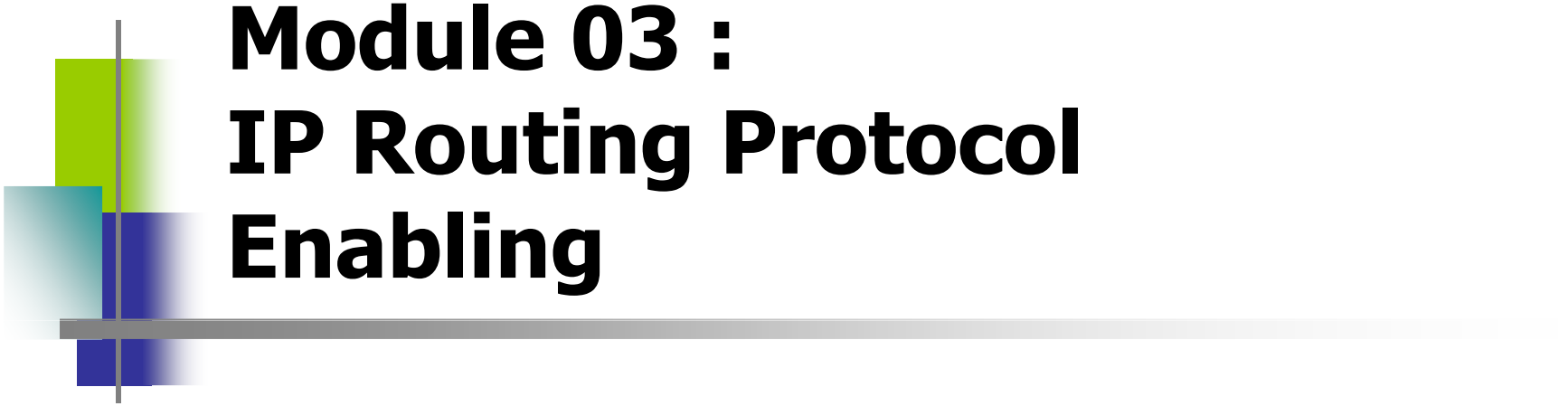
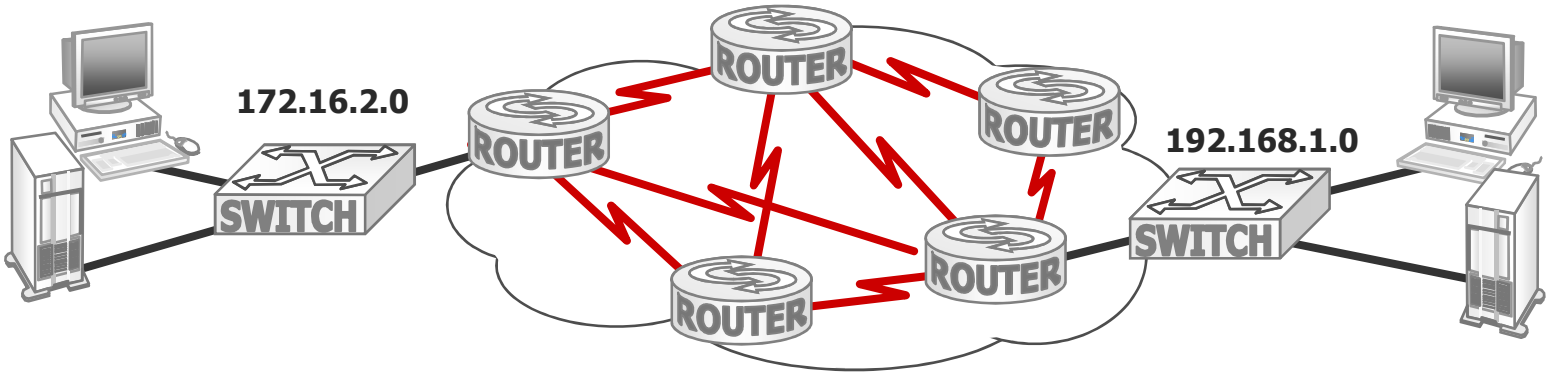


# Module 03 : IP Routing Protocol Enabling

# **Routing**의 개념 소개

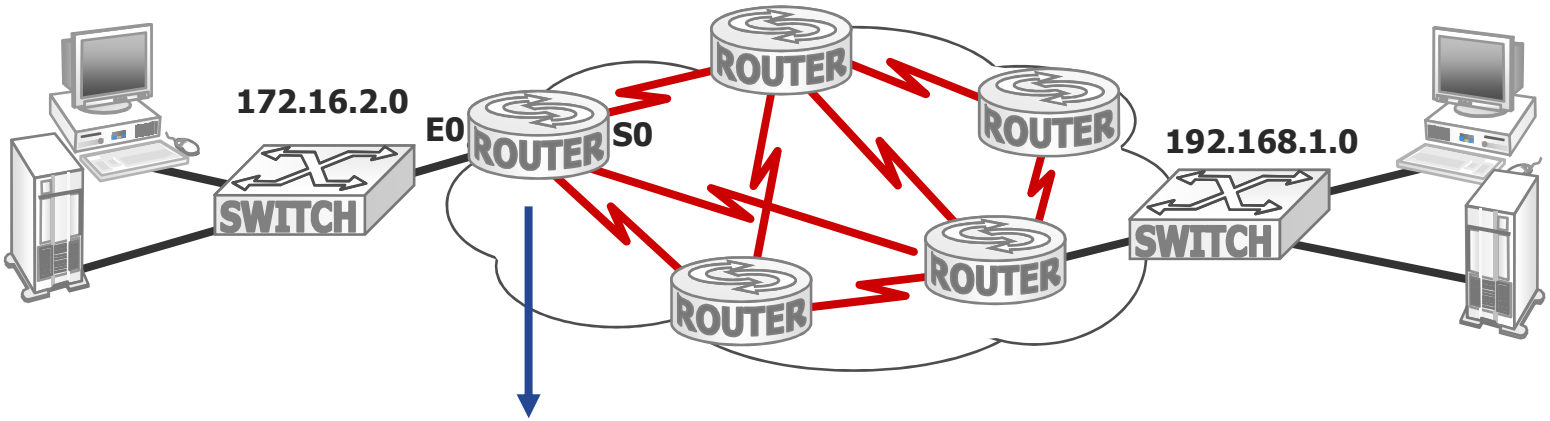
- **Routing** 개요
- **Static**과 **Dynamic Routing**
- **Static Route** 설정하기
- **Default Route** 설정하기
- **Dynamic Routing** 개요
- **IP Classless** 소개
- **Inter-VLAN Routing**

# Routing 개요




- 데이터를 최적의 경로를 선택하여 목적지까지 이송하는 전 절차
- **IP Protocol**이 올라가있는 **Router, Computer** 또는 **Host, L-3 Ethernet Switch**들이 이러한 작업을 수행 가능
- 라우터가 데이터를 **Routing**하기 위하여 알아야 하는 것
  - 소스와 목적지 주소
  - 입·출력 인터페이스 형태
  - 가능성 있는 모든 경로**(route)**들에 대한 정보 수집
  - 가능성 있는 모든 경로 중에 최적의 경로 선택
  - 지속적인 네트워크 상태를 확인하고 유지하는 것

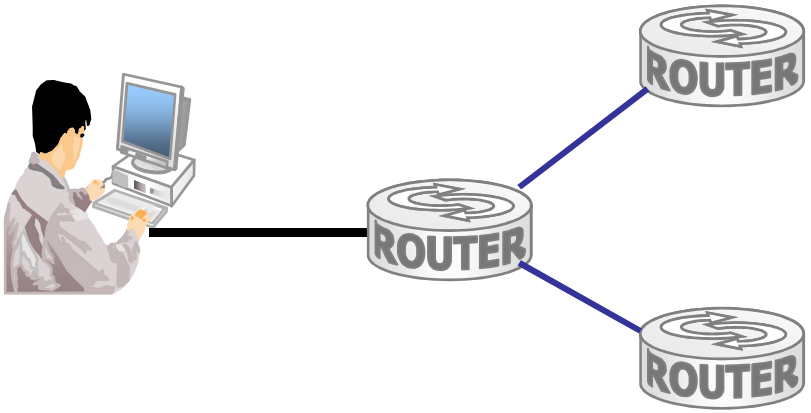
# Routing 개요

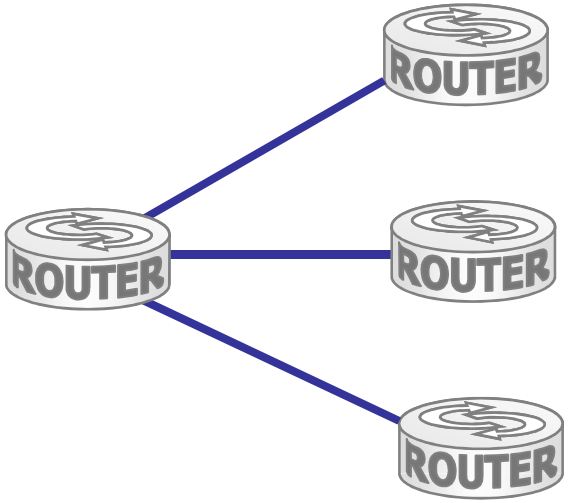



| 네트워크 프로토콜 | 목적지 네트워크 | 나가는 인터페이스 |
|---|---|---|
| **Connected** | **172.16.2.0** | **E0** |
| **Learned** | **192.168.1.0** | **S0** |

라우터는 직접 연결된 곳 이외의 다른 목적지에 대해서는 반드시 **Connected, Static, Dynamic Routing Protocol, Redistribution** 과 같은 다양한 방법으로 학습 할 수 있다.

# Static과 Dynamic Routing



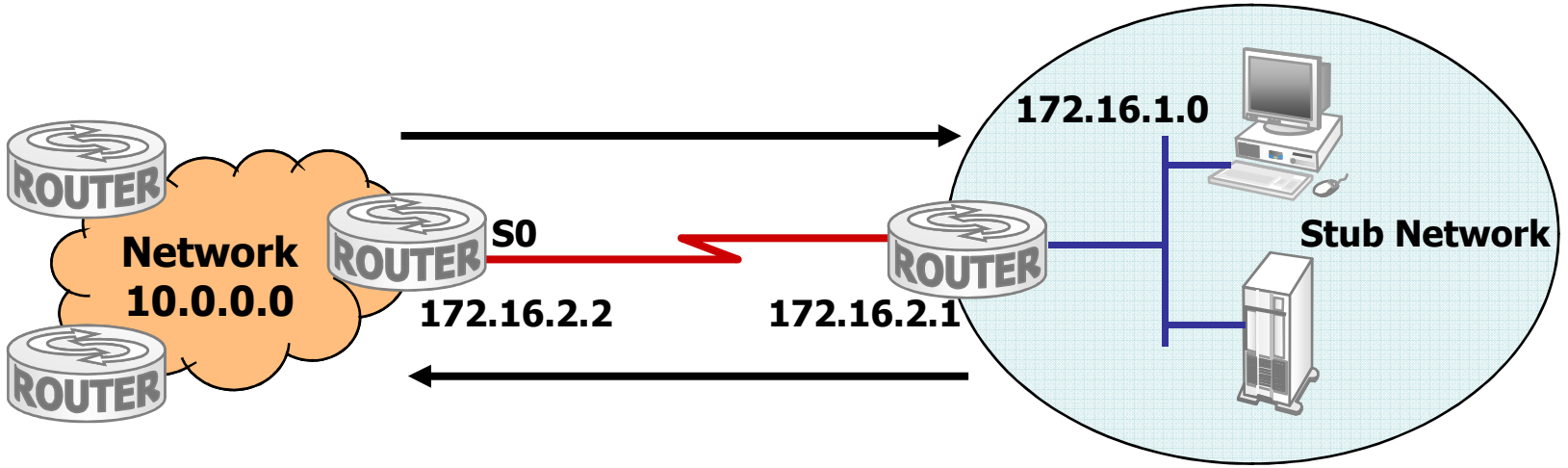

## Static Routing

- 관리자가 직접 수동으로 **Router**에게 필요한 **Route** 정보들을 입력한다.
- **Network**의 변화에 대해 **Router**는 자동으로 반응하지 못하며 관리자가 직접 **Network**의 변화를 **Router**에 설정해야 한다.

## Dynamic Routing

- **Routing Protocol**을 이용하여 자동으로 **Route** 정보들을 수집한다.
- **Network** 변화에 대해 자동으로 반응한다.

# Static Route 설정하기

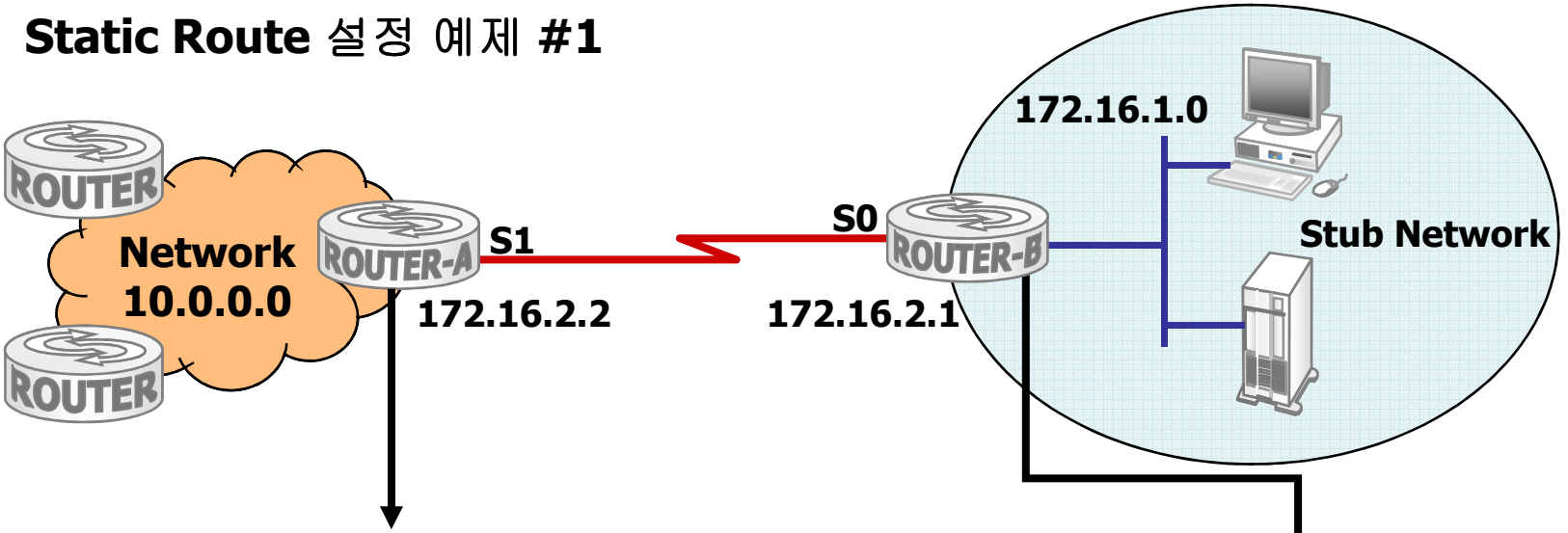



**Router#(config)ip route *network* [*mask* ] {*address* | *interface*} [*distance*] [permanent]**

| | |
|---|---|
| ***Network*** | 도착지 **Network** |
| ***Mask*** | 도착지 **Network**의 **Subnet mask** |
| ***Address*** | 도착지 **network**로 도달하기 위한 **Next-hop address** |
| ***Interface*** | 도착지 **network**로 도달하기 위한 **Next-hop Router**와 연결된 **local interface** |
| ***Distance*** | 정의된 **Route**의 **Administrator Distance** 값 |
| ***Permanent*** | 정의된 **Static Route**가 **Routing table**에서 제거되지 않도록 한다. |

# Static Route 설정하기

## Static Route 설정 예제 #1




```
Router-A(config)#ip route 172.16.1.0  255.255.255.0  172.16.2.1
```

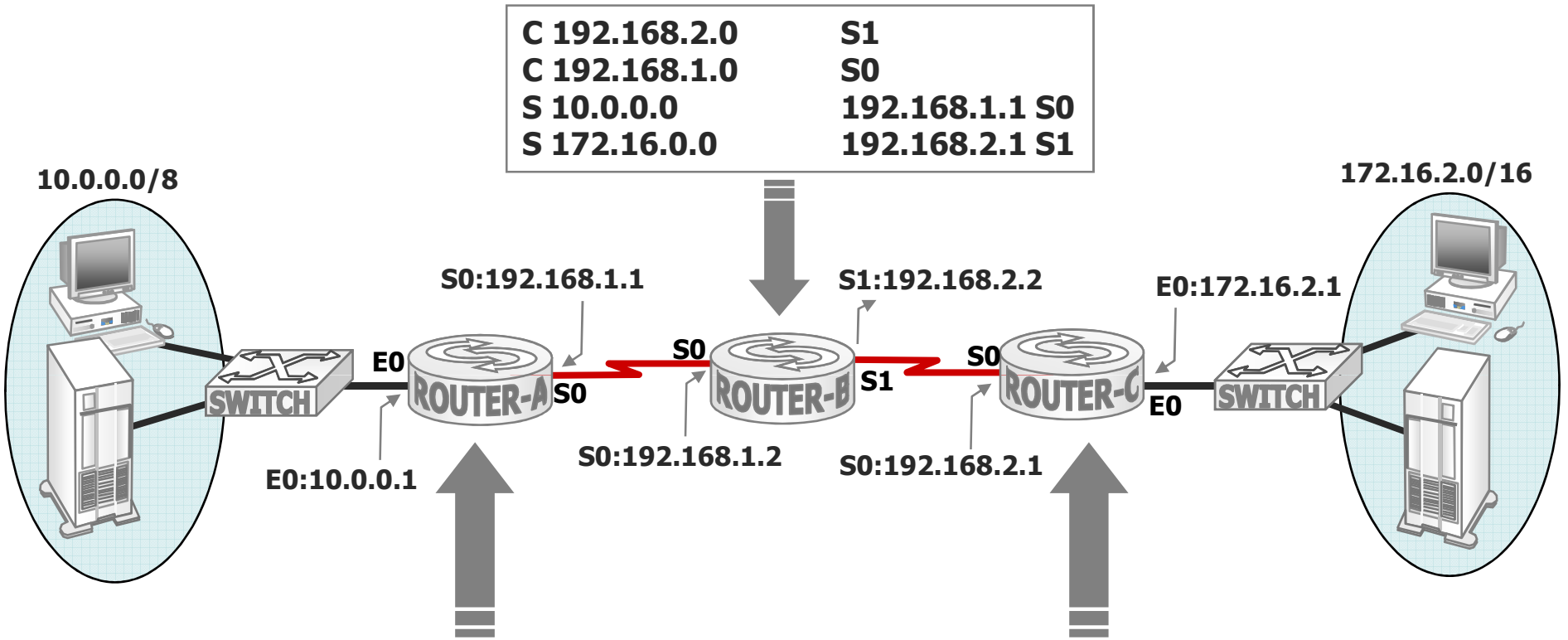
```
Router(config)#ip route 10.0.0.0  255.0.0.0  172.16.2.2
```

```
Router(config)#ip route 10.0.0.0  255.0.0.0  Serial 0
```

위 구성의 결과로 **Router-A**와 **Router-B**는 자신들이 포함된 모든 **Network**에 대해 통신이 가능한가 **?**

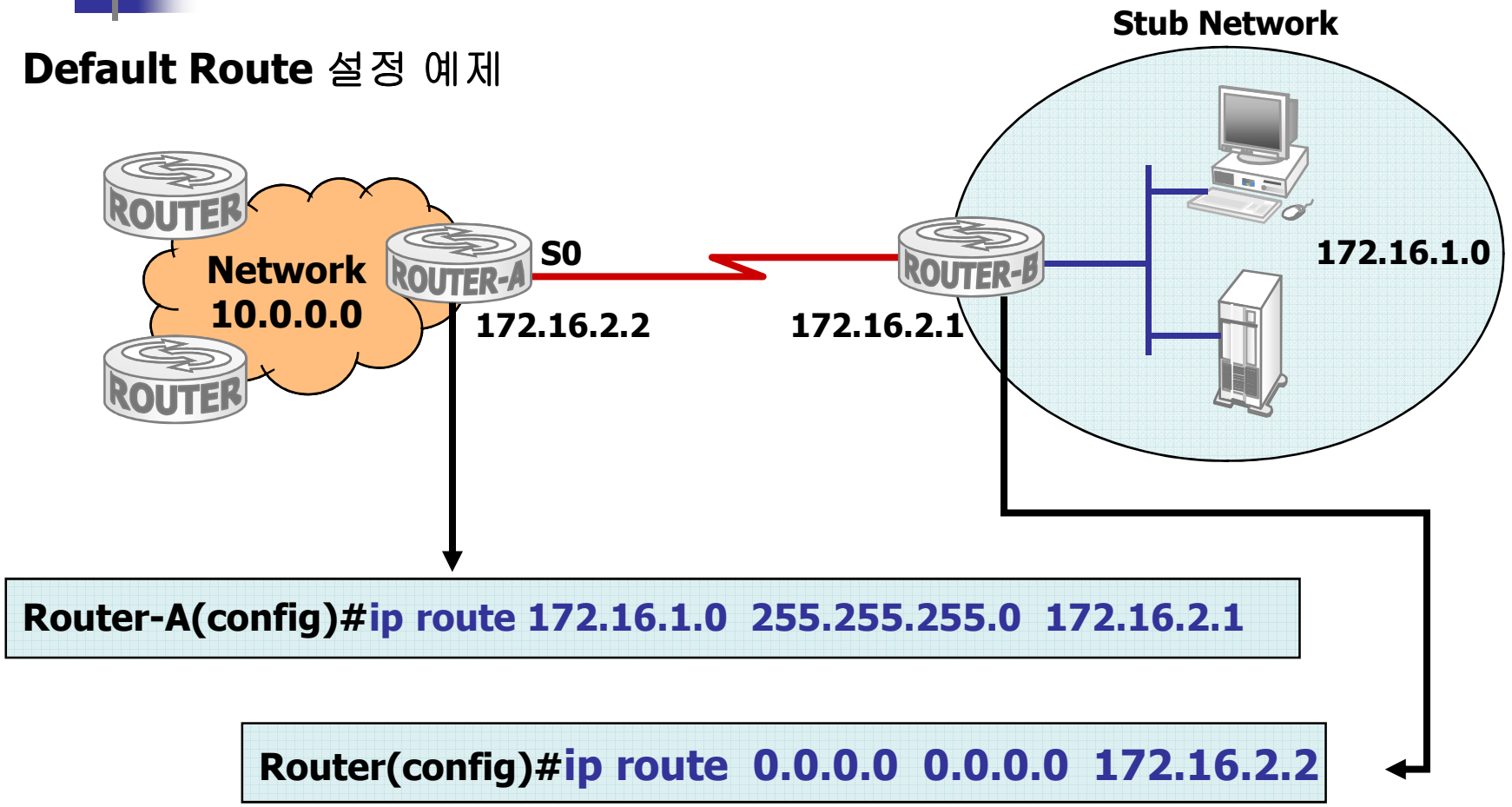
# Static Route 설정하기

## Static Route 설정 예제 #2



# Default Route 설정하기

**Default Route** 설정 예제



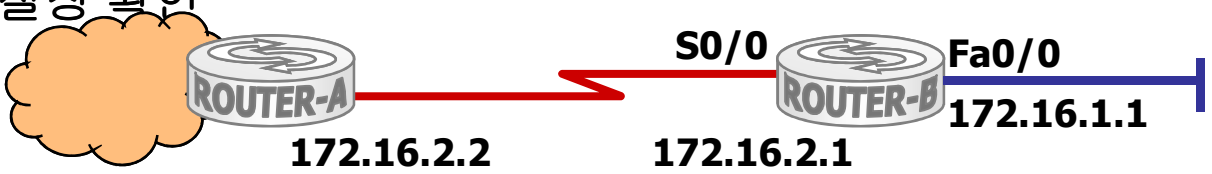

**Router-A(config)#ip route 172.16.1.0 255.255.255.0 172.16.2.1**

**Router(config)#ip route 0.0.0.0 0.0.0.0 172.16.2.2**

위의 구성의 결과로 **Stub Network**의 모든 **Host**들은 전체 **Network**에 대해 통신이 가능하다 .

# Default Route 설정하기

## Default Route 설정 확인

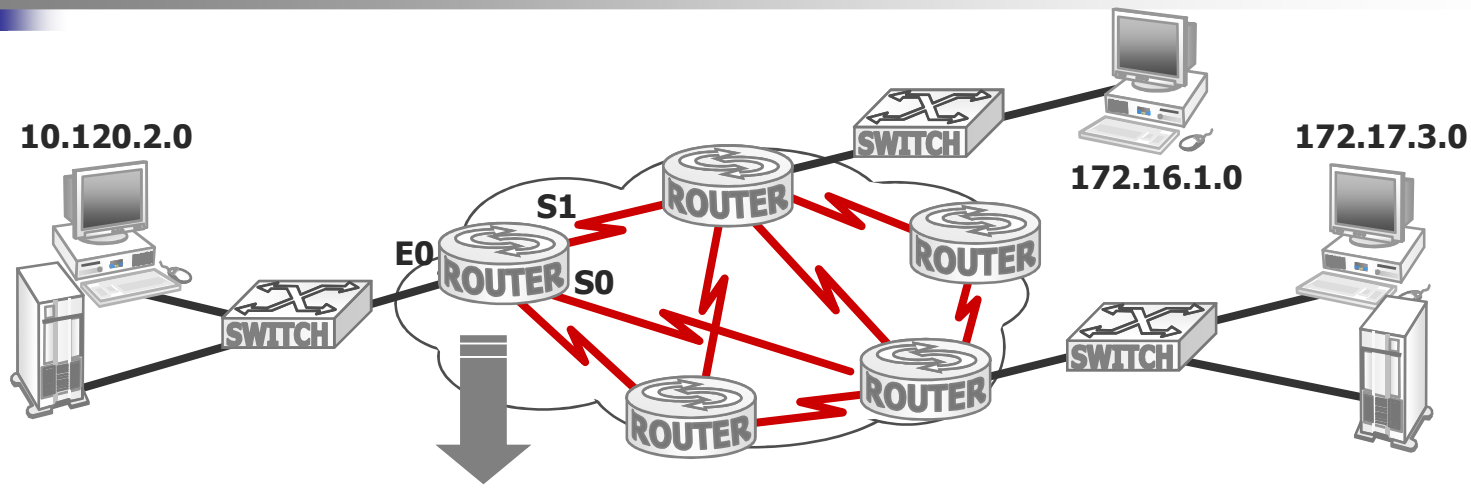



```
Router-B#config t
Router-B(config)#ip route 0.0.0.0 0.0.0.0 172.16.2.2
Router-B(config)#exit
Router-B#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, 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

    172.16.0.0/24 is subnetted, 3 subnets
C       172.16.1.0 is directly connected, FastEthernet0/0
C       172.16.2.0 is directly connected, Serial0/0
S*      0.0.0.0/0 [ 1/0 ] via 172.16.2.2
Router-B#
```

# 개관

- **Routing**의 개념 소개
- **Distance Vector Routing Protocol**
- **Link State & Hybrid Routing Protocol**
- **RIP** 구성하기
- **IGRP** 구성하기
- **VLSM** 개요

# Dynamic Routing 개요




| 라우팅 프로토콜 | 목적지 네트워크 | 나가는 인터페이스 |
|---|---|---|
| **Connected** | **10.120.2.0** | **E0** |
| **RIP** | **172.16.2.0** | **S0** |
| **IGRP** | **172.17.3.0** | **S1** |

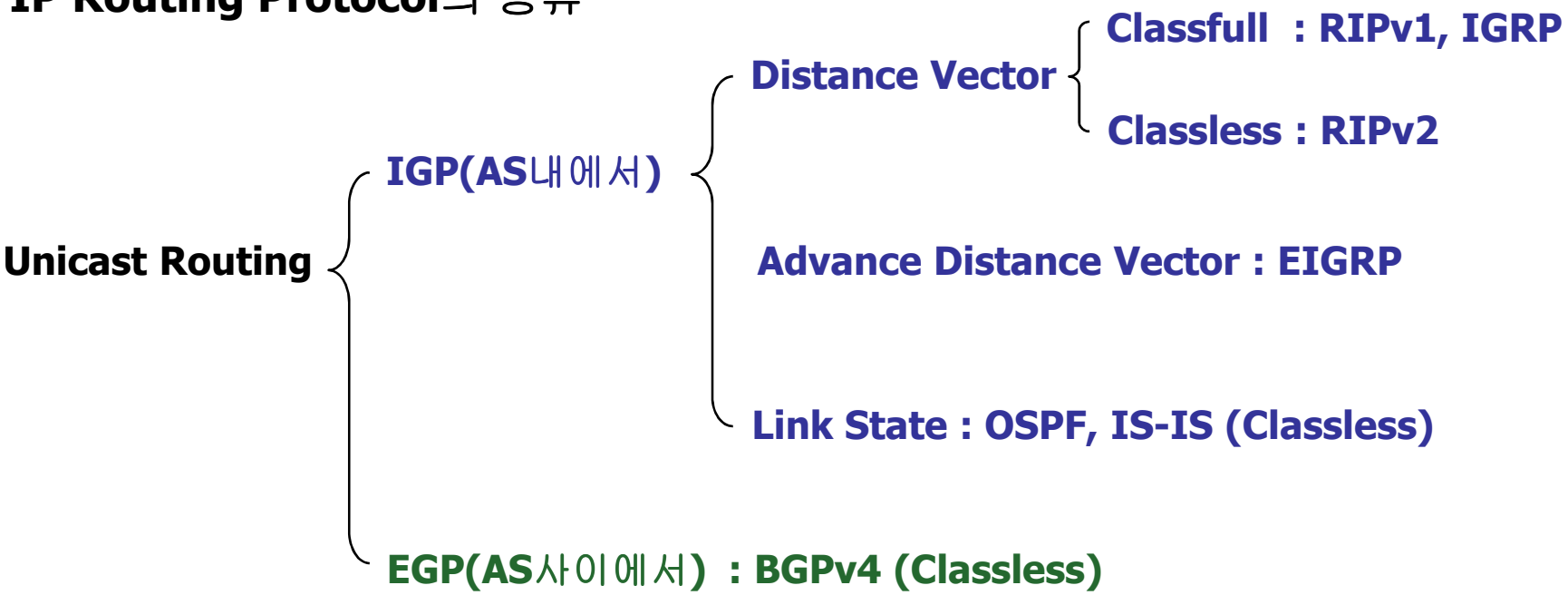
- **Dynamic Routing Protocol**은 네트워크 정보를 교환하여 최적의 경로를 결정하고, 라우팅 테이블을 지속적으로 유지한다.
- 하나의 경로가 결정되면 라우터는 **Routed Protocol**들을 라우트 할 수 있다.

◆ **Routed Protocol : IP,IPX, AppleTalk**

◆ **Routing protocol : RIP, IGRP, EIGRP,OSPF,IS-IS,BGP, DVMRP, MOSPF, PIM Dense & Sparse**

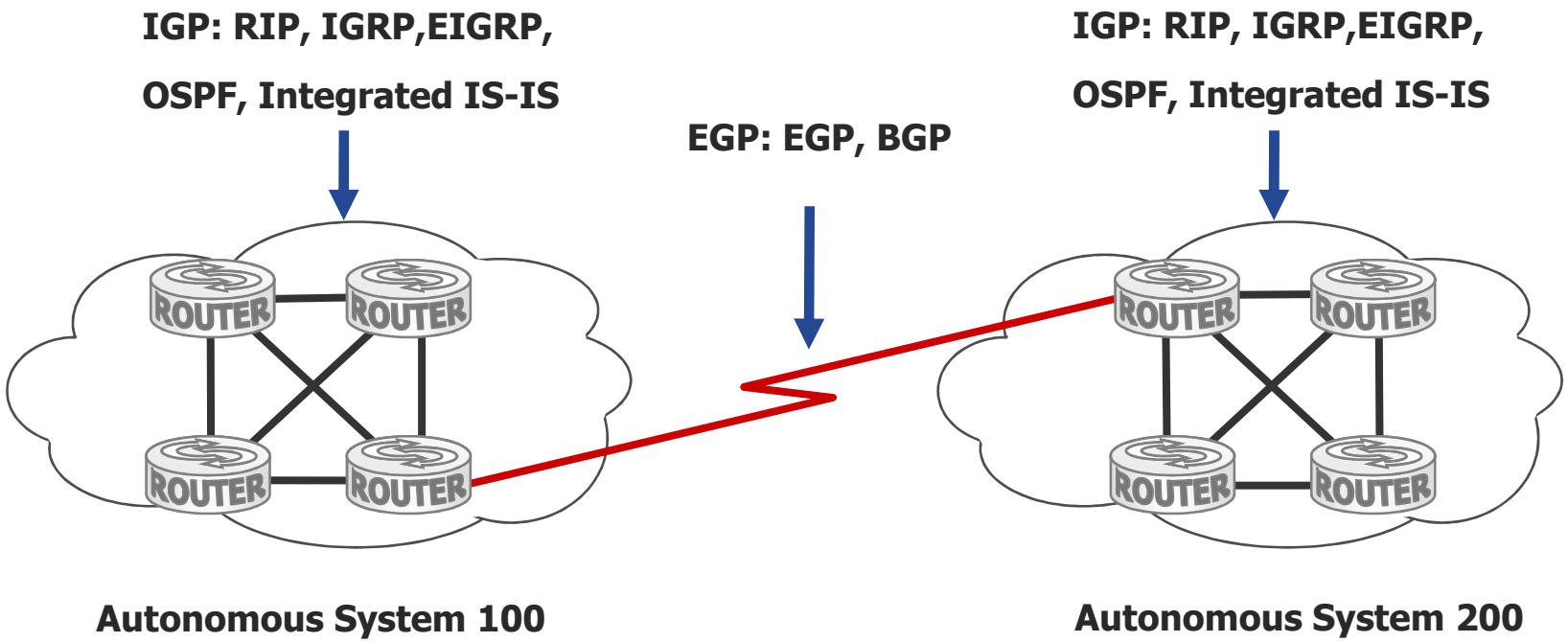
# Dynamic Routing 개요

**IP Routing Protocol의 종류**




**Multicast Routing :** **DVMRP, MOSPF, PIM Dense & Sparse**

# Dynamic Routing 개요



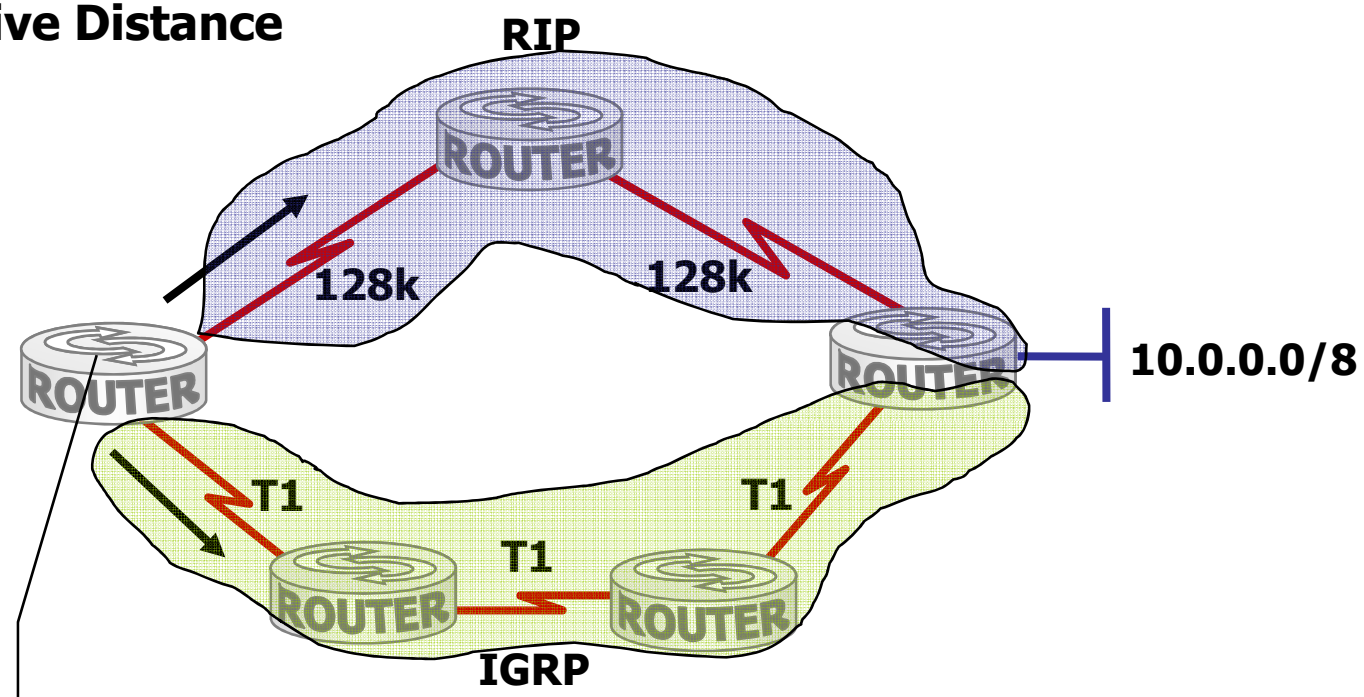

- **Autonomous System**은 일반적인 관리영역 하에 있는 네트워크들의 집합
- **IGP**들은 **Autonomous System**안에서 운영
- **EGP**들은 다른 **Autonomous System**간에 운영

# Dynamic Routing 개요

## Administrative Distance

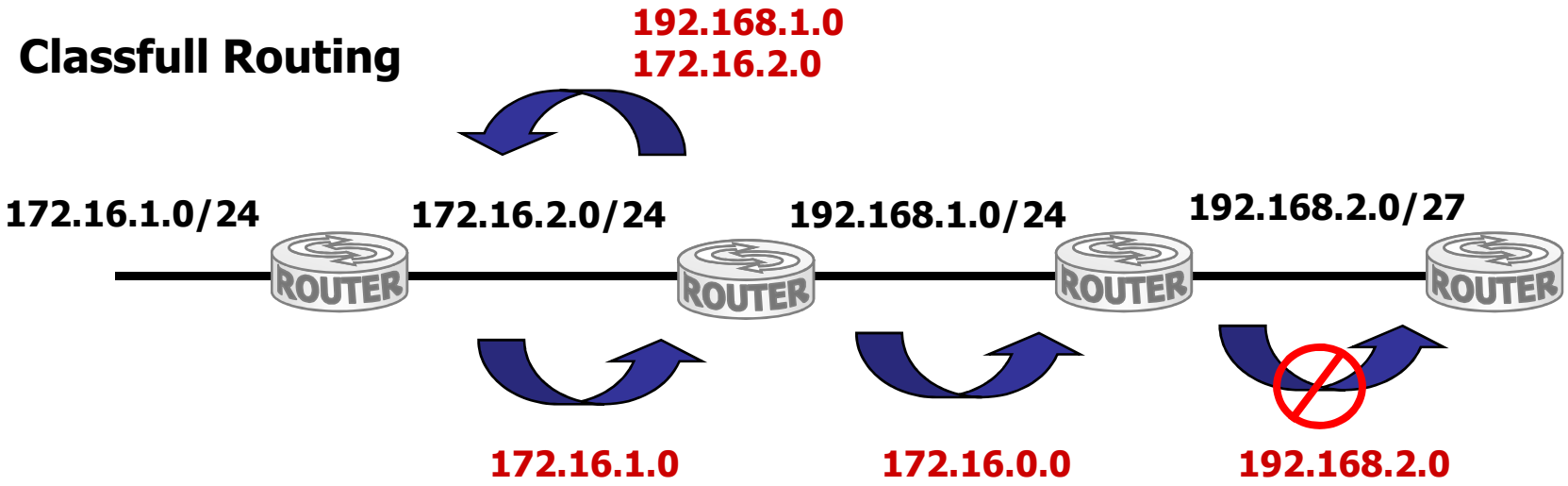
**RIP Administrative Distance = 120**

**IGRP Administrative Distance = 100**



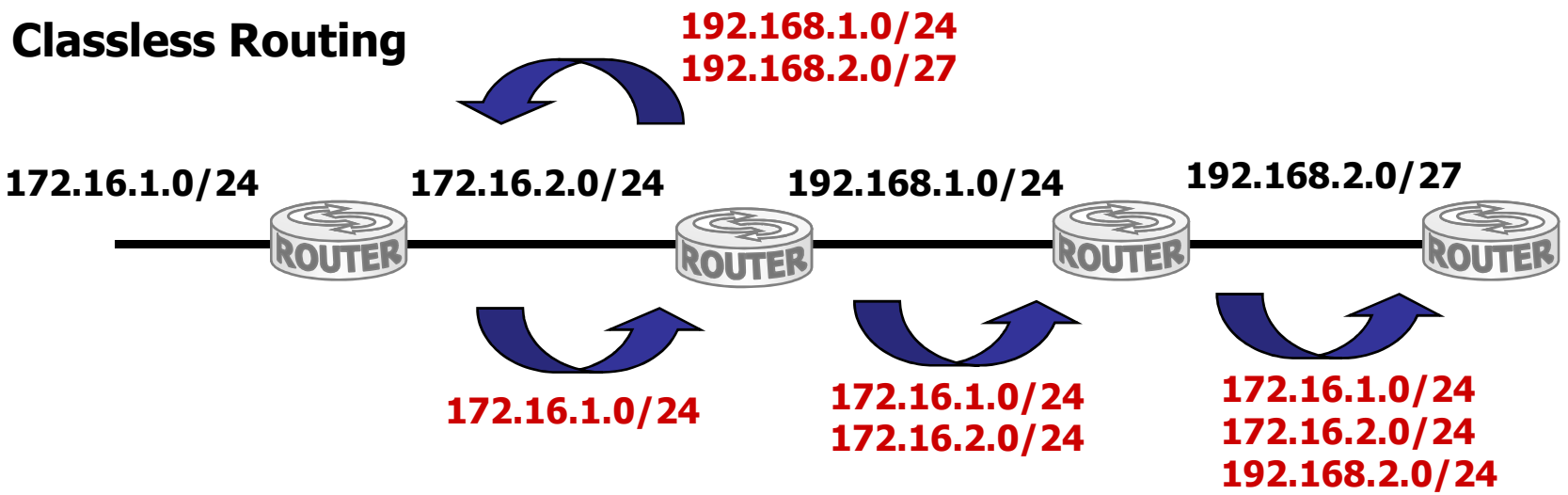

**10.0.0.0/8**로 가기 위해서 어떤 **Routing Protocol**이 학습한 경로를 더 신뢰할 것인가 **?**

# Dynamic Routing 개요




- **Routing** 정보 전달 시에 **Subnet mask** 정보를 전달 하지 않는다.
- 같은 **network**에 연결된 **Router**들은 같은 **Subnet mask**로 설정 되어 있다고 가정한다.
- **Network**이 다른 **router**와 **Routing** 정보 교환 시는 자동으로 **Classfull** 경계를 기반으로 **Summary**된 정보를 전달한다.
- **RIP Version 1**과 **IGRP**가 이곳에 속한다.

# Dynamic Routing 개요




- **Routing** 정보 전달 시에 **Subnet mask** 정보를 함께 전달한다.
- **network**에 연결된 **Router**들은 다양한 **Subnet mask**로 설정 되어 있을 수 있다. **(VLASM** 지원**)**
- **Network**이 다른 **router**와 **Routing** 정보 교환 시는 수동으로 **Summary**된 정보를 전달 할 수도 있다.
- **RIP Version 1**과 **IGRP**를 제외한 모든 **Routing Protocol**이 이를 지원 한다.

## Routing Protocol들의 비교 #1

| 특징들 | RIPv1 | RIPv2 | IGRP | EIGRP** | OSPF |
|---|---|---|---|---|---|
| Distance vector | x | x | x | x | |
| Link-state | | | | | x |
| Classful (auto route summ.) | x | x | x | x | |
| Classless (VLSM support) | | x | | x | x |
| Proprietary | | | x | x | |
| Scalability | Small | Small | Med. | Large | Large |
| Convergence time | Slow | Slow | Slow | Fast | Fast |

** EIGRP는Advanced Distance Vector Protocol

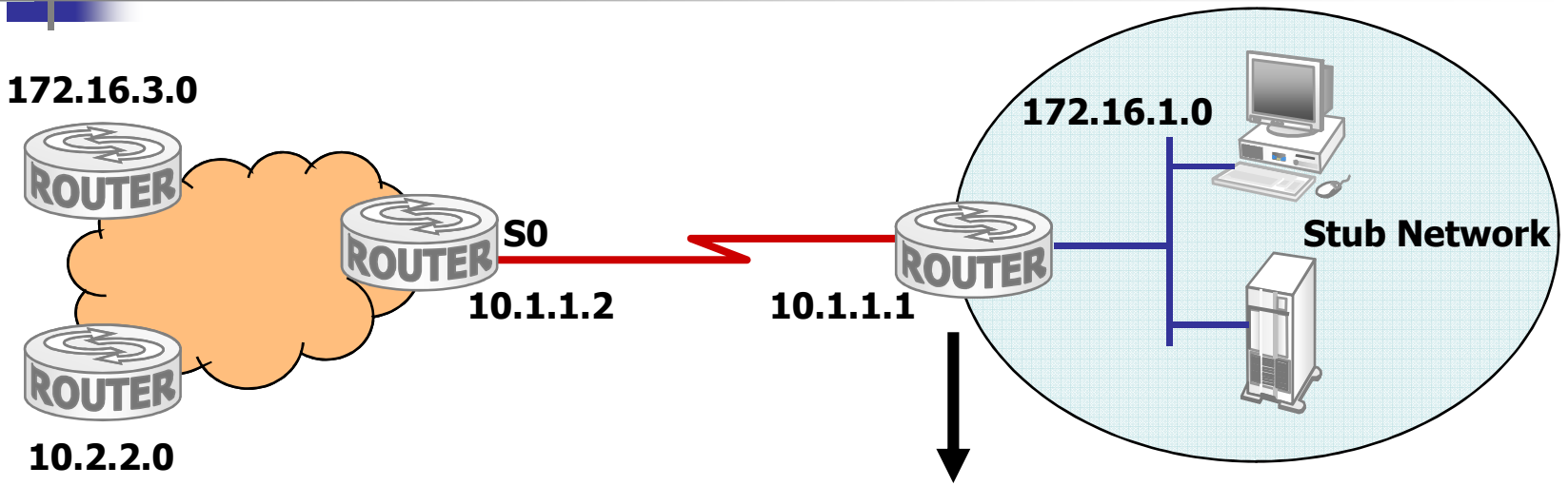
# Dynamic Routing 개요

## Routing Protocol들의 비교 #2

| Characteristic | RIPv1 | RIPv2 | IGRP | EIGRP ** |
|---|---|---|---|---|
| Count to infinity | x | x | x | |
| Split horizon | x | x | x | x |
| Hold-down timer | x | x | x | |
| Triggered updates with route poisoning | x | x | x | x |
| Load balancing—Equal paths | x | x | x | x |
| Load balancing—Unequal paths | | | x | x |
| VLSM support | | x | | x |
| Routing algorithm | B-F | B-F | B-F | DUAL |
| Metric | Hops | Hops | Comp | Comp |
| Hop count limit | 16 | 16 | 100 | 100 |
| Scalability | Med | Med | Large | Large |

** EIGRP 는 Advanced Distance Vector Protocol (Hybrid)

# IP Classless 개요



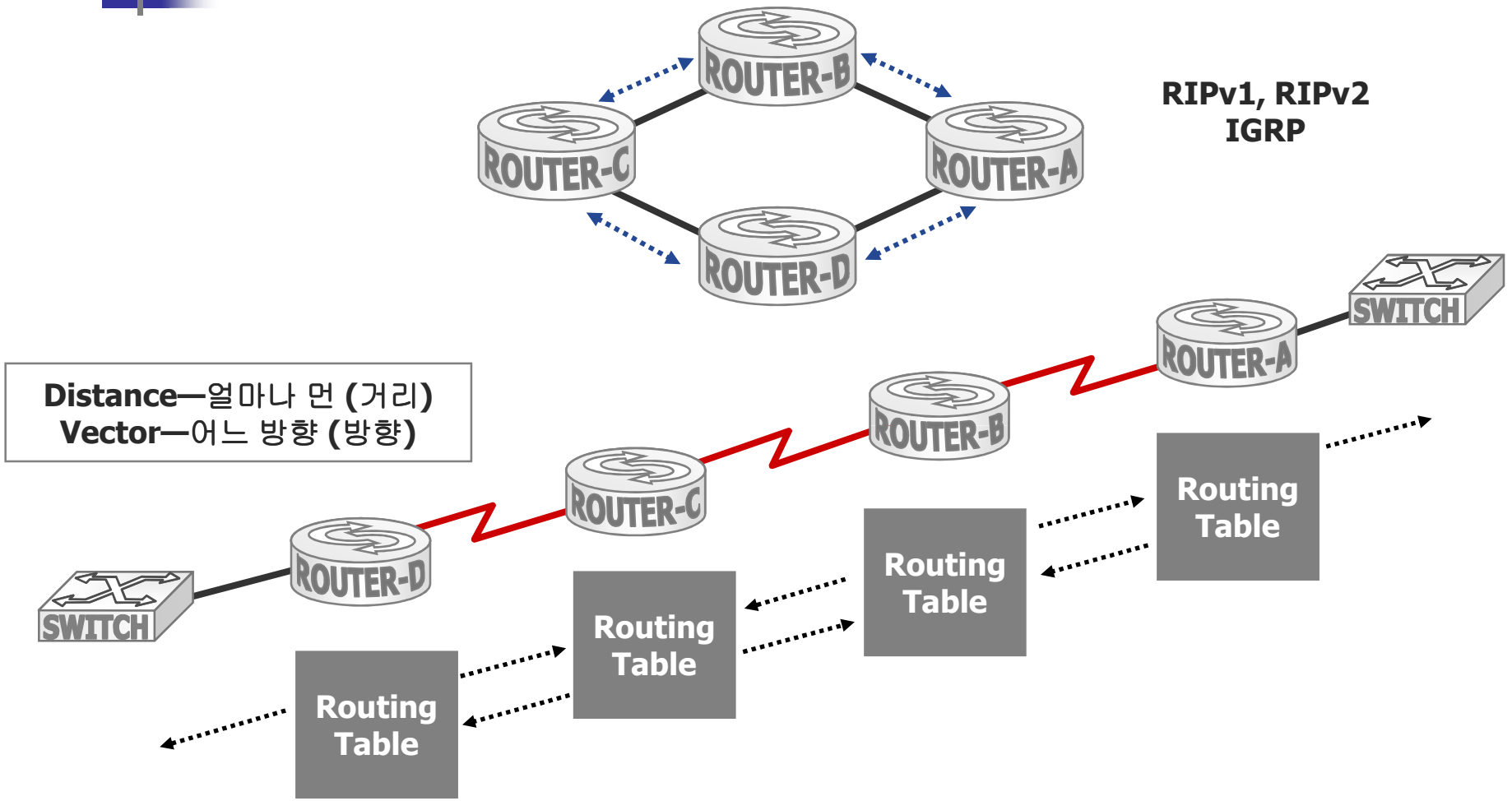

```
Router(config)#ip route 0.0.0.0  0.0.0.0  10.1.1.2
Router(config)#ip classless
Router#ping 10.2.2.0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
.....
```

```
Router(config)#ip route 0.0.0.0  0.0.0.0  10.1.1.2
Router(config)#no ip classless
Router#ping 10.2.2.0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!!
```

# Distance Vector Routing

- **Distance Vector Routing** 개요
- **Distance Vector**의 경로 선택
- **Routing** 정보 관리
- **Distance Vector**의 **Routing Loop**
- **Routing Loop** 문제 해결
- **Distance Vector**의 **Operation**

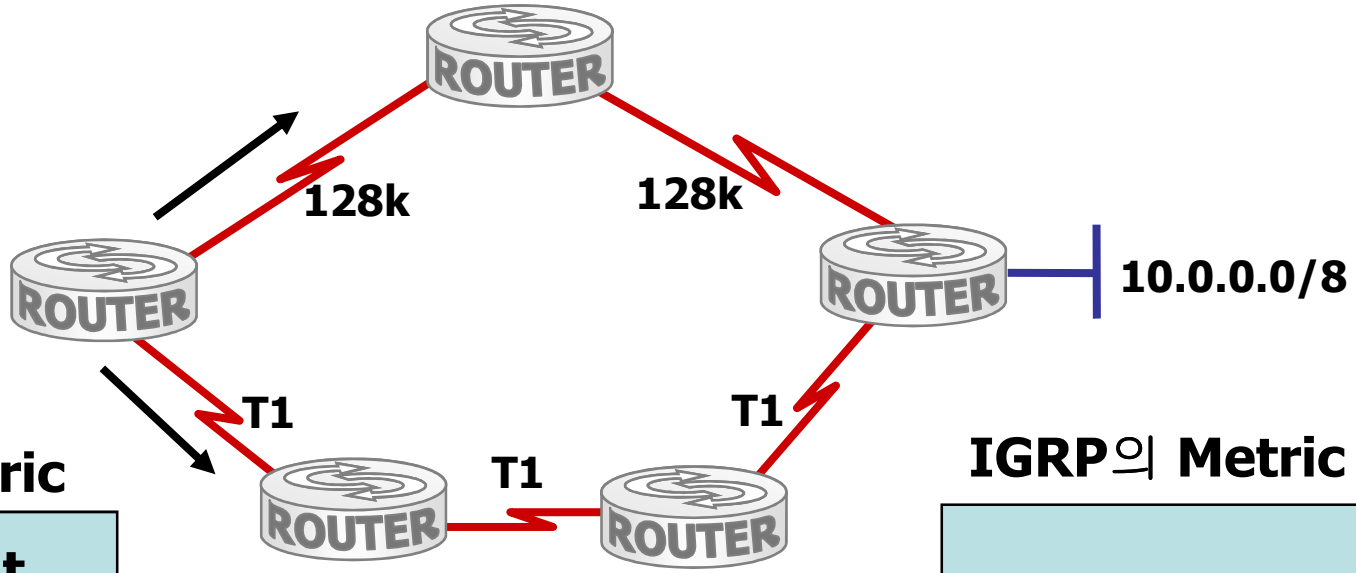
# Distance Vector Routing의 개요




- 최적의 라우팅 리스트를 만든 후, 주기적으로 라우팅 테이블을 인접관계에 있는 라우터에게 전달한다.

# Distance Vector의 경로 선택

## Distance Vector의 Metric




**RIP의 Metric**

**Hop Count**

**IGRP의 Metric**

**Bandwidth**
**Delay**
**Load**
**Reliability**
**MTU**

# Distance Vector Routing의 개요

## Distance Vector의 경로 정보 수집


**Routing Table**

| | | |
|---|---|---|
| 10.1.0.0 | E0 | 0 |
| 10.2.0.0 | S0 | 0 |
| 10.3.0.0 | S0 | 1 |
| 10.4.0.0 | S0 | 2 |

**Routing Table**

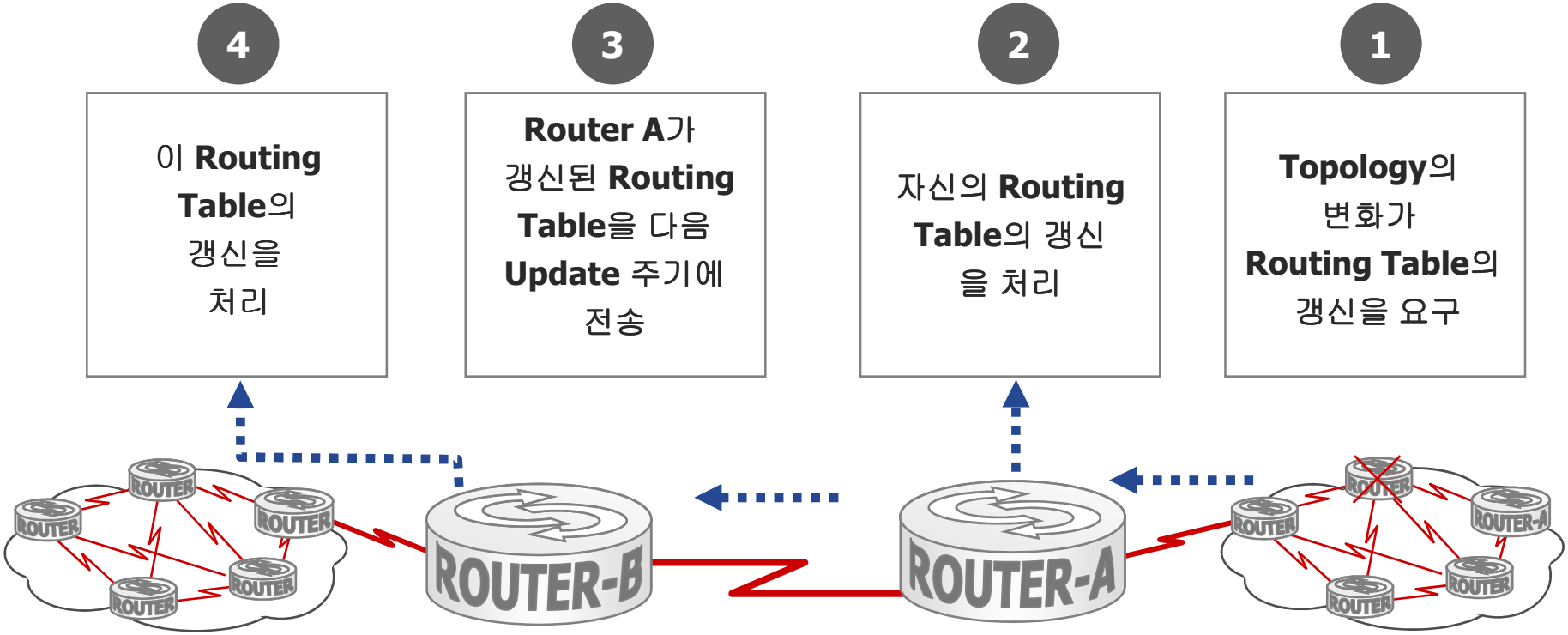
| | | |
|---|---|---|
| 10.2.0.0 | S0 | 0 |
| 10.3.0.0 | S1 | 0 |
| 10.4.0.0 | S1 | 1 |
| 10.1.0.0 | S0 | 1 |

**Routing Table**

| | | |
|---|---|---|
| 10.3.0.0 | S0 | 0 |
| 10.4.0.0 | E0 | 0 |
| 10.2.0.0 | S0 | 1 |
| 10.1.0.0 | S0 | 2 |

**Router**들은 **Network**상에 각 **Destination**에 대해 최적의 경로를 선택 후 이를 관리 및 유지 한다.

# Routing 정보 관리




- **Update** 절차는 라우터에서 다른 라우터로 **Step-by-Step**으로 진행

# Distance Vector의 Routing Loop

## Routing Loop 예제


**Routing Table**

| | | |
|---|---|---|
| 10.1.0.0 | E0 | 0 |
| 10.2.0.0 | S0 | 0 |
| 10.3.0.0 | S0 | 1 |
| 10.4.0.0 | S0 | 2 |

**Routing Table**

| | | |
|---|---|---|
| 10.2.0.0 | S0 | 0 |
| 10.3.0.0 | S1 | 0 |
| 10.4.0.0 | S1 | 1 |
| 10.1.0.0 | S0 | 1 |

**Routing Table**

| | | |
|---|---|---|
| 10.3.0.0 | S0 | 0 |
| 10.4.0.0 | E0 | 0 |
| 10.2.0.0 | S0 | 1 |
| 10.1.0.0 | S0 | 2 |

일반적인 상황에서의 **Routing Table**

# Distance Vector의 Routing Loop

## Routing Loop 예제 (계속..)


**Routing Table**

| | | |
|---|---|---|
| 10.1.0.0 | E0 | 0 |
| 10.2.0.0 | S0 | 0 |
| 10.3.0.0 | S0 | 1 |
| 10.4.0.0 | S0 | 2 |

**Routing Table**

| | | |
|---|---|---|
| 10.2.0.0 | S0 | 0 |
| 10.3.0.0 | S1 | 0 |
| 10.4.0.0 | S1 | 1 |
| 10.1.0.0 | S0 | 1 |

**Routing Table**

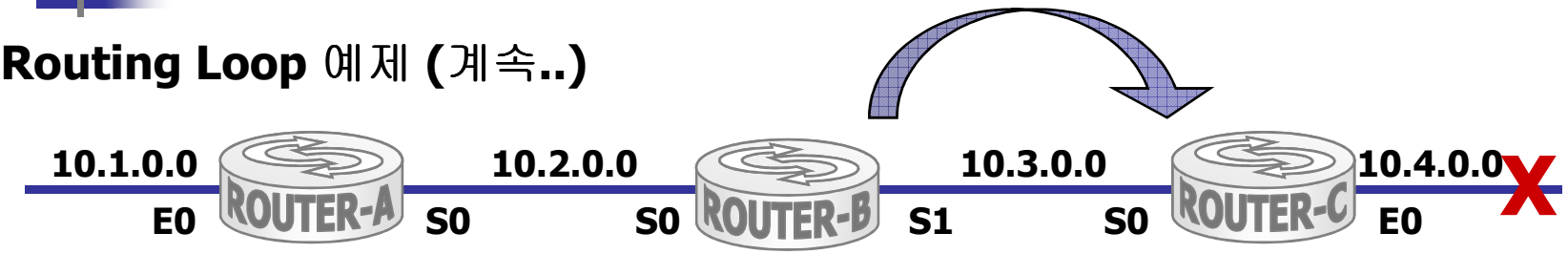
| | | |
|---|---|---|
| 10.3.0.0 | S0 | 0 |
| ~~10.4.0.0~~ | ~~E0~~ | Down |
| 10.2.0.0 | S0 | 1 |
| 10.1.0.0 | S0 | 2 |

◆ **Router-C**에서 **Down**된 **Ethernet** 구간은 **Routing Table**에서 경로 제거

◆ 이러한 **Topology**의 변화가 다른 **Router**에게 얼마나 빨리 전달 되는가 **?**

# Distance Vector의 Routing Loop

## Routing Loop 예제 (계속..)




**Routing Table**

| | | |
|---|---|---|
| 10.1.0.0 | E0 | 0 |
| 10.2.0.0 | S0 | 0 |
| 10.3.0.0 | S0 | 1 |
| 10.4.0.0 | S0 | 2 |

**Routing Table**

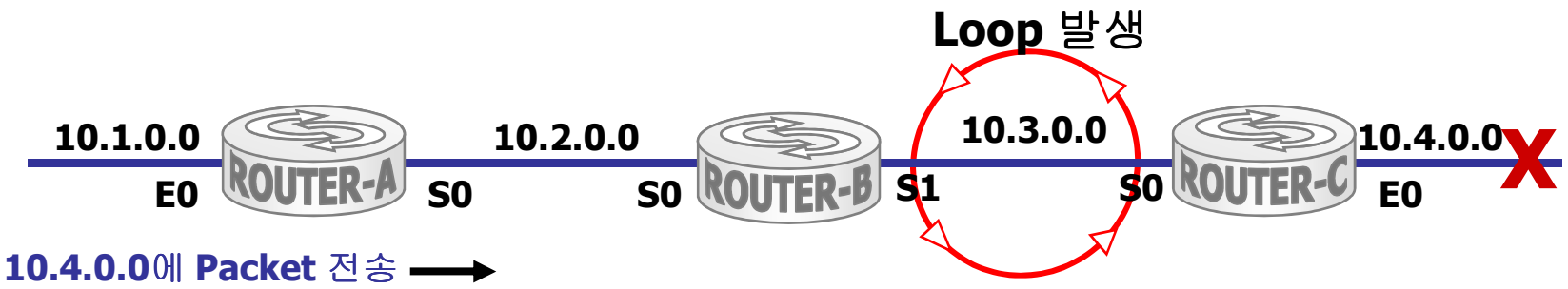
| | | |
|---|---|---|
| 10.2.0.0 | S0 | 0 |
| 10.3.0.0 | S1 | 0 |
| 10.4.0.0 | S1 | 1 |
| 10.1.0.0 | S0 | 1 |

**Routing Table**

| | | |
|---|---|---|
| 10.3.0.0 | S0 | 0 |
| 10.4.0.0 | S0 | 2 |
| 10.2.0.0 | S0 | 1 |
| 10.1.0.0 | S0 | 2 |

- **Distance Vector**는 **Network Topology**를 이해하지 못한다.
- **Router-C**는 **Router-B**에서 받은 **Routing** 정보를 기반으로 **10.4.0.0**에 도달 할 수 있는 또 다른 경로가 있다고 판단한다.

# Routing Loop 문제 해결




**Routing Table**

| | | |
|---|---|---|
| 10.1.0.0 | E0 | 0 |
| 10.2.0.0 | S0 | 0 |
| 10.3.0.0 | S0 | 1 |
| 10.4.0.0 | S0 | 4 |

**Routing Table**

| | | |
|---|---|---|
| 10.2.0.0 | S0 | 0 |
| 10.3.0.0 | S1 | 0 |
| 10.4.0.0 | S1 | 3 |
| 10.1.0.0 | S0 | 1 |

**Routing Table**

| | | |
|---|---|---|
| 10.3.0.0 | S0 | 0 |
| 10.4.0.0 | S0 | 2 |
| 10.2.0.0 | S0 | 1 |
| 10.1.0.0 | S0 | 2 |

**Router-B와 Router-C 사이에서 Routing Loop 발생**

# Distance Vector의 Routing Loop

## Routing Loop 예제 (계속..)


**Routing Table**

| | | |
|---|---|---|
| 10.1.0.0 | E0 | 0 |
| 10.2.0.0 | S0 | 0 |
| 10.3.0.0 | S0 | 1 |
| 10.4.0.0 | S0 | 4 |

**Routing Table**

| | | |
|---|---|---|
| 10.2.0.0 | S0 | 0 |
| 10.3.0.0 | S1 | 0 |
| 10.4.0.0 | S1 | 3 |
| 10.1.0.0 | S0 | 1 |

**Routing Table**

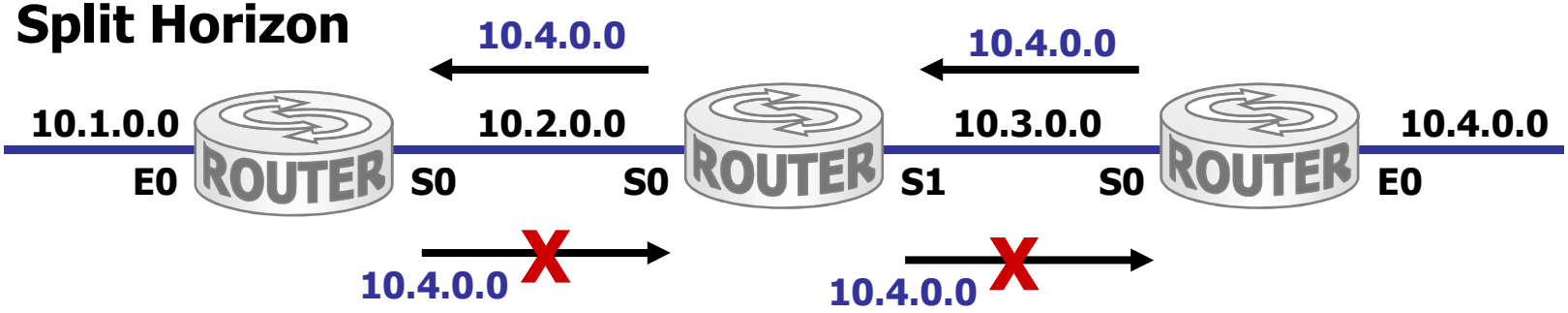
| | | |
|---|---|---|
| 10.3.0.0 | S0 | 0 |
| 10.4.0.0 | S0 | 2 |
| 10.2.0.0 | S0 | 1 |
| 10.1.0.0 | S0 | 2 |

◆ 결국 **Hop Count**가 **Maximum=16** 까지 증가 되어서야 **10.4.0.0**의 **Network**이 도달 할 수 없음을 모든 **Router**가 인지하게 된다.

◆**Hop Count**가 **Maximum**까지 증가되면 **Routing Table**에서 해당 경로를 제거한다.

# Routing Loop 문제 해결

## - Split Horizon




**Routing Table**

| | | |
|---|---|---|
| 10.1.0.0 | E0 | 0 |
| 10.2.0.0 | S0 | 0 |
| 10.3.0.0 | S0 | 1 |
| 10.4.0.0 | S0 | 2 |

**Routing Table**

| | | |
|---|---|---|
| 10.2.0.0 | S0 | 0 |
| 10.3.0.0 | S1 | 0 |
| 10.4.0.0 | S1 | 1 |
| 10.1.0.0 | S0 | 1 |

**Routing Table**

| | | |
|---|---|---|
| 10.3.0.0 | S0 | 0 |
| 10.4.0.0 | E0 | 0 |
| 10.2.0.0 | S0 | 1 |
| 10.1.0.0 | S0 | 2 |

특정 **interface**에서 받아온 **Route** 정보는 차후에 그 **interface**를 통해 다시 전달되지 않는다.

# Routing Loop 문제 해결

## - Route Poisoning


**Routing Table**

| | | |
|---|---|---|
| 10.1.0.0 | E0 | 0 |
| 10.2.0.0 | S0 | 0 |
| 10.3.0.0 | S0 | 1 |
| 10.4.0.0 | S0 | 16 |

**Routing Table**

| | | |
|---|---|---|
| 10.2.0.0 | S0 | 0 |
| 10.3.0.0 | S1 | 0 |
| 10.4.0.0 | S1 | 16 |
| 10.1.0.0 | S0 | 1 |

**Routing Table**

| | | |
|---|---|---|
| 10.3.0.0 | S0 | 0 |
| 10.4.0.0 | E0 | Down |
| 10.2.0.0 | S0 | 1 |
| 10.1.0.0 | S0 | 2 |

최대 홉 가운트 보다 대개 **1**만큼 크게 설정해서 적용한다.

# Routing Loop 문제 해결

## - Poison Reverse


Poison Reverse

**Routing Table**

| | | |
|---|---|---|
| 10.1.0.0 | E0 | 0 |
| 10.2.0.0 | S0 | 0 |
| 10.3.0.0 | S0 | 1 |
| 10.4.0.0 | S0 | 16 |

**Routing Table**

| | | |
|---|---|---|
| 10.2.0.0 | S0 | 0 |
| 10.3.0.0 | S1 | 0 |
| 10.4.0.0 | S1 | Possibly Down |
| 10.1.0.0 | S0 | 1 |

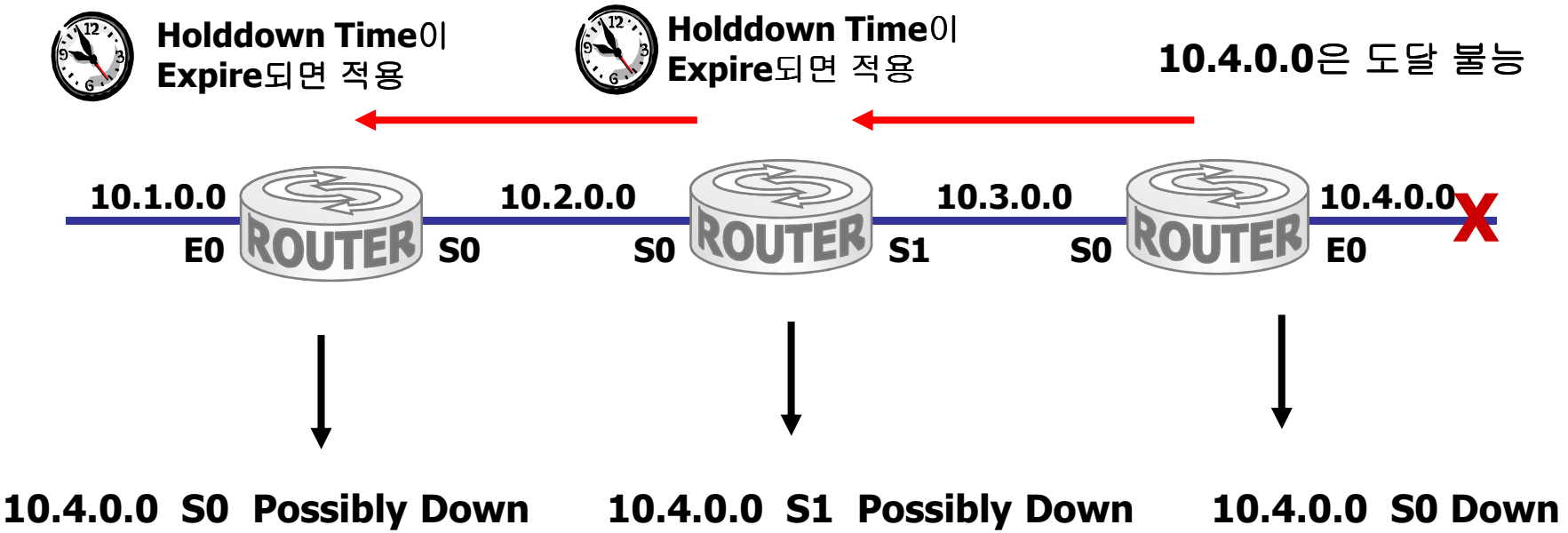
**Routing Table**

| | | |
|---|---|---|
| 10.3.0.0 | S0 | 0 |
| 10.4.0.0 | E0 | infinity |
| 10.2.0.0 | S0 | 1 |
| 10.1.0.0 | S0 | 2 |

해당 경로의 홉 수를 **16**으로 정정하여 전체 네트워크에 전송한다. 그러나 포이즌 리버스는 패킷 수가 증가된다는 단점이 있다.

# Routing Loop 문제 해결

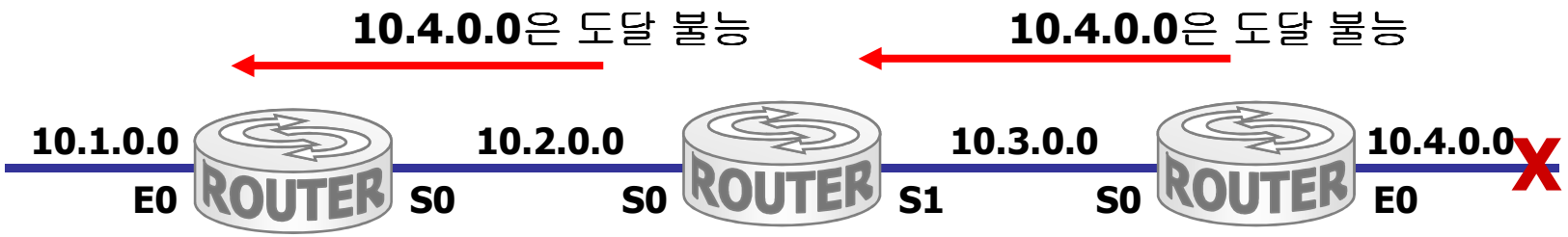
## - Hold down Timer




**Router**가 특정 **Link**의 **Fail**을 전달 받은 후에 해당 경로를 **Routing Table**에서 바로 제거하지 않고 특정 시간 동안 그 정보의 사실을 확인 하기 위해 기다린다. 이는 **Topology**의 변화 정보를 검증하는 용도이다.

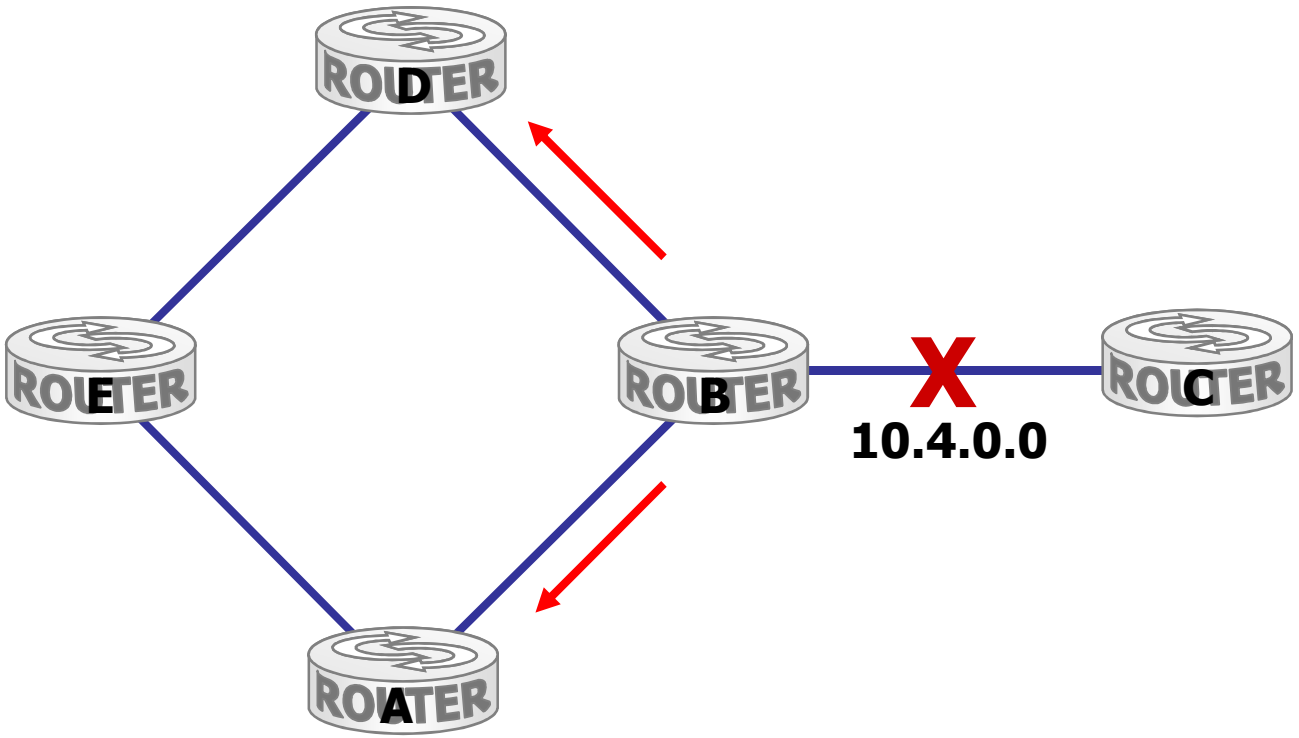
# Routing Loop 문제 해결

## - Triggered Update

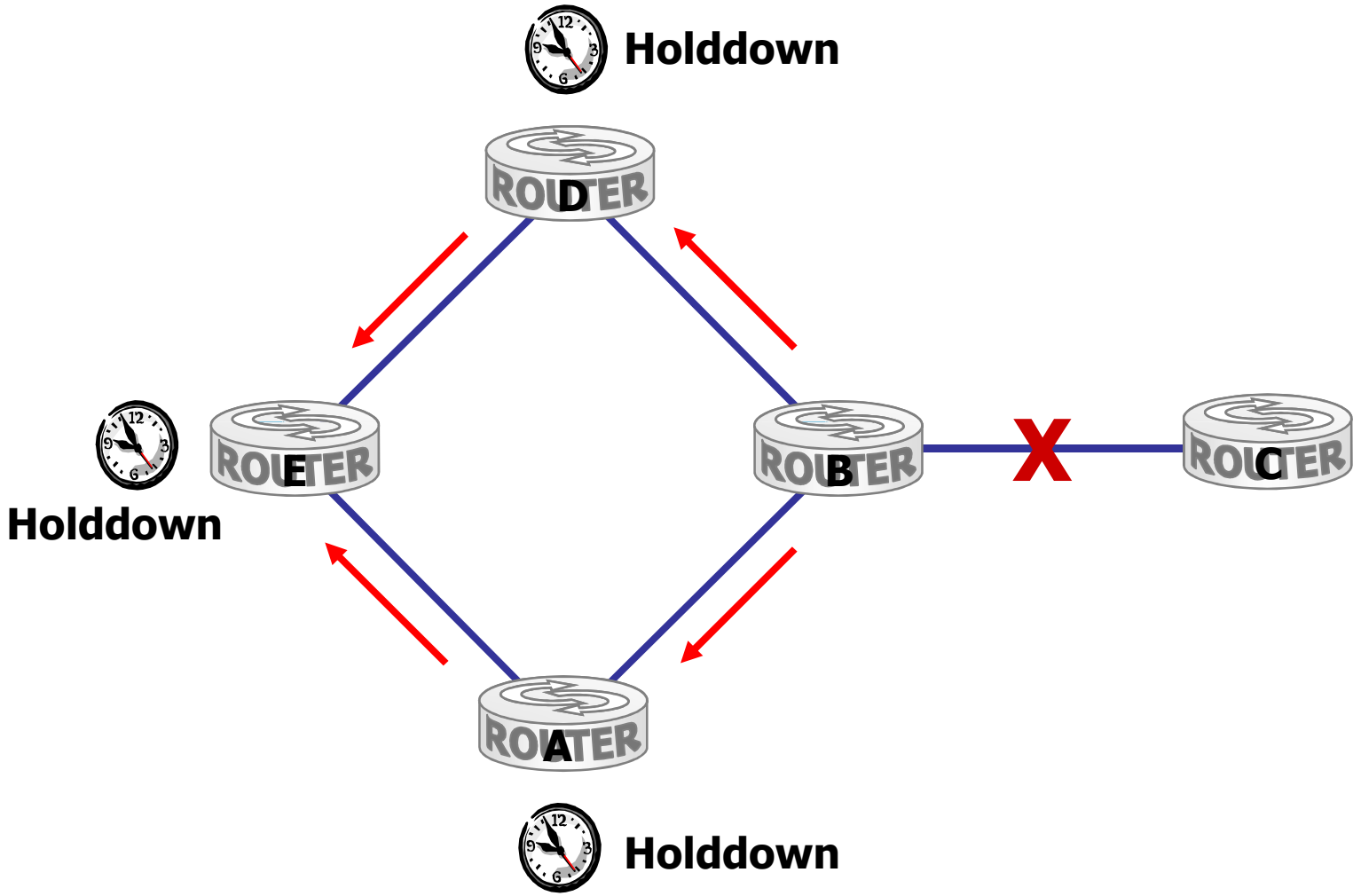



**Topology**의 변화를 즉시 이웃한 **Router**에게 알려준다.

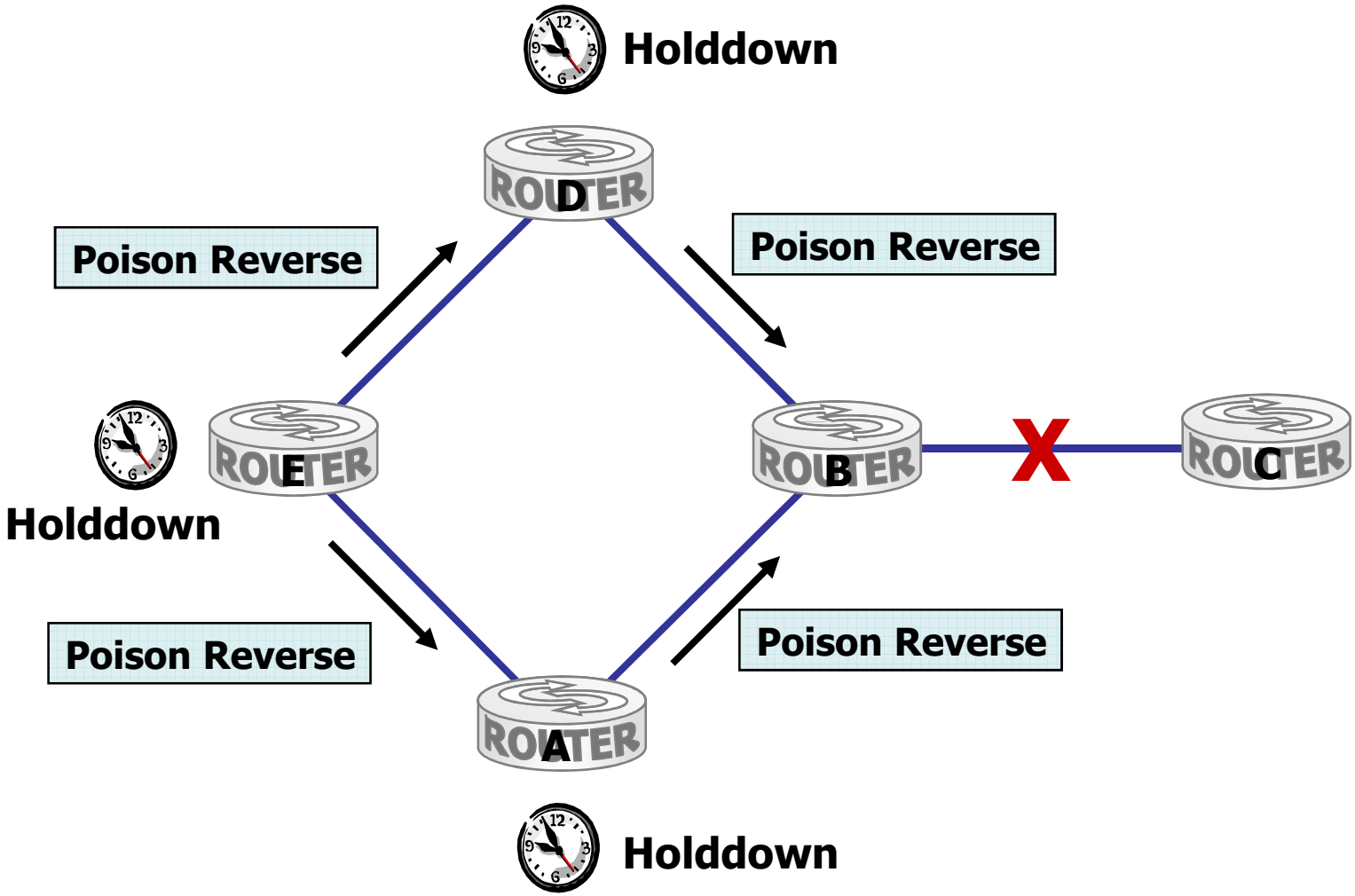
# Distance Vector의 Operation



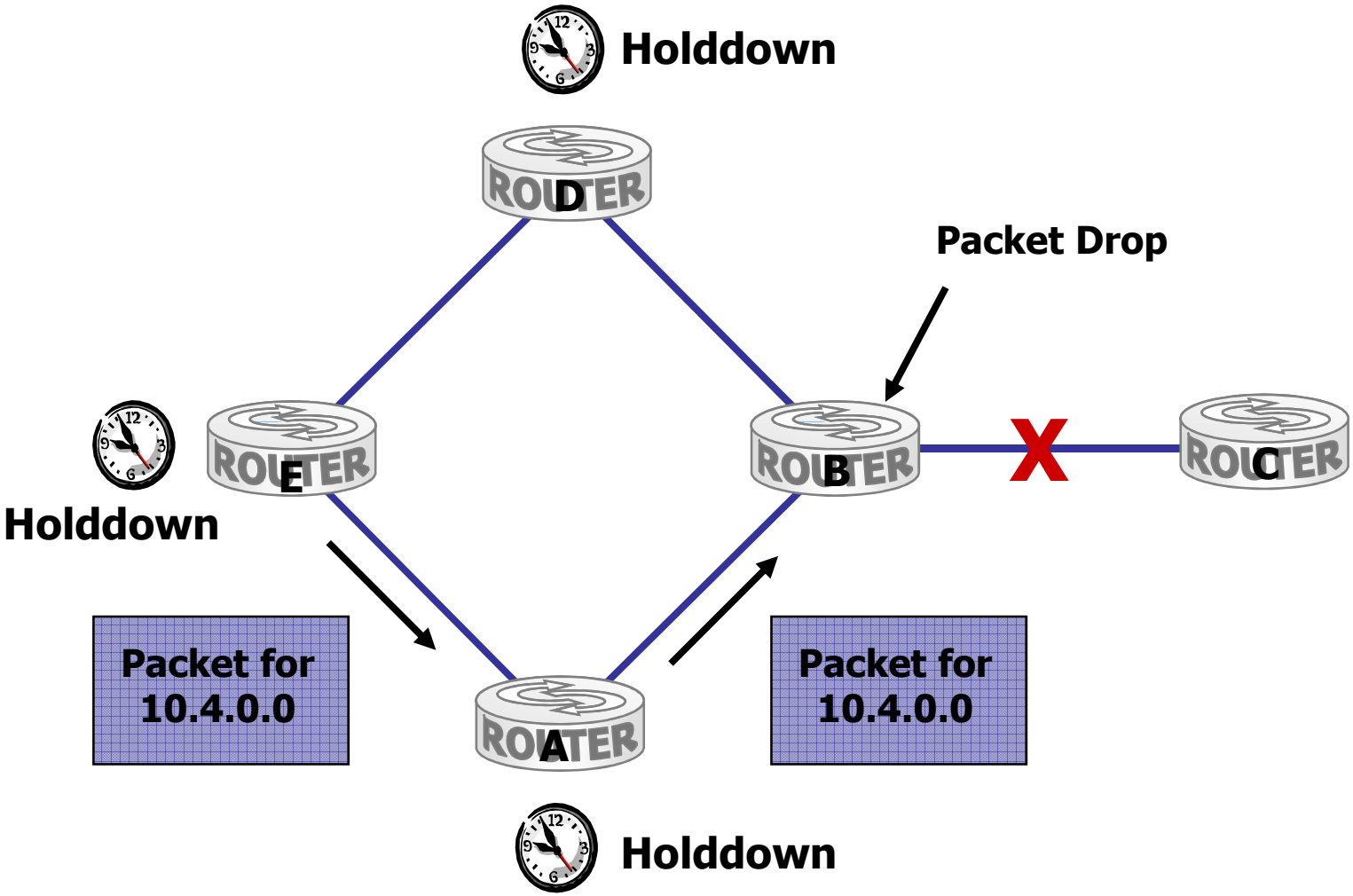
# Distance Vector의 Operation



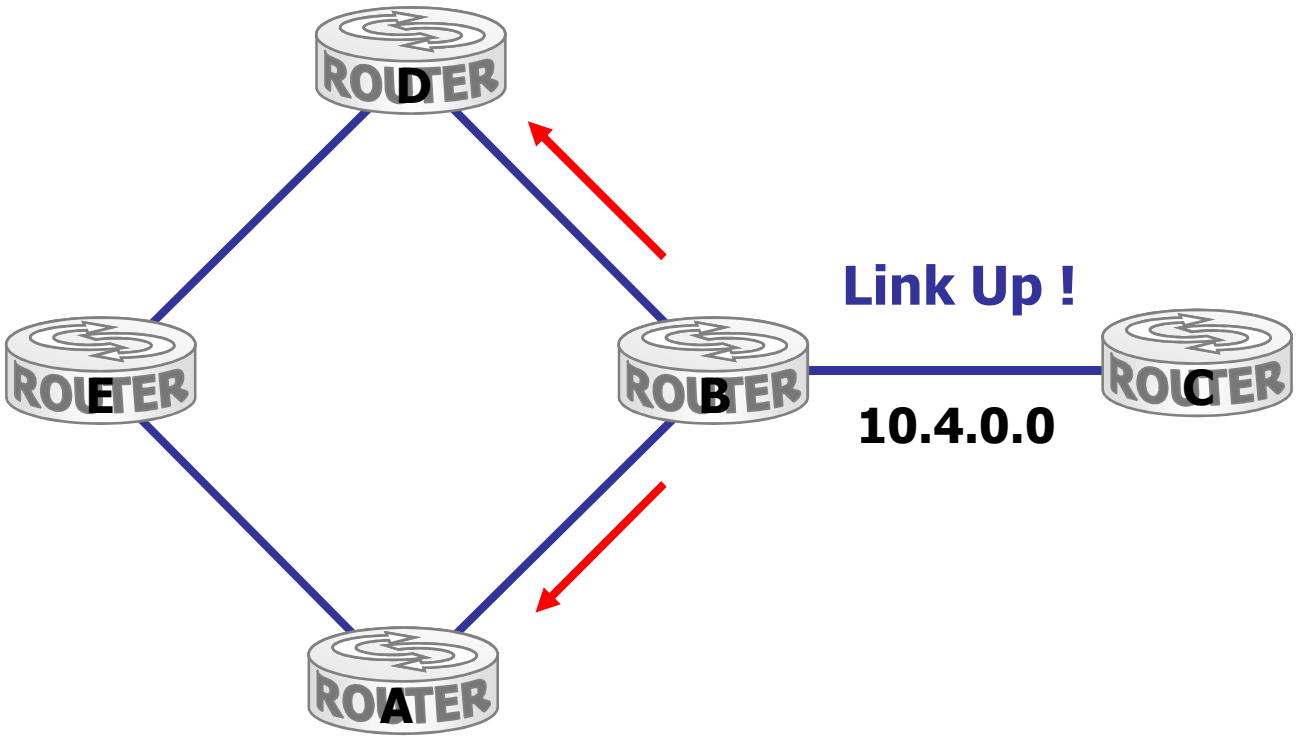
# Distance Vector의 Operation



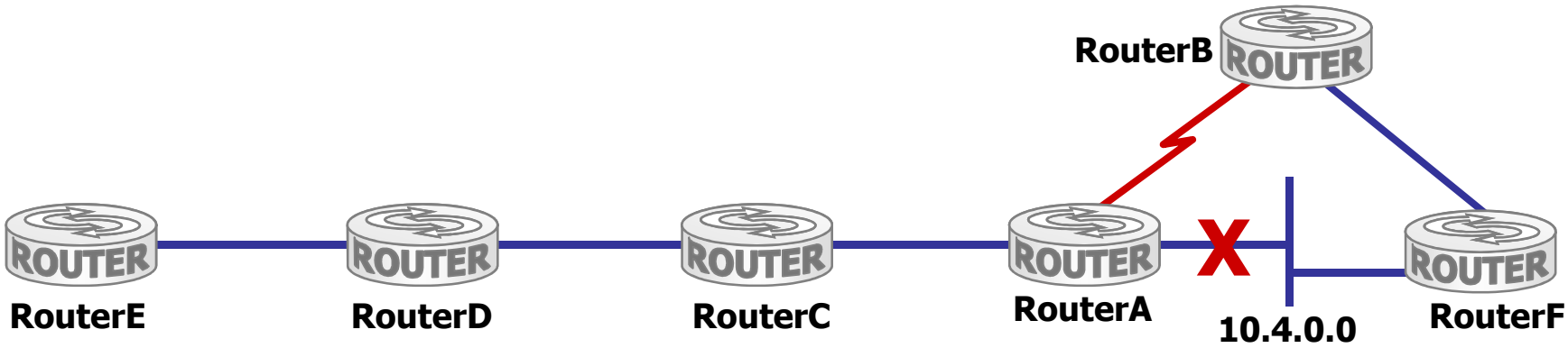
# Distance Vector의 Operation



# Distance Vector의 Operation



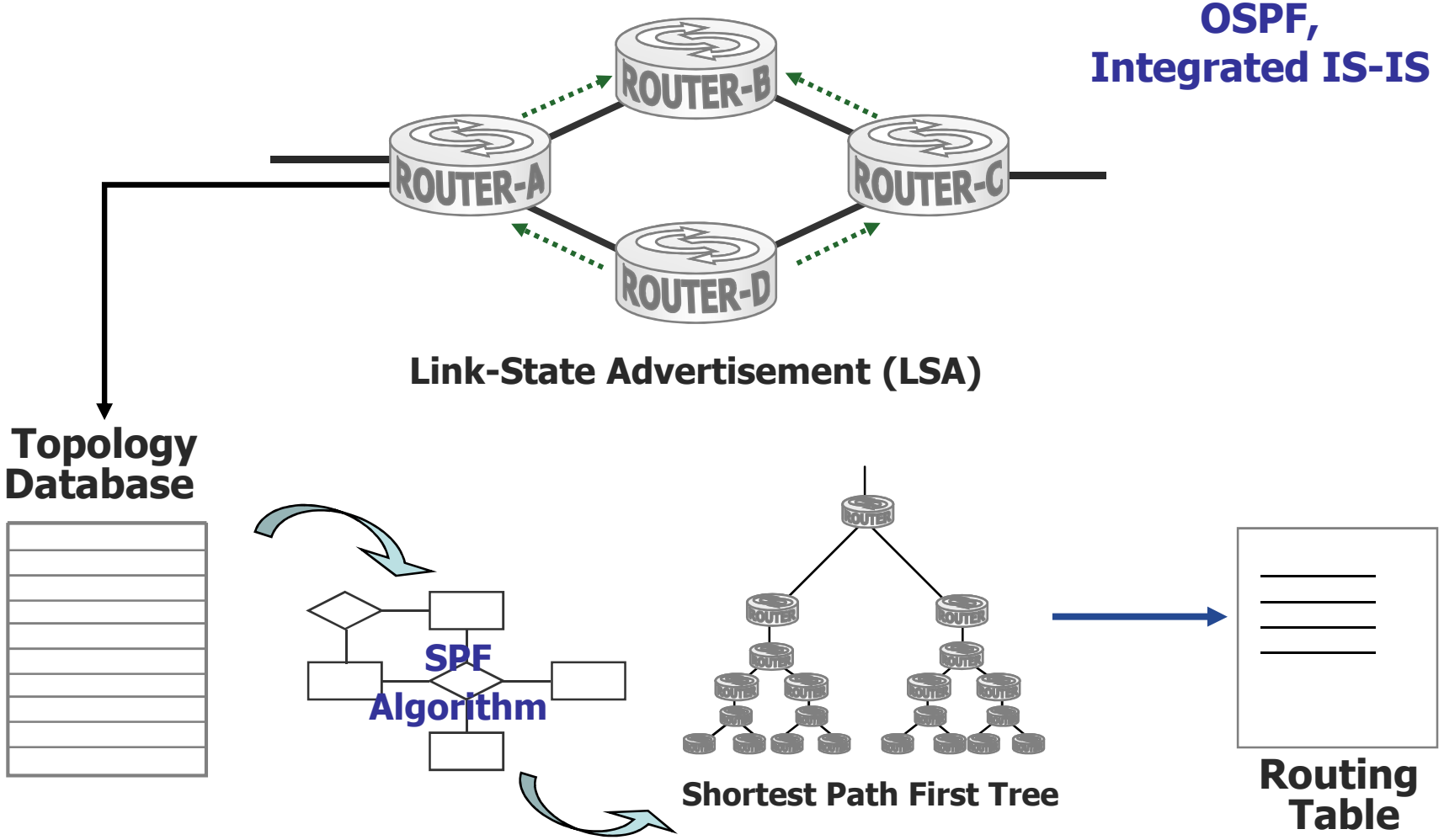
# Distance Vector의 Operation



# Link-State & Hybrid Routing

- **Link-State Routing Protocol** 개요
- **Link-State**의 **SPF** 알고리즘
- **Link-State** 의 계층적 구조
- **Hybrid Routing Protocol**

# Link-State Routing Protocol



# Link-State Routing Protocol

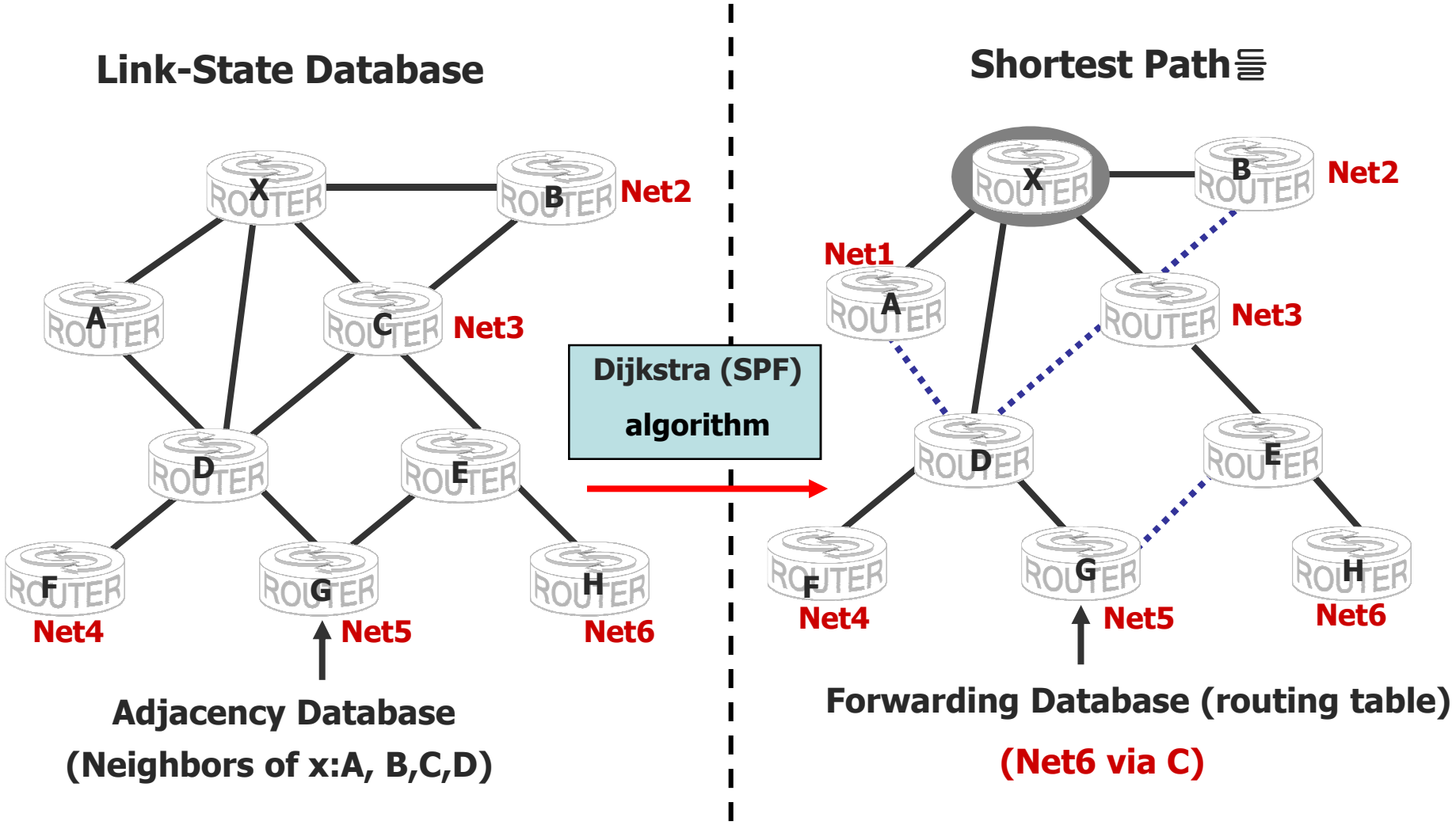
## 장점

- **Fast Convergence**
  **Topology** 변화에 대해 빠른 반응을 수행한다.
- **Routing Loop**
  **Topology**를 이해하므로 **SPF** 알고리즘에서 **Routing Loop**을 방지한다.
- 계층적 **Design**에 따라 **Network** 확장성이 보장된다.

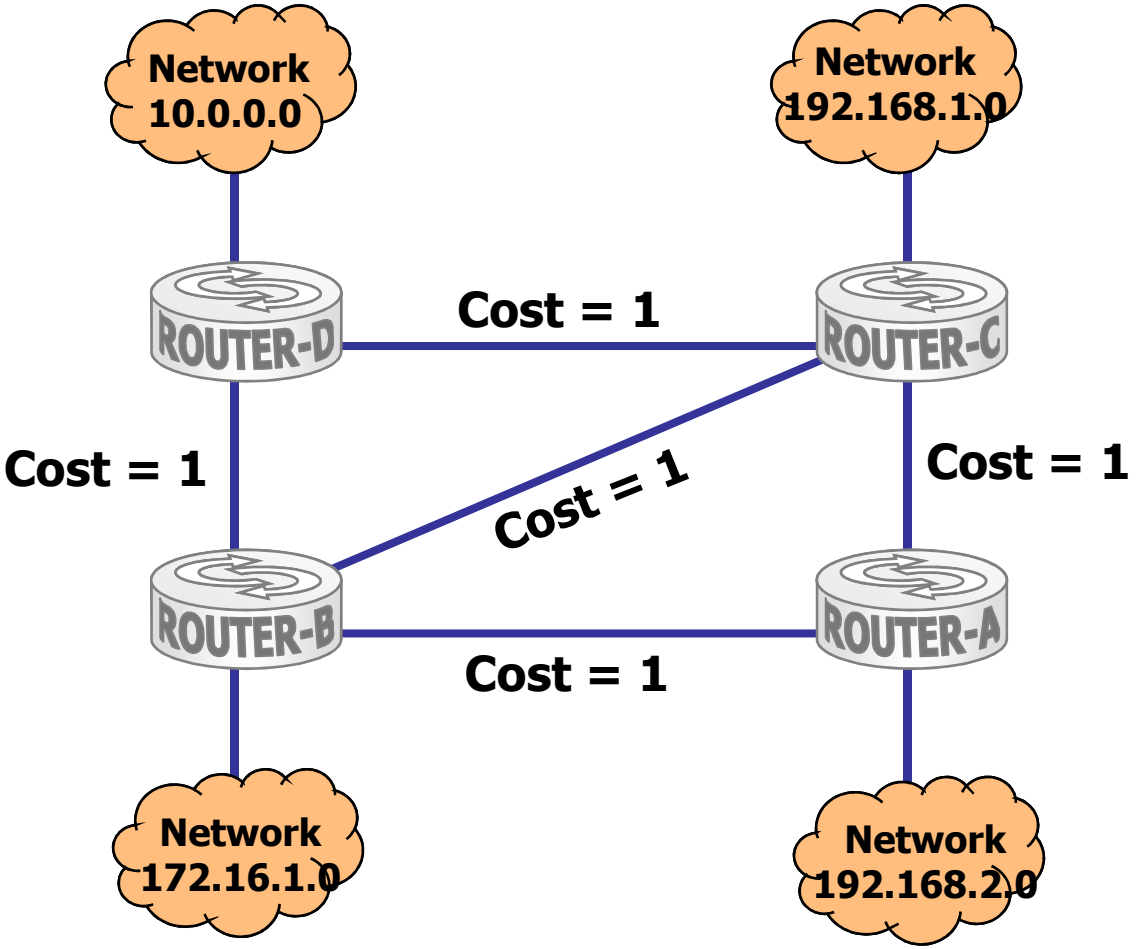
## 단점

- **Router**의 내부 **Resource** 소모가 많다.
  **CPU** → 잦은 **SPF** 알고리즘 수행
  **Memory** → **Network Topology** 정보 관리
- 반드시 계층적 **Design Rule**을 따라야 한다.
- 경우에 따라서는 많은 **Tuning Option**을 이해해야 한다.

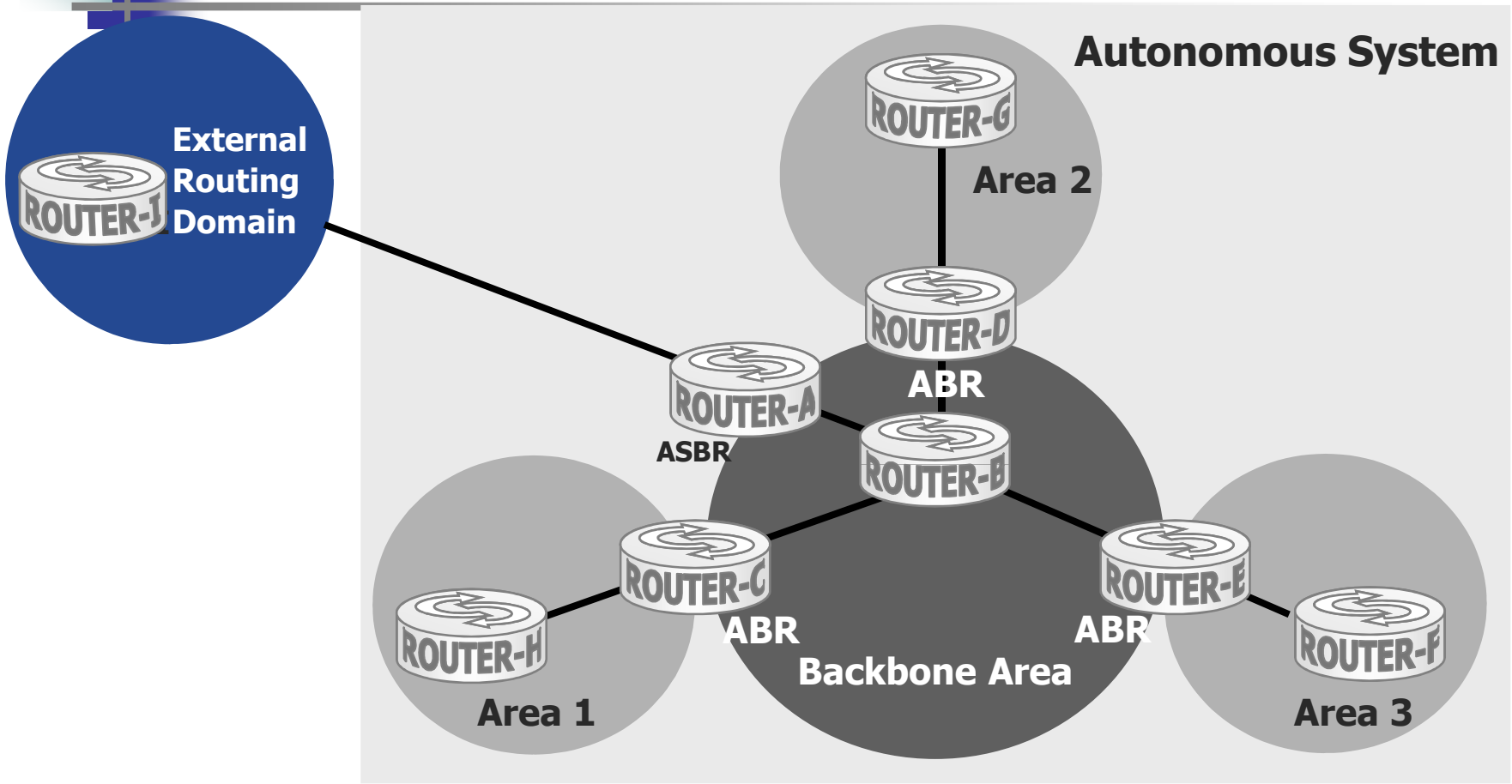
# Link-State의 SPF 알고리즘



# Link-State의 SPF 알고리즘



# Link-State의 계층적 구조




- 효과적인 **Route Summarization**을 통해 **Routing Table Size** 절감
- **Area**안에서의 **Topology** 변화와 관련된 **Traffic**을 지역적으로 제한한다.
- **Router**의 **Processor**와 **Memory** 자원 절감
- **Routing Update Traffic**줄임

# Hybrid Routing Protocol

**Balanced Hybrid Routing**

**=**

**Distance Vector**의

**Bandwidth, Delay**를

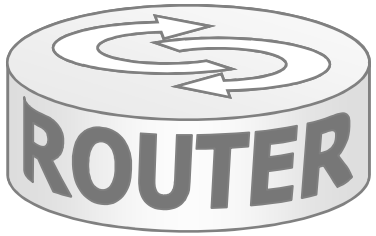
기준으로 경로 선택

**+**

**Link State**처럼 **Topology** 변화가

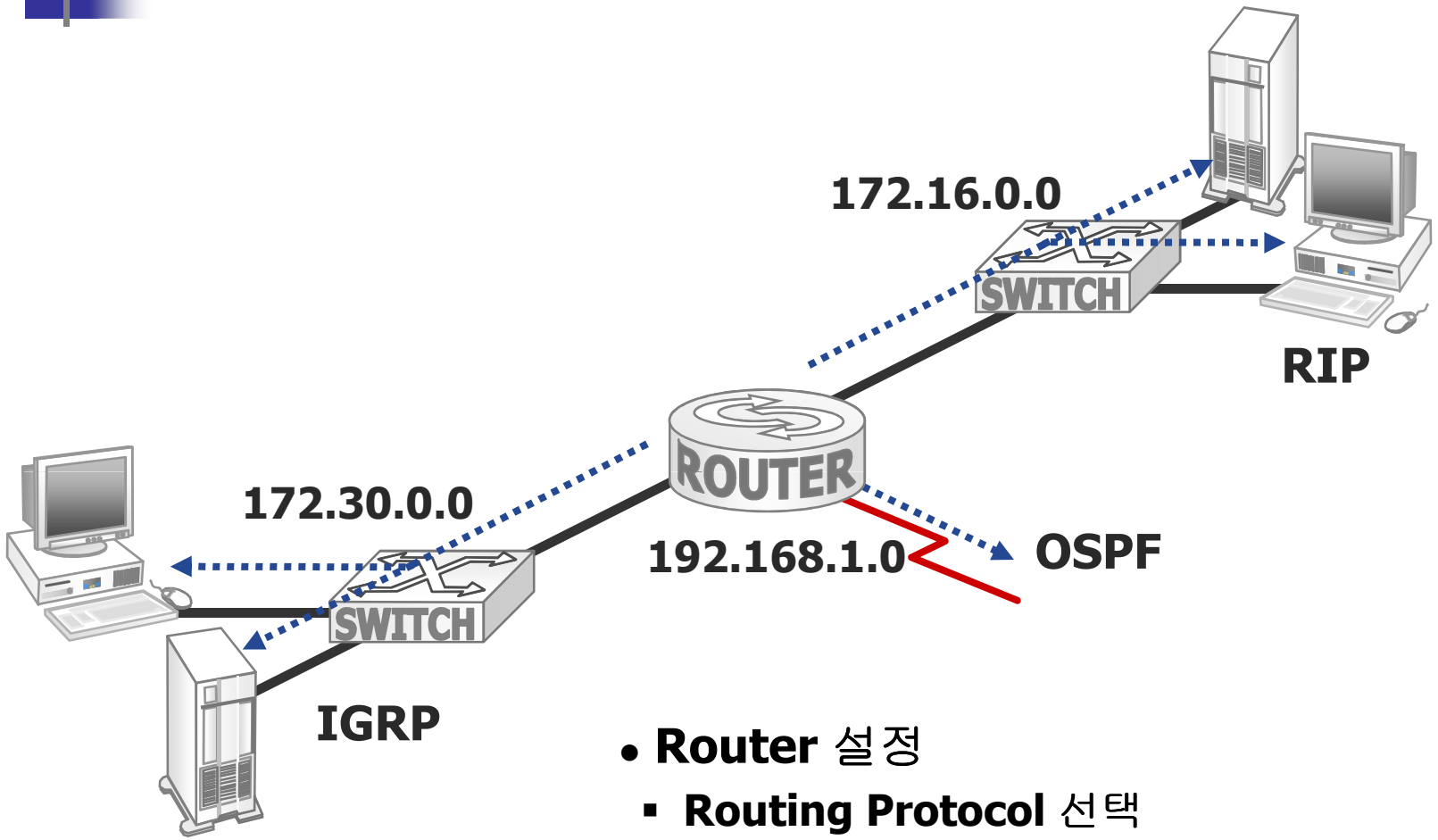
있을 때 빠른 라우팅 반영

**EIGRP**
**BGP4**




- **Distance-Vector**와 **Link-State** 라우팅 속성을 공유

# Dynamic Routing 설정하기




- **Router** 설정
  - **Routing Protocol** 선택
  - **Network**와 **Interface** 지정
  - **Summarization**과 같은 기타 옵션을 지정

# Dynamic Routing 설정하기

**Router(config)#router *protocol* [*keyword*]**

◆ **IP Routing Protocol**을 정의

**Router(config-router)#network *network-number [options]***

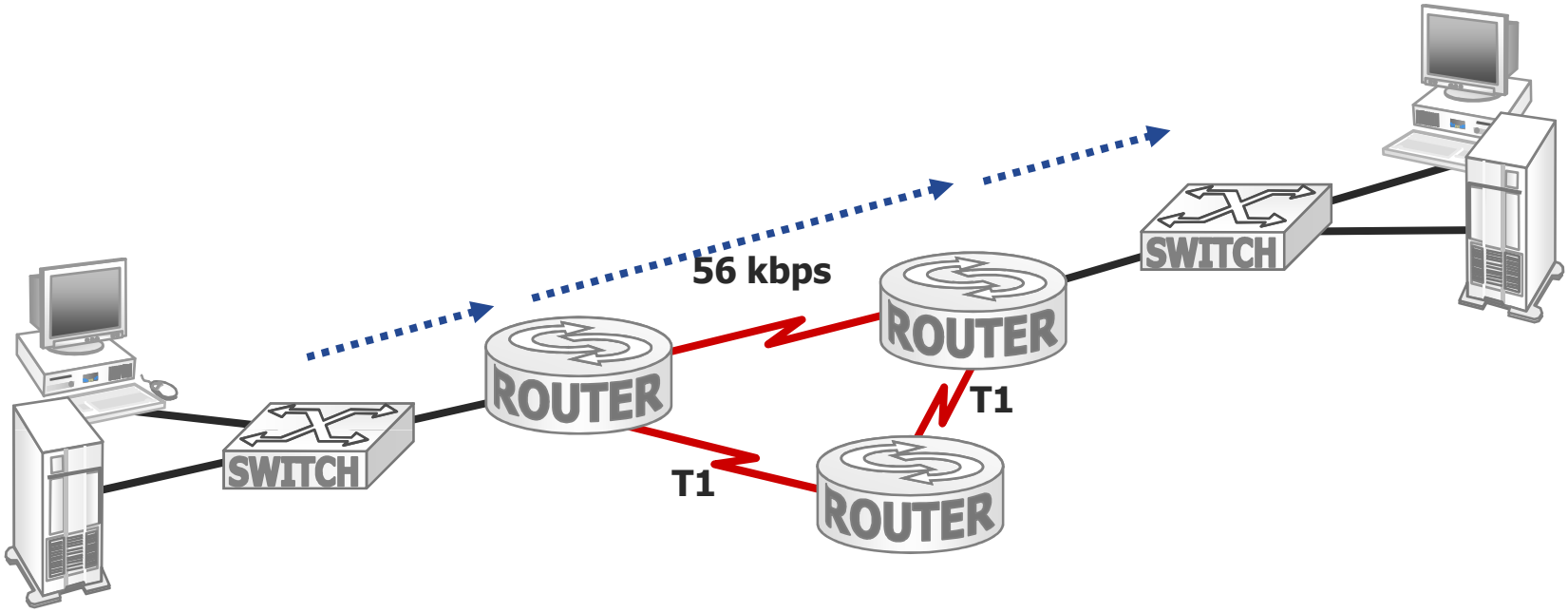
◆ 각각의 **IP Routing Process**에 반드시 설정해야 하는 명령어

◆ 자신이 가진 **Network**를 알리고, 이 **Network**에서 파생된 **ip address**가 할당된 **Interface**로 **Routing** 정보를 전송

# **RIP** 구성하기

- **RIP**의 개요
- **Dynamic Routing** 설정하기
- **RIP** 설정하기
- **RIP**의 구성 검증

# RIP의 개요




- **Hop Count**를 **Metric**으로 사용하여 최적 경로 선택
- 매 **30**초마다 **Routing Update**를 내보냄
- 최대 **6**개의 **Cost Equal Path** 지원

# RIP 설정하기

- **RIP routing process**시작

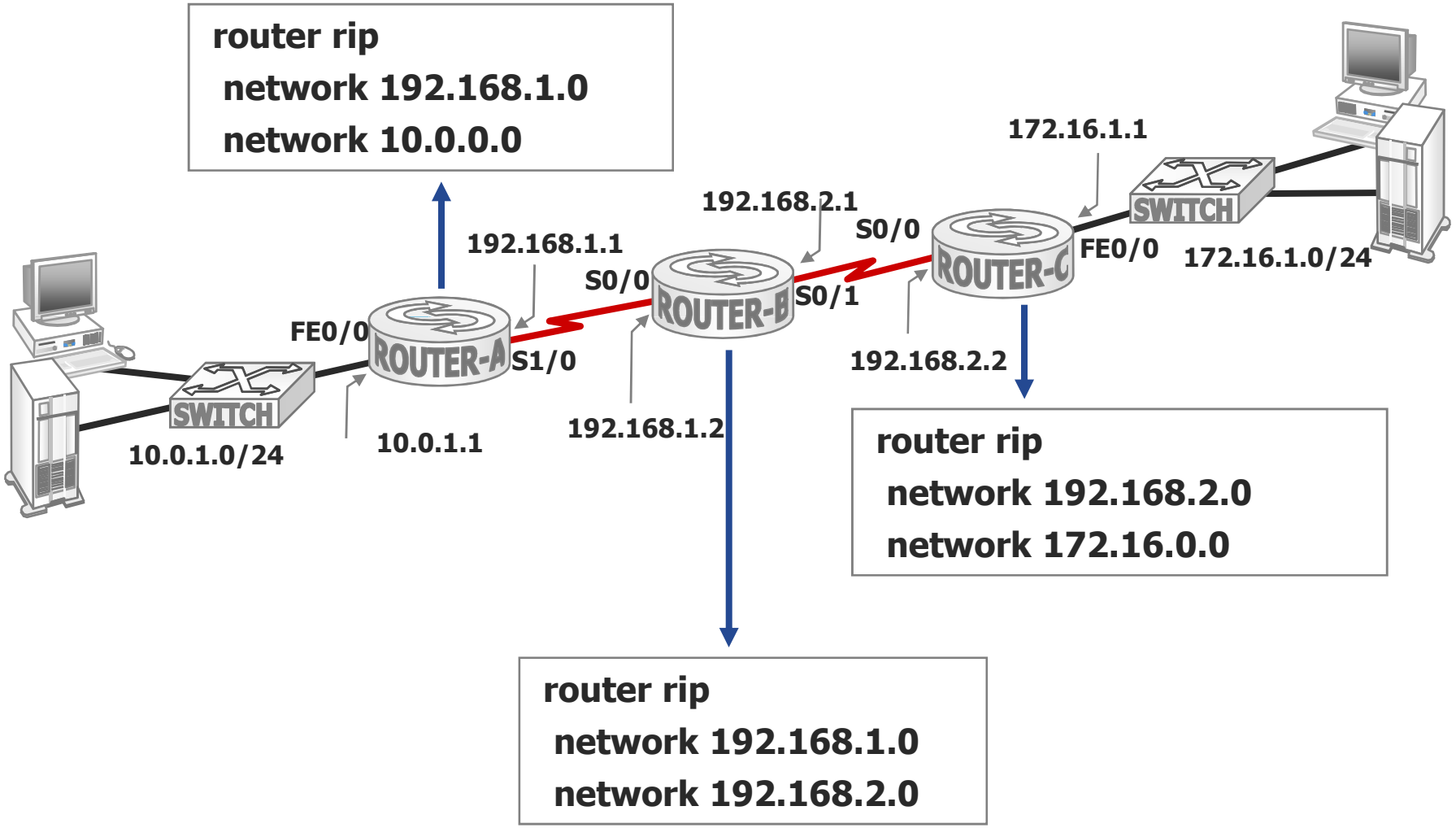
```
Router(config)#router rip
```

```
Router(config-router)#network network-number
```

◆ 참여할 연결된 네트워크들을 선택

◆ **Network Number**는 반드시 **Major Classful Network Number**로 설정해야 한다**.**

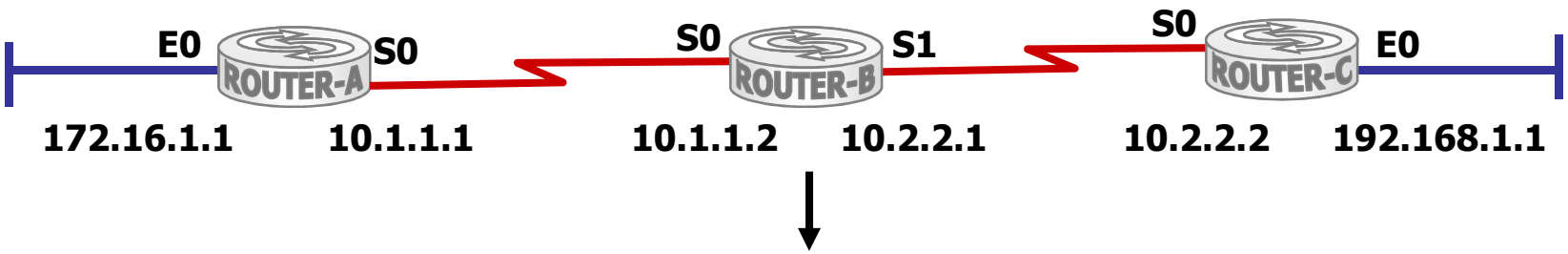
## RIP 설정 예제


router rip
network 192.168.1.0
network 10.0.0.0
192.168.1.1
FE0/0
S0/0
S1/0
ROUTER-A
SWITCH
10.0.1.0/24
10.0.1.1
192.168.2.1
S0/0
S0/1
ROUTER-B
192.168.1.2
172.16.1.1
FE0/0
ROUTER-C
SWITCH
172.16.1.0/24
192.168.2.2
router rip
network 192.168.2.0
network 172.16.0.0
router rip
network 192.168.1.0
network 192.168.2.0

# RIP의 구성 검증

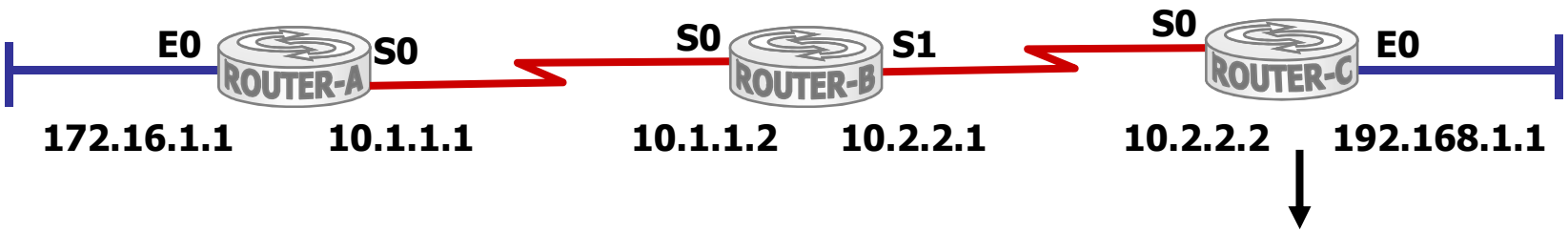

```
Router-A#show ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 5 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Redistributing: rip
  Default version control: send version 1, receive any version
    Interface          Send  Recv  Triggered RIP  Key-chain
    Ethernet0          1     1 2
    Serial0            1     1 2
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
    1722.16.0.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.1.1.2              120         00:00:18
  Distance: (default is 120)
```

# RIP의 구성 검증




```
Router-B#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
    10.0.0.0/24 is subnetted, 2 subnets
C    10.1.1.0/24 is directly connected, Serial0
C    10.2.2.0/24 is directly connected, Serial1
R    172.16.1.0/24  [120/1] via 10.1.1.1, 00:00:18, serial0
R    192.168.1.0/24  [120/1] via 10.2.2.2, 00:00:20, serial1

Router-B#clear ip route *
```

# RIP의 구성 검증

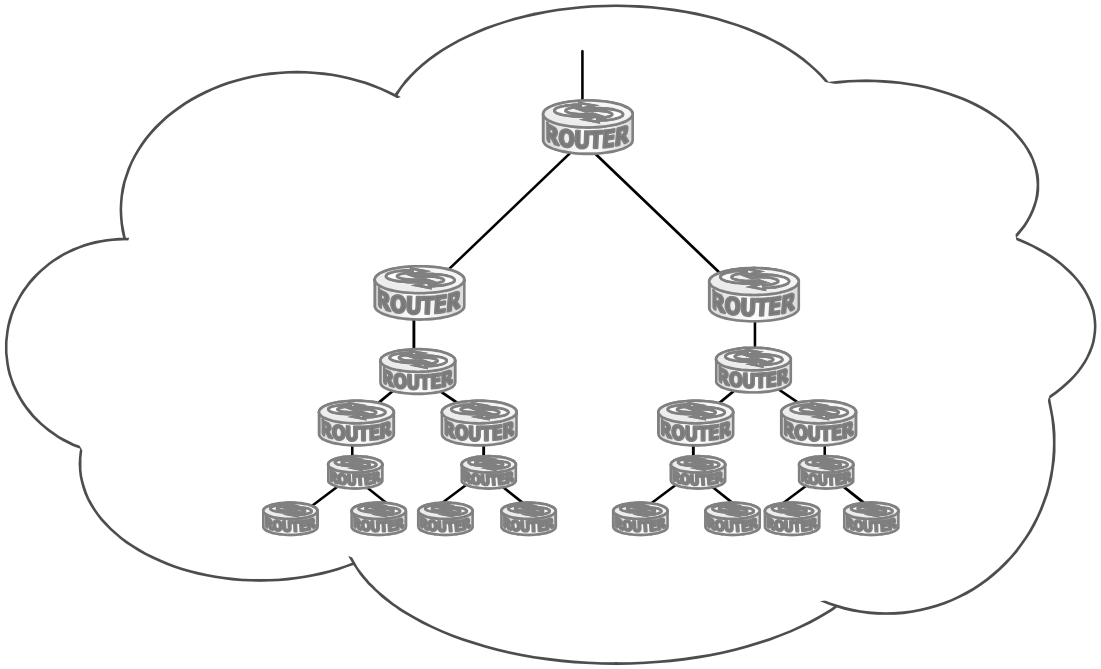



```
Router-C#debug ip rip
RIP protocol debugging is on
Router-C#
00:27:58: RIP: sending v1 update to 255.255.255.255 via Ethernet0 (192.168.1.1)
00:27:58: RIP: build update entries
00:27:58:         network 172.16.0.0 metric 3
00:27:58:         network 10.1.1.0 metric 2
00:27:58:         network 10.2.2.0 metric 1
00:27:58: RIP: sending v1 update to 255.255.255.255 via Serial0 (10.2.2.2)
00:27:58: RIP: build update entries
00:27:58:         network 192.168.1.0 metric 1
00:28:00: RIP: received v1 update from 10.2.2.2 on Serial0
00:28:00:        10.0.0.0 in 1 hops
00:28:00:        172.16.0.0 in 2 hops
.
.
<중간 생략>
```

# **IGRP** 구성하기

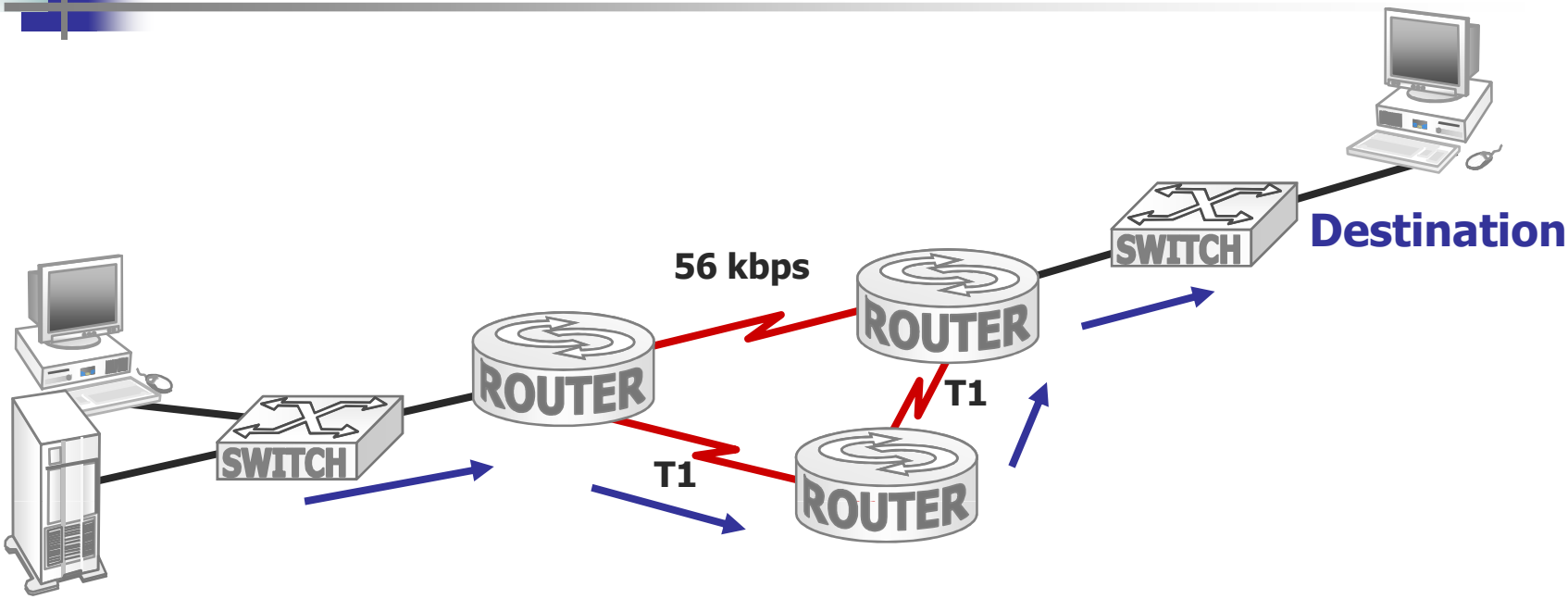
- **IGRP**의 개요
- **IGRP**의 **Metric** 소개
- **IGRP**의 다중 경로 지원
- **IGRP** 설정하기
- **IGRP** 구성 검증

# IGRP의 개요



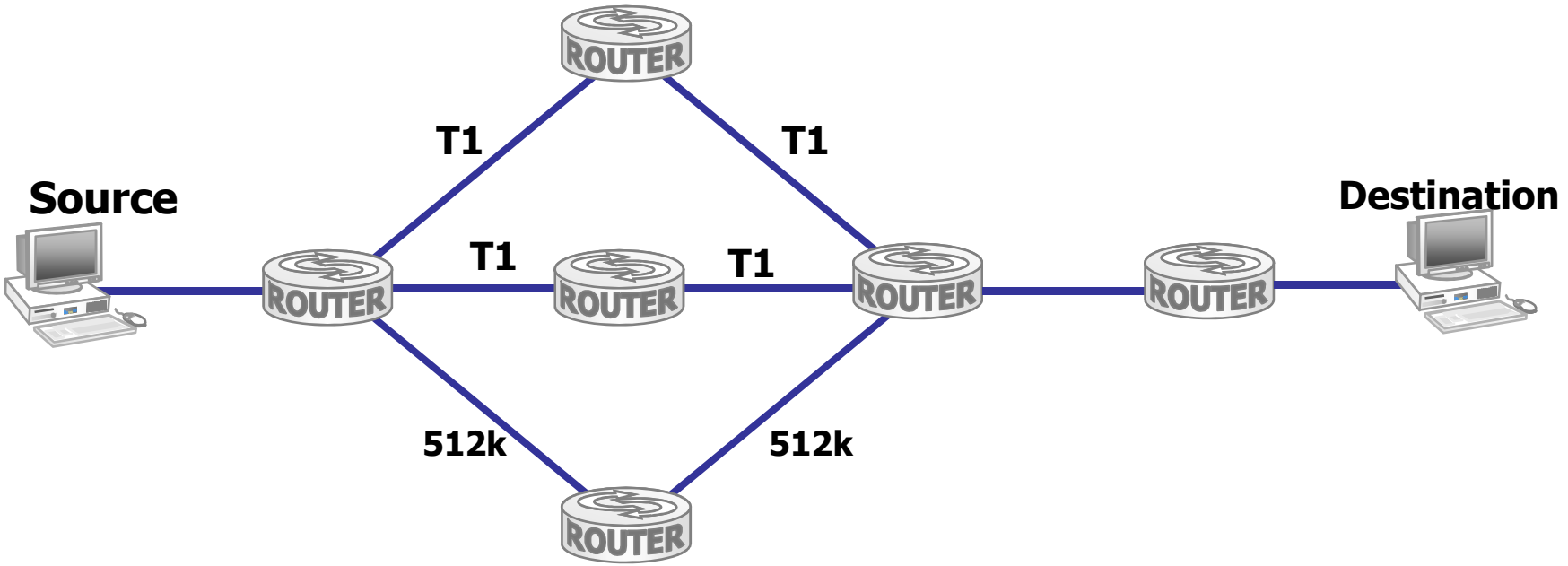

- **RIP**보다 대형 **Network** 지원을 위해 개발
- 다양한 **Metric** 속성 지원
- **Multiple-path** 지원
- **Cisco Router**에서만 동작

# IGRP의 Metric 소개




- **Bandwidth**
- **Delay**
- **Reliability**
- **Loading**
- **MTU**

# IGRP의 다중 경로 지원




- 최대 **6**개까지의 다중 경로 지원 **(Default = 4**개**)**
- **Unequal Multiple-path**를 지원 할 수 있다**.**
- **Unequal Multiple-path**에 참여되는 경로는 **Source**보다 **Destination**에 가까워야 한다**.**

# IGRP 설정하기

**IGRP**를 **Routing Protocol**로 사용한다.

```
Router(config)#router igrp autonomous-system
```

**IGRP**가 수행되는 **Network** 구간 정의

```
Router(config-router)#network network-number
```

**IGRP**가 수행되는 **Network** 구간 정의

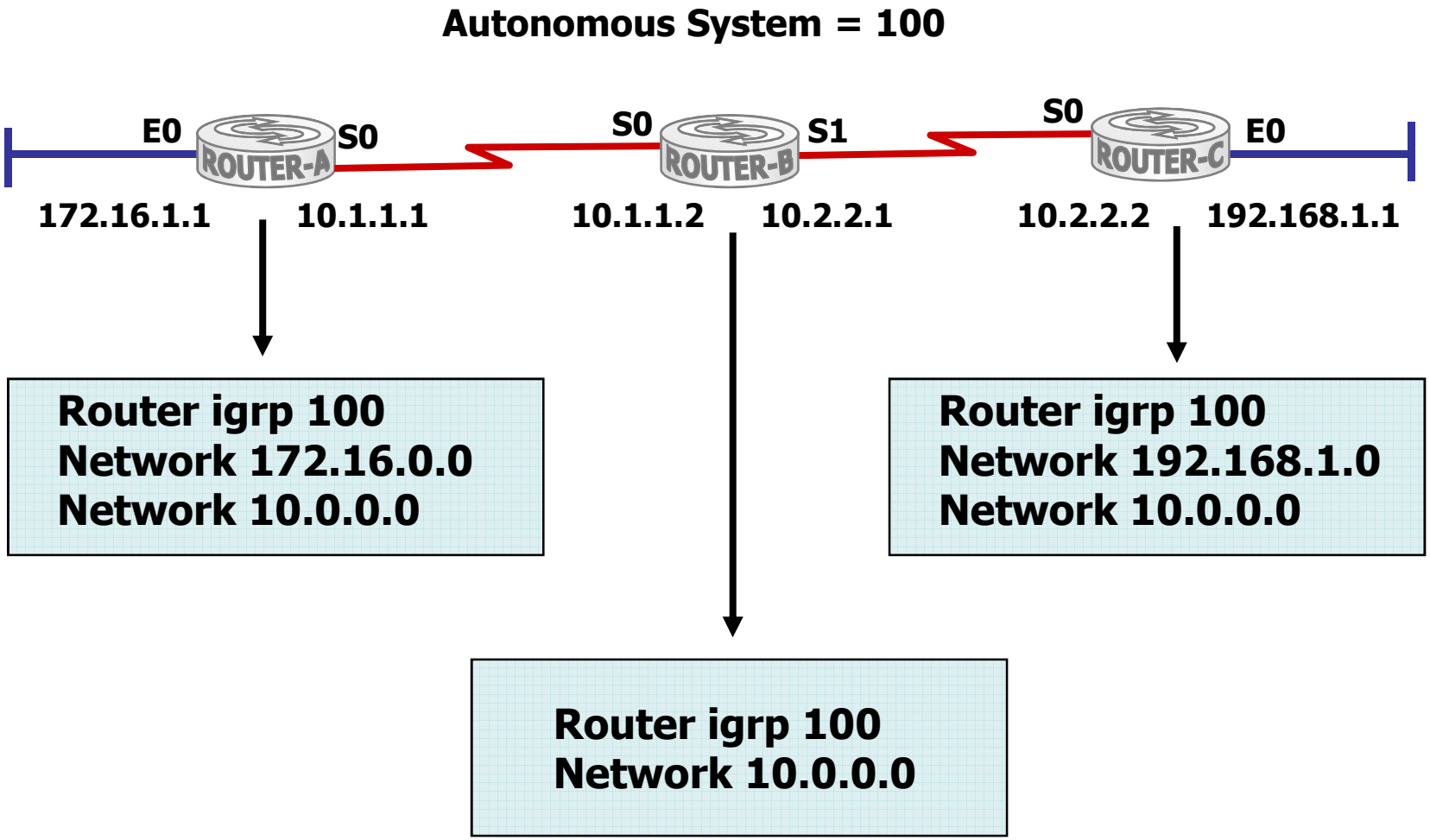
```
Router(config-router)#variance multiplier
```

**IGRP**가 수행되는 **Network** 구간 정의

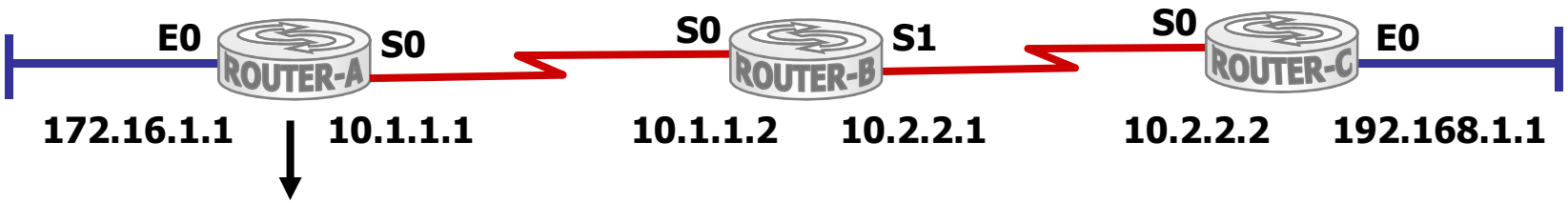
```
Router(config-router)#traffic-share { balanced | min }
```

# IGRP 설정하기

## IGRP 구성 예제

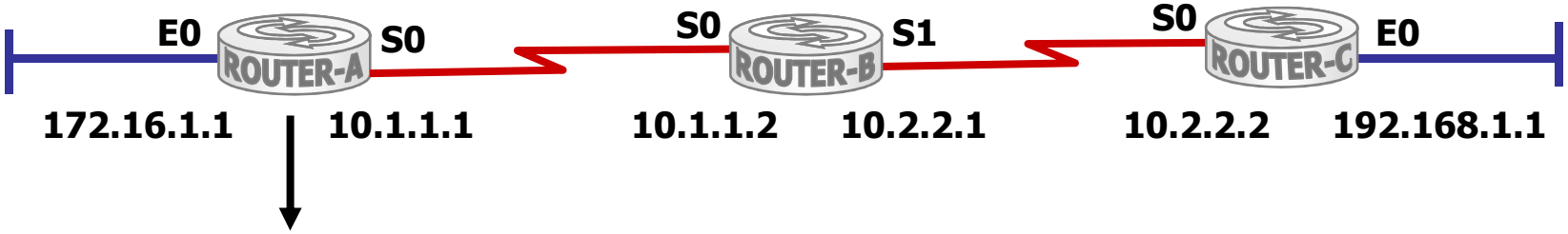


# IGRP 구성 검증



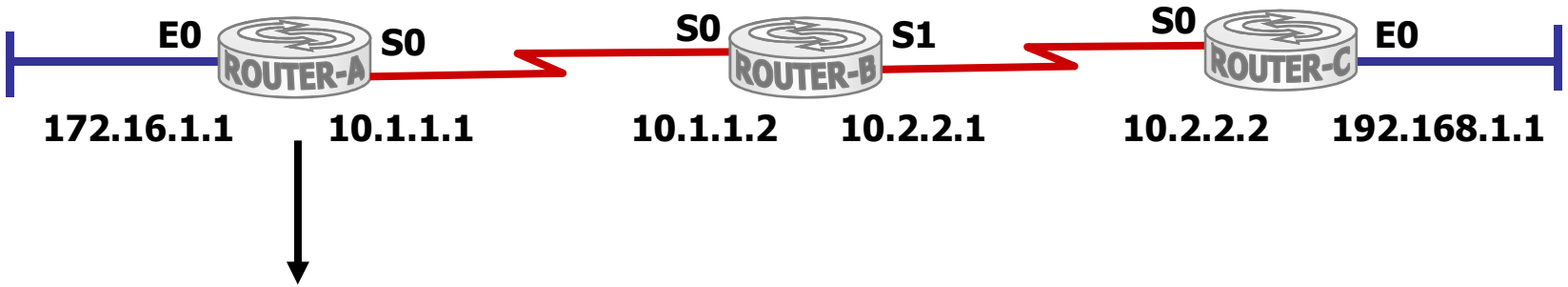

```
Router-A#show ip protocols
Routing Protocol is "igrp 100"
  Sending updates every 90 seconds, next due in 21 seconds
  Invalid after 270 seconds, hold down 280, flushed after 630
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  IGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  IGRP maximum hopcount 100
  IGRP maximum metric variance 1
  Redistributing: igrp 100
  Routing for Networks:
    10.0.0.0
    172.16.0.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.1.1.2            100      00:01:01
  Distance: (default is 100)
```

# IGRP 구성 검증



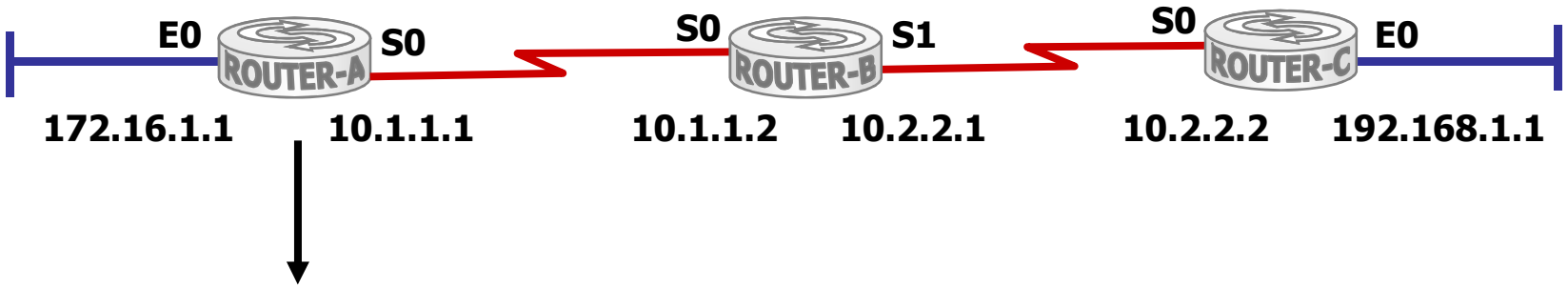

```
Router-A#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2,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

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.1.0 is directly connected, Ethernet0
     10.0.0.0/24 is subnetted, 2 subnets
I       10.2.2.0 [100/90956] via 10.1.1.2, 00:00:23, Serial0
C       10.1.1.0 is directly connected, Serial2
I    192.168.1.0/24 [100/91056] via 10.1.1.2, 00:00:23, Serial0
```

# IGRP 구성 검증



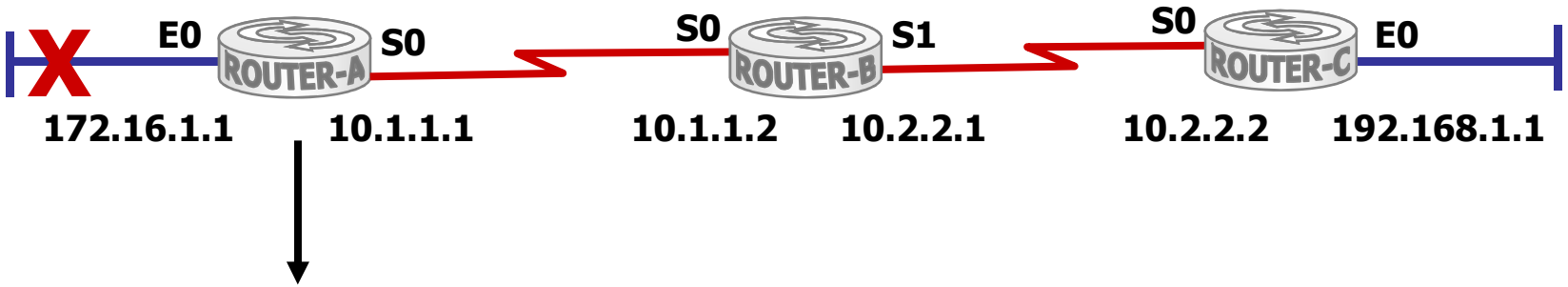

```
Router-A#debug ip igrp transactions
IGRP protocol debugging is on
Router-A#
00:21:06: IGRP: sending update to 255.255.255.255 via Ethernet0 (172.16.1.1)
00:21:06:          network 10.0.0.0, metric=88956
00:21:06:          network 192.168.1.0, metric=91056
00:21:07: IGRP: sending update to 255.255.255.255 via Serial0 (10.1.1.1)
00:21:07:          network 172.16.0.0, metric=1100
00:21:16: IGRP: received update from 10.1.1.2 on Serial0
00:21:16:          subnet 10.2.2.0, metric 90956 (neighbor 88956)
00:21:16:         network 192.168.1.0, metric 91056 (neighbor 89056)
```

# IGRP 구성 검증



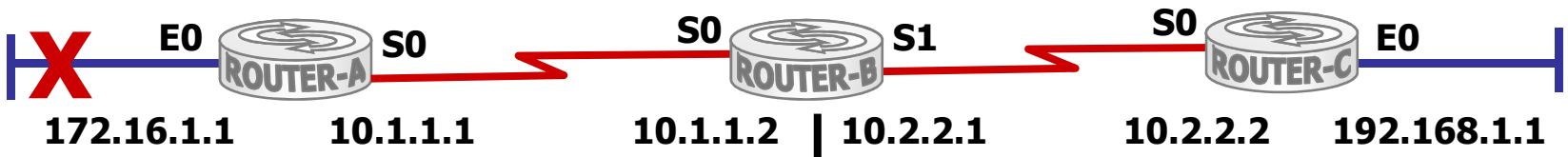

```
Router-A#debug ip igrp events
IGRP event debugging is on
Router-A#
00:23:44: IGRP: sending update to 255.255.255.255 via Ethernet0 (172.16.1.1)
00:23:44: IGRP: Update contains 0 interior, 2 system, and 0 exterior routes.
00:23:44: IGRP: Total routes in update: 2
00:23:44: IGRP: sending update to 255.255.255.255 via Serial0 (10.1.1.1)
00:23:45: IGRP: Update contains 0 interior, 1 system, and 0 exterior routes.
00:23:45: IGRP: Total routes in update: 1
00:23:48: IGRP: received update from 10.1.1.2 on Serial0
00:23:48: IGRP: Update contains 1 interior, 1 system, and 0 exterior routes.
00:23:48: IGRP: Total routes in update: 2
```

# IGRP 구성 검증



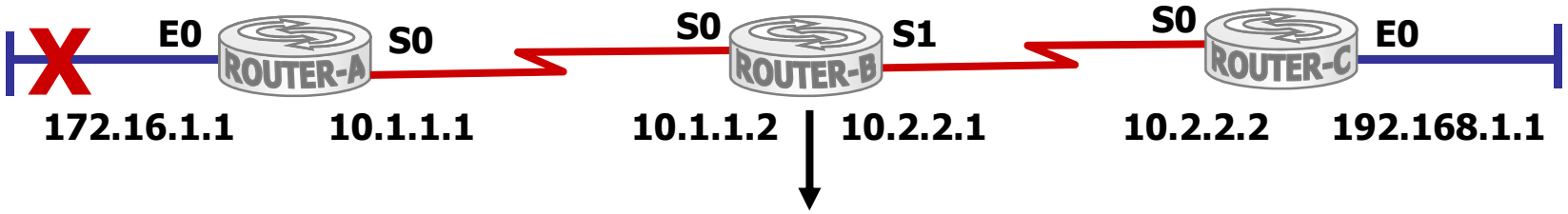

```
RouterA# debug ip igrp trans
00:31:15: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0, changed state to
down
00:31:15: IGRP : edition is now 3
00:31:15: IGRP : sending update to 255.255.255.255 via Serial0 (10.1.1.1)
00:31:15:          network 172.16.0.0, metric=4294967295
00:31:16: IGRP : Update contains 0 interior, 1 system, and 0 exterior routes.
00:31:16: IGRP : Total routes in update: 1
00:31:16: IGRP : broadcasting request on Serial0
00:31:16: IGRP : received update from 10.1.1.2 on Serial0
00:31:16:          subnet 10.2.2.0, metric 90956 (neighbor 88956)
00:31:16:          network 172.16.0.0, metric 4294967295 (inaccessible)
00:31:16:          network 192.168.1.0, metric 91056 (neighbor 89056)
00:31:16: IGRP : Update contains 1 interior, 2 system, and 0 exterior routes.
00:31:16: IGRP : Total routes in update: 3
```

# IGRP 구성 검증



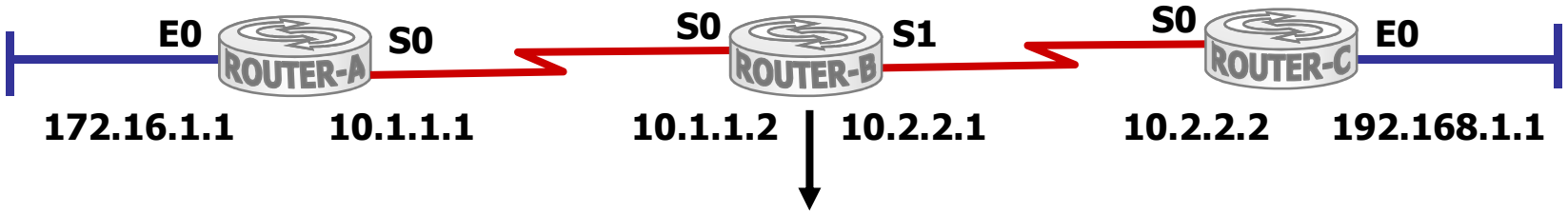

```
Router-B#debug ip igrp trans
IGRP protocol debugging is on
Router-B#
1d19h: IGRP: sending update to 255.255.255.255 via Serial0 (10.1.1.2)
1d19h:         subnet 10.2.2.0, metric=88956
1d19h:         network 192.168.1.0, metric=89056
1d19h: IGRP: sending update to 255.255.255.255 via Serial1 (10.2.2.1)
1d19h:         subnet 10.1.1.0, metric=88956
1d19h:         network 172.16.0.0, metric=89056
1d19h: IGRP: received update from 10.1.1.1 on Serial0
1d19h:         network 172.16.0.0, metric 4294967295 (inaccessible)
1d19h: IGRP: edition is now 10
1d19h: IGRP: sending update to 255.255.255.255 via Serial0 (10.1.1.2)
1d19h:         subnet 10.2.2.0, metric=88956
1d19h:         network 172.16.0.0, metric=4294967295
1d19h:         network 192.168.1.0, metric=89056
1d19h: IGRP: sending update to 255.255.255.255 via Serial1 (10.2.2.1)
1d19h:         subnet 10.1.1.0, metric=88956
1d19h:         network 172.16.0.0, metric=4294967295
```

# IGRP 구성 검증




```
Router-B#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
       <중간 생략>

Gateway of last resort is not set

I    172.16.0.0/16 is possibly down, routing via 10.1.1.1, Serial0
     10.0.0.0/24 is subnetted, 2 subnets
C       10.1.1.0 is directly connected, Serial0
C       10.2.2.0 is directly connected, Serial1
I    192.168.1.0/24 [100/89056] via 10.2.2.3, 00:00:14, Serial1

Router-B#ping 172.16.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Router-B#
```

# IGRP 구성 검증

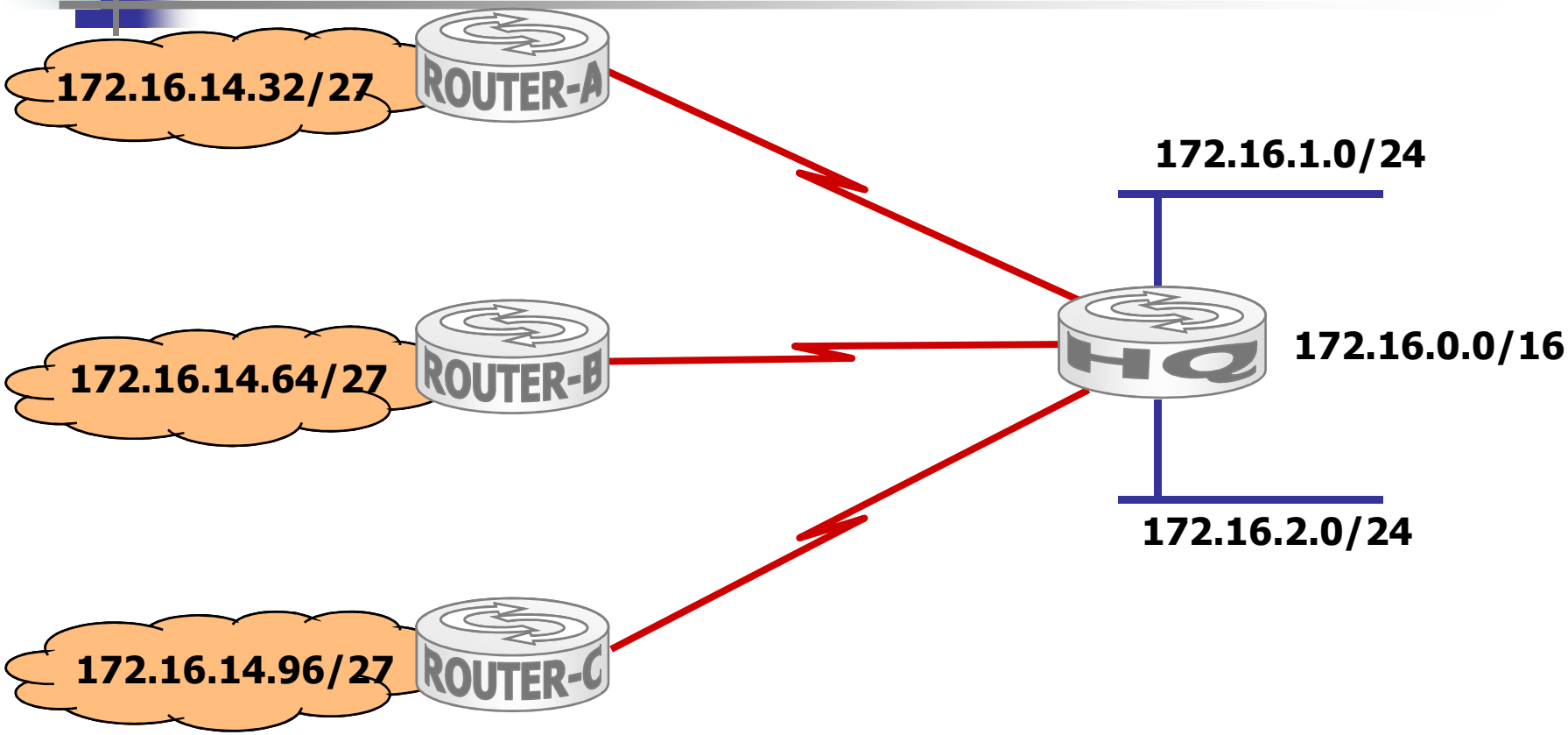



```
Router-B#
1d20h: IGRP: received update from 10.1.1.1 on Serial0
1d20h:      network 172.16.0.0, metric 89056 (neighbor 1100)
Router-B#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
          <중간 생략>
Gateway of last resort is not set

I    172.16.0.0/16 is possibly down, routing via 10.1.1.1, Serial0
     10.0.0.0/24 is subnetted, 2 subnets
C       10.1.1.0 is directly connected, Serial0
C       10.2.2.0 is directly connected, Serial1
I    192.168.1.0/24 [100/89056] via 10.2.2.3, 00:00:18, Serial1
Router-B#ping 172.16.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/38/48 ms
```

# VLSM과 Route Summarization

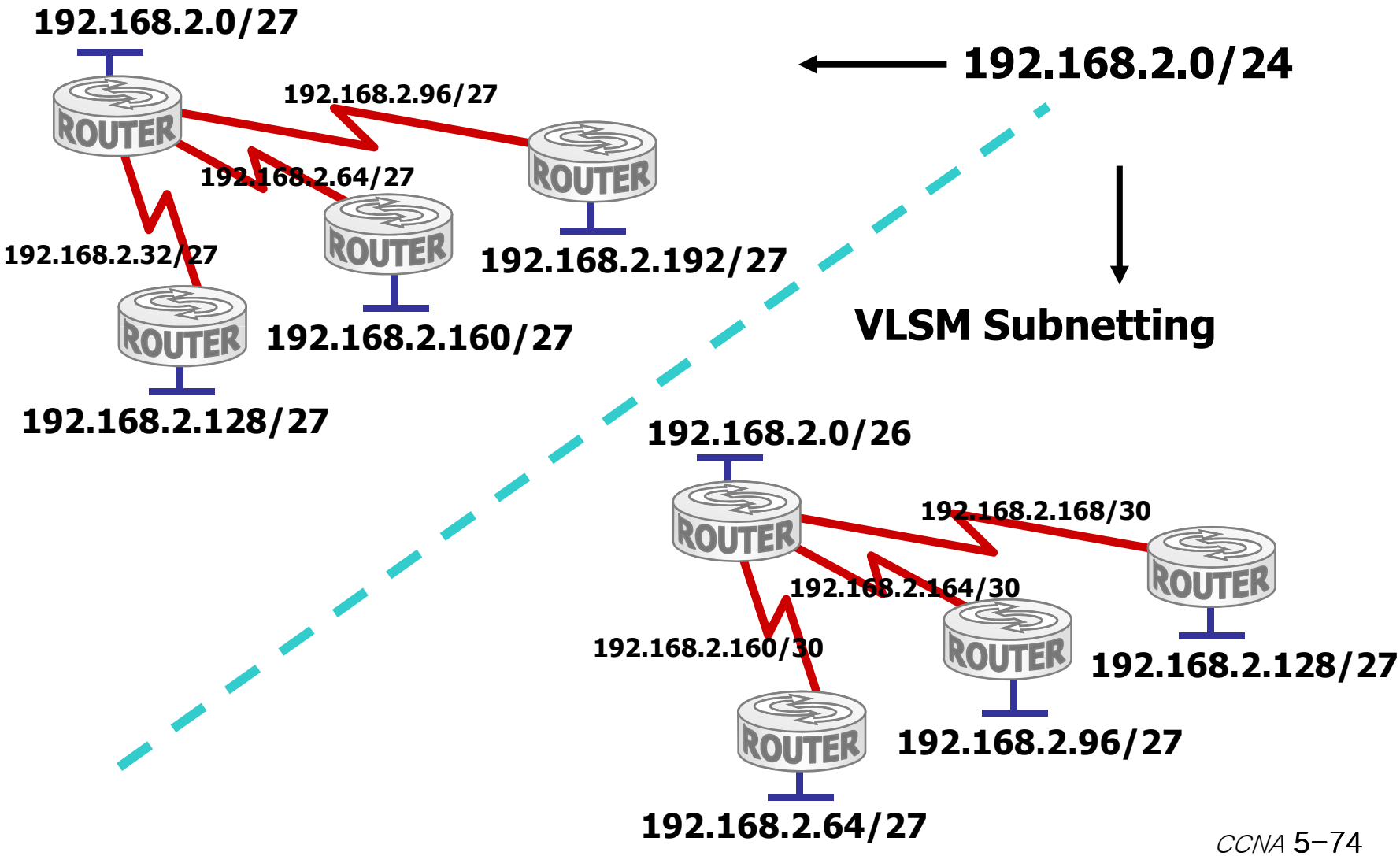
- **VLSM** 개요
- **Standard**와 **VLSM Subnetting** 방법
- **Route Summarization**

# VLSM 개요

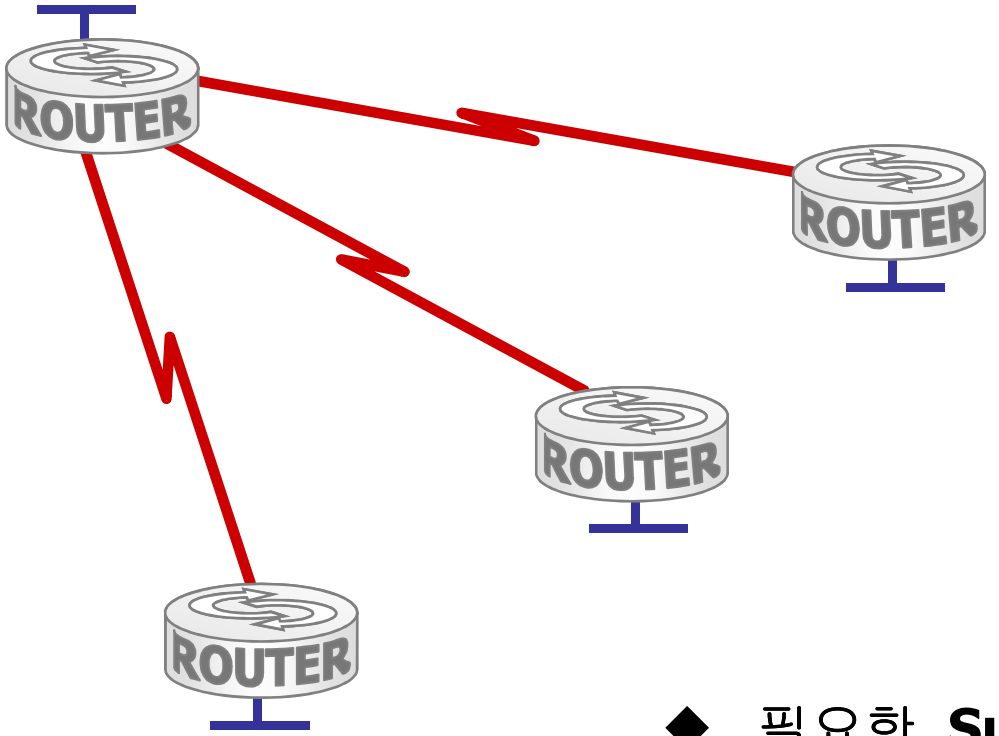


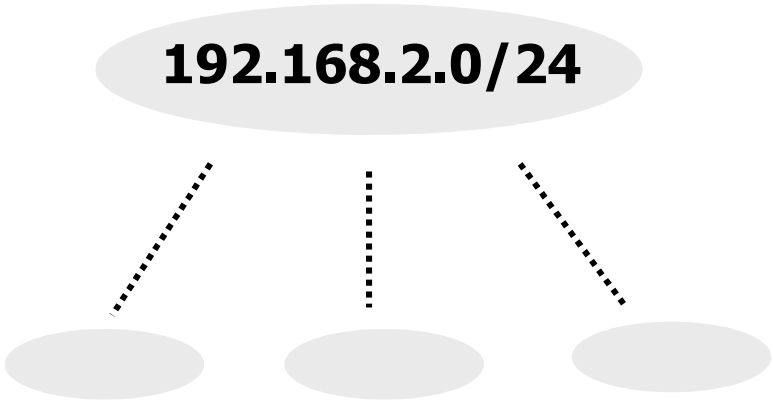

- **IP Addrress** 배분을 효율적으로 사용 가능 하다.
- **Route Summarization**의 효율성을 극대화 할 수 있다.

# Standard / VLSM Subnetting



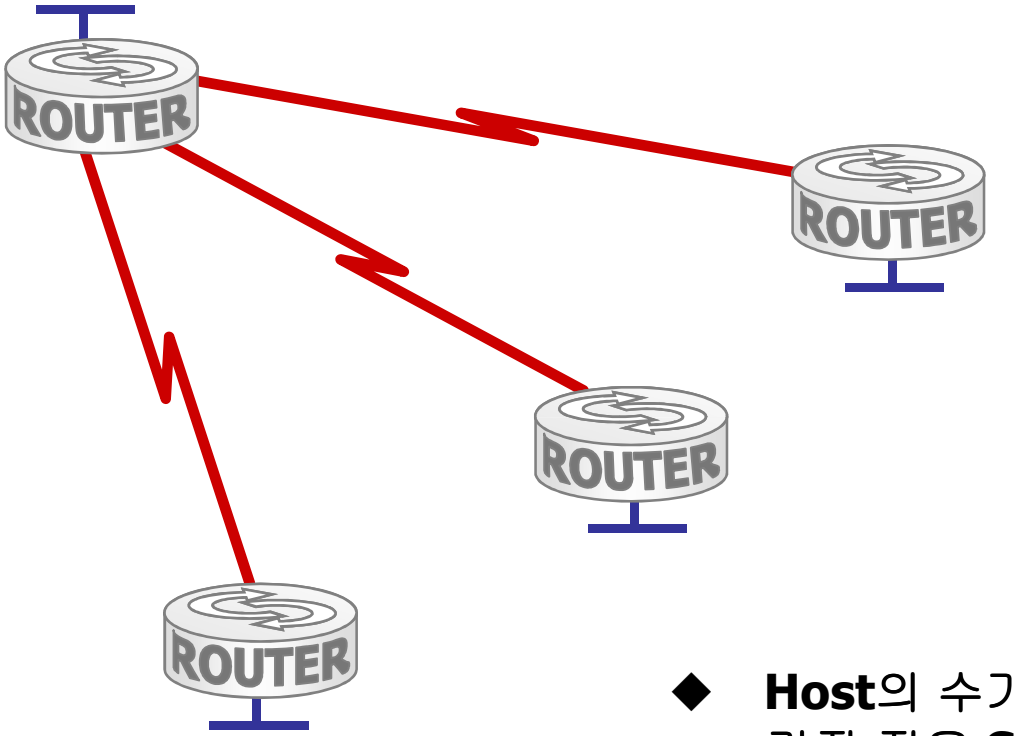
# Standard / VLSM Subnetting

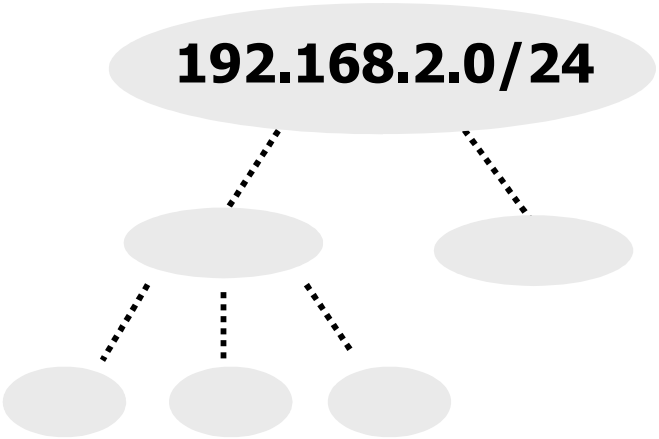



- 필요한 **Subnet**의 수를 결정
- **Subnetting**을 위한 **Subnet Bit**의 수 결정
- **IP Subnet**의 범위 확인

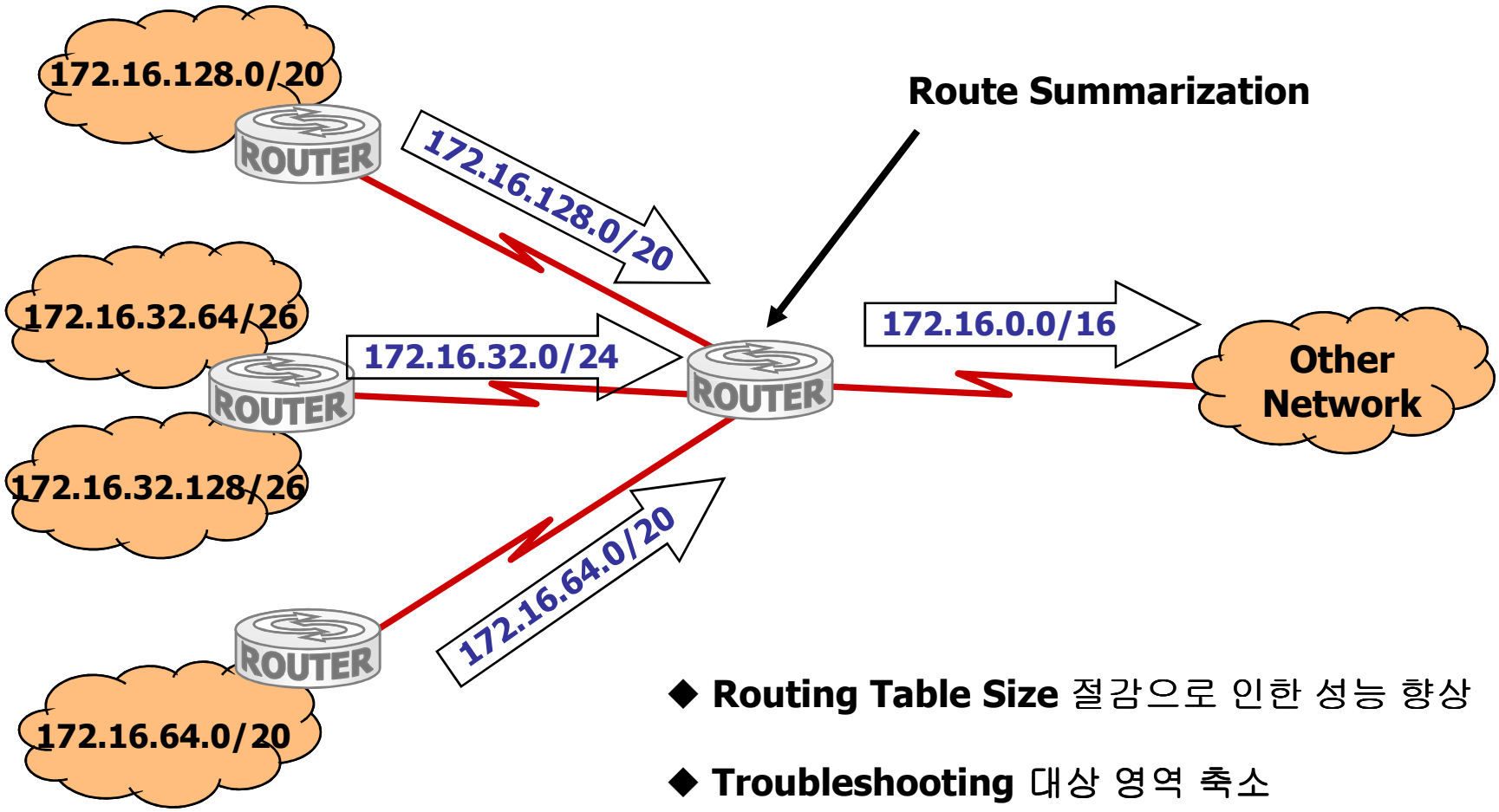
# Standard / VLSM Subnetting




- **Host**의 수가 가장 많은 **Subnet**와 **Host**의 수가 가장 적은 **Subnet**들을 계층적으로 구분
- 각 **Subnet**들 위한 **Subnet Bit**의 수 결정
- **IP Subnet**의 범위 확인

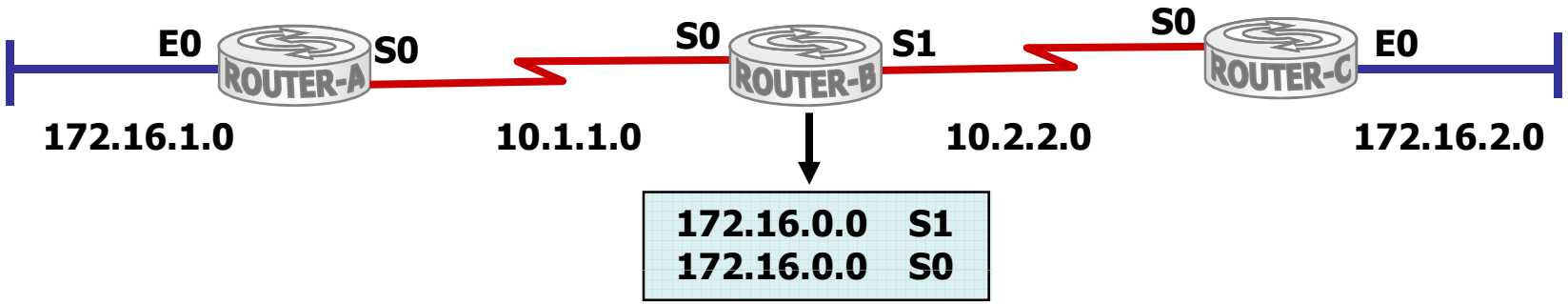
# Route Summarization



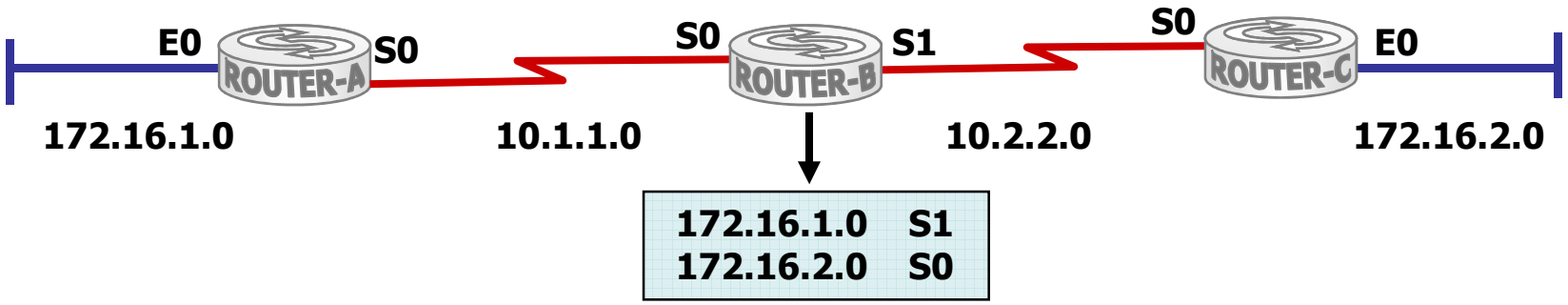

- **Routing Table Size** 절감으로 인한 성능 향상
- **Troubleshooting** 대상 영역 축소
- 계층적 **Network Design** 요구

# Route Summarization

**RIP Version 1과 IGRP의 경우**




**RIP V2, EIGRP, OSPF, IS-IS의 경우**



# Distance Vector Routing LAB

- **RIPv1** 구성하기
- **IGRP** 구성하기
- **RIPv2** 구성하기 및 **Summary**