

9차시 수업

10월 25일 (화) - 송준규

| | | | |
|------|-------|---|---------------|
| 등교수업 | 10/25 | 화 | 16:30 ~ 18:30 |
| 등교수업 | | | |
| 등교수업 | 10/27 | 목 | 15:30 ~ 18:30 |
| 등교수업 | | | |
| 등교수업 | | | |
| 등교수업 | 10/28 | 금 | 15:30 ~ 18:30 |
| 등교수업 | | | |
| 등교수업 | | | |
| 등교수업 | 11/1 | 화 | 16:30 ~ 18:30 |

- } 이번 시간 (Inter-Vlan + Frame-Relay)
- } 다음 시간 (Frame-Relay, Server Service)
- } 다음 다음 시간 (Config)
- } 온라인 강의 대체 예정

-복습 Start-

심화

네트워크 이론 복습: 이제는 모르는 내용이 없길 바랍니다.

Network (Connecting)

LAN(Local) v.s **WAN(Wide)**

Internet (집합체)

Protocol (규칙)

통신 방식 3가지 (Unicast, Multicast, Broadcast)

MAC(2계층) v.s. IP address(3계층)

ARP (MAC<->IP)

OSI 7 Layer v.s. TCP / IP Layer

7th Layer:Application

6th Layer:Presentation

5th Layer:Session 4th Layer:Application

4th Layer:Transport 3rd Layer: Transport

3rd Layer:Network 2nd Layer: Internet

2nd Layer:DataLink 1st Layer: Network Access

1st Layer:Physical

How we count Network

= How we count Broadcast Domain

IP addresses are in same network,

Network-ID have to be Same but

Host filed have to be Different !

PDU: Protocol Data Unit - Header

4th Layer / Segment

3rd Layer / Packet

2nd Layer / Frame

1st Layer / Bit

Ethernet: 동작 방식: CSMA/CD : 눈치껏 알아서

계층별 장비:

3rd Layer / Router

2nd Layer / Switch

1st Layer / Repeater

When we use CrossOver cable and DirectCable?

/ IP 주소 사용 성질 다를때 Direct

/ 같을때 Crossover

IP: Internet Protocol ?

Subnetmask is used to divide an IP address into two

parts -> IP = Network part + Host part

FLSM : 고정 길이 서브네팅 (같은 크기의 서브넷)

VLSM : 가변 길이 서브네팅 (크기가 고정되지 않은 서브넷)

Class A ~ C

Prefix /8, /16, /24 : Subnetmask의 1의 개수

뜯어보기

TCP: 3-way-handshake
reliable +
flow, congestion control

Subnetting 문제 풀이: 꼭 스스로 풀어 보시면 좋겠습니다.

문제가 더 필요한 학생들은, 2차시에 업로드 해둔 문제지 활용 부탁드립니다.
(죄송하지만, 악필입니다. 노력해서 작성했으니, 양해 부탁드립니다.)

1. 다음 2진수를 10진수로, 10진수를 2진수로 표현 하시오.
11100101: 232:

1. 11100101 2진수 → 10진수
 ↳ 3개자리: 224, + 4 + 1 = 229 easy
 10진수 → 2진수.
 232 = 224 + 8
 ↳ 3개자리 + 8
11101000
 10000000: 128
 11000000: 192
 11100000: 224
 11110000: 240
 11111000: 248
 11111100: 252
 11111110: 254
 11111111: 255

2. C class, 각 서브넷에 61개의 호스트가 연결되어야 한다면,
이때 서브넷 마스크와 서브넷 개수를 구하여라.

2. C class → 각 Subnet 61개 host 이데,
 Subnet mask, Subnet 개수
 FLSM 진행!
 1단계: 61 + 2 (network - id, broadcast id)
 2단계: 63 포함하는 2ⁿ 이 되찾음: 64 (=2⁶)
 3단계: host 기준: 뒤에서부터 6칸 ←
 C class: 11111111.11111111.11111111.00000000
 ~~~~~  
 추가된 network id bit = 2bit  
 = 늘어난 Subnet 개수 = 2<sup>2</sup> = 4  
 ↳ 이부분까지 Subnet mask.  
 이때, Subnet mask = 255.255.255.192 (2개자리)

# Subnetting 문제 풀이

3. B class 주소를 가지고 서브넷 마스크 255.255.255.248으로 서브넷을 만들었을 때, 추가된 서브넷의 수와 각 서브넷 당 사용 가능한 호스트의 수를 작성하시오.

3. Bclass  $\rightarrow$  255.255.255.248, 이때  
늘어난 Subnet 개수, 사용가능한 host 개수.

Bclass prefix = /16 (248 = 5개 자리)  
255.255.255.248 prefix = /24+5 = /29

29-16 = 13 : 늘어난 network 비트

$\therefore$  늘어난 Subnet 개수 =  $2^{13}$

/29 means, host bit = 32-29 = 3 bit

$\therefore$  사용가능한 host 개수 =  $2^3 - 2$  (net-id, broad)  
= 8-2 = 6

4. IP 주소가 111.222.33.199(255.255.255.192)이라면, 해당 주소의 네트워크 id는 어떻게 될 것인가?

4. 111.222.33.199 (255.255.255.192)의 net-id  
IP address And Subnetmask = net id  
 $\therefore$  111.222.33. 11000111  
And 192.192.192. 11000000  
 $\hline$  111.222.33.11000000  
 $\therefore$  net id = 111.222.33.192/26

5. IP 주소는 111.222.33.42에, 서브넷 마스크는 255.255.255.248이다. 본 네트워크의 브로드 캐스트 주소는?

5. 111.222.33.42  $\rightarrow$  255.255.255.248 이때,  
broadcast 주소는? (5개 자리)  
1단계: net-id 구하기 (And 연산) / 42 = 32+10  
11.222.33.00101010 2단계: host bit  
And 192.192.192.11111000 '1'로 채우기  
 $\hline$  111.222.33.00101000 : net-id  
00101111 : broadcast =  
net-bit host bit 111.222.33.47/29



# Subnetting 문제 풀이

6. 주식회사 "Sunrin"회사와 사무실에 VLSM이 필요한 시점입니다.  
 192.168.120.0/24 네트워크를 정보팀 60대 / 숲팀 30대 /  
 IT경팀 4대 / 콘디팀 2 대의 PC가 사용 가능하도록 서브네팅합니다.  
 (공식 풀이와 정석 풀이의 결과는 동일합니다.)  
 (공식 풀이또한 정석 풀이를 원리로 두기에, 올바른 접근입니다.)

6번. 공식풀이 → '3기역' 2단계: -3  
 1단계 동일 → 64, 32, 8, 4 → 61, 29, 5, 1  
 (700여개 시작 → 다음 IP) +1  
 ① net-id: 0 사용가능범위: 1~(1+61)62  
 broadcast: 63  
 다음 IP (+1)  
 ② net-id: 64 사용가능범위: 65~(65+29)94  
 broadcast: 95  
 ③ net-id: 96 사용가능범위: 97~(97+5)102  
 broadcast: 103  
 ④ net-id: 104 사용가능범위: 105~(105+1)106  
 broadcast: 107  
 단, 공식은 본인 입맛대로 알아서 사용한다!

6. VLSM, [60대, 30대, 4대, 2대]-정석풀이  
 1단계: +2 (net-id, broadcast)를 포함하는 2<sup>n</sup>의 숫자.  
 62 ≤ 64, 32 ≤ 32, 6 ≤ 8, 4 ≤ 4  
 ① 2<sup>6</sup> ② 2<sup>5</sup> ③ 2<sup>3</sup> ④ 2<sup>2</sup>  
 2단계: host 기준: 뒤에서 접근  
 [다음 IP]  
 ① 192.168.120.0 00000000  
 00 111 111 (2<sup>6</sup>-1) 6칸  
 ② 0 10000000  
 010 111 111 (2<sup>5</sup>-1) 5칸  
 net-id = 192.168.120.0/26 net-id = .64/27  
 broadcast = .63 broadcast = .95  
 사용가능범위 = .1~.62 사용가능범위 = .65~.94  
 [다음 IP]  
 ③ 0 11000000  
 111 3칸  
 net-id = .96/29  
 broadcast = .103  
 사용가능범위 = .97~.102  
 ④ 0 1101000  
 11 2칸  
 net-id = .104/30  
 broadcast = .107  
 사용가능범위 = .105~.106

## 1. IP Settings: 이제 IP 할당은 익숙해 지셨길 바랍니다.

```
en // Router Left
conf t
int se0/0/0
ip add 20.0.0.1 255.255.255.0
```

```
int g0/0
ip add 10.0.0.254 255.255.255.0
```

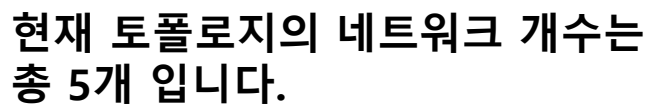
```
en // Router Center
conf t
int se0/0/0
ip add 20.0.0.254 255.255.255.0
```

```
int se0/0/1
ip add 40.0.0.254 255.255.255.0
```

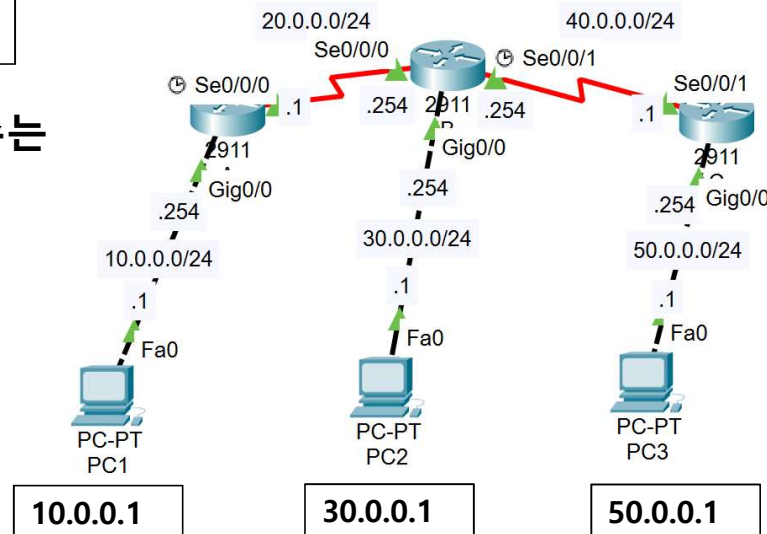
```
int g0/0
ip add 30.0.0.254 255.255.255.0
```

```
en // Router Right
conf t
int se0/0/1
ip add 40.0.0.1 255.255.255.0
```

```
int g0/0
ip add 50.0.0.254 255.255.255.0
```



10.0.0.0/24  
20.0.0.0/24  
30.0.0.0/24  
40.0.0.0/24  
50.0.0.0/24





## 2-1. Static Routing:

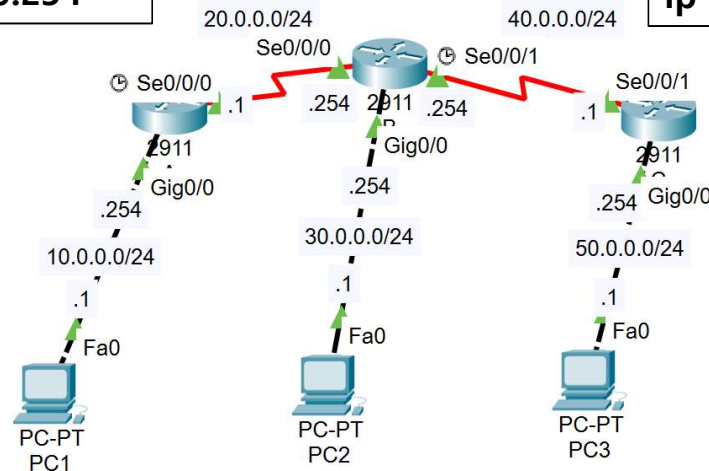
이웃하지 않아 모르는 네트워크를 알려준다고 생각합시다.

```
conf t // Static Routing
ip route [net-id] [sm] [상대편 Router ip]
```

```
conf t // router Center
ip route 10.0.0.0 255.255.255.0 20.0.0.1
ip route 50.0.0.0 255.255.255.0 40.0.0.1
```

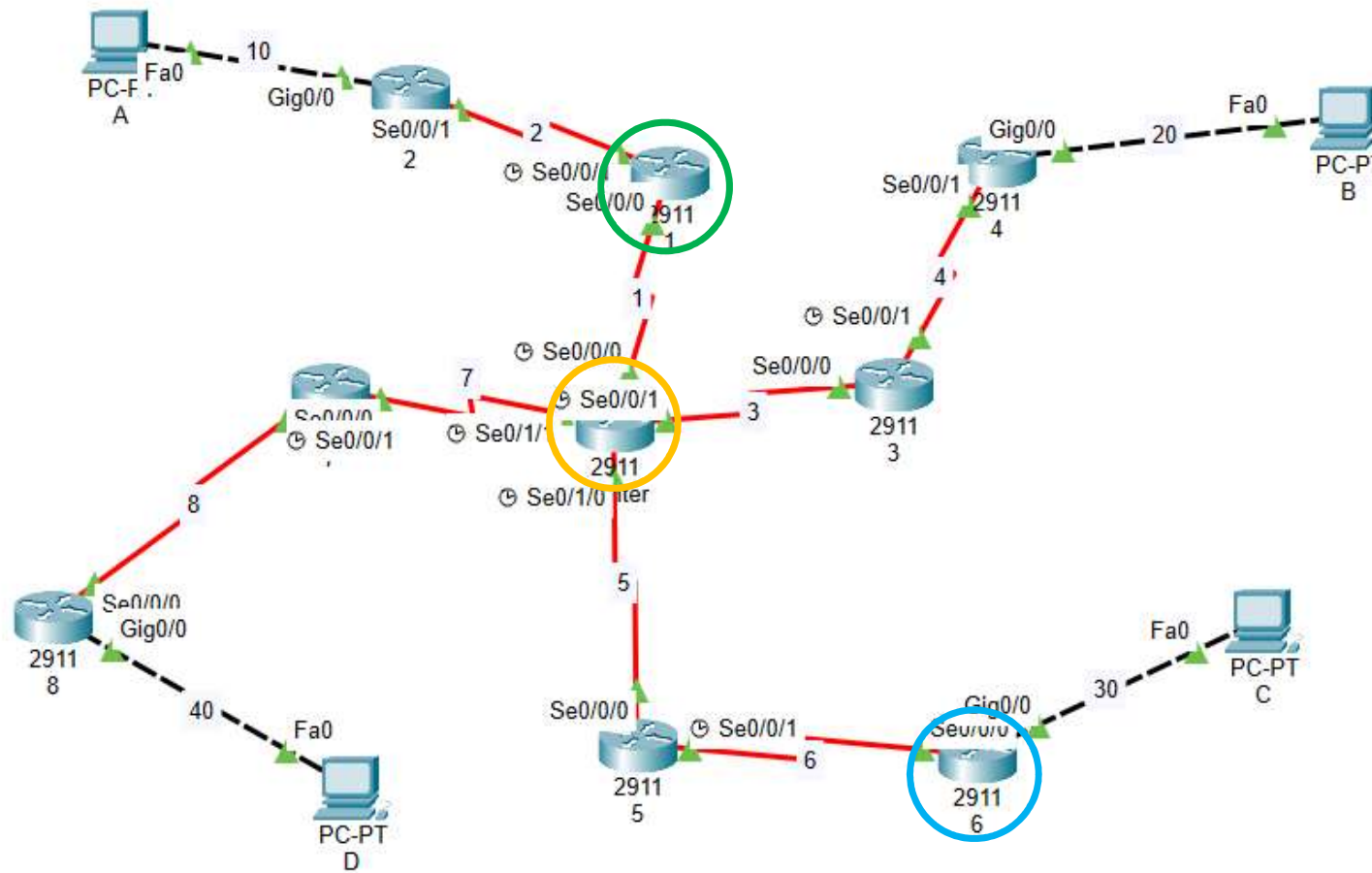
```
conf t // Router Left
ip route 30.0.0.0 255.255.255.0 20.0.0.254
ip route 40.0.0.0 255.255.255.0 20.0.0.254
ip route 50.0.0.0 255.255.255.0 20.0.0.254
```

```
conf t// Router Right
ip route 10.0.0.0 255.255.255.0 40.0.0.254
ip route 20.0.0.0 255.255.255.0 40.0.0.254
ip route 30.0.0.0 255.255.255.0 40.0.0.254
```



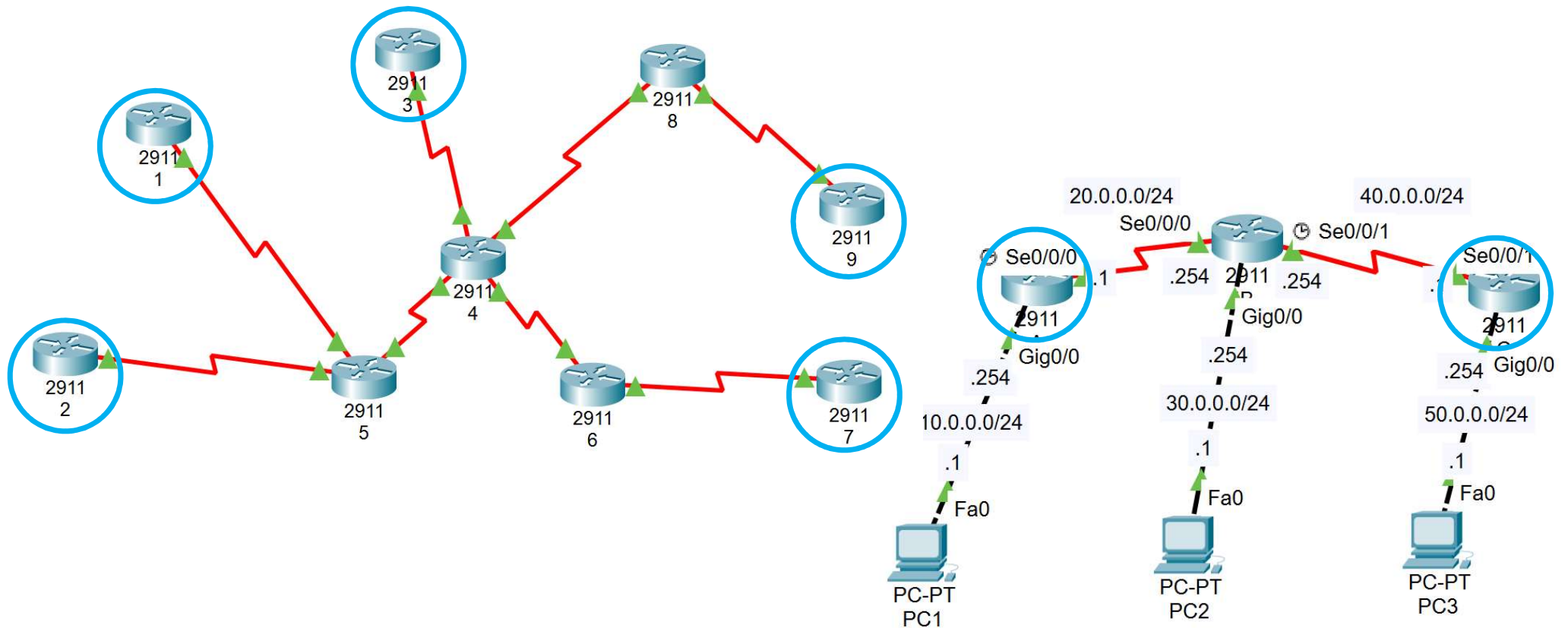
## 2-2. Static Routing

초록, 주황, 파랑 라우터를 기준으로 어떤 네트워크에 대한 Static Routing이 필요한지 생각해 봅니다.



## 2-2. Stub Network / 단말 라우터

아래 토폴로지에서 단말 라우터를 찾아봅니다.



## 2-3. Default Routing

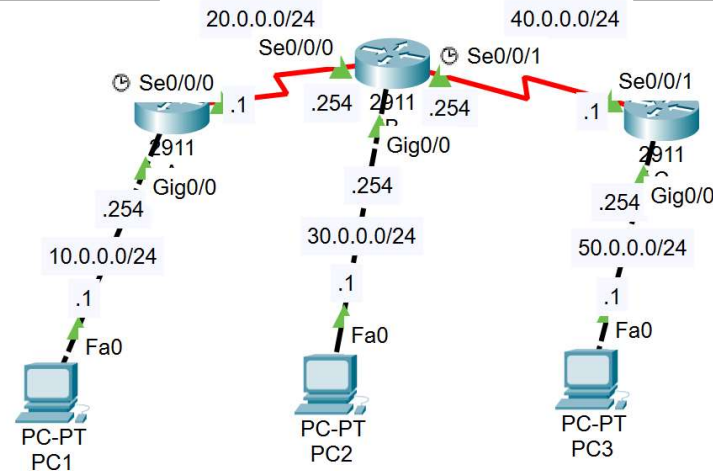
단말 라우터에서 다른 모든 정보가 한 포트만 들어올때 Default Routing을 사용합니다.

```
conf t // Static Default Routing
ip route 0.0.0.0 0.0.0.0 [상대편 Router ip]
```

```
conf t // router Center
ip route 10.0.0.0 255.255.255.0 20.0.0.1
ip route 50.0.0.0 255.255.255.0 40.0.0.1
```

```
conf t // Router Left
ip route 0.0.0.0 0.0.0.0 20.0.0.254
```

```
conf t// Router Right
ip route 0.0.0.0 0.0.0.0 40.0.0.254
```



# 3-1. Dynamic Routing (RIPv2)

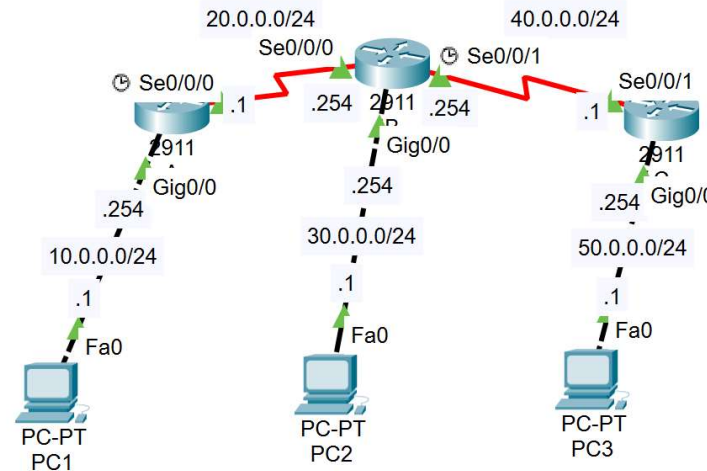
꼭 따로 라우팅 설정 해보시길 바랍니다.

RIPv2 동작 방식  
+  
Loop 방지

```
router rip
version 2 // version 2에서는 VLSM 지원
network [인접한 네트워크 id]
no auto-summary // for VLSM
```

```
conf t // router Center
router rip
version 2
network 20.0.0.0
network 30.0.0.0
network 40.0.0.0
no auto-summary
```

```
conf t // Router Left
router rip
v 2
net 10.0.0.0
net 20.0.0.0
no au
```



```
conf t// Router Right
router rip
v 2
net 40.0.0.0
net 50.0.0.0
no au
```

## 3-2. Dynamic Routing (EIGRP)

꼭 따로 라우팅 설정 해보시길 바랍니다.

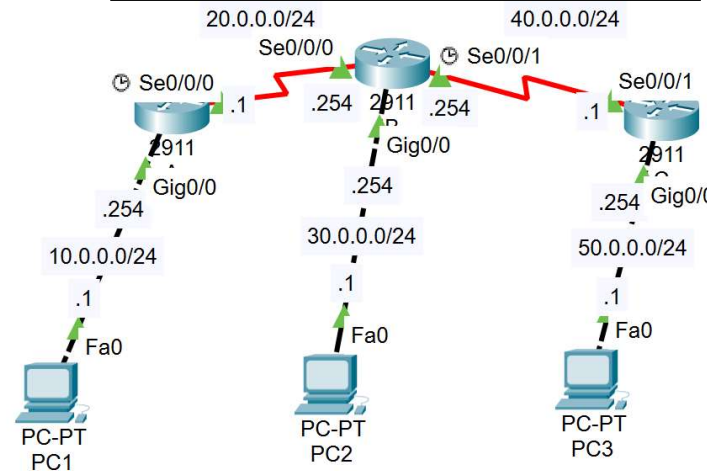
EIGRP 동작 방식  
+  
Metric, DUAL  
(Feasible)  
Successor

```
router eigrp [AS number]  
network [인접한 네트워크 id] [Wildcard mask]  
no auto-summary // for VLSM
```

```
conf t // Router Left  
router eigrp 2  
net 10.0.0.0 0.0.0.255  
net 20.0.0.0 0.0.0.255  
no au
```

```
conf t // router Center  
router eigrp 2  
network 20.0.0.0 0.0.0.255  
network 30.0.0.0 0.0.0.255  
network 40.0.0.0 0.0.0.255  
no auto-summary
```

```
conf t// Router Right  
router eigrp 2  
net 40.0.0.0 0.0.0.255  
net 50.0.0.0 0.0.0.255  
no au
```





# 3-3. Dynamic Routing (OSPF)

꼭 따로 라우팅 설정 해보시길 바랍니다.

OSPF 동작방식  
+  
DR, BDR  
Area

사전 단계: Loopback 주소  
router ospf [process-id]  
router-id [ip address]  
network [인접한 네트워크 id] [Wildcard mask] area [area]

```
conf t // Router Left
int lo 0
ip add 1.1.1.1 255.255.255.0
```

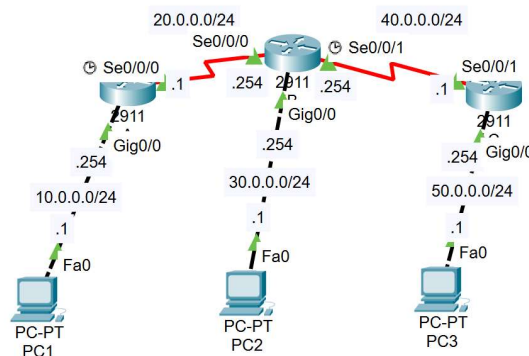
```
router ospf 1
router-id 1.1.1.1
net 10.0.0.0 0.0.0.255 area 0
net 20.0.0.0 0.0.0.255 area 0
```

```
conf t // router Center
int lo 0
ip add 2.2.2.2 255.255.255.0

router ospf 1
router-id 2.2.2.2
network 20.0.0.0 0.0.0.255 area 0
network 30.0.0.0 0.0.0.255 area 0
network 40.0.0.0 0.0.0.255 area 0
```

```
conf t// Router Right
int lo 0
ip add 3.3.3.3 255.255.255.0
```

```
router ospf 1
router-id 3.3.3.3
net 40.0.0.0 0.0.0.255 area 0
net 50.0.0.0 0.0.0.255 area 0
```

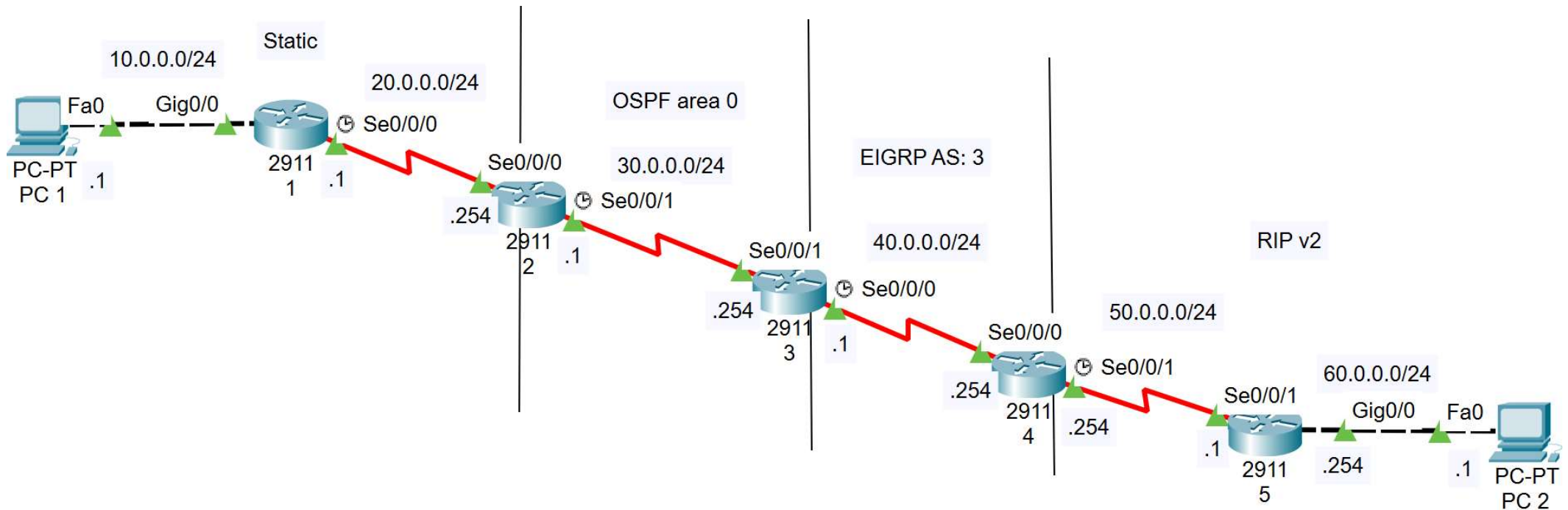


## 4-1. Redistribute (IP Settings)

재분배 설정 또한 꼭 해보시길 바랍니다.

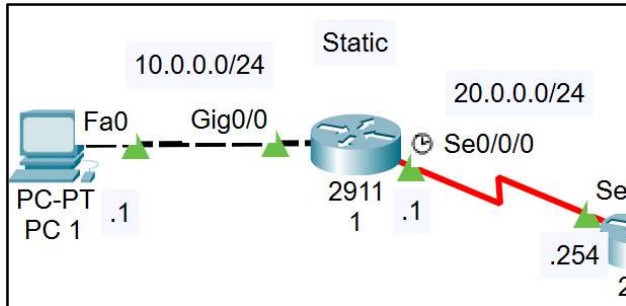
재분배는 서로 다른 Routing Protocol간의 정보 전달을 위해 설정합니다.

우선, 아래와 같이 IP를 부여해 봅니다.



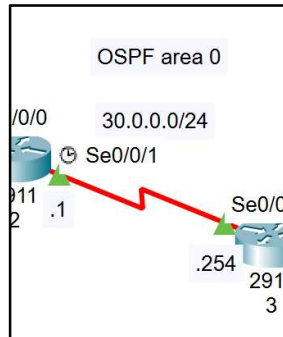
# 4-2-1. Redistribute (Routing)

각 Routing Protocol 마다 Routing을 진행합니다.

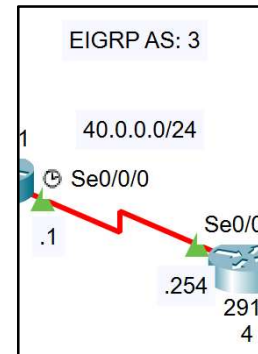


**conf t // Router "1" // 단말 라우터 입니다.**  
**ip route 0.0.0.0 0.0.0.0 20.0.0.254**

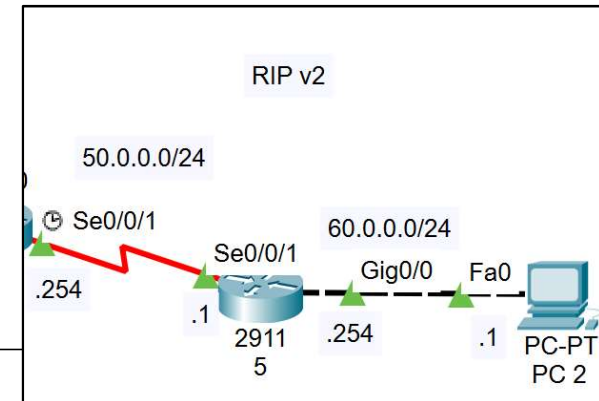
**conf t// Router "2"**  
**ip route 10.0.0.0 255.255.255.0 20.0.0.1**



**conf t // Router "2"**  
**router ospf 1**  
**net 30.0.0.0 0.0.0.255 area 0**

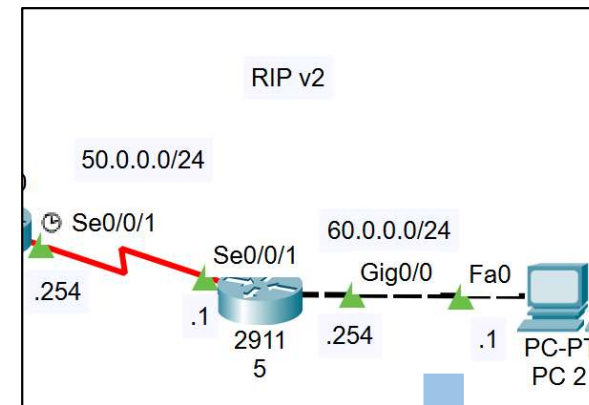
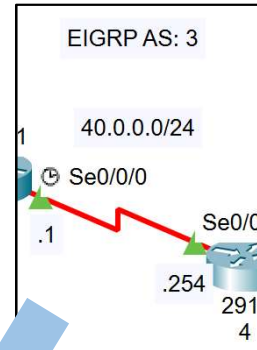
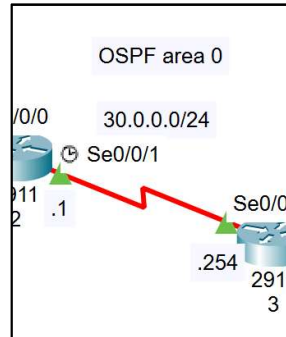
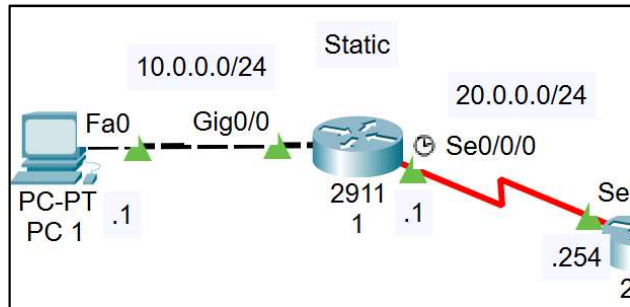


**conf t// Router "3"**  
**router ospf 1**  
**net 30.0.0.0 0.0.0.255 area 0**



## 4-2-2. Redistribute (Routing)

각 Routing Protocol 마다 Routing을 진행합니다.



```
conf t // Router "3"  
router eigrp 3  
net 40.0.0.0 0.0.0.255  
no au
```

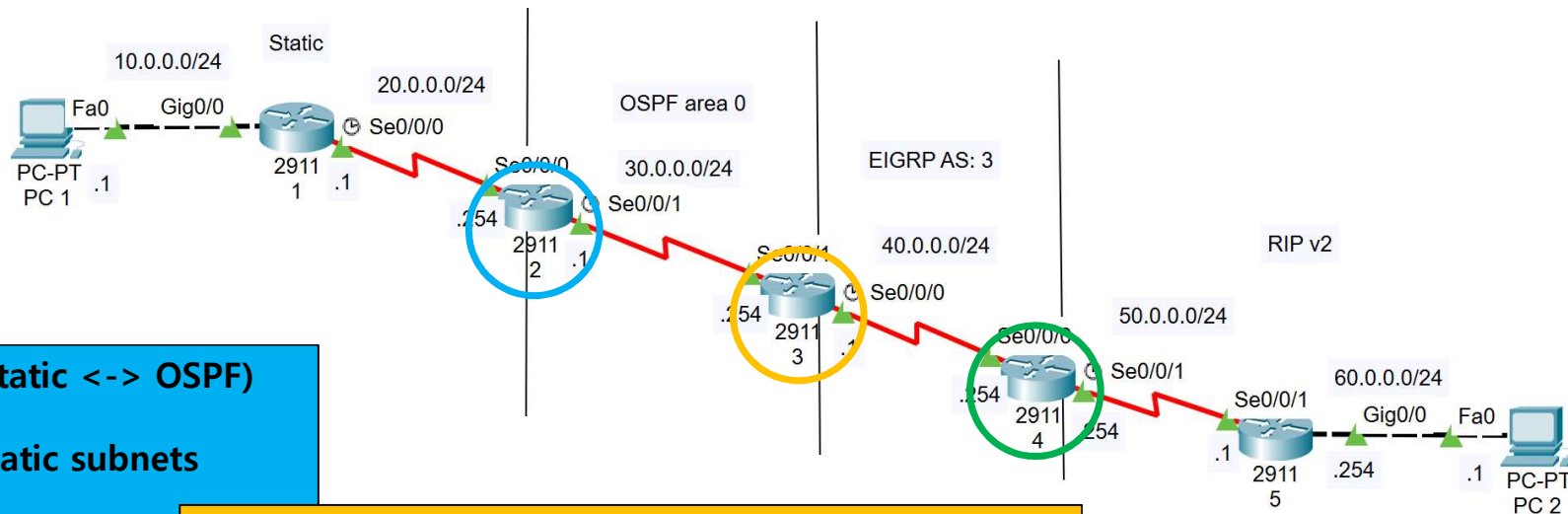
```
conf t// Router "4"  
router eigrp 4  
net 40.0.0.0 0.0.0.255  
no au
```

```
conf t // Router "4"  
router rip  
version 2  
net 50.0.0.0  
no au
```

```
conf t// Router "5"  
router rip  
v 2  
net 50.0.0.0  
net 60.0.0.0  
no au
```

## 4-3. Redistribute (main)

IP Setting과 아래 command만 똑같이 입력해도 PC끼리 통신이 가능합니다.  
꼭 한번 설정 해보시길 바랍니다.



```
// Router 2 (Static <-> OSPF)
router ospf 1
redistribute static subnets
```

```
// router static ???
```

```
// Router 3 (OSPF <-> EIGRP)
router ospf 1
redistribute eigrp 3 subnets
```

```
router eigrp 3
redistribute ospf 1 metric 3000 1000 255 1 1500
```

```
// Router 4 (EIGRP <-> RIP)
router eigrp 3
redistribute rip metric 3000 1000 255 1 1500
```

```
router rip
redistribute eigrp 3 metric 1
```

## 4-4. Redistribution Example

Metric Change

재분배 Metric 값 너무 어려워 하지말고, 서로 같은 정보가 되기 위해, 형식을 맞춰 준다고 생각합니다.

OSPF: no auto-summary / **subnets** / "OSPF랑 같이 정보 나란히 하려면, VSLM 한다는 딱지 달고와!"  
EIGRP: Metric / **Bandwith, Delay, Reliability, Load, MTU** / "EIGRP랑 같이 정보 나란히 하려면, 5가지 Metric 값 달고와!"  
RIP: **Hop Counts(max: 15)** / "RIP랑 같이 정보 나란히 하려면, Hop count Metric 값으로 달고와!"

```
// Router A (Static <-> OSPF:1)
router ospf 1
redistribute static subnets
```

```
// Router A (Static <-> EIGRP:3)
router eigrp 3
redistribute static metric 3000 1000 255 1 1500
```

```
// Router A (Static <-> RIP)
router rip
redistribute static metric 1
```

```
// Router A (EIGRP:3 <-> OSPF:1)
router ospf 1
redistribute eigrp 3 subnets
```

```
// Router A (OSPF:1 <-> EIGRP:3)
router eigrp 3
redistribute ospf 1 metric 3000 1000 255 1 1500
```

```
// Router A (EIGRP:3 <-> RIP)
router rip
redistribute eigrp 3 metric 1
```

```
// Router A (RIP <-> OSPF:1)
router ospf 1
redistribute rip subnets
```

```
// Router A (RIP <-> EIGRP:3)
router eigrp 3
redistribute rip metric 3000 1000 255 1 1500
```

```
// Router A (OSPF:1 <-> RIP)
router rip
redistribute ospf 3 metric 1
```

```
// Router A (OSPF:2 <-> OSPF:1)
router ospf 1
redistribute ospf 2 subnets
```

```
// Router A (EIGRP:2 <-> EIGRP:3)
router eigrp 3
redistribute eigrp 2 metric 3000 1000 255 1 1500
```

```
// Router A (RIP <-> RIP) ???
router rip
redistribute rip metric 1
```



## 4-5. Redistribution (Practice)

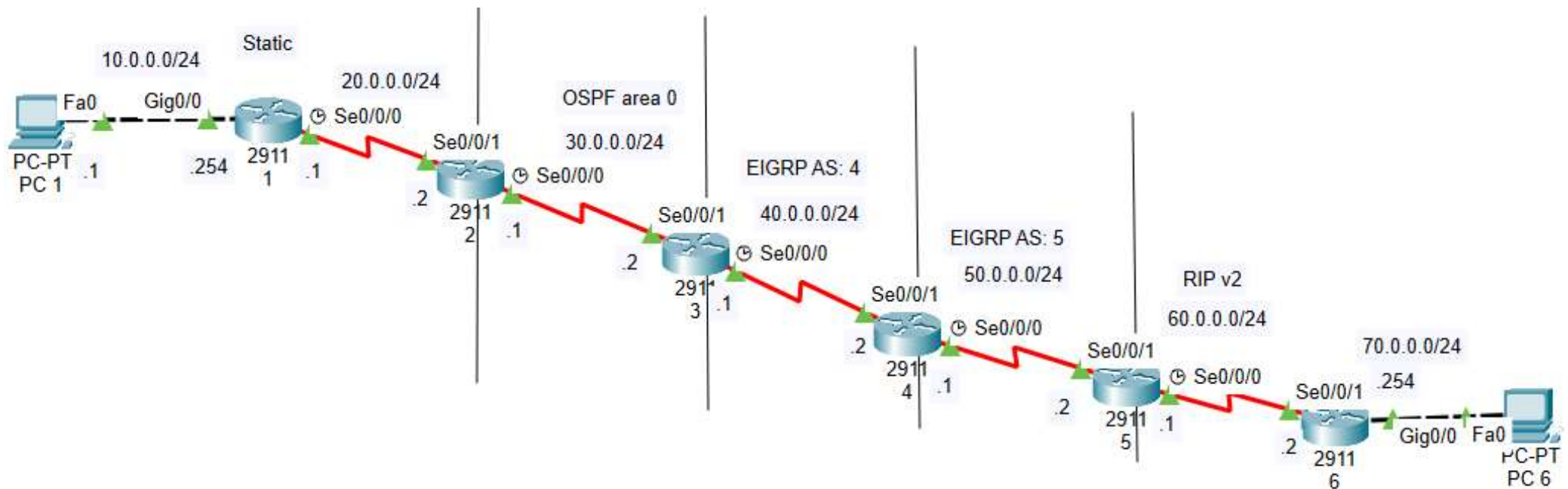
수업시간에 진행했던 과제 입니다. 8차시에 IP 부여된 version도 업로드 되어 있고,  
모든 라우팅이 완료된 version도 업로드 되어 있습니다. 복습한다는 생각으로 한번더 Config 해보시길 바랍니다.

## tips for ip settings:

**se0/0/0: N.0.0.1**

**se0/0/1: N.0.0.2**

**OSPF process-id: 1 / eigrp metric: 3000 1000 255 1 1500 / RIP metric: 2**

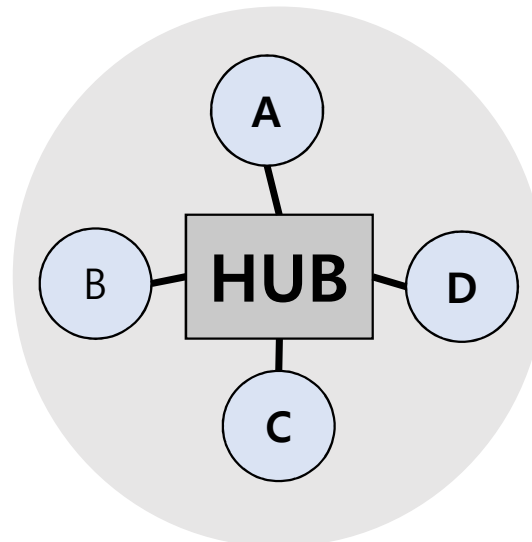


# 5-1. Ethernet Hub (Multiport Repeater)

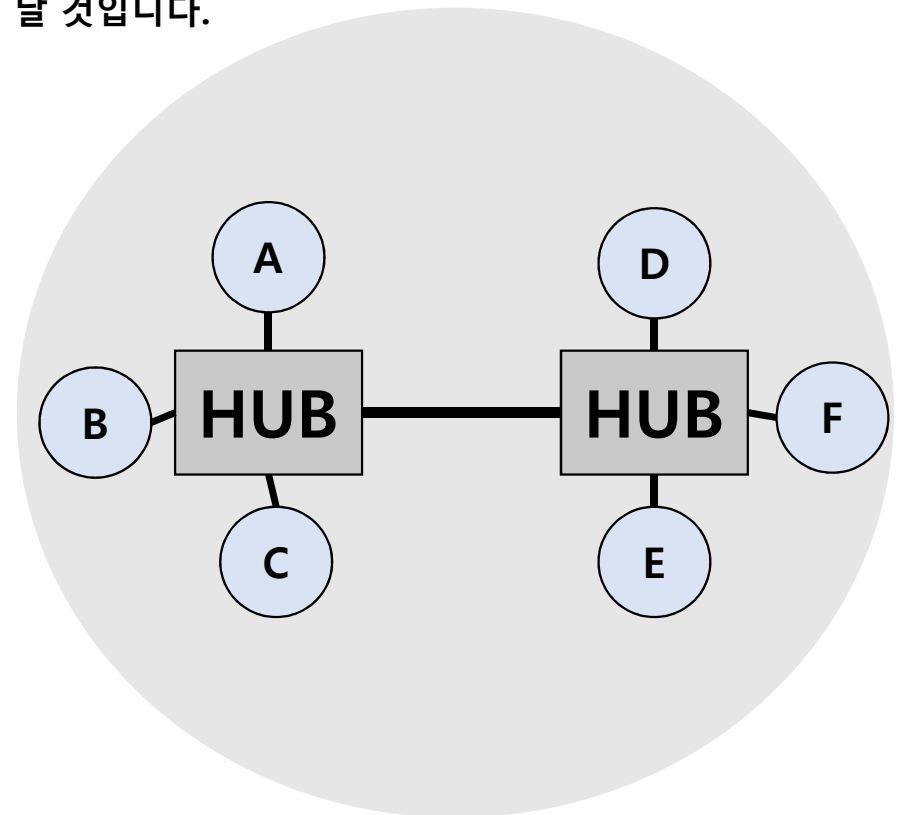
## CSMA/CD

CSMA/CD에서 "눈치껏 알아서 통신하기"가 기억이 난다면,  
한쌍의 장비가 통신 중일 때, 다른 장비들은 동시에 통신 못한다는 것도 기억 날 것입니다.  
이렇게 동시에 통신 하지 못하는 범위를 Collision Domain 이라고 합니다.

허브는 통신이 가능하도록 연결 해주는 장비입니다.  
허브는 허브를 통해서도 연장이 가능합니다.  
그러나 Collision Domain이 그만큼 커지게 됩니다.

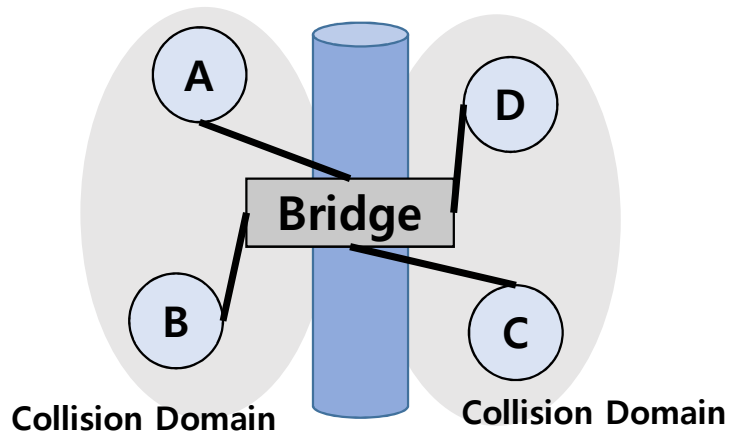


Collision Domain



Collision Domain

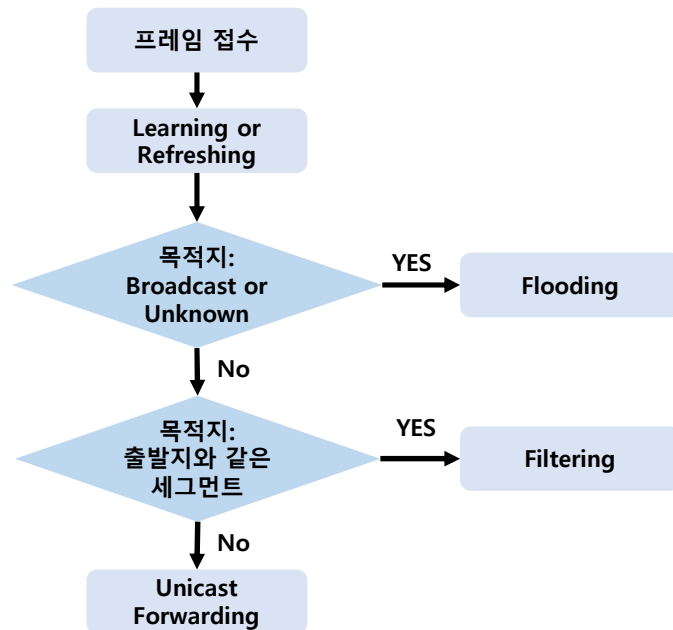
## 5-2. Bridge



쉽게 말해서 Hub의 동시 통신이 안되는 기능을 개선하여, 탄생한 친구가 Bridge 입니다.

Bridge는 동시 통신을 보장해 줍니다.

Bridge는 아래의 단계를 통해 동시 통신을 보장합니다.  
다음 Page에 순서도 또한 표기를 해두었습니다.



1. Learning / 모르면 배운다 (프레임 접수)
2. Flooding / 모르면 뿌린다.

3. Forwarding / 해당 세그먼트로 건네준다.
4. **Filtering** / 목적지 세그먼트가 같을 경우, 다른 포트로 못 건너게 막는다  
(**Collision Domain 분리**)

5. Aging (Table 최적화)

Summary:

Hub: 연결만 가능

Bridge: Collision Domain을 나눈다.

## 6. Summary + Quiz

Summary:

Hub: Multiport Repeater

Bridge: Devide Collision Domain

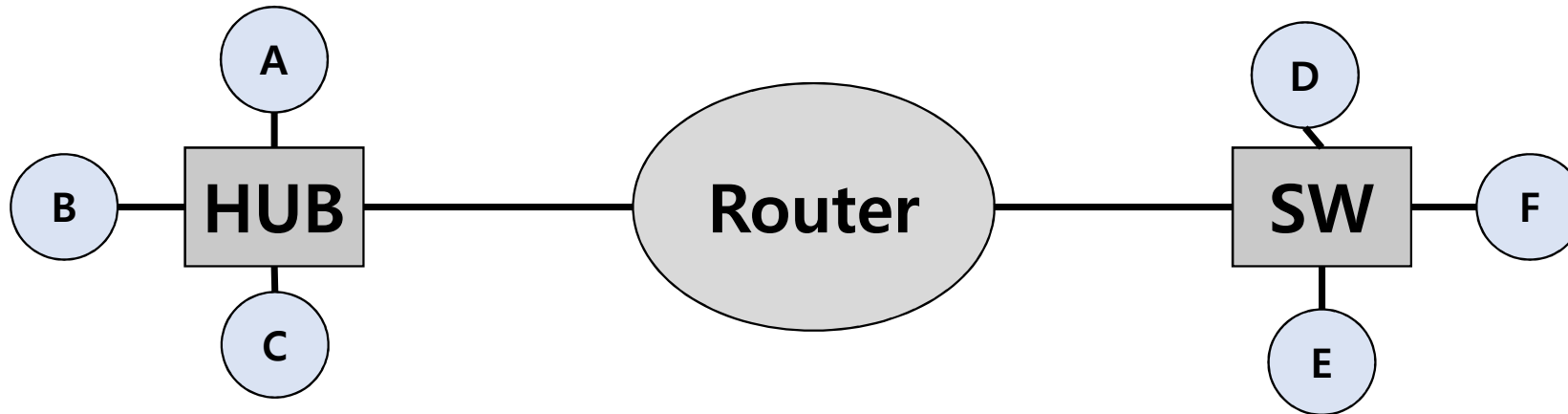
Switch: Bridge + more Functions (포트별로)

Router: Devide Broadcast Domain

동시 통신을 보장하기 위해 HUB에서 Bridge로,  
보다 더 논리적인 기능을 추가하여 Switch로 발전되었습니다.  
추가된 기능은 추후에 더 알아 볼 예정입니다.

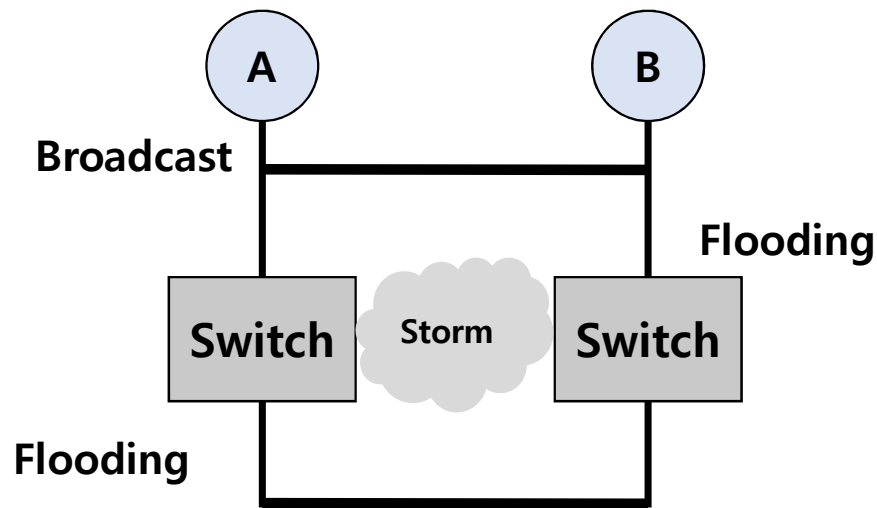
Switch는 Port별로 Collision Domain을 나눠준 답니다.

Quiz : How many Broadcast domain, Collision Domain in this topology?



# 7. Broadcast Storm

5-2 Bridge Flow-chart 에서 살펴본 바와 같이,  
목적지가 Broadcast 일경우, Switch는 Flooding 하게 됩니다.  
그러나 아래의 그림처럼 경로가 2개인 경우, 서로 Flooding을 계속하여 Loop가 발생하게 됩니다.  
이를 해결하기 위한 프로토콜이 Switch 의 Spanning Tree Protocol 줄여서, STP 입니다.



To Solve -> **STP**  
Spanning Tree Protocol

# 8-1. STP (Prepare)

## Broadcast Storm Solve -> STP

STP를 설정하기 위해서는 Bridge ID와 Path Cost를 알아야 합니다.  
Bridge ID는 우선순위와 Mac 주소를 합친 값이며,  
Path Cost는 1000Mbps를 대역폭으로 나눈 값을 정수로 표현한 것입니다.  
(결과가 소수점으로 나오는 값은 컴퓨터에서 처리가 곤란하기에,  
옆의 표에 작성된 바와 같이 적절한 정수와 대응되어 있습니다.)

### 1. Bridge ID (BID)

Priority (2Byte)

Mac Address  
(6Byte)

### 2. Path Cost

1000Mbps /  
Bandwidth

정수 / 소수점 Issue

| Speed    | Port Cost | Comment              |
|----------|-----------|----------------------|
| 10 Mbps  | 100       | Ethernet             |
| 20 Mbps  | 56        | EtherChannel         |
| 30 Mbps  | 47        | EtherChannel         |
| 40 Mbps  | 41        | EtherChannel         |
| 50 Mbps  | 35        | EtherChannel         |
| 54 Mbps  | 33        | 802.11 wireless      |
| 60 Mbps  | 30        | EtherChannel         |
| 70 Mbps  | 26        | EtherChannel         |
| 80 Mbps  | 23        | EtherChannel         |
| 100 Mbps | 19        | Fast Ethernet        |
| 200 Mbps | 12        | Fast EtherChannel    |
| 300 Mbps | 9         | Fast EtherChannel    |
| 400 Mbps | 8         | Fast EtherChannel    |
| 500 Mbps | 7         | Fast EtherChannel    |
| 600 Mbps | 6         | Fast EtherChannel    |
| 700 Mbps | 5         | Fast EtherChannel    |
| 800 Mbps | 5         | Fast EtherChannel    |
| 1 Gbps   | 4         | Gigabit Ethernet     |
| 2 Gbps   | 3         | Gigabit EtherChannel |
| 10 Gbps  | 2         | 10G Ethernet         |
| 20 Gbps  | 1         | 20G EtherChannel     |
| 40 Gbps  | 1         | 40G EtherChannel     |



## 8-2. STP (Rules)

1번 규칙: 네트워크당 하나의 **Root Bridge**를 갖는다.

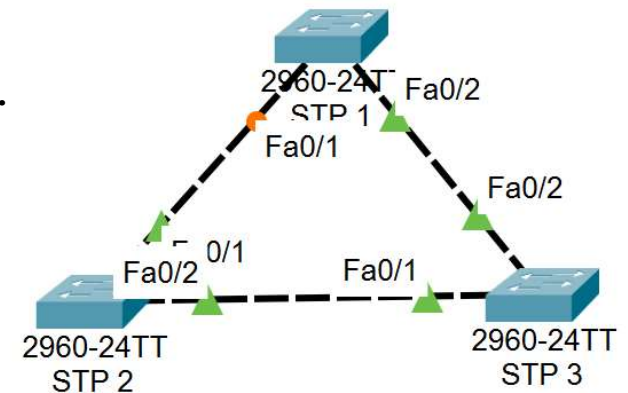
2번 규칙: Non Root Bridge는 하나의 **Root Port**를 갖는다.

3번 규칙: Segment당 하나씩의 **Designated Port**를 갖는다.

\*. Root port, Designated Port 가 아닌 포트 차단

STP를 설정하기에 앞서 STP 세계에는 다음과 같은 규칙이 존재합니다.  
규칙을 통해 Root Bridge, Root Port, Designated Port가 있다는 것을 확인 할 수 있습니다.  
그렇다면, 각각의 역할은 어떻게 선정이 되는지, 한번 살펴 보도록 합시다.

아래 그림을 살펴보면, STP1 Switch의 Fa0/1 포트가 주황 빛을 내고 있습니다.  
이는 STP의 결과라고 할 수 있습니다.  
추후에 나올 내용을 잘 따라간다면, 그 과정을 이해할 수 있을 것 입니다.



# 8-3. STP (Root Bridge)

1번 규칙: 네트워크당 하나의 **Root Bridge**를 갖는다.

뜯어보기

BPDUs (Bridge Protocol Data Units)

Root  
BID

Sender  
BID

Port  
ID

Root Path  
Cost

## Root Bridge 선정 기준: 낮은 BID

Root Bridge의 선정 기준은 낮은 BID입니다.

여기서 잠깐, BID는 Bridge ID로, Priority와 Mac Address를 합친 값입니다.

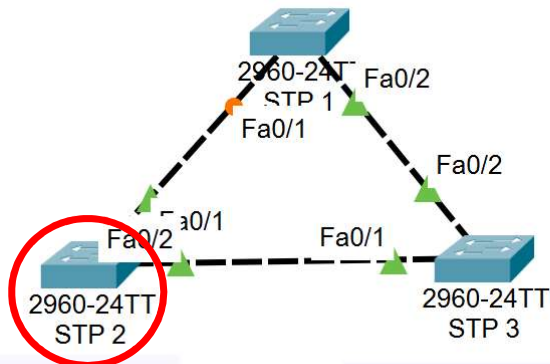
왼쪽 그림에서는 Priority는 같으나, Mac Address로 인해 STP2 Switch가 Root Bridge입니다.

오른쪽 그림에서는 Mac Address가 높아도 낮은 Priority로 인해, STP3 Switch가 Root Bridge입니다.

Switch들끼리 Root Bridge를 선정할때 우측 상단의 BPDUs라는 DATA를 주고 받으며 선정하게 됩니다.

Root Bridge는 Root BID, Sender BID, Port ID, Root Path Cost 등등 다양한 정보를 포함하고 있습니다.

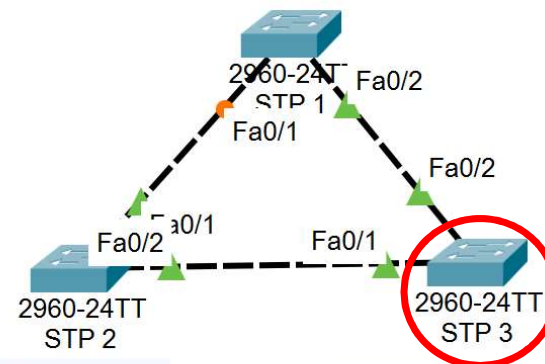
BID: 32768.2222.2222.2222



BID: 32768.1111.1111.1111

BID: 32768.3333.3333.3333

BID: 32768.2222.2222.2222



BID: 32768.1111.1111.1111

BID: 0.3333.3333.3333

## 8-4. STP (Root port)

2번 규칙: Non Root Bridge는 하나의 **Root Port**를 갖는다.

**Root Port 선정 기준: Root Path Cost가 낮은 포트**

그 다음은, Root Port를 선정하는 것 입니다. Root Path Cost란, Root Bridge 까지의 Cost를 의미 합니다.  
100Mbps 회선은 Cost가 19이며, 1Gbps 회선은 Cost가 4입니다.

왼쪽 그림을 먼저 보면, 우선 Root Bridge는 STP3 Switch 입니다. (by. Priority)

그렇다면 STP2, STP1 Switch는 Non Root Bridge로서 2번 규칙에 의해 Root Port를 하나씩 가져야 합니다.

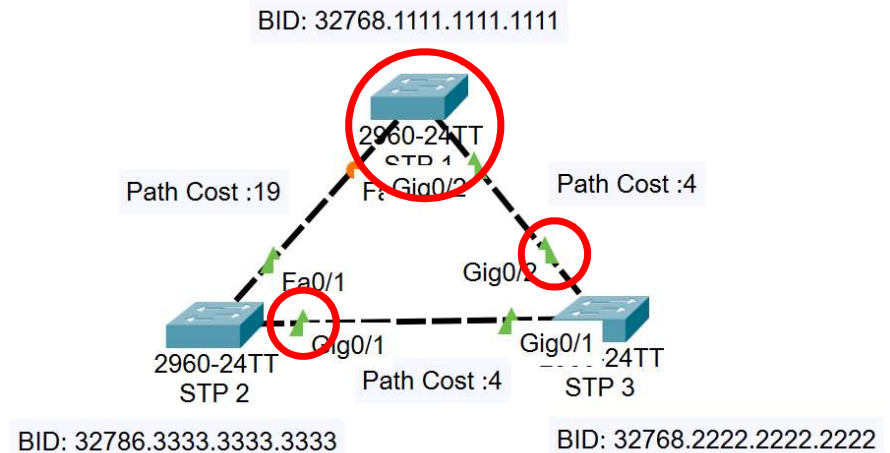
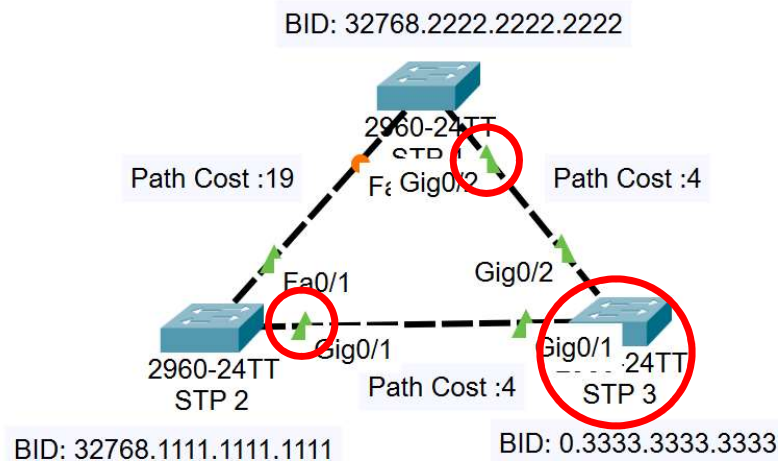
각각 Cost가 적은 길을 고르게 되면, 아래 그림과 같이 Root Port를 선정할 수 있습니다.

오른쪽 그림에서는 STP1 Switch가 Root Bridge 입니다. (by. Mac Address) 그러나 주의 해야할 점이 있습니다.

STP2 Switch에서 STP1로 가는 경로가 fa0/1로 가는 경우, STP3을 거쳐가는 경우 2개가 존재합니다.

Cost값을 계산해보니 STP3을 거쳐가는 것이 회선 개수는 더 많지만 Cost가 더 적게 나옵니다. (19>4+4)

이럴 경우에는 Cost를 기준으로 Root Port를 선정해야 합니다. 결론적으로 아래와 같이 Root Port를 선정 할수 있습니다.



## 8-5. STP (Designated Port)

3번 규칙: Segment당 하나씩의 **Designated Port**를 갖는다.

Designated Port 선정 기준:

1. 누가 더 작은 Root BID를 가졌는가?
2. Root path Cost가 누가 더 작은가?
3. 누구의 BID가 더 낮은가?
4. 누구의 포트 ID가 더 낮은가?

마지막 단계입니다. 현재 빨간 색 동그라미로 선정할 수 있는 Root Bridge와 Root port를 선정해 두었습니다.

3번째 규칙에 의해 각 Segment당 Designated Port만 선정해 주면 됩니다.

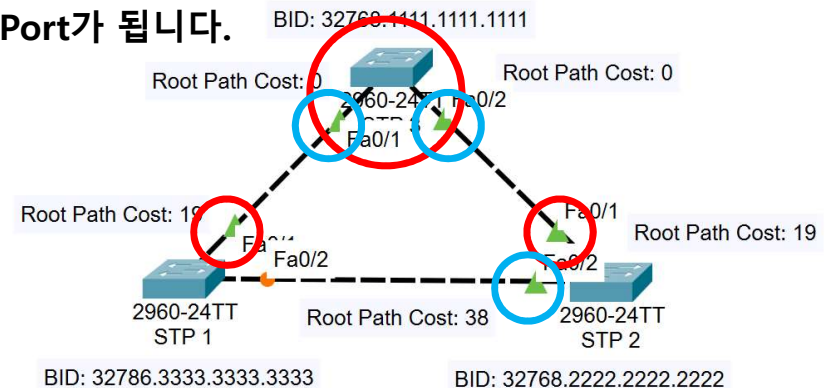
여기서 Segment란, 의도와 상관없이 회선에 의해 신호를 받는 범위, 쉽게 말해 회선 하나를 뜻합니다.

2번 선정기준을 통해 Root path Cost가 0인 Root Bridge의 포트들을 Designated Port로 선정합니다.

3번 선정기준을 통해 Root path Cost가 같은 STP1, STP2 Switch 간의 Designated Port를 선정합니다.

STP2의 BID가 더 낮기에 STP2의 FA0/2 Port를 Designated Port로 선정합니다.

그렇게 어떠한 Port로도 선정이 안된 STP1 Switch의 Fa0/2는 Non Designated Port가 됩니다.



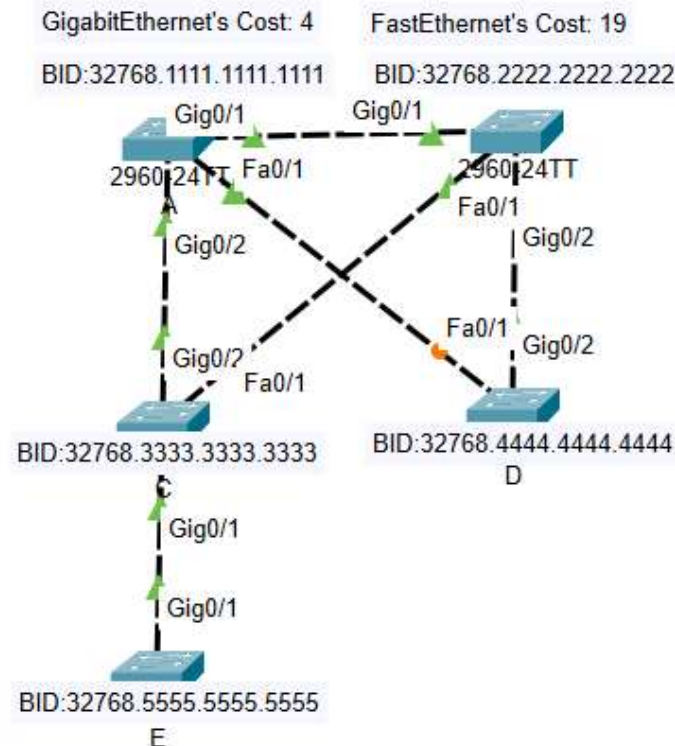
## 8-6. STP (Summary + Quiz)

- 1번 규칙: 네트워크당 하나의 **Root Bridge**를 갖는다.
- 2번 규칙: Non Root Bridge는 하나의 **Root Port**를 갖는다.
- 3번 규칙: Segment당 하나씩의 **Designated Port**를 갖는다.
- \*. Root port, Designated Port 가 아닌 포트 차단

1. **Root Bridge** 선정 기준: 낮은 BID
2. **Root Port** 선정 기준: Root Path Cost가 낮은 포트
3. **Designated Port** 선정 기준:
  1. 누가 더 작은 Root BID를 가졌는가?
  2. Root path Cost가 누가 더 작은가?
  3. 누구의 BID가 더 낮은가?
  4. 누구의 포트 ID가 더 낮은가?

cf)  
Why STP? for prevent Broadcast Storm  
BID? Priority + Mac Address  
Path Cost?  $1000\text{Mbps} / \text{Bandwidth} \rightarrow$  정수 대응 값  
BPDU? STP정보를 위한 Unit

지금까지 잘 따라 오셨다면, STP를 완료할 수 있습니다.  
아래 그림에서 Root Bridge, Root Port, Designated Port,  
Non designated port를 선정해 봅니다.  
다음 Page에 정답이 있습니다.



## 8-7. STP (Status)

그렇게 설정된 Non Designated Port는 앞서 차단 한다고 이야기 했었는데,  
정확히는 Blocking 상태가 됩니다.  
Status는 아래의 설명 참고 부탁드립니다.

(데이터 전송 여부 / MAC Address 학습 여부 / BPDU 송수신 여부)

**1. Disabled** (x/x/x): 고장 or 의도적 shutdown

(Non Designated: Blocking 상태)

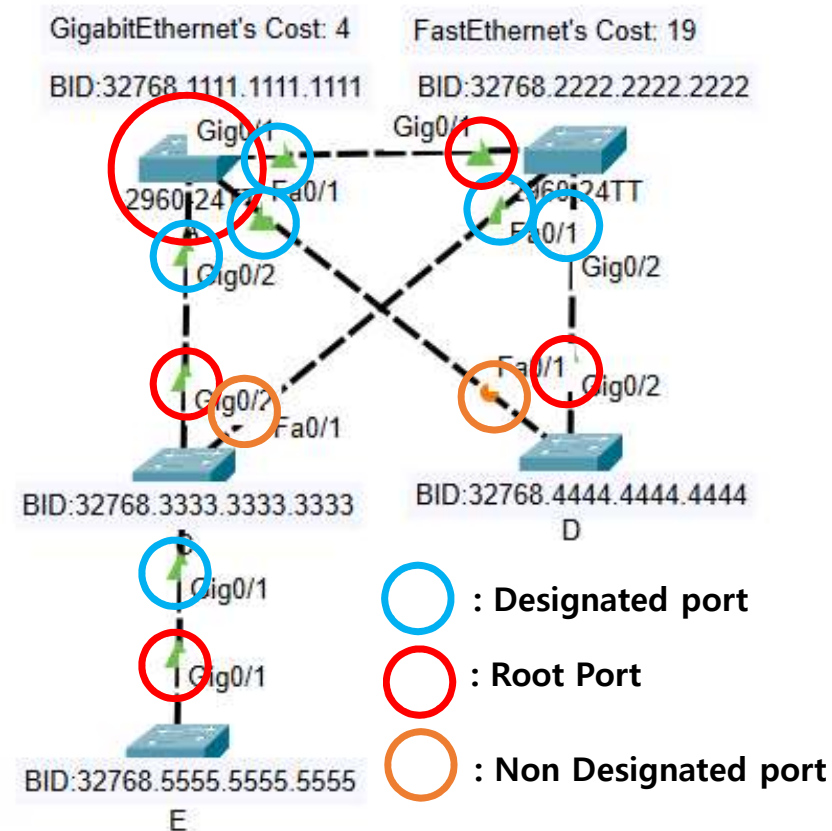
**2. Blocking** (x/x/o): 전원 On or Disabled 해제

**3. Listening** (x/x/o): Root, Designated Port 설정 완료 후 진입

**4. Learning** (x/o/o): Listening 상태 15초 이상 유지 후 진입

**5. Forwarding** (o/o/o): Learning 상태 15초 이상 유지 후 진입

여기까지, 여태까지 제가 수업했던 내용들 입니다.  
복습 꼭 하시고, 다음 시간에 뵙도록 하겠습니다.  
감사합니다 :)





**-복습 End-**

# 9-1. VTP(Vlan Trunking Protocol)

**Vlan** :Switch에서만 지원 (Bridge, Hub 미지원)  
:논리적으로 Broadcast Domain을 나눈다.

```
Switch# show vlan // vlan 정보 조회  
Switch(config)# vlan [N] // 삭제시, no vlan [N]  
Switch(config-vlan)# name [Name]
```

**VTP(Vlan Trunking Protocol)**: Switch간 vlan 정보를 항상 일치 시켜 주는 프로토콜  
(Why? 모든 Switch에 Vlan 생성하는 것은 귀찮기 때문)

VTP 정보: **Trunk port** 로만 전달 가능

VTP: Domain, Password 값이 같아야 VTP 진행 가능

-다른 네트워크는 라우터를 통해야만 연결이 가능하다 (라우팅 필요)

## 9-2. VTP(VTP mode)

VTP mode: **Server / Transparent / Client**

Server: 생성, 변경, 전달 가능

Transparent: 본인 저장 X, 전달만 가능

Client: 연결된 스위치에 전달, 수신 가능

Switch(config)#vtp mode server / transparent / client (3중1택)

Switch(config)#vtp domain [Domain]

Switch(config)#vtp password [Password]

#Deafult Value:

VTP Domain Name = null

VTP version = 1 (v2, Token ring vlan 지원)

VTP Mode = Server

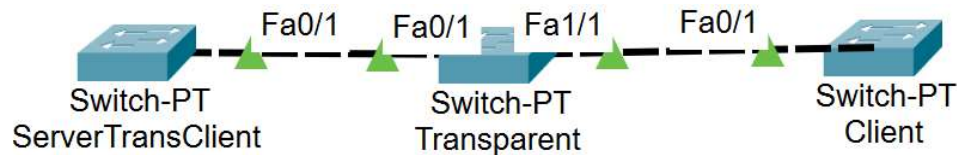
## 9-2. VTP(VTP mode)

VTP mode: **Server / Transparent / Client**

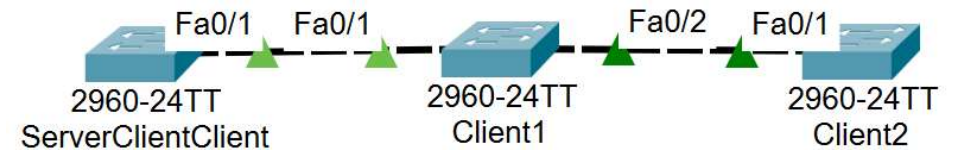
Server: 생성, 변경, 전달 가능

Transparent: 본인 저장 X, 전달만 가능

Client: 연결된 스위치에 전달, 수신 가능



| Server#show vlan | Transparent#show vlan   | Client#show vlan |
|------------------|-------------------------|------------------|
| VLAN Name        | VLAN Name               | VLAN Name        |
| 1 default        | 1 default               | 1 default        |
| 10 Red           | 1002 fddi-default       | 10 Red           |
| 20 Green         | 1003 token-ring-default | 20 Green         |
| 30 Blue          | 1004 fddinet-default    | 30 Blue          |
|                  | 1005 trnet-default      |                  |



| Server#show vlan | Client1#show vlan | Client2#show vlan |
|------------------|-------------------|-------------------|
| VLAN Name        | VLAN Name         | VLAN Name         |
| 1 default        | 1 default         | 1 default         |
| 40 Hour          | 40 Hour           | 40 Hour           |
| 50 Minute        | 50 Minute         | 50 Minute         |
| 60 Second        | 60 Second         | 60 Second         |

## 9-3. VTP(Switchport mode)

// 특정 Vlan 적용 시, (1개)

```
Switch(config)# int [interface]
```

```
Switch(config-if)# switchport mode access
```

```
Switch(config-if)# switchport access vlan [N]
```

// Trunk port 적용 시, (N개)

```
Switch(config)# int [interface]
```

```
Switch(config-if)# switchport mode trunk
```

Trunk + Nonegotiate: 안정적  
Dynamic + Nonegotiate: 오류

DTP(Dynamic Trunking Protocol)

1. **access**: Never trunk, Never DTP send

: Switch(config-if)# switchport mode access

2. **trunk**: DTP send, 상대방 DTP ignore

: Switch(config-if)# switchport mode trunk

3. dynamic (auto): DTP send X, DTP reply only // **default**

: Switch(config-if)# switchport mode dynamic auto

4. dynamic (desirable): DTP send, DTP reply

: Switch(config-if)# switchport mode dynamic desirable

5. Nonegotiate: disable DTP

: Switch(config-if)# switchport nonegotiate

## cf) DTP 더 알아보기

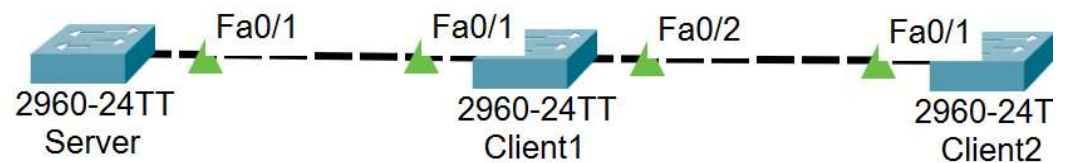
요약: **auto-auto**, **[Any]-Access** 제외하고 Trunk port로 사용 가능

| Administrative Mode    | Auto               | Desirable          | Trunk (on)         | Access (off)       | Non-Negotiate (access) | Non-Negotiate (trunk) |
|------------------------|--------------------|--------------------|--------------------|--------------------|------------------------|-----------------------|
| Auto                   | Static access (NT) | Trunk              | Trunk              | Static access      | Static access          | Unexpected Results    |
| Desirable              | Trunk              | Trunk              | Trunk              | Static access      | Static access          | Unexpected Results    |
| Trunk (on)             | Trunk              | Trunk              | Trunk              | Unexpected Results | Unexpected Results     | Trunk                 |
| Access (off)           | Static access      | Static access      | Unexpected Results | Static access      | Static access          | Unexpected Results    |
| Non-Negotiate (access) | Static access      | Static access      | Unexpected Results | Static access      | Static access          | Unexpected Results    |
| Non-Negotiate (trunk)  | Unexpected Results | Unexpected Results | Trunk              | Unexpected Results | Unexpected Results     | Trunk                 |

## 9-4. VTP(Lab)

Situation: VTP 덕분에 Server, Client, Client 상황에서 Vlan 6개의 정보를 서로 일치시키고 있다.

1. VTP Mode, Domain, Password 설정 ( $3 \times 3 = 9$ )
2. VTP Server에 6개의 Vlan 정보 생성 (6)
3. 각 Segment 별로 Trunk port 지정 (2) // 17



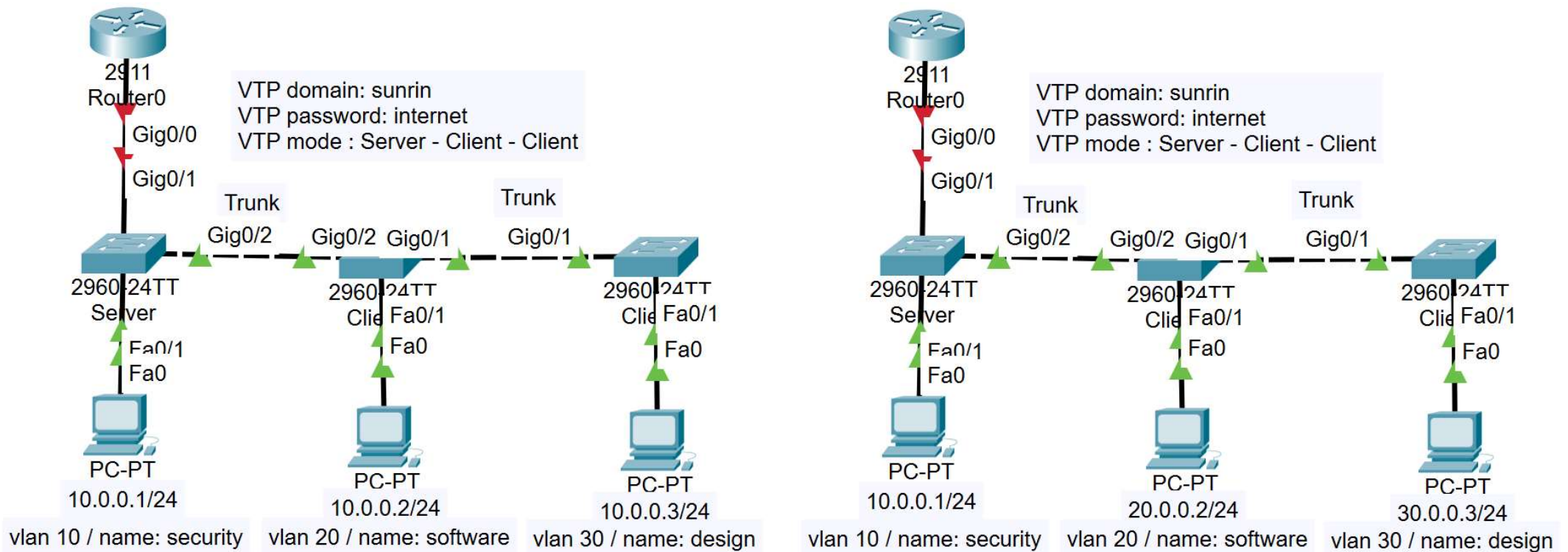
If same Situation에서, VTP가 없다면?

1. Switch 별 Vlan 정보 생성 ( $6 \times 3 = 18$ )
2. 각 포트별 Vlan 적용 ( $2 \times 6 \times 2 = 24$ ) // 42
- \*. Vlan 정보 수정시 반영 X



## 9-5. VTP(Practice)

Port 번호를 아래 Topology와 맞추어 설정합니다. / 왼쪽 오른쪽 토폴로지를 비교해 봅니다.  
VTP 설정 후, PC끼리 Ping을 보내봅니다.





## 9-6. Inter-Vlan(Practice)

Inter-Vlan을 적용하여 모든 PC 끼리 통신이 가능하도록 설정합니다.  
Gateway는 해당 네트워크의 사용가능한 마지막 IP를 사용합니다.  
Routing protocol은 RIPv2를 사용합니다.

1. Router와 만나는 Switch: Trunk port 지정

2. Router에 Sub-Interface 생성

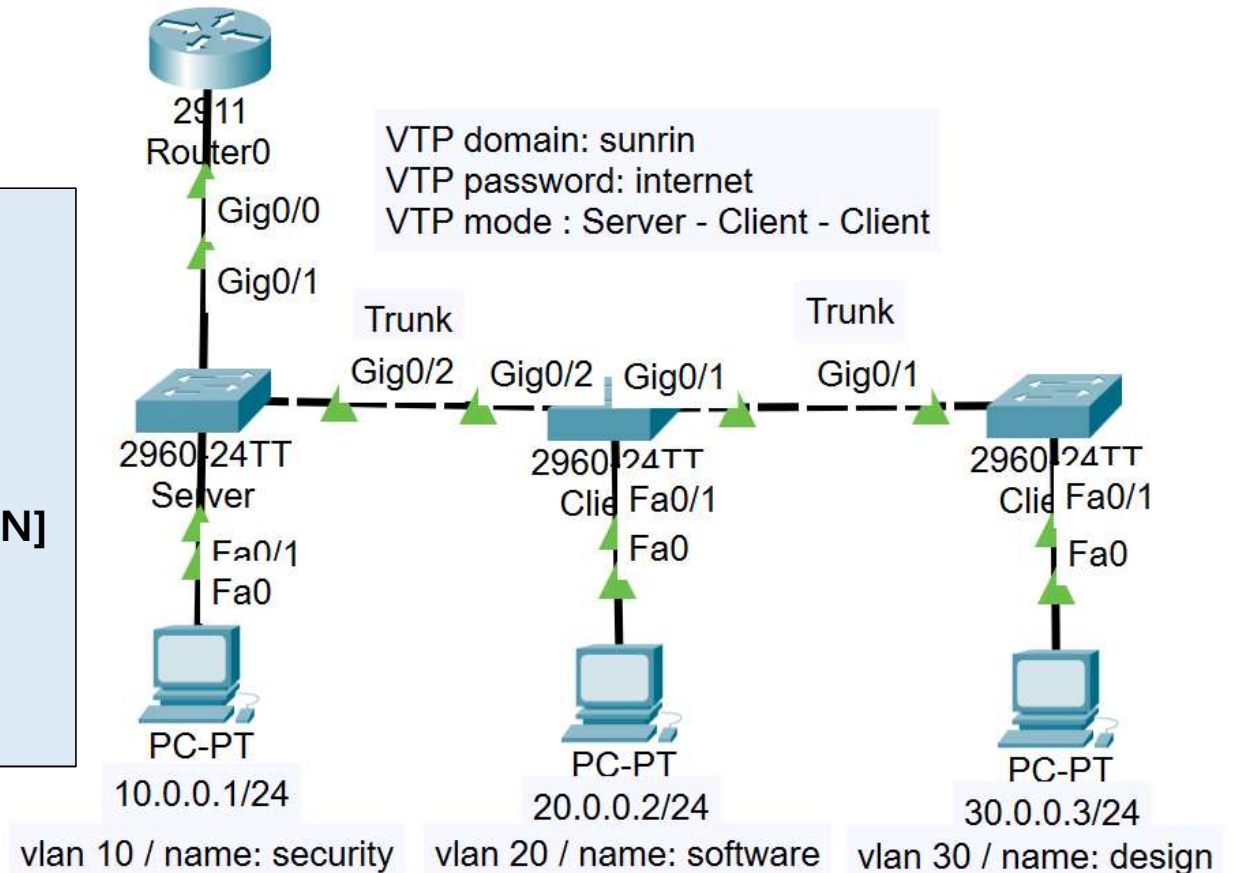
Router(config-if)#int gi0/0.[N]

3. Sub-Interface에 vlan 정보 입력

Router(config-subif)#encapsulation dot1Q [N]

4. Sub-Interface에 IP 정보 입력

5. Routing table Update



## 9-7. Summary + Quiz

Summary: [Inter-vlan + VTP]

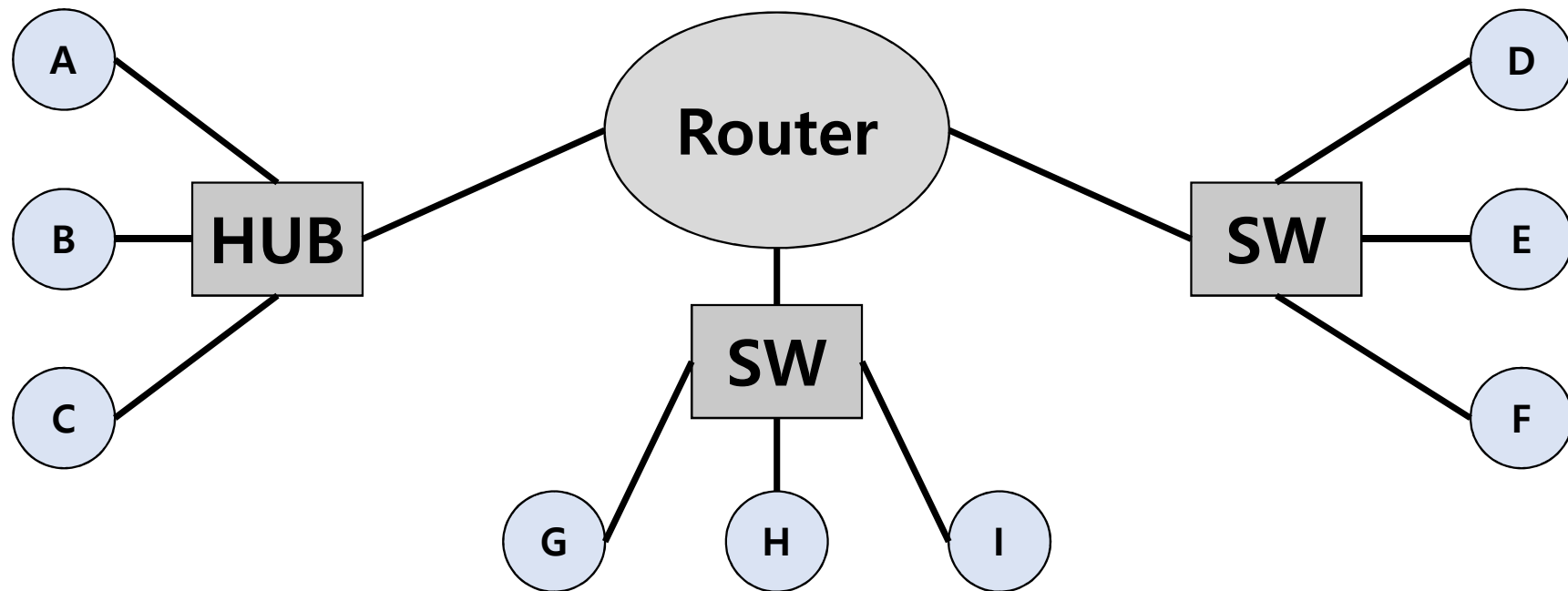
VTP mode Server / Transparent / Client

Domain , Password

Trunk / Access / Dynamic (auto, desirable)

sub-interface (encapsulation dot1q) -> Routing

Quiz : How many **Broadcast domain**, **Collision Domain** in this topology?



# 10-1. WAN(Prepare)

Summary: LAN처럼 WAN에도 2계층 Encapsulation 방식이 있고, 그 중에는 HDLC, PPP, Frame-Relay와 같은 방식 들이 있다.

WAN <-> LAN

WAN

: 멀리 떨어진 LAN과 LAN 사이 연결  
: 직접 케이블을 깔 수 없을 때  
3계층 <-> 2계층: 속도 차이

1. Leased Line: 전용선 (비용, 안정)

└ PPP, HDLC

2. Circuit Switching (예약 시간 비례/회선 회수)

└ PPP, HDLC

3. Packet Switching (패킷 비례/ 회선 공유)

└ Frame-Relay (Router 필요)

\*. VPN <-> 전용선

Packet -> Router -> 2계층 Header를 벗김 -> 3계층 주소 확인 -> 2계층 새로운 정보 생성 -> 외부

WAN's 2계층 Protocol (Not 3계층) // encapsulation 방식

1. ATM (음성, 영상/ 53 byte Cell)

2. **HDLC** (High-level DataLink Control / Cisco 전용: **Default** / 표준 HDLC와 다름)

2. **PPP** (타사 장비 사용 가능/ **PAP, CHAP** / more better than HDLC / 기능)

3. **Frame-Relay** (오류 검출 drop -> 속도 get)

이론

## 10-2. WAN(PPP)

Summary: PPP는 NCP, LCP를 통해 다양한 기능을 제공한다.  
1. PAP, 2. CHAP 인증 기능 지원

PPP (Point to Point)

1. **NCP** (Network Control Protocol)

서로 다른 3계층 protocol encapsulation 지원

= Multi Protocol 지원

= 타사 장비 끼리 사용 가능하다

2. **LCP** (Link Control Protocol)

보안, 인증, 압축, 에러 검출, 다중 링크

└ 2-1. PPP PAP

└ 2-2. PPP CHAP

PPP 세션 구축 단계

1단계: 데이터 링크 계층의 세션 구축



2단계: 보안 인증



3단계: 네트워크 계층의 세션 구축

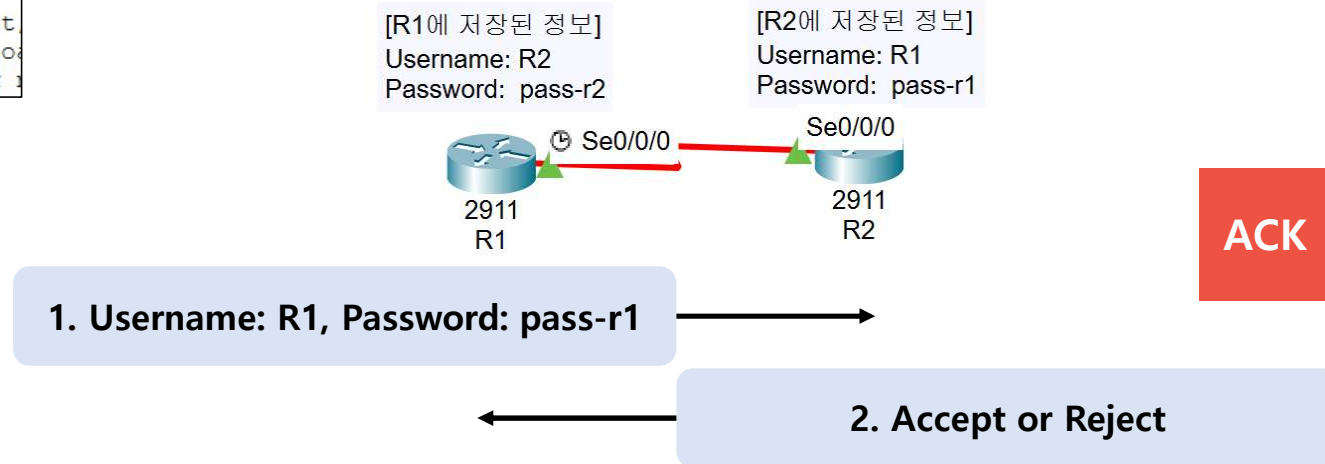
## 10-3. WAN(PPP PAP)

### PAP (Password Authentication Protocol)

2-way hand shake 방식으로, 최초 연결 시에만 한번 인증 검사,  
username, password 평문으로 전달

```
Router#show interfaces se0/0/0
Serial0/0/0 is administratively
(disabled)
  Hardware is HD64570
  MTU 1500 bytes, BW 1544 Kbit,
    reliability 255/255, txload
  Encapsulation HDLC, loopback
```

```
Router#show interfaces se0/0/0
Serial0/0/0 is administratively
(disabled)
  Hardware is HD64570
  MTU 1500 bytes, BW 1544 Kbit,
    reliability 255/255, txload
  Encapsulation PPP, loopback
```



```
R1(config)# username R2 password pass-r2
```

```
R1(config)#int se0/0/0
```

```
R1(config-if)#encapsulation ppp
```

```
R1(config-if)#ppp authentication pap
```

```
R1(config-if)#ppp pap sent-username R1 password pass-r1
```

```
R2(config)# username R1 password pass-r1
```

```
R2(config)#int se0/0/0
```

```
R2(config-if)#encapsulation ppp
```

```
R2(config-if)#ppp authentication pap
```

```
R2(config-if)#ppp pap sent-username R2 password pass-r2
```

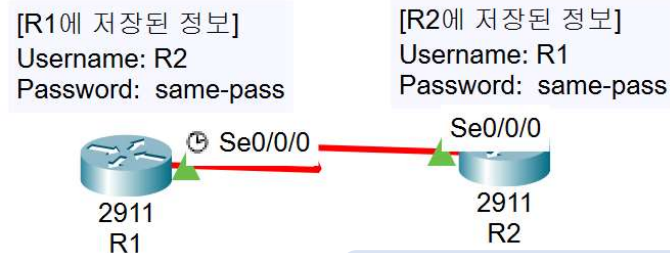
# 10-4. WAN(PPP CHAP)

**CHAP** (Challenge Handshake Authentication Protocol)

3-way handshake 방식으로 주기적 인증 검사

username, password MD5 hash값으로 전송

hostname으로 peer 식별, 동일한 암호 사용



1. Challenge  
(with Label.R2)

2. Response  
(Label.R2.same-pass.Hash with Label.R1)

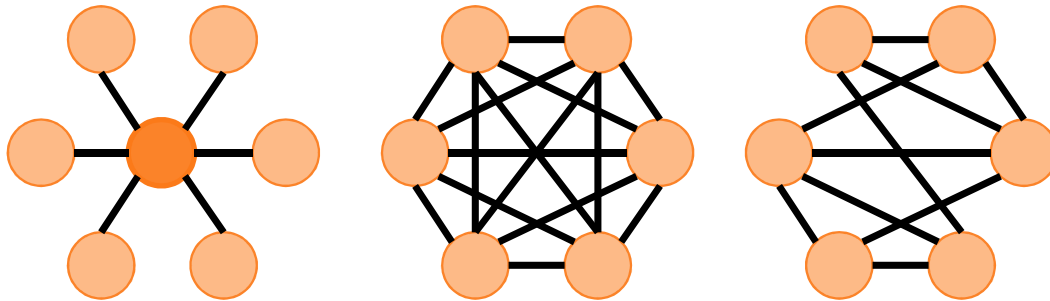
3. Accept or Reject  
(Hash from R2 v.s. R1.same-pass.Hash)

```
Router(config)# hostname R1
R1(config)#username R2 password same-pass
R1(config)#int se0/0/0
R1(config-if)#encapsulation ppp
R1(config-if)#ppp authentication chap
```

```
Router(config)# hostname R2
R2(config)#username R1 password same-pass
R2(config)#int se0/0/0
R2(config-if)#encapsulation ppp
R2(config-if)#ppp authentication chap
```

# 10-5. Frame-Relay(Prepare)

**Frame-Relay**: WAN Encapsulation (가상 전용선 제공)  
Routing X so, **Router Needed** for Routing(Packet Switching)  
종류: Hub-And-Spoke, Full-mesh, Partial-mesh



X.25 Packet Switching의 overhead(오류 책임 X->다른 계층에 전담, Drop) 제거 -> Frame-Relay탄생  
/\*

cf) Frame-Relay 관련 용어 정리

**PVC** (Permanent Virtual Circuit): 고정 가상 회선

**DLCI** (Data Link Connection Identifier): PVC 구별을 위한 식별 주소

**LMI** (Local Management Interface): PVC 상태 정보를 위한 Protocol

**Inverse ARP** (Address Resolution Protocol): DLCI, Router IP 자동 연동 Protocol // **Dynamic Mapping**

**FECB** (Forward Explicit Congestion Notification): 트래픽 혼잡시 데이터 수신 Router에 통지

**BECD** (Backward Explicit Congestion Notification): 트래픽 혼잡시 데이터 송신 Router에 통지

**DE** (Discard Eligibility): 트래픽 혼잡시 가장 우선적으로 버려지는 Frame 설정 비트

**CIR** (Committed Information Rate): Frame-Relay에서 허용되는 전송 속도 \*/

## 10-6. Frame-Relay(Split-horizon)

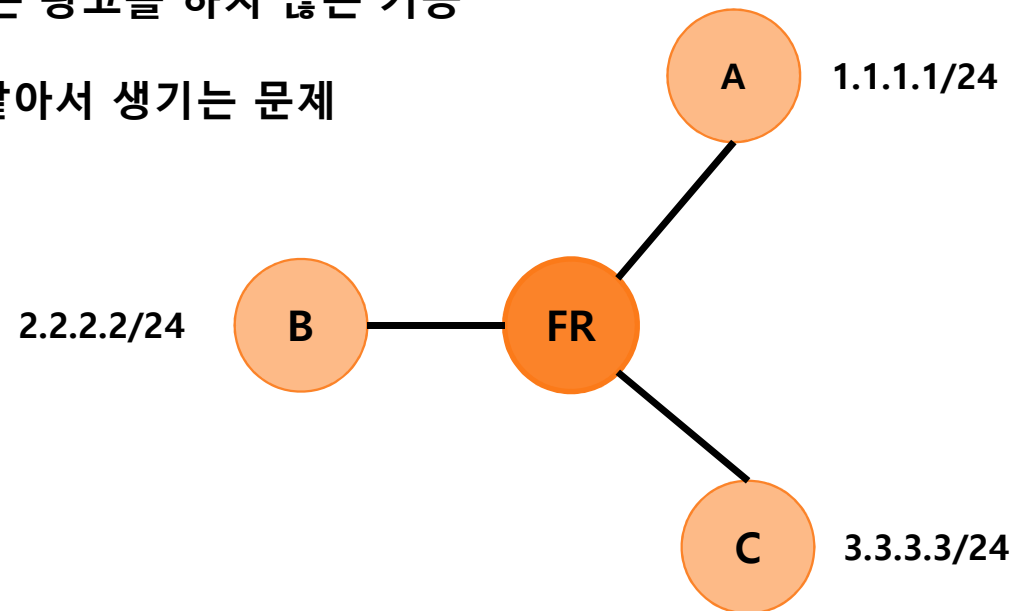
**Split-horizon:** Distance Vector 알고리즘을 사용하는 라우터에서  
특정 정보가 특정 포트에서 들어왔다면, 해당 포트로는 광고를 하지 않는 기능

Frame-Relay의 정보가 들어온 port 나가는 port가 같아서 생기는 문제

Solution 1. Sub-Interface를 선언하여, port 분리

Solution 2. Split horizon 기능 비활성화

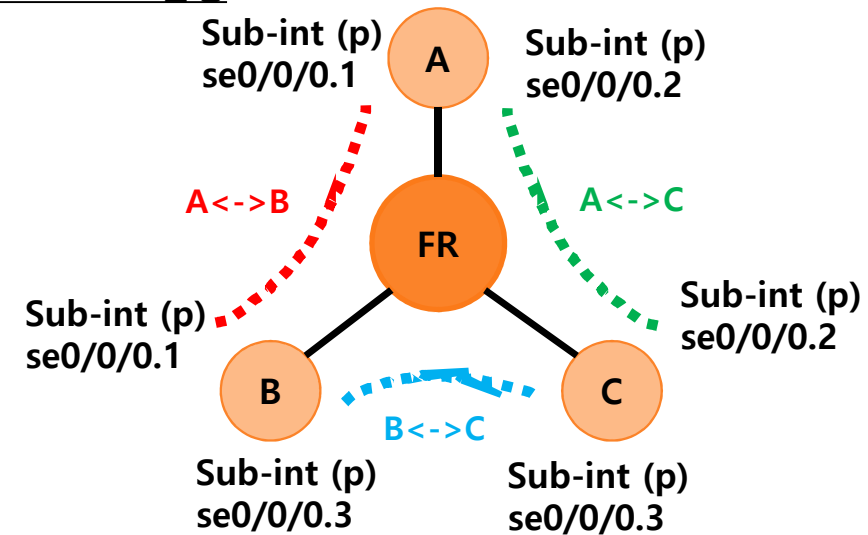
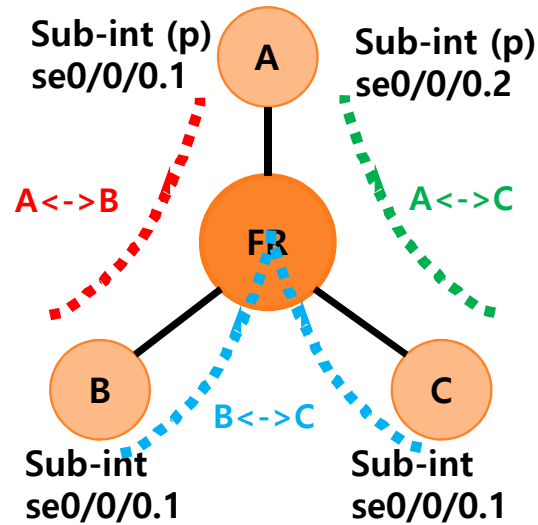
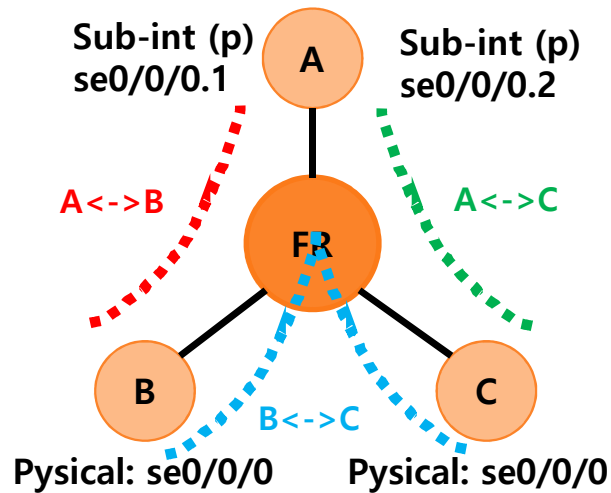
Solution 3. Link State 방식 라우팅 사용



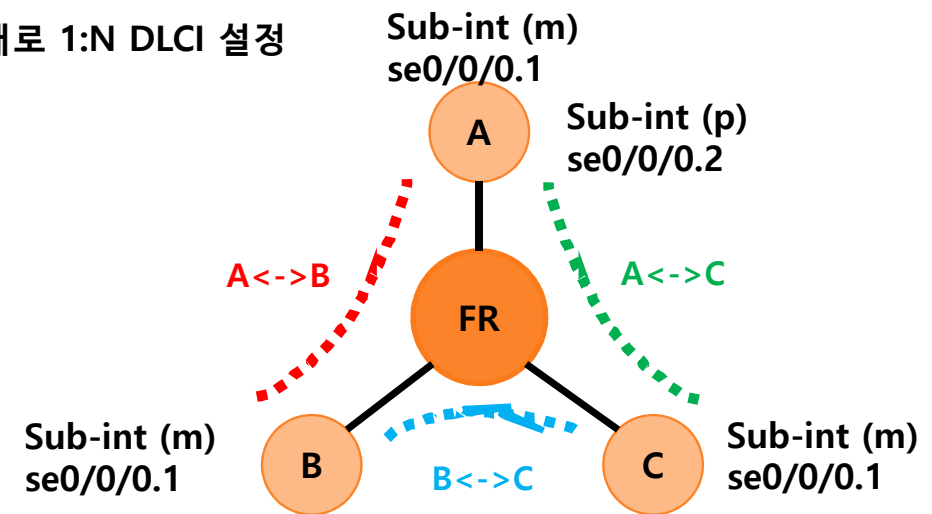
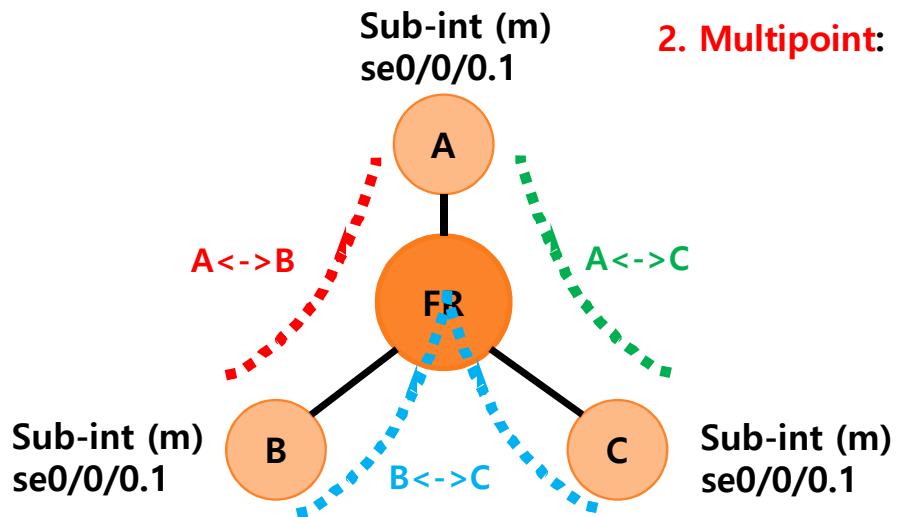


## 10-7. Frame-Relay(2 Types)

### 1. Point-to-Point: Port 1개와 port 1개를 1:1로 DLCI 설정



### 2. Multipoint: Port 1개를 Port 여러개로 1:N DLCI 설정



## 10-8-1. Frame-Relay(Point to Point Settings)

**1. Point-to-Point:** Port 1개와 port 1개를 1:1로 DLCI 설정 (via)

```
int se0/0/0
encapsulation frame-relay
no frame-relay inverse-arp // DLCI 자동 적용 방지

int se0/0/0.1 point-to-point // port 번호 상관 X
ip add [IP] [SM] // ip 부여 (Port 마다 다른 Subnet)
frame-relay interface-dlci [DLCI] // Sub-interface 에서만 사용 가능

[Routing]
```

**2. Multipoint:** Port 1개와 port N개를 1:N로 DLCI 설정 (via)

```
int se0/0/0
encapsulation frame-relay
no frame-relay inverse-arp // DLCI 자동 적용 방지

int se0/0/0.1 multipoint
ip add [IP] [SM] // ip 부여 (동일한 Subnet)
frame-relay map ip [목적지 frame-relay ip] [DLCI] broadcast // broadcast+multicast 허용 / Physical, Multiport 일때

[Routing]
```

## 10-8-2. Frame-Relay(Point to Point via)

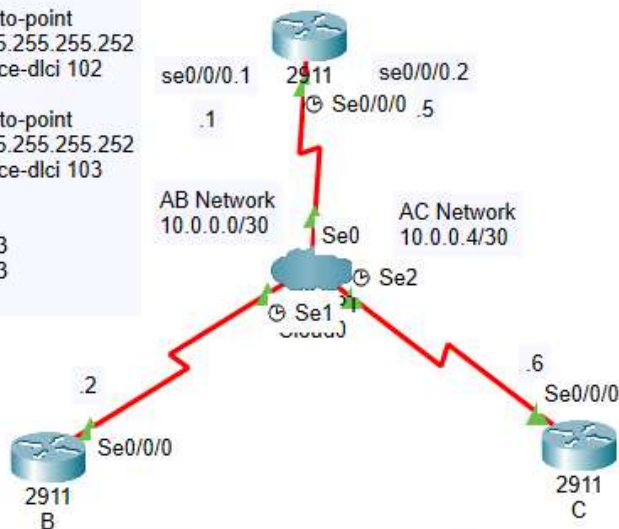
### 1. Point-to-Point: Port 1개와 port 1개를 1:1로 DLCI 설정 (via)

```
int se0/0/0
encapsulation frame-relay
no frame-relay inverse-arp
```

```
int se0/0/0.1 point-to-point
ip add 10.0.0.1 255.255.255.252
frame-relay interface-dlci 102
```

```
int se0/0/0.2 point-to-point
ip add 10.0.0.5 255.255.255.252
frame-relay interface-dlci 103
```

```
router eigrp 1
net 10.0.0.0 0.0.0.3
net 10.0.0.4 0.0.0.3
no au
```



```
int se0/0/0
ip add 10.0.0.2 255.255.255.252
encapsulation frame-relay
```

```
router eigrp 1
net 10.0.0.0 0.0.0.3
no au
```

```
int se0/0/0
ip add 10.0.0.6 255.255.255.252
encapsulation frame-relay
```

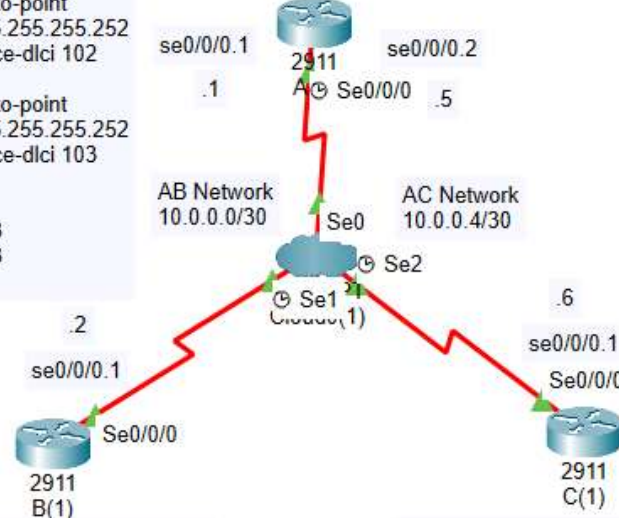
```
router eigrp 1
net 10.0.0.4 0.0.0.3
no au
```

```
int se0/0/0
encapsulation frame-relay
no frame-relay inverse-arp
```

```
int se0/0/0.1 point-to-point
ip add 10.0.0.1 255.255.255.252
frame-relay interface-dlci 102
```

```
int se0/0/0.2 point-to-point
ip add 10.0.0.5 255.255.255.252
frame-relay interface-dlci 103
```

```
router eigrp 1
net 10.0.0.0 0.0.0.3
net 10.0.0.4 0.0.0.3
no au
```



```
int se0/0/0
encapsulation frame-relay
no frame-relay inverse-arp
```

```
int se0/0/0.1 point-to-point
ip add 10.0.0.2 255.255.255.252
frame-relay interface-dlci 201
```

```
router eigrp 1
net 10.0.0.0 0.0.0.3
no au
```

| Frame Relay |         |         |         |
|-------------|---------|---------|---------|
| Serial0     |         | Serial0 |         |
| Port        | Sublink | Port    | Sublink |
| 1           | Serial0 | Serial1 | 201     |
| 2           | Serial0 | Serial2 | 301     |

| INTERFACE |  | DLCI | Name |
|-----------|--|------|------|
| Serial0   |  | 102  | 102  |
| Serial0   |  | 103  | 103  |
| INTERFACE |  | DLCI | Name |
| Serial0   |  | 201  | 201  |
| INTERFACE |  | DLCI | Name |
| Serial0   |  | 301  | 301  |
| Serial1   |  |      |      |
| Serial2   |  |      |      |

# 10-8-3. Frame-Relay(Point to Point mesh)

## 1. Point-to-Point: Port 1개와 port 1개를 1:1로 DLCI 설정 (mesh)

```

int se0/0/0
encapsulation frame-relay
no frame-relay inverse-arp

int se0/0/0.1 point-to-point
ip add 10.0.0.1 255.255.255.252
frame-relay interface-dlci 102

int se0/0/0.2 point-to-point
ip add 10.0.0.5 255.255.255.252
frame-relay interface-dlci 103

router eigrp 1
net 10.0.0.0 0.0.0.3
net 10.0.0.4 0.0.0.3
no au
    
```

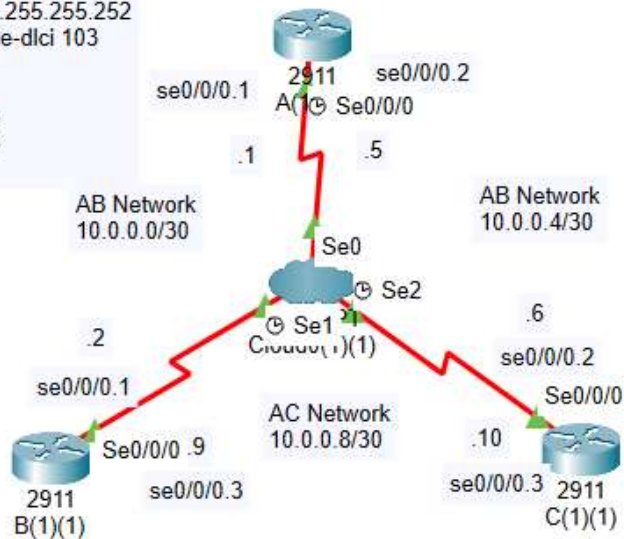
```

int se0/0/0
encapsulation frame-relay
no frame-relay inverse-arp

int se0/0/0.1 point-to-point
ip add 10.0.0.2 255.255.255.252
frame-relay interface-dlci 201

int se0/0/0.3 point-to-point
ip add 10.0.0.9 255.255.255.252
frame-relay interface-dlci 203

router eigrp 1
net 10.0.0.0 0.0.0.3
net 10.0.0.8 0.0.0.3
no au
    
```



```

int se0/0/0
encapsulation frame-relay
no frame-relay inverse-arp

int se0/0/0.1 point-to-point
ip add 10.0.0.6 255.255.255.252
frame-relay interface-dlci 301

int se0/0/0.3 point-to-point
ip add 10.0.0.10 255.255.255.252
frame-relay interface-dlci 302

router eigrp 1
net 10.0.0.4 0.0.0.3
net 10.0.0.8 0.0.0.3
no au
    
```

| Frame Relay |         |      |         |
|-------------|---------|------|---------|
| Serial0     |         | <->  | Serial0 |
| Port        | Sublink | Port | Sublink |
| 1           | Serial0 | 102  | Serial1 |
| 2           | Serial0 | 103  | Serial2 |
| 3           | Serial1 | 203  | Serial2 |

| INTERFACE |  | DLCI | Name |
|-----------|--|------|------|
| Serial0   |  | 102  | 102  |
|           |  | 103  | 103  |

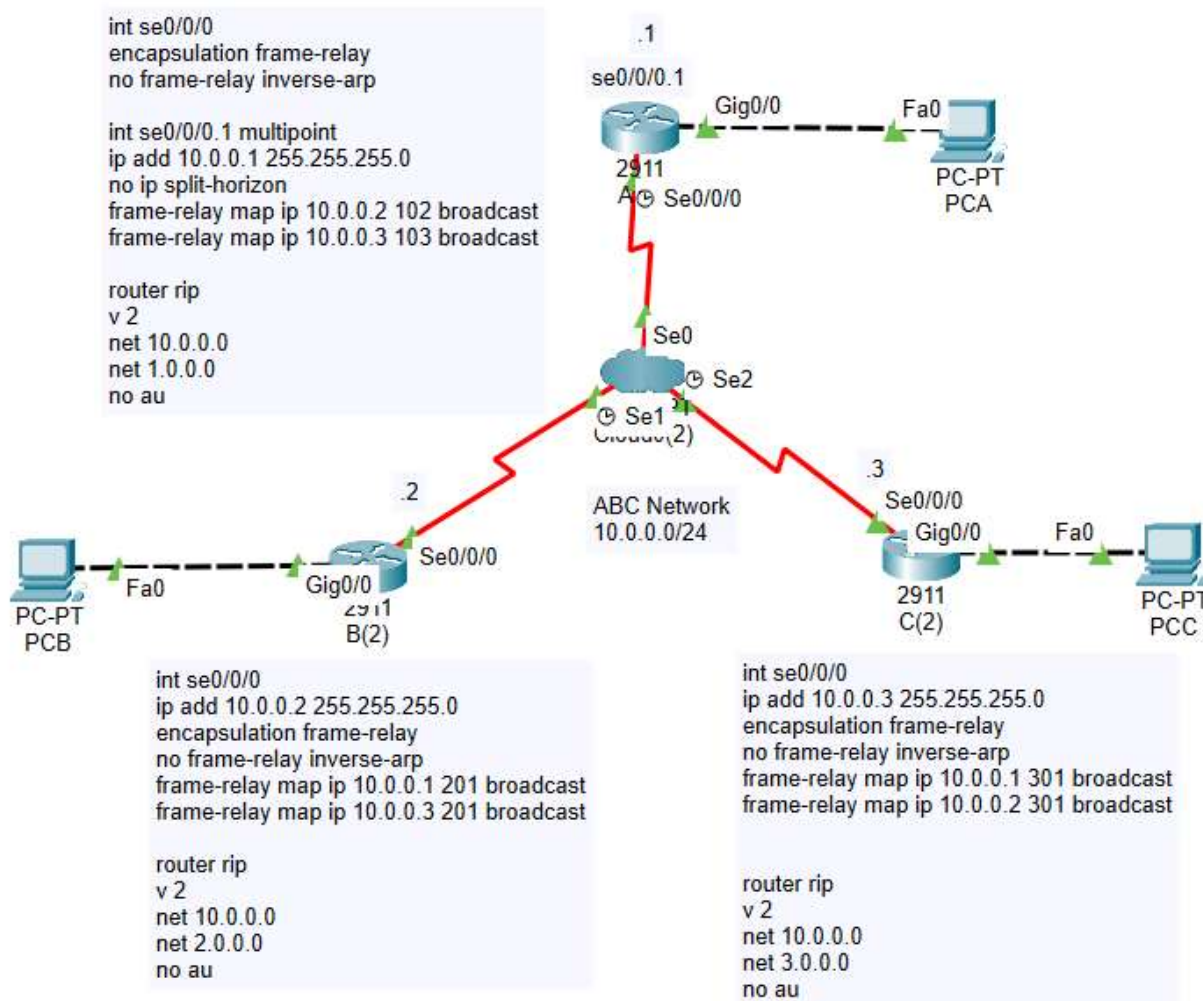
| INTERFACE |  | DLCI | Name |
|-----------|--|------|------|
| Serial0   |  | 201  | 201  |
| Serial1   |  | 203  | 203  |

| INTERFACE |  | DLCI | Name |
|-----------|--|------|------|
| Serial0   |  | 301  | 301  |
| Serial1   |  | 302  | 302  |

# 10-8-4. Frame-Relay(Multipoint via)

## 2. Multipoint: Port 1개와 port N개를 1:N로 DLCI 설정 (via)



| Frame Relay |         |      |         |
|-------------|---------|------|---------|
| Serial0     |         | <->  | Serial0 |
| Port        | Sublink | Port | Sublink |
| 1           | Serial0 | 102  | Serial1 |
| 2           | Serial0 | 103  | Serial2 |

|           |  |      |      |
|-----------|--|------|------|
| INTERFACE |  | DLCI | Name |
| Serial0   |  | 102  | 102  |
|           |  | 103  | 103  |

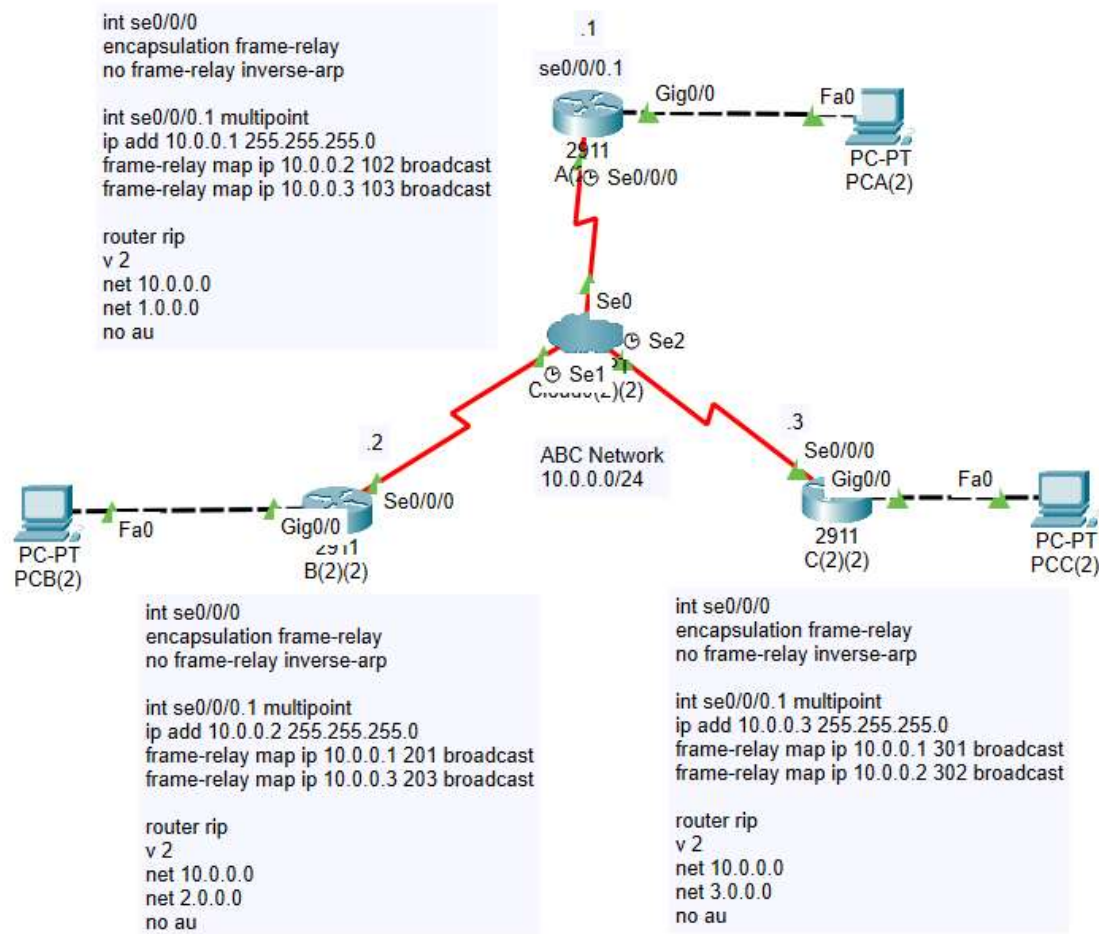
|           |  |      |      |
|-----------|--|------|------|
| INTERFACE |  | DLCI | Name |
| Serial0   |  | 201  | 201  |
| Serial1   |  |      |      |

|           |  |      |      |
|-----------|--|------|------|
| INTERFACE |  | DLCI | Name |
| Serial0   |  | 301  | 301  |
| Serial1   |  |      |      |
| Serial2   |  |      |      |



# 10-8-5. Frame-Relay(Multipoint mesh)

## 2. Multipoint: Port 1개와 port N개를 1:N로 DLCI 설정 (mesh)



| Frame Relay |         |      |         |
|-------------|---------|------|---------|
| Serial0     |         | <--> | Serial0 |
| Port        | Sublink | Port | Sublink |
| 1           | Serial0 | 102  | Serial1 |
| 2           | Serial0 | 103  | Serial2 |
| 3           | Serial1 | 203  | Serial2 |

| INTERFACE |  | DLCI | Name |
|-----------|--|------|------|
| Serial0   |  | 102  | 102  |
|           |  | 103  | 103  |

| INTERFACE |  | DLCI | Name |
|-----------|--|------|------|
| Serial0   |  | 201  | 201  |
|           |  | 203  | 203  |

| INTERFACE |  | DLCI | Name |
|-----------|--|------|------|
| Serial1   |  | 301  | 301  |
|           |  | 302  | 302  |

## 10-8-6. Frame-Relay(Practice)

