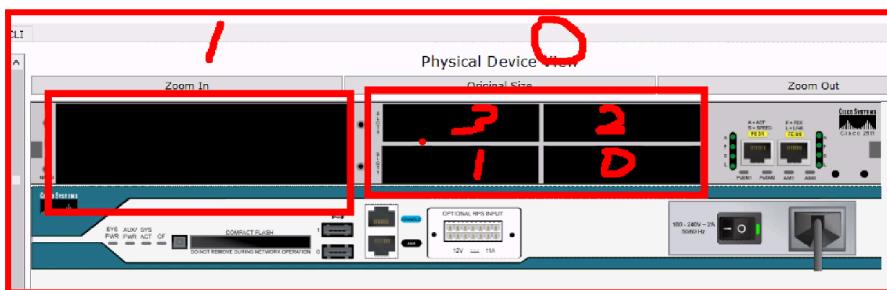
- 0번 슬롯에 1번, 0번포트이다 : F O/O (FASTETHERNET 0번슬롯 0번포트)



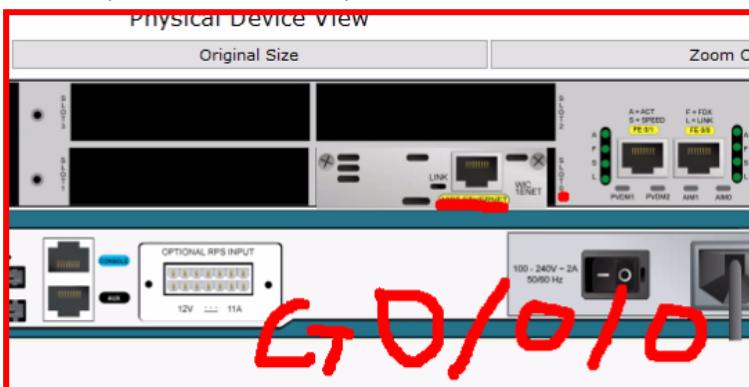
- 1번슬롯에 0,1,2,3 포트



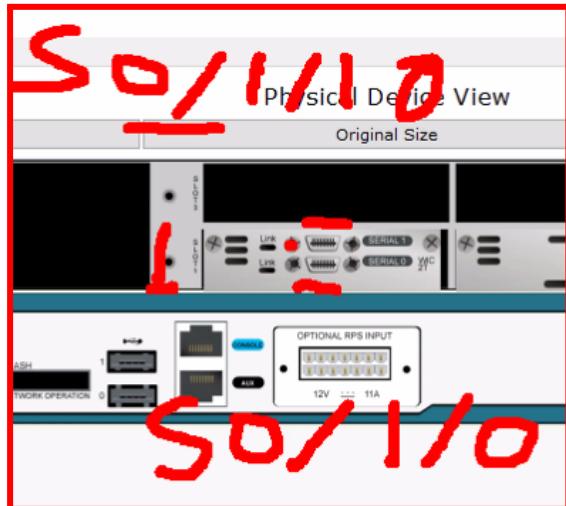
- #모듈, #슬롯, #포트



- G 0/0/0 (GIGAETHERNET)



-S 0 / 1 / 0, S 0 / 1 / 1 (Serial)



명령어 실습

R1 (라우터) → CLI 접속

라우터 부팅:

```
Copyright (c) 2000 by cisco Systems, Inc.
Initializing memory for ECC
...
c2811 processor with 524288 Kbytes of main memory
Main memory is configured to 64 bit mode with ECC enabled

 Readonly ROMMON initialized

 Self decompressing the image :
#####
[OK]

 Restricted Rights Legend

Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706
```

라우터의 시리얼 번호:

```
Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-I-M), Version 12.2(28), RELEASE SOFTWARE (fc5)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2005 by cisco Systems, Inc.
Compiled Wed 27-Apr-04 19:01 by miwang

Cisco 2621 (MPC860) processor (revision 0x200) with 253952K/8192K bytes of memory
.
Processor board ID JAD05190MTZ (4292891495)
M860 processor: part number 0, mask 49
Bridging software.
X.25 software, Version 3.0.0.
2 FastEthernet/IEEE 802.3 interface(s)
4 Low-speed serial(sync/async) network interface(s)
32K bytes of non-volatile configuration memory.
63488K bytes of ATA CompactFlash (Read/Write)

--- System Configuration Dialog ---

Continue with configuration dialog? [yes/no]:
```

이더넷 (둘다 같은 의미):

1. 랜카드 장비: Ethernet II
2. IP 장비: IEEE

```
Processor board ID JAD05190MTZ (4292891495)
M860 processor: part number 0, mask 49
Bridging software.
X.25 software, Version 3.0.0.
2 FastEthernet/IEEE 802.3 interface(s)
4 Low-speed serial(sync/async) network interface(s)
32K bytes of non-volatile configuration memory.
63488K bytes of ATA CompactFlash (Read/Write)
```

FastEthernet, IEEE 사용

```
ios (tm) c2600 software (c2600-1-m), version 12.2(20), RELEASE SOFTWARE
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2005 by cisco Systems, Inc.
Compiled Wed 27-Apr-04 19:01 by miwang

Cisco 2621 (MPC860) processor (revision 0x200) with 253952K/8192K bytes
memory
.
Processor board ID JAD05190MTZ (4292891495)
M860 processor: part number 0, mask 49
Bridging software.
X.25 software, Version 3.0.0.
2 FastEthernet/IEEE 802.3 interface(s)
4 Low-speed serial(sync/async) network interface(s)
32K bytes of non-volatile configuration memory.
63488K bytes of ATA CompactFlash (Read/Write)
```

RAM.

저장소 크기

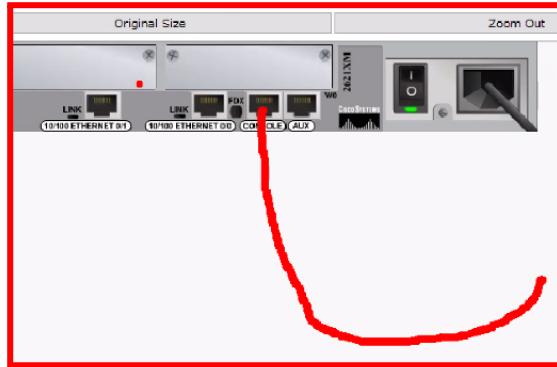
- 63488k bytes

```
Cisco 2621 (MPC860) processor (revision 0x200) with 253952K/8192K bytes of memory
```

- 라우터 메모리 크기 (RAM)
- 휘발성
- 동적 데이터

명령어: Show user

- 접속한 유저를 프린트해준다
- *은 본인이다.
- 0 con 0 :콘솔 포트를 이용해서 접속



```
Continue with configuration dialog? [yes/no]: no
```

```
Press RETURN to get started!
```

```
Router>show user
Line      User      Host(s)      Idle      Location
* 0 con 0          idle           00:00:00

Interface    User      Mode      Idle      Peer Address
Router>
```

ex)

```
Router>show user
Line      User      Host(s)      Idle      Location
* 0 con 0          idle           00:00:00
  66 vty 33  admin     1.4.5.23
```

- Id: admin
- IP: 1.4.5.23
- vty: virtual terminal 을 이용해서 접속

명령어 : show privilege

- 권한설정
 - Cisco에는 0~15까지의 권한이 있다.
 - 15는 모든 명령어 사용 (root)
 - 1~14는 제한적이다.

```
Router>show privilege  
Current privilege level is 1  
Router>
```

명령어: enable

- '#' 은 관리자라는 의미를 많이 가진다.

```
Router>enable  
Router#
```

- enable → show privilege:

```
Router#show privilege  
Current privilege level is 15  
Router#
```

- 또는 en으로 해도 축약가능하다:

en
ena
enab
enabl
enable

명령어: exit

- 나가기
 - 엔터시 재접속 가능

```
Router#exit  
  
Router con0 is now available  
  
Press RETURN to get started.
```

명령어: e?

- e와 관련된 명령어를 보여준다

Router>e?
enable exit

명령어: 탭 (tab) 키

- 자동완성

```
Router>en  
Router>ena  
Router>enable
```

명령어 : ?

- 사용 가능한 명령어 리스트를 보여준다

```
Router#?  
Exec commands:  
<1-99>      Session number to resume  
auto          Exec level Automation  
clear         Reset functions  
clock          Manage the system clock  
configure     Enter configuration mode  
connect        Open a terminal connection  
copy           Copy from one file to another  
debug          Debugging functions (see also 'undebbug')  
delete         Delete a file  
dir            List files on a filesystem  
disable        Turn off privileged commands  
disconnect    Disconnect an existing network connection  
enable         Turn on privileged commands  
erase          Erase a filesystem  
exit           Exit from the EXEC  
logout         Exit from the EXEC  
mkdir          Create new directory  
more           Display the contents of a file  
no             Disable debugging informations  
ping           Send echo messages  
reload         Halt and perform a cold restart  
--More--
```

- 명령어를 계속 쓰고 '?' 를 써도 적용된다:
 - cr (Carriage Return) 은 명령어를 치라는 의미다.

```
Router#show ip in  
Router#show ip int ?  
Ethernet          IEEE 802.3  
FastEthernet     FastEthernet IEEE 802.3  
GigabitEthernet GigabitEthernet IEEE 802.3z  
Loopback         Loopback interface  
Serial           Serial  
Tunnel           Tunnel interface  
Virtual-Access   Virtual Access interface  
Virtual-Template Virtual Template interface  
brief            Brief summary of IP status and configuration  
<cr>  
Router#show ip int bri ?  
<cr>
```

키보드 shortcut command:

- Ctrl + a (명령어 칠때 제일 앞 글자로 이동)
- Ctrl + e (명령어 칠때 제일 뒷 글자로 이동)
- Ctrl + shift + 6 (명령어 종료)
- Ctrl + C (취소)

명령어 오타

- 오타가 났을때 앞에서부터 알려준다.
- ^ 마커로 표시
- % 로 나오면 보통 오류 메세지

```
Router#show iq route
^
% Invalid input detected at '^' marker.
```

```
Router#show ip roue
^
% Invalid input detected at '^' marker.
```

- Ambiguous command

- 축약을 너무 심하게 해서 나온다.

```
Router#show i
% Ambiguous command: "show i"
Router#
```

```
Router#show i?
interfaces ip
```

- Incomplete command

- 명령어를 덜 쓸때 나오는 오류

```
Router#show ip
% Incomplete command.
Router#
```

1. 명령 프롬프트 모드

```
Router> User Mode : 라우터 접속 초기 프롬프트(명령어가 제한되어 있음)

Router>enable

Router# Privilege Exec Mode : 관리자 실행 모드(모든 명령어 가능함)

- show : 정적 정보 확인
- debug : 동적 정보 확인
- copy : 저장, 복사
- erase : 삭제
- reload : 재부팅
- ping : ping 테스트
- telnet : 텔넷 접속

Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
Router(config)# Global Configuration Mode : 전체 설정 모드(라우터 환경 설정이 가능함)
```

100M, 21초라는 결과가 있을때:

- show (100M)
- debug (21초)

[] 안에 들어와있으면 엔터치면 자동적용:

```
Router#reload
Proceed with reload? [confirm]
Router#
```

reload

- 재부팅

```
Router#reload
Proceed with reload? [confirm]
System Bootstrap, Version 12.1(3r)T2, RELEASE SOFTWARE (fc1)
Copyright (c) 2000 by cisco Systems, Inc.
Initializing memory for ECC

c2811 processor with 524288 Kbytes of main memory
main memory is configured to 64 bit mode with ECC enabled

Readonly ROMMON initialized

Self decompressing the image :
#####
##### [OK]

Restricted Rights Legend
```

conf t

- 설정 커맨드
- Configuration
 - 환경 설정
- Global Configuration Mode
 - 전체 설정 모드(라우터 환경 설정이 가능함)

```
Router#conf t  
Enter configuration commands, one per line. End with CNTL/Z.  
Router(config)#
```

- 명령어를 치면 리눅스환경에서 환경설정 파일을 열어서 수정하는것처럼 해준다.
 - 리눅스처럼 재시작도 필요없다

리눅스 시스템	웹 서버	httpd
---------	------	-------

```
vi /etc/httpd/httpd.conf
```

```
vi /etc/running-config
```

- 설정하고 치면 바로바로 변경된다. (실시간)
- Ctrl Z 사용시 관리자 모드로 나가진다.

```
Router(config-line)#^Z  
Router#  
%SYS-5-CONFIG_I: Configured from console by console  
  
Router#
```

2. 라우터 호스트 네임 설정

호스트 이름 변경:

```
Router(config)#hostname ABC  
ABC(config)#[
```

호스트 이름 삭제:

```
ABC(config)#no hostname ABC  
Router(config)#A
```

호스트 이름 덮어쓰기도 가능:

```
Router(config)#hostname HQ  
HQ(config)#hostname R1  
R1(config)#[
```

```
Router(config)#hostname ABC  
ABC(config)#  
ABC(config)#no hostname ABC // 'no' 명령어로 설정을 삭제할 수 있음  
Router(config)#  
Router(config)#hostname HQ  
HQ(config)#  
HQ(config)#hostname R1 // 덮어쓰기로 이름을 변경할 수 있음  
R1(config)#end  
R1#  
R1#show run
```

3. RAM & NVRAM

1) RAM

- 라우터의 램에 명령어가 들어가면 바로 설정된다.
 - 휘발성 메모리이다 (재부팅하면 기본값으로 돌아간다)
 - running-config 파일에 저장된다.

The screenshot shows three separate windows of the Windows Notepad application. The first window, titled '[RAM]', contains the command '@ running-config' with the word 'running' underlined in red. The second window, titled '[NVRAM]', contains the command '@ startup-config'. The third window, also titled '[NVRAM]', contains a configuration script for a router named 'Router'. The script includes commands like 'enable', 'conf t', 'configure terminal', and 'hostname ABC'. It then changes the hostname to 'HQ' and finally sets it back to 'R1'. The line 'hostname R1' is circled in red.

```
Router>
Router>
Router>enable
Router#
Router#
Router#confi
Router#configure te
Router#configure terminal
Router#
Router#
Router#
Router#conf t
Enter configuration commands, o
Router(config)#hostname ABC
ABC(config)#
ABC(config)#no hostname ABC
Router(config)#hostname HQ
HQ(config)#
HQ(config)hostname R1
R1(config)#
R1(config)#
R1(config)#

```

라우터가 동작하기 위해서 사용하는 메모리이다. 라우터 설정 내용이 'running-config' 파일에 저장되고 라우터는 이 파일 내용을 참고하여 동작한다. 정보 확인 명령어는 'show running-config' 이지만 너무 길기 때문에 'show run' 명령어를 권장한다. RAM에 저장된 내용들은 라우터가 재부팅되면 초기화되기 때문에 'running-config' 내용도 초기화된다.

2) NVRAM

- 램과 다르게 비휘발성이다.
 - NVRAM에 저장된 값을 RAM으로 보내서 실행시킨다.
 - 만약 NVRAM이 없으면 기본값으로 나온다.
 - startup-config 파일에 저장된다.

```
Router>
Router>
Router>enable
Router#
Router#
Router#confi
Router#configure te
Router#configure terminal
Router#
Router#
Router#
Router#
Router#conf t
Enter configuration commands, one
Router(config)#hostname ABC
ABC(config)#
ABC(config)#no hostname ABC
Router(config)#hostname HQ
HQ(config)#
HQ(config)#hostname R1
R1(config)#
R1(config)#
R1(config)#
R1#
%SYS-5-CONFIG_I: Configured from
```

RAM의 'running-config' 파일을 'startup-config' 파일로 저장하여 보관하는 메모리이다. 라우터가 부팅될 때, 마지막에 NVRAM을 확인하여 'startup-config' 설정 내용을 RAM 'running-config'로 복원시킨다. 정보 확인 명령어는 'show startup-config' 이지만 너무 길기 때문에 'show start' 명령어를 권장한다. NVRAM에 저장된 파일들은 라우터가 재부팅되어도 유지된다.

show run

- 설정한 내용을 확인할 수 있다.

```
R1#show run
Building configuration...

Current configuration : 671 bytes
!
version 12.2
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname R1
!
!
!
!
!
!
!
no ip cef
no ipv6 cef
!
!
--More--
```

3) RAM 'running-config'을 NVRAM 'startup-config'로 저장하는 명령어

'copy running-config startup-config' 명령어를 이용하여 저장하지만, 너무 길기 때문에 다음과 같이 축약하는 것을 권장한다.

```
R1#copy run start
Destination filename [startup-config]? 엔터
R1#show start
R1#reload
Proceed with reload? [confirm] 엔터
```

4) NVRAM 'startup-config' 삭제 명령어

라우터를 초기화하려면, 다음과 같이 NVRAM에 저장된 'startup-config' 파일을 삭제하고 재부팅한다.

```
R1>enable  
R1#show start  
R1#erase startup-config  
Erasing the nvram filesystem will remove all configuration files! Continue? [confirm] 엔터  
  
R1#reload  
Proceed with reload? [confirm] 엔터
```

erase start-config file

```
R1#erase start  
Erasing the nvram filesystem will remove all configuration files! Continue?  
[confirm]  
[OK]  
Erase of nvram: complete  
%SYS-7-NV_BLOCK_INIT: Initialized the geometry of nvram  
R1#
```

4. 라우터 환경 구성

- exec-timeout 0 0 (idle timeout을 없애는 커マン드)

```
Router>enable
Router#conf t
Router(config)#hostname R1
R1(config)#
R1(config)#no ip domain-lookup      // 관리자 모드에서 문자 입력시 DNS 요청을 실시하지 않는다.
R1(config)#
R1(config)#line con 0
R1(config-line)#exec-timeout 30 30    // 30 분 30초동안 입력 없으면, 콘솔을 종료한다.
R1(config-line)#
R1(config-line)#exec-timeout 0 0      // 0 분 0초로 설정하면, 콘솔을 종료하지 않는다.
R1(config-line)#
R1(config-line)#logg syn            // 콘솔 작업시 명령어 및 출력 내용에 대한 라인을 정리한다.
R1(config-line)#
R1(config-line)# end
R1#
R1#show run
```

- R1 설정을 참고하여 R2 와 R3 에도 설정을 실시한다.

- Idle timeout 설정

```
Router>en
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#line con 0
Router(config-line)#exec-timeout 30 30
Router(config-line)#
Router(config-line)#exec-timeout 0 0
```

- log를 출력할때 더럽게 나온다. (로그로 인해 명령어를 칠때 불편하다)

```
Router(config-if)#
Router(config-if)#int lo 3
Router(config-if)#
%LINK-5-CHANGED: Interface Loopback
Router(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol
ip address |
```

- 이런식으로 깔끔하게 만들수도 있다. (logg syn 명령어 사용)

```
Router(config-if)#  
Router(config-if)#int lo 3  
  
Router(config-if)#  
%LINK-5-CHANGED: Interface Loopback3, changed state to up  
  
%LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback3, changed state to up  
ip address  
% Incomplete command.  
Router(config-if)#  
Router(config-if)#  
Router(config-if)#  
Router(config-if)#line con 0  
Router(config-line)#logg syn  
Router(config-line)#exit  
Router(config)#  
Router(config)#int lo 5  
  
Router(config-if)#  
%LINK-5-CHANGED: Interface Loopback5, changed state to up  
  
%LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback5, changed state to up  
  
Router(config-if)#ip address |
```

콘솔에다가 'dsadsdasda'같이 문자열을 입력하면:

- DNS요청을 서버에다가 보낸다.
 - IP를 알아내서 Telnet으로 접속하는 용도
- 그러나 1분정도 걸려서 조금 불편하다.

```
Router#dsadsdasdasdsad  
Translating "dsadsadasdsad"...domain server (255.255.255.255) % Name lookup  
aborted
```

- 따라서 이 명령어로 막아준다

```
Router#conf t  
Enter configuration commands, one per line. End with CNTL/Z.  
Router(config)#no ip domain-lookup  
Router(config)#^Z  
Router#  
%SYS-5-CONFIG_I: Configured from console by console  
  
Router#  
Router#
```

한번에 라우터 설정법 (Shift + insert key):

```
en
conf t
hostname R2
no ip domain-lookup
!
line con 0
exc-timeout 0 0
log syn
end
!
```

```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname R2
R2(config)#no ip domain-lookup
R2(config)#!
R2(config)#line con 0
R2(config-line)# exc-timeout 0 0
^
% Invalid input detected at '^' marker.

R2(config-line)# logg syn
R2(config-line)# end
R2#!
R2#
SYS-5-CONFIG_I: Configured from console by console
R2#
```

```
en
conf t
hostname R3
no ip domain-lookup # DNS 안보내게 설정
!
line con 0
exc-timeout 0 0
log syn
end
!
```

```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname R3
R3(config)#no ip domain-lookup
R3(config)#!
R3(config)#line con 0
R3(config-line)# exc-timeout 0 0
^
% Invalid input detected at '^' marker.

R3(config-line)# logg syn
R3(config-line)# end
R3#!
R3#
SYS-5-CONFIG_I: Configured from console by console
R3#
```

- ! 은 엔터라는 뜻이다.

5. 라우터 패스워드 설정

```
R1#conf t
R1(config)#
R1(config)#enable secret cisco          // User 모드에서 관리자 모드로 전환할 때 패스워드를 요구한다.
R1(config)#
R1(config)#line con 0
R1(config-line)#password ciscocon      // 콘솔을 접속할 때 패스워드를 요구한다.
R1(config-line)#login
R1(config-line)#
R1(config-line)#line vty 0 4           // Telnet, SSH 와 같은 원격 터미널로 접속할 때 패스워드를 요구한다.
R1(config-line)#password ciscovty
R1(config-line)#login
R1(config-line)#end
R1#
R1#show run
```

- 콘솔을 종료하고 다시 접속하여 패스워드 설정 테스트를 실시한다.

```
R1#exit
R1 con0 is now available

Press RETURN to get started.

User Access Verification

Password: ciscocon

R1>
R1>enable
Password: cisco
R1#
```

- R1 설정을 참고하여 R2 와 R3 에도 설정 및 테스트를 실시한다.

show run으로 설정 값 확인:

```
line con 0
password ciscocon
logging synchronous
login
!
line aux 0
!
line vty 0 4
login
!
!
!
end
```

R3#

비밀번호 설정 확인:

```
User Access Verification
```

```
Password:
```

R3>

- 패스워드는 덮어쓰기할 수 있다. (재설정할때)

```
R3(config)#line vty ?
<0-15> First Line number
R3(config)#line vty 0 4
R3(config-line)#password ciscovty
R3(config-line)#login
R3(config-line)#

```

- vty 0 4는 가상 포트이다 (5개 사용)

설정 확인:

```
!
line con 0
password ciscocon
logging synchronous
login
!
line aux 0
!
line vty 0 4
password ciscovty
login
!
!
!
end
```

비밀번호 enable (관리자모드)

```
R3#conf t  
Enter configuration commands, one per line.  End with CNTL/Z.  
R3(config)#enable secret cisco  
R3(config)#^Z  
R3#
```

```
enable secret 5 $1$mERr$hx5rVt7rPNoS4wqbXXX7m0
```

- 해시값으로 저장된다.

Console, 관리자 비번 둘다 설정된다.

```
User Access Verification  
  
Password:  
  
R3>en  
Password:  
R3#
```

6. 패스워드 문자 암호화 실시

```
R1#show run
Building configuration...

Current configuration : 790 bytes
!
version 12.2
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname R1
!
!
!
enable secret 5 $1$mERr$hx5rVt7rPNoS4wqbXKX7m0
```

~ 중간 생략 ~

```
line con 0
exec-timeout 0 0
password ciscocon
login
!
line aux 0
!
line vty 0 4
password ciscovty
login
!
!
!
end
```

- 'service password-encryption' 명령어를 실행하여 콘솔 패스워드와 VTY 패스워드 문자를 암호화한다.

```
R1#conf t
R1(config)#service password-encryption
R1(config)#end
R1#
```

- 'show run' 명령어를 실행하여 콘솔 패스워드와 VTY 패스워드 문자가 암호화되었는지 확인한다.

```
R1#show run
Building configuration...
Current configuration : 824 bytes
!
version 12.2
no service timestamps log datetime msec
no service timestamps debug datetime msec
service password-encryption
!
hostname R1
!
```

```
!
enable secret 5 $1$mERr$hx5rVt7rPNoS4wqbXKX7m0

~ 중간 생략 ~

line con 0
exec-timeout 0 0
password 7 0822455D0A1606181C
login
!
line aux 0
!
line vty 0 4
password 7 0822455D0A1613030B
login
!
!
```

[참고] cisco password 7 복호화 사이트

<https://www.firewall.cx/cisco-technical-knowledgebase/cisco-routers/358-cisco-type7-password-crack.html>

콘솔 패스워드와 VTY 패스워드를 입력하여 복호화를 실시한다.

Ensure you only enter the **encrypted password**. For example, for the code below, you would paste the **yellow highlighted** portion. **Do not** include anything before the encrypted password.

username fcx password 7 **0/09285E4B1E18091B5C0814**

Encrypted Password:

Decrypted Password:

Ensure you only enter the **encrypted password**. For example, for the code below, you would paste the **yellow highlighted** portion. **Do not** include anything before the encrypted password.

username fcx password 7 **0709285F4B1E18091B5C0814**

Encrypted Password:

Decrypted Password:

<Review>

```
Router>en
Router#
Router#
Router#conf
Configuring from terminal, memory, or network [terminal]? t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname ?
    WORD This system's network name
Router(config)#hostname R1
R1(config)#
R1(config)#
R1(config)#
R1(config)#no ip domain-lookup
R1(config)#enable secret cisco
R1(config)#line con 0
R1(config-line)#password ciscocon
R1(config-line)#login
R1(config-line)#exec-timeout 0 0
R1(config-line)#[
```

enable secret cisco

- 콘솔 비번 설정

password ciscocon

- 관리자 비번 설정

login

- 로그인 설정

exec-timeout 0 0

- 자동 로그아웃 끄기

```
R1(config-line)#logg syn
R1(config-line)#kube vty 0 4
^
% Invalid input detected at '^' marker.

R1(config-line)#line vty 0 4
R1(config-line)#password ciscovty
R1(config-line)#login
R1(config-line)#[
```

logg syn

- 로그 라인정리

line vty 0 4

- 0 ~ 4 가상 포트까지 설정

password ciscovty

- 비번설정

login

- 로그인

```

R1(config)#
R1(config)#hostname R1
R1(config)#
R1(config)#
R1(config)#no ip domain-lookup
R1(config)#
R1(config)#
R1(config)#enable secret cisco
R1(config)#
R1(config)#line con 0
R1(config-line)#password ciscocon
R1(config-line)#login
R1(config-line)#exec-timeout 0 0
R1(config-line)#logg syn
R1(config-line)#
R1(config-line)#line vty 0 4
R1(config-line)#password ciscovt
R1(config-line)#login
R1(config-line)#
R1(config-line)#^Z
R1#
SYS-5-CONFIG_I: Configured from console by consol

```

- 이런 데이터는 램 메모리에 올라가 있다.

RAM안에는

- ‘running config’ 파일이 있다

휘발성이므로,

NVRAM

- ‘startup-config’ 파일을 쓴다

```

en
conf t
hostname R3
enable secret cisco
no ip domain-lookup
!
line con 0
password ciscocon
login
exec-timeout 0 0
logg syn
!
line vty 0 4
password ciscovt
login
end
!
```

gateway 설정:

```
R1>en
Password:
R1#
R1#sjsdlsdlhksdlhksdalhksdsdlhklshl
R1#
R1#
R1#show ip int brief
Interface          IP-Address      OK? Method Status  Protocol
FastEthernet0/0    unassigned      YES unset   administratively down down
FastEthernet0/1    unassigned      YES unset   administratively down down
Serial1/0          unassigned      YES unset   administratively down down
Serial1/1          unassigned      YES unset   administratively down down
Serial1/2          unassigned      YES unset   administratively down down
Serial1/3          unassigned      YES unset   administratively down down
R1#
R1#
```

L1 L2

레이어 1계층(전기신호)

- administratively down
- 명령어를 통해서 강제 다운 상태

레이어 2계층

- 만약 ethernet 프로토콜을 쓰면 up이라고 뜬다.

```
!
interface FastEthernet0/0
no ip address
duplex auto
speed auto
shutdown
!
interface FastEthernet0/1
no ip address
duplex auto
speed auto
shutdown
!
interface Serial1/0
no ip address
clock rate 2000000
shutdown
!
interface Serial1/1
no ip address
clock rate 2000000
shutdown
!
interface Serial1/2
no ip address
clock rate 2000000
shutdown
!
```

- 라우터 인터페이스는 기본적으로 닫혀있다.
- 공통적으로 shutdown이 있다.

여는방법:

%Link-5 ...

- ### - 전기신호가 열림

%LineProto ...

- up 상태로 바뀜

스위치는 기본적으로 열려있는 상태이다.

확인 (레이어 2계층이 하나 만들어짐):

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	13.13.10.1	YES	manual	up	up
FastEthernet0/1	unassigned	YES	unset	administratively down	down
Serial1/0	unassigned	YES	unset	administratively down	down
Serial1/1	unassigned	YES	unset	administratively down	down
Serial1/2	unassigned	YES	unset	administratively down	down
Serial1/3	unassigned	YES	unset	administratively down	down

라우터의 가장 중요한 정보:

```
R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
      * - candidate default, U - per-user static route, o - ODR
      P - periodic downloaded static route

Gateway of last resort is not set

      13.0.0.0/24 is subnetted, 1 subnets
C        13.13.10.0 is directly connected, FastEthernet0/0
```

7. FastEthernet 0/0 인터페이스 설정

```
R1#show arp
Protocol Address          Age (min)  Hardware Addr   Type    Interface
Internet 13.13.10.1       -          00D0.BA76.9D01  ARPA   FastEthernet0/0
Internet 13.13.10.2       1          0001.96E4.7109  ARPA   FastEthernet0/0
Internet 13.13.10.3       2          000C.8514.73C1  ARPA   FastEthernet0/0
R1#
```

```
conf t
line con 0
exec-timeout 0 0
logg syn
end
!
```

```
@ R2
conf t
int fa0/0
ip address 13.13.20.1 255.255.255.0
no shutdown
end
!
```

```
@R3
conf t
int fa0/0
ip address 13.13.20.1 255.255.255.0
no shutdown
end
!
```

```
R2, R3#show run
R2, R3#show ip int brief
R2, R3#show ip route
R2, R3#ping 13.13.z.z
```

<pre> @ R2 conf t int fa0/0 ip address 13.13.20.1 255.255.255.0 no shutdown end ! @ R3 conf t int fa0/0 ip address 13.13.30.1 255.255.255.0 no shutdown end !</pre>	<pre> @ R2, R3 conf t line con 0 exec-timeout 0 0 logg syn end ! R2,R3#show run R2,R3#show ip int brief R2,R3#show ip route R2,R3#ping 13.13.z.z </pre>
---	--

- 라우터에다가 설정

<pre> R1#conf t R1(config)#int fa0/0 R1(config-if)#ip address 13.13.10.1 255.255.255.0 R1(config-if)#no shutdown R1(config-if)#end R1# R1#show run ~ 중간 생략 ~ ! interface FastEthernet0/0 ip address 13.13.10.1 255.255.255.0 duplex auto speed auto !</pre>

```
R1#show ip int brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	13.13.10.1	YES	manual	up	up
FastEthernet0/1	unassigned	YES	unset	administratively down	down
Serial1/0	unassigned	YES	unset	administratively down	down
Serial1/1	unassigned	YES	unset	administratively down	down
Serial1/2	unassigned	YES	unset	administratively down	down
Serial1/3	unassigned	YES	unset	administratively down	down

```
R1#show int fa0/0
```

```
FastEthernet0/0 is up, line protocol is up (connected)
```

```
Hardware is Lance, address is 00d0.ba76.9d01 (bia 00d0.ba76.9d01)
```

```
Internet address is 13.13.10.1/24
```

```
MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
```

```
reliability 255/255, txload 1/255, rxload 1/255
```

```
Encapsulation ARPA, loopback not set
```

```
~ 중간 생략 ~
```

R2:

```
R2#show ip int brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	13.13.20.1	YES	manual	up	up
FastEthernet0/1	unassigned	YES	unset	administratively down	down
Serial1/0	unassigned	YES	unset	administratively down	down
Serial1/1	unassigned	YES	unset	administratively down	down
Serial1/2	unassigned	YES	unset	administratively down	down
Serial1/3	unassigned	YES	unset	administratively down	down

```
R2#show ip route
```

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
      * - candidate default, U - per-user static route, o - ODR
      P - periodic downloaded static route
```

```
Gateway of last resort is not set
```

```
    13.0.0.0/24 is subnetted, 1 subnets
```

```
C      13.13.20.0 is directly connected, FastEthernet0/0
```

```
R2#ping 13.13.20.2
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 13.13.20.2, timeout is 2 seconds:
```

```
.!!!!
```

```
Success rate is 80 percent (4/5), round-trip min/avg/max = 0/1/4 ms
```

처음에는 ARP 실패가 되기 때문에 안된다.

IP address 빼는법:

```
R2#conf t  
Enter configuration commands, one per line. End with CNTL/Z.  
R2 (config)#int fa0/0  
R2 (config-if)#no ip address
```

shutdown 다시 넣는법:

```
R2# conf t  
Enter configuration commands, one per line. End with CNTL/Z.  
R2 (config)#int fa0/0  
R2 (config-if)#shutdown
```

8. R1-R2 WAN 구간 인터페이스 설정

R1 구간:

show int s1/0

```
R1# show int s1/0  
Serial1/0 is administratively down, line protocol is down (disabled)  
Hardware is HD64570  
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,  
reliability 255/255, txload 1/255, rxload 1/255  
Encapsulation HDLC, loopback not set, keepalive set (10 sec)  
Last input never, output never, output hang never  
Last clearing of "show interface" counters never  
Input queue: 0/75/0 (size/max/drops); Total output drops: 0  
Queueing strategy: weighted fair  
Output queue: 0/1000/64/0 (size/max total/threshold/drops)  
Conversations 0/0/256 (active/max active/max total)  
Reserved Conversations 0/0 (allocated/max allocated)  
Available Bandwidth 1158 kilobits/sec  
5 minute input rate 0 bits/sec, 0 packets/sec  
5 minute output rate 0 bits/sec, 0 packets/sec  
0 packets input, 0 bytes, 0 no buffer  
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles  
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort  
0 packets output, 0 bytes, 0 underruns  
0 output errors, 0 collisions, 2 interface resets  
0 output buffer failures, 0 output buffers swapped out  
0 carrier transitions  
--More--
```

HDLC를 사용한다. ARP라는건 없다



동일하게 IP설정과 no shutdown을 해준다.
반대쪽이 닫혀있기때문에 no shutdown을 해도 down으로 나온다
반대쪽을 열어주면 up으로 나온다.

따라서 연결이 되어있지 않기 때문에 핑테스트가 안된다:

```

R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
      * - candidate default, U - per-user static route, o - ODR
      P - periodic downloaded static route

Gateway of last resort is not set

      13.0.0.0/24 is subnetted, 1 subnets
C          13.13.10.0 is directly connected, FastEthernet0/0
R1#ping 13.13.12.2

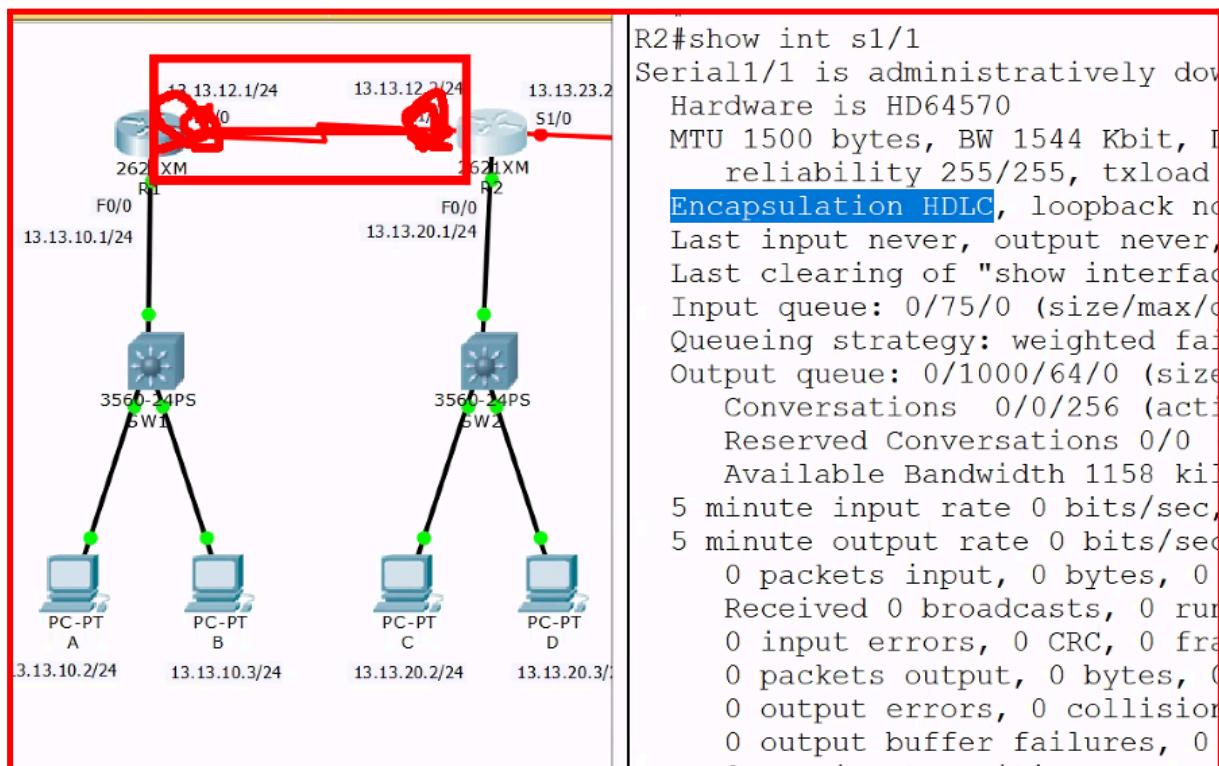
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 13.13.12.2, timeout is 2 seconds:
.....Success rate is 0 percent (0/5)

R1#

```

R2:

R1 과 R2은 같은 프로토콜을 사용해야 통신이 가능하다.



```

R2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#int s1/1
R2(config-if)#ip address 13.13.12.2 255.255.255.0
R2(config-if)#no shutdown

R2(config-if)#
%LINK-5-CHANGED: Interface Serial1/1, changed state to up

R2(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1, changed state to up

R2(config-if)#

```

똑같이 IP, no shutdown 해준다.

```

interface Serial1/1
  ip address 13.13.12.2 255.255.255.0
  clock rate 2000000
!

```

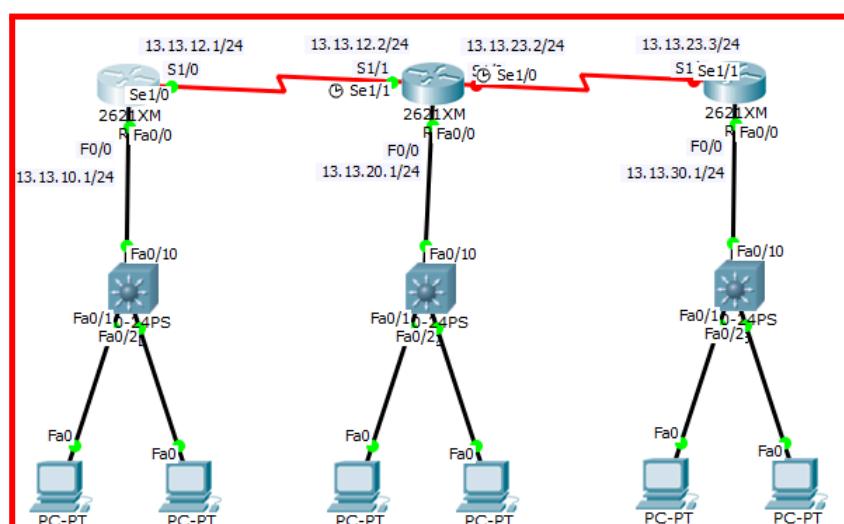
설정완료

```

R2# show ip int brief
Interface          IP-Address      OK? Method Status           Protocol
FastEthernet0/0    13.13.20.1      YES manual up            up
FastEthernet0/1    unassigned      YES unset administratively down down
Serial1/0          unassigned      YES unset administratively down down
Serial1/1          13.13.12.2      YES manual up            up
Serial1/2          unassigned      YES unset administratively down down
Serial1/3          unassigned      YES unset administratively down down
R2#

```

R1, R2가 연결되었음으로 Status, Protocol 둘다 up으로 나온다.



이런식으로 초록불로 연결된다.

```
R2#ping 13.13.12.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 13.13.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 3/7/10 ms

R2#
```

핑테스트 성공 R2 → R1

R1:

```
R1#show ip int brief
Interface          IP-Address      OK? Method Status        Protocol
FastEthernet0/0    13.13.10.1     YES manual up           up
FastEthernet0/1    unassigned      YES unset   administratively down down
Serial1/0          13.13.12.1     YES manual up           up
Serial1/1          unassigned      YES unset   administratively down down
Serial1/2          unassigned      YES unset   administratively down down
Serial1/3          unassigned      YES unset   administratively down down
R1#
```

Up 된거 확인가능

```
R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
      * - candidate default, U - per-user static route, o - ODR
      P - periodic downloaded static route

Gateway of last resort is not set

      13.0.0.0/24 is subnetted, 2 subnets
C        13.13.10.0 is directly connected, FastEthernet0/0
C        13.13.12.0 is directly connected, Serial1/0
R1#
```

WAN 네트워크가 연결된걸 확인가능하다.

명령어 복붙으로 한번에 설정하기:

```
en
conf t
hostname R3
no ip domain-lookup
enable secret cisco
!
line con 0
password ciscocon
login
exec-timeout 0 0
logg syn
!
line vty 0 4
password ciscovty
login
!
```

@ R1

```
conf t
int fa0/0
ip address 13.13.10.1 255.255.255.0
no shutdown
!
int s1/0
ip address 13.13.12.1 255.255.255.0
no shutdown
!
```

@ R2

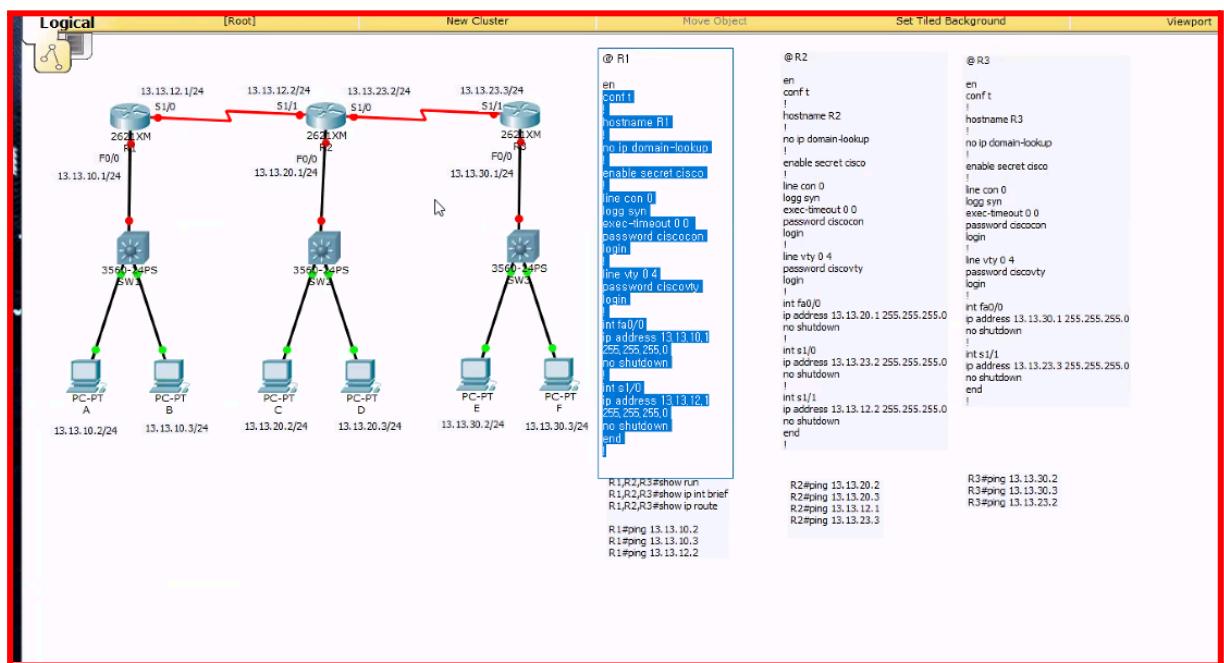
```
conf t
int fa0/0
ip address 13.13.20.1 255.255.255.0
no shutdown
!
int s1/0
ip address 13.13.23.2 255.255.255.0
no shutdown
!
int s1/1
ip address 13.13.12.2 255.255.255.0
no shutdown
!
```

@ R3

```
conf t
int fa0/0
ip address 13.13.30.1 255.255.255.0
no shutdown
!
int s1/1
ip address 13.13.23.3 255.255.255.0
no shutdown
!
```

같은 네트워크끼리 통신시키는 작업을 L2작업이라고 한다.

10-1번 명령어 참고



04/17/25

Salt Key:

```
!
enable secret 5 $1$mERr$hx5rVt7rPNoS4wqbXKX7m0
!
```

1 : MD5

hostname RI		
cisco	md5	adsADESAD
cisco	md5	adsADESAD
user1	abc+cisco	md5 SDF!@\$@ASCSDA\$
user3	xyz+cisco	md5 DFDV!#\$ASFDD\$

mERrcisco

비밀번호를 받아서 해시값으로 저장한다. 그러나 이것만으로는 취약함으로 솔트키라는 걸 추가해서 보완한다.

hostname RI		
cisco	md5	adsADESADASDF
cisco	md5	adsADESADASDF
user1	abc+cisco	md5 SDF!@\$@ASCSDA\$
user3	xyz+cisco	md5 DFDV!#\$ASFDD\$

mERrcisco

이런식으로 앞에 mERrci솔트키를 추가해서 해시값을 받아서 저장한다.

만약 PC에 방화벽이 켜져 있으면 ARP테이블을 확인:

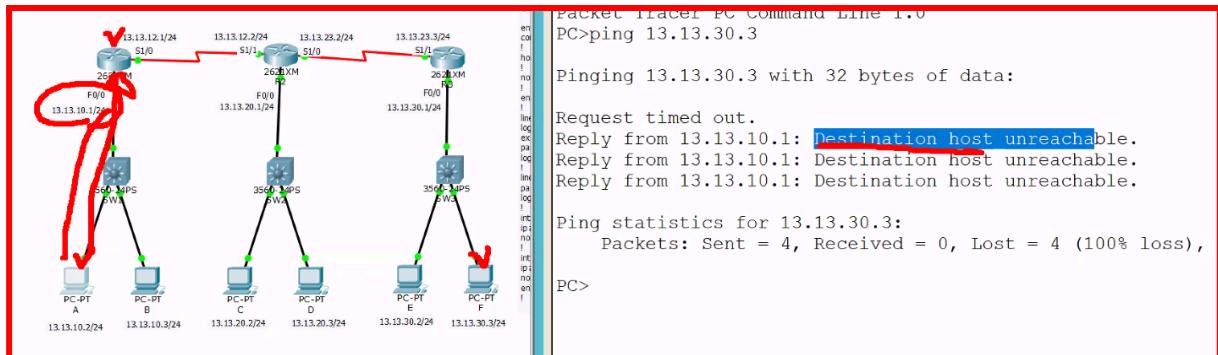
```
R1#show arp
Protocol Address          Age (min)  Hardware Addr   Type    Interface
Internet 13.13.10.1        -          00D0.BA76.9D01 ARPA   FastEthernet0/0
Internet 13.13.10.2        0          0001.96E4.7109 ARPA   FastEthernet0/0
Internet 13.13.10.3        0          000C.8514.73C1 ARPA   FastEthernet0/0
R1#
R1#
R1#
```

제2장 정적 경로 및 기본 경로 구성

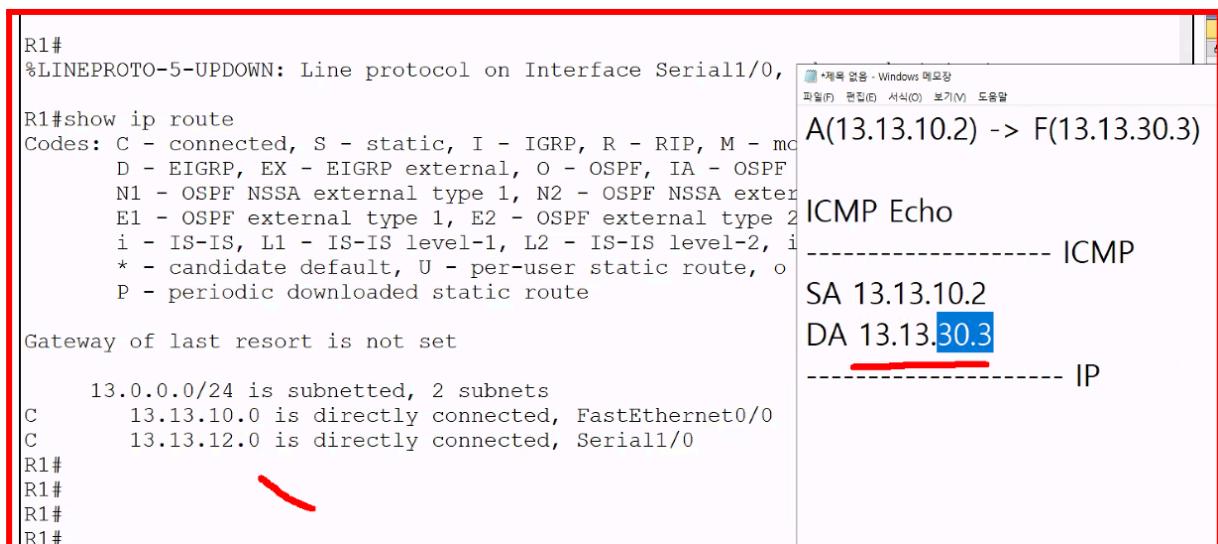
1. 정적 경로 구성

- '10-1.정적 경로 및 기본 경로 구성.pkt' 파일을 실행하여 기본 설정을 실시하고 정적 경로를 설정한다

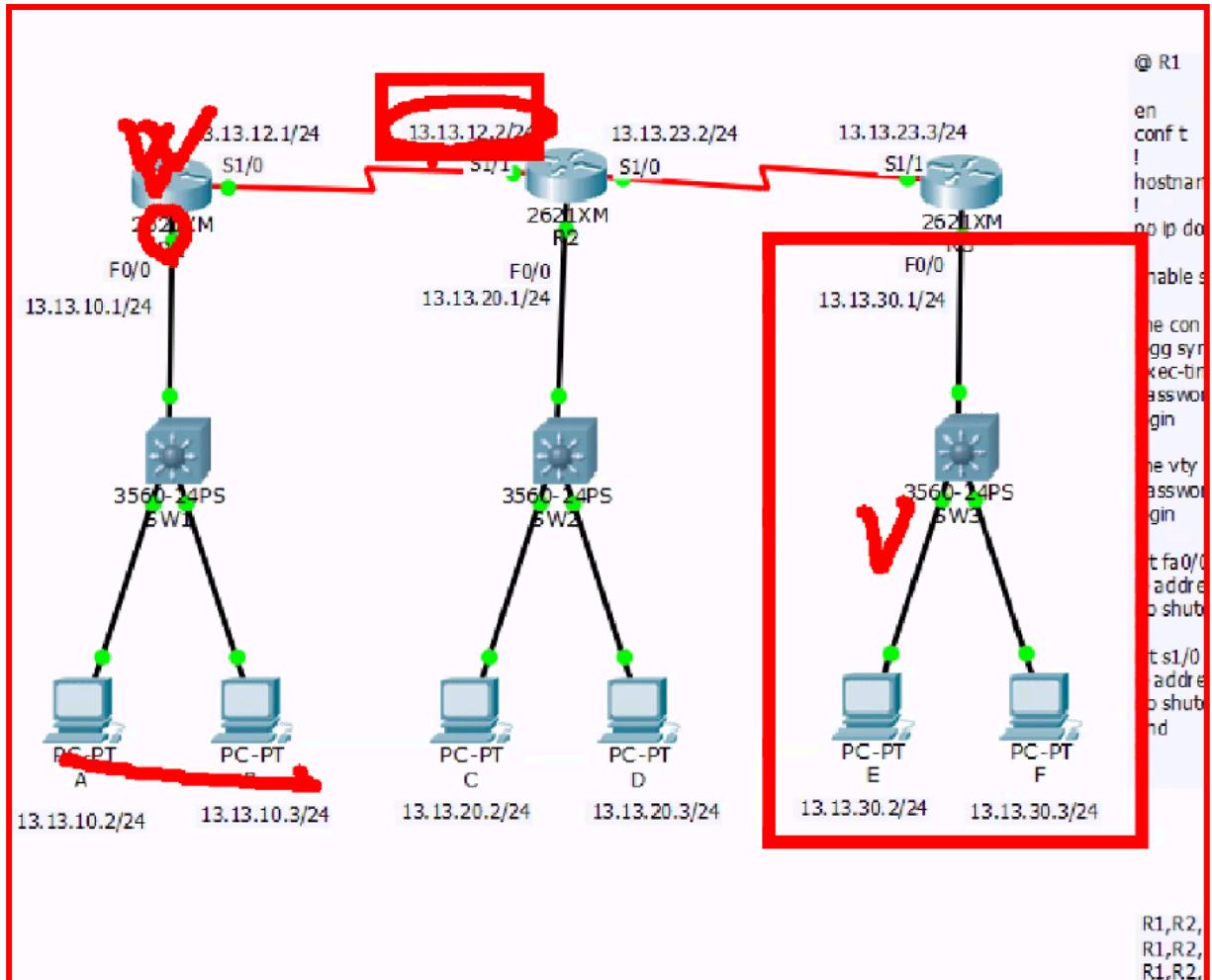
13.13.30.3으로 핑을 보낼 경우:



- R1에서 받아서 다시 돌려준다.



- 라우터에 대한 목적지 정보가 없기 때문에 드랍처리 해버린다.
- R1라우터에 목적지 30.3에 대한 경로가 없기 때문에 처리가 안된다.



R1 → Nexthop(Gateway) → 13.13.30.2

루트를 정해줘야 한다.

```

R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#
R1(config)#ip route ?
  A.B.C.D Destination prefix
R1(config)#ip route 13.13.30.0 ?
  A.B.C.D Destination prefix mask
R1(config)#ip route 13.13.30.0 255.255.255.0 ?
  A.B.C.D      Forwarding router's address
  Ethernet      IEEE 802.3
  FastEthernet   FastEthernet IEEE 802.3
  GigabitEthernet GigabitEthernet IEEE 802.3z
  Loopback      Loopback interface
  Null          Null interface
  Serial         Serial
R1(config)#ip route 13.13.30.0 255.255.255.0 13.13.12.2
R1(config)#

```

Nexthop: 13.13.12.2

R1 → 13.13.30.0(네트워크 이름) 으로 보내려 하는데 Nexthop13.13.12.2이다.

show run 으로 설정확인:

```
!
interface Serial1/2
no ip address
shutdown
!
interface Serial1/3
no ip address
shutdown
!
ip classless
ip route 13.13.30.0 255.255.255.0 13.13.12.2
!
I
ip flow-export version 9
!
!
```

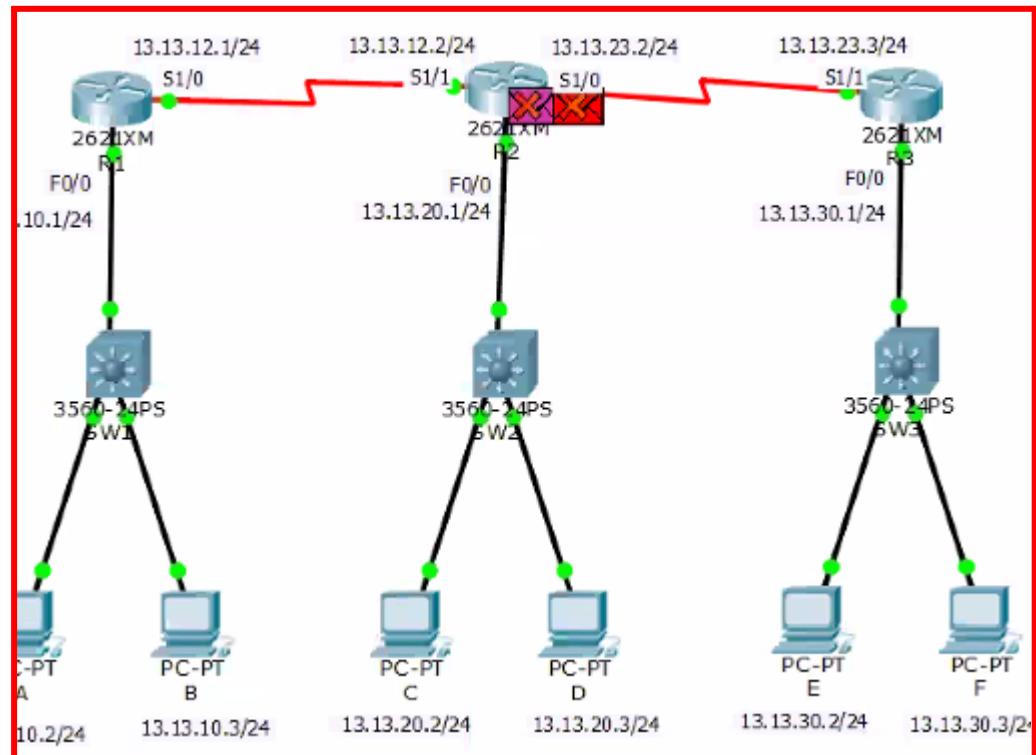
show ip route 확인:

```
Gateway of last resort is not set

      13.0.0.0/24 is subnetted, 3 subnets
C        13.13.10.0 is directly connected, FastEthernet0/0
C        13.13.12.0 is directly connected, Serial1/0
S          13.13.30.0 [1/0] via 13.13.12.2
R1#
```

- 게이트웨이 추가 완료.
- 패킷을 출력하는 인터페이스 정보가 안나온다.
 - 13.13.30.0 (목적지) 13.13.12.0 (경유지)를 통해서 나간다.

그러나 A PC부터 13.13.30.2까지 핑이 안나간다:



R2도 똑같이 설정을 해줘야 한다.

```
R2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#ip route 13.13.30.0 255.255.255.0 13.13.23.3
R2(config)#end
R2#
*SYS-5-CONFIG_I: Configured from console by console
R2#
```

show run

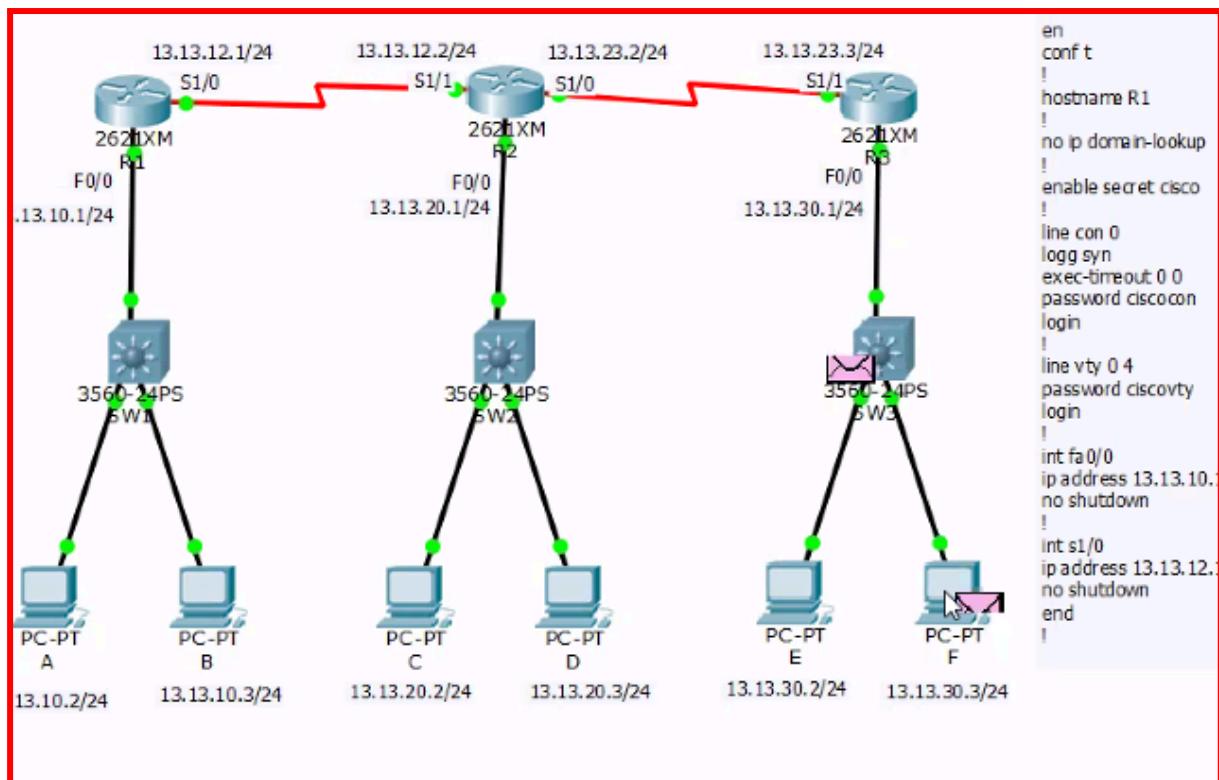
```
!
ip classless
ip route 13.13.30.0 255.255.255.0 13.13.23.3
!
ip flow-export version 9
!
```

show ip route

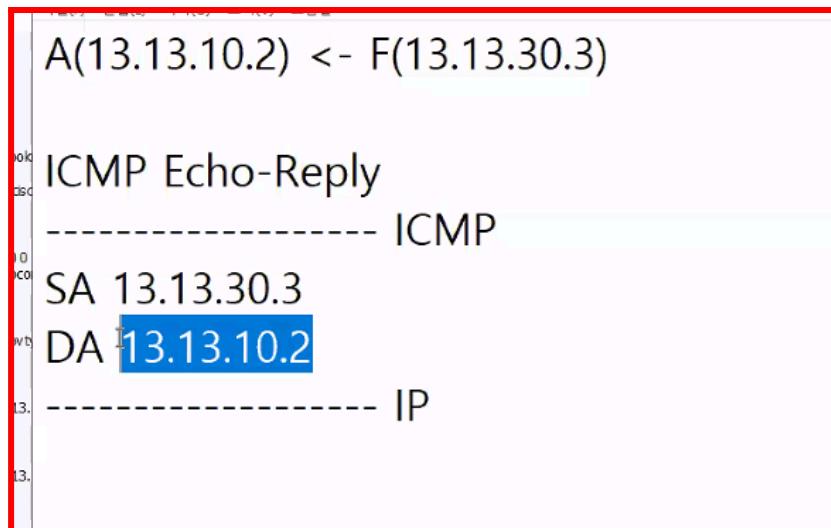
```
Gateway of last resort is not set

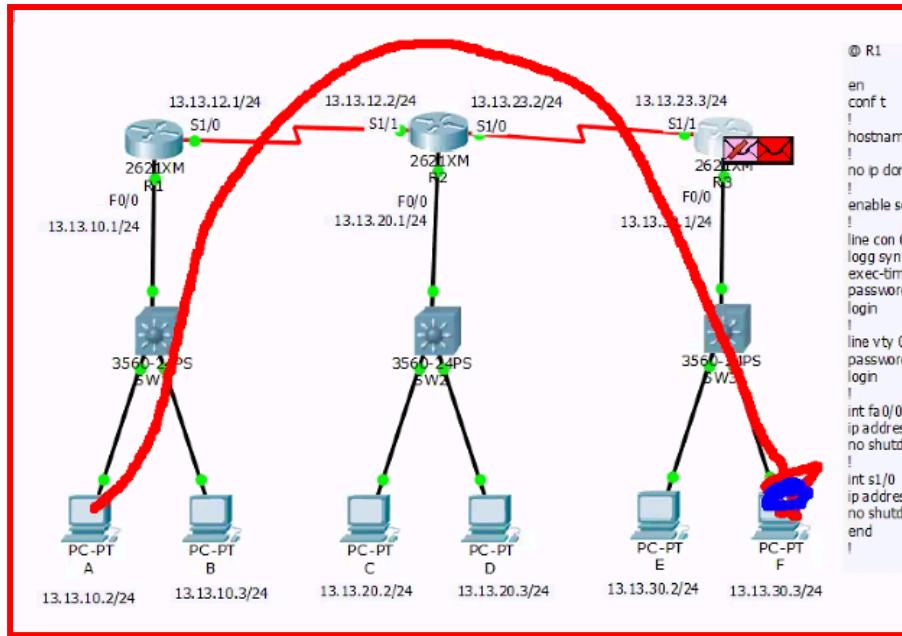
      13.0.0.0/24 is subnetted, 4 subnets
C        13.13.12.0 is directly connected, Serial1/1
C        13.13.20.0 is directly connected, FastEthernet0/0
C        13.13.23.0 is directly connected, Serial1/0
S        13.13.30.0 [1/0] via 13.13.23.3
R2#
```

그러나 아직도 안된다.



F PC까지는 나가지만 echo reply를 주지 않는다 (A까지 안준다)





여기까지는 이상이 없지만 돌아가는 경로가 없다.

- R3에서 경로를 추가해야한다.

R3 → R2

```

R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#ip route 13.13.10.0 255.255.255.0 13.13.23.2
R3(config)#end
R3#
SYS-5-CONFIG_I: Configured from console by console

```

show ip route

```

Gateway of last resort is not set

      13.0.0.0/24 is subnetted, 3 subnets
S        13.13.10.0 [1/0] via 13.13.23.2
C        13.13.23.0 is directly connected, Serial1/1
C        13.13.30.0 is directly connected, FastEthernet0/0
R3#

```

R2 → R1

```

R2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#ip route 13.13.10.0 255.255.255.0 13.13.12.1

```

show run

```

ip classless
ip route 13.13.30.0 255.255.255.0 13.13.23.3
ip route 13.13.10.0 255.255.255.0 13.13.12.1
!

```

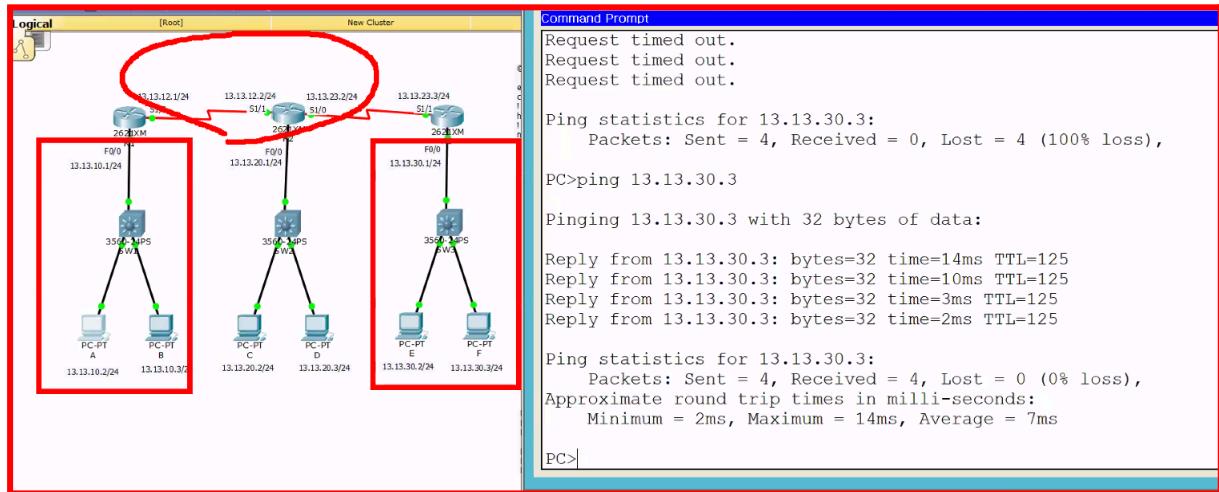
show ip route

```
Gateway of last resort is not set

      13.0.0.0/24 is subnetted, 5 subnets
S        13.13.10.0 [1/0] via 13.13.12.1
C        13.13.12.0 is directly connected, Serial1/1
C        13.13.20.0 is directly connected, FastEthernet0/0
C        13.13.23.0 is directly connected, Serial1/0
S        13.13.30.0 [1/0] via 13.13.23.3
```

같은방식으로 추가해준다.

이제 외부망을 통해서 통신이 가능하다:



R1 → R2

R2 → R3

까지 해주면 모든 네트워크가 사용 가능하다.

A>ping 13.13.23.3

안되는 이유?

R1,R2,R3#show ip route

```
PC>ping 13.13.23.3
Pinging 13.13.23.3 with 32 bytes of data:
Reply from 13.13.10.1: Destination host unreachable.

Ping statistics for 13.13.23.3:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

R1에서 경로가 없다.

```
Gateway of last resort is not set
      13.0.0.0/24 is subnetted, 4 subnets
C        13.13.10.0 is directly connected, FastEthernet0/0
C        13.13.12.0 is directly connected, Serial1/0
S          13.13.20.0 [1/0] via 13.13.12.2
S          13.13.30.0 [1/0] via 13.13.12.2
R1#
```

23.3에 대한 경로가 없다.

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#ip route 13.13.23.0 255.255.255.0 13.13.12.2
```

- 경로추가

```
!
ip classless
ip route 13.13.30.0 255.255.255.0 13.13.12.2
ip route 13.13.20.0 255.255.255.0 13.13.12.2
ip route 13.13.23.0 255.255.255.0 13.13.12.2
!
ip flow-export version 9
!
```

- 라우팅 테이블 확인

잘 나간다:

```
PC>ping 13.13.23.3

Pinging 13.13.23.3 with 32 bytes of data:

Reply from 13.13.23.3: bytes=32 time=2ms TTL=253
Reply from 13.13.23.3: bytes=32 time=4ms TTL=253
Reply from 13.13.23.3: bytes=32 time=6ms TTL=253
```

13.13.12.1

안되는 이유?

R1,R2,R3#show ip route

```
PC>ping 13.13.12.1

Pinging 13.13.12.1 with 32 bytes of data:

Reply from 13.13.30.1: Destination host unreachable.

Ping statistics for 13.13.12.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
    ...
```

R3에 12.1로 보내는 패킷이 없다.

```
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#ip route 13.13.12.0 255.255.255.0 13.13.23.2
R3(config)#

```

```
ip classless
ip route 13.13.10.0 255.255.255.0 13.13.23.2
ip route 13.13.20.0 255.255.255.0 13.13.23.2
ip route 13.13.12.0 255.255.255.0 13.13.23.2
!
```

```

Gateway of last resort is not set

  13.0.0.0/24 is subnetted, 5 subnets
S       13.13.10.0 [1/0] via 13.13.23.2
S       13.13.12.0 [1/0] via 13.13.23.2
S       13.13.20.0 [1/0] via 13.13.23.2
C       13.13.23.0 is directly connected, Serial1/1
C       13.13.30.0 is directly connected, FastEthernet0/0
R3#

```

The image shows three separate terminal windows, each titled "IOS Command Line Interface".

- R3:** Displays configuration commands like "line aux 0", "line vty 0 4", and "password ciscovtv". It also shows the output of "show ip route" which lists several routes including 13.13.10.0/24, 13.13.12.0/24, 13.13.20.0/24, 13.13.23.0/24, and 13.13.30.0/24.
- R1:** Displays configuration commands like "line aux 0", "line vty 0 4", and "password ciscovtv". It also shows the output of "show ip route" which lists several routes including 13.13.10.0/24, 13.13.12.0/24, 13.13.20.0/24, 13.13.23.0/24, and 13.13.30.0/24.
- R2:** Displays configuration commands like "line aux 0", "line vty 0 4", and "password ciscovtv". It also shows the output of "show ip route" which lists several routes including 13.13.10.0/24, 13.13.12.0/24, 13.13.20.0/24, 13.13.23.0/24, and 13.13.30.0/24.

- 각 라우터마다 5개의 네트워크가 다 등록되어 있다.
- 모든 네트워크끼리 통신이 가능해졌다.

LoopBack

- 가상 인터페이스 (테스트용)

```
@ R3
conf t
!
int lo 1
ip address 168.126.63.1 255.255.255.0
!
int lo 2
ip address 8.8.8.8 255.255.255.0
!
int lo 3
ip address 121.160.42.1 255.255.255.0
!
int lo 4
ip address 61.42.100.1 255.255.255.0
end
```

```
interface Loopback1
ip address 168.126.63.1 255.255.255.0
!
interface Loopback2
ip address 8.8.8.8 255.255.255.0
!
interface Loopback3
ip address 121.160.42.1 255.255.255.0
!
interface Loopback4
ip address 61.42.100.1 255.255.255.0
!
```

```
R3# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
      * - candidate default, U - per-user static route, o - ODR
      P - periodic downloaded static route

Gateway of last resort is not set

      8.0.0.0/24 is subnetted, 1 subnets
C        8.8.8.0 is directly connected, Loopback2
      13.0.0.0/24 is subnetted, 5 subnets
S          13.13.10.0 [1/0] via 13.13.23.2
S          13.13.12.0 [1/0] via 13.13.23.2
S          13.13.20.0 [1/0] via 13.13.23.2
C        13.13.23.0 is directly connected, Serial1/1
C        13.13.30.0 is directly connected, FastEthernet0/0
      61.0.0.0/24 is subnetted, 1 subnets
C        61.42.100.0 is directly connected, Loopback4
      121.0.0.0/24 is subnetted, 1 subnets
C        121.160.42.0 is directly connected, Loopback3
--More--
```

추가해준다

```
R2(config)#ip route 168.126.63.0 255.255.255.0 13.13.23.3
R2(config)#ip route 8.8.8.0 255.255.255.0 13.13.23.3
R2(config)#ip route 121.160.42.0 255.255.255.0 13.13.23.3
R2(config)#ip route 61.42.100.0 255.255.255.0 13.13.23.3
```

show run

```
ip classless
ip route 13.13.30.0 255.255.255.0 13.13.23.3
ip route 13.13.10.0 255.255.255.0 13.13.12.1
ip route 168.126.63.0 255.255.255.0 13.13.23.3
ip route 8.8.8.0 255.255.255.0 13.13.23.3
ip route 121.160.42.0 255.255.255.0 13.13.23.3
ip route 61.42.100.0 255.255.255.0 13.13.23.3
!
ip flow-export version 9
!
```

show ip route

```
R2#
R2#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
      * - candidate default, U - per-user static route, o - ODR
      P - periodic downloaded static route

Gateway of last resort is not set

      8.0.0.0/24 is subnetted, 1 subnets
S        8.8.8.0 [1/0] via 13.13.23.3
      13.0.0.0/24 is subnetted, 5 subnets
S        13.13.10.0 [1/0] via 13.13.12.1
C        13.13.12.0 is directly connected, Serial1/1
C        13.13.20.0 is directly connected, FastEthernet0/0
C        13.13.23.0 is directly connected, Serial1/0
S        13.13.30.0 [1/0] via 13.13.23.3
      61.0.0.0/24 is subnetted, 1 subnets
S        61.42.100.0 [1/0] via 13.13.23.3
      121.0.0.0/24 is subnetted, 1 subnets
S        121.160.42.0 [1/0] via 13.13.23.3
--More--
```

- 경로 등록완료

```

R2#ping 8.8.8.8
Translating "8.8.8.8"
% Unrecognized host or address or protocol not running.

R2#ping 8.8.8.8

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.8, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/11/18 ms

R2#ping 121.160.42.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 121.160.42.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/9/17 ms

```

핑테스트 성공

기본경로 설정

```

R1(config)#ip route 0.0.0.0 0.0.0.0 13.13.12.2
R1(config)#end
R1#
%SYS-5-CONFIG_I: Configured from console by console

R1#

```

- 목적지가 뭐가되든간에 12.2한테 보내라
- 0.0.0.0은 모든 IP를 이야기한다.

```

R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
      * - candidate default, U - per-user static route, o - ODR
      P - periodic downloaded static route

Gateway of last resort is 13.13.12.2 to network 0.0.0.0

      13.0.0.0/24 is subnetted, 5 subnets
C        13.13.10.0 is directly connected, FastEthernet0/0
C        13.13.12.0 is directly connected, Serial1/0
S          13.13.20.0 [1/0] via 13.13.12.2
S          13.13.23.0 [1/0] via 13.13.12.2
S          13.13.30.0 [1/0] via 13.13.12.2
S*        0.0.0.0/0 [1/0] via 13.13.12.2
R1#

```

- * 이 붙어있다
- 기본경로로 설정됨.

```
Gateway of last resort is 13.13.12.2 to network 0.0.0.0
```

- 만약 경로가 없을시, 최후의 수단으로 사용하도록 설정된다.

```

PC>netstat -r

Route Table
-----
Interface List
0x1 ..... PT TCP Loopback interface
0x2 ...00 16 6f 0d 88 ec ..... PT Ethernet interface
-----
Active Routes:
Network Destination      Netmask        Gateway        Interface Metric
          0.0.0.0          0.0.0.0    13.13.10.1   13.13.10.2     1
Default Gateway:         13.13.10.1
-----
Persistent Routes:
None

PC>ipconfig

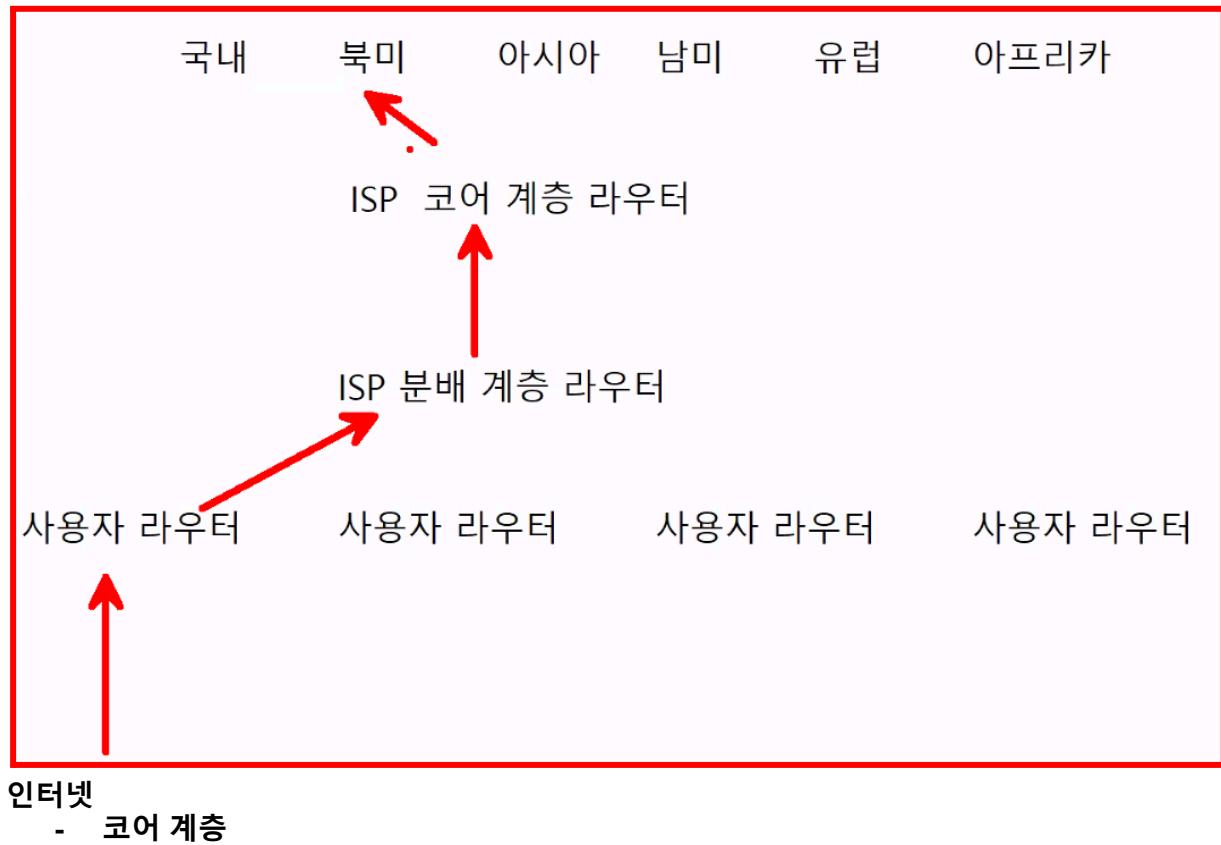
FastEthernet0 Connection:(default port)

Link-local IPv6 Address.....: FE80::201:96FF:FE4:7109
IP Address.....: 13.13.10.2
Subnet Mask.....: 255.255.255.0
Default Gateway.....: 13.13.10.1

PC>

```

- PC에서도 기본경로가 똑같이 적용된다.



사용자 라우터
- 기본경로

2. 정적 경로 설정 유형

1) 넥스트 흡 지정 방식

정적 경로는 관리자가 목적지 네트워크 정보와 넥스트-흡 정보를 파악하여 직접 설정하는 방식이다. 또한 요청 패킷을 전송하는 경로 뿐만 아니라 응답 패킷이 돌아올수 있도록 경로를 설정해야 한다.

1) 넥스트-흡 지정 방식

R1#conf t

R1(config)#ip route 13.13.30.0 255.255.255.0 13.13.12.2

Gateway of last resort is not set

13.0.0.0/24 is subnetted, 5 subnets

C 13.13.10.0 is directly connected, FastEthernet0/0

C 13.13.12.0 is directly connected, Serial1/0

S 13.13.20.0 [1/0] via 13.13.12.2

S 13.13.23.0 [1/0] via 13.13.12.2

S 13.13.30.0 [1/0] via 13.13.12.2

- via - ~~를 통해서 나간다.
- 만약 13.13.12.0 이 없어지면 (Connected 경로) 라우팅 테이블이 등록이 되지 않는다.

R1#show ip route

Gateway of last resort is not set

13.0.0.0/24 is subnetted, 5 subnets

C 13.13.10.0 is directly connected, FastEthernet0/0

C 13.13.12.0 is directly connected, Serial1/0

S 13.13.20.0 [1/0] via 13.13.12.2

S 13.13.23.0 [1/0] via 13.13.12.2

S 13.13.30.0 [1/0] via 13.13.12.2

S	: Static 경로
13.13.30.0	: 목적지 네트워크
[1/	: 정적 경로의 신뢰도(0~255)
/0]	: 메트릭
via	: 넥스트-흡 라우터 표시
13.13.12.2	: 넥스트-흡 IP 주소

- 신뢰도 0은 직접연결이다
- 메트릭이란
 - 지정된 경로까지 갈때 드는 비용 (cost)이다.

2) 인터페이스 지정 방식

```
@ R1
conf t
ip route 13.13.30.0 255.255.255.0 s1/0
!
```

```
ip route 13.13.30.0 255.255.255.0 13.13.12.2
!
```

- 설정할때는 위에꺼가 편하지만 공부할때는 밑에꺼가 더 편하다.

```
R1#conf t
R1(config)#no ip route 13.13.30.0 255.255.255.0 13.13.12.2
R1(config)#ip route 13.13.30.0 255.255.255.0 s1/0
R1(config)#end
R1#show ip route
```

Gateway of last resort is not set	
13.0.0.0/24 is subnetted, 5 subnets	S : Static 경로
C 13.13.10.0 is directly connected, FastEthernet0/0	13.13.30.0 : 목적지 네트워크
C 13.13.12.0 is directly connected, Serial1/0	directly connected : Connected 경로처럼 출력
S 13.13.20.0 [1/0] via 13.13.12.2	Serial1/0 : 패킷을 전송하는 인터페이스
S 13.13.23.0 [1/0] via 13.13.12.2	
S 13.13.30.0 is directly connected, Serial1/0	

3. 정적 기본 경로

- 패킷을 전송하기 위한 경로가 라우팅 테이블에 없을 경우, 가장 마지막에 사용하는 경로이다. 그렇기 때문에
- 사용자 라우터에는 기본 경로를 설정하여 ISP 라우터로 패킷을 전송할 수 있도록 구성하고 있다.
- 또한 시스템에서 기본 게이트웨이 IP 주소를 지정하면 시스템 라우팅 테이블에 기본 경로가 등록된다.
- R1에서 기존에 설정한 정적 경로를 삭제하고 정적 기본 경로를 설정한다.

```
R1#show ip route
```

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route

Gateway of last resort is 13.13.12.2 to network 0.0.0.0

13.0.0.0/24 is subnetted, 2 subnets
C 13.13.10.0 is directly connected, FastEthernet0/0
C 13.13.12.0 is directly connected, Serial1/0
S* 0.0.0.0/0 [1/0] via 13.13.12.2

0.0.0.0/0 정적경로 (다 보낸다)

Ex) 기본 경로를 주로 사용하는 라우터는 어떤 라우터인가?

(사용자)	(ISP)	(ISP Backbone)
Access 라우터-----	Distribution 라우터-----	Core 라우터-----
		인터넷(다른 지역/국가)

4. 라우팅 테이블 경로 검색 순서

1) 통기스트 매치를

- a) 라우팅 테이블에 경로를 검색할 때 가장 먼저 검사하며 패킷의 목적지 IP 주소에 대한 상세 경로를 먼저 사용하는 규칙이다.

Ex) 목적지 주소가 '192.168.30.1'인 패킷을 라우팅할 때 어떤 경로를 사용하는가?

- ① S 192.168.30.0/24 [1/0] via 13.13.12.2
- ② S 192.168.30.0/26 [1/0] via 13.13.102.2

192.168.30. 00000000

192.168.30.00 000000 <- 192.168.30.0

~

192.168.30.00 111111 <- 192.168.30.63

- 아래꺼가 아이피 범위가 더 상세하다 (0~255 vs 0~63).

2) 신뢰도 (Administrative Distance)

라우팅 테이블에 등록할 경로의 신뢰도를 의미한다. 범위는 '0~255'까지이며 신뢰도 값이 작은 경로가 라우팅 테이블에 우선적으로 등록된다. 다음은 경로에 대한 신뢰도 기본값이다.

Connected	0
Static	1
EIGRP	90
OSPF	110
RIP	120

Ex) '13.13.30.0/24' 네트워크에 대해서 Static 경로와 RIP 경로가 있다면, 라우팅 테이블에 등록되는 경로는?

- Static

3) 메트릭 (신뢰도, 룽기스트가 같을때)

로컬 라우터에서 목적지까지 도달하는데 필요한 비용이다. 경로를 선출할 때 사용하며 값이 작을 수록 최적 경로로 선출되어 라우팅 테이블에 등록된다.

Ex) 다음 중 라우팅 테이블에 등록되는 경로는 무엇인가?

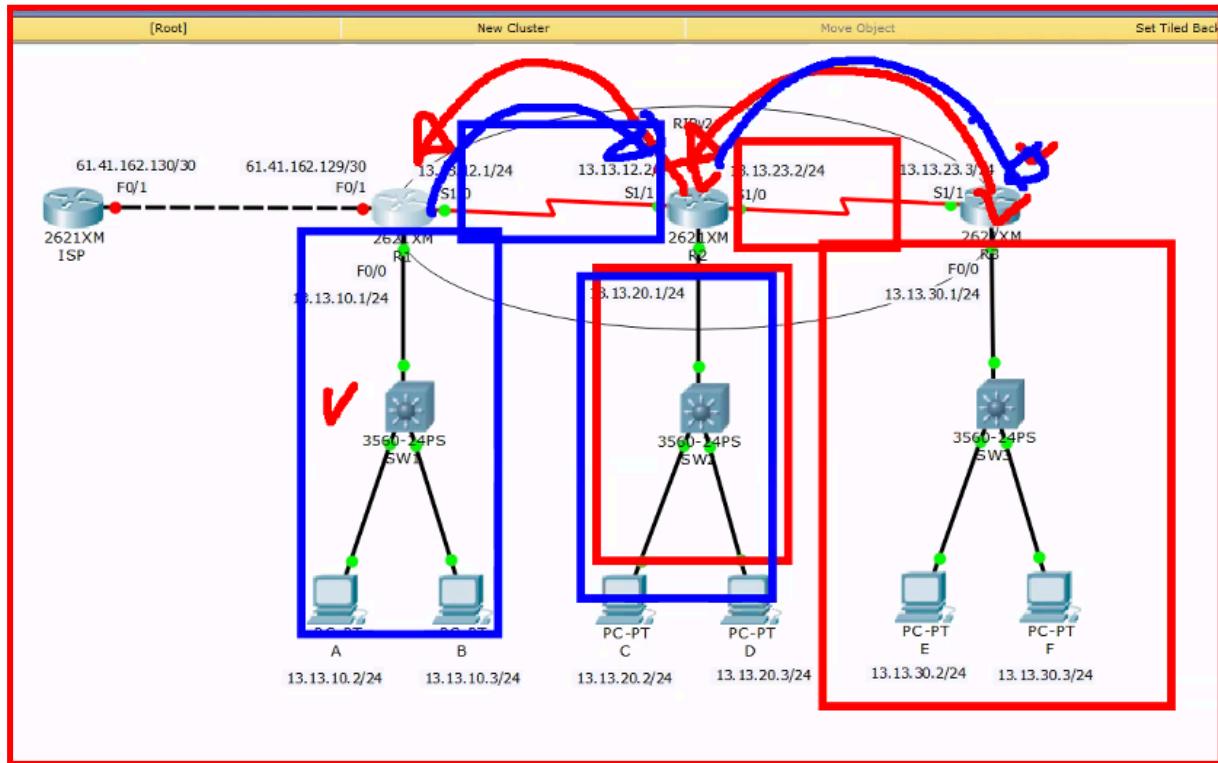
- ① R 13.13.30.0 [120/2] via 13.13.12.2, Serial1/0
- ② R 13.13.30.0 [120/5] via 13.13.14.4, Serial1/1

1번

- 룽기스트 매치율 [신뢰도 / 메트릭]

제3장 RIPv1 라우팅 프로토콜 (시험에 안나옴)

동적경로란?:



- 3번 라우터가 2번라우터한테 이런 네트워크로 구성되어 있다고 알려주는것
- 각각의 라우터들이 자기 네트워크를 알려준다. (라우팅 업데이트)
- 남의 네트워크가 아니라 자기 네트워크를 다른 라우터에게 알려주는것 (더 편하다)

라우팅

- 라우팅 테이블 → 패킷을 검색해서 → 내보내는것

```
R1#
R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
      * - candidate default, U - per-user static route, o - ODR
      P - periodic downloaded static route

Gateway of last resort is not set

      13.0.0.0/24 is subnetted, 2 subnets
C        13.13.10.0 is directly connected, FastEthernet0/0
C        13.13.12.0 is directly connected, Serial1/0
R1#
```

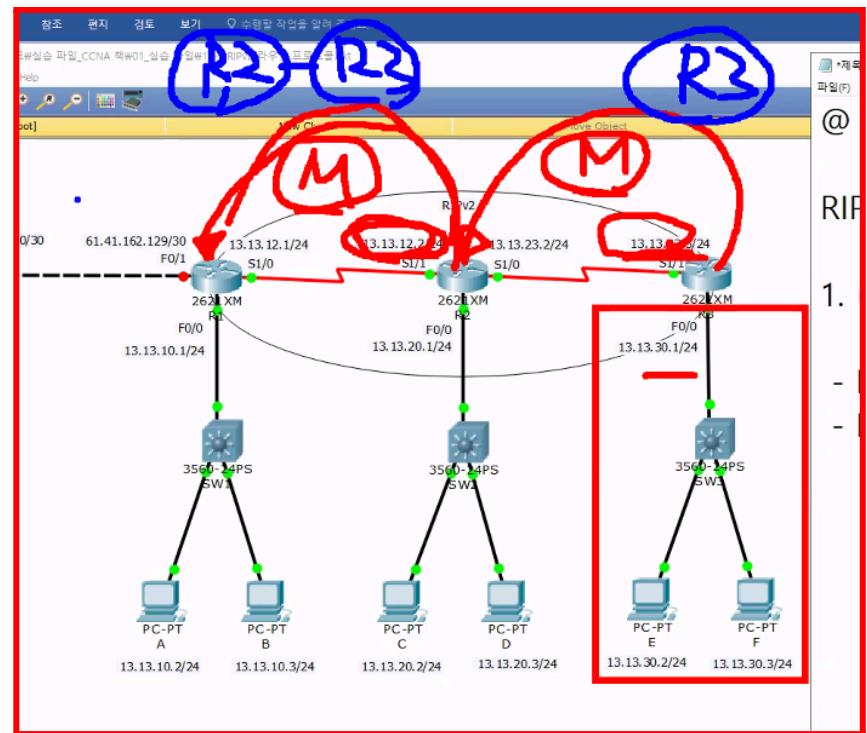
동적 경로 이용할 때 필요한 것

- 라우팅 프로토콜
 - RIPv1, RIPv2, IGRP, EIGRP, OSPF, ISIS, BFPv4
 - 중요한 것
 - OSPF, BGPv4
 - KT company: ISIS

- 공통적인 특징:

1. 라우팅 업데이트 동작 및 관리방식

- a. Distance Vector
 - i. 옆에 있는 메트릭 정보, Next-hop를 다른 라우터들에게 알려준다
 - ii. 그러나 몇 번 라우터인지는 모른다
 - iii. 가까이 가야지 보이는 지도이다. (비교적 부하가 적다)
- b. Link State
 - i. 모든 메트릭 정보, Next-hop를 다른 라우터들에게 알려준다
 - ii. 몇 번 라우터인지를 알려준다.
 - iii. 전체를 다 볼 수 있는 지도이다 (부하가 조금 있다)
 - iv. 모든 메트릭 정보를 볼 수 있기 때문에 조금 더 편리하다.



* 경로만 있으면 어떤것이든 상관없이 갈 수 있다.

2. 서브넷 처리

- Classful Routing Protocol (RIPv1, IGRP)
 - 13.13.30.0 / 24 ← 13.0.0.0
 - 사용안한다.
- Classless Routing Protocol (RIPv2, EIGRP, OSPF, ISIS)
 - 13.13.30.0 / 24 ← 13.13.30.0/24
 - 이걸 사용한다.
 - 네트워크를 서브넷이 아닌 클래스로 처리하며 라우팅 업데이트 시 서브넷 마스크가 포함되지 않는다.
 - VLSM 환경에서 라우팅 업데이트가 불가능하며, CIDR 기능을 지원하지 않는다.

3. 사용 구간

- IGP (RIPv1, RIPv2, IGRP, EIGRP, OSPF, ISIS)
 - 라우팅 업데이트 속도가 빠르다
 - 많은 양의 업데이트가 불가하다.
 - 몇천 개, 몇만 개
 - 장비 부하
 - 네트워크 망을 만들 때 사용
- EGP (BGPv4)
 - 스피드는 느리지만 파워가쎈
 - ISP업체 ↔ ISP업체 라우팅 업데이트 할 때 사용
 - 망과 망을 업데이트 할 때 사용

4. 주요 라우팅 프로토콜

- OSPF, BGPv4 ← 이것만 쓴다.

1. RIPv1 (Routing Information Protocol Version 1)

- Distance Vector
- Classful Routing Protocol
- IGP

1) RIPv1 라우팅 설정 방법

'network' 명령어를 이용하여 로컬 네트워크 서브넷을 다음과 같이 원본 클래스 이름으로 설정한다.

```
Router(config)#router rip
Router(config-router)#network A.0.0.0
Router(config-router)#network B.B.0.0
Router(config-router)#network C.C.C.0
Router(config-router)#end
```

0~127	0
128~191	10
192~223	110

a, b, c 클래스

F0/0	13.13.10.1/24	13.13.10.0/24
F0/1	121.160.1.1/24	121.160.1.0/24
S1/0	183.45.21.2/24	183.45.21.0/24
S1/1	198.133.219.133/24	198.133.219.9/24

```
router rip
network 13.0.0.0
network 121.0.0.0
network 183.45.0.0
network 198.133.219.0
```

2) R1, R2, R3 RIPv1 라우팅 프로토콜 설정

a) R1, R2, R3 로컬 네트워크에 할당한 서브넷의 원본 클래스가 '13.0.0.0' A 클래스이므로 설정이 동일하다.

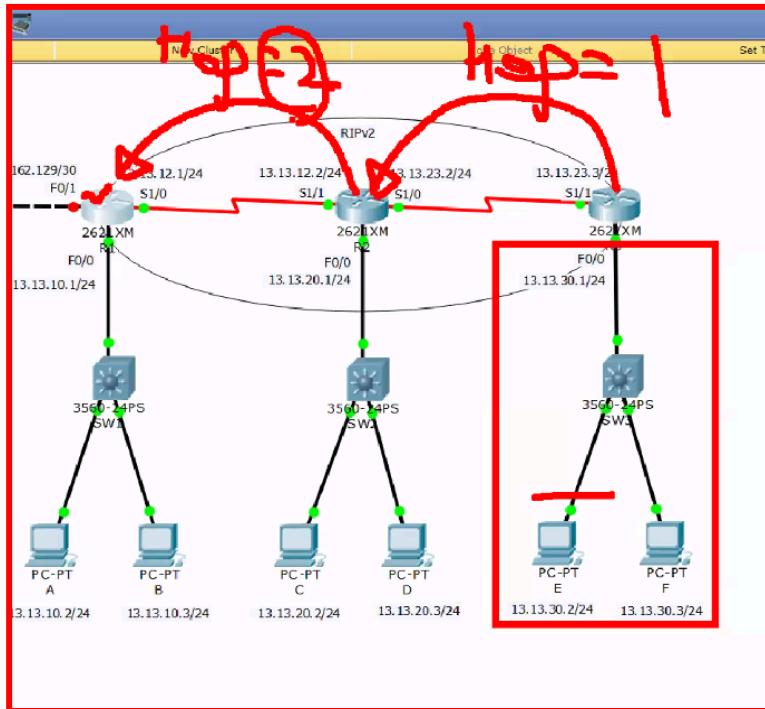
```
R1,R2,R3#conf t
R1,R2,R3(config)#router rip
R1,R2,R3(config-router)#network 13.0.0.0
R1,R2,R3(config-router)#end
R1,R2,R3#
```

```
R1,R2,R3#show run
```

```
~ 중간 생략 ~
!
router rip
  network 13.0.0.0
!
```

메트릭

- 메트릭 단위는 라우터 개수를 의미하는 흙(Hop)을 사용하며 범위는 0 부터 16 이다. 실제 사용 가능한 범위는 1 부터 15이며 흙이 16 이면 더 이상 도달할 수 없다는 의미이다
- 16은 사용할 수 없다. (도달할 수 없을 때 사용, 라우팅 테이블에서 삭제시키는 기능)
- 0 보다 작은건 1이다.
- 결국 사용하는건 1~15이다.



라우터 2개 지나면 있다.

4. 라우팅 업데이트 방식

- 라우팅 업데이트시 목적지 IP 주소를 브로드캐스트(255.255.255.255)로 설정하여 30초마다 주기적으로 라우팅 업데이트를 실시한다. 그렇기 때문에 RIPv1 라우팅 업데이트가 전송될 필요 없는 내부 네트워크 인터페이스는 'passive-interface' 명령어를 이용하여 전송되지 않도록 차단하는 것을 권장한다

```
R1#conf t
R1(config)#service timestamps debug datetime msec
R1(config)#router rip
R1(config-router)#passive-interface fa0/0
R1(config-router)#end
R1#
```

show ip protocols

```
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 23 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Redistributing: rip
  Default version control: send version 1, receive any version
    Interface      Send   Recv   Triggered RIP  Key-chain
    Serial1/0        1       2     1
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    13.0.0.0
```

Passive Interface(s):

FastEthernet0/0

Routing Information Sources:

Gateway	Distance	Last Update
13.13.12.2	120	00:00:20

Distance: (default is 120)

debug ip rip

- 명령어를 실행하여 RIPv1 라우팅 업데이트 동작 디버깅을 실시한다.

undebbug all

- 라우팅 업데이트 동작 확인이 완료되었다면, 'undebbug all' 명령어를 실행하여 디버깅을 종료한다.

R2, R3에서 RIPv1 라우팅 업데이트가 F0/0 인터페이스로 전송되지 않도록 'passive-interface' 설정을 실시한다

```
R2#conf t  
R2(config)#router rip  
R2(config-router)#passive-interface fa0/0  
R2(config-router)#end  
R2#
```

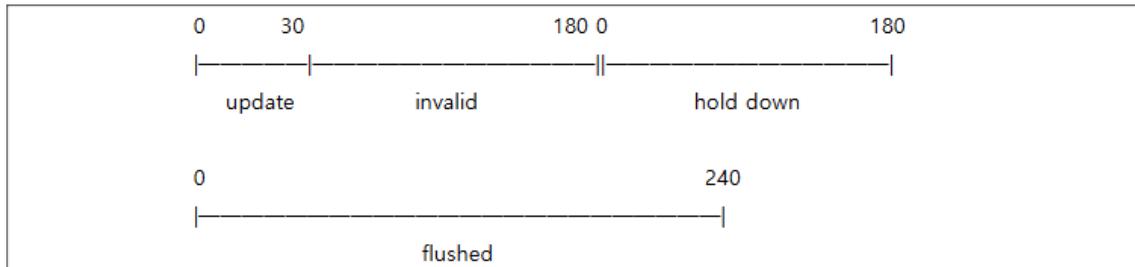
```
R3#conf t  
R3(config)#router rip  
R3(config-router)#passive-interface fa0/0  
R3(config-router)#end  
R3#
```

```
R2,R3#show run  
R2,R3#show ip protocol
```

5. 균등 로드 분산

- 목적지 네트워크에 대한 경로의 메트릭이 동일한 경우 자동으로 구현된다.

- update(30 초) : 라우팅 업데이트 주기
- invalid(180 초) : update 타이머 이내에 라우팅 업데이트를 못받으면, 기회를 더 제공하는 시간
- hold down(180 초) : invalid 타이머 이내에 라우팅 업데이트를 못받으면, 경로를 삭제 대기 시간
- flushed(240 초) : 경로를 라우팅 테이블에서 삭제하는 타이머



6. RIP 타이머

- RIP 경로를 라우팅 업데이트하거나 RIP 경로를 유지 및 삭제할 때 사용하는 타이머이다.

7. RIP 삭제

```
R1,R2,R3#conf t  
R1,R2,R3(config)#no router rip  
R1,R2,R3(config)#end  
R1,R2,R3#
```

제4장 RIPv2 라우팅 프로토콜 (시험에 안나옴)

1. RIPv2(Routing Information Protocol Version 2)

- Distance Vector
- **Classless Routing Protocol (바뀐점)**
- auto-summary -> no auto-summary
- IGP

1) RIPv2 라우팅 프로토콜 설정방법

'network' 명령어를 이용하여 로컬 네트워크 서브넷을 다음과 같이 원본 클래스 이름으로 설정한다.

```
Router(config)#router rip
Router(config-router)#version 2
Router(config-router)#no auto-summary
Router(config-router)#network A.0.0.0
Router(config-router)#network B.B.0.0
Router(config-router)#network C.C.C.0
Router(config-router)#end
```

- version 2
- no auto-summary
- 이것만 바꿨다.

2. Classless Routing Protocol

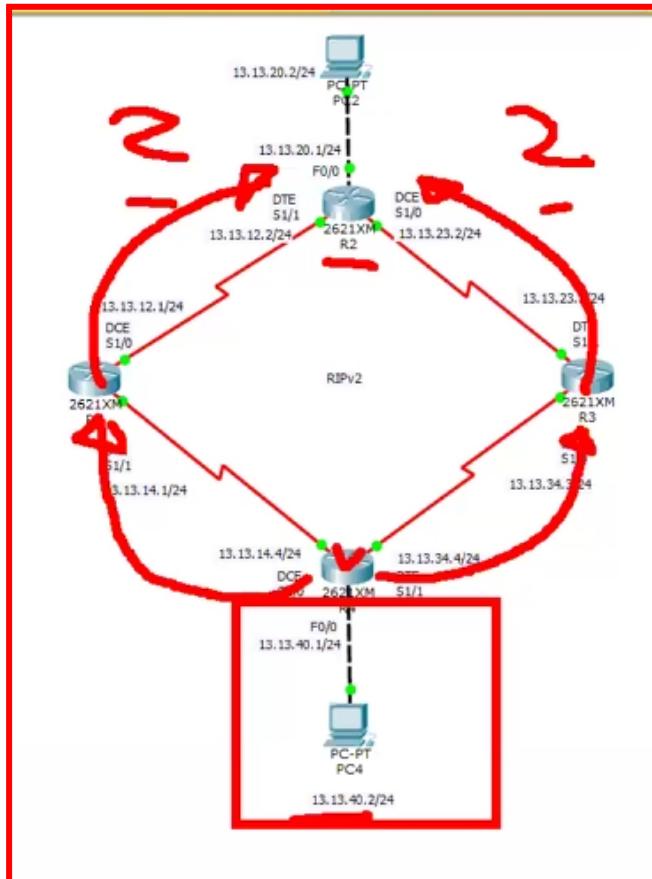
서브넷 마스크를 이용하여 네트워크를 서브넷으로 처리하며 라우팅 업데이트시 서브넷 마스크가 포함된다.
VLSM 환경에서 라우팅 업데이트가 가능하며, CIDR 기능을 지원한다.

4. 라우팅 업데이트 방식

라우팅 업데이트시 목적지 IP 주소를 멀티캐스트(224.0.0.9)로 설정하여 30 초마다 주기적으로 라우팅 업데이트를 실시한다. 그렇기 때문에 RIPv2 라우팅 업데이트가 전송될 필요 없는 내부 네트워크 인터페이스는 'passive-interface' 명령어를 이용하여 전송되지 않도록 차단하는 것을 권장한다.

```
R1#conf t
R1(config)#service timestamps debug datetime msec
R1(config)#router rip
R1(config-router)#passive-interface fa0/0
R1(config-router)#end
R1#
```

5. 균등 로드 분산



- 이 경우 두 경로 다 네스트홉이 2개이다.
- 메트릭이 동일하다.

```
R2#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
      * - candidate default, U - per-user static route, o - ODR
      P - periodic downloaded static route

Gateway of last resort is not set

      13.0.0.0/24 is subnetted, 6 subnets
C        13.13.12.0 is directly connected, Serial1/1
R        13.13.14.0 [120/1] via 13.13.12.1, 00:00:01, Serial1/1
C        13.13.20.0 is directly connected, FastEthernet0/0
C        13.13.23.0 is directly connected, Serial1/0
R        13.13.34.0 [120/1] via 13.13.23.3, 00:00:28, Serial1/0
R        13.13.40.0 [120/2] via 13.13.23.3, 00:00:28, Serial1/0
                           [120/2] via 13.13.12.1, 00:00:01, Serial1/1

R2#
```

이 경우 둘다 등록되며, 패킷을 보낼 때 반반씩 보낸다. (분산처리로 보낸다) - load distribution

7. RIP 컨버전스

1) 루트 포이즌(Route Poison)

장애가 발생한 RIP 네트워크 정보에 대해서 Hop=16 정보를 업데이트하는 동작이다.
Hop=16 인 RIP 네트워크는 더 이상 도달이 불가능한 네트워크를 의미한다.

2) 리버스 포이즌(Reverse Poison)

루트 포이즌에 대한 응답이다.
Hop=16에 대한 응답이며, 네트워크 도달 불가능한 정보를 역으로 업데이트하는 동작이다.

```
장애 발생          hop=16 ->          hop=16 ->
13.13.10.0/24---[F0/0]R1[S1/0]-----[S1/1]R2[S1/0]-----[S1/1]R3
                           <- hop=16           <- hop=16
```

8. RIP 라우팅 업데이트 방지

1) Split-Horizon

RIP 라우팅 업데이트 루프를 방지하는 기능이다.
라우팅 정보를 수신한 인터페이스로 라우팅 업데이트가 나가는 것을 차단한다.

```
R3#show ip int s1/1
```

```
~ 중간 생략 ~
```

```
Local Proxy ARP is disabled
Security level is default
Split horizon is enabled
```

2) Hop Count Limit

Hop 범위를 0~16으로 제한하는 기능이다.
라우팅 업데이트 루프가 발생하여, Hop=16으로 된 RIP 경로를 라우팅 테이블에 삭제한다.

- Split Horizon이 있기 때문에 안쓴다.

제5장 EIGRP 라우팅 프로토콜

EIGRP(Enhanced Interior Gateway Routing Protocol)

- Cisco 전용 라우팅 프로토콜
- Advanced Distance Vector
- Classless Routing Protocol, VLSM, CIDR
- auto-summary → no auto-summary
- IGP
 - 빠른 업데이트를 지향, 많은 양 X

1) EIGRP 라우팅 프로토콜 설정 방법:

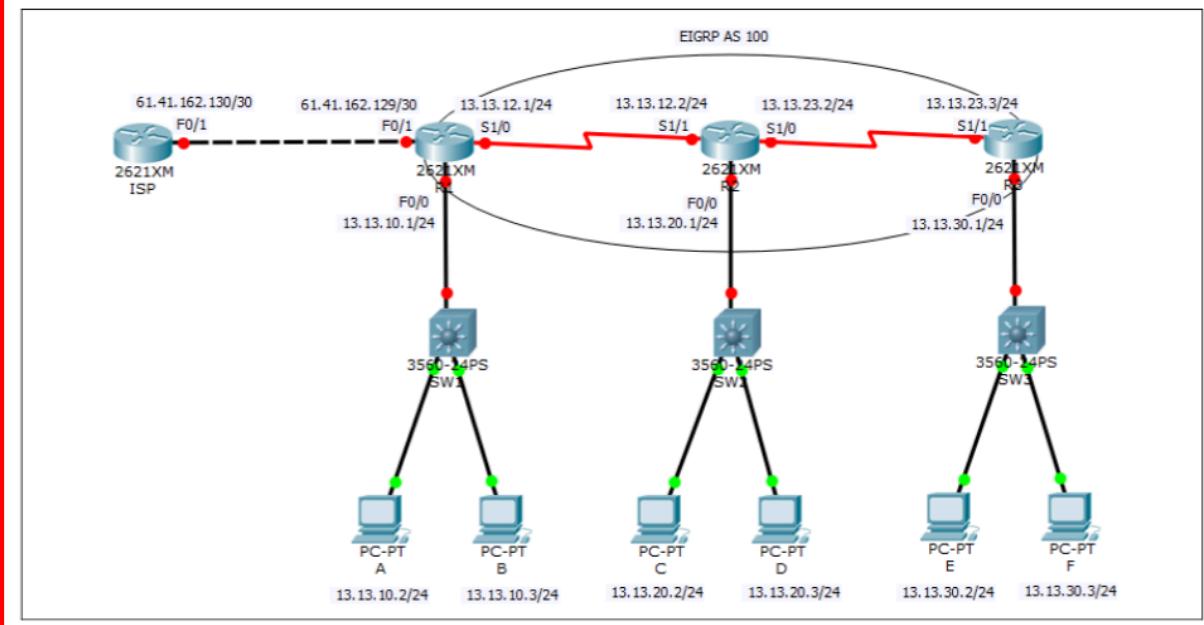
'network' 명령어를 이용하여 로컬 네트워크 서브넷을 다음과 같이 원본 클래스 이름으로 설정한다.
EIGRP에서는 AS 번호를 이용하여 EIGRP 라우팅 프로토콜 프로세스를 구분한다. 이때, 인접 라우터와 AS 번호가 동일해야지만 EIGRP 네이버를 성립하고 라우팅 업데이트를 실시한다.

```
Router(config)#router eigrp [AS Number 1~65535]
Router(config-router)#no auto-summary
Router(config-router)#network A.0.0.0
Router(config-router)#network B.B.0.0
Router(config-router)#network C.C.C.0
Router(config-router)#end
```

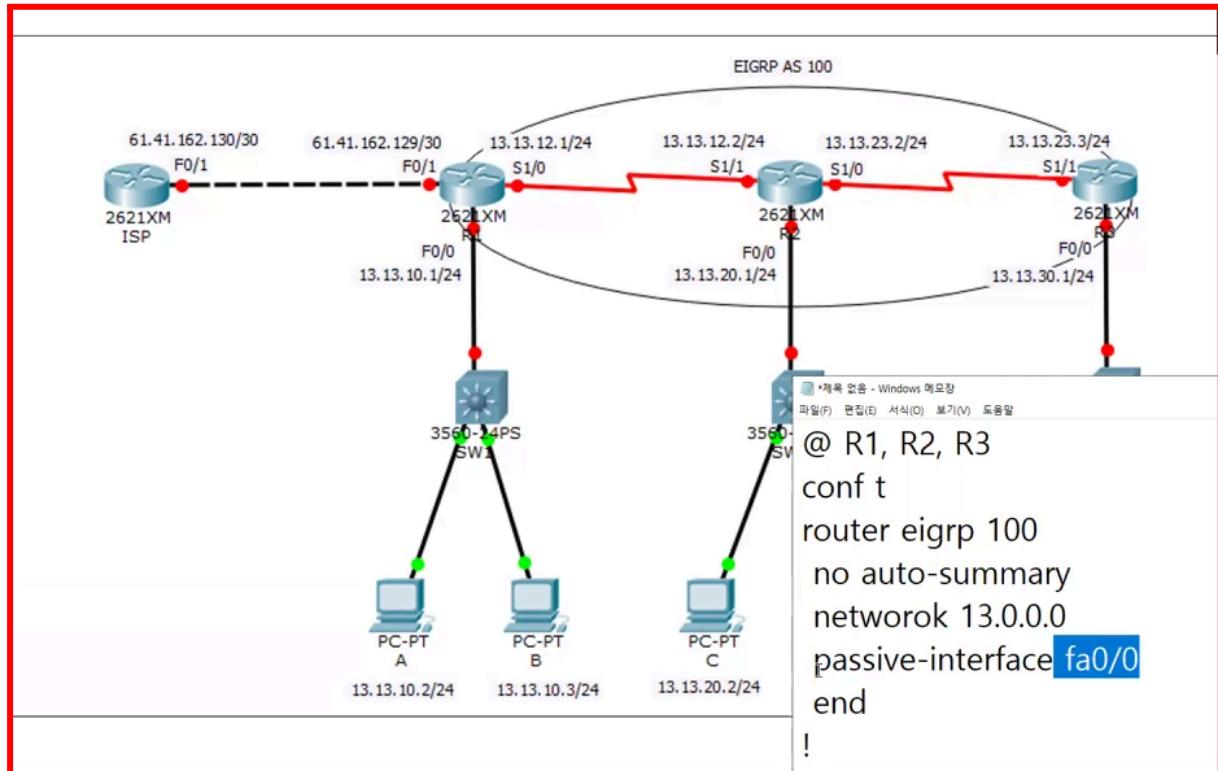
- AS 번호: 망 식별 번호 (국가 코드 같은 개념)
- 번호가 같지 않으면 라우팅 업데이트가 안된다.
- A ~ B 클래스별 설정 방법이 다르다.
- 이 네트워크는 자기 자신을 설정해서 다른 라우터에 뿌리는 개념이다.

예제)

'13-1.EIGRP 라우팅 프로토콜.pkt' 파일을 실행하여 기본 설정을 실시하고 EIGRP 라우팅 프로토콜을 설정한다.



위 예제에서는 IP 대역이 13번대 이기 때문에 설정법이 같다:



```
파일(F) 편집(E) 서식(O) 보기(V) 도움말
@ R1, R2, R3
conf t
router eigrp 100
no auto-summary
network 13.0.0.0
passive-interface fa0/0
end
!
show run
show ip eigrp neighbor
show ip route
!
```

- 명령어
- 밑에꺼로 정보 확인까지 가능하다.

```

R1#
R1#show ip ei
R1#show ip eigrp nei
R1#show ip eigrp neighbors
IP-EIGRP neighbors for process 100
  H   Address           Interface      Hold Uptime      SRTT      RTO      Q      Seq
      (sec)             (ms)          Cnt  Num
  0  13.13.12.2        Se1/0          10  00:06:27    40  1000  0  5

R1#
R1#
R1#

```

- show ip eigrp neighbor
- 연결되어있는 라우터를 보여준다. (R1기준)

```

R1#
R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
      * - candidate default, U - per-user static route, o - ODR
      P - periodic downloaded static route

Gateway of last resort is not set

      13.0.0.0/24 is subnetted, 5 subnets
C        13.13.10.0 is directly connected, FastEthernet0/0
C        13.13.12.0 is directly connected, Serial1/0
D        13.13.20.0 [90/2172416] via 13.13.12.2, 00:06:44, Serial1/0
D        13.13.23.0 [90/2681856] via 13.13.12.2, 00:06:44, Serial1/0
D        13.13.30.0 [90/2684416] via 13.13.12.2, 00:06:42, Serial1/0
R1#

```

- show ip route 경우
- D로 뜨는걸 볼 수 있다.

2. 네이버 성립 및 라우팅 업데이트

- 네이버 관계를 성립하고 라우팅 업데이트를 실시한다.
- 주기적인 전체 라우팅 업데이트를 하지 않고 추가된 부분만 라우팅 업데이트한다.

3. 네이버(neighbor) 성립 조건

- 인접 라우터와 AS 번호와 K 상수값이 동일해야한다.

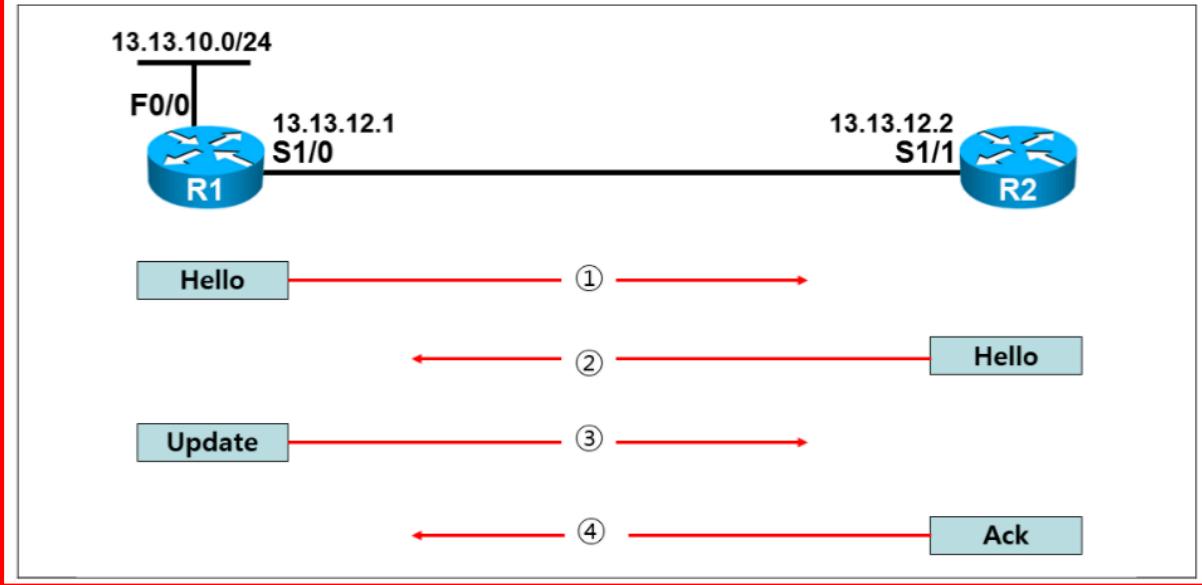
```

R1#show ip protocols
Routing Protocol is "eigrp 100"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
~ 중간 생략 ~

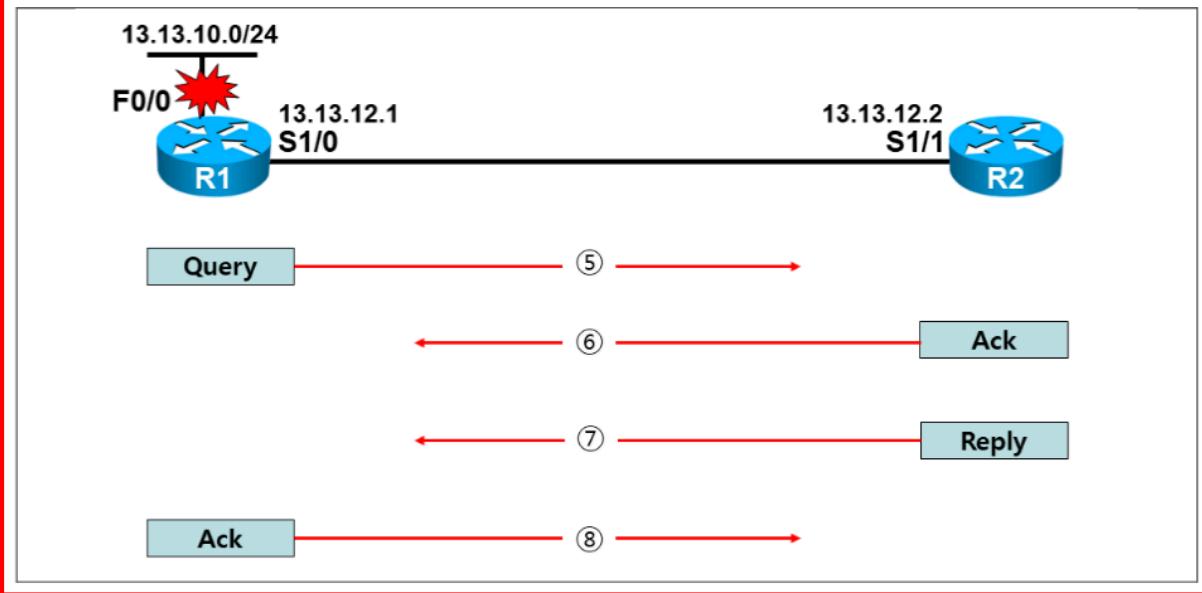
```

4. EIGRP 동작 과정

EIGRP는 Hello 패킷을 교환하여 네이버를 성립하고 라우팅 업데이트를 실시한다.



네이버에게 라우팅 업데이트를 실시한 로컬 네트워크에 장애가 발생하면 Query 패킷을 전송하여 도달 불가능한 정보를 알려주면서 대체 경로 질의한다. Query 패킷을 수신한 네이버 라우터는 Reply 패킷을 이용하여 도달 가능한 경로 정보 또는 도달 불가능한 정보를 응답한다.



5. EIGRP 패킷 유형

유형	내용
Hello	네이버 관계를 성립하기 위해서 교환하며, 주기적인 교환으로 네이버 관계를 유지한다.
Update	네이버 관계를 성립한 라우터 간에 라우팅 업데이트할 때 전송한다.
Query	EIGRP 네트워크 장애 발생시, 네이버에게 도달 불가능한 정보 및 대체 경로를 질의한다.
Reply	Query 패킷에 대한 도달 가능한 경로 정보 또는 도달 불가능한 정보를 응답한다.
Ack	Update, Query, Reply 패킷을 수신하면 Ack 패킷을 전송하여 수신 확인을 알린다.

6. EIGRP 토플로지 테이블

EIGRP에 포함된 네트워크 및 EIGRP 경로를 관리하는 테이블이다. 토플로지 테이블 등록된 경로 중에 최적 경로를 선출하여 라우팅 테이블에 등록한다.

R1#**show ip eigrp topology**

IP-EIGRP Topology Table for AS(100)/ID(13.13.12.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,

r - reply Status, s - sia Status

~ 중간 생략 ~

P 13.13.30.0/24, 1 successors, FD is 2684416

via 13.13.12.2 (2684416/2172416), Serial1/0

- P : Passive 상태, 경로 계산이 완료된 상태(라우팅 테이블 등록된 상태)
- A : Active 상태, 경로 계산이 진행되는 상태(라우팅 테이블 등록 및 삭제 불가능 상태)
- 13.13.30.0/24 : 목적지 네트워크 이름
- successors : 최적 경로 상에 네이버 라우터(최적 경로)
- FD is 2684416 : 로컬 라우터에서 목적지까지 EIGRP 메트릭
- via 13.13.12.2 : 경유하는 라우터(넥스트-홉 라우터)
- (2684416/ : FD 메트릭, 로컬 라우터에서 목적지까지 EIGRP 메트릭
- /2172416) : AD 메트릭, 네이버 라우터에서 목적지까지 EIGRP 메트릭
- Serial1/0 : 라우팅에 의해서 패킷을 출력하는 인터페이스

|—FD(2684416)—|

|—AD(2172416)—|

R1[S1/0]-----[S1/1] R2[S1/0]-----[S1/1]R3[F0/0]-----| 13.13.30.0/24

7. EIGRP Dual 알고리즘

EIGRP 토플로지 정보를 기반으로 경로를 선출하는 알고리즘이며, EIGRP 경로 선출 과정은 다음과 같다.

- ① FD 메트릭이 가장 작은 경로를 최적 경로로 선출한다.
- ② 최적 경로의 FD 메트릭보다 AD 메트릭이 작은 경로를 후속 경로로 선출한다.

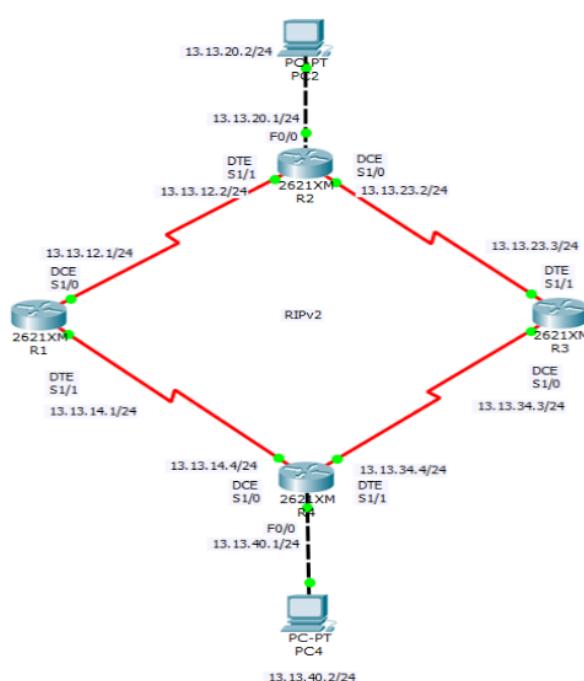
Ex) 다음 중 후속 경로가 있는 경우는 몇번인가?

Router#show ip eigrp topology all-link

- ① P 13.13.30.0/24, 1 successors, FD is 2809856
 - via 13.13.12.2 (2809856/2297856), Serial1/0
 - via 13.13.13.3 (2993144/2809856), Serial1/1
- ② P 13.13.30.0/24, 1 successors, FD is 2809856
 - via 13.13.12.2 (2809856/2297856), Serial1/0
 - via 13.13.13.3 (2993144/1809856), Serial1/1
- ③ P 13.13.30.0/24, 1 successors, FD is 2809856
 - via 13.13.12.2 (2809856/2297856), Serial1/0
 - via 13.13.13.3 (3193144/2909856), Serial1/1
- ④ P 13.13.30.0/24, 2 successors, FD is 2809856
 - via 13.13.12.2 (2809856/2297856), Serial1/0
 - via 13.13.13.3 (2809856/2297856), Serial1/1

8. 균등 로드 분산

목적지 네트워크에 대한 경로의 메트릭이 동일한 경우 자동으로 구현된다.



9. 비균등 로드 분산

대기하고 있는 후속 경로를 라우팅 테이블에 등록하여, 최적 경로와 후속 경로를 동시에 사용할 수 있다.

R2-R3-R4 구간의 대역폭을 다음과 같이 2048kbps로 변경한다.

10. EIGRP 메트릭

EIGRP 메트릭은 'Bandwidth'와 'Delay' 값을 기반으로 계산한다.

메트릭 요소	내용
Vector 메트릭	MTU, Bandwidth, Delay, reliability, load
K 상수	K1=1, K2=0, K3=1, K4=0, K5=0

11. 수동 요약 기능

라우팅 업데이트시 요약 경로만 업데이트하여 라우팅 테이블의 등록된 경로 개수를 최적화시킬 수 있다.

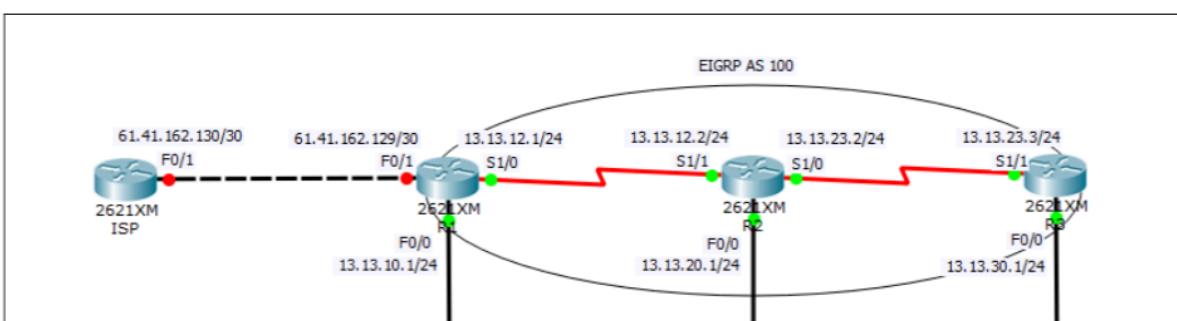
```
R1#conf t
R1(config)#int lo 1
R1(config-if)#ip address 128.28.8.1 255.255.255.0
R1(config-if)#int lo 2
R1(config-if)#ip address 128.28.9.1 255.255.255.0
R1(config-if)#int lo 3
R1(config-if)#ip address 128.28.10.1 255.255.255.0
R1(config-if)#int lo 4
R1(config-if)#ip address 128.28.11.1 255.255.255.0
R1(config-if)#int lo 5
R1(config-if)#ip address 128.28.12.1 255.255.255.0
R1(config-if)#
R1(config-if)#router eigrp 100
R1(config-router)#network 128.28.0.0
R1(config-router)#passive-interface lo1
R1(config-router)#passive-interface lo2
R1(config-router)#passive-interface lo3
R1(config-router)#passive-interface lo4
R1(config-router)#passive-interface lo5
R1(config-router)#end
R1#
```

12. EIGRP 외부 경로

경로 유형	내용
D	동일한 AS 안에서 라우팅 업데이트를 실시한 EIGRP 내부 경로이며 신뢰도는 90 이다.
D EX	다른 환경에서 라우팅 업데이트를 실시한 EIGRP 외부 경로이며 신뢰도는 170 이다.

```
R3#conf t
R3(config)#int lo 1
R3(config-if)#ip address 100.100.1.1 255.255.255.0
R3(config-if)#int lo 2
R3(config-if)#ip address 100.100.2.1 255.255.255.0
R3(config-if)#int lo 3
R3(config-if)#ip address 100.100.3.1 255.255.255.0
R3(config-if)#
R3(config-if)#router rip
R3(config-router)#version 2
R3(config-router)#no auto-summary
R3(config-router)#network 100.0.0.0
R3(config-router)#
R3(config-router)#router eigrp 100
R3(config-router)#redistribute rip metric 1544 2000 255 1 1500
R3(config-router)#end
R3#
```

13. EIGRP 기본 경로 라우팅 업데이트



```
R1#conf t
R1(config)#int fa0/1
R1(config-if)#ip address 61.41.162.129 255.255.255.252
R1(config-if)#no shutdown
R1(config-if)#
R1(config-if)#ip route 0.0.0.0 0.0.0.0 61.41.162.130
R1(config)#
R1(config)#router eigrp 100
R1(config-router)#redistribute static // 기본 경로를 EIGRP로 라우팅 업데이트하는 명령어
R1(config-router)#end
R1#
```

14. EIGRP 삭제

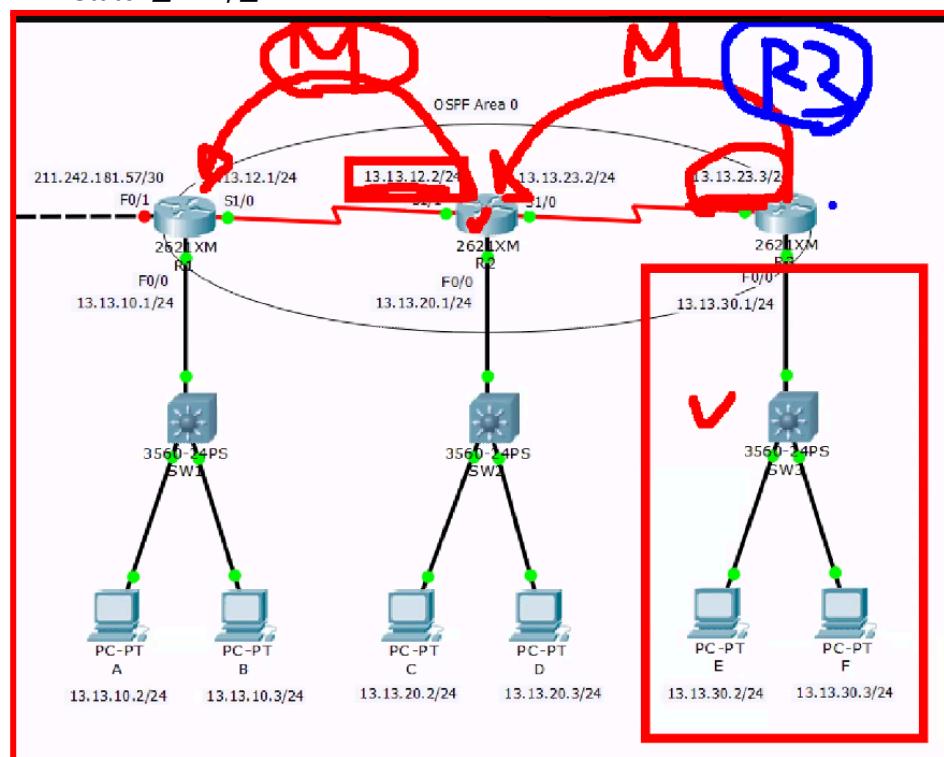
```
R1,R2,R3#conf t
R1,R2,R3(config)#no router eigrp 100
R1,R2,R3(config)#end
R1,R2,R3#
```

제6장 OSPF 라우팅 프로토콜

모든 장비에서 거의 다 OSPF를 쓴다. (미국 등)

1. OSPF(Open Shortest Path First)

a. Link-State 알고리즘



- 30번 네트워크는 R3에 있다는걸 알려준다.
- 또한 몇번 라우터에 연결되어 있다는걸 알려준다.
- 메트릭, 넥스트홉 외에 이걸 알려준다는거에 차이가 있다.

b. Classless Routing Protocol

- i. subnet mask 확인 후 몇 비트 자리인지 알 수 있다.
- ii. 별로 중요한 기능은 아니다.

13.13.10.0/24 <- A 클래스

13.13.10.0/24 <- 13.13.10.0/24

- 확실하게 알려준다.

- c. VLSM, CIDR
- d. IGP
 - 속도는 빠르지만 많은 양의 업데이트가 불가능하다.
- e. SPF 알고리즘을 사용하는 개방된 라우팅 프로토콜
- f. SPF를 사용하는 개방형 알고리즘이다.

2. 라우터 아이디(Router-ID)

- OSPF 라우터를 구분하기 위한 식별자이다
- 아이디 형식은 IPv4 주소 형식을 사용하며 라우터 아이디가 중복되면 네이버를 성립하지 않는다

1) 물리적 인터페이스를 이용한 라우터 아이디 선출

물리적 인터페이스만 있을 경우, 인터페이스 중에 IP 주소가 가장 높은 IP 주소로 선출한다.

```
F0/0 : 13.13.10.1  
S1/0 : 13.13.12.1 // 라우터 아이디로 선출됨
```

2) Loopback 인터페이스를 이용한 라우터 아이디 선출

Loopback 인터페이스가 있을 경우, Loopback 인터페이스 중에 IP 주소가 가장 높은 IP 주소로 선출한다.

```
F0/0 : 13.13.10.1  
S1/0 : 13.13.12.1  
Lo1 : 11.11.11.11 // 라우터 아이디로 선출됨
```

- 단, 물리적 인터페이스 및 Loopback 인터페이스가 Down 상태인 인터페이스의 IP 주소로는 산출하지 않는다.

3) 'router-id' 명령어를 이용한 수동 선출

```
R1(config)#router ospf 1  
R1(config-router)#router-id 1.1.1.1 // 라우터 아이디로 선출됨
```

- 하지만 모든 범위가 되지는 않는다 (VALID 범위 내에서 가능)

Ex) R1에서 OSPF를 설정한 경우, 라우터 아이디는 어떻게 되는가?

```
R1#show ip int brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	unassigned	YES	unset	administratively down	down
FastEthernet0/1	13.13.23.3	YES	DHCP	up	up
Serial1/0	211.241.22.1	YES	maunal	down	down
Serial1/1	unassigned	YES	unset	administratively down	down
Serial1/2	unassigned	YES	unset	administratively down	down
Serial1/3	unassigned	YES	unset	administratively down	down
Loopback0	13.13.3.3	YES	manual	administratively down	down

- FastEthernet0/1 13.13.23.3 YES DHCP up up

```

!
router ospf 100
  router-id 10.6.15.201
  log-adjacency-changes
  area 10 nssa +
  passive-interface f9/1
  network 10.6.15.248 0.0.0.3 area 10
  network 10.6.15.201 0.0.0.0 area 10
  network 10.7.10.208 0.0.0.3 area 10
  network 10.7.10.212 0.0.0.3 area 10
  network 10.7.22.255 0.0.0.63 area 10
}

```

4. management link 설정
- BEQ-01 IGP연동

- 대부분 이렇게 직접 router-id를 설정한다.

3. OSPF 설정

1) OSPF 라우팅 프로토콜 설정 방법

```

Router(config)#router ospf [1~65535 Process-ID]
Router(config-router)#router-id [IPv4 주소 형식]
Router(config-router)#network [로컬 네트워크] [와일드카드 마스크] area [0-4294967295 area 주소]
Router(config-router)#end

```

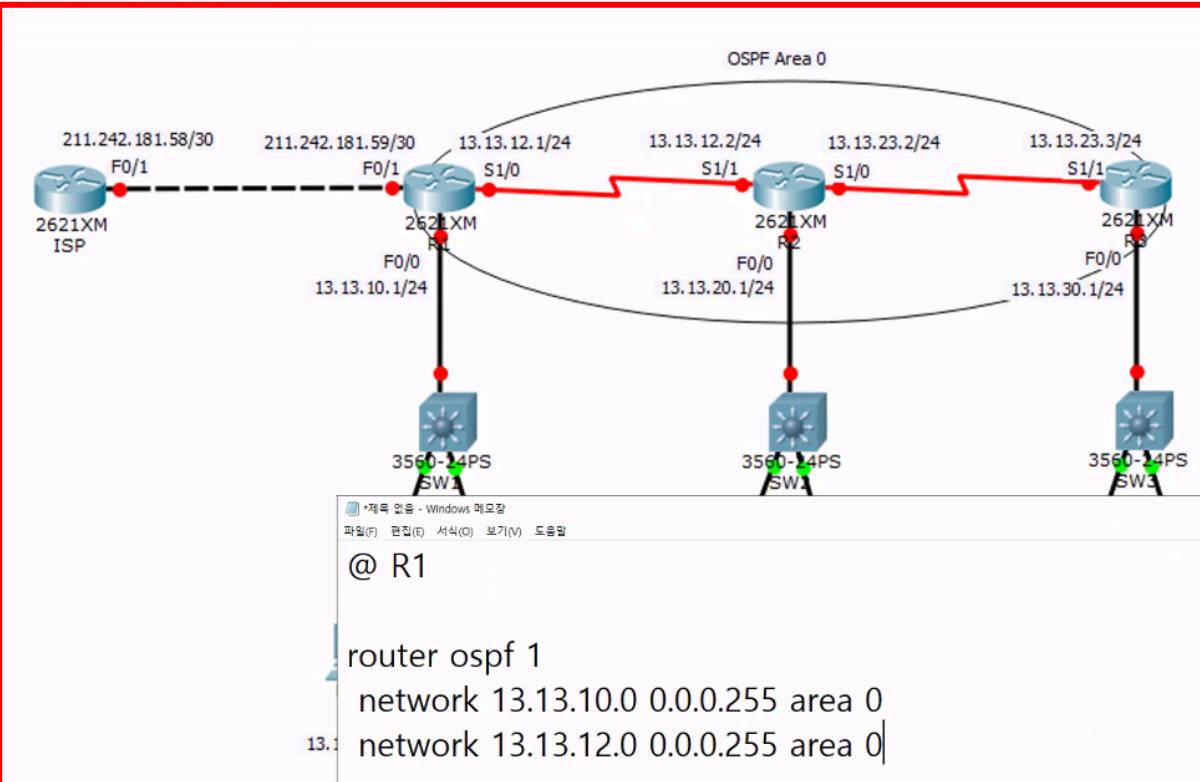
- process id는 ospf를 여러개를 설정할 수 있게 도와준다. 또한 업데이트를 지역별로 할수 있게 도와준다.
- OSPF 프로세스 아이디를 지원하기 때문에 라우터에 여러 개의 OSPF 라우팅 프로토콜을 설정할 수 있다
- area 주소 - ospf에서 사용하는 주소

와일드카드 마스크 vs 서브넷 마스크:

서브넷 마스크	와일드카드 마스크
255.255.255.255	0.0.0.0
255.255.255.0	0.0.0.255
255.255.0.0	0.0.255.255
255.0.0.0	0.255.255.255
0.0.0.0	255.255.255.255
255.255.255.252	0.0.0.3
255.255.255.224	0.0.0.31
255.255.248.0	0.0.7.255

- 0과 1을 반대로 쓰면된다.

2) R1, R2, R3 OSPF 라우팅 프로토콜 설정



- 정확하게 OSPF에 해당되는 IP, area를 넣어야 한다.
- 필요할때 지역분배가 가능하다.

ex) R1

R1, R2, R3에서 각각의 라우터 아이디를 '1.1.1.1', '2.2.2.2', '3.3.3.3'으로 하여 OSPF Area 0 환경을 구성한다.

```

R1#conf t
R1(config)#router ospf 1
R1(config-router)#router-id 1.1.1.1
R1(config-router)#network 13.13.10.0 0.0.0.255 area 0
R1(config-router)#network 13.13.12.0 0.0.0.255 area 0
R1(config-router)#end
R1#

```

```

R1#show run
~ 중간 생략 ~
!
router ospf 1
router-id 1.1.1.1
log-adjacency-changes
network 13.13.10.0 0.0.0.255 area 0
network 13.13.12.0 0.0.0.255 area 0

```

ex) R2

```
R2#conf t
R2(config)#router ospf 1
R2(config-router)#router-id 2.2.2.2
R2(config-router)#network 13.13.12.0 0.0.0.255 area 0
R2(config-router)#network 13.13.20.0 0.0.0.255 area 0
R2(config-router)#network 13.13.23.0 0.0.0.255 area 0
R2(config-router)#end
R2#
```

```
R2#show run
```

```
~ 중간 생략 ~
!
router ospf 1
  router-id 2.2.2.2
  log-adjacency-changes
  network 13.13.12.0 0.0.0.255 area 0
  network 13.13.20.0 0.0.0.255 area 0
  network 13.13.23.0 0.0.0.255 area 0
```

- R3 등 다른 라우터도 이런식으로 설정이 가능하다.

네이버 테이블 확인

3) R1, R2, R3 OSPF 네이버 테이블 확인

```
R1#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
2.2.2.2	0	FULL/ -	00:00:30	13.13.12.2	Serial1/0

```
R2#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
3.3.3.3	0	FULL/ -	00:00:33	13.13.23.3	Serial1/0
1.1.1.1	0	FULL/ -	00:00:31	13.13.12.1	Serial1/1

```
R3#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
2.2.2.2	0	FULL/ -	00:00:39	13.13.23.2	Serial1/1

show ip ospf database router
show ip ospf database

IOS Command Line [Root] New Cluster Move Obj

Logical

Options: (No TOS-capability, DC)
LS Type: Router Links
Link State ID: 3.3.3.3
Advertising Router: 3.3.3.3
LS Seq Number: 80000004
Checksum: 0x8b69
Length: 60
Number of Links: 3

Link connected to: a Stub Network
(Link ID) Network/subnet number: 13.13.30.0
(Link Data) Network Mask: 255.255.255.0
Number of TOS metrics: 0
TOS 0 Metrics: 1

Link connected to: another Router (point-to-point)
(Link ID) Neighboring Router ID: 2.2.2.2
(Link Data) Router Interface address: 13.13.23.3
Number of TOS metrics: 0
TOS 0 Metrics: 64

Link connected to: a Stub Network
(Link ID) Network/subnet number: 13.13.23.0
(Link Data) Network Mask: 255.255.255.0
Number of TOS metrics: 0
TOS 0 Metrics: 64

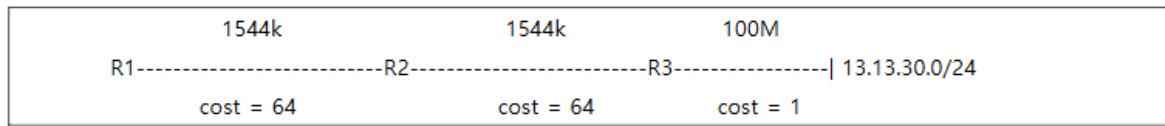
1#

- 이런식으로 R1라우터에서 R3 라우터에 연결되어있는 허브까지 다 알수 있다.

4. OSPF 메트릭

- OSPF 메트릭 단위는 'Cost'이며 'bandwidth' 기반으로 계산한다. 계산식은 다음과 같으며 로컬 라우터에서 목적지까지 Cost 값을 더한 값을 메트릭으로 사용한다
- $\text{Cost} = 10^8 / \text{Bandwidth}$

Ex) R1에서 '13.13.30.0/24'까지 OSPF 메트릭(Cost)은 얼마인가?



5. OSPF 신뢰도

- OSPF 경로의 신뢰도는 '110'으로 설정되어 있다.

R1#show ip route ospf

```
13.0.0.0/24 is subnetted, 5 subnets
O 13.13.20.0 [110/65] via 13.13.12.2, 00:38:04, Serial1/0
O 13.13.23.0 [110/128] via 13.13.12.2, 00:38:04, Serial1/0
O 13.13.30.0 [110/129] via 13.13.12.2, 00:37:51, Serial1/0
```

[참고] 경로 신뢰도

Connected	0
Static	1
EIGRP	90
OSPF	110
RIP	120
EIGRP External	170

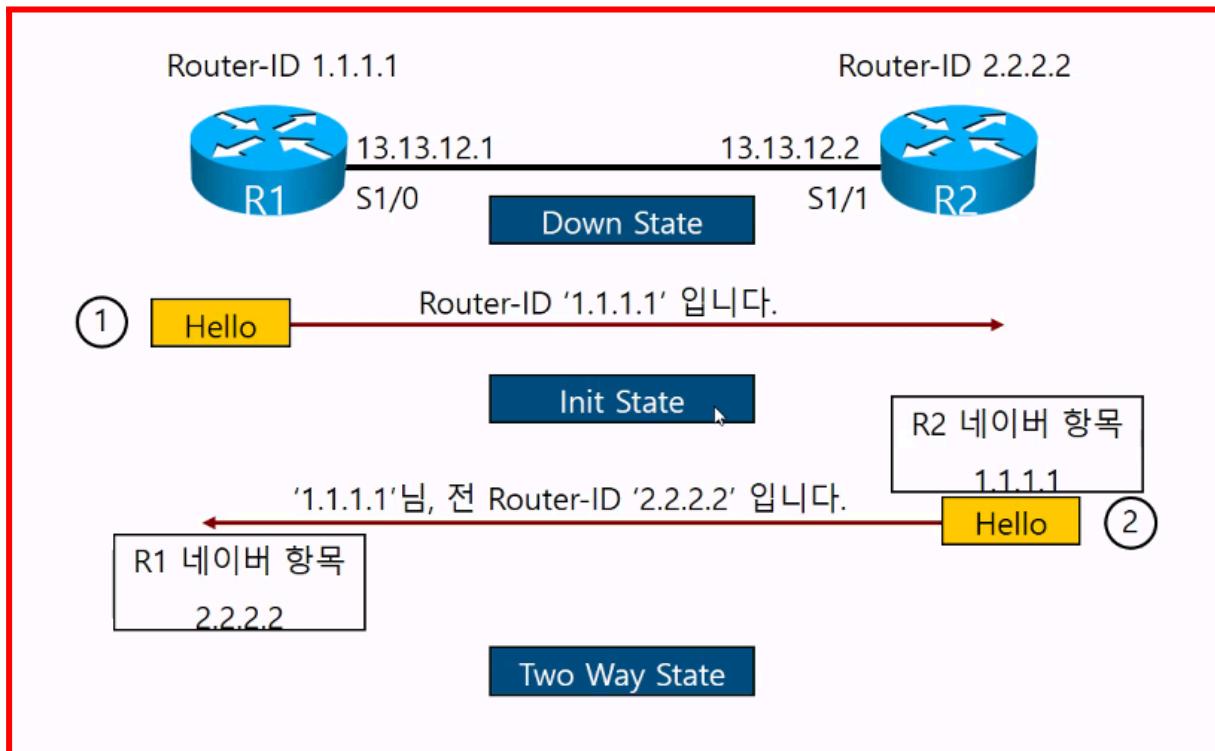
Ex1) 현재 구성된 환경에서 EIGRP 100 라우팅 업데이트 환경을 구성하면 라우팅 테이블에는 어떤 경로가 등록되는가?

- EIGRP로 등록된다 (신뢰도가 더 높다)

Ex2) 현재 구성된 환경에서 RIPv2 라우팅 업데이트 환경을 구성하면 라우팅 테이블에는 어떤 경로가 등록되는가?

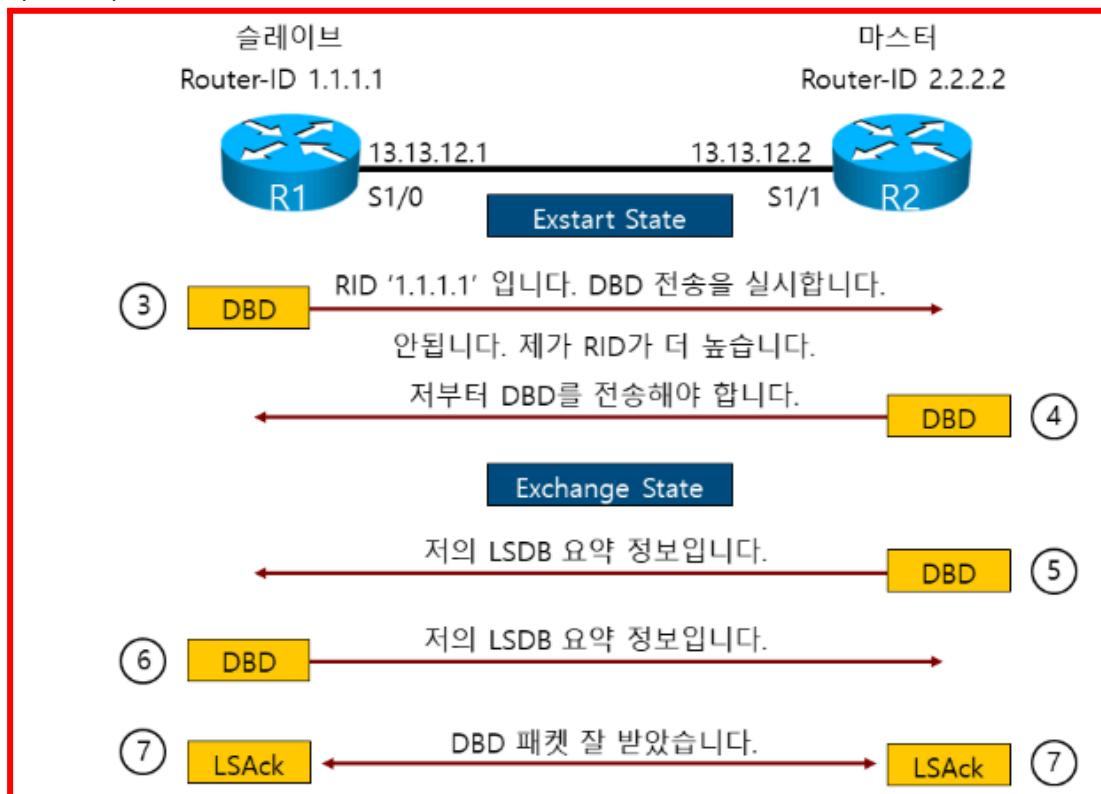
- OSPF로 된다 (신뢰도가 높다)

6. OSPF 동작 과정



- 혼자만 보내면 Init State (자기 옆에 누가 있는지 모른다)
- 쌍방향으로 보내면 Two Way State (자기 옆에 누가 있는지 안다)

다음 단계:



- 라우터 아이디가 더 크면 마스터가 된다.



- 만약에 슬레이브가 3번 라우터에 대한 정보가 없으면 요청을 한다. (loading state)
- 받으면 Full State
- LSR link state reloading
- LSU link state update

7. OSPF 패킷 유형

```
R1#
```

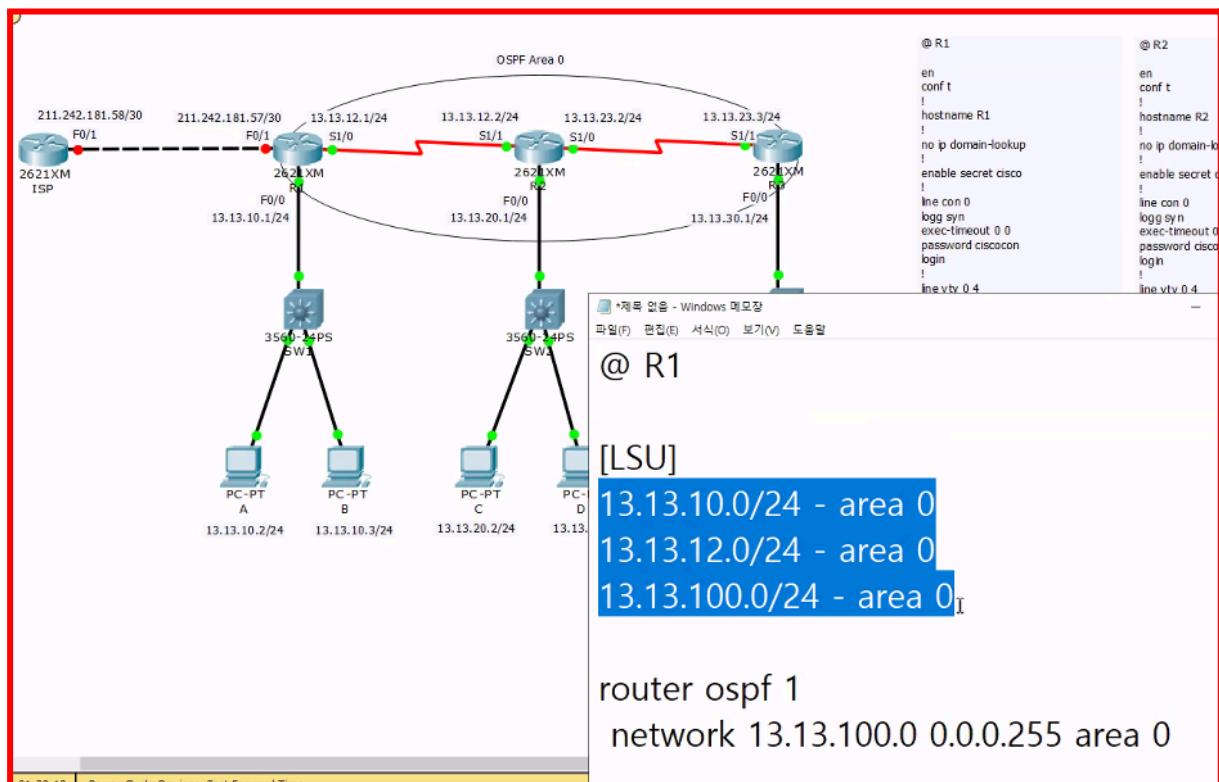
```
R1#
```

```
01:18:06: OSPF: DR/BDR election on FastEthernet0/0
01:18:06: OSPF: Elect BDR 0.0.0.0
01:18:06: OSPF: Elect DR 1.1.1.1
01:18:06: DR: 1.1.1.1 (Id) BDR: none
01:18:06: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Serial1/0 from FULL to DOWN, Neighbor Down:
Adjacency forced to reset
01:18:06: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Serial1/0 from FULL to DOWN, Neighbor Down:
Interface down or detached
01:18:06: OSPF: Build router LSA for area 0, router ID 1.1.1.1, seq 0x80000005
01:18:11: OSPF: Send DBD to 2.2.2.2 on Serial1/0 seq 0x5287 opt 0x00 flag 0x7 len 32
01:18:11: OSPF: Rcv DBD from 2.2.2.2 on Serial1/0 seq 0x3b97 opt 0x00 flag 0x7 len 32  mtu 1500 state
EXSTART
01:18:11: OSPF: NBR Negotiation Done. We are the SLAVE
01:18:11: OSPF: Send DBD to 2.2.2.2 on Serial1/0 seq 0x3b97 opt 0x00 flag 0x2 len 32
01:18:11: OSPF: Rcv DBD from 2.2.2.2 on Serial1/0 seq 0x3b98 opt 0x00 flag 0x3 len 92  mtu 1500 state
EXCHANGE
01:18:11: OSPF: Send DBD to 2.2.2.2 on Serial1/0 seq 0x3b98 opt 0x00 flag 0x0 len 32
01:18:11: OSPF: Rcv DBD from 2.2.2.2 on Serial1/0 seq 0x3b99 opt 0x00 flag 0x1 len 32  mtu 1500 state
```

EXCHANGE

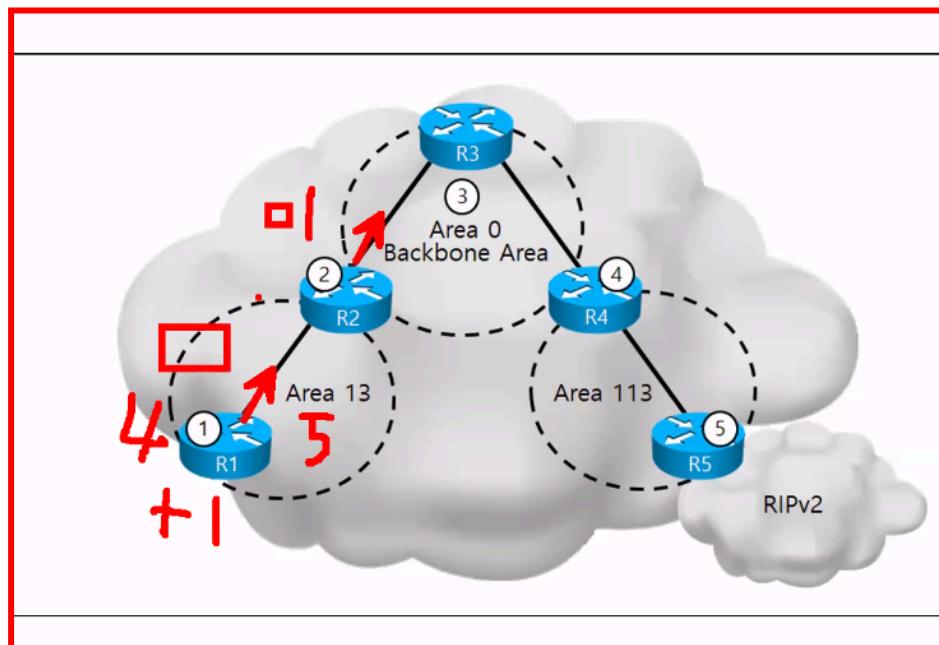
```
01:18:11: OSPF: Send DBD to 2.2.2.2 on Serial1/0 seq 0x3b99 opt 0x00 flag 0x0 len 32
01:18:11: Exchange Done with 2.2.2.2 on Serial1/0
01:18:11: OSPF: Database request to 2.2.2.2
01:18:11: OSPF: sent LS REQ packet to 224.0.0.5, length 36
01:18:11: OSPF: Send DBD to 2.2.2.2 on Serial1/0 seq 0x3b99 opt 0x00 flag 0x0 len 32
01:18:11: OSPF: Build router LSA for area 0, router ID 1.1.1.1, seq 0x80000005
01:18:11: Synchronized with 2.2.2.2 on Serial1/0, state FULL
01:18:11: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Serial1/0 from LOADING to FULL, Loading Done
01:18:11: OSPF: Build router LSA for area 0, router ID 1.1.1.1, seq 0x80000006
```

8. OSPF Area 설계



- OSPF에서는 같은 area 안에서 업데이트 진행시(라우터 추가 등) 모든 추가된 ospf 를 업데이트 시킨다.
- 추가된것만 넘기는게 아니라 전체를 다 넘긴다.

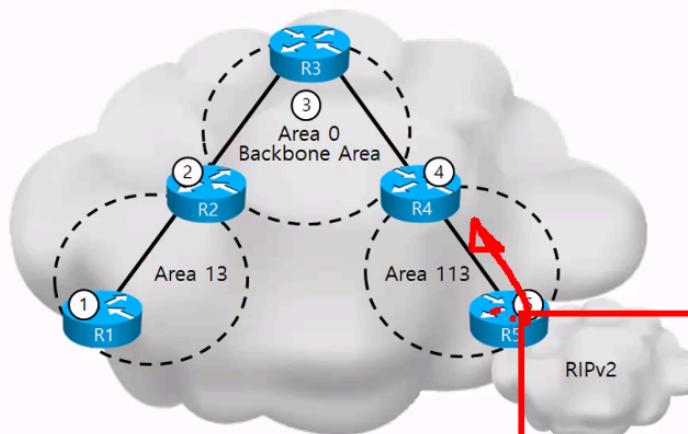
다중 Area 설계:



- 모든 정보가 불필요하게 업데이트 되는것을 막기 위해 모든 Area 는 Backbone Area 를 담당하는 Area 0 을 경유하도록 설계해야 한다.
- 이 경유하는 위치의 라우터를 ABR 이라고 한다.

ASBR

다중 Area 요소	내용
Backbone Area	모든 Area 가 경유하는 Area 이며, Area 0 이 담당한다.
ABR	Area 0 과 다른 Area 경계 사이에 위치한 라우터이다.
ASBR	외부 네트워크 정보를 OSPF 환경으로 라우팅 업데이트하는 라우터이다.



- LSU 패킷 사이즈를 줄이기 위해서 이런 설계를 한다.

9. OSPF 테이블 유형

1) 네이버 테이블

네이버 관계를 성립한 인접 라우터와의 상태 정보를 관리한다. 네이버 관계가 해지되면 네이버 테이블에 등록된 인접 라우터의 정보는 삭제된다.

R1#show ip ospf neighbor

① Neighbor ID	② Pri	③ State	④ Dead Time	⑤ Address	⑥ Interface
2.2.2.2	0	FULL/ -	00:00:30	13.13.12.2	Serial1/0

- ① 네이버 관계를 성립한 인접 라우터 R2의 라우터 아이디이다.
- ② 네이버 라우터 R2의 OSPF 우선 순위 값이다.
- ③ 네이버 라우터 R2와의 상태 정보와 DR/BDR 선출 내용을 의미한다. 현재 DR/BDR 선출이 없는 상태이다.
- ④ Hello 패킷을 수신하지 못하면 네이버를 해지하는 Dead 타이머이다.
- ⑤ 네이버 라우터 R2의 IP 주소이다.
- ⑥ 네이버 라우터 R2와 연결된 인터페이스이다.

2) 데이터베이스 테이블(Link-State Database)

OSPF 링크-상태 정보를 관리하며 SPF 알고리즘을 이용하여 최적 경로를 선출한다

R1#show ip ospf database

OSPF Router with ID (1.1.1.1) (Process ID 1)					
Router Link States (Area 0)					
Link ID	ADV Router	Age	Seq#	Checksum	Link count
1.1.1.1	1.1.1.1	29	0x80000006	0x00b678	3
2.2.2.2	2.2.2.2	22	0x80000008	0x0060ae	5
3.3.3.3	3.3.3.3	21	0x80000006	0x00876b	3

R1 Area 0 링크(3 개)	R2 Area 0 링크(5 개)	R3 Area 0 링크(3 개)
13.13.10.0/24	13.13.12.0/24	13.13.23.0/24
13.13.12.0/24	13.13.23.0/24	13.13.30.0/24
R2 와 연결된 S1/0 주소 13.13.12.1	R3 와 연결된 S1/0 주소 13.13.23.2 R1 과 연결된 S1/1 주소 13.13.12.2	R2 와 연결된 S1/1 주소 13.13.23.3

10. OSPF 경로 유형

OSPF는 Area 지역 내부 또는 외부에 따라서 광고하는 LSA 광고 유형이 다르며, 데이터베이스에서 각각 별도로 관리하기 때문에 경로 유형이 다양하다.

이름	코드	내용
Intra-Area 경로	O	같은 Area 네트워크
Inter-Area 경로	O IA	다른 Area 네트워크
External 경로	O E1, O E2	외부 네트워크

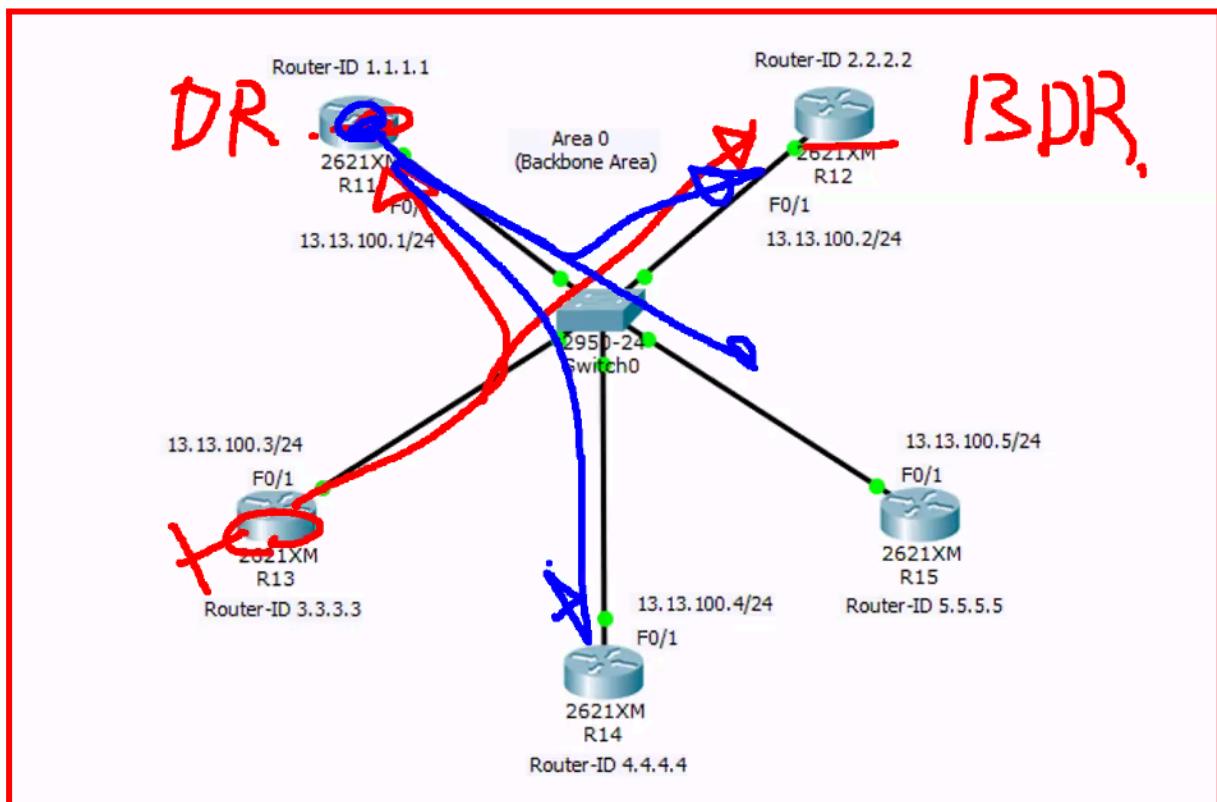
R3에서 다음과 같은 설정을 실시하면 Area 13 네트워크 정보를 Area 0으로 광고하므로 ABR 라우터를 수행하며 RIPv2 외부 네트워크 정보를 광고하므로 ASBR 라우터를 수행한다.

정보확인을 위한 코드

```
Gateway of last resort is not set
[SPF] 13.0.0.0/24 is subnetted, 5 subnets
도 [O] 13.13.10.0 [110/65] via 13.13.12.1, 01:27:06, Serial1/1
C 13.13.12.0 is directly connected, Serial1/1
C 13.13.20.0 is directly connected, FastEthernet0/0
C 13.13.23.0 is directly connected, Serial1/0
O 13.13.30.0 [110/65] via 13.13.23.3, 01:28:13, Serial1/0
    100.0.0.0/24 is subnetted, 3 subnets
O E2 100.100.1.0 [110/20] via 13.13.23.3, 00:19:30, Serial1/0
O E2 100.100.2.0 [110/20] via 13.13.23.3, 00:19:30, Serial1/0
O E2 100.100.3.0 [110/20] via 13.13.23.3, 00:19:30, Serial1/0
    200.200.1.0/32 is subnetted, 1 subnets
O IA 200.200.1.1 [110/65] via 13.13.23.3, 00:19:20, Serial1/0
    200.200.2.0/32 is subnetted, 1 subnets
O IA 200.200.2.1 [110/65] via 13.13.23.3, 00:19:20, Serial1/0
    200.200.3.0/32 is subnetted, 1 subnets
O IA 200.200.3.1 [110/65] via 13.13.23.3, 00:19:20, Serial1/0
R2#
```

- O는 같은 네트워크
- O IA는 다른 AREA 네트워크
- O E2는 외부 네트워크

11. DR/BDR 선출



- 반장/부반장 이라고 생각하면 쉽다.
- 하나의 네트워크에 여러개의 라우터를 넣는 설계의 경우 선출
- DR 하나만 설정하면 DR 라우터가 나머지 라우터한테 다 광고해준다.

DR 선출 과정은 다음과 같다.

- ① OSPF 우선 순위 값이 큰 라우터를 DR로 선출하고 두 번째 라우터를 BDR로 선출한다.
- ② 우선 순위 값이 동일하면 라우터 아이디 값이 큰 라우터를 DR로 선출하고 두 번째 라우터를 BDR로 선출한다.
- ③ DR과 BDR 아닌 라우터들은 'DROTHER'라고 한다.

DR과 BDR은 다음과 같이 OSPF 우선 순위 값을 조정하여 수동으로 선출하는 것을 권장한다.

- 우선순위값이 같으면 라우터 아이디가 가장 큰 라우터를 DR로 설정한다.
- 우선순위값을 변경해서 원하는 위치의 장비를 DR, BDR로 변경 가능하다.

DR에 장애가 생겨서 BDR이 DR이 된다

>> 장애복구 >>

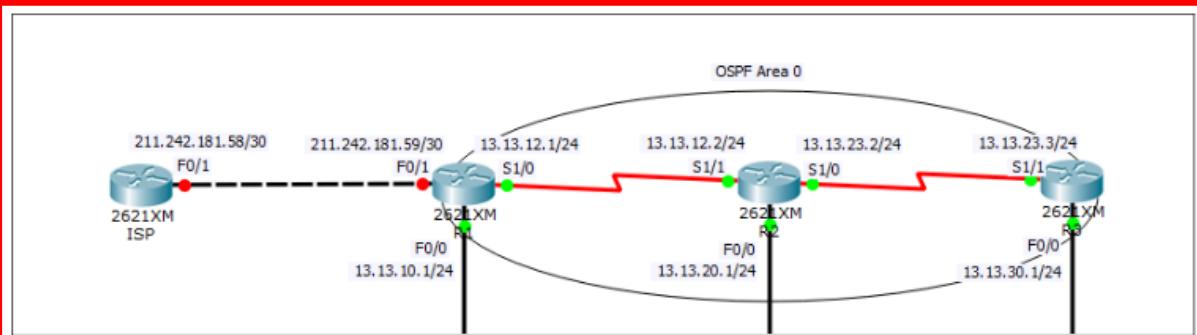
아직 BDR >> DR된 라우터가 DR이 된다.

다시 DR에 장애가 생긴다

>> 장애복구 >>

BDR이 DR이 된다.

12. OSPF 기본 경로 라우팅 업데이트



```
R1#conf t
R1(config)#int fa0/1
R1(config-if)#ip address 211.242.181.57 255.255.255.252
R1(config-if)#no shutdown
R1(config-if)#
R1(config-if)#ip route 0.0.0.0 0.0.0.0 211.242.181.58
R1(config)#
R1(config)#router ospf 1
R1(config-router)#default-information originate // 기본 경로를 OSPF로 라우팅 업데이트하는 명령어
R1(config-router)#end
```

13. OSPF 삭제

```
R1,R2,R3#conf t
R1,R2,R3(config)#no router ospf 1
R1,R2,R3(config)#end
R1,R2,R3#
```

제7장 ACL (Access Control List-접근 제어 항목)

1. ACL

2. 와일드카드 마스크

- 서브넷마스크와 다르게 0과 1이 반대로 되어있다.
- 규칙이 없는 마스크

ACL은 와일드카드 마스크를 이용하여 IP 주소 및 서브넷을 정의한다. 와일드카드 마스크는 서브넷 마스크처럼 '1'이 연속되어야 하는 규칙이 없는 마스크이기 때문에 서브넷 마스크로 간결하게 정의할 수 없는 서브넷 또는 IP 주소들을 간결하게 설정할 수 있다.

비트 처리	서브넷 마스크	와일드카드 마스크
크기	32bit	32bit
공통비트	1	0
비공통 비트	0	1
규칙	1이 연속되어야 함	규칙이 없음

서브넷 마스크 → 와일드카드 마스크

ex1)

255.255.255.255	0.0.0.0
255.255.255.0	0.0.0.255
255.255.0.0	0.0.255.255
255.0.0.0	0.255.255.255
0.0.0.0	255.255.255.255
255.255.255.252	0.0.0.3
255.255.255.248	0.0.0.7
255.255.255.240	0.0.0.15
255.255.255.224	0.0.0.31
255.255.255.128	0.0.0.127
255.255.254.0	0.0.1.255
255.255.240.0	0.0.15.255

Ex2) 192.168.1.0/24 ~ 192.168.255.0/24 중에 3 번째 옥텟이 홀수인 서브넷을 한줄로 설정하여라

192.168.0000000 1.0
192.168.0000001 1.0
192.168.0000010 1.0
~
192.168.1111111 1.0
-----> 192.168.1.0 0.0.254.255
0. 0.1111111 0.255 <- 0.0.254.255

Ex3) 192.168.1.0/24 ~ 192.168.255.0/24 중에 3 번째 옥텟이 짹수인 서브넷을 한줄로 설정하여라

```
192.168.0000001 0.0  
192.168.0000010 0.0  
192.168.0000011 0.0  
~  
192.168.1111111 0.0  
-----> 192.168.0.0 0.0.254.255  
0. 0.1111111 0.255 <- 0.0.254.255
```

Ex4) 192.168.112.32 ~ 192.168.112.63 IP 주소를 한줄로 설정하여라.

```
192.168.112.001 00000  
192.168.112.001 00001  
192.168.112.001 00010  
~  
192.168.112.001 11111  
-----> 192.168.112.32 0.0.0.31  
0. 0. 0.000 11111 <- 0.0.0.31
```

Ex5) A 클래스 IP 주소를 한줄로 설정하여라.

```
0.0.0.0 ~ 127.255.255.255  
0 0000000. 0 1111111.  
0 1111111.  
  
0.0.0.0 127.255.255.255
```

Ex6) B 클래스 IP 주소를 한줄로 설정하여라.

```
128.0.0.0 ~ 191.255.255.255  
10 000000. 10 111111.  
00 111111.  
  
128.0.0.0 63.255.255.255
```

Ex7) C 클래스 IP 주소를 한줄로 설정하여라.

192.0.0.0 ~ 223.255.255.255
110 00000. ¹110 11111.
000 11111.

|192.0.0.0 31.255.255.255

Ex8) A 클래스 사설 IP 주소를 한줄로 설정하여라.

0.0.0.0 ~ 127.255.255.255
0 0000000. 0 1111111.
0 1111111.

0.0.0.0 127.255.255.255

Ex9) B 클래스 사설 IP 주소를 한줄로 설정하여라

172.16.0.0 ~ 172.31.255.255

172.0001 0000.0.0

172.0001 0001.0.0

172.0001 0010.0.0

~

172.0001 1111.0.0

-----> 172.16.0.0 0.15.255.255

0.0000 1111.255.255 <- 0.15.255.255¹

Ex10) C 클래스 사설 IP 주소를 한줄로 설정하여라.

192.168.0.0 ~ 192.168.255.255

192.168.0.0 0.0.255.255

Ex11) IP 주소 전체를 학줄로 설정하여라.

0.0.0.0 255.255.255.255 -> any

Ex12) 13.13.10.100 IP 주소 1 개를 설정하여라

13.13.10.100 0.0.0.0 -> host 13.13.10.100

Ex13) 199.172.1.0/24, 199.172.3.0/24 를 학줄로 설정하여라

199.172.000000 0 1.0
199.172.000000 1 1.0
-----> 199.172.1.0 0.0.2.255
0. 0. 000000 1 0.255 ← 0.0.2.255

Ex14) 199.172.1.0/24 ~ 199.172.3.0/24, 199.172.8.0/24 ~ 199.172.11.0/24 를 학줄로 설정하여라

199.172.0000 0 0 01.0
199.172.0000 0 0 10.0
199.172.0000 0 0 11.0
199.172.0000 1 0 00.0
199.172.0000 1 0 01.0
199.172.0000 1 0 10.0
199.172.0000 1 0 11.0
-----> 199.172.0.0 0.0.11.255
0. 0. 0000 1 0 11.255 ← 0.0.11.255

Ex15) 199.172.5.0/24, 199.172.7.0/24, 199.172.10.0/24, 199.172.14.0/24 를 두줄로 설정하여라.

199.172.000001 0 1.0
199.172.000001 1 1.0
-----> 199.172.5.0 0.0.2.255
0. 0. 000000 1 0.255 ← 0.0.2.255

199.172.000001 0 10.0
199.172.000001 1 10.0
-----> 199.172.10.0 0.0.4.255
0. 0. 000000 1 00.255 ← 0.0.4.255

ACL (Access Control List) - 접근 제어 항목

네트워크에서 전송되는 트래픽을 제어하는 것은 보안적인 관점에서 중요한 이유이다. 이때, ACL은 트래픽 필터링과 방화벽을 구축하는데 가장 중요할 요소일 뿐만 아니라 라우팅 환경에서 서브넷과 호스트를 정의하는 경우에도 필요하다

3. ACL 설정시 파악할 요소

- ① 출발지와 목적지를 파악한다.
- ② 패킷을 허용(permit) 할 것인지, 차단(deny) 할 것인지 파악한다.
- ③ ACL를 인바운드로 적용할 것인지, 아웃바운드로 적용할 것인지 파악한다.
- ④ 추가적으로 파악할 요소로는 서비스 유형 및 포트 번호, TCP 플래그 정보, IP Fragments 정보 등이 있다.
- ⑤ ACL은 패킷을 허용하거나 차단할 때, IP 헤더부터 IP 상위 계층 헤더 정보까지만 검사한다.

프로토콜	계층	보안 정책
HTTP, FTP, TELNET....	L7	IDS/IPS(snort, suricata), WAF(웹 애플리케이션 방화벽)
TCP, UDP, ICMP, EIGRP, OSPF....	L4	ACL, Firewall, IDS/IPS(snort, suricata)
IP	L3	ACL, Firewall, IDS/IPS(snort, suricata)

- 1,2,3은 기본적으로 파악해야 한다.
- ACL은 검사할 수 있는 범위가 있다 (L4까지만 검사가 가능하다)
 - IP header, TCP header, UDP, ICMP 까지만 검사를 한다.
 - HTTP 안에 들어있는 건 검사를 못 한다. (L7쪽은 불가능하다)

4. ACL 처리 과정 및 주의 사항

1) 서브넷 범위가 작은 항목부터 설정해야 한다.

13.13.0.0/16	차단
13.13.30.0/24	허용
13.13.0.0/16	허용
13.13.30.0/24	차단

설정된 순서대로 동작을 한다. 따라서 순서가 중요하다.

ex) 범위가 큰 서브넷을 먼저 설정:

```
R1#conf t
R1(config)#access-list 10 permit 13.13.0.0 0.0.255.255
R1(config)#access-list 10 deny 13.13.30.0 0.0.0.255
R1(config)#int s1/0
R1(config-if)#ip access-group 10 in
R1(config-if)#end
R1#show ip access-lists
Standard IP access list 10
    10 permit 13.13.0.0 0.0.255.255
    20 deny 13.13.30.0 0.0.0.255
```

- 범위가 큰 서브넷을 허용하는 설정이 순서 번호 10 번에 있기 때문에 F(13.13.30.3)를 차단하지 않는다.
 - 따라서 범위가 크게 앞에 있으면 안된다. (뒤에 있는 범위까지 닿지 않는다.)
- 10, 20은 검사하는 순서이다. (sequence number, 10→20)
- 설정한 아이피는 출발지이다.

- ACL 을 삭제한다.

```
R1#conf t
R1(config)#no access-list 10
```

ex) 범위가 작은 서브넷을 먼저 설정:

```
R1(config)#access-list 10 deny 13.13.30.0 0.0.0.255
R1(config)#access-list 10 permit 13.13.0.0 0.0.255.255
R1(config)#
R1(config)#end
R1#
*SYS-5-CONFIG_I: Configured from console by console

R1#show ip access-list
Standard IP access list 10
    10 deny 13.13.30.0 0.0.0.255
    20 permit 13.13.0.0 0.0.255.255
R1#
```

- 범위가 작은 서브넷 → 큰 서브넷으로 확장되었음으로 차단된다.

2) ACL 마지막 항목 'deny any' 처리

1. 전체차단하는 경우

```
R1#conf t
R1(config)#access-list 10 deny 13.13.30.0 0.0.0.255
R1(config)#end
R1#show ip access-lists
Standard IP access list 10
 10 deny 13.13.30.0 0.0.0.255
  // 마지막에 'deny any' 처리 실시
```

- 보이지는 않지만 마지막에 모든 아이피를 차단시킨다.

2. 나머지 전체 허용이 필요한 경우

```
R1#conf t
R1(config)#access-list 10 deny 13.13.30.0 0.0.0.255      // 설정되어 있음
R1(config)#access-list 10 permit any                      // 설정 추가
R1(config)#end
R1#show ip access-lists
Standard IP access list 10
  10 deny 13.13.30.0 0.0.0.255 (11 match(es))
  20 permit any
```

- 전체차단을 걸고 허용하는 문구를 써줘야 그 아이피 대역을 제외한 나머지 대역들을 허용시켜준다.
- access-list 10 permit any

3) 항목 부분 추가 및 부분 삭제 불가능

1. 항목 추가 불가능

- 출발지 '13.13.20.x'인 패킷을 차단하는 ACL 설정을 추가한다.

```
R1#conf t
```

```
R1(config)#access-list 10 deny 13.13.20.0 0.0.0.255
```

```
R1(config)#end
```

```
R1#show ip access-lists
```

```
Standard IP access list 10
```

```
 10 deny 13.13.30.0 0.0.0.255 (11 match(es))
```

```
 20 permit any (9 match(es))
```

```
 30 deny 13.13.20.0 0.0.0.255
```

- 30번에 추가가 되었음으로 차단이 되지 않는다 (전 20번에서 다 허용을 시켜버린다 -전체허용)

2. 항목 부분 삭제 불가능

```
R1#conf t
```

```
R1(config)#no access-list 10 deny 13.13.20.0 0.0.0.255
```

```
R1(config)#end
```

```
R1#show ip access-lists
```

```
// ACL 항목 1개를 삭제하면 ACL 전체 항목을 삭제하기 때문에 특정 항목만 삭제할 수 없다.
```

- 하나삭제하면 전체가 다 삭제된다.
- 부분삭제가 되지 않는다

4) Named ACL 을 이용한 항목 부분 추가 및 삭제

- Named ACL 를 이용하면 순서 번호를 직접 입력할 수 있기 때문에 ACL 항목을 부분 추가하거나 부분 삭제를 할 수 있다

1) 순서 번호를 이용한 항목 부분 추가

① 순서 번호를 이용한 항목 부분 추가

```
R1#conf t
R1(config)#ip access-list standard 10
R1(config-std-nacl)#?
<1-2147483647> Sequence Number
default      Set a command to its defaults
deny         Specify packets to reject
exit          Exit from access-list configuration mode
no            Negate a command or set its defaults
permit        Specify packets to forward
remark        Access list entry comment
R1(config-std-nacl)#15 deny 13.13.20.0 0.0.0.255
R1(config-std-nacl)#end
```

R1#show ip access-lists

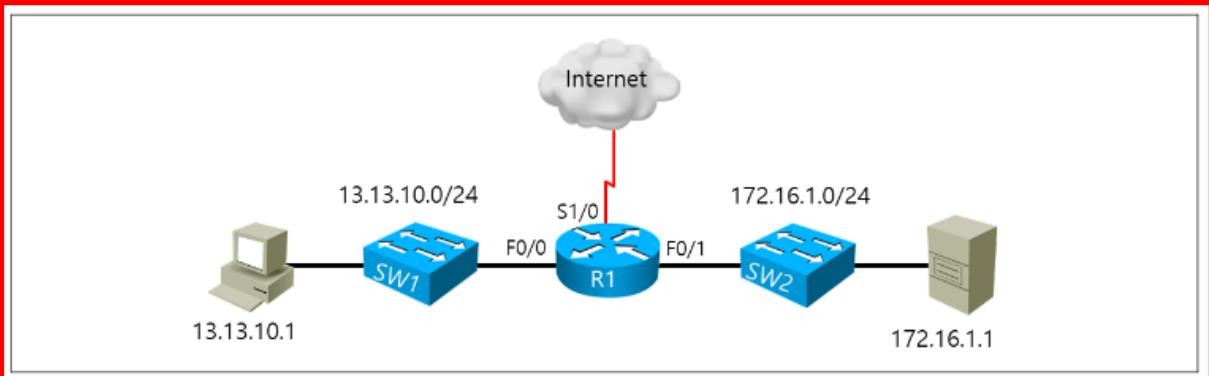
```
Standard IP access list 10
 10 deny 13.13.30.0 0.0.0.255
15 deny 13.13.20.0 0.0.0.255
 20 permit any
```

2) 순서 번호를 이용한 부분 삭제

```
R1#conf t
R1(config)#ip access-list standard 10
R1(config-std-nacl)#no 15
R1(config-std-nacl)#end
R1#show ip access-lists
Standard IP access list 10
 10 deny 13.13.30.0 0.0.0.255
 20 permit any
```

5. Standard ACL

- ACL 번호는 1~99 까지이며, 패킷의 출발지만 검사하여 패킷을 허용하거나 차단한다
- IP header의 출발지만 검사한다. (Source)



Ex1) '13.13.10.0/24' 사용자들이 '172.16.1.1' 서버에 접근하는 것을 차단한다. 단, 인터넷은 되어야한다.

```
access-list 10 deny 13.13.10.0 0.0.0.255
access-list 10 permit any
!
int fa0/1
ip access-group 10 out
```

- 출발지 10점대는 차단
- 나머지는 허용
- fa0/1에 적용을 시켜서 외부에는 접근가능하게 허용

Ex2) '13.13.10.0/24' 사용자들에 대해서 인터넷 사용을 제한하며, '172.16.1.1' 서버로는 접근이 가능하도록 한다.

```
access-list 10 deny 13.13.10.0 0.0.0.255
access-list 10 permit any
!
int fa0/1
ip access-group 10 out
```

- s1/0이다. 틀림.

Ex3) '172.16.1.1' 서버는 인터넷과 연결된 외부 사용자들에게 서비스가 되지 않도록 하며, 오직 '13.13.10.0/24' 사용자들에게만 서비스가 가능하도록 한다.

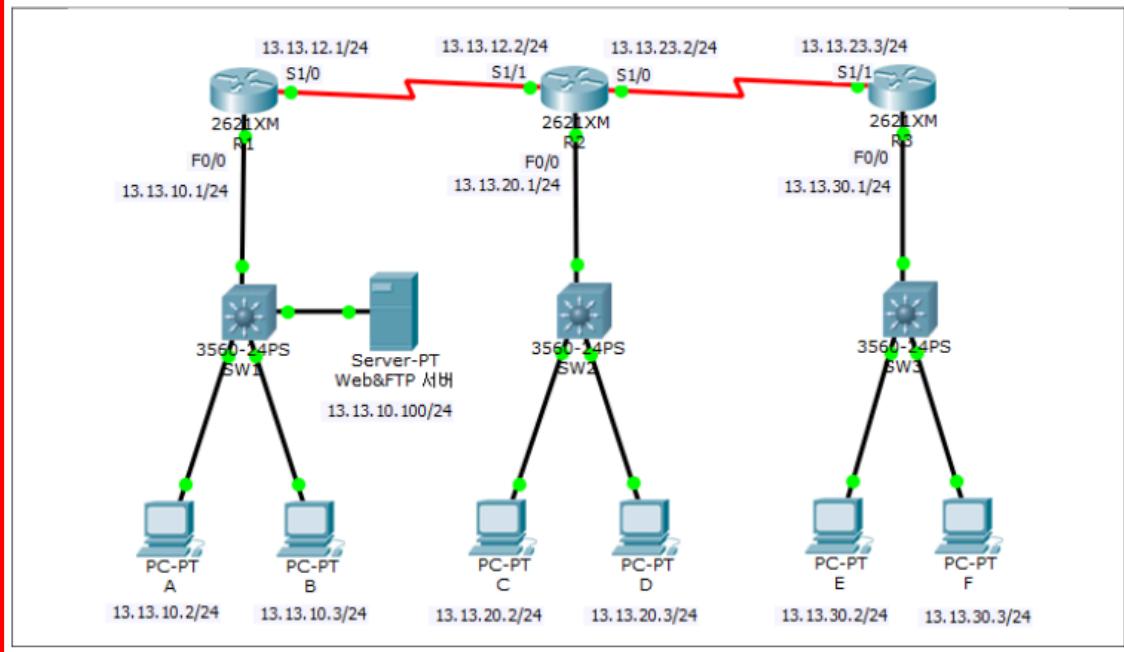
응답 패킷을 아웃바운드로 차단한 경우	요청 패킷을 아웃바운드로 차단한 경우
<pre>access-list 10 deny host 172.16.1.1 access-list 10 permit any ! int s1/0 ip access-group 10 out</pre>	<pre>access-list 10 permit 13.13.30.0 0.0.0.255 ! int fa0/1 ip access-group 10 out</pre>

- 오른쪽이 조금 더 낫다.

Ex4)

Ex4) Standard ACL 예제

- 출발지 '13.13.30.0/24'인 패킷만 '13.13.10.0/24' 서브넷으로 접근하는 것을 차단한다.
- 나머지 패킷들은 허용한다. R1에서 ACL을 구성하도록 한다.



@ R1

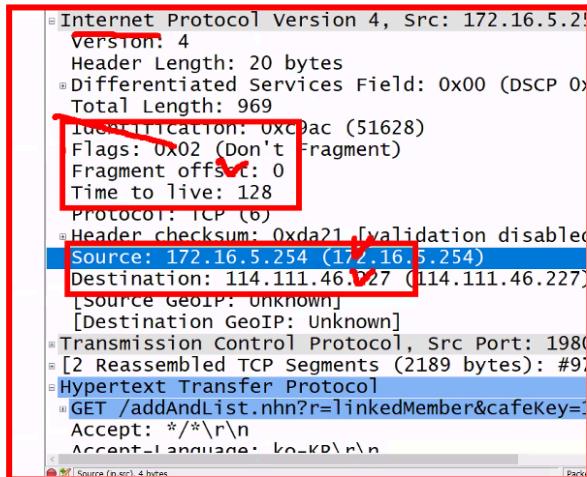
```
conf t
access-list 10 deny 13.13.30.0 0.0.0.255
access-list 10 permit any
!
int s1/0
ip access-group 10 in
end
```

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#access-list 10 deny 13.13.30.0 0.0.0.255
R1(config)#access-list 10 permit any
R1(config)#!
R1(config)#int s1/0
R1(config-if)# ip access-group 10 in
R1(config-if)# end
R1#
%SYS-5-CONFIG_I: Configured from console by console

R1#show ip access-lists
Standard IP access list 10
    10 deny 13.13.30.0 0.0.0.255
    20 permit any
R1#show ip access-lists
Standard IP access list 10
    10 deny 13.13.30.0 0.0.0.255 (96 match(es))
    20 permit any
R1#
```

6. Extended ACL

- ACL 항목으로 사용할 수 있는 범위는 '100~199'까지이며, 출발지 및 목적지를 정의할 뿐만 아니라, 패킷이 사용하는 프로토콜과 애플리케이션 프로토콜 포트 번호를 정의하기 때문에 네트워크를 통하여 전송하는 다양한 트래픽들을 검사할 수 있다. 그리고 다양한 옵션이 제공되므로 시간대별 ACL 필터링, TCP Flag 제어, QoS 관련 설정에서도 사용할 수 있다



- Standard와 달리 다양한 패킷 검사가 가능하다.

Ex1) 출발지 '13.13.10.1'인 PC 가, FTP 서버 '172.16.1.1'로 접근하는 트래픽만 차단한다.

프로토콜	출발지 IP 주소	출발지 포트	목적지 IP 주소	목적지 포트
tcp	13.13.10.1	any	172.16.1.1	20, 21

```

access-list 110 deny tcp host 13.13.10.1 host 172.16.1.1 range 20 21
access-list 110 permit ip any any
!
int fa0/0
ip access-group 110 in

```

Ex2) 출발지 '13.13.10.1'인 PC 가, 웹-서버 '172.16.1.1'로 접근하는 트래픽만 허용한다.

프로토콜	출발지 IP 주소	출발지 포트	목적지 IP 주소	목적지 포트
tcp	13.13.10.1	any	172.16.1.1	80

```

access-list 110 permit tcp host 13.13.10.1 host 172.16.1.1 eq 80
!
int fa0/0
ip access-group 110 in

```

Ex3) '172.16.1.1'로 전송하는 ICMP 를 차단하고 나머지는 허용한다. 단, 서버는 외부로 Ping 이 되어야 한다.

프로토콜	출발지 IP 주소	출발지 포트	목적지 IP 주소	목적지 포트
icmp	any	-	172.16.1.1	-

```

access-list 110 deny icmp any host 172.16.1.1 echo
access-list 110 permit ip any any
!
int fa0/1
ip access-group 110 out

```

- 서버로 가는 echo만 차단하면 서버로 가는 응답만 차단되고 돌아오는 내용은 차단되지 않는다.

언니.			
프로토콜 출발지 IP 주소 출발지 포트	13.13.10.1	172.16.1.1	
icmp any -	ICMP Echo-Reply	ICMP Echo	
access-list 110 deny icmp any host 172.16.1.1 echo	SA 13.13.10.1	SA 172.16.1.1	
access-list 110 permit ip any any	DA 172.16.1.1	DA 13.13.10.1	
!			
int fa0/1			

- 외부에서 오는 에코만 차단시킨다.

Ex4) 출발지 '13.13.10.1'인 PC 가, '172.16.1.0/24'로 접근하는 트래픽을 차단하고 나머지는 허용한다.

프로토콜	출발지 IP 주소	출발지 포트	목적지 IP 주소	목적지 포트
ip	13.13.10.1	-	172.16.1.0/24	-
access-list 110 deny ip host 13.13.10.1 172.16.1.0 0.0.0..255				
access-list 110 permit ip any any				
!				
int fa0/0				
ip access-group 110 in				

Ex5) '13.13.10.1' PC 가 웹-서버 '172.16.1.1'로부터 다운로드하는 트래픽을 차단하고 나머지는 허용한다.

프로토콜	출발지 IP 주소	출발지 포트	목적지 IP 주소	목적지 포트
tcp	172.16.1.1	80	13.13.10.1	any
access-list 110 deny tcp host 172.16.1.1 eq 80 host 13.13.10.1				
access-list 110 permit ip any any				
!				
int fa0/1				
ip access-group 110 in				

Ex6) '13.13.10.0/24' 네트워크에서 '172.16.1.1'로 접근하는 것을 차단하고 나머지는 허용한다.

프로토콜	출발지 IP 주소	출발지 포트	목적지 IP 주소	목적지 포트
ip	13.13.10.0/24	0	172.16.1.1	-
access-list 110 deny ip 13.13.10.0 0.0.0.255 host 172.16.1.1				
access-list 110 permit ip any any				
!				
int fa0/0				
ip access-group 110 in				

Ex7) Extended ACL

- ① 출발지 '13.13.10.0/24' 서브넷이 FTP 서버 '172.16.1.1'로 접근하는 것을 허용한다.
- ② 단, 출발지 '13.13.10.1' 호스트가 FTP 서버 '172.16.1.1'로 접근하는 것을 차단한다.
- ③ 외부 사용자가 인터넷을 통하여 '172.16.1.1' 서버로 Telnet 접속하는 것을 차단한다.
- ④ 나머지 패킷들은 접근이 가능하도록 허용한다.

```
access-list 110 deny tcp host 13.13.10.1 host 172.16.1.1 range 20 21
access-list 110 permit tcp 13.13.10.0 0.0.0.255 host 172.16.1.1 range 20 21      // 마지막에 전체 허용이 있기 때문에 설정할 필요 없음
access-list 110 deny tcp any host 172.16.1.1 eq 23
access-list 110 permit ip any any
!
int fa0/1
ip access-group 110 out
```

- 범위 작은 2번부터 설정

Ex8) Extended ACL

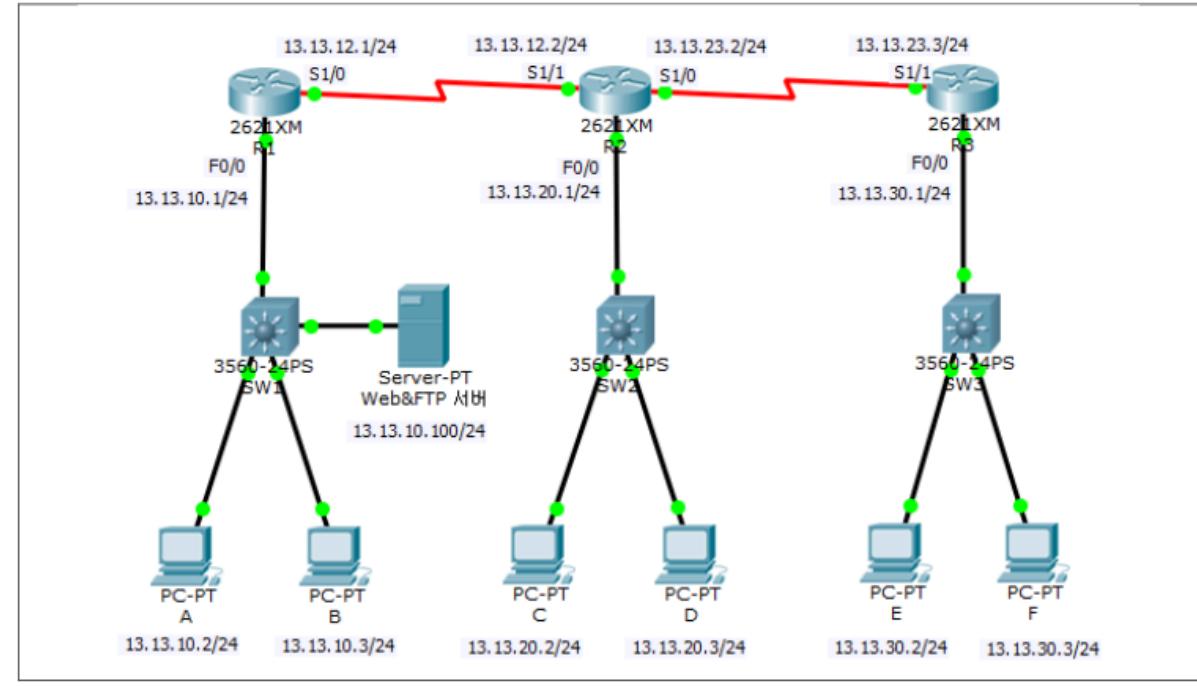
- ① 출발지 '13.13.10.1' 호스트가 웹-서버 '172.16.1.1'로 접근하는 것을 차단한다.
- ② 출발지 '13.13.10.0/24' 서브넷이 웹-서버 '172.16.1.1'로 접근하는 것은 허용한다.
- ③ 외부 사용자가 인터넷을 통하여 '172.16.1.1' 서버로 전송하는 ICMP 패킷을 차단한다.
- ④ 단, '172.16.1.1' 서버는 외부로 Ping 이 되어야 한다.
- ⑤ 나머지 패킷들은 접근이 가능하도록 허용한다.

```
access-list 110 deny tcp host 13.13.10.1 host 172.16.1.1 eq 80
access-list 110 permit tcp 13.13.10.0 0.0.0.255 host 172.16.1.1 eq 80      // 마지막에 전체 허용이 있기 때문에 설정할 필요 없음
access-list 110 deny icmp any host 172.16.1.1 echo
access-list 110 permit ip any any
!
int fa0/1
ip access-group 110 out
```

- 범위 작은 1번부터 설정

Ex9) Extended ACL 예제

- 출발지 '13.13.30.0/24'인 패킷이 내부 로컬 네트워크 '13.13.10.1'로 Telnet 접속되는 것을 차단한다.
- 외부에서 내부 서버 '13.13.10.100'으로 Ping 되는 것을 차단하여라. 단, 서버는 외부로 Ping 이 되어야 한다.
- 출발지 '13.13.20.0/24'인 패킷이 내부 웹서버 '13.13.10.100'으로 접근하는 것을 차단하여라.
- 나머지 패킷은 허용한다.
- R1에서 ACL를 최대한 간결하게 구성하며, R1 Serial 1/0 인터페이스에 적용하여라.



@ R1

```

conf t
access-list 110 deny tcp 13.13.30.0 0.0.0.255 host 13.13.10.1 eq 23
access-list 110 deny icmp any host 13.13.10.100 echo
access-list 110 deny tcp 13.13.20.0 0.0.0.255 host 13.13.10.100 eq 80
access-list 110 permit ip any any
!
int s1/0
ip access-group 110 in
end
!
```

[확인 작업]

- | | |
|----|--|
| 차단 | F -> Desktop -> Command Prompt -> telnet 13.13.10.1 |
| 차단 | D, F -> Desktop -> Command Prompt -> ping 13.13.10.100 |
| 허용 | 내부 서버 -> Desktop -> Command Prompt -> ping 13.13.20.3, ping 13.13.30.3 |
| 차단 | D -> Desktop -> Web Browser -> http://13.13.10.100 |

제8장 DHCP

(Dynamic Host Configuration Protocol)

1. DHCP(Dynamic Host Configuration Protocol)

- 시스템은 TCP/IP 네트워크에 참여하기 위해서 IP 주소 정보가 필요하다. 여기서 IP 주소 정보란 IP 주소, 서브넷 마스크, 기본 게이트웨이 주소, DNS 서버 주소 정보를 의미한다. 시스템이 필요한 IP 주소 정보는 관리자가 직접 설정할 수 있으나, 시스템이 많은 환경에서는 직접 설정하는 것은 번거로운 작업이기 때문에 DHCP 서비스를 이용하여 동적으로 할당해야 한다.
- 원래는 BootStrap protocol 이라고 했었다.
 - 부팅할 때 받아와서 부트스트랩이였다.

1) DHCP 개요

DHCP는 네트워크 관리자들이 IP 주소를 중앙 서버에서 관리하여 클라이언트에게 자동으로 IP 주소 정보를 할당하는 기능을 수행한다. 클라이언트에게 할당한 IP 주소 정보는 일정한 기간 동안에만 유효하도록 하는 임대 서비스 방식으로 동작한다.

2) DHCP 구성 요소

DHCP는 '서버'와 '클라이언트'로 구성된다. 서버는 UDP 포트 번호 67번을 사용하며, 클라이언트는 UDP 포트 번호 68번을 사용한다. DHCP 서버는 원도우, 리눅스, 라우터에서 구성할 수 있으며, 이더넷 장치를 갖고 있는 시스템들은 DHCP 클라이언트로 동작할 수 있다. DHCP 서버와 클라이언트가 서로 다른 네트워크 상에 있다면, 'DHCP Relay Agent'를 구성하여 DHCP 메세지가 유니캐스트로 전송될 수 있도록 해야 한다.

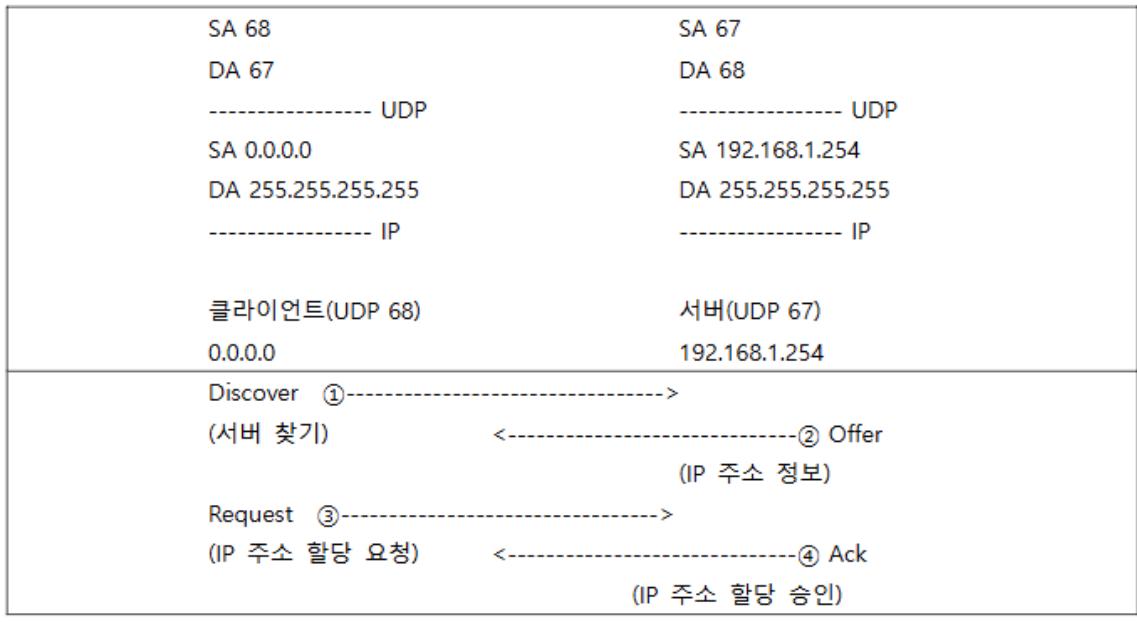
3) DHCP 메시지 및 교환 설정

DHCP 서버와 클라이언트는 다음과 같은 4개의 DHCP 메세지를 교환하여 IP 주소 할당하고 설정한다.

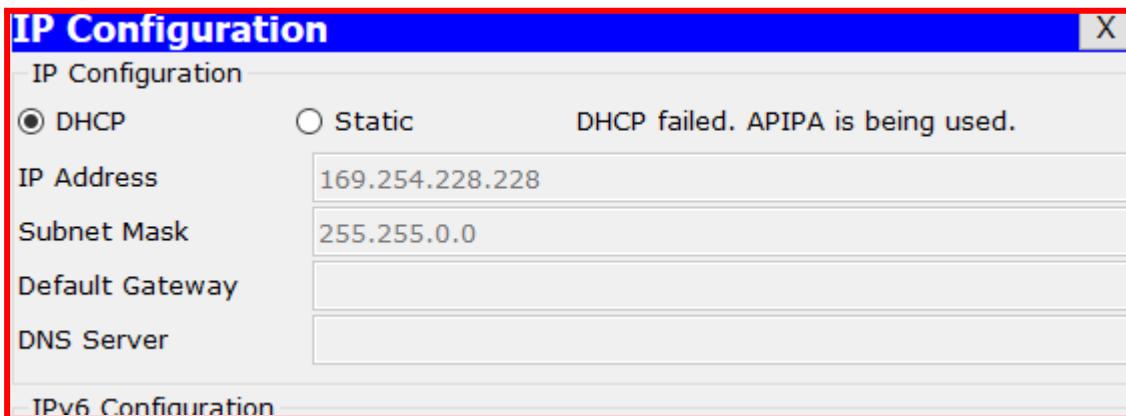
Time	Source	Destination	Protocol	Length	Info
16 206.289	0.0.0.0	255.255.255.255	DHCP	618	DHCP Discover
18 210.548	192.168.1.254	255.255.255.255	DHCP	342	DHCP Offer
19 210.572	0.0.0.0	255.255.255.255	DHCP	618	DHCP Request
20 210.596	192.168.1.254	255.255.255.255	DHCP	354	DHCP ACK

- ① Discover 클라이언트가 서버를 찾기 위해서 전송하는 메세지이다.
- ② Offer 서버가 클라이언트에게 IP 주소 정보를 알리기 위해서 전송하는 메세지이다.
- ③ Request 클라이언트가 서버에게 Offer 메세지 안에 있는 IP 주소 사용 허가를 요청하는 메세지이다.
- ④ Ack 서버가 클라이언트에게 IP 주소 정보 사용 허가를 승인하는 메세지이다.

'16-1.DHCP 캡처 내용.pcap' 파일을 실행하여 DHCP 메세지 내용과 메세지 교환 과정을 확인한다.



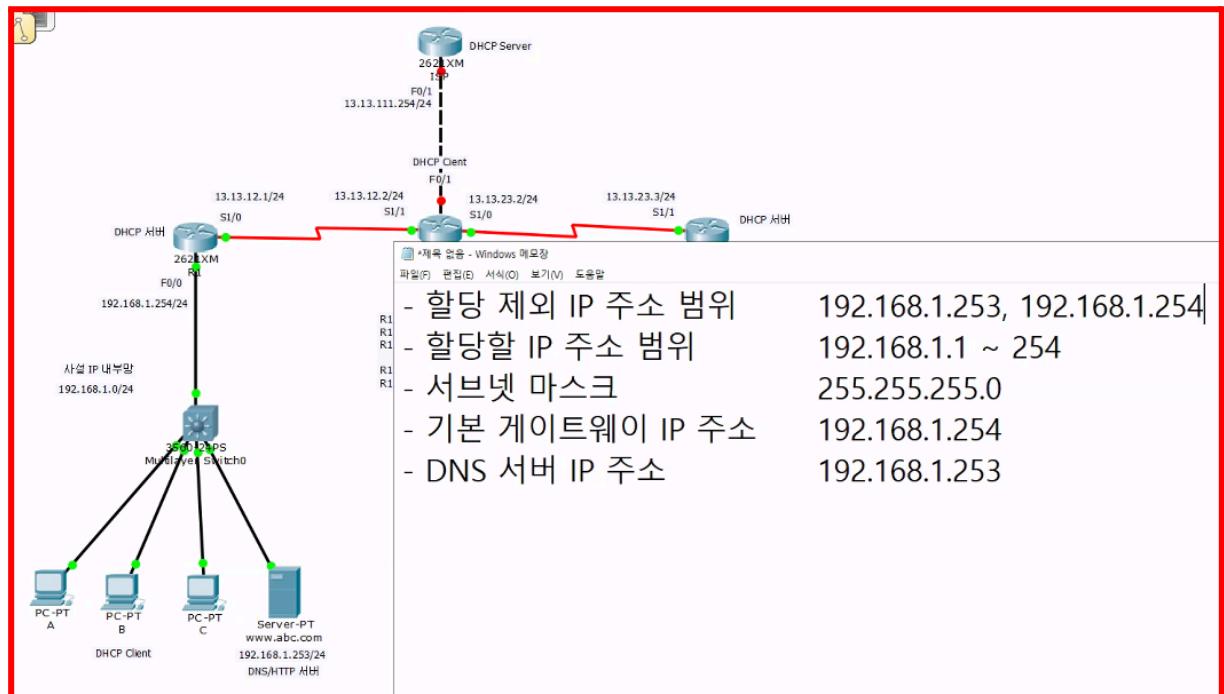
- Offer 안에 IP, 서브넷 마스크 등 필요한 정보를 보낸다.
- 출발지 IP는 서버의 IP이다.
- 클라이언트가 IP 설정을 받아서 자동설정해주는것까지 DHCP 서비스 이다.



- DHCP서버가 없는 상태이다
- 169.254 아이피 대역은 offer를 받지 못했다는것을 의미한다.

2. DHCP 구성

1) DHCP 서버 구성:



명령어:

```
@ R1
conf t
ip dhcp excluded-address 192.168.1.253 192.168.1.254
!
ip dhcp pool NET192
network 192.168.1.0 255.255.255.0
default-router 192.168.1.254
dns-server 192.168.1.253
end
!
```

show run:

```
!
ip dhcp excluded-address 192.168.1.253 192.168.1.254
!
ip dhcp pool NET192
network 192.168.1.0 255.255.255.0
default-router 192.168.1.254
dns-server 192.168.1.253
!
!
!
```

이제 아이피를 자동으로 받아온다:

The screenshot shows two windows. The top window is titled 'IP Configuration' and has tabs for Physical, Config, Desktop, and Custom Interface. It is set to 'IP Configuration' mode, with 'DHCP' selected. The IP Address is 192.168.1.1, Subnet Mask is 255.255.255.0, Default Gateway is 192.168.1.254, and DNS Server is 192.168.1.253. The bottom window is a 'Command Prompt' window titled 'Command Prompt'. It displays the output of the command 'PC>ipconfig /all'. The output shows the connection 'FastEthernet0 Connection:(default port)' with its specific DNS suffix, physical address (00D0.BCB4.E4E4), link-local IPv6 address (FE80::2D0:BCFF:FEB4:E4E4), and various IP configurations matching the ones in the IP Configuration window.

```
Packet Tracer PC Command Line 1.0
PC>ipconfig /all

FastEthernet0 Connection:(default port)

Connection-specific DNS Suffix...:
Physical Address.....: 00D0.BCB4.E4E4
Link-local IPv6 Address....: FE80::2D0:BCFF:FEB4:E4E4
IP Address.....: 192.168.1.1
Subnet Mask.....: 255.255.255.0
Default Gateway.....: 192.168.1.254
DNS Servers.....: 192.168.1.253
DHCP Servers.....: 192.168.1.254
DHCPv6 Client DUID.....: 00-01-00-01-BA-96-46-EE-00-D0-BC-B4-E4-E4

PC>
```

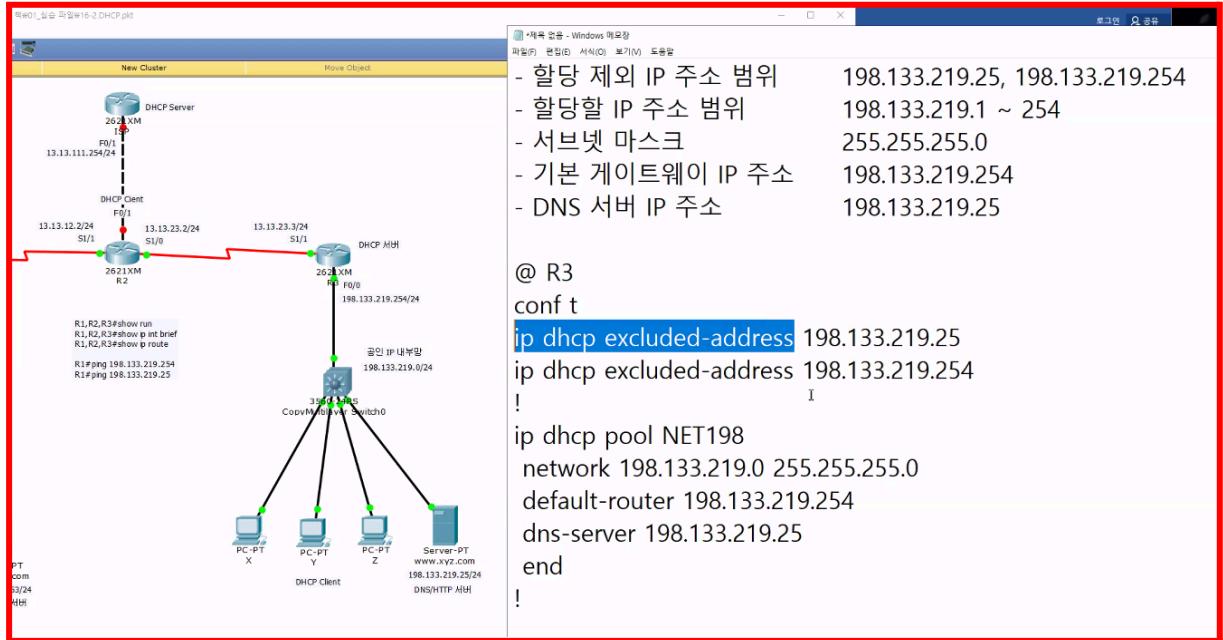
- 보통 Gateway가 DHCP 역할을 많이 한다.

show ip dhcp binding

```
R1#show ip dhcp binding
IP address      Client-ID/          Lease expiration      Type
               Hardware address
192.168.1.1    00D0.BCB4.E4E4      --
192.168.1.2    0090.216A.0A2D      --
192.168.1.3    0060.5C84.02D1      --
R1#
R1#
R1#
```

- 임대정보를 반환해준다.

반대쪽 라우터 설정:



명령어:

```

@ R3
conf t
ip dhcp excluded-address 198.133.219.25
ip dhcp excluded-address 198.133.219.254
!
ip dhcp pool NET198
network 198.133.219.0 255.255.255.0
default-router 198.133.219.254
dns-server 198.133.219.25
end
!
```

show run:

```

enable secret 5 $1$mERr$hx5rVt7rPNoS4wqbXKX7m0
!
!
ip dhcp excluded-address 198.133.219.25
ip dhcp excluded-address 198.133.219.254
!
ip dhcp pool NET198
network 198.133.219.0 255.255.255.0
default-router 198.133.219.254
dns-server 198.133.219.25
!
```

IP Configuration

IP Configuration

DHCP Static

IP Address	198.133.219.1
Subnet Mask	255.255.255.0
Default Gateway	198.133.219.254
DNS Server	198.133.219.25

IPv6 Configuration

```
PC>ipconfig /all

FastEthernet0 Connection:(default port)

Connection-specific DNS Suffix...:
Physical Address.....: 0006.2A9C.CD45
Link-local IPv6 Address....: FE80::206:2AFF:FE9C:CD45
IP Address.....: 198.133.219.1
Subnet Mask.....: 255.255.255.0
Default Gateway.....: 198.133.219.254
DNS Servers.....: 198.133.219.25
DHCP Servers.....: 198.133.219.254
DHCPv6 Client DUID.....: 00-01-00-01-89-40-CD-1E-00-06-2A-9C-CD-45
```

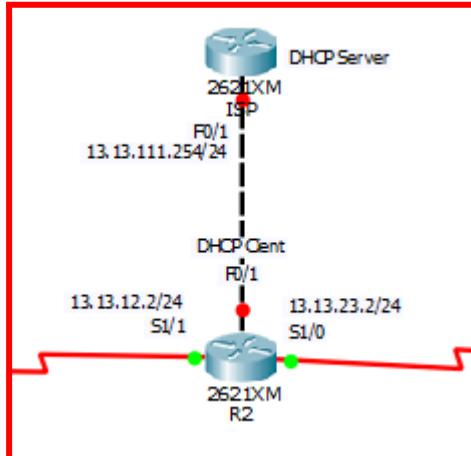
```
PC>
```

- 할당된거 확인

로그확인:

```
R3#show ip dhcp binding
IP address      Client-ID/          Lease expiration      Type
                  Hardware address
198.133.219.1   0006.2A9C.CD45    --                  Automatic
198.133.219.2   00D0.583B.8398    --                  Automatic
198.133.219.3   00D0.FF88.676D    --                  Automatic
R3#
```

마지막으로 ISP 라우터 설정:



```
ISP#conf t
Enter configuration commands, one per line. End with CNTL/Z.
ISP(config)#ip dhcp pool NET13
ISP(dhcp-config)#network 13.13.111.0 255.255.255.0
ISP(dhcp-config)#default-router 13.13.111.254
ISP(dhcp-config)#end
ISP#
%SYS-5-CONFIG_I: Configured from console by console
ISP#
```

show run:

```
!
ip dhcp pool NET13
  network 13.13.111.0 255.255.255.0
  default-router 13.13.111.254
!
```

R2 라우터 설정:

```
R2>en
Password:
R2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#int fa0/1
R2(config-if)#ip address ?
  A.B.C.D  IP address
    dhcp    IP Address negotiated via DHCP
R2(config-if)#ip address dhcp
R2(config-if)#no shutdown

R2(config-if)#
*LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up

*LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to
up

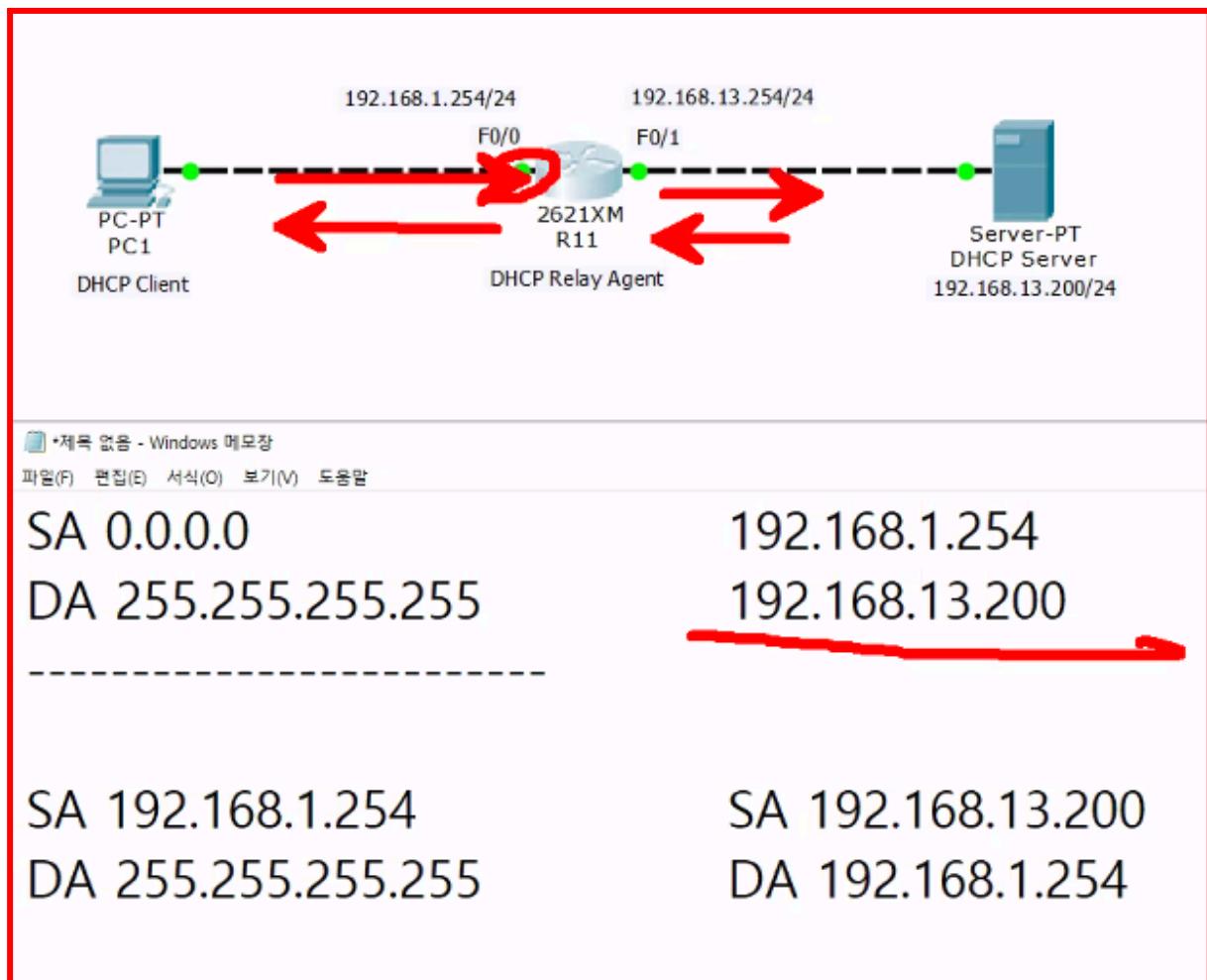
R2(config-if)#
*DHCPC-6-ADDRESS_ASSIGN: Interface FastEthernet0/1 assigned DHCP address
13.13.111.1, mask 255.255.255.0, hostname R2

R2(config-if)#

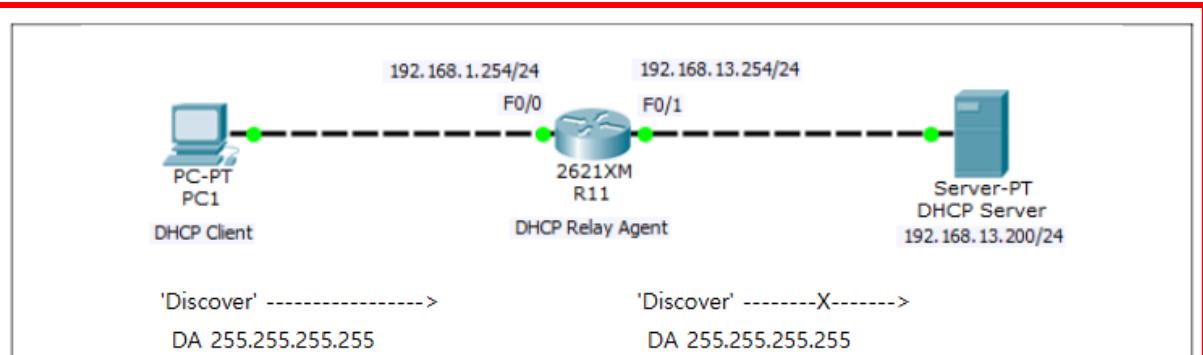
```

2) DHCP Relay Agent

DHCP 메시지들은 브로드캐스트를 이용하여 전송한다. 다음과 같이 DHCP 서버와 클라이언트가 서로 다른 네트워크 환경에 있다면, PC1이 전송한 'Discover' 메시지가 라우터를 통해서 DHCP 서버까지 전송되지 않기 때문에 PC1은 IP 주소 정보를 할당 받을 수 없다.

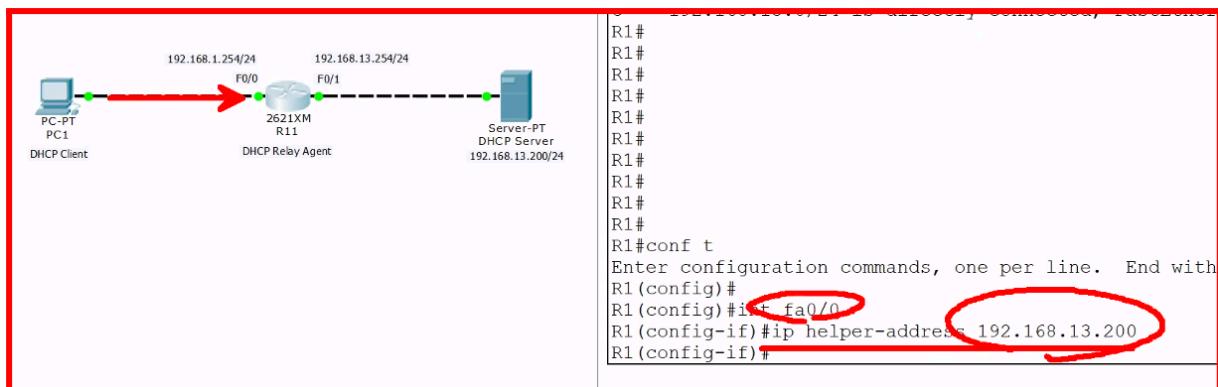


- 브로드캐스트로 받아서 라우터에서 유니캐스트로 만들어준다.
- 그리고 유니캐스트로 받아서 라우터에서 브로드캐스트로 만들어준다.
- DHCP 메세지를 릴레이로 연결해주는것을 DHCP Relay Agent라고 한다.



이런 경우, R11 을 DHCP Relay Agent 로 구성하여 F0/0 인터페이스로 수신하는 'Discover' 메세지의 목적지 주소(255.255.255.255)'를 DHCP 서버 IP 주소(192.168.13.200)로 변환하여 유니캐스트로 전송하면 해결할 수 있다.

'16-3.DHCP Relay Agent(pkt)' 파일을 실행하여 R11 을 DHCP Relay Agent 로 구성한다.



- 이런식으로 세팅을 해놓으면 R11 을 DHCP Relay Agent 로 구성한다.
- 유니캐스트로 바꿔서 설정한다.

제9장 NAT

(Network Address Translation)

1. NAT

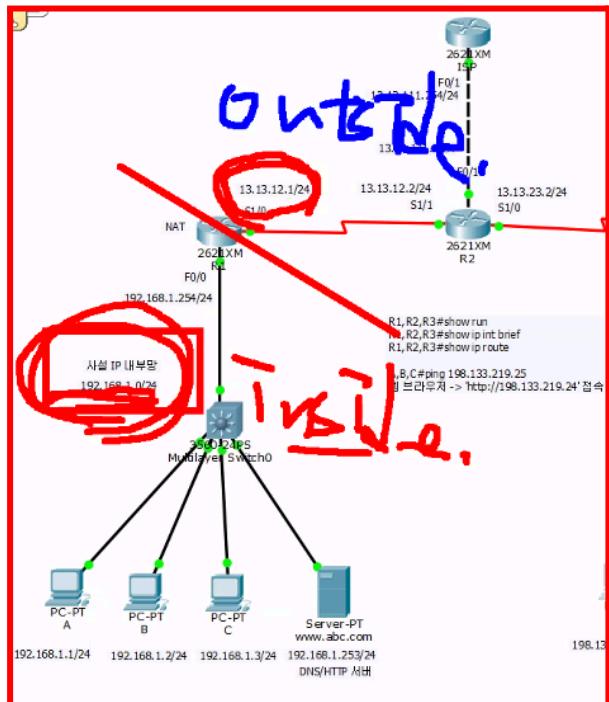
NAT는 IP 주소 및 포트 번호를 변환하는 네트워크 서비스이며, 내부 사설 IP 주소를 사용하는 환경에서 데이터를 인터넷으로 전송할 때 출발지 사설 IP 주소를 공인 IP 주소로 변환한다. NAT를 구성하면 내부 네트워크는 다음과 같은 장점을 갖는다.

목적	내용
보안	사설 IP 주소를 사용하기 때문에 외부에서 접근 자체가 불가능하다.
IP 주소 고갈	사설 IP 주소를 사용해도 인터넷이 되기 때문에 공인 IP 주소를 사용할 필요가 없다.
데이터 전송	출발지 사설 IP 주소를 공인 IP 주소로 변환하기 때문에 응답 패킷이 다시 돌아올 수 있다.

사설IP → 공인 IP로 보낼 때 문제는 없다 (목적지 IP만 잘 되어 있으면 나간다)
하지만 다시 사설IP로 돌아올 때 다른 곳으로 보내진다.

- 따라서 NAT를 설정한다.
- 출발지 주소를 공인IP로 변경해서 내보낸다 (라우터로 유도한다.)
- 이렇게 사용하면 공인IP를 절약할 수 있다.
- 또한 보안 측면에서 사설 IP를 사용하면 더 안전하다 (외부에서 바로 접근이 불가하다)

2. NAT 구성 요소



- inside : inside local address (inside에서 쓰는 주소)
- outside : inside global address (outside에서 쓰는 주소)
- In → Out : 출발지 변경
- Out → In : 목적지 변경

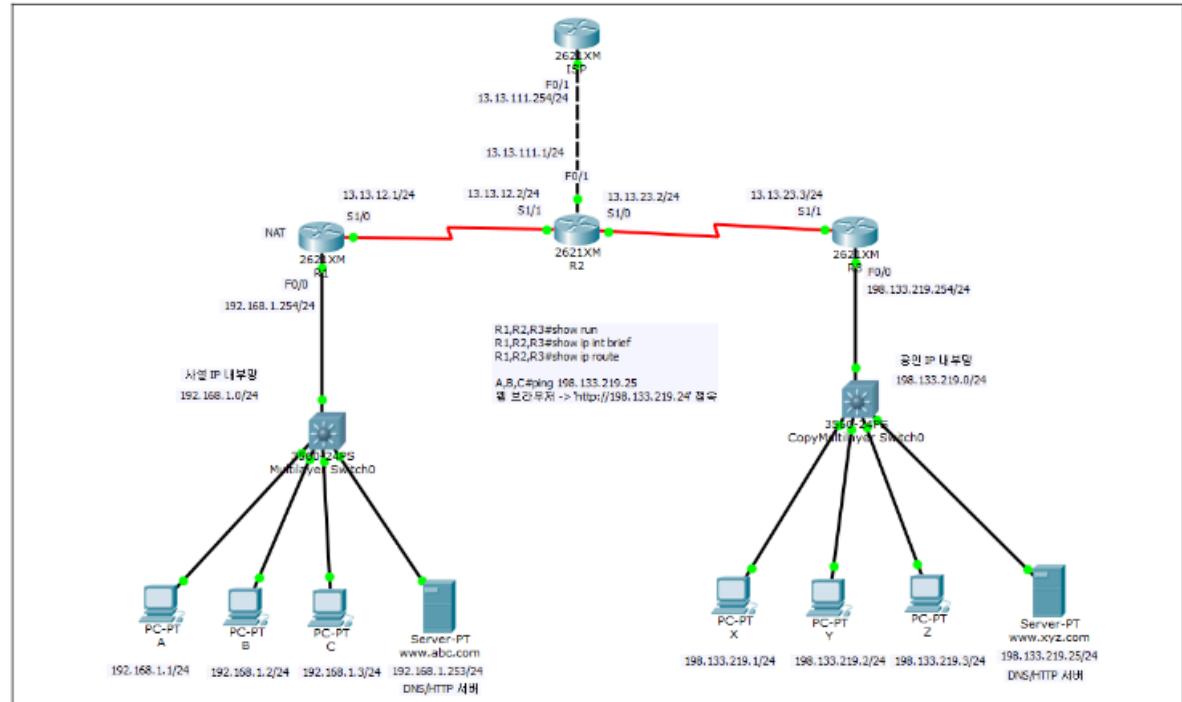
Cisco 라우터에서는 동적 NAT 와 정적 NAT 가 있다. 동적 NAT 는 내부 네트워크 시스템들이 인터넷을 하기 위해서 구성이 필요하며, 정적 NAT 는 외부 네트워크에서 내부 서버(Ex: 웹 서버)로 접근하기 위해서 구성이 필요하다. 리눅스에서 NAT 를 구현할 경우, 동적 NAT 는 '**마스커레이드(Masquerade)**'라고 하며, 정적 NAT 는 '**포트 포워딩(Port Forwarding)**'이라고 한다.

Cisco 라우터에서 NAT 를 구성할 경우, 다음과 같은 NAT 구성 요소들을 파악해야 한다.

NAT 요소	내용
NAT Inside	<ul style="list-style-type: none">- 출발지 주소를 변경할 시스템들이 있는 내부 네트워크- Ex) 사설 IP 주소를 사용하는 내부망
NAT Outside	<ul style="list-style-type: none">- 출발지 주소가 변환되어 패킷이 전송되는 외부 네트워크- Ex) 공인 IP 주소를 사용하는 외부망
Inside Local 주소	<ul style="list-style-type: none">- NAT Inside 에서 사용하는 로컬 네트워크 주소- Ex) 사설 IP 주소
Inside Global 주소	<ul style="list-style-type: none">- Nat Outside 로 패킷이 전송될때 변환되는 주소- Ex) 공인 IP 주소
Inside -> Outside 패킷 전송	<ul style="list-style-type: none">- 패킷의 출발지 주소를 변경한다.
Outside -> Inside 패킷 전송	<ul style="list-style-type: none">- 패킷의 목적지 주소를 변경한다.

ex)

'17-1.NAT.pkt' 파일을 실행하여 R1에서 NAT를 구성한다.



- 콘솔 패스워드는 'ciscocon'이며, 관리자 패스워드는 'cisco'이다.
- A~C_PC에서 198.133.219.25'로 Ping이 되는지 확인한다. (안되는게 정상이다.)
- A~C_PC에서 브라우저를 실행하여 'http://198.133.219.25'로 접속되는지 확인한다. (안되는게 정상이다.)

Inside Local 192.168.1.0/24
Outside 13.13.12.1

```
@ R1
conf t
access-list 10 permit 192.168.1.0 0.0.0.255
!
ip nat inside source list 10 interface s1/0 overload
!
int fa0/0
 ip nat inside
!
int s1/0
 ip nat outside
end
!
```

```

R1#show run
~ 중간 생략 ~
!
interface FastEthernet0/0
ip address 192.168.1.254 255.255.255.0
ip nat inside
duplex auto
speed auto
!
interface FastEthernet0/1
no ip address
duplex auto
speed auto
shutdown
!
interface Serial1/0
ip address 13.13.12.1 255.255.255.0
ip nat outside
!
~ 중간 생략 ~
!
ip nat inside source list 10 interface Serial1/0 overload
ip classless
ip route 0.0.0.0 0.0.0.0 13.13.12.2
!
```

R1#show ip nat translations

Pro	Inside global	Inside local	Outside local	Outside global
tcp	13.13.12.1:1024	192.168.1.3:1025	198.133.219.25:80	198.133.219.25:80
tcp	13.13.12.1:1025	192.168.1.2:1025	198.133.219.25:80	198.133.219.25:80
tcp	13.13.12.1:1026	192.168.1.1:1026	198.133.219.25:80	198.133.219.25:80

- NAT table
- NAT가 작동되는지 확인할 수 있는 테이블이다.

A PC:

Physical Config CLI		IOS Com	
<pre>login ! line aux 0 ! line vty 0 4 password ciscovty login ! ! !</pre>		A(192.168.1.1) -> 198.133.219.25	
		SA 1025	SA 1025
		DA 80	DA 80
		----- TCP	
		SA 192.168.1.1	SA 13.13.12.1
		DA 198.133.219.25	DA 198.133.219.25
		----- IP	
<pre>R1# R1#show ip nat tr R1#show ip nat translations</pre>		13.13.12.1 <- 198.133.219.25	
Pro	Inside global	Inside local	Out
tcp	13.13.12.1:1024	192.168.1.2:1025	198
tcp	13.13.12.1:1025	192.168.1.1:1025	198
tcp	13.13.12.1:1026	192.168.1.3:1025	198
		SA 80	SA 80
		DA 1025	DA 1025
		----- TCP	
		SA 198.133.219.25	SA 198.133.219.25
		DA 13.13.12.1	DA 192.168.1.1
		----- IP	
<pre>R1# R1# R1# R1# R1# R1#</pre>			

- 포트번호를 이용해서 정확하게 inside local 주소를 알 수 있다.

B PC:

Physical Config CLI		IOS Com	
<pre>login ! line aux 0 ! line vty 0 4 password ciscovty login ! !</pre>		B(192.168.1.2) -> 198.133.219.25	
		SA 1025	SA 1024
		DA 80	DA 80
		----- TCP	
		SA 192.168.1.2	SA 13.13.12.1
		DA 198.133.219.25	DA 198.133.219.25
		----- IP	
<pre>R1# R1#show ip nat tr R1#show ip nat translations</pre>		13.13.12.1 <- 198.133.219.25	
Pro	Inside global	Inside local	Out
tcp	13.13.12.1:1024	192.168.1.2:1025	198
tcp	13.13.12.1:1025	192.168.1.1:1025	198
tcp	13.13.12.1:1026	192.168.1.3:1025	198
		SA 80	SA 80
		DA 1024	DA 1025
		----- TCP	
		SA 198.133.219.25	SA 198.133.219.25
		DA 13.13.12.1	DA 192.168.1.1
		----- IP	
<pre>R1# R1# R1# R1# R1# R1#</pre>			

- overload 명령어가 포트번호를 담당한다.

포트번호까지 담당하면:

PNAT 라고한다. 또는 리눅스에서는 포트포워딩이라고 한다.

4. 정적 NAT 구성

외부 네트워크에서 내부 서버로 접근이 필요한 경우, 정적 NAT를 구성해야 한다.

Inside Local : 192.168.1.253(웹 서버)

Inside Global : 13.13.12.100

```
R1#conf t
R1(config)#ip nat inside source static 192.168.1.253 13.13.12.100
R1(config)#
R1(config)#int fa0/0
R1(config-if)#ip nat inside
R1(config-if)#
R1(config-if)#int s1/0
R1(config-if)#ip nat outside
R1(config-if)#end
R1#
```

R1#show run

```
~ 중간 생략 ~
!
ip nat inside source list 10 interface Serial1/0 overload
ip nat inside source static 192.168.1.253 13.13.12.100
ip classless
ip route 0.0.0.0 0.0.0.0 13.13.12.2
```

R1#show ip nat translations

Protocol	Inside global	Inside local	Outside local	Outside global
---	13.13.12.100	192.168.1.253	---	---
tcp	13.13.12.1:1024	192.168.1.3:1025	198.133.219.25:80	198.133.219.25:80
tcp	13.13.12.1:1025	192.168.1.2:1025	198.133.219.25:80	198.133.219.25:80
tcp	13.13.12.1:1026	192.168.1.1:1026	198.133.219.25:80	198.133.219.25:80

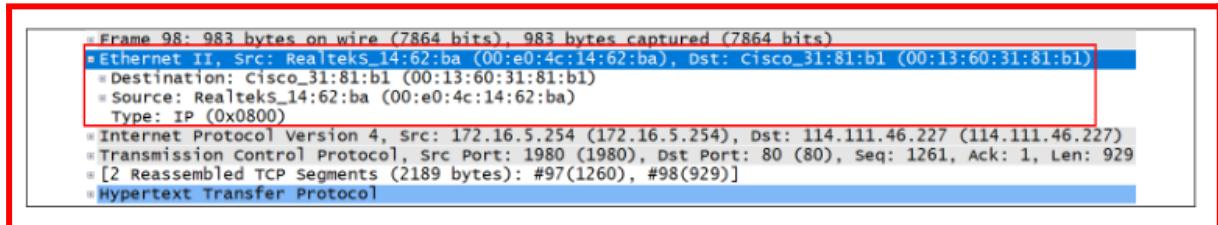
2부: 전적 경로, OSPF 만 알고있자

제3부 LAN 스위칭

제1장 스위치 장비 특징

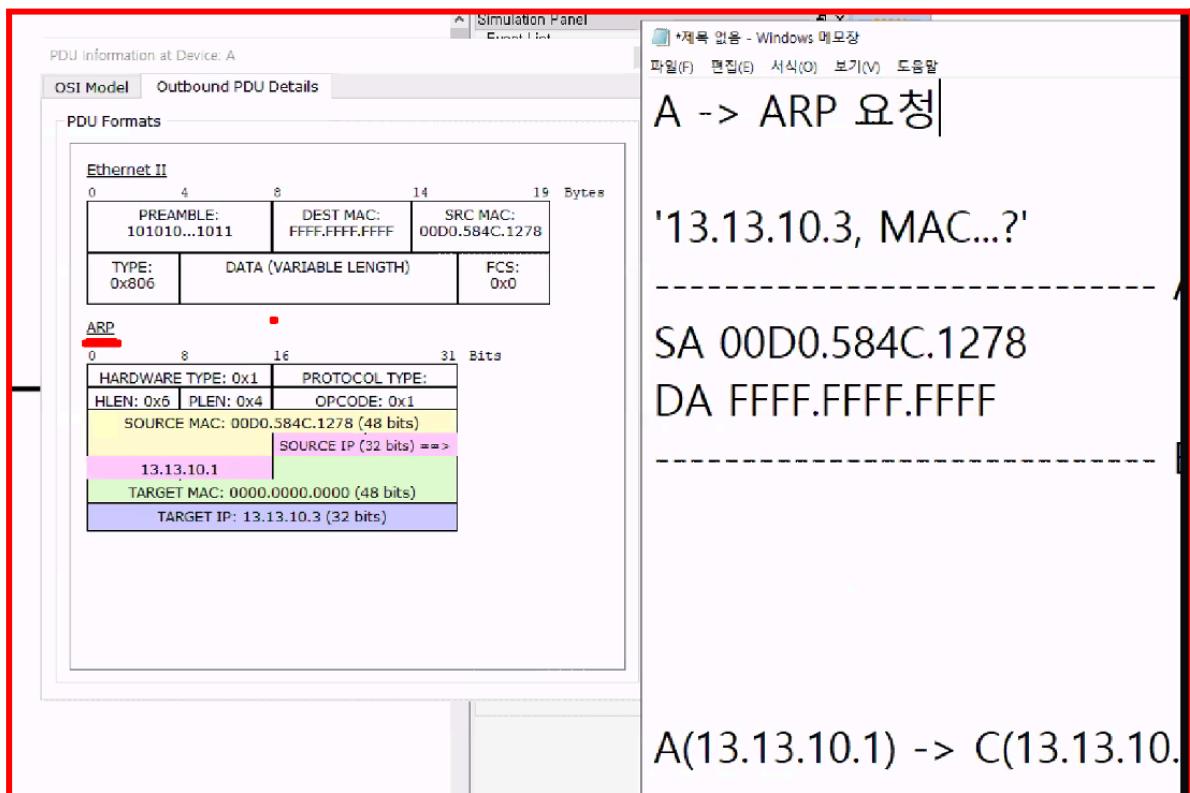
1. 스위치(Switch)

스위치는 Ethernet 헤더의 목적지 MAC 주소를 MAC 주소 테이블을 참조하여 프레임을 전송 처리하는 Layer 2 계층 장비이다. 이러한 데이터 전송 처리 방식을 '스위칭'이라고 하며, 스위칭할 때 참조하는 MAC 주소 테이블 정보는 관리자가 설정하지 않아도 자동으로 생성된다



```
Frame 98: 983 bytes on wire (7864 bits), 983 bytes captured (7864 bits)
* Ethernet II, Src: RealtekS_14:62:ba (00:e0:4c:14:62:ba), Dst: Cisco_31:81:b1 (00:13:60:31:81:b1)
  □ Destination: Cisco_31:81:b1 (00:13:60:31:81:b1)
  □ Source: RealtekS_14:62:ba (00:e0:4c:14:62:ba)
  □ Type: IP (0x0800)
* Internet Protocol Version 4, Src: 172.16.5.254 (172.16.5.254), Dst: 114.111.46.227 (114.111.46.227)
* Transmission Control Protocol, Src Port: 1980 (1980), Dst Port: 80 (80), Seq: 1261, Ack: 1, Len: 929
* [2 Reassembled TCP Segments (2189 bytes): #97(1260), #98(929)]
* Hypertext Transfer Protocol
```

- L2 처리는 스위칭이라고 생각하자.
- 스위치는 Ethernet 헤더만 본다.

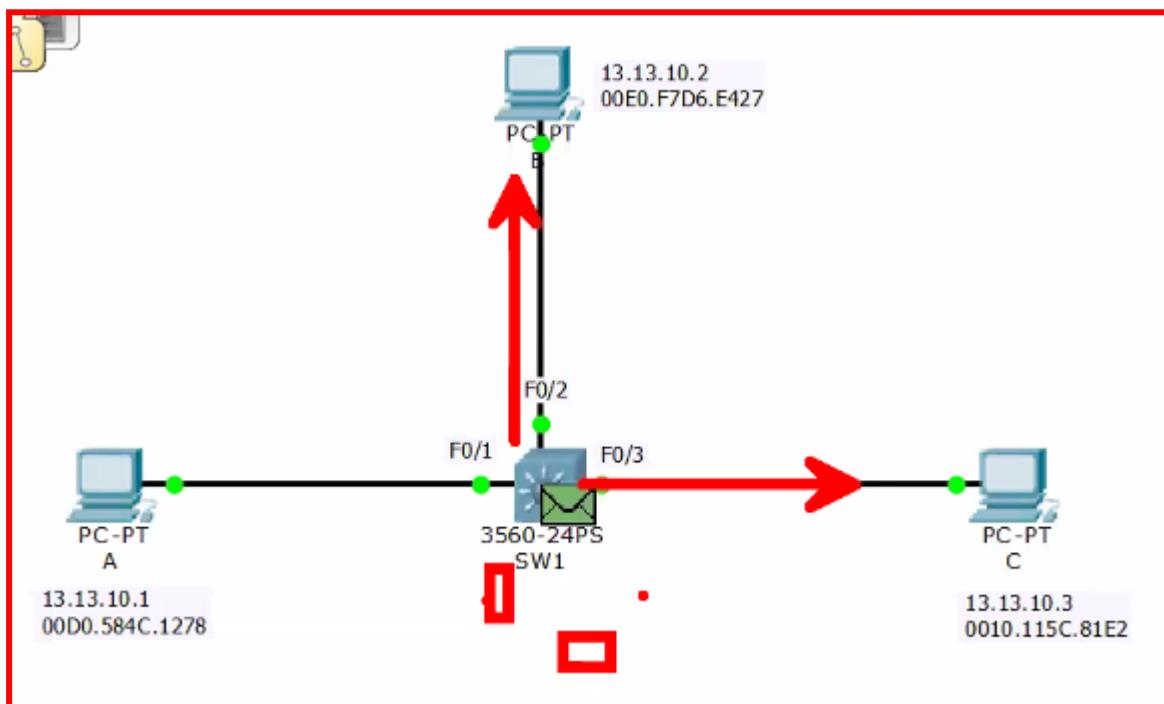


- ARP 브로드캐스 요청

Learning

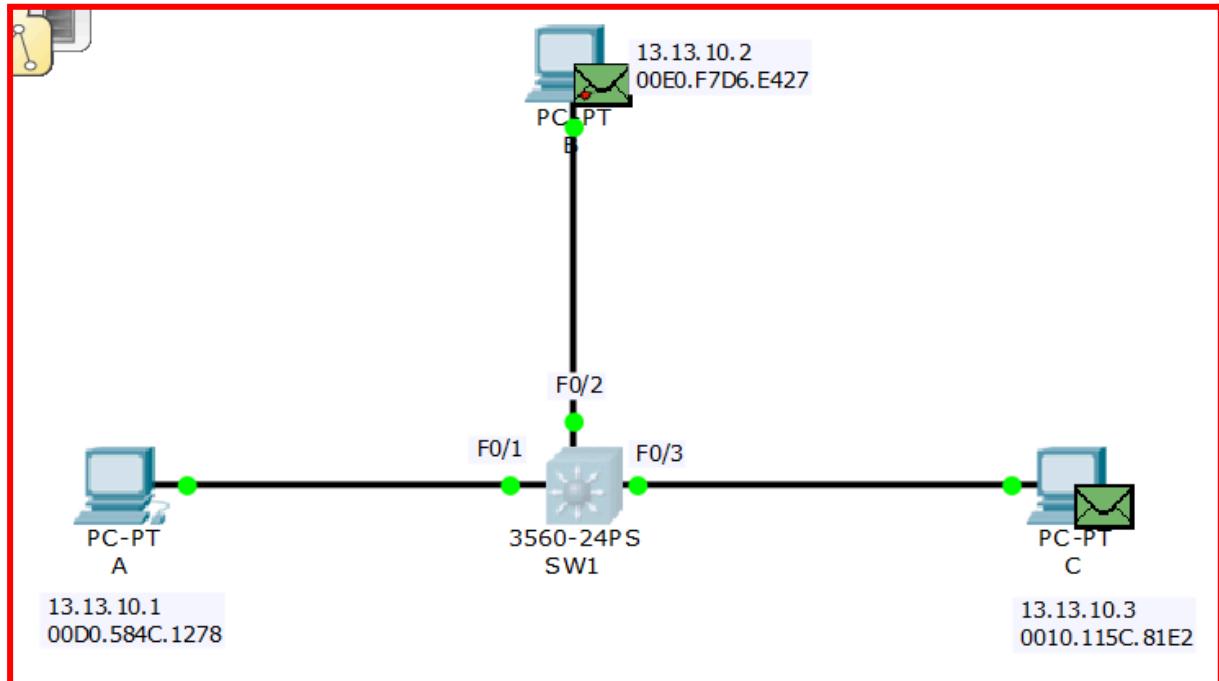
```
SW1>en
Password:
Password:
SW1#en
SW1#show mac address-table
      Mac Address Table
-----
Vlan      Mac Address          Type      Ports
----      -----              -----      -----
SW1#
SW1#
SW1#show mac address-table
      Mac Address Table
-----
Vlan      Mac Address          Type      Ports
----      -----              -----      -----
      1      00d0.584c.1278    DYNAMIC   Fa0/1
SW1#
```

- SW1 > show mac address-table
- 브로드캐스트로 A PC에서 맥어드레스를 보내면 학습을 한다. (Learning)

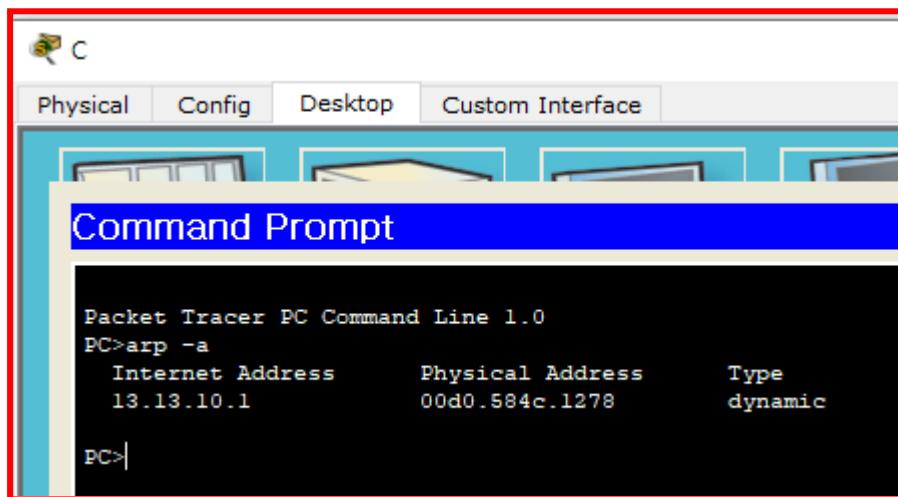


- 그런다음 ARP를 복제해서 모든 곳에 내보낸다 (브로드캐스트)

Flooding



- 이런식으로 브로드캐스트 동작을 받아서 넘기는걸 플러딩(flooding) 이라고 한다.
- 도착한 다음 맥어드레스가 자기가 아니면 드랍처리를 한다.



- 찾는 PC가 자기가 맞으면 맥어드레스 정보를 등록한다.
- 그리고 ARP 응답을 보낸다.
- 출발지 맥 : C PC, 도착지: A PC (자기일 때 찾고 있던 PC)

The screenshot shows a Cisco IOS CLI session on the left and a Windows clipboard on the right.

Cisco IOS CLI Session:

```

Password:
Password:
Password:
% Bad secrets

SW1>en
Password:
SW1#
SW1#show mac address-table
Mac Address Table
-----
Vlan      Mac Address          Type      Ports
-----  -----
SW1#
SW1#
SW1#show mac address-table
Mac Address Table
-----
Vlan      Mac Address          Type      Ports
-----  -----
1        00d0.584c.1278        DYNAMIC   Fa0/1
SW1#

```

Windows Clipboard (ARP Response):

ARP 응답 <- C
'13.13.10.3, MAC 0010.115C.81E2
----- ARP
SA 0010.115C.81E2
DA 00D0.584C.1278
----- ETH

Windows Clipboard (ICMP Echo Request):

A(13.13.10.1) -> C(13.13.10.3)
----- ICMP
SA 13.13.10.1
DA 13.13.10.3

Vlan	Mac Address	Type	Ports
1	0010.115c.81e2	DYNAMIC	Fa0/3
1	00d0.584c.1278	DYNAMIC	Fa0/1

- 다시 ARP를 APC 쪽으로 보내면 Mac Address를 업데이트한다.

The screenshot shows a Cisco IOS CLI session on the left and a terminal window on the right.

Cisco IOS CLI Session:

```

PC-PT
A
13.13.10.1
00D0.584C.1278

```

Terminal Window (PC-A ARP Request):

```

DNS Servers.....: 0.0.0.0
DHCP Servers.....: 0.0.0.0
DHCPv6 Client DUID...: 00-01-00-01-39-46-9B-49-00

PC>arp -a
No ARP Entries Found
PC>show arp
Invalid Command.

PC>arp - a
Invalid Command.

PC>arp - a
Invalid Command.

PC>arp -a
Internet Address      Physical Address      Type
13.13.10.3            0010.115c.81e2      dynamic
PC>

```

- 마지막으로 스위치에서 A PC로 정보를 보낸다 (A PC는 C PC의 맥 어드레스 정보를 업데이트한다)

Aging Timer

Vlan	Mac Address	Type	Ports
1	0010.115c.81e2	DYNAMIC	Fa0/3
1	00d0.584c.1278	DYNAMIC	Fa0/1

SW1#

<- aging 타이머 시작 (300초)

282초

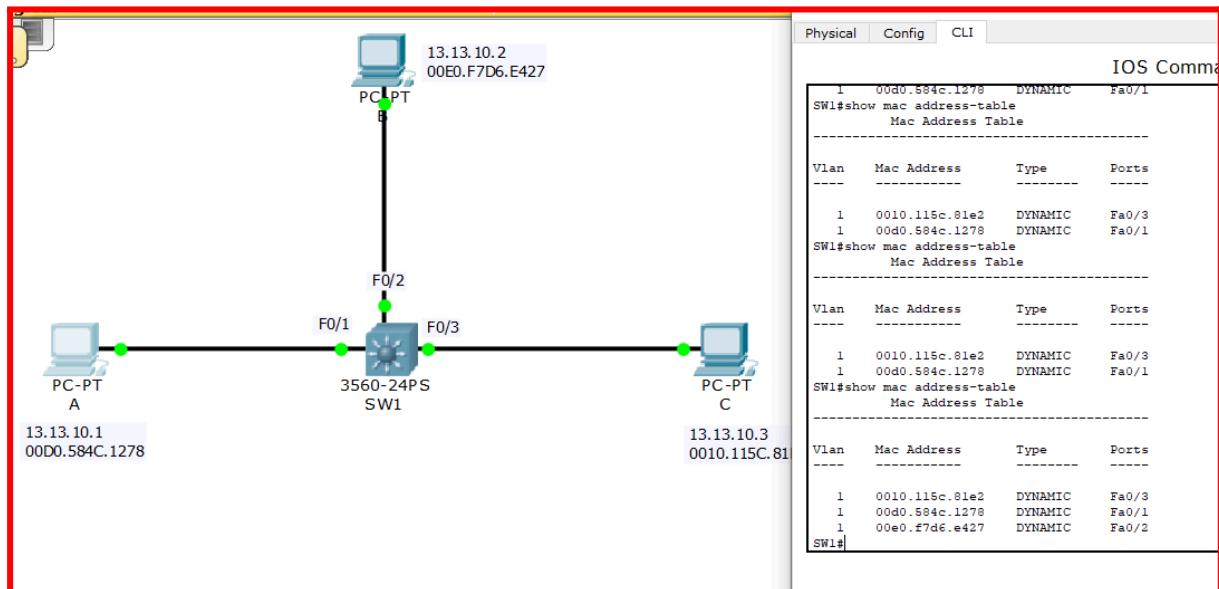
- 자 그리고 원하는 ICMP를 C PC로 보낼 수 있다.
- 만약 스위치에 등록된 맥 어드레스로 정보를 안보내주면 300초뒤에 사라진다

Forwarding

Vlan	Mac Address	Type	Ports	DA 13.13.10.3
1	00d0.584c.1278	DYNAMIC	Fa0/1	IP SA 00D0.584C.1278
SW1#				DA 0010.115c.81e2
				----- ETH
Vlan	Mac Address	Type	Ports	
1	0010.115c.81e2	DYNAMIC	Fa0/3	
1	00d0.584c.1278	DYNAMIC	Fa0/1	<- aging 타이머 시작 (300초)
SW1#				

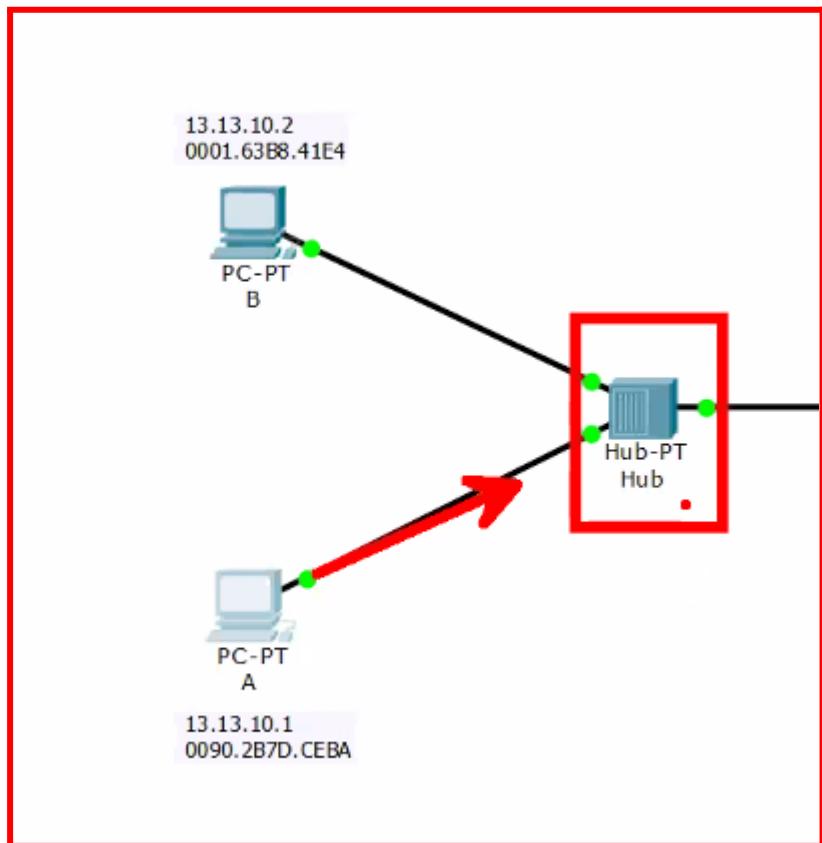
도착

- 맥 주소가 맞으면 설정된 포트로 정보를 보내는것을 포워딩이라고 한다.

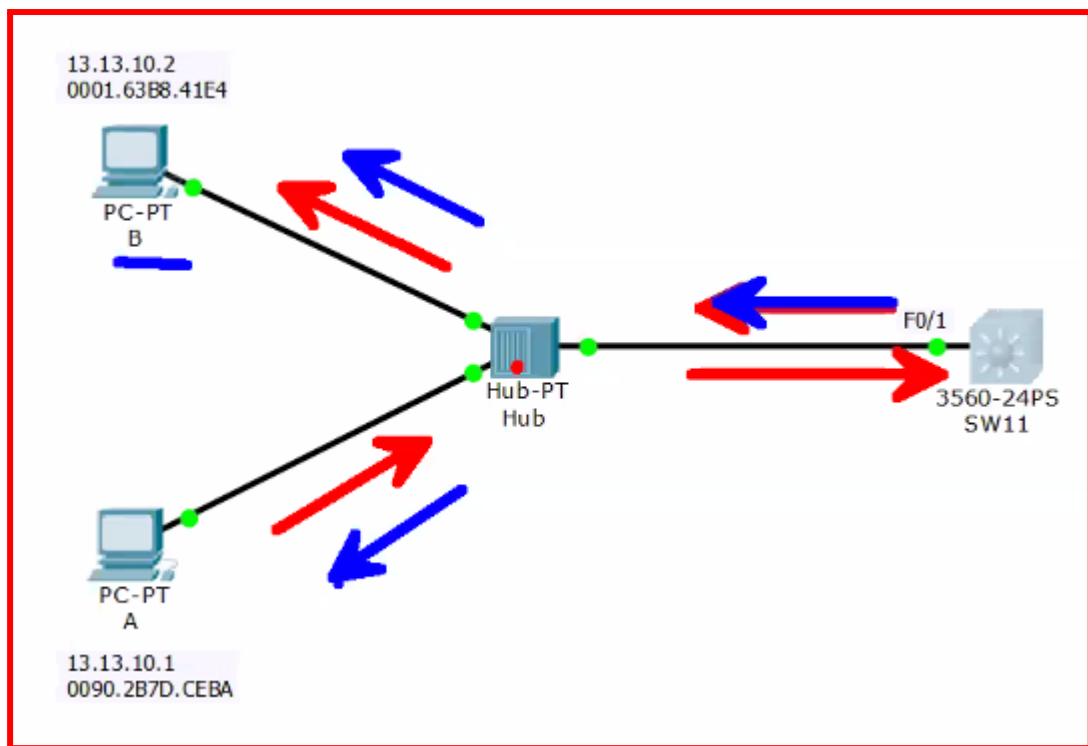


- 핑을 보내면 맥어드레스 테이블이 자동으로 업데이트를 시켜준다.

HUB

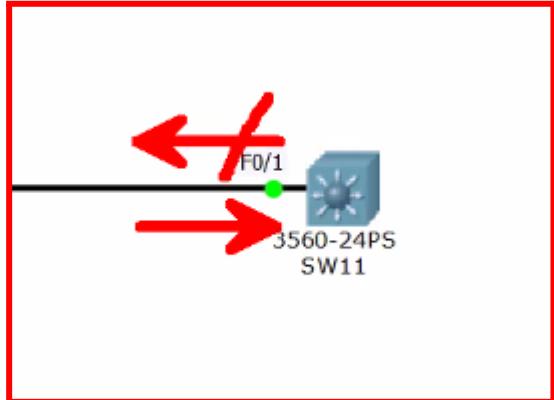


- L1 장비 (케이블하고 다를게 없음)
- 전기신호 보내줌
- 전기 들어오면 전체적으로 다 뿌려 버린다. (유니캐스트가 불가능한 장비)



- 이런식으로 허브를 사용하면 루프가 발생해버린다. (모든 방향으로 전기신호를 보낸다)

Filtering(필터링) 기능



- 이 상황을 방지하기위해 허브가 에초에 차단을 시켜버린다.
- 루프가 발생될것같으면 안나가게 자기가 막아버린다.

Transparent Bridging 특성

- Learning
- Flooding
- Forwarding
- Aging
- Filtering
 - 자동으로 해주는 기능들 (연결하면 자동으로 학습해준다)
 - 사용하지 않으면 삭제, 갱신까지 해준다.
 - 스위치에서 제공하는 기능

제2장 VLAN(Virtual LAN)

VLAN(Virtual LAN):

내부 네트워크 환경에 시스템이 많이 있거나 또는 추가되면 브로드캐스트 플러딩이 비례적으로 증가되므로 대역폭 부족 현상, 전송 장비 부하라는 문제가 발생될 수 있다. 그리고 스위치로 구성된 내부 네트워크는 하나의 브로드캐스트 도메인으로 동작하기 때문에 시스템들 간에 유니캐스트 접근 자체가 가능하므로 보안적인 측면에서도 문제가 발생될 수 있다.

이러한 문제를 해결하기 위해서 스위치에는 VLAN 기능을 지원한다. VLAN 기능을 이용하면 내부 네트워크를 여러 개의 논리적인 네트워크로 분리할 수 있기 때문에 브로드캐스트 플러딩을 최소화하고 서로 다른 VLAN 간에 유니캐스트 접근을 차단시킬 수 있다.

VLAN을 구성한 내부 네트워크는 다음과 같은 장점을 갖게된다.

- ① 논리적인 브로드캐스트 도메인을 분할하여 브로드캐스트 플러딩을 최소화한다.
- ② 서로 다른 VLAN 간에 브로드캐스트가 차단되므로 ARP 학습에 의한 유니캐스트 접근이 불가능하다.
- ③ Spanning-Tree 이중화 환경에서 VLAN 로드 분산이 가능하다.
- ④ 논리적인 브로드캐스트 도메인이기 때문에 위치상 제약이 없으며, 관리가 효율적이다.

- 같은 VLAN에 있는 PC끼리만 브로드캐스팅이 가능하다. (같은 네트워크로 만들어준다)
- 라우터를 통해서만 다른 VLAN끼리 통신이 가능하며, ACL로 이걸 막을 수도 있다.

2. VLAN Database

- 기본적으로 VLAN 1, VLAN 1002~1005가 있으며, 이 5개의 VLAN은 삭제 및 수정이 불가능하다. VLAN은 1~4094 범위 내에서 사용할 수 있으며, 스위치에서 사용 가능한 VLAN은 최대 1005개로 제한되어 있다. VLAN 1~1005까지를 Standard VLAN이라고 하며, VLAN 1006~4094까지를 Extended VLAN이라고 한다.

VLAN-ID : 12bit(0~4095)

1~1005	Standard VLAN
1006~4094	Extended VLAN

SW1#show vlan brief

VLAN	Name	Status	Ports
1	default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4 Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9, Fa0/10, Fa0/11, Fa0/12 Fa0/13, Fa0/14, Fa0/15, Fa0/16 Fa0/17, Fa0/18, Fa0/19, Fa0/20 Fa0/21, Fa0/22, Fa0/23, Fa0/24 Gig0/1, Gig0/2
1002	fdci-default	active	
1003	token-ring-default	active	
1004	fddinet-default	active	
1005	trnet-default	active	

- VLan 1 : 브로드캐스트가 공유되는 포트
- VLAN1 안에 있는 포트들은 같은 네트워크가 된다.

```
SW1#conf t
SW1(config)#no vlan 1
Default VLAN 1 may not be deleted.

SW1(config)#
SW1(config)#vlan 1
SW1(config-vlan)#name ABC
Default VLAN 1 may not have its name changed.

SW1(config-vlan)#end
SW1#
```

- VLAN 데이터베이스에 기본적으로 생성되어 있는 VLAN 1, VLAN 1002~1005는 삭제하거나 수정할 수 없다.

Static vs dynamic VLAN

static vlan

```
int fa0/1
switchport access vlan 11
```

dynamic vlan

0001.1111.1111	vlan 11
0002.2222.222	vlan 12

- static은 각 VLAN을 포트마다 설정
- Dynamic은 각 VLAN을 MAC 주소마다 설정 (어느 포트에다가 꽂든 VLAN이 안바뀜)

3. VLAN 생성

```
SW1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)#vlan 11
SW1(config-vlan)#vlan 12
SW1(config-vlan)#+Z
SW1#
%SYS-5-CONFIG_I: Configured from console by console

SW1#show vlan brief

VLAN Name          Status    Ports
----  -----
1    default        active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                           Fa0/5, Fa0/6, Fa0/7, Fa0/8
                           Fa0/9, Fa0/10, Fa0/11, Fa0/12
                           Fa0/13, Fa0/14, Fa0/15, Fa0/16
                           Fa0/17, Fa0/18, Fa0/19, Fa0/20
                           Fa0/21, Fa0/22, Fa0/23, Fa0/24
                           Gig0/1, Gig0/2
11   VLAN0011       active
12   VLAN0012       active
1002  fddi-default  active
1003  token-ring-default  active
1004  fddinet-default  active
1005  trnet-default  active
SW1#
```

VLAN 이름 변경:

```
SW1#conf t
Enter configuration commands,
SW1(config)#vlan 11
SW1(config-vlan)#name VLAN_A
SW1(config-vlan)#vlan 12
SW1(config-vlan)#name VLAN_B
SW1(config-vlan)#vlan 13
SW1(config-vlan)#name VLAN_C
SW1(config-vlan)#end
SW1#
```

이름 변경완료:

```
SW1#show vlan brief

VLAN Name          Status    Ports
----  -----
1    default        active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                           Fa0/5, Fa0/6, Fa0/7, Fa0/8
                           Fa0/9, Fa0/10, Fa0/11, Fa0/12
                           Fa0/13, Fa0/14, Fa0/15, Fa0/16
                           Fa0/17, Fa0/18, Fa0/19, Fa0/20
                           Fa0/21, Fa0/22, Fa0/23, Fa0/24
                           Gig0/1, Gig0/2
11   VLAN_A         active
12   VLAN_B         active
13   VLAN_C         active
1002  fddi-default  active
1003  token-ring-default  active
1004  fddinet-default  active
1005  trnet-default  active
SW1#
```

4. VLAN Access 설정:

- range라는 키워드로 한번에 설정 가능

@ SW1 설정:

```
conf t
int range fa0/1 - 3, fa0/7, fa0/10 - 13
switchport mode access
switchport access vlan 11
!
int range fa0/4 - 6, fa0/8 - 9, fa0/14 - 17
switchport mode access
switchport access vlan 12
end
!
```

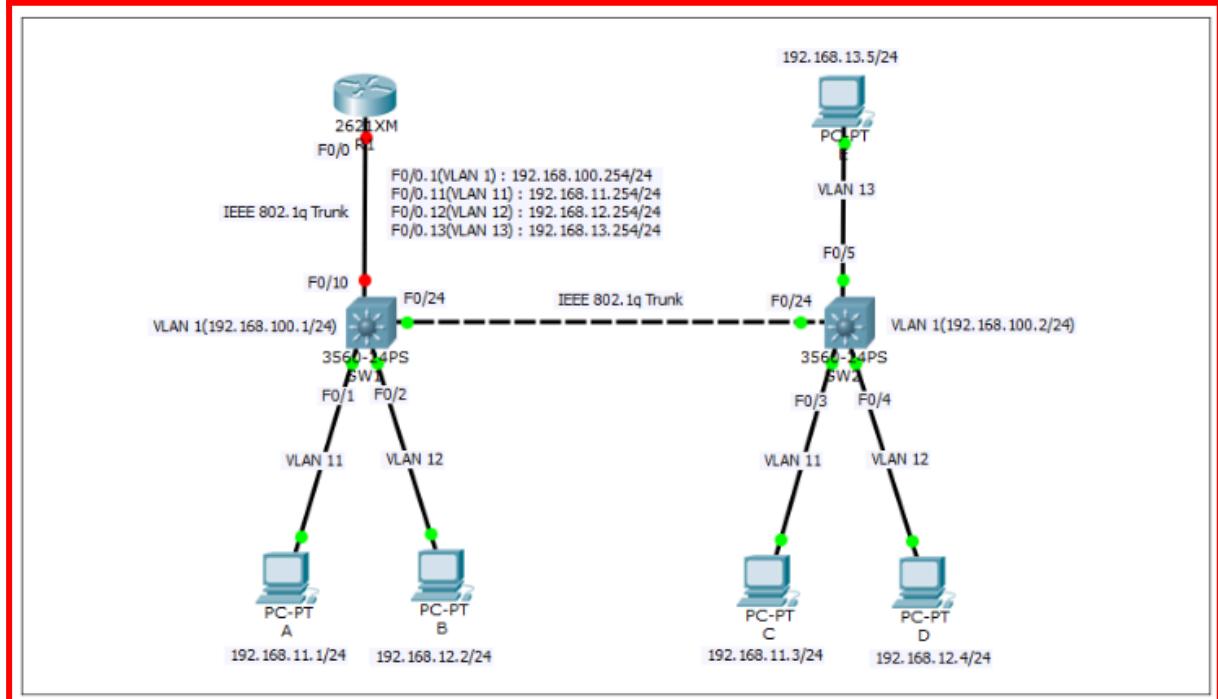
```
SW1#show vlan brief
```

VLAN	Name	Status	Ports
1	default	active	Fa0/18, Fa0/19, Fa0/20, Fa0/21 Fa0/22, Fa0/23, Fa0/24, Gig0/1 Gig0/2
11	VLAN_A	active	Fa0/1, Fa0/2, Fa0/3, Fa0/7 Fa0/10, Fa0/11, Fa0/12, Fa0/13
12	VLAN_B	active	Fa0/4, Fa0/5, Fa0/6, Fa0/8 Fa0/9, Fa0/14, Fa0/15, Fa0/16 Fa0/17
13	VLAN_C	active	
1002	fddi-default	active	
1003	token-ring-default	active	
1004	fdnet-default	active	
1005	trnet-default	active	

@ SW1 설정 삭제:

```
conf t
int range fa0/1 - 3, fa0/7, fa0/10 - 13
no switchport mode access
no switchport access vlan 11
!
int range fa0/4 - 6, fa0/8 - 9, fa0/14 - 17
no switchport mode access
no switchport access vlan 12
end
!
```

설정 예시)

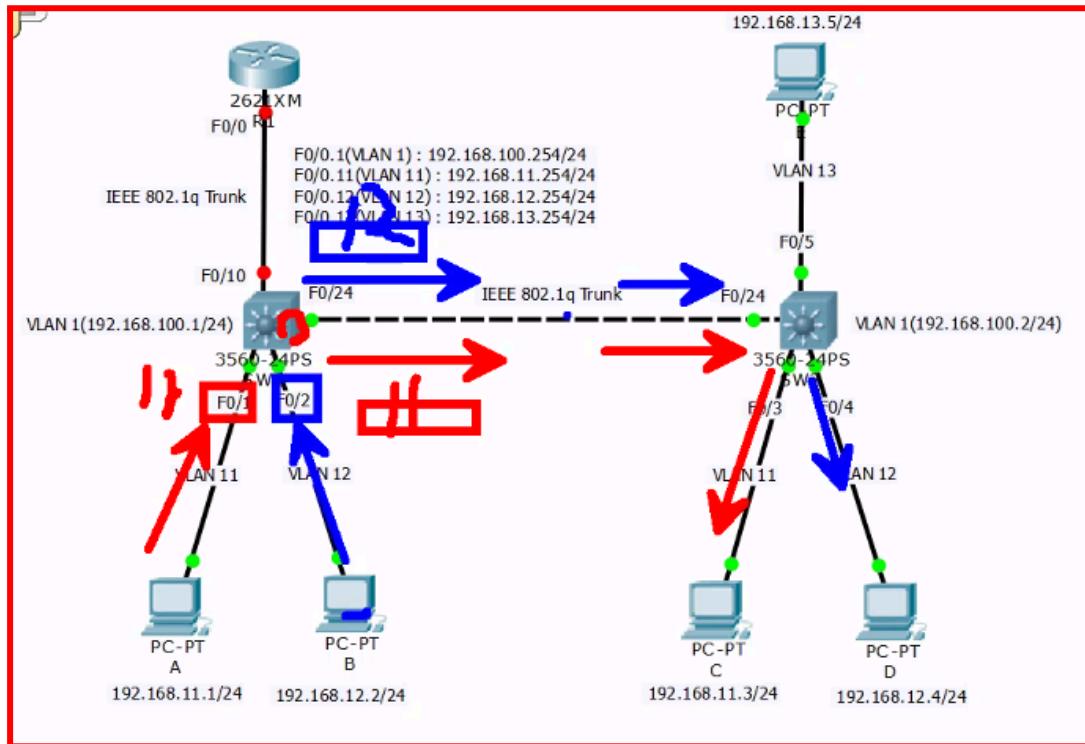


SW1, SW2에서 스위치 포트를 각각의 VLAN 으로 매핑되도록 액세스 설정을 실시한다.

```
@ SW 1  
conf t  
int fa0/1  
switchport mode access  
switchport access vlan 11  
!  
int fa0/2  
switchport mode access  
switchport access vlan 12  
!  
  
@ SW 2  
conf t  
int fa0/3  
switchport mode access  
switchport access vlan 13  
!  
int fa0/4  
switchport mode access  
switchport access vlan 14  
!  
int fa0/5  
switchport mode access  
switchport access vlan 15  
end  
!
```

5. 트렁크 구성

- 하나의 링크를 이용하여 서로 다른 VLAN 이더넷 프레임들을 전송 처리하는 기능



- VLAN ID를 부착을 한 다음 하나의 링크를 이용하여 다른 VLAN으로 보내는 기능.
- 프로토콜
 - IEEE 802.17 트렁크 프로토콜
 - 아이디를 하나 얹어주는 느낌
 - Cisco ISL 트렁크 프로토콜
 - 이건 통째로 Ethernet 프로토콜 위에 감싼다. (비효율적)

IEEE 802.17 트렁크 프로토콜

```

Frame 25: 118 bytes on wire (944 bits), 118 bytes captured (944 bits)
Ethernet II, Src: c0:00:10:a0:00:01 (c0:00:10:a0:00:01), Dst: c0:01:10:a0:00:01 (c0:01:10:a0:00:01)
  802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 11
    000. .... .... .... = Priority: Best Effort (default) (0)
    ...0 .... .... .... = CFI: Canonical (0)
    .... 0000 0000 1011 = ID: 11
  Type: IP (0x0800)
Internet Protocol Version 4, Src: 192.168.11.1 (192.168.11.1), Dst: 192.168.11.2 (192.168.11.2)
Internet Control Message Protocol

```

- 4byte짜리 IEEE 802.17 트렁크 프로토콜

```

000. .... .... .... = Priority: Best Effort (default) (0)
...0 .... .... .... = CFI: Canonical (0)
.... 0000 0000 1011 = ID: 11
Type: IP (0x0800)

```

- 아이디는 12비트 (0 - 4094)
- PC가 패킷을 받으면 목적지 Mac Address 확인 → VLAN ID 확인 후 decapsulation
- 801.1q TAG (붙인다고 해서 tag라고 불린다)

SW1, SW2 F0/24 포트에 트렁크 설정:

```
@ SW1, SW2
conf t
int fa0/24
switchport trunk encapsulation dot1q
switchport mode trunk
end
```

트렁크 설정 확인:

```
SW1#show int trunk
Port      Mode      Encapsulation  Status      Native vlan
Fa0/24    on        802.1q         trunking    1

Port      Vlans allowed on trunk
Fa0/24    1-1005

Port      Vlans allowed and active in management domain
Fa0/24    1,11,12,13

Port      Vlans in spanning tree forwarding state and not pruned
Fa0/24    1,11,12,13
SW1#
```

- Mode on : 직접 수동설정됨
- Encalpsulation : 트렁크 프로토콜
- Native VLAN : 나중에
- 기본적으로 모든 VLAN이 트렁크 사용가능 (원래는 1 - 4094)
- VLAN 데이터베이스에 등록된 VLAN만 트렁크 설정 가능

6. PC IP 주소 설정

각각의 VLAN 들은 논리적으로 분리된 네트워크이므로 네트워크 이름이 중복되면 안된다. 즉, IP 주소를 각각의 VLAN 마다 서로 다른 서브넷으로 할당해야 한다. 다음 표를 참고하여 각각의 PC에 IP 주소 정보를 설정한다.

PC 클릭 -> Desktop -> IP Configuration

VLAN	PC	IP/Prefix	Gateway
vlan 11 (192.168.11.0/24)	A	192.168.11.1/24	192.168.11.254
	C	192.168.11.3/24	
vlan 12 (192.168.12.0/24)	B	192.168.12.2/24	192.168.12.254
	D	192.168.12.4/24	
vlan 13 (192.168.13.0/24)	E	192.168.13.5/24	192.168.13.254

7. Inter-VLAN 구성:

- **Vlan + Router 설정 완료 시 Inter-VLAN** 이라고 부른다

1) SW1 F0/10 포트 트렁크 설정

- 라우터를 이용하여 각각 VLAN에 대한 기본 게이트웨이를 구성한다. 스위치는 라우터와 연결된 스위치 포트에 트렁크를 구성하고 라우터는 스위치와 연결된 인터페이스를 서브 인터페이스로 분리하여 트렁크와 VLAN 게이트웨이 IP 주소를 설정한다

```
SW1#conf t
SW1(config)#int fa0/10
SW1(config-if)#switchport trunk encapsulation dot1q
SW1(config-if)#switchport mode trunk
SW1(config-if)#end
SW1#
```

2) R1 F0/0 서브-인터페이스 트렁크 및 VLAN 게이트웨이 IP 주소 설정:

```
@ R1
conf t
int fa0/0
no shutdown!
!
int fa0/0.1
encapsulation dot1q 1
ip address 192.168.100.254 255.255.255.0
!
int fa0/0.11
encapsulation dot1q 11
ip address 192.168.11.254 255.255.255.0
!
int fa0/0.12
encapsulation dot1q 12
ip address 192.168.12.254 255.255.255.0
!
int fa0/0.13
encapsulation dot1q 13
ip address 192.168.13.254 255.255.255.0
!
```

8. Inter-VLAN 구성 확인:

Inter-VLAN 을 구성하면, 각각의 PC 들은 인터넷이 가능하다. 또한, R1 라우팅 테이블에 각각의 VLAN 에 대한 경로가 있기 때문에 서로 다른 VLAN 간에도 유니캐스트도 가능해진다. 만약, 서로 다른 VLAN 간에 유니캐스트 접근을 차단하려면, R1 에서 ACL 를 이용하여 차단해야 한다. A_PC 에서 다른 VLAN E_PC 로 Ping 테스트를 실시한다. (PC 클릭 -> Desktop -> Command Prompt)

```
A_PC>ping 192.168.13.5
Pinging 192.168.13.5 with 32 bytes of data:

Reply from 192.168.13.5: bytes=32 time=1ms TTL=127
Reply from 192.168.13.5: bytes=32 time=0ms TTL=127
Reply from 192.168.13.5: bytes=32 time=0ms TTL=127
Reply from 192.168.13.5: bytes=32 time=0ms TTL=127

Ping statistics for 192.168.13.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

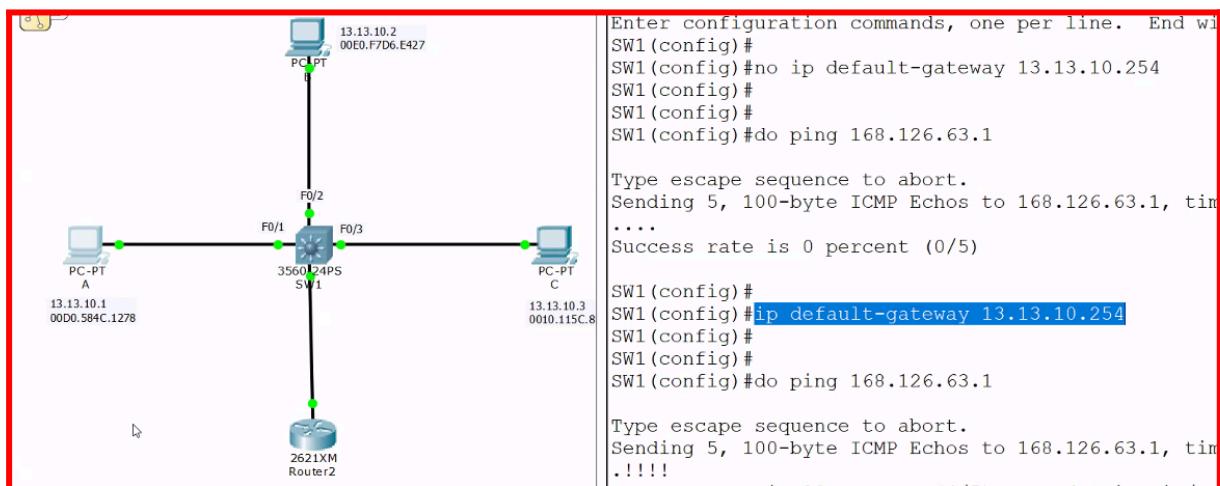
ARP 테이블을 확인하면 VLAN 11 게이트웨이 IP 주소와 MAC 주소가 학습된 것을 확인할 수 있다.

```
A_PC>arp -a
Internet Address      Physical Address      Type
  192.168.11.3          0006.2ad7.0638      dynamic
  192.168.11.254        0001.9610.d701      dynamic
```

VLAN 기능 요약:

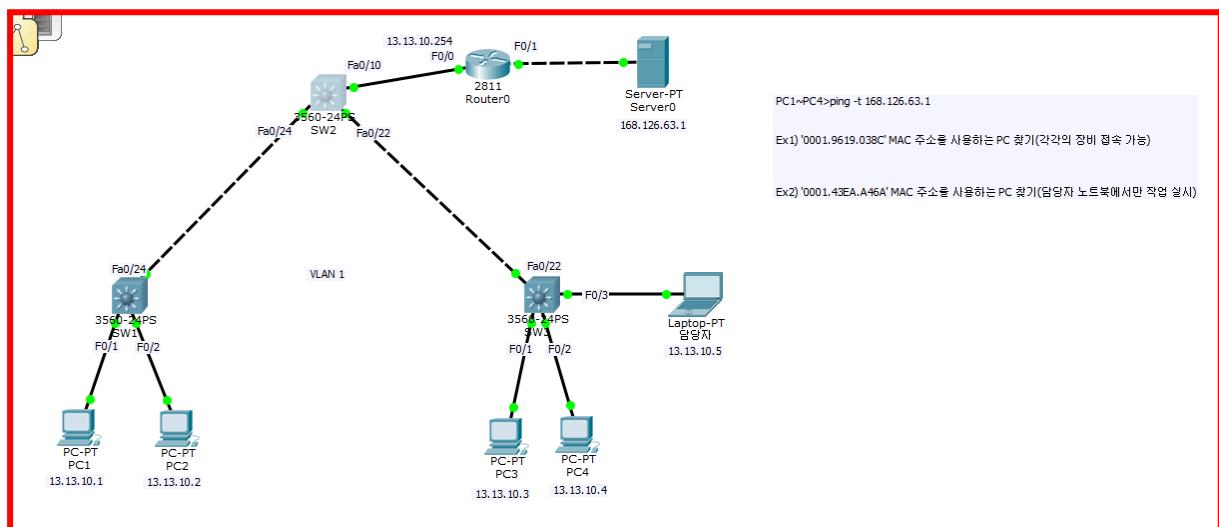


- A, B, C는 같은 VLAN에 있고 등록이 되어 있으므로 통신이 가능하다.
- 따라서 관리자는 스위치에 Telnet 접속을 개인 PC에서 가능하다.
- 관리목적으로 사용이 가능하다.



- 스위치에서도 게이트웨이를 지정할 수 있다.
- 나머지는 라우터랑 다 비슷하다.

Ex1) '0001.9619.038C' MAC 주소를 사용하는 PC 찾기(각각의 장비 접속 가능)



```

Router0
Physical Config CLI
IOS Command Line Interface

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state
up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state
up

R1>en
Password:
R1#
R1#
R1#show arp
Protocol Address Age (min) Hardware Addr Type Interface
Internet 13.13.10.1 3 000B.BE12.90C0 ARPA FastEthernet0/0
Internet 13.13.10.2 3 0001.43EA.A46A ARPA FastEthernet0/0
Internet 13.13.10.3 3 0001.9619.038C ARPA FastEthernet0/0
Internet 13.13.10.4 3 00E0.B049.32C6 ARPA FastEthernet0/0
Internet 13.13.10.254 - 000C.CF94.2401 ARPA FastEthernet0/0
Internet 168.126.63.1 3 000C.CFE0.C069 ARPA FastEthernet0/1
Internet 168.126.63.254 - 000C.CF94.2402 ARPA FastEthernet0/1
R1#

```

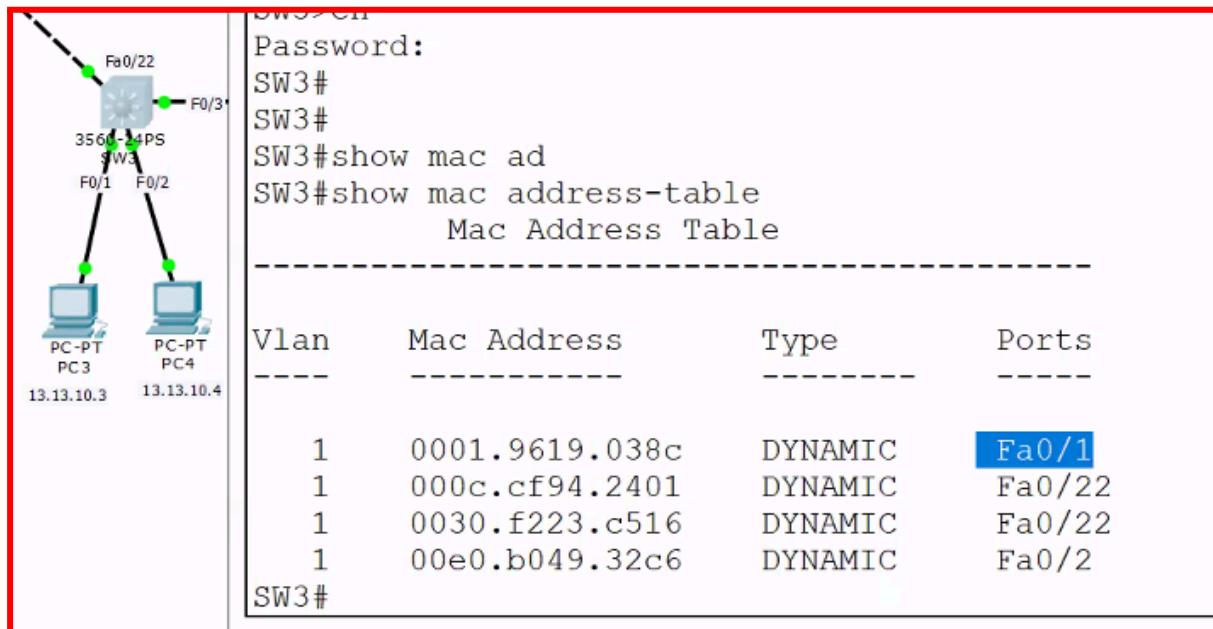
- 위에서부터 아래로 찾아보자

2번 스위치 확인

```
SW2>
SW2>en
Password:
SW2#
SW2#show mac address-table
Mac Address Table
-----
Vlan      Mac Address          Type      Ports
----      -----              ----      -----
1        0001.43ea.a46a    DYNAMIC   Fa0/24
1        0001.9619.038c    DYNAMIC   Fa0/22
1        000b.be12.90c0    DYNAMIC   Fa0/24
1        000c.cf94.2401    DYNAMIC   Fa0/10
1        0090.0cb2.8518    DYNAMIC   Fa0/24
1        00e0.b049.32c6    DYNAMIC   Fa0/22
SW2#
```

- 22번포트에 있다.

3번 스위치 찾자:



- 3번 PC를 찾았다.
- 위에서부터 찾으면 금방 찾는다.

담당자 노트북에서만 찾아보자:

```
PC>telnet 13.13.10.254
Trying 13.13.10.254 ...Open

User Access Verification

Password:
R1>enable cisco
^
* Invalid input detected at '^' marker.

R1>enable cisco
^
* Invalid input detected at '^' marker.

R1>enable
Password:
R1#show arp
Protocol Address Age (min) Hardware Addr Type Interface
Internet 13.13.10.1 19 000B.BE12.90C0 ARPA FastEthernet0/0
Internet 13.13.10.2 19 0001.43EA.A46A ARPA FastEthernet0/0
Internet 13.13.10.3 19 0001.9619.038C ARPA FastEthernet0/0
Internet 13.13.10.4 19 00E0.B049.32C6 ARPA FastEthernet0/0
Internet 13.13.10.5 0 0001.C7E1.E3D5 ARPA FastEthernet0/0
Internet 13.13.10.254 - 000C.CF94.2401 ARPA FastEthernet0/0
Internet 168.126.63.1 19 000C.CFE0.C069 ARPA FastEthernet0/1
Internet 168.126.63.254 - 000C.CF94.2402 ARPA FastEthernet0/1
```

- telnet으로 접속
- 맥 어드레스 찾기

2번스위치 접속:

하지만 VLAN 설정이 되지 않았다.

```
!
interface Vlan1
no ip address
shutdown
!
ip classless
!
ip flow-export version 9
!
```

따라서 관리자 노트북에서 접속 불가능

스위치마다 관리자 설정을 해준다:

```
@ SW1
conf t
int vlan 1
ip address 13.13.10.101 255.255.255.0
no shutdown
end
```

```
@ SW2
conf t
int vlan 1
ip address 13.13.10.102 255.255.255.0
no shutdown
end
```

```
@ SW3
conf t
int vlan 1
ip address 13.13.10.103 255.255.255.0
no shutdown
end
```

2번 스위치:

```
SW2>en
Password:
SW2#show mac
SW2#show mac ad
SW2#show mac address-table
      Mac Address Table
-----
Vlan     Mac Address          Type      Ports
----     -----              -----    -----
1        0001.43ea.a46a    DYNAMIC   Fa0/24
1        0001.9619.038c    DYNAMIC   Fa0/22
1        0001.c7e1.e3d5    DYNAMIC   Fa0/22
1        000b.be12.90c0    DYNAMIC   Fa0/24
1        000c.cf94.2401    DYNAMIC   Fa0/10
1        0090.0ccb2.8518    DYNAMIC   Fa0/24
1        00e0.b049.32c6    DYNAMIC   Fa0/22
SW2#
```

- 이제 접속 가능하다

3번 스위치 접속:

```
password.  
SW3>en  
Password:  
Password:  
SW3#show mac address  
SW3#show mac address-table  
      Mac Address Table  
-----  
Vlan     Mac Address          Type      Ports  
----  -----  -----  
 1    0001.9619.038c  DYNAMIC   Fa0/1  
 1    0001.c7e1.e3d5  DYNAMIC   Fa0/3  
 1    000c.cf94.2401  DYNAMIC   Fa0/22  
 1    0030.f223.c516  DYNAMIC   Fa0/22  
 1    00e0.b049.32c6  DYNAMIC   Fa0/2  
SW3#
```

- 관리자 권한으로 관리자 PC에서 접속 가능하다.

show cdp neighbor

```
User Access Verification  
  
Password:  
R1>en  
Password:  
R1#  
R1#show arp  
Protocol  Address          Age (min)  Hardware Addr  Type  Interface  
Internet  13.13.10.1       26        000B.BE12.90C0  ARPA  FastEthernet0/0  
Internet  13.13.10.2       26        0001.43EA.A46A  ARPA  FastEthernet0/0  
Internet  13.13.10.3       26        0001.9619.038C  ARPA  FastEthernet0/0  
Internet  13.13.10.4       26        00E0.B049.32C6  ARPA  FastEthernet0/0  
Internet  13.13.10.5       7         0001.C7E1.E3D5  ARPA  FastEthernet0/0  
Internet  13.13.10.254     -          000C.CF94.2401  ARPA  FastEthernet0/0  
Internet  168.126.63.1      26        000C.CFE0.C069  ARPA  FastEthernet0/1  
Internet  168.126.63.254    -          000C.CF94.2402  ARPA  FastEthernet0/1  
R1#  
R1#show cdp neighbor  
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge  
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone  
Device ID      Local Intrfce     Holdtme   Capability  Platform  Port ID  
SW2           Fas 0/0          176        S.RP        3560      Fas 0/10  
R1#
```

- 연결상태, 뭐가 연결되어있는지 알 수 있다.

show cdp neighbor detail

```
R1#show cdp neighbor detail  
  
Device ID: SW2  
Entry address(es):  
  IP address : 13.13.10.102  
Platform: cisco 3560, Capabilities:  
Interface: FastEthernet0/0, Port ID (outgoing port): FastEthernet0/10  
Holdtime: 173  
  
Version :  
Cisco IOS Software, C3560 Software (C3560-ADVIPSERVICESK9-M), Version 12.2(37)SE1, RELEASE SOFTWARE  
(fc1)  
Copyright (c) 1986-2007 by Cisco Systems, Inc.  
Compiled Thu 05-Jul-07 22:22 by pt_team  
  
advertisement version: 2  
Duplex: full  
R1#
```

- 더 자세한 정보가 나온다.
 - 운영체제 정보, 관리자 IP 등 (CISCO 장비인지 아닌지도 알수 있다.)

cdp는 보안문제때문에 지금은 잘 사용하지 않는다.

```
SW3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW3(config)#
SW3(config)#no cdp en
SW3(config)#no cdp
% Incomplete command.
SW3(config)#no cdp run
SW3(config)#

```

- CDP 동작을 다 끈다

LLDP 라는 유사한 동작이 있다. (사용을 잘 안한다)

Ex2) '0001.43EA.A46A' MAC 주소를 사용하는 PC 찾기(담당자 노트북에서만 작업 실시)

```

TOS3 Command Line Interface
SW1#show vlan br

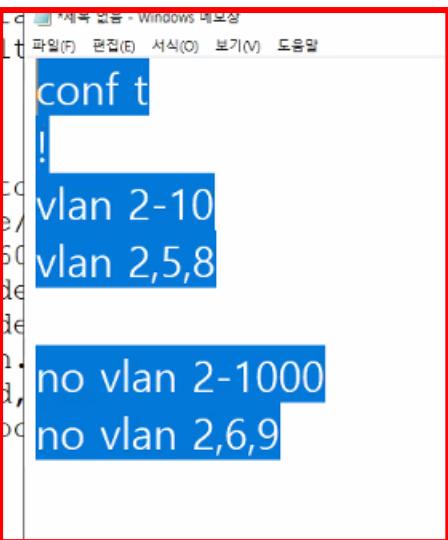
VLAN Name          Status    Ports
----- -----
1     default       active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                           Fa0/5, Fa0/6, Fa0/7, Fa0/8
                           Fa0/9, Fa0/10, Fa0/11, Fa0/12
                           Fa0/13, Fa0/14, Fa0/15, Fa0/16
                           Fa0/17, Fa0/18, Fa0/19, Fa0/20
                           Fa0/21, Fa0/22, Fa0/23, Fa0/24
                           Gig0/1, Gig0/2
11    VLAN_A        active
12    VLAN_B        active
1002  fddi-default  active
1003  token-ring-default  active
1004  fddinet-default  active
1005  trnet-default   active
SW1#show flash

System flash directory:
File  Length  Name/status
3    8662192  c3560-adviservicesk9-mz.122-37.SE1.bin
2    28282    sigdef-category.xml
1    227537   sigdef-default.xml
5    676      vlan.dat
[8918687 bytes used, 55097697 available, 64016384 total]
63488K bytes of processor board System flash (Read/Write)

SW1#

```

- vlan 정보는 show run에 나오지 않고 show vlan brief에 저장된다.
- vlan.dat 파일에 모든 vlan 정보가 따로 저장되기 때문이다.
- flash 메모리는 vlan.dat에 저장되고
- access 정보는 저장이 안된다. (포트 등)



```

conf t
!
vlan 2-10
!
vlan 2,5,8
!
no vlan 2-1000
!
no vlan 2,6,9

```

- 이런식으로 연속적으로 vlan을 한번에 생성도 가능하다.

switchport

```
SW1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)#int fa0/1
SW1(config-if)#switchport mode access
^
* Invalid input detected at '^' marker.

SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 11
SW1(config-if)#do show vlan brief

VLAN Name          Status      Ports
-----  -----
1    default        active     Fa0/2, Fa0/3, Fa0/4, Fa0/5
                               Fa0/6, Fa0/7, Fa0/8, Fa0/9
                               Fa0/10, Fa0/11, Fa0/12, Fa0/13
                               Fa0/14, Fa0/15, Fa0/16, Fa0/17
                               Fa0/18, Fa0/19, Fa0/20, Fa0/21
                               Fa0/22, Fa0/23, Fa0/24, Gig0/1
                               Gig0/2
11   VLAN_A         active     Fa0/1
12   VLAN_B         active
1002 fddi-default   active
1003 token-ring-default active
1004 fddinet-default active
1005 trnet-default   active
SW1(config-if)#switchport access vlan 12
SW1(config-if)#do show vlan brief
```

- 가상환경에서 포트를 바꿔줄 수 있다.
- vlan1 → vlan11

switchport 정보확인:

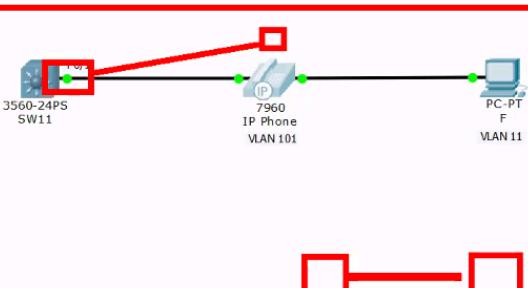
```
SW1#show int fa0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 11 (VLAN_A)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
Trunking VLANs Enabled: All
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL
Protected: false
--More--
```

9. Voice VLAN & Native VLAN

'19-3.Voice VLAN&Native VLAN.pkt' 파일을 실행하여 Voice VLAN 과 Native VLAN 을 구성한다.



- ip 전화기를 통해서 컴퓨터에 연결한다
- 단점은 보안에 조금 취약하다 (음성을 PC로 가는 도중에 가로채기가 가능)



```

name VLAN_PC
vlan 101
name VLAN_Phone
!
int fa0/1
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk native vlan 11
switchport voice vlan 101
end
!
```

- 전화기 폼포트는 트렁크로 기본값이 되어있다.

Untagged Frame

- VLAN 아이디가 안붙어있는 데이터
- **Untagged Frame → Native VLAN 처리**. (.1q는 이걸 처리해준다)

PC → VLAN 11 (untagged frame)

VOIP → VLAN 101 (tagged)

```

SW11# show int fa0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 11
Voice VLAN: 101
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: non
Administrative private-vlan trunk encapsulation: d
Administrative private-vlan trunk normal VLANs: no
Administrative private-vlan trunk private VLANs: n
Operational private-vlan: none
Trunking VLANs Enabled: All
Pruning VLANs Enabled: 2-1001

```

```

name VLAN_PC
vlan 101
name VLAN_Phone
!
int fa0/1
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk native vlan 11
switchport voice vlan 101
end
!
```

Untagged Frame -> Native VLAN 처리

왼쪽(초창기) → 오른쪽(현재)

```
Packet Tracer Student
3560-24PS SW11

@ SW11
conf t
vlan 11
name VLAN_PC
vlan 101
name VLAN_Phone
!
int fa0/1
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk native vlan 11
switchport voice vlan 101
end
!

Untagged Frame -> Native VLAN 처리
```

```
@ SW11

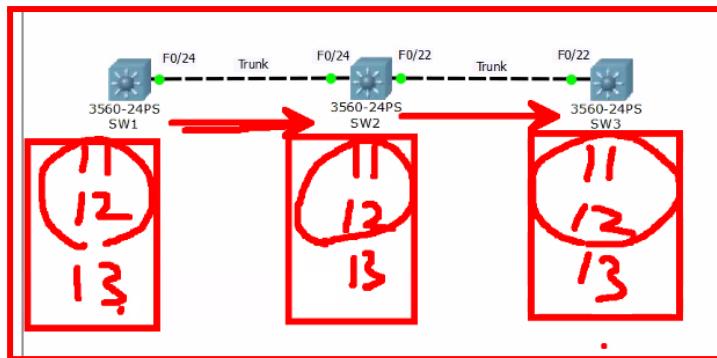
int fa0/1
switchport mode access
switchport access vlan 11
switchport voice vlan 101
end
```

- 더 간편해졌다.

제3장 VTP(Vlan Trunk Protocol)

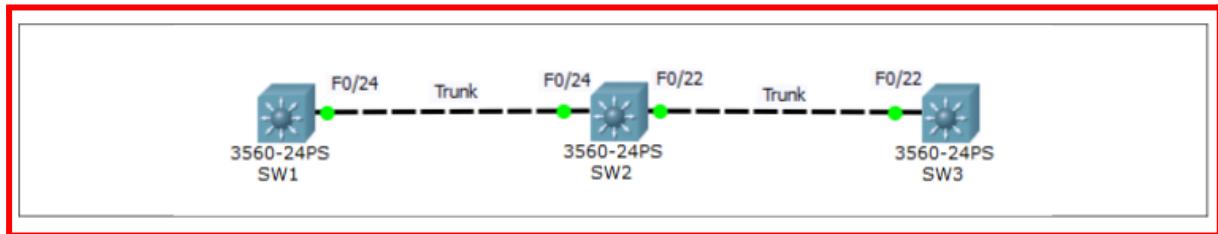
1. VTP(Vlan Trunk Protocol)

- 트렁크로 연결된 스위치들 간에 VLAN 정보를 공유할 수 있도록 메시지를 생성하여 전송하는 프로토콜이다.
- 하나의 스위치에서 VLAN을 생성하거나, 삭제 또는 수정을 실시하면, 트렁크로 연결된 다른 스위치에게 변경된다.
- VLAN 데이터베이스 정보를 공유하는 기능을 담당한다. 이때, VLAN 공유는 동기화 방식으로 진행한다
- Cisco 장비에만 지원된다. (다른회사 제품에서는 지원이 안된다)
- 요즘은 잘 안쓰는 추세다 (vlan 추가시 기존환경이 다 없어진다)



- VLAN 데이터베이스가 추가가 되었을 때 Trunk간 연동되어서 동기화 시켜준다.
- 덮어쓰기 된다.
- 장점:
 - 각각의 스위치마다 vlan을 만들 필요가 없어진다.
 - 트렁크 사용이 되지 않는 VLAN이 안나온다

예제: 20-1 VTP.pkt 파일



트렁크 설정:

```
@SW1
conf t
int fa0/24
switchport trunk encapsulation dot1q
switchport mode trunk
!
```

```
@SW2
conf t
int range fa0/22, fa0/24
switchport trunk encapsulation dot1q
switchport mode trunk
!
```

```
@SW3
conf t
int fa0/22
switchport trunk encapsulation dot1q
switchport mode trunk
!
```

관리용 IP 설정:

```
@SW1
conf t
int vlan 1
ip address 192.168.100.1 255.255.255.0
no shutdown
!
@SW2
conf t
int vlan 1
ip address 192.168.100.2 255.255.255.0
no shutdown
!
@SW3
conf t
int vlan 1
ip address 192.168.100.3 255.255.255.0
no shutdown
```

이름, PW 설정:

```
@ SW1, SW2, SW3
conf t
vtp domain CCNA
vtp password cisco
end
```

VTP 사용 조건

1. 스위치간 트렁크로 구성
2. VTP domain 이름 설정 및 동일
 - a. show vtp status
 - i. VTP domain name 확인
 - ii. 모두 같은 이름이어야 한다.
3. VTP Password 설정
 - a. show vtp password
 - i. 없으면 다 없는상태로

```
SW1#show vtp status
VTP Version : 2
Configuration Revision : 0
Maximum VLANs supported locally : 1005
Number of existing VLANs : 5
VTP Operating Mode : Server
VTP Domain Name : CCNA
VTP Pruning Mode : Disabled
VTP V2 Mode : Disabled
VTP Traps Generation : Disabled
MD5 digest : 0x8C 0x29 0x40 0xDD 0x7F 0x7A 0x63 0x17
Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00
Local updater ID is 192.168.100.1 on interface V11 (lowest numbered VLAN interface
found)
SW1#
```

- 이름 설정 완

```
found)
SW1#show vtp password
VTP Password: cisco
SW1#
```

- 비번 설정 완

동기화 확인법:

```
SW1#show vtp status
VTP Version : 2
Configuration Revision : 0
Maximum VLANs supported locally : 1005
Number of existing VLANs : 5
VTP Operating Mode : Server
VTP Domain Name : CCNA
VTP Pruning Mode : Disabled
VTP V2 Mode : Disabled
VTP Traps Generation : Disabled
MD5 digest : 0x8C 0x29 0x40 0xDD 0x7F 0x7A 0x63 0x17
Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00
Local updater ID is 192.168.100.1 on interface V11 (lowest numbered VLAN interface
found)
SW1#show vtp password
VTP Password: cisco
SW1#
```

- revision 횟수 확인
- vlan database가 변경되면 1씩 증가한다. (이름, vlan 추가, 삭제 등)
- 1씩 증가될때마다 vtp 메세지가 바로 나간다.

VTP 동작 모드

```
SW1#conf t
SW1(config)#vtp mode ?
client      Set the device to client mode.
server      Set the device to server mode.
transparent Set the device to transparent mode.
```

① VTP mode server (기본 모드)

- VLAN 데이터베이스 읽기/쓰기 가능
- VLAN 사용 가능, VLAN 생성/삭제/수정 가능
- VLAN 정보 광고/일치/전달 가능

② VTP mode client

- VLAN 데이터베이스 읽기 가능, 쓰기 불가능
- VLAN 사용 가능, VLAN 생성/삭제/수정 불가능
- VLAN 정보 광고 X, 일치/전달 가능

③ VTP mode transparent

- VLAN 데이터베이스 읽기/쓰기 가능
- VLAN 사용 가능, VLAN 생성/삭제/수정 가능
- VLAN 정보 광고/일치 X, 전달 가능

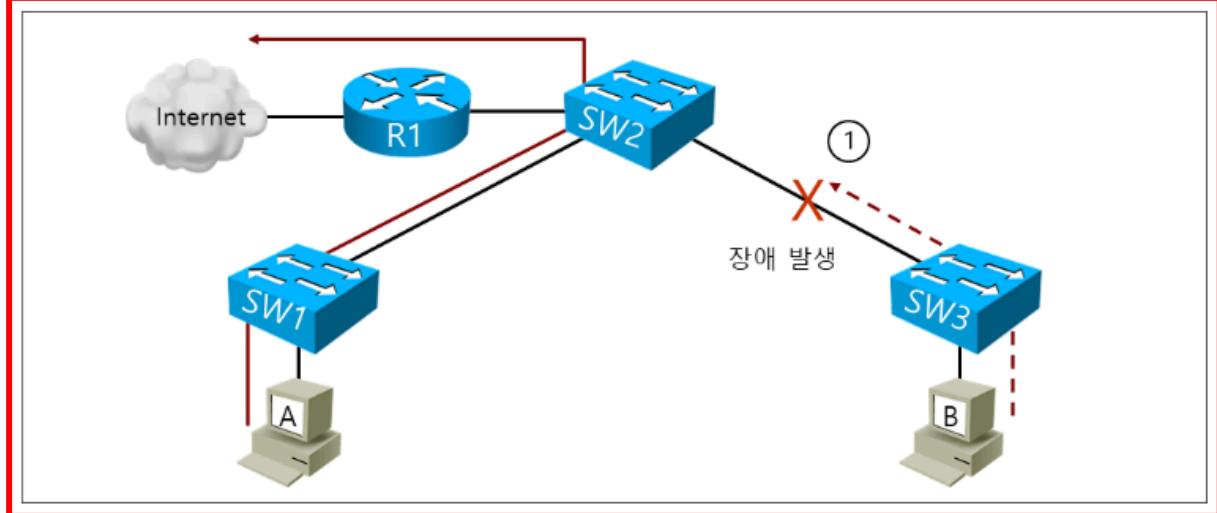
VTP 공유 과정

```
SW1#show vtp status
VTP Version          : 2
Configuration Revision : 0 // Revision 값이 높은쪽으로 VLAN 정보를 동기화함
Maximum VLANs supported locally : 1005
Number of existing VLANs   : 5
~ 중간 생략 ~
```

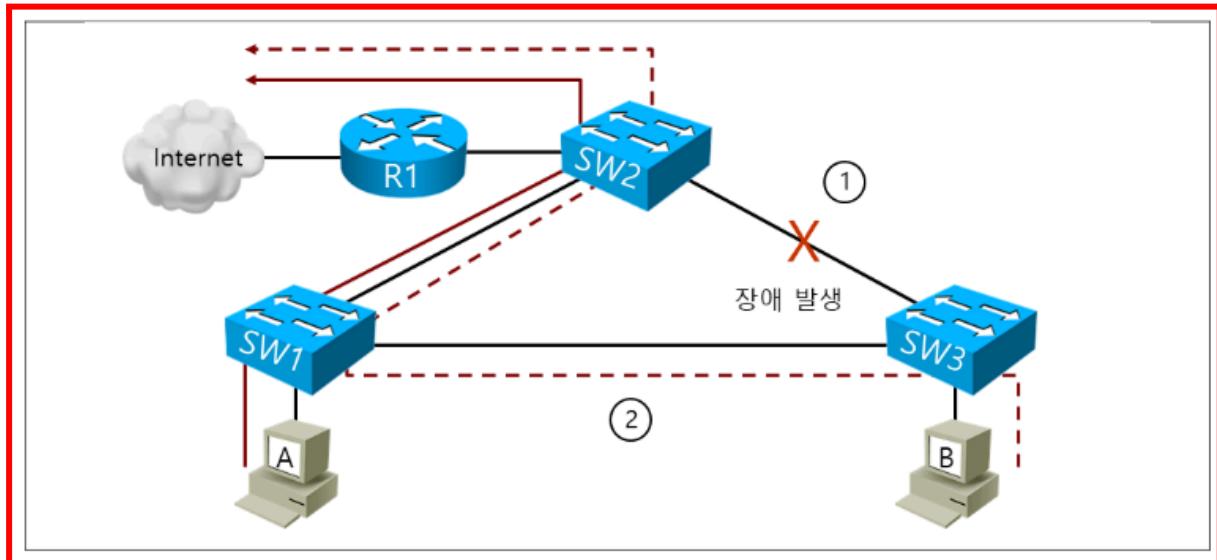
'Configuration Revision'은 VLAN 데이터베이스가 변경될때 마다 '1'씩 증가시키면서 VTP 메세지를 광고한다.
VTP 메세지를 수신한 스위치는 자신의 'Configuration Revision' 값과 비교하여 높은 쪽으로 VLAN 정보를 동기화한다.

제4장 IEEE 802.1d STP

1. 스위치 이중화 링크 필요성



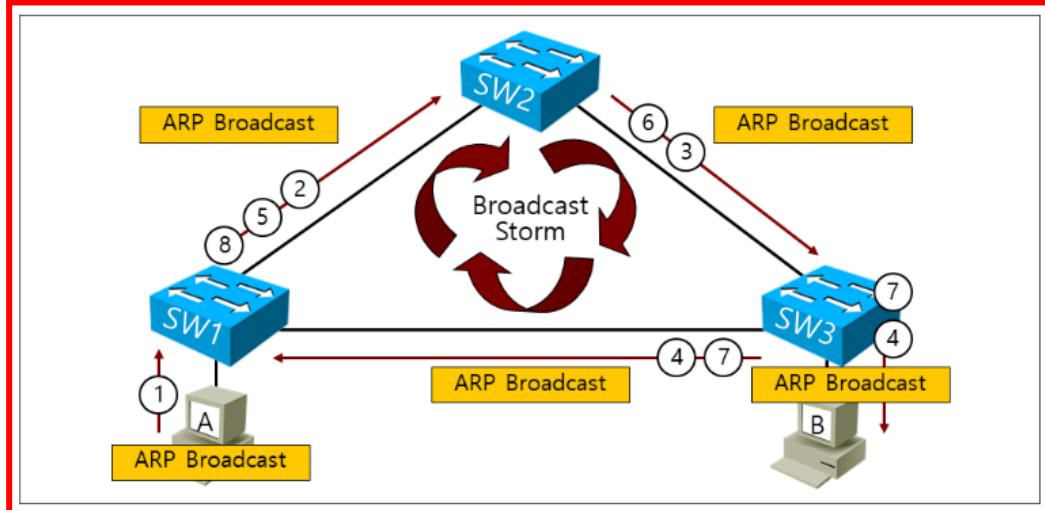
이중화 링크 구성 (케이블 이중화)



2. 브리징 루프

- 스위치는 브로드캐스트를 수신하면 트렁크 포트 또는 같은 VLAN 액세스 포트로 플러딩하기 때문에 이중화링크를 구성할 내부 네트워크 환경에서 브리징 루프가 발생한다.

1) 브로드캐스트 스톰(broadcast storm)



- 스위치는 브로드캐스트를 받으면 모든 방향으로 전달한다.
- 따라서 CPU가 99%까지 올라간다.

2) MAC 플래핑

스위치는 수신한 프레임의 출발지 MAC 주소가 자신의 MAC 주소 테이블에 학습되어 있지 않았다면, 출발지 MAC 주소와 프레임을 수신한 포트 번호를 MAC 주소 테이블에 등록한다. 만약, 특정 포트로 MAC 주소가 학습되어 에이징 타이머가 동작 중일 때, 출발지 MAC 주소가 동일한 이더넷 프레임을 다른 포트로 학습한다면, 다음과 같은 MAC 플래핑 현상이 발생한다.

SW1#

```
1d03h: %SW_MATM-4-MACFLAP_NOTIF: Host 0019.aaff.41c0 in vlan 1 is flapping between port Fa0/24 and port Fa0/20
```

- Mac address 학습이 24 - 20 으로 왔다갔다 반복한다.
- 스위치가 유니캐스트가 한개가 나오기 위해서 계속 업데이트를 실시하기 때문에 일어나는 현상이다.

3) 브리징 루프 원인 및 해결 방법

- 1. 이중화 구성 2. 플러딩
- 해결방안:
 - 이중화 리스트 포기
 - 스위치의 플러딩 기능을 없애기
(하지만 없애지는 못한다- 스위치의 특성)
 - 하지만 브로드캐스팅 요청을 못한다.
- 따라서 결국 해결하려면 스위치의 한쪽 포트는 사용하지 말아야 한다.
(논리적으로 차단)
 - 업인데 프레임 송수신만 되지 않도록 차단 (Blocking)
 - 뭔가 들어오면 버려버리고 나갈려고하면 차단시킨다.
 - 다른 링크가 장애가 생기면 열린다

3. IEEE 802.1d STP(Spanning-Tree Protocol)

- 이 Blocking 포트를 관리해주는 역할을 한다.
- BPDU라는 메시지를 전송하여 물리적인 연결 상태와 내부 토플로지 변경 사항 및 루프의 위치를 결정하여 스위치 포트를 차단한다.
- 아무리 복잡하게 연결해도 Spanning Tree가 자동으로 생성한다.
- 802.1은 날짜
- 포트가 하나라도 업이 되어있으면 spanning tree가 활성화된다.

```
SW2#show spanning-tree vlan1
^
% Invalid input detected at '^' marker.

SW2#show spanning-tree vlan 1
VLAN0001
  Spanning tree enabled protocol ieee
    Root ID    Priority    32769
                Address     0001.9608.DC2A
                Cost         19
                Port        24 (FastEthernet0/24)
                Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

    Bridge ID   Priority    32769  (priority 32768 sys-id-ext 1)
                Address     0030.A383.1759
                Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
                Aging Time  20

  Interface      Role Sts Cost      Prio.Nbr Type
  -----  -----
  Fa0/10        Desg FWD 19      128.10    P2p
  Fa0/22        Altn BLK 19      128.22    P2p
  Fa0/24        Root FWD 19      128.24    P2p

SW2#
```

- Spanning tree 찾는법
- 22번 포트가 blocking 이 되어있다. (Role as alternative)

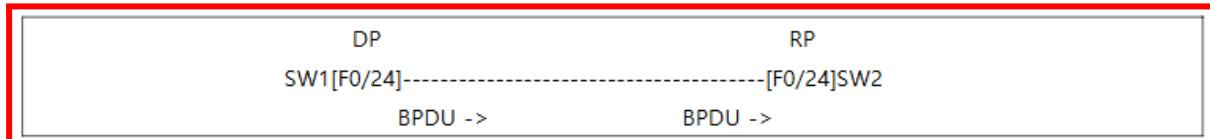
BPDU(Bridge Protocol Data Unit)

- BPDU는 IEEE 802.1d STP를 지원하는 스위치들 사이에서 전송하는 메세지이다.
- STP는 BPDU 메세지를 스위치들 간에 전송하여 루프가 없는 경로를 구성하여 네트워크 토플로지에 대한 모니터링을 실시한다. BPDU는 설정 BPDU(Configuration BPDU)와 TCN BPDU(Topology Change Notification)가 있다

항목	내용
Protocol Identifier	프로토콜 아이디(항상 '0'으로 설정되어 있음)
Protocol Version	STP 버전('0'이면 IEEE 802.1d, '2'이면 IEEE 802.1w RSTP, '3'이면 IEEE 802.1s MSTP이다.)
BPDU Type	BPDU 타입('0x00'이면 설정 BPDU, '0x80'이면 TCN BPDU)
BPDU flags	토플로지 변화를 알리는 BPDU 플래그('0x01'이면 TC, '0x80'이면 TCA)
Root Identifier	루트 브리지로 선출된 스위치의 브리지 아이디
Root Path Cost	루트 브리지까지 Cost 값
Bridge Identifier	루트 브리지로 가는 경로 직전에 있는 스위치의 브리지 아이디
Port Identifier	BPDU를 전송한 해당 스위치의 포트 아이디
Message Age	루트 브리지까지 스위치 개수
Max Age	수신한 BPDU 정보를 보관하는 최대 시간
Hello Time	BPDU 전송하는 주기
Forward Dely	스위치 포트가 포워딩까지 전환되는데 걸리는 시간

STP 포트 유형

- DP(Designated Port) 포트는 BPDU를 송신하는 포트이며, RP(Root Port) 포트는 루트 브리지 쪽에 연결되어 BPDU를 수신하는 포트이다. 루트 브리지까지의 Cost 값이 낮은 포트를 RP(Root Port)로 선정한다.



Cost

스위치에서 사용하는 메트릭이다.

별도의 계산식은 없으며 다음과 같이 이더넷 장치별로 Cost 값이 정해져있다.

스위치 포트	Bandwidth	Cost
Ethernet	10M	100
FastEthernet	100M	19
GigaEthernet	1000M	4
10GigaEthernet	10000M	2

Ex) SW3에서 SW1 까지 Cost 값은 얼마인가?



브리지 아이디, Port ID

4) 브리지 아이디

스위치 식별자이다. 브리지 아이디는 64bit로 되어 있으며, 우선 순위 16bit와 MAC 주소 48bit 조합으로 구성되어 있다. 우선 순위 기본값은 '32768'로 설정되어 있다.

5) Port-ID

스위치 포트마다 갖고 있는 포트 우선 순위 값이다. 우선 순위 기본값은 '128.포트 번호'로 설정되어 있다.

Ex) F0/24 -> 128.24

STP 정보 확인

- show spanning-tree vlan 1

non-root bridge

- 블라킹을 가지고 있는 스위치

root bridge

- 메인 루트 브리지

backup root bridge

- 블라킹을 가지고 있지 않지만 루트는 아닌 브리지

STP를 이용한 스위치 포트 Blocking 과정

1. 루트 브리지 선출

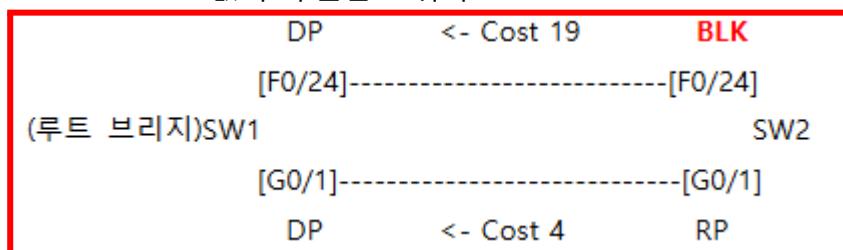
- 브리지 아이디 → 우선 순위가 가장 낮은 스위치
- 브리지 아이디 → MAC 주소가 가장 낮은 스위치
 - BPDU를 스위치끼리 서로 교환해서 더 좋은놈을 루트 브리지로 만든다.

2. DP, RP

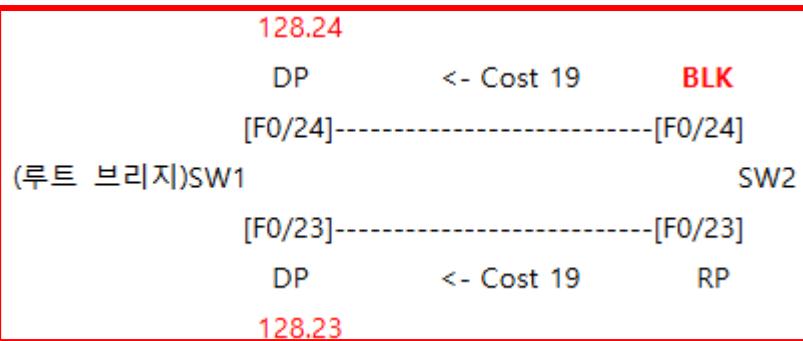
- Cost 값이 더 낮은곳을 RP로 설정한다.
- RP는 루트 브리지와 Cost값이 낮은 스위치 포트

3. Blocking

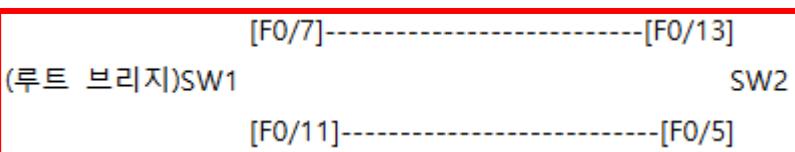
- Root Path Cost 값이 더 높은 스위치 포트



- Cost라는 값을 보고 결정한다.
- Bridge id 값이 높은 스위치의 포트가 blocking이 된다.
- Blocking port 앞에 있는 스위치는 DP이다.



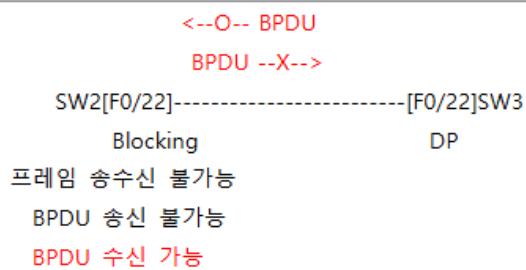
- 같은 Cost 일때는 어떻게 하냐?
 - 포트 아이디 값이 높은 스위치가 Block이 된다.
 - 앞에꺼를 본다.
 - 0/24가 blocking이 된다.



- 이 경우 0/5가 Blocking 이된다.

Blocking 포트

Blocking 포트는 브리징 루프를 방지하기 위해서 논리적으로 차단한 상태이므로 프레임 송수신이 불가능하다. 또한, Blocking 포트는 BPDU 송신이 안되며, 상대방 스위치가 송신한 BPDU 만 수신한다.



STP 루트 브리지 및 Blocking 수동 설정

브리지 아이디의 우선 순위 값을 변경하여 루트 브리지와 Blocking 포트를 변경할 수 있다. 다음 조건에 맞게 루트 브리지와 Blocking 포트를 변경한다.

Root Bridge	SW2
Backup Root Bridge	SW3
Non Root Bridge	SW1(F0/20 BLK)

```
SW2#conf t
SW2(config)#spanning-tree vlan 1 priority 4096      // spanning-tree vlan 1 root primary
SW2(config)#end
```

```
SW3#conf t
SW3(config)#spanning-tree vlan 1 priority 16384      // spanning-tree vlan 1 root secondary
SW3(config)#end
```

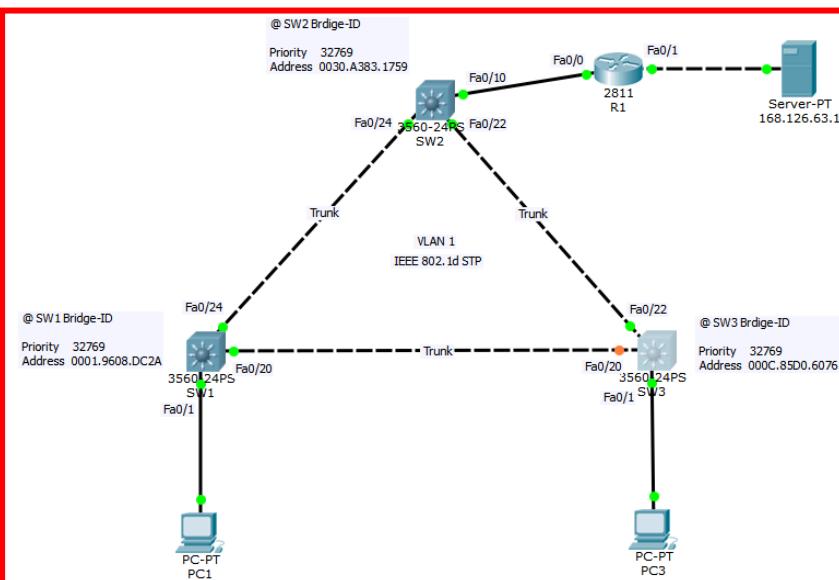
- priority 값을 수정해서 변경시킨다.

```
SW2#show spanning-tree vlan 1
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority  4097
              Address   0030.A383.1759
              This bridge is the root
              Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority  4097  (priority 4096 sys-id-ext 1)
              Address   0030.A383.1759
              Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time 20

  Interface      Role Sts Cost      Prio.Nbr Type
  -----  -----
  Fa0/10        Desg FWD 19       128.10    P2p
  Fa0/22        Desg FWD 19       128.22    P2p
  Fa0/24        Desg FWD 19       128.24    P2p

SW2#
```



STP 타이머

타이머	내용
Hello	BPDUs 전송 주기(기본값 2초)
Forward Delay	스위치 포트가 Forwarding 상태로 전환되는데 필요한 시간(기본값 15초)
Max Age	Blocking 포트로 새로운 BPDUs를 수신하면, 기존의 BPDUs와 비교하는 시간(기본값 20초)

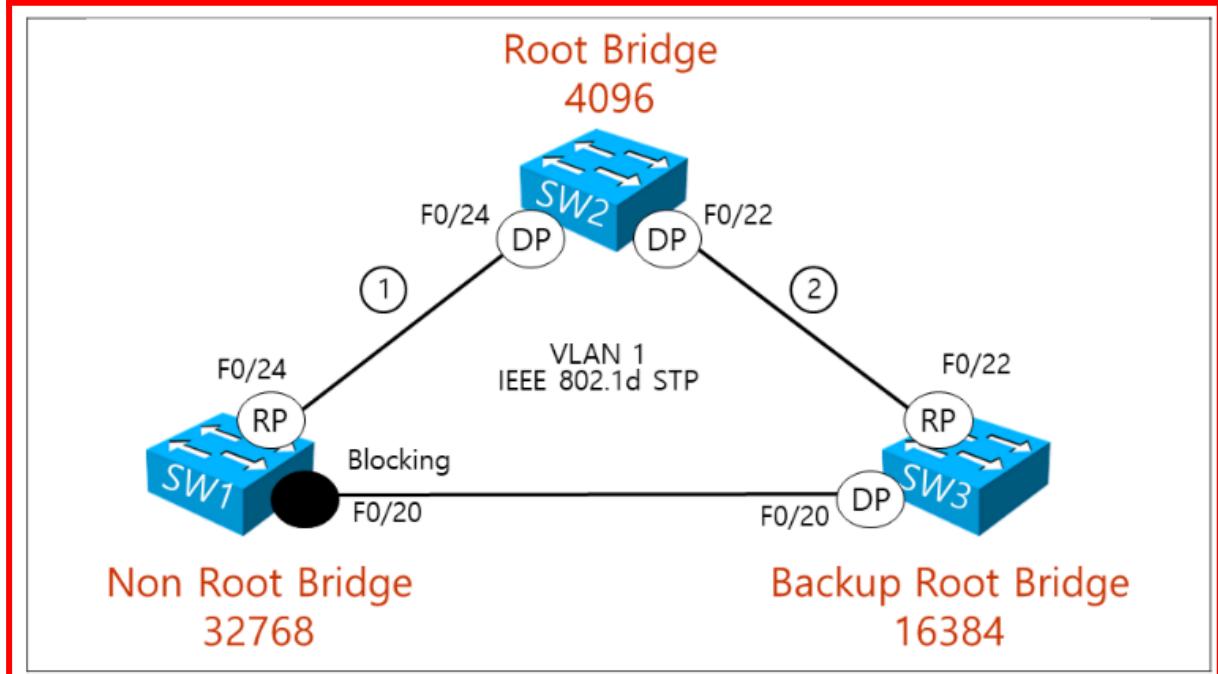
```
SW1#show spanning-tree vlan 1
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    4097
              Address     0030.A383.1759
              Cost         19
              Port        24 (FastEthernet0/24)
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
              Address     0001.9608.DC2A
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  20

  Interface      Role Sts Cost      Prio.Nbr Type
  -----  -----
  Fa0/24        Root FWD 19        128.24   P2p
  Fa0/20        Altn BLK 19        128.20   P2p
  Fa0/1         Desg FWD 19        128.1    P2p

SW1#
```

STP 포트 상태 변환



1 구간에 장애가 발생할 경우

1. SW1에서 STP 디버깅을 실시한다.
2. SW2 F0/24 포트를 'shutdown' 한다.
3. SW1 F0/20 BLK 포트가 Listening(15초) Learning(15초) Forwarding 순으로 이전하는 내용을 확인한다.
4. SW1 F0/20 포트가 Forwarding 상태로 전환 되었는지 확인한다

2 구간에 장애가 발생한 경우

1. SW1에서 STP 디버깅을 실시한다.
2. SW2 F0/22 포트를 'shutdown' 한다.
3. SW1 F0/20 BLK 포트가 Blocking 20초 유지 Listening(15초) Learning(15초) Forwarding 순으로 이전하는 내용을 확인한다.
4. SW1 F0/22 포트가 Forwarding 상태로 전환 되었는지 확인한다

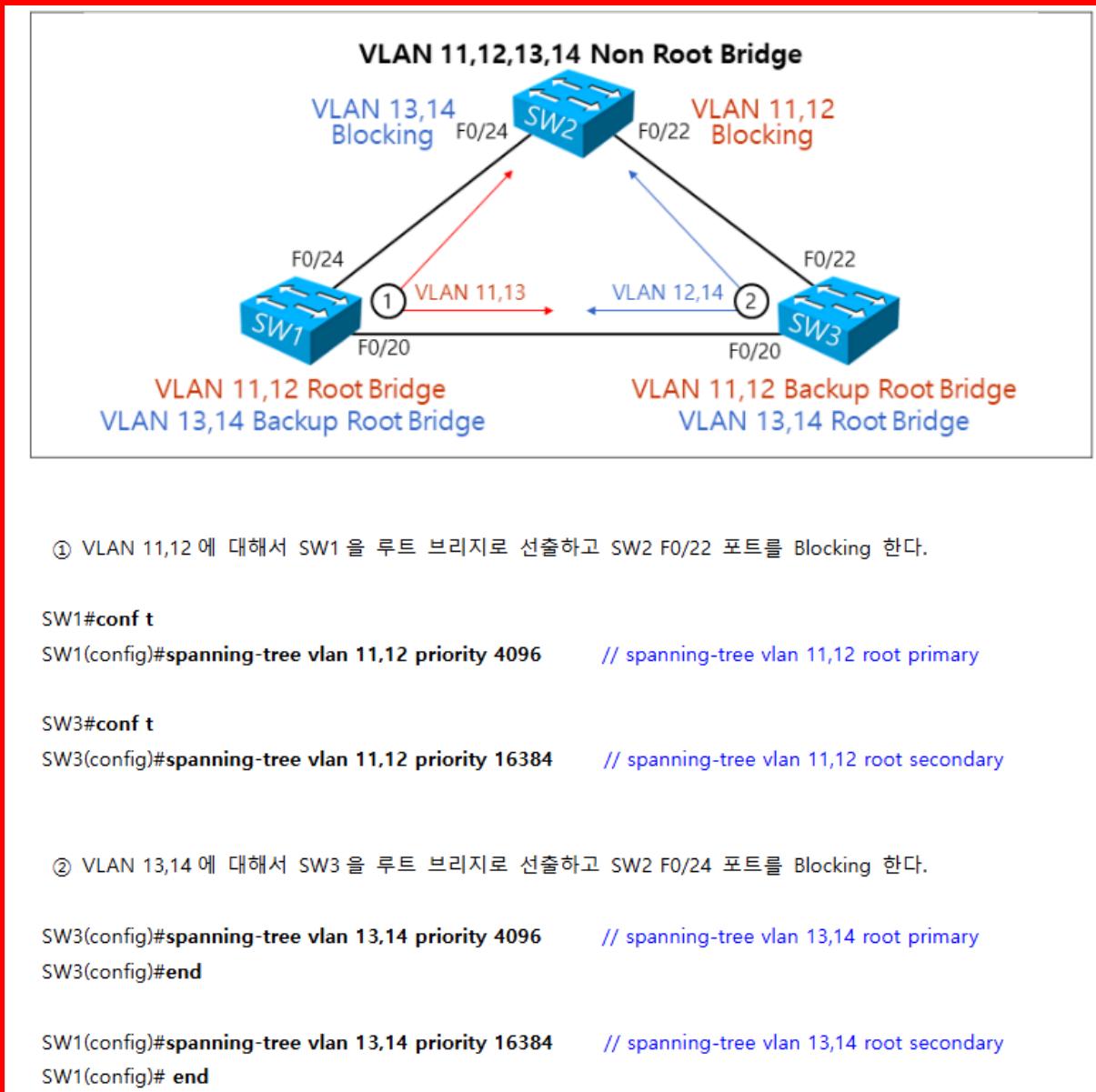
스위치 포트에 노드가 연결될 경우

1. SW1에서 STP 디버깅을 실시한다.
2. PC가 연결된 SW1 F0/1 포트를 'shutdown' 하고 Down 로그 메시지가 출력되면 'no shutdown' 한다.
3. SW1 F0/1 포트가 Listening(15초) Learning(15초) Forwarding 순으로 이전하는 내용을 확인한다.

제6장 PVST (Per VLAN Spanning-Tree)

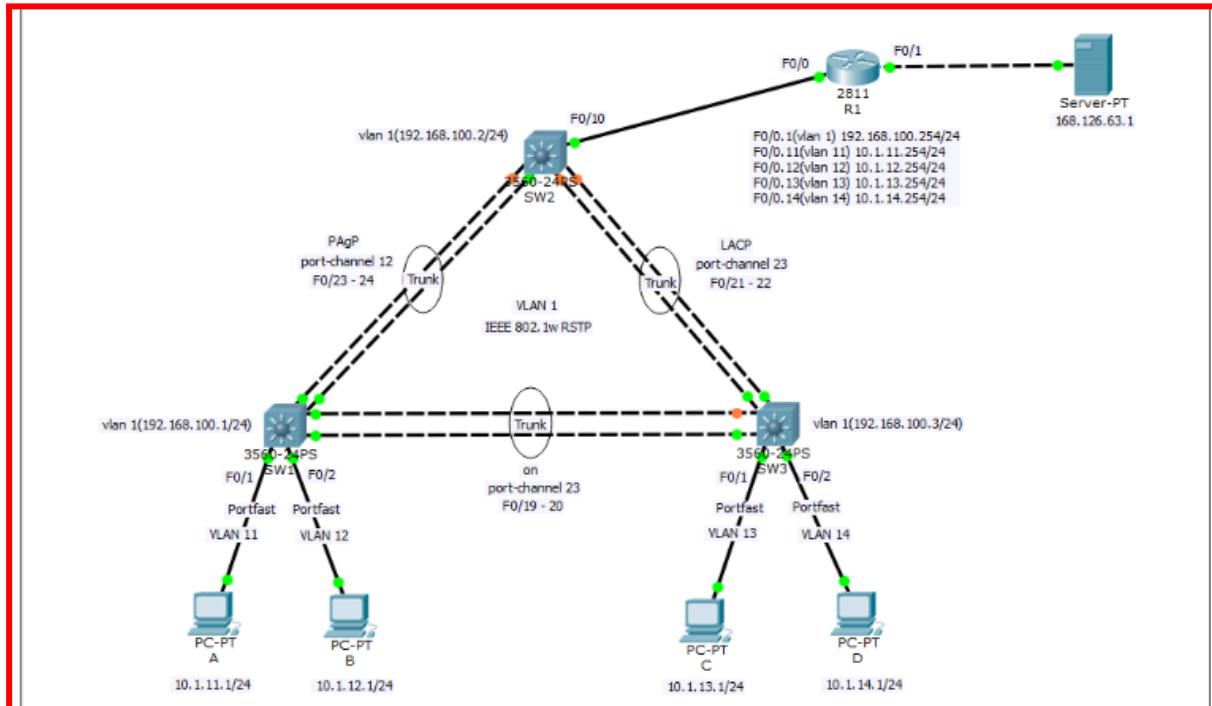
2. PVST(Per VLAN Spanning-Tree)를 이용한 VLAN 로드 분산

- PVST란 각각의 VLAN에 대해서 STP를 지원하는 STP를 의미한다. 그렇기 때문에 각각의 VLAN마다 루트 브리지를 다르게 선출할 수 있으며, 대체 포트도 서로 다르게 설정할 수 있기 때문에 VLAN 로드 분산이 가능하다.



제7장 Etherchannel

이더채널은 여러 개의 스위치 포트를 논리적인 하나의 포트로 구성하여 대역폭을 확장하는 기능이다. 예를 들어 스위치 F0/1~F0/4 포트를 이더채널 구성하면, 400M 대역폭을 제공하는 논리적인 포트를 구성할 수 있다. 또한, F0/1 포트가 장애가 발생되면, 300M 대역폭을 유지하여 사용할 수 있다.



이더채널은 액세스 포트와 트렁크 포트에서 구성이 가능하다. 액세스 포트로 이더채널을 구성할 경우, 스위치 포트의 Speed, Duplex-Mode, VLAN-ID가 동일해야 하며, 트렁크 포트로 이더체널을 구성할 경우, Trunk 프로토콜과 Native VLAN, 트렁크 사용이 가능한 VLAN-ID가 동일해야 한다.

Half-duplex

- 무전기

Full-duplex

- 전화기

2. 이더채널 구성

이더체널을 구성하기 위해서는 'port-channel'이라는 논리적인 인터페이스를 생성하고 이더체널 프로토콜을 이용하여 이더체널 멤버 포트를 협의해야 한다.

프로토콜	동작 모드	내용
PAgP	desirable	상대방 스위치와 협의하여 이더체널을 시작한다.
	auto	상대방 스위치가 desirable 모드인 경우 이더체널을 시작한다.
LACP	active	상대방 스위치와 협의하여 이더체널을 시작한다.
	passive	상대방 스위치가 active 모드인 경우 이더체널을 시작한다.
수동	on	이더체널 프로토콜을 사용하지 않고 이더체널을 시작한다.

이더채널 구성 설정 명령어:

```
@SW1, SW2 (PAgP)
conf t
int port-channel 12
switchport trunk encapsulation dot1q
switchport mode trunk
!
int range fa0/23 - 24
switchport trunk encapsulation dot1q
switchport mode trunk
channel-protocol pagp
channel-group 12 mode desirable
end
```

```
@SW2, SW3 (LACP)
conf t
int port-channel 23
switchport trunk encapsulation dot1q
switchport mode trunk
!
int range fa0/21 - 22
switchport trunk encapsulation dot1q
switchport mode trunk
channel-protocol lacp
channel-group 23 mode active
end
```

```
@SW1, SW3 (수동모드)
conf t
int port-channel 13
switchport trunk encapsulation dot1q
switchport mode trunk
!
int range fa0/19 - 20
switchport trunk encapsulation dot1q
switchport mode trunk
channel-group 13 mode on
end
```

SW1#show etherchannel summary

Flags: D - down P - in port-channel
I - stand-alone s - suspended
H - Hot-standby (LACP only)
R - Layer3 S - Layer2
U - in use f - failed to allocate aggregator
u - unsuitable for bundling
w - waiting to be aggregated
d - default port

Number of channel-groups in use: 2

Number of aggregators: 2

Group	Port-channel	Protocol	Ports
12	Po12(SU)	PAgP	Fa0/23(P) Fa0/24(P)
13	Po13(SU)	-	Fa0/19(P) Fa0/20(P)

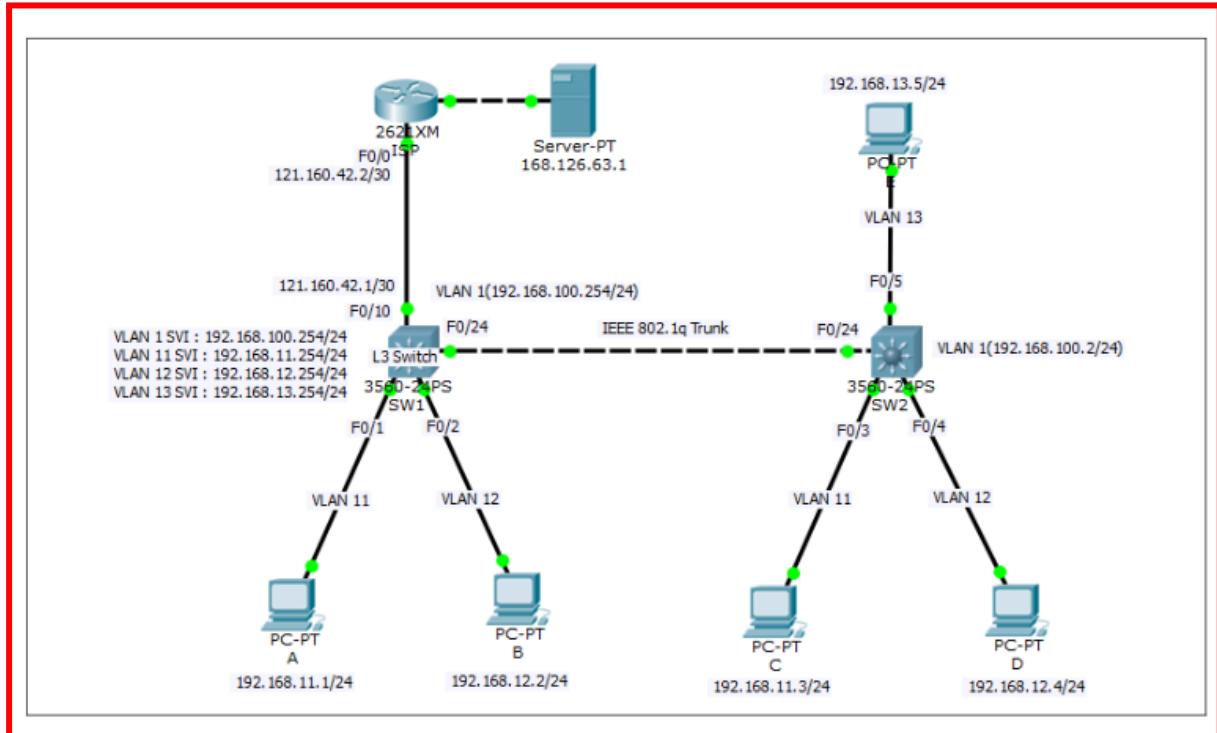
- 설정 확인 명령어

제8장 L3 스위치

L3 스위치:

- L2 스위치 → L3 스위치로 동작
- Multilayer switch 라고도 함
- 라우터처럼 라우팅이 가능한 멀티레이어 스위치
 - 라우팅 테이블
 - 게이트웨이 수행
 - VLAN routing
- 게이트웨이의 역할을 할 뿐만 아니라 각각의 VLAN과 통신이 가능하다.

ex)



vlan 설정 이후 SW1, 2에 라우팅 설정

```
conf t
!
ip routing // ip 라우팅 활성화
!
int vlan 1
ip address 192.168.100.254 255.255.255.0
no shutdown
!
int vlan 11 // SVI 인터페이스
ip address 192.168.11.254 255.255.255.0
!
int vlan 12
ip address 192.168.12.254 255.255.255.0
!
int vlan 13
ip address 192.168.13.254 255.255.255.0
!
```

```

int vlan 1
ip address 192.168.100.254 255.255.255.0
no shutdown
!
int vlan 11 // SVI 인터페이스
ip address 192.168.11.254 255.255.255.0
!
int vlan 12
ip address 192.168.12.254 255.255.255.0
!
int vlan 13
ip address 192.168.13.254 255.255.255.0
!
int fa0/10 // Routed 인터페이스(L3 포트)
no switchport
ip address 121.160.42.1 255.255.255.252
!
```

- Routed 인터페이스 설정

```

SW1(config)#router
% Incomplete command.
SW1(config)#router ?
  bgp    Border Gateway Protocol (BGP)
  eigrp  Enhanced Interior Gateway Routing Protocol (EIGRP)
  ospf   Open Shortest Path First (OSPF)
  rip    Routing Information Protocol (RIP)
SW1(config)#router
```

- 라우팅 프로토콜도 사용 가능

```

SW1(config)#ip rout
SW1(config)#ip route ?
  A.B.C.D Destination prefix
SW1(config)#ip route 0.0.0.0 0.0.0.0 121.160.42.2
SW1(config)#end
SW1#
%SYS-5-CONFIG_I: Configured from console by console
```

- 동적 경로 설정도 가능

라우터 사용하듯이 사용하면 된다.

NAT 설정

```
@SW1
conf t
access-list 10 permit 192.168.0.0 0.0.255.255
!
ip nat inside source list 10 interface fa0/10 overload
!
ip nat outside
!
int vlan 10
ip nat inside
!
int vlan 1
ip nat inside
!
int vlan 11
ip nat inside
!
int vlan 12
ip nat inside
!
int vlan 13
ip nat inside
end
```

DHCP 설정

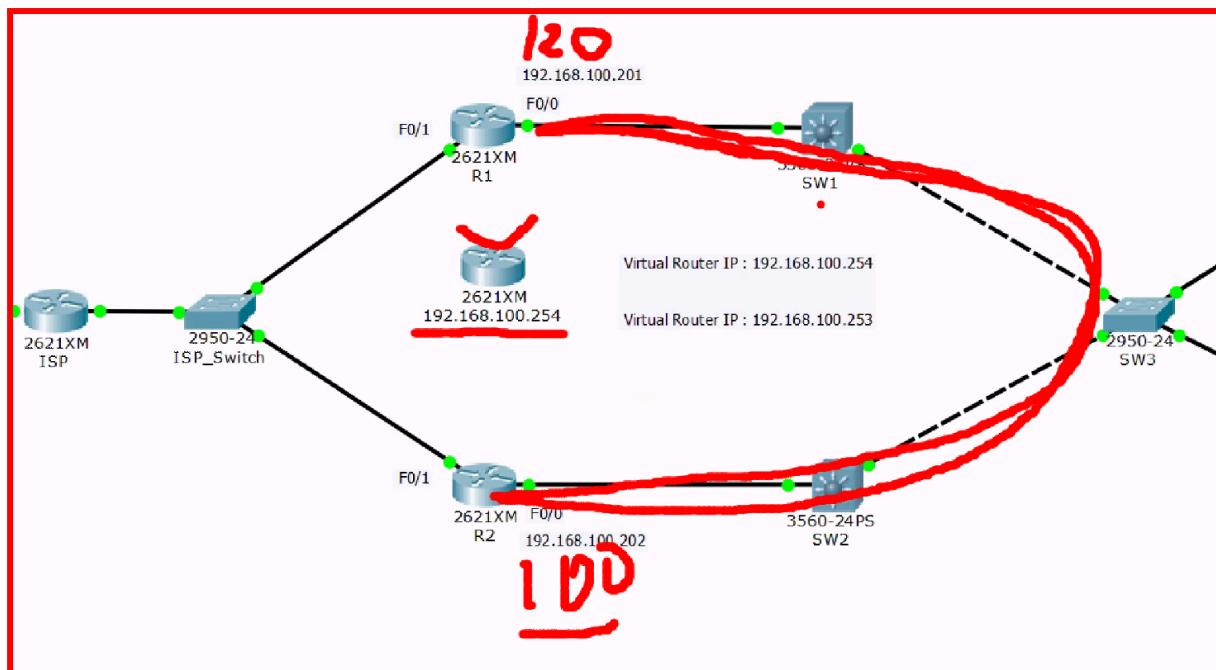
```
@SW1
conf t
ip dhcp excluded-address 192.168.11.254
ip dhcp excluded-address 192.168.12.254
ip dhcp excluded-address 192.168.13.254
!
ip dhcp pool VLAN11
network 192.168.11.0 255.255.255.0
default-router 192.168.11.254
dns-server 168.126.63.1
!
ip dhcp pool VLAN12
network 192.168.12.0 255.255.255.0
default-router 192.168.12.254
dns-server 168.126.63.1
!
ip dhcp pool VLAN13
network 192.168.13.0 255.255.255.0
default-router 192.168.13.254
dns-server 168.126.63.1
!
```

제9장 HSRP (Hot Standby Router Protocol) 이중화

1. HSRP를 이용한 이중화 구성 (Cisco 전용 프로토콜)

내부 시스템들은 인터넷과 같은 외부 네트워크로 패킷을 전송할 때, 게이트웨이를 이용하여 전송한다. 만약, 게이트웨이 장애가 발생하면, 내부 시스템들은 외부 네트워크 접근이 불가능하기 때문에 게이트웨이를 2개 이상 구축하여 사용하는 것을 권장한다. 게이트웨이를 2개 이상 구축하여 가용성이 보장된 내부 네트워크 환경을 게이트웨이 이중화(Gateway Redundancy) 환경이라고 한다.

HSRP는 시스코에서 개발한 게이트웨이 이중화 기능이다. HSRP는 게이트웨이를 수행하는 라우터가 장애가 발생하면, 대기하고 있는 라우터가 즉각적으로 게이트웨이를 수행할 수 있도록 처리한다.



- R1, R2 라우터 HSRP hello message 교환
- R1, R2 라우터 우선순위 비교 (기본값 100)
- R1 라우터가 virtual router를 땡겨간다 (기본값 105라고 가정)
 - R1 라우터는 Active 라우터가 된다.
 - R2 라우터는 Standby 라우터가 된다.
- A, B PC는 virtual 라우터를 gateway로 지정한다.
- R1라우터에 장애가 생기면 R2가 라우터를 땡겨간다 (우선순위 -10)
 - R2 → Active / R1 → Standby
 - Gateway는 안변한다.
 - Arp업뎃을 안해도 되어서 지연시간이 적다.
- 다시 장애가 복구가 될시 R1라우터가 Active로 변한다 (우선순위 +10)

HSRP는 Cisco 전용 프로토콜이다.

- 다른 장비들은?
- VRRP (virtual router redundancy protocol)
- 하나는 있다.

preempt 키워드

- 장애가 복구되었을때 다시 virtual 라우터를 가져올려면 필요하다.

HSRP Track 설정

```
R1#  
R1#  
R1#show int fa0/1  
FastEthernet0/1 is up, line protocol is up  
Hardware is Lance, address is 0001.6492.48  
Internet address is 121.160.31.1/24  
MTU 1500 bytes, BW 100000 Kbit, DLY 100 us  
reliability 255/255, txload 1/255, rxlo  
Encapsulation ARPA, loopback not set  
ARP type: ARPA, ARP Timeout 04:00:00,  
Last input 00:00:08, output 00:00:05, output  
Last clearing of "show interface" counters  
Input queue: 0/75/0 (size/max/drops); Total  
Queueing strategy: fifo  
Output queue :0/40 (size/max)  
5 minute input rate 0 bits/sec, 0 packets/  
5 minute output rate 0 bits/sec, 0 packets  
    0 packets input, 0 bytes, 0 no buffer  
    Received 0 broadcasts, 0 runts, 0 giant  
    0 input errors, 0 CRC, 0 frame, 0 overrun  
    0 input packets with dribble condition  
    0 packets output, 0 bytes, 0 underruns  
    0 output errors, 0 collisions, 1 interf  
    0 babbles, 0 late collision, 0 deferred  
    0 lost carrier, 0 no carrier  
--More--  
@ R1  
conf t  
track 10 interface fa0/1 line-protocol  
!  
int fa0/0  
standby 1 ip 192.168.100.254  
standby 1 priority 105  
standby 1 preempt  
standby 1 track 10 decrement 30  
end  
!
```

- interface fa0/1가 다운되면
- track 10번을 30 빼겠다(우선순위)

```
@ R1  
conf t  
int fa0/0  
standby 1 ip 192.168.100.254  
standby 1 priority 105  
standby 1 preempt  
standby 1 track fa0/1  
end  
!
```

```
@ R2  
conf t  
int fa0/0  
standby 1 ip 192.168.100.254  
standby 1 preempt  
end  
!
```

- 같은 그룹 번호여야한다.

설정확인:

```
R1#show standby br
          P indicates configured to preempt.
          |
Interface  Grp  Pri  P State      Active           Standby           Virtual IP
Fa0/0       1     105 P Active    local            192.168.100.202 192.168.100.254
R1#
```

- P → Preempt
- R1 → Active (local)

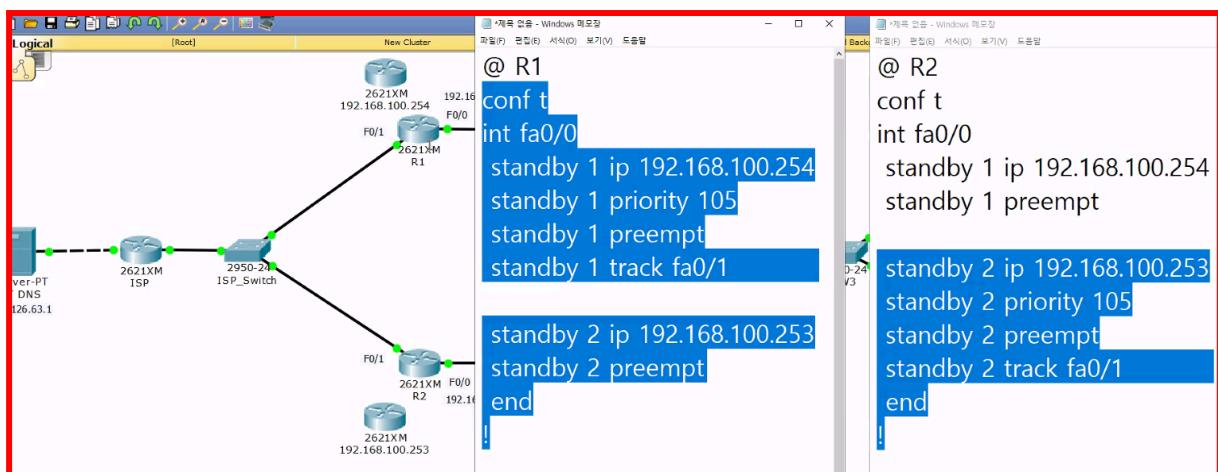
```
R2#show standby br
          P indicates configured to preempt.
          |
Interface  Grp  Pri  P State      Active           Standby           Virtual IP
Fa0/0       1     100 P Standby   192.168.100.201 local            192.168.100.254
R2#
```

- R2 → Standby
- 201 is active

show standby fa0/0

```
R1#show standby fa0/0
FastEthernet0/0 - Group 1 (version 2)
  State is Active
    6 state changes, last state change 00:28:39
  Virtual IP address is 192.168.100.254
  Active virtual MAC address is 0000.0C9F.F001
    Local virtual MAC address is 0000.0C9F.F001 (v2 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 1.055 secs
  Preemption enabled
  Active router is local
  Standby router is 192.168.100.202
  Priority 105 (configured 105)
    Track interface FastEthernet0/1 state Up decrement 10
  Group name is hsrp-Fa0/0-1 (default)
R1#
```

- 더 상세하게 보는 명령어
- Active
- 6 state change
- virtual router mac address



- 이런식으로 트래픽을 분산시키기 위해서 virtual router를 2개씩 만들수 있다.
- 하나가 고장나면 virtual router 2개를 한개의 라우터로 옮긴다.
- 다시 정상화되면 원래대로 옮긴다.