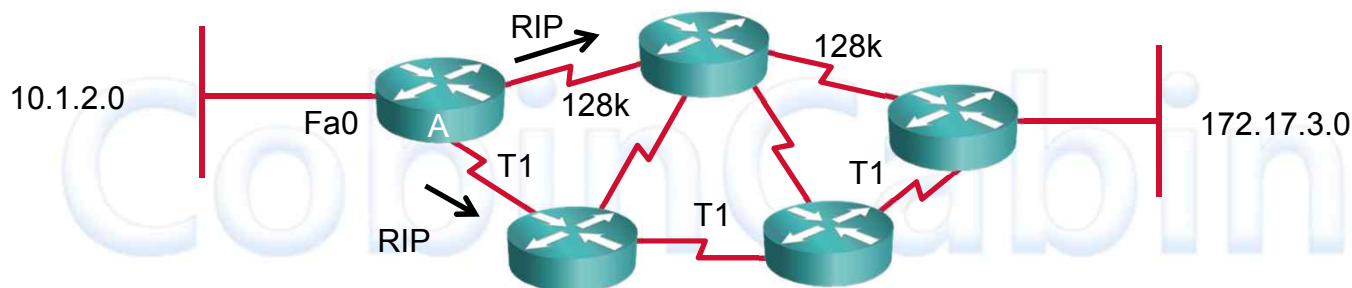


# Distance Vector Routing

## Distance Vector Routing 개요

### Metric



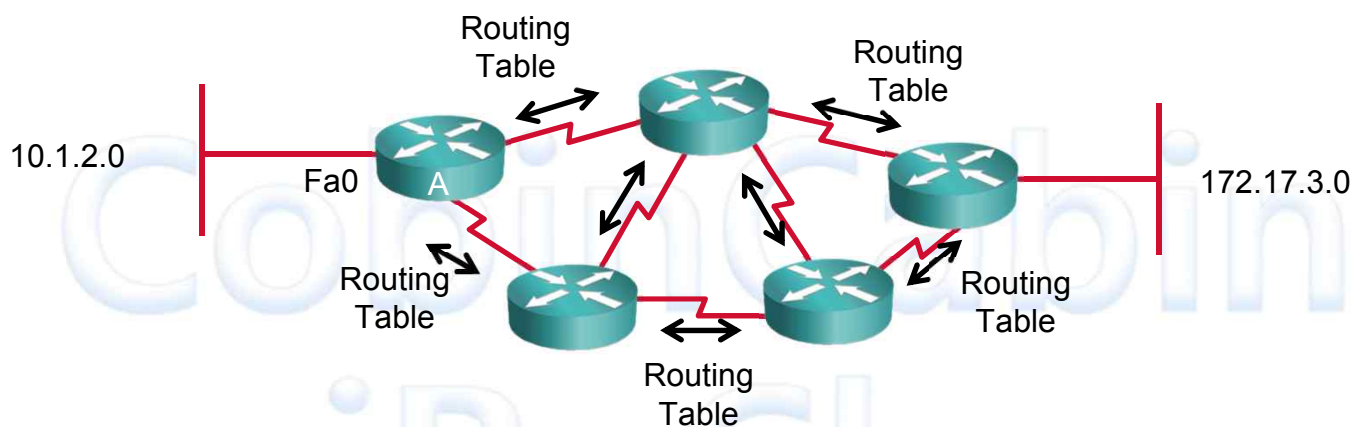
- Metric
  - 목적지 네트워크까지 도달하는데 지불해야 하는 비용
  - 목적지 네트워크까지 존재하는 여러 개의 경로 중에 하나의 경로를 선택할 때 metric값이 낮은 것을 선택한다

- RIP Metric
  - Hop Count
- EIGRP Metric
  - Bandwidth, Delay
  - Load, Reliability

# Distance Vector Routing

## Distance Vector Routing의 개요

### 경로 정보 수집

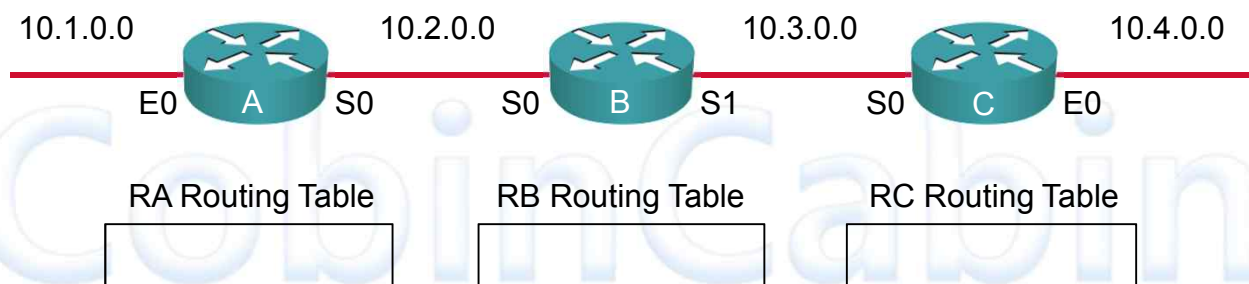


- 최적의 라우팅 리스트를 만든 후, 주기적으로 라우팅 테이블을 인접관계에 있는 라우터에게 전달
- RIP은 30초, IGRP는 90초 마다 자기가 학습한 내용을 인접 장비에게 전송

# Distance Vector Routing

## Distance Vector Routing의 개요

### 경로 정보 수집

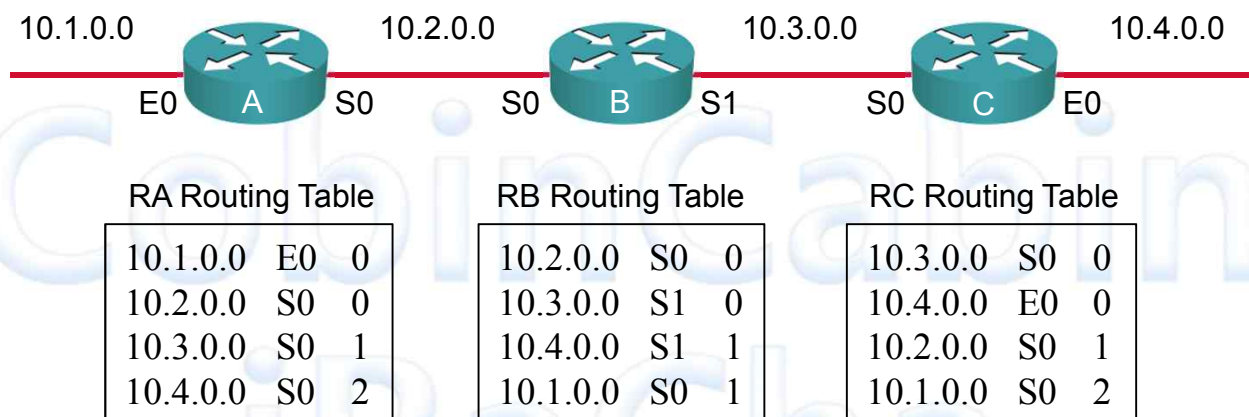


- Router들은 Destination Network에 대해 최적의 경로를 선택 후 이를 관리 및 유지 한다
  - Connected Network 정보 학습
  - Route 초기 정보 교환
  - Route 정보 교환

# Distance Vector Routing

## Distance Vector Routing의 개요

### 경로 정보 수집

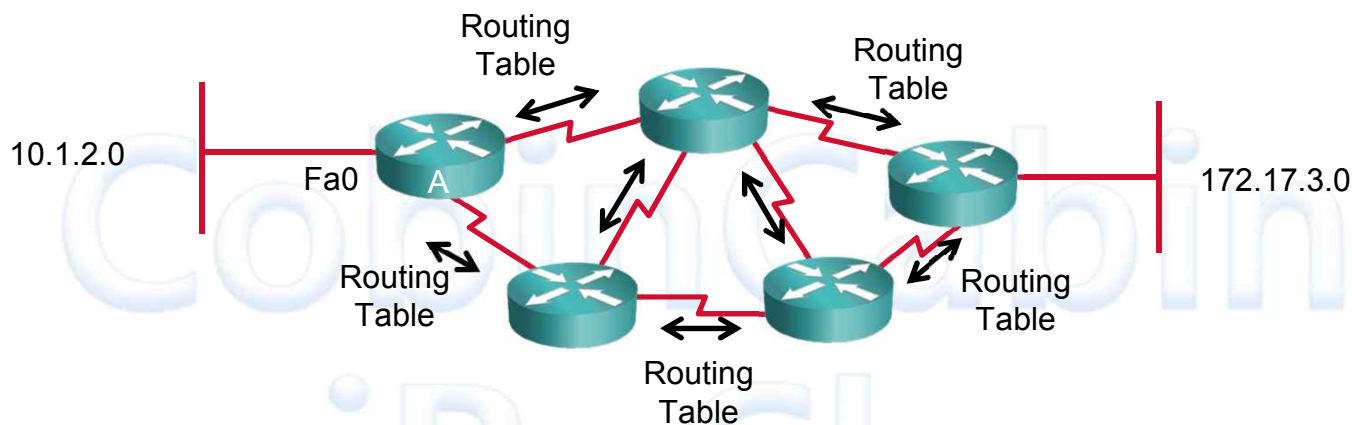


- 수렴
  - 모든 장비가 변경된 네트워크 정보를 학습하고, 학습한 정보를 경로 선택에 적용할 수 있는 상태

# Distance Vector Routing

## Routing 정보 관리

### Routing Information up-date



- Update 방식
  - 라우터에서 다른 라우터로 Hop-by-Hop으로 진행
- Update 순서
  - 1.Topology의 변화가 Routing Table의 갱신을 요구
  - 2.자신의 Routing Table의 갱신을 처리
  - 3.Router A가 갱신된 Routing Table을 다음 Update 주기에 전송
  - 4.Routing Table의 갱신을 처리

# *Distance Vector Routing*

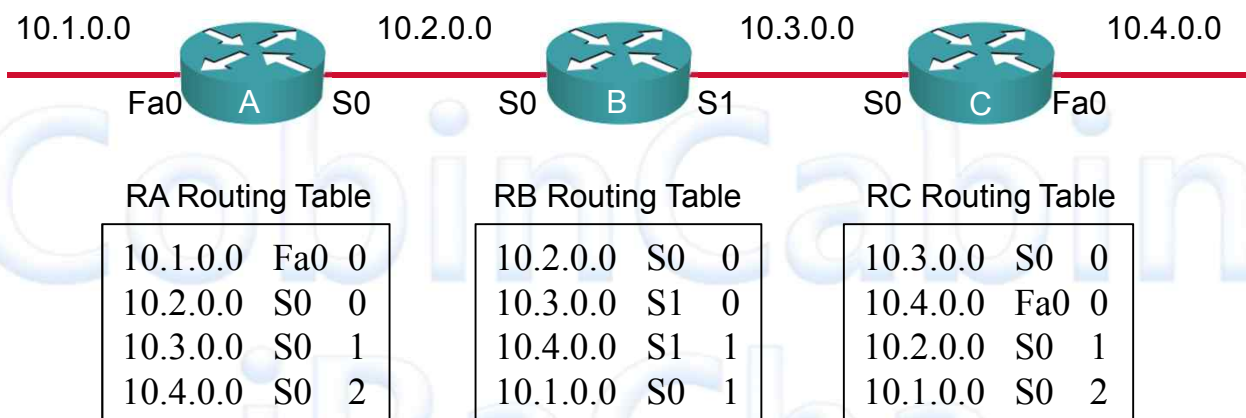
## **Routing 정보 관리**

- Update 정보를 수신한 장비는 수신 내용을 적용해서 네트워크 경로를 결정한 다음 다른 라우터로 전송 (Hop-by-Hop)
- 주기적 업데이트
  - 30초/90초 간격으로 네트워크정보를 전부 브로드캐스트, 또는 멀티캐스트
  - RIP : 30sec
  - IGRP : 90sec
- 타이머
  - Invalid 180초
  - Hold down 180초
  - Flush 240초

# Distance Vector Routing

## Distance Vector / Routing Loop

### Routing Loop 예제



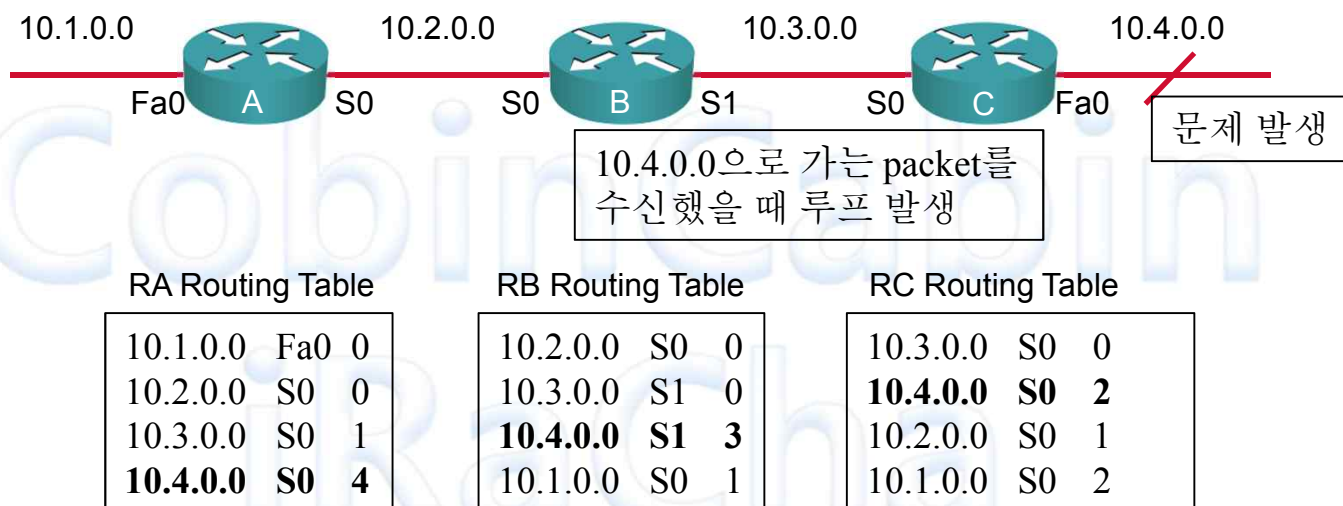
일반적인 상황에서의 Routing Table

- Routing Loop
  - Packet이 특정 장비를 따라 계속 맴도는 현상
  - Loop가 발생하면 목적지 네트워크에 도달할 수 없다.

# Distance Vector Routing

## Distance Vector / Routing Loop

### Routing Loop 예제 (계속)



- Hop Count

- 패킷이 목적지 네트워크로 전달 되는 동안 거쳐가는 장비의 수를 말하며, 최대 15대를 넘어서 전달 될 수 없다

- Hop Count = 16

- 도달할 수 없는 네트워크를 의미한다

- 해결 방법

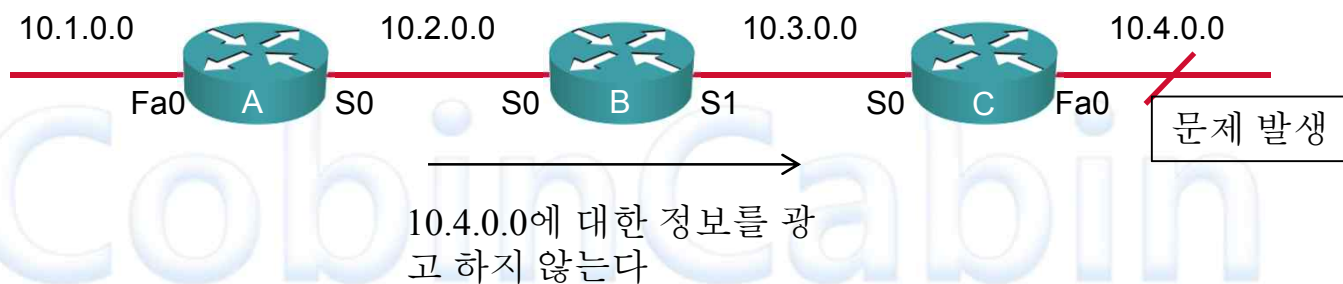
- Split Horizon
- Route Poisoning
- Poison Reverse
- Hold Down Timer
- Triggered Update



# Distance Vector Routing

## Routing Loop 문제 해결

### Split Horizon



RA Routing Table

10.1.0.0	Fa0	0
10.2.0.0	S0	0
10.3.0.0	S0	1
10.4.0.0	S0	2

RB 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

RC Routing Table

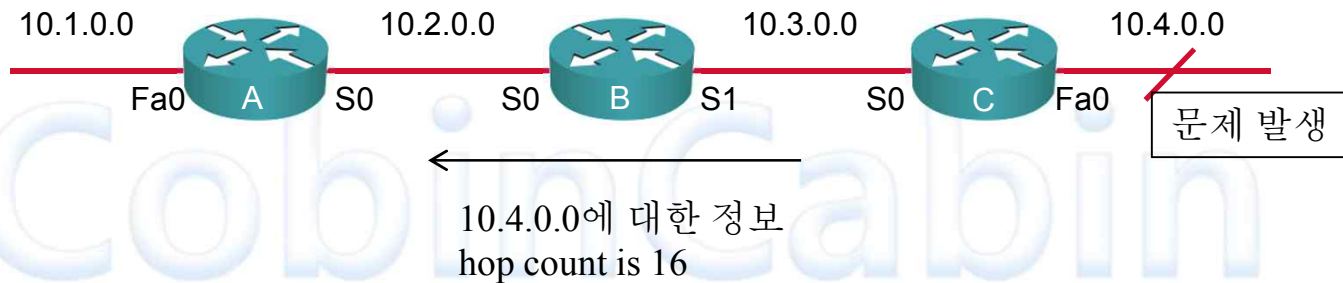
10.3.0.0	S0	0
10.4.0.0	Fa0	0
10.2.0.0	S0	1
10.1.0.0	S0	2

- 특정 interface에서 받아온 Route정보는 수신 interface를 통해 다시 전달하지 않는다

# Distance Vector Routing

## Routing Loop 문제 해결

### Route Poisoning



RA Routing Table

10.1.0.0	Fa0	0
10.2.0.0	S0	0
10.3.0.0	S0	1
<b>10.4.0.0</b>	<b>S0</b>	<b>16</b>

RB Routing Table

10.2.0.0	S0	0
10.3.0.0	S1	0
<b>10.4.0.0</b>	<b>S1</b>	<b>16</b>
10.1.0.0	S0	1

RC Routing Table

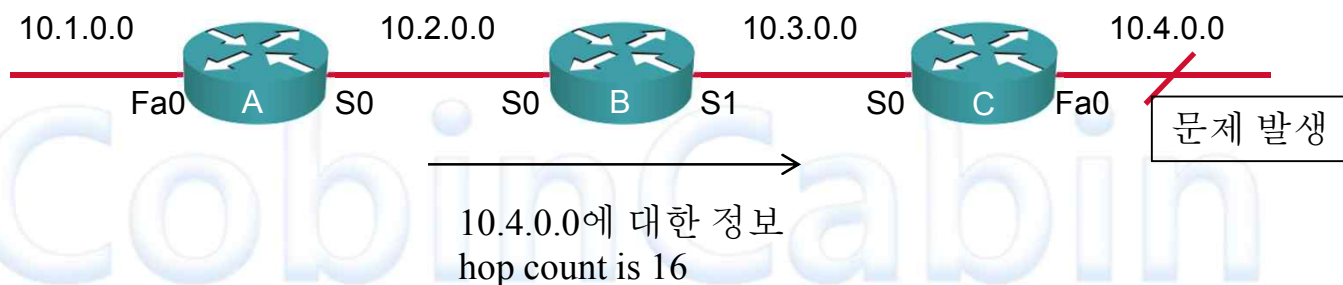
10.3.0.0	S0	0
<b>10.4.0.0</b>	<b>Fa0</b>	<b>제거</b>
10.2.0.0	S0	1
10.1.0.0	S0	2

- 문제가 발생한 네트워크의 metric 값을 무한대 값으로 광고한다

# Distance Vector Routing

## Routing Loop 문제 해결

### Poison Reverse



RA Routing Table

10.1.0.0	Fa0	0
10.2.0.0	S0	0
10.3.0.0	S0	1
10.4.0.0	S0	2

RB 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

RC Routing Table

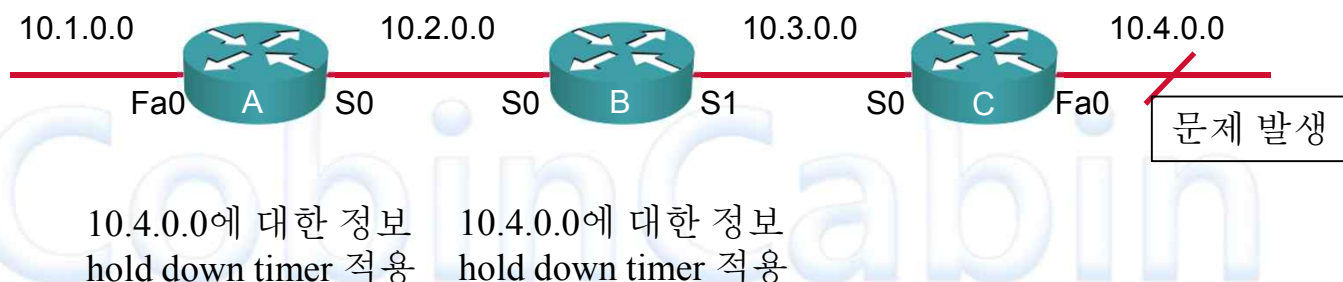
10.3.0.0	S0	0
<b>10.4.0.0</b>	<b>Fa0</b>	<b>제거</b>
10.2.0.0	S0	1
10.1.0.0	S0	2

- 수신된 정보에 대해 metric 값을 무한대로 재 광고 한다.
- Split Horizon보다 우선 한다

# Distance Vector Routing

## Routing Loop 문제 해결

### Hold down Timer



RA Routing Table

10.1.0.0	Fa0	0
10.2.0.0	S0	0
10.3.0.0	S0	1
10.4.0.0	S0	2

RB 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

RC Routing Table

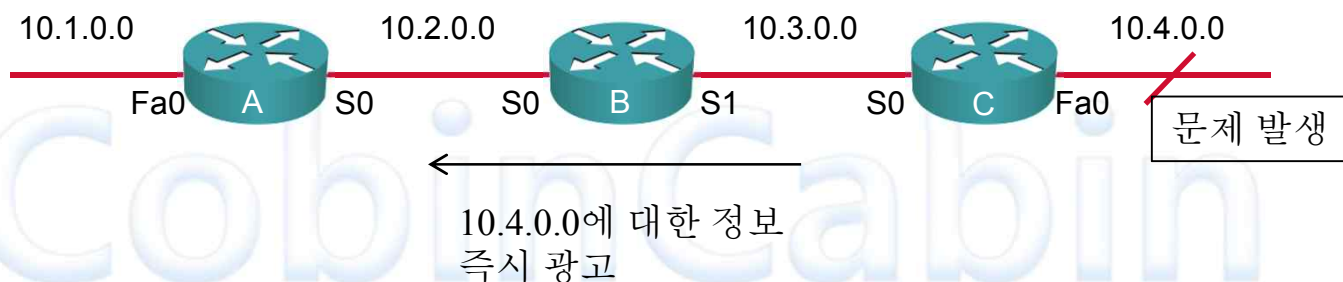
10.3.0.0	S0	0
<b>10.4.0.0</b>	<b>Fa0</b>	<b>제거</b>
10.2.0.0	S0	1
10.1.0.0	S0	2

- Router가 특정 Link의 Fail을 전달 받은 후에 해당 경로를 Routing Table에서 바로 제거하지 않고 특정 시간 동안 그 정보의 사실을 확인하기 위해 기다린다
- 이는 Topology의 변화 정보를 검증하는 용도이다
- 현재 경로와 같거나 불리한 정보를 수신하면 모두 무시한다

# Distance Vector Routing

## Routing Loop 문제 해결

### Triggered Update



RA Routing Table

10.1.0.0	Fa0	0
10.2.0.0	S0	0
10.3.0.0	S0	1
<b>10.4.0.0</b>	<b>S0</b>	<b>16</b>

RB Routing Table

10.2.0.0	S0	0
10.3.0.0	S1	0
<b>10.4.0.0</b>	<b>S1</b>	<b>16</b>
10.1.0.0	S0	1

RC Routing Table

10.3.0.0	S0	0
<b>10.4.0.0</b>	<b>Fa0</b>	<b>제거</b>
10.2.0.0	S0	1
10.1.0.0	S0	2

- Topology의 변화를 즉시 이웃한 Router에게 알려준다