

# TCP/IP 프로토콜

## -11장 유니캐스트 라우팅 프로토콜-

이 정 민([jeongmin@pel.sejong.ac.kr](mailto:jeongmin@pel.sejong.ac.kr))

세종대학교 프로토콜공학연구실

# 목 차

---

- 자율 시스템
- 거리 벡터 라우팅
- 링크 상태 라우팅
- 경로 벡터 라우팅

# 목 차

---

- 자율 시스템
- 거리 벡터 라우팅
- 링크 상태 라우팅
- 경로 벡터 라우팅

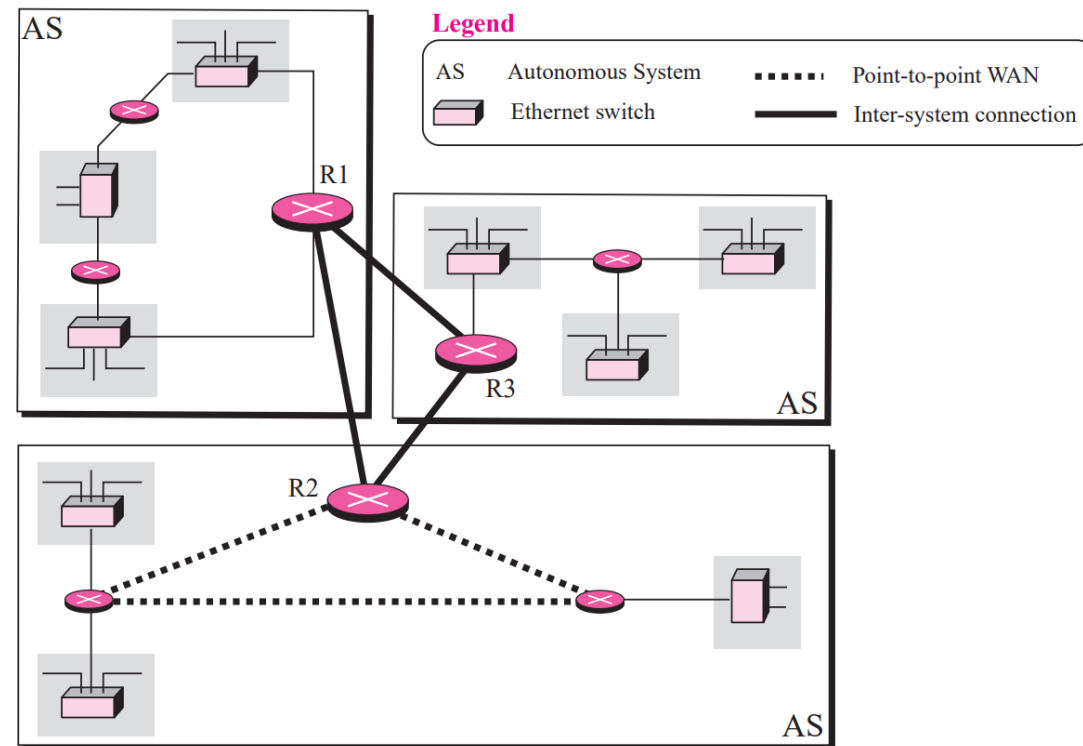
# 자율 시스템 (1/2)

- 정의

- 단일 관리 기관 하에 있는 라우터와 네트워크의 그룹

- 도입 배경

- 근래의 인터넷은 한 가지 라우팅 프로토콜로 모든 라우터의 라우팅 테이블 갱신 작업을 수행하기에는 부족할 정도로 확장됨

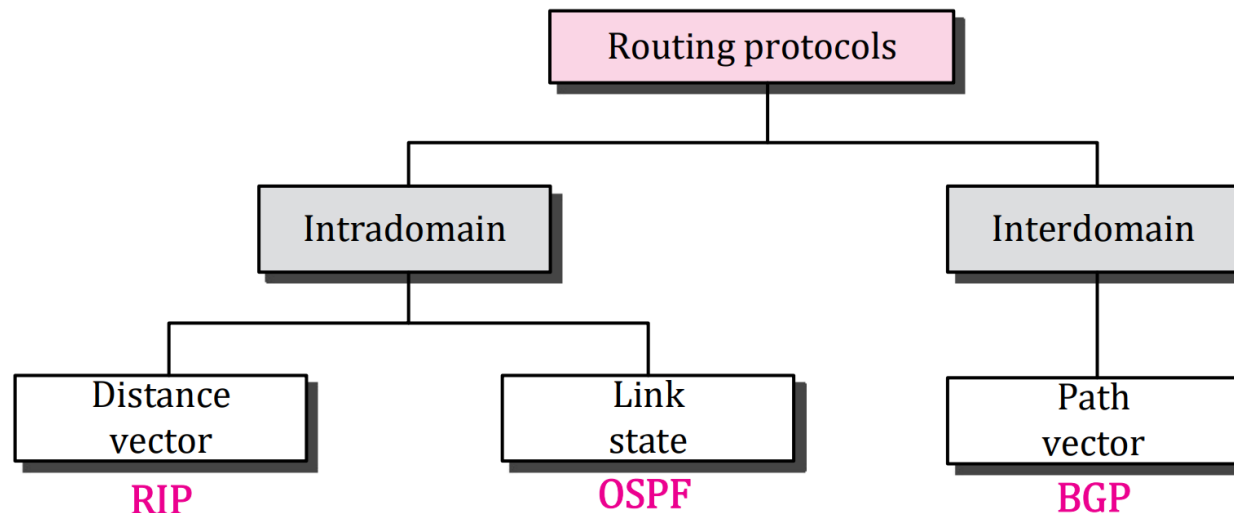


# 자율 시스템 (2/2)

## • 특징

- 자율 시스템 내 라우팅을 처리하기 위해, 하나 이상의 도메인 내 라우팅 프로토콜이 사용됨
- 자율 시스템 간 라우팅을 처리하기 위해, 하나의 도메인 간 라우팅 프로토콜이 사용됨

## • 라우팅 프로토콜의 종류



# 목 차

---

- 자율 시스템
- 거리 벡터 라우팅
- 링크 상태 라우팅
- 경로 벡터 라우팅

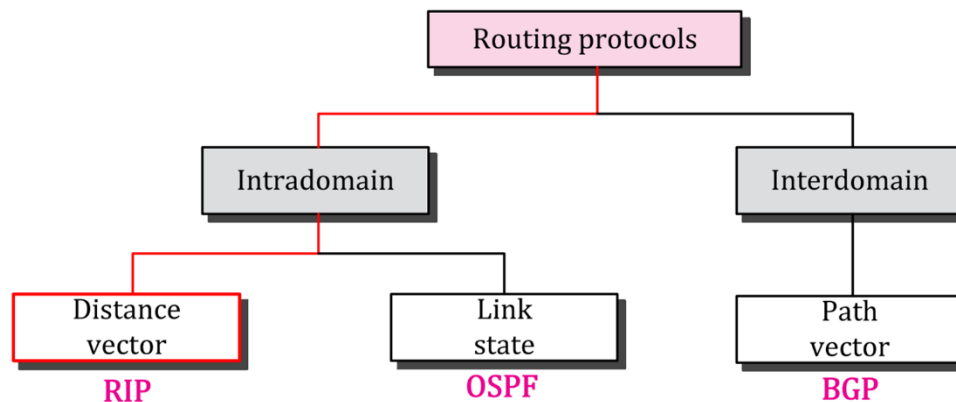
# 거리 벡터 라우팅 (1/27)

- 정의

- 인접 라우터와 정보를 공유하여 목적지까지의 경로를 결정하는 라우팅 방식

- 특징

- 소규모 네트워크에 적합함
- Bellman-Ford 알고리즘을 사용함
- RIP (Routing Information Protocol)로 구현됨



# 거리 벡터 라우팅 (2/27)

## • Bellman-Ford 알고리즘 (1/2)

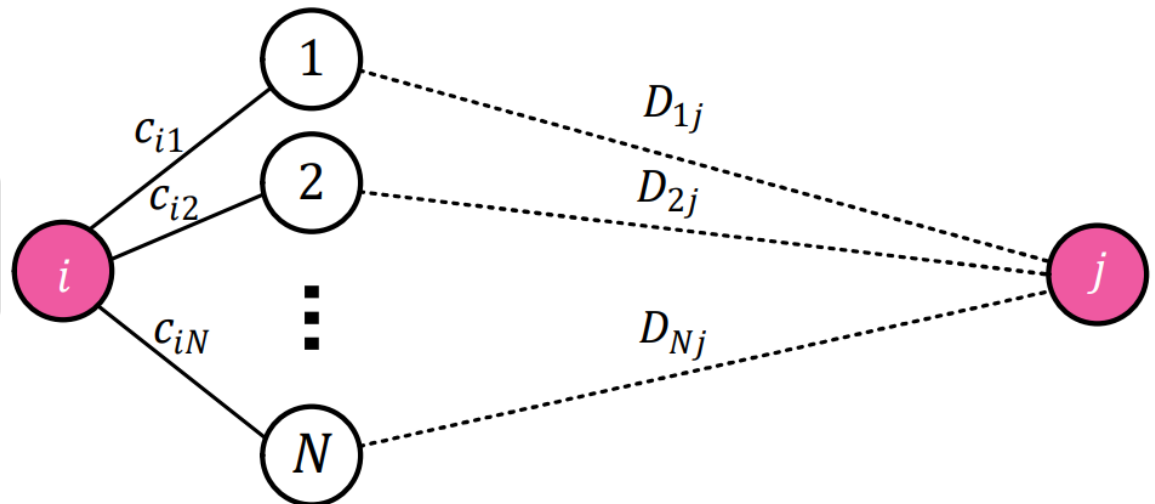
### • 정의

- 임의의 두 노드 쌍 간의 비용을 안다면 두 노드 간의 최소 비용 (최단 거리)를 찾을 수 있는 알고리즘

$$D_{ij} = \text{minimum} \{ (c_{i1} + D_{1j}), (c_{i2} + D_{2j}) \dots (c_{iN} + D_{Nj}) \}$$

### Legend

$D_{ij}$  Shortest distance between  $i$  and  $j$   
 $c_{ij}$  Cost between  $i$  and  $j$   
 $N$  Number of nodes



# 거리 벡터 라우팅 (3/27)

## • Bellman-Ford 알고리즘 (2/2)

### • 과정

1. 각 노드에서 자신 간의 최단 거리 및 비용을 0으로 초기화함
2. 각 노드에서 다른 노드와의 최단 거리를 무한대로 설정함
3. 각 노드에서 다른 노드와의 비용을 주어진 값으로 설정하되, 연결이 없으면 무한대로 설정함
4. 최단 거리 벡터에 변경 사항이 없을 때까지 다음 알고리즘을 반복함

$$D_{ij} = \min (c[i][1] + D[1][j], \dots, c[i][N] + D[N][j])$$

```
1 Bellman_Ford ()
2 {
3     // Initialization
4     for (i = 1 to N; for j = 1 to N)
5     {
6         if (i == j) D[i][j] = 0   c[i][j] = 0
7         else       D[i][j] = ∞   c[i][j] =
8             cost between i and j
9     }
10    // Updating
11    repeat
12    {
13        for (i = 1 to N; for j = 1 to N)
14        {
15            D[i][j] = min(c[i][1] + D[1][j],
16                ..., c[i][N] + D[N][j])
17        } // end for
18    } until (there was no change in previous
19        iteration)
20 } // end Bellman-Ford
```

# 거리 벡터 라우팅 (4/27)

---

- 알고리즘 (1/7)

- Bellman-Ford 알고리즘의 변형

- 각 라우터에서 이웃 노드로부터 정보를 수신하면 비동기적으로 라우팅 테이블을 변경함에 따라, 절차를 분산적으로 수행함
- 비용 설정
  - 비용은 일반적으로 홉 수로 사용됨에 따라, 연결된 두 노드간 비용이 1로 설정됨
- 테이블 전달
  - 라우터가 자신의 테이블을 갱신한 결과를 이웃 노드에게 전달하여 이웃 노드가 라우팅 테이블을 계산할 수 있도록 함
- 변수 추가
  - 각 라우터는 경로에 대해 목적지 망, 비용, 다음 홉 정보를 가짐에 따라, 관련 변수(Table.dest, Table.cost, Table.next)를 추가함
  - 이웃 노드로부터 받은 각 경로에 대한 정보 R에 대해, R.dest, R.cost 정보를 추가함

# 거리 벡터 라우팅 (5/27)

- 알고리즘 (2/7)

- 의사 코드: 초기화

```
1 Distance_Vector_Algorithm ()
2 {
3     // At startup
4     for (i = 1 to N)           // N is number of ports
5     {
6         Table[i].dest = address of the attached network
7         Table[i].cost = 1
8         Table[i].next = -      // Means at home
9         Send(R[i]) // Send a record R about each row to each neighbor
10    } // end for loop
```

- 라우터는 각 인터페이스에 직접 연결된 망들로 패킷을 라우팅하는데 사용되는 초기 라우팅 테이블을 생성함
- 라우팅 테이블에 있는 각 항목에 대해 목적지 망과 이웃 노드로의 비용을 전송함
- 두 노드 간 비용을 1로 설정하고, 관련 변수 Table, R을 사용함

# 거리 벡터 라우팅 (6/27)

- 알고리즘 (3/7)
- 의사 코드: 갱신 (1/2)

```
12 // Updating
13 repeat (forever)
14 {
15     Wait for arrival of a record R from a neighbor
16     Update (R, Table) // Call update module
17     for (i = 1 to N) // N is the current table size
18     {
19         Send(R[i]) //Send a record R about each row to each neighbor
20     }
21 } // end repeat
22 } // end Distance_Vector
```

- 각 라우터는 이웃으로부터 레코드 정보를 받으면 자신의 라우팅 테이블을 갱신함
- 자신의 테이블을 갱신한 결과를 이웃 노드에게 전달함

# 거리 벡터 라우팅 (7/27)

## • 알고리즘 (4/7)

### • 의사 코드: 갱신 (2/2)

- 레코드가 도착하면, 라우터는 라우팅 테이블에 있는 목적지 주소를 검색함

```
24 Update (R, Table)      // Update module
25 {
26     Search Table for a destination matching the one in R
27     if (destination is found in row i)
28     {
29         if (R.cost + 1 < Table[i].cost or R.next == Table[i].next)
30         {
31             Table[i].cost = R.cost + 1
32             Table[i].next = Address of sending router
33         }
34         else discard the record // No change is needed
35     }
36     else
37     {
38         // Insert the new router
39         Table[N+1].dest = R.dest
40         Table[N+1].cost = R.cost + 1
41         Table[N+1].next = Address of sending router
42         Sort(Table, orderby=destination_addresses)
43         Sort the table according to destination address
44     }
45 } // end of Update module
```

목적지 주소가 발견되면, 두 가지 경우를 점검함

레코드 값에 1을 더한 값이 테이블에 있는 비용보다 적으면, 이웃이 목적지로 가는 더 좋은 경로를 찾았음을 의미함

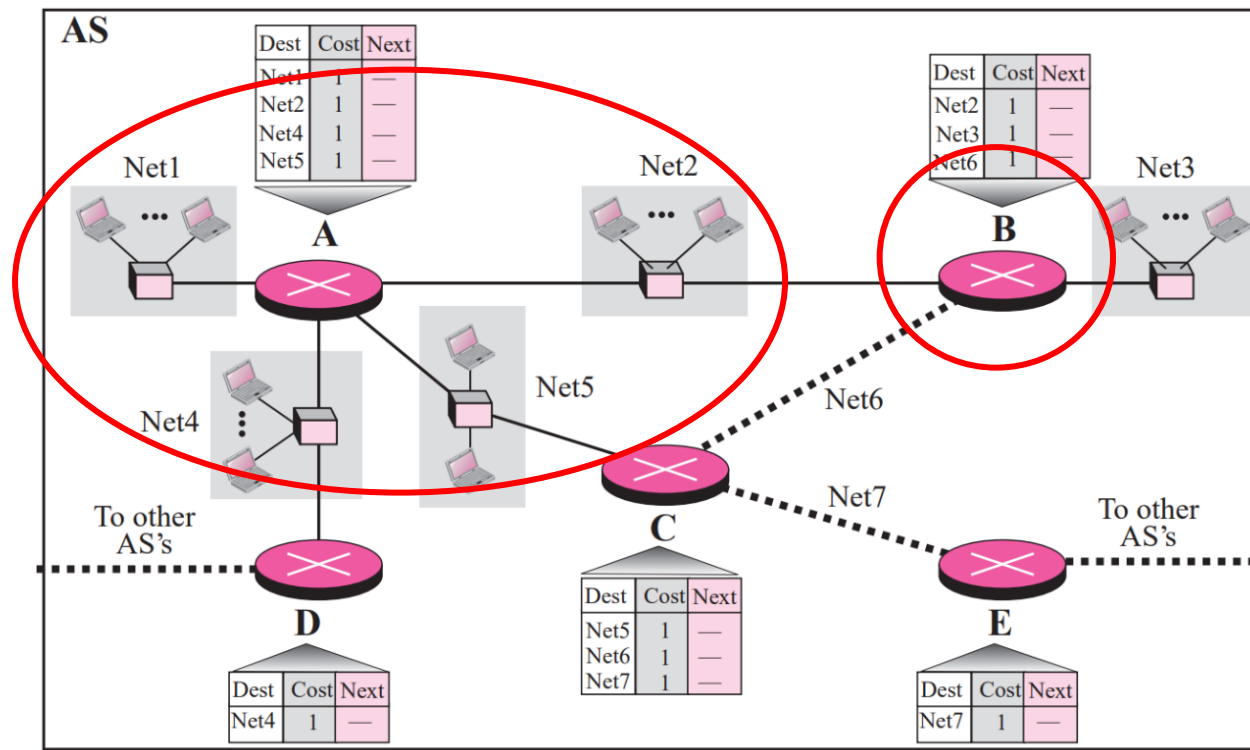
다음 홉이 같다면, 망의 일부에서 변경 사항이 있었음을 의미함 (e.g., 다음 홉으로 가는 경로가 더 이상 존재하지 않는 경우)

목적지 주소가 라우팅 테이블에 존재하지 않으면, 비용을 1 더하여 새로 추가함

# 거리 벡터 라우팅 (8/27)

- 알고리즘 (5/7)
  - 예제 11.1, 11.2 (1/3)

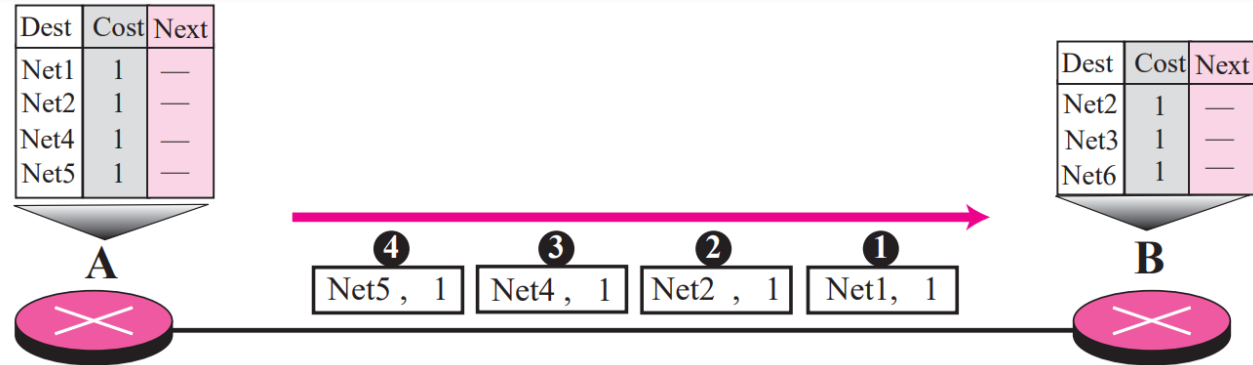
다음과 같은 네트워크에서 라우터 A가 이웃 노드인 라우터 B에 Net1, Net2, Net4, Net5 항목을 전송할 때, 라우터 B에서 라우팅 테이블의 변화를 보이시오.



# 거리 벡터 라우팅 (9/27)

- 알고리즘 (6/7)

- 예제 11.1, 11.2 (2/3)



Routing Table B	Routing Table B	Routing Table B	Routing Table B																																																																					
<table border="1"> <thead> <tr><th>Dest</th><th>Cost</th><th>Next</th></tr> </thead> <tbody> <tr><td>Net1</td><td>2</td><td>A</td></tr> <tr><td>Net2</td><td>1</td><td>—</td></tr> <tr><td>Net3</td><td>1</td><td>—</td></tr> <tr><td>Net6</td><td>1</td><td>—</td></tr> </tbody> </table> <p>After receiving record 1</p>	Dest	Cost	Next	Net1	2	A	Net2	1	—	Net3	1	—	Net6	1	—	<table border="1"> <thead> <tr><th>Dest</th><th>Cost</th><th>Next</th></tr> </thead> <tbody> <tr><td>Net1</td><td>2</td><td>A</td></tr> <tr><td>Net2</td><td>1</td><td>—</td></tr> <tr><td>Net3</td><td>1</td><td>—</td></tr> <tr><td>Net6</td><td>1</td><td>—</td></tr> </tbody> </table> <p>After receiving record 2</p>	Dest	Cost	Next	Net1	2	A	Net2	1	—	Net3	1	—	Net6	1	—	<table border="1"> <thead> <tr><th>Dest</th><th>Cost</th><th>Next</th></tr> </thead> <tbody> <tr><td>Net1</td><td>2</td><td>A</td></tr> <tr><td>Net2</td><td>1</td><td>—</td></tr> <tr><td>Net3</td><td>1</td><td>—</td></tr> <tr><td>Net4</td><td>2</td><td>A</td></tr> <tr><td>Net6</td><td>1</td><td>—</td></tr> </tbody> </table> <p>After receiving record 3</p>	Dest	Cost	Next	Net1	2	A	Net2	1	—	Net3	1	—	Net4	2	A	Net6	1	—	<table border="1"> <thead> <tr><th>Dest</th><th>Cost</th><th>Next</th></tr> </thead> <tbody> <tr><td>Net1</td><td>2</td><td>A</td></tr> <tr><td>Net2</td><td>1</td><td>—</td></tr> <tr><td>Net3</td><td>1</td><td>—</td></tr> <tr><td>Net4</td><td>2</td><td>A</td></tr> <tr><td>Net5</td><td>2</td><td>A</td></tr> <tr><td>Net6</td><td>1</td><td>—</td></tr> </tbody> </table> <p>After receiving record 4</p>	Dest	Cost	Next	Net1	2	A	Net2	1	—	Net3	1	—	Net4	2	A	Net5	2	A	Net6	1	—
Dest	Cost	Next																																																																						
Net1	2	A																																																																						
Net2	1	—																																																																						
Net3	1	—																																																																						
Net6	1	—																																																																						
Dest	Cost	Next																																																																						
Net1	2	A																																																																						
Net2	1	—																																																																						
Net3	1	—																																																																						
Net6	1	—																																																																						
Dest	Cost	Next																																																																						
Net1	2	A																																																																						
Net2	1	—																																																																						
Net3	1	—																																																																						
Net4	2	A																																																																						
Net6	1	—																																																																						
Dest	Cost	Next																																																																						
Net1	2	A																																																																						
Net2	1	—																																																																						
Net3	1	—																																																																						
Net4	2	A																																																																						
Net5	2	A																																																																						
Net6	1	—																																																																						

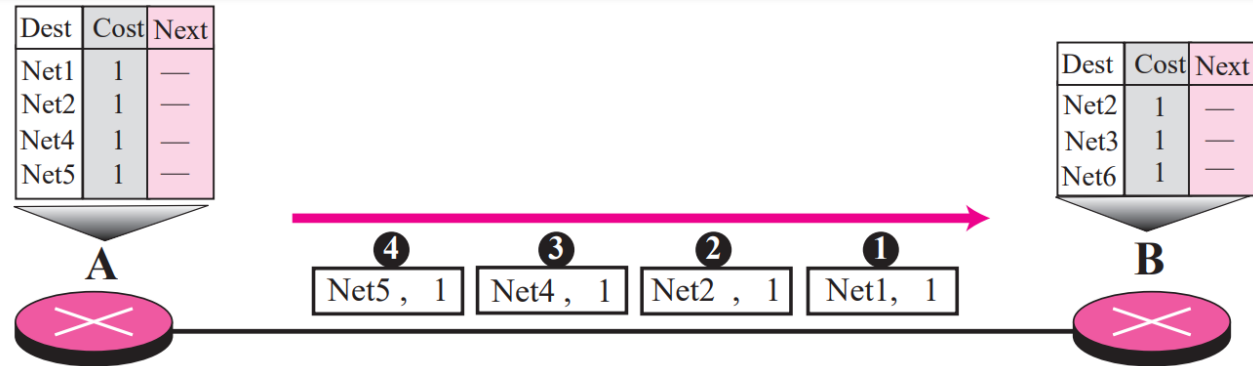
- 풀이 (1/2)

- 라우터 B가 레코드 1을 수신할때, Net1으로 가는 경로를 라우팅 테이블에서 검색하고 찾을 수 없기 때문에 비용을 1 더해서 (A와 B 사이의 거리) 기록하게 됨
- 레코드 2를 수신하면 경로가 있음을 확인하고, 비용을 1 더해서 테이블에 있던 값과 비교, 테이블에 있는 값보다 크기 때문에 해당 레코드를 버림

# 거리 벡터 라우팅 (10/27)

- 알고리즘 (6/7)

- 예제 11.1, 11.2 (3/3)



Routing Table B	Routing Table B	Routing Table B	Routing Table B																																																																					
<table border="1"> <thead> <tr><th>Dest</th><th>Cost</th><th>Next</th></tr> </thead> <tbody> <tr><td>Net1</td><td>2</td><td>A</td></tr> <tr><td>Net2</td><td>1</td><td>—</td></tr> <tr><td>Net3</td><td>1</td><td>—</td></tr> <tr><td>Net6</td><td>1</td><td>—</td></tr> </tbody> </table> <p>After receiving record 1</p>	Dest	Cost	Next	Net1	2	A	Net2	1	—	Net3	1	—	Net6	1	—	<table border="1"> <thead> <tr><th>Dest</th><th>Cost</th><th>Next</th></tr> </thead> <tbody> <tr><td>Net1</td><td>2</td><td>A</td></tr> <tr><td>Net2</td><td>1</td><td>—</td></tr> <tr><td>Net3</td><td>1</td><td>—</td></tr> <tr><td>Net6</td><td>1</td><td>—</td></tr> </tbody> </table> <p>After receiving record 2</p>	Dest	Cost	Next	Net1	2	A	Net2	1	—	Net3	1	—	Net6	1	—	<table border="1"> <thead> <tr><th>Dest</th><th>Cost</th><th>Next</th></tr> </thead> <tbody> <tr><td>Net1</td><td>2</td><td>A</td></tr> <tr><td>Net2</td><td>1</td><td>—</td></tr> <tr><td>Net3</td><td>1</td><td>—</td></tr> <tr><td>Net4</td><td>2</td><td>A</td></tr> <tr><td>Net6</td><td>1</td><td>—</td></tr> </tbody> </table> <p>After receiving record 3</p>	Dest	Cost	Next	Net1	2	A	Net2	1	—	Net3	1	—	Net4	2	A	Net6	1	—	<table border="1"> <thead> <tr><th>Dest</th><th>Cost</th><th>Next</th></tr> </thead> <tbody> <tr><td>Net1</td><td>2</td><td>A</td></tr> <tr><td>Net2</td><td>1</td><td>—</td></tr> <tr><td>Net3</td><td>1</td><td>—</td></tr> <tr><td>Net4</td><td>2</td><td>A</td></tr> <tr><td>Net5</td><td>2</td><td>A</td></tr> <tr><td>Net6</td><td>1</td><td>—</td></tr> </tbody> </table> <p>After receiving record 4</p>	Dest	Cost	Next	Net1	2	A	Net2	1	—	Net3	1	—	Net4	2	A	Net5	2	A	Net6	1	—
Dest	Cost	Next																																																																						
Net1	2	A																																																																						
Net2	1	—																																																																						
Net3	1	—																																																																						
Net6	1	—																																																																						
Dest	Cost	Next																																																																						
Net1	2	A																																																																						
Net2	1	—																																																																						
Net3	1	—																																																																						
Net6	1	—																																																																						
Dest	Cost	Next																																																																						
Net1	2	A																																																																						
Net2	1	—																																																																						
Net3	1	—																																																																						
Net4	2	A																																																																						
Net6	1	—																																																																						
Dest	Cost	Next																																																																						
Net1	2	A																																																																						
Net2	1	—																																																																						
Net3	1	—																																																																						
Net4	2	A																																																																						
Net5	2	A																																																																						
Net6	1	—																																																																						

- 풀이 (2/2)

- 라우터 B가 레코드 3을 수신할 때, 테이블에 Net4가 없기 때문에 비용을 1 더하여 테이블에 추가됨
- 라우터 B가 레코드 4를 수신할 때, 테이블에 Net5가 없기 때문에 비용을 1 더하여 테이블에 추가됨

# 거리 벡터 라우팅 (11/27)

---

- 무한대로의 카운트 (Count to Infinity) 문제 (1/6)
  - 링크의 고장 등으로 인해 비용이 무한대로 바뀌었을 때, 이를 모든 라우터가 인식하는데 시간이 소요되어 발생하는 문제
- 발생 원인
  - 거리 벡터 라우팅에서 비용 감소 정보는 빠르게 확산되나, 비용 증가와 같은 정보는 갱신이 느리기 때문임

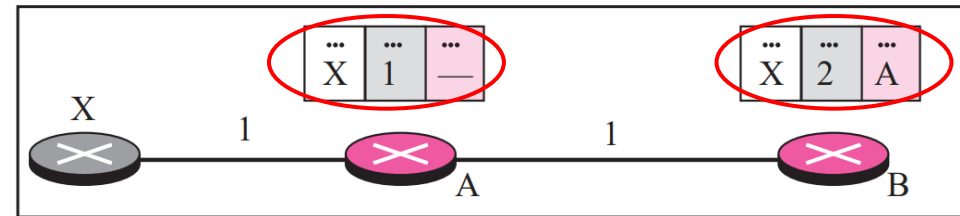
# 거리 벡터 라우팅 (12/27)

- 무한대로의 카운트 문제 (2/6)

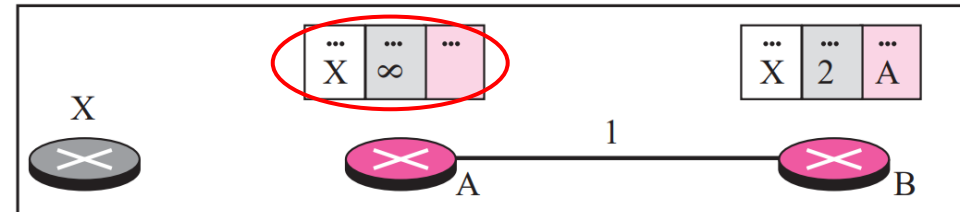
- 두 노드 불안전성 (1/3)

- 문제 상황 (1/2)

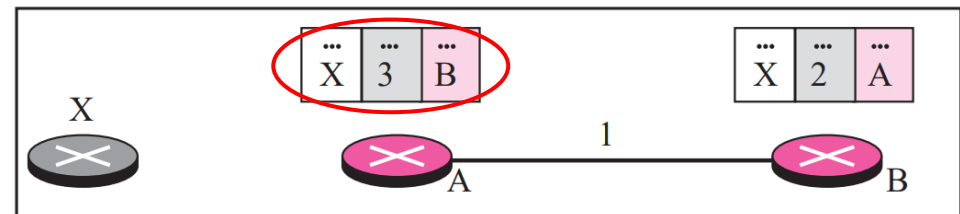
1. A와 B는 노드 X에 도달하는 방법을 알고 있음



2. A와 X 사이의 링크가 실패하여, A는 자신의 테이블을 변경함



3. B가 A의 라우팅 테이블 정보를 받기 전에, 자신의 라우팅 테이블을 보냄



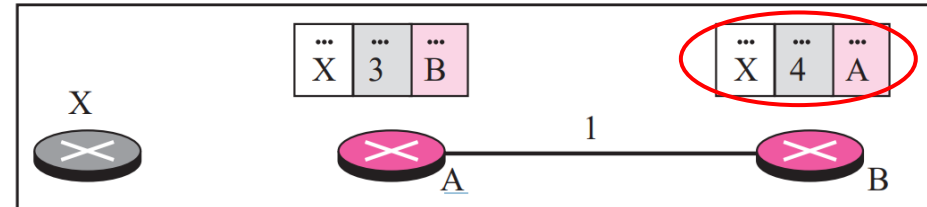
# 거리 벡터 라우팅 (13/27)

- 무한대로의 카운트 문제 (3/6)

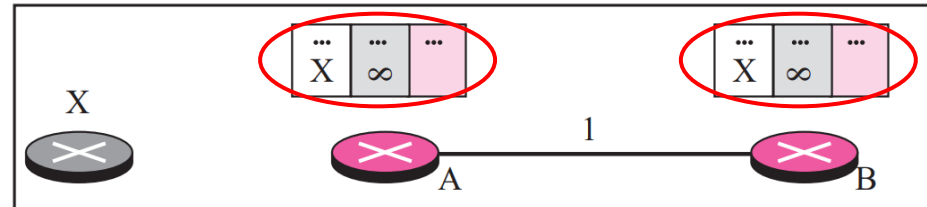
- 두 노드 불안전성 (2/3)

- 문제 상황 (2/2)

4. A는 B의 라우팅 테이블 정보를 받고 자신의 라우팅 테이블을 갱신한 내용을 B에게 보냄



5. 위 과정을 반복하여, 최종적으로 무한대의 비용이 되어서야 X에 도달할 수 없음을 알게 됨



# 거리 벡터 라우팅 (14/27)

---

- 무한대로의 카운트 문제 (4/6)
  - 두 노드 불안전성 (3/3)
    - 해결 방안
      - 무한대의 정의
        - 거리 벡터 라우팅 구현 대부분에서 16을 무한대로 정의하여 사용함
          - 16을 무한대로 정의할 경우, 망의 크기가 15 홉을 넘어설 수 없음
      - 수평 분할 (Split horizon)
        - 인터페이스를 통해 테이블을 플러딩 하는 대신에, 각 인터페이스를 통해 자신 테이블의 일부만을 전송함
          - e.g., B가 X에 도달하는 최적의 경로가 A를 거치는 것이라는 걸 알고 A에 다시 알리지 않음
        - 보통 거리 벡터 라우팅은 타이머를 사용하기 때문에 수평 분할 정책과 타이머로 인한 경로 제거를 구분하기 어렵다는 문제가 존재함
          - e.g., B가 X에 관한 정보를 A에게 보내지 않으면, A는 이것이 수평 분할 정책에 의해서 인지, B가 최근에 X에 관한 정보를 받지 못해서 인지 예측할 수 없음
          - 포이즌 리버스 (Poison reverse) 정책과 결합하여 해당 문제를 해결함
      - 포이즌 리버스
        - 이웃 라우터들에게 자신이 경유지로 사용되는 경로의 거리를 무한대로 알림
          - e.g., B가 여전히 X에 대한 정보를 보내지만, 송신자가 A인 경우 거리 값을 무한대로 설정해서 보냄

# 거리 벡터 라우팅 (15/27)

- 무한대로의 카운트 문제 (5/6)

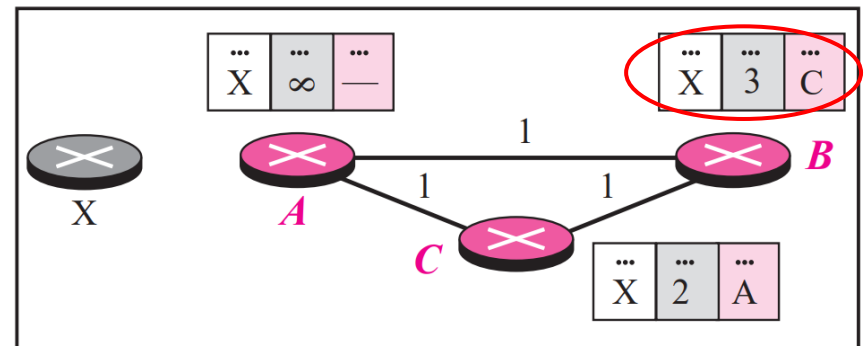
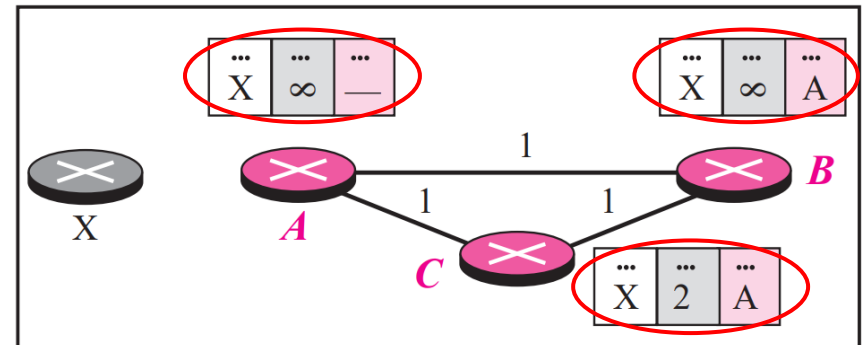
- 세 노드 불안전성 (1/2)

- 문제 상황 (1/2)

1. X가 도달 가능하지 않음을 발견한 후, A가 B, C에게 패킷을 전송해 상황을 알림

2. B는 즉각적으로 테이블을 갱신했으나, C는 패킷이 유실되어 상황을 알지 못함

3. C는 B에게 라우팅 테이블을 보내 X로의 경로를 알리고, B는 테이블을 갱신함



# 거리 벡터 라우팅 (16/27)

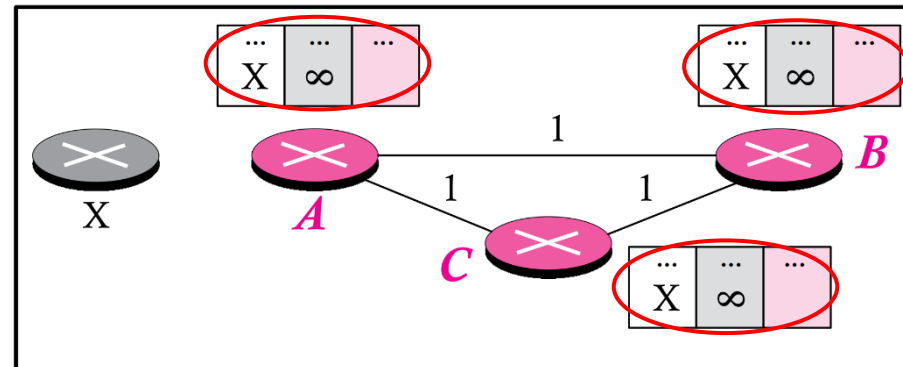
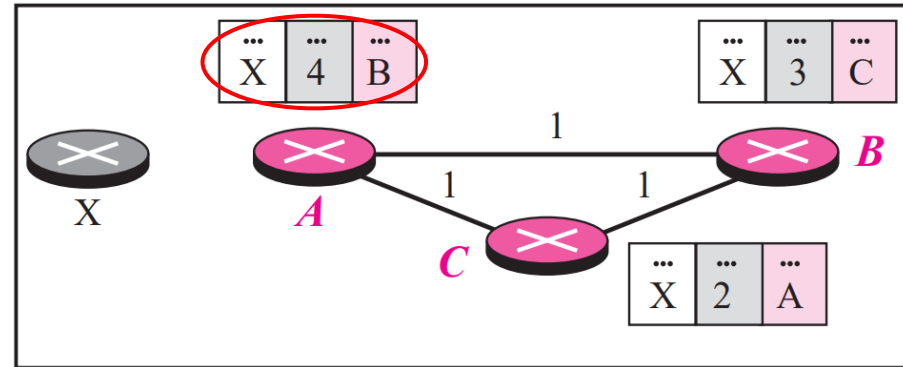
- 무한대로의 카운트 문제 (6/6)

- 세 노드 불안전성 (2/2)

- 문제 상황 (2/2)

4. B가 이 정보를 A에게 보내어, A는 테이블을 갱신함

5. 루프가 반복되어 각 노드의 비용이 무한대에 다다르면 중지됨



- 해결 방안

- 세 노드 간의 불안전성 문제는 해결이 보장되지 않음

# 거리 벡터 라우팅 (17/27)

---

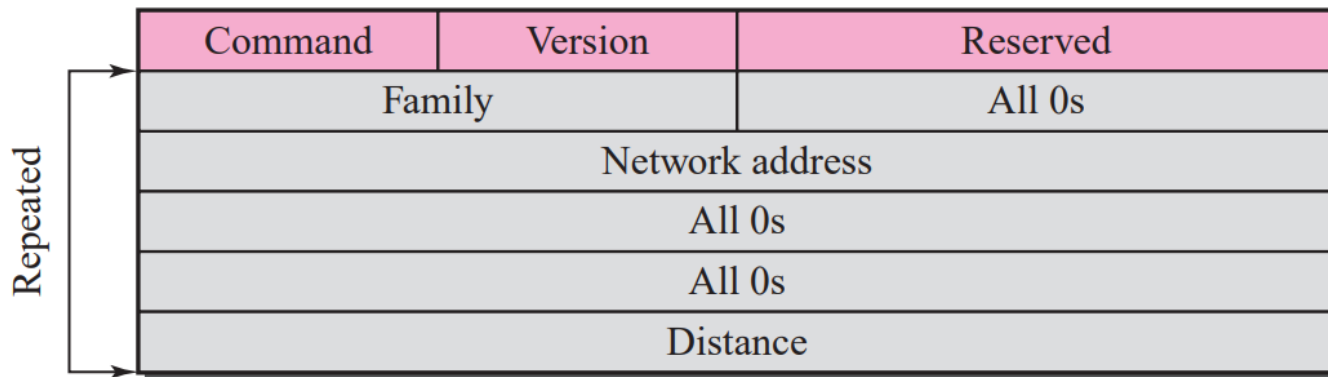
- RIP (Routing Information Protocol)
  - 정의
    - 자율 시스템 내에서 사용되는 거리 벡터 라우팅 기반 프로토콜
  - 특징
    - 홉 수를 메트릭\*으로 사용함
    - 무한대를 16으로 정의함
      - 자율 시스템 내에서 최대 15홉의 경로를 가짐

\*메트릭 (Metric): 망을 통해 전달되는 비용

# 거리 벡터 라우팅 (18/27)

- RIPv1 (1/6)
  - 메시지 형태

필드명	크기	설명
명령 (Command)	8 bits	메시지의 종류를 나타냄 요청은 1, 응답은 2의 값을 가짐
버전 (Version)	8 bits	RIP 버전을 나타냄
계열 (Family)	16 bits	사용된 프로토콜의 계열을 정의함 TCP/IP는 2의 값을 가짐
네트워크 주소 (Network address)	14 bytes	목적지 네트워크의 주소를 정의함 IPv4의 경우 4바이트를 사용하고 나머지를 0으로 채움
거리 (Distance)	32 bits	송신 라우터부터 목적지 네트워크까지의 홉 수를 나타냄



# 거리 벡터 라우팅 (19/27)

## • RIPv1 (2/6)

### • 요청

- 새로 생기거나 시간이 만료된 항목을 가진 라우터에 의해 전송됨
- 특정 항목이나 모든 항목에 대해 문의할 수 있음

### • 응답

- 요청에서 지정된 목적지에 대한 정보를 포함함
- 갱신 패킷
  - 30초 주기로 전송됨
  - 모든 라우팅 테이블을 다루는 정보를 포함함

a. Request for some

Com: 1	Version	Reserved
Family		All 0s
Network address		
All 0s		
All 0s		
All 0s		

Repeated

b. Request for all

Com: 1	Version	Reserved
Family		All 0s
All 0s		
All 0s		
All 0s		

c. Response

Com: 2	Version	Reserved
Family		All 0s
Network address		
All 0s		
All 0s		
Distance		

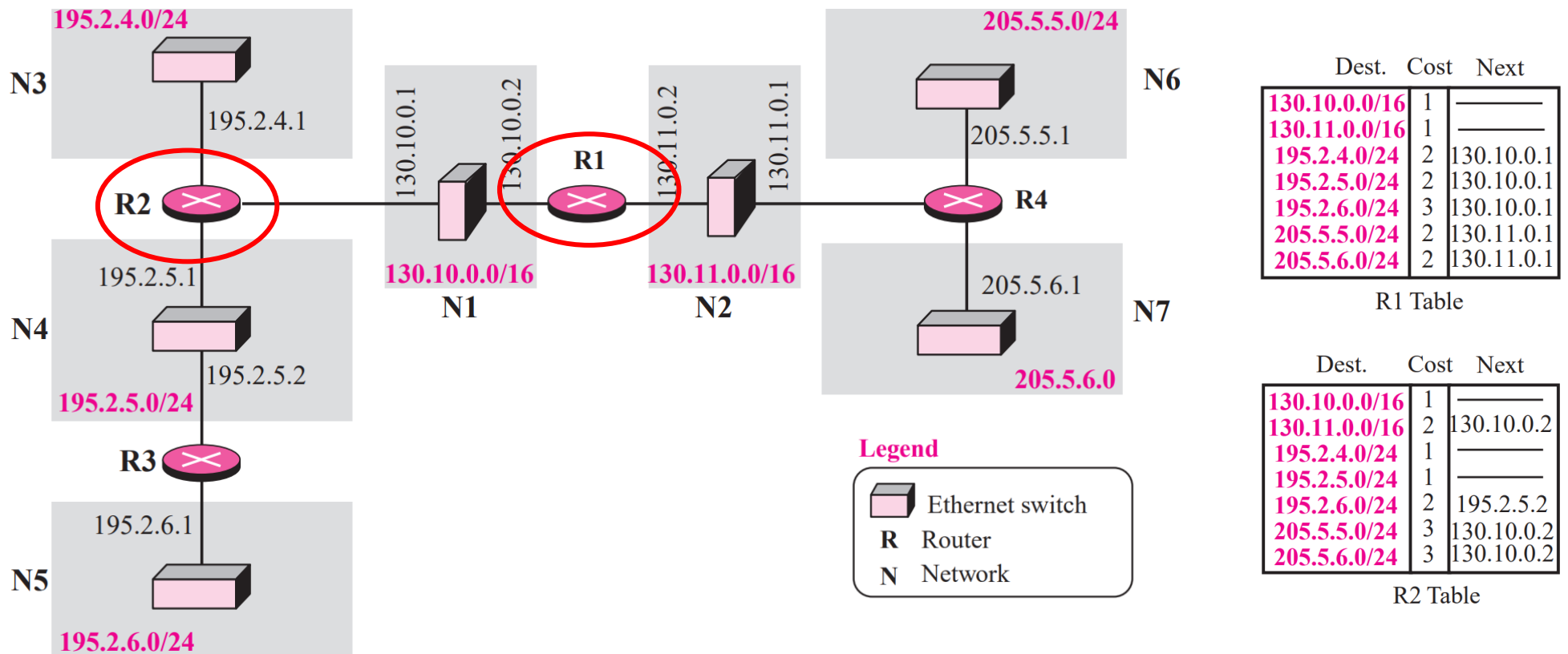
Repeated

# 거리 벡터 라우팅 (20/27)

## • RIPv1 (3/6)

### • 예제 11.4 (1/2)

다음 상황에서 라우터 R1에서 R2로 보내지는 갱신 메시지를 보이고, 이에 대해 설명하여라.



Dest.	Cost	Next
130.10.0.0/16	1	_____
130.11.0.0/16	1	_____
195.2.4.0/24	2	130.10.0.1
195.2.5.0/24	2	130.10.0.1
195.2.6.0/24	3	130.10.0.1
205.5.5.0/24	2	130.11.0.1
205.5.6.0/24	2	130.11.0.1

R1 Table

Dest.	Cost	Next
130.10.0.0/16	1	_____
130.11.0.0/16	2	130.10.0.2
195.2.4.0/24	1	_____
195.2.5.0/24	1	_____
195.2.6.0/24	2	195.2.5.2
205.5.5.0/24	3	130.10.0.2
205.5.6.0/24	3	130.10.0.2

R2 Table

# 거리 벡터 라우팅 (21/27)

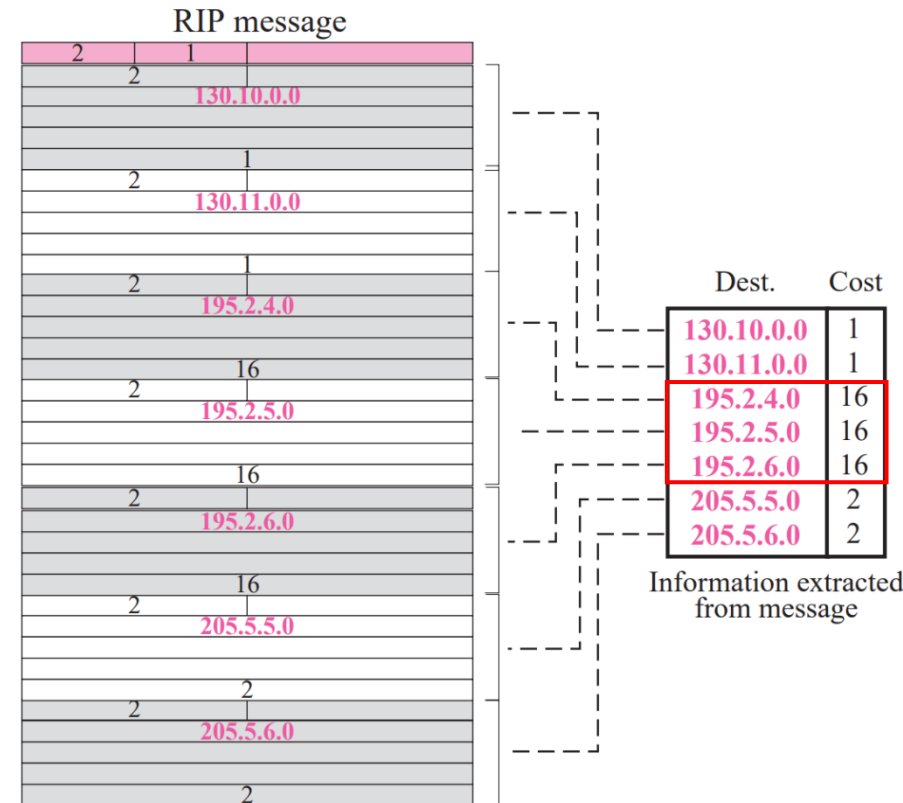
- RIPv1 (4/6)

- 예제 11.4 (2/2)

다음 상황에서 라우터 R1에서 R2로 보내지는 갱신 메시지를 보이고, 이에 대해 설명하여라.

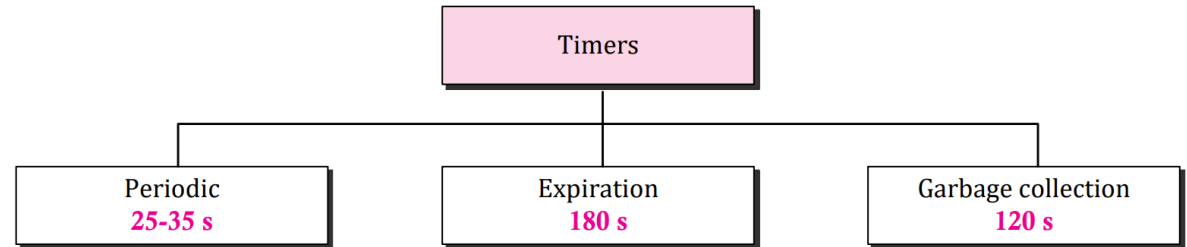
- 풀이

- 수평 분할과 포이즌 리버스정책을 염두에 두고 R1은 195.240.0, 195.2.6.0 정보를 R2로 얻었음에 따라, 무한대 값인 16으로 대체함
    - RIP 메시지를 전송하는 IP 데이터그램의 송신자 주소를 다음 홉 주소로 사용함
    - 응답 데이터는 R1을 기준으로 하는 값이므로, R2는 각 홉 수를 1씩 증가시켜서 계산함



# 거리 벡터 라우팅 (22/27)

- RIPv1 (5/6)



- 타이머

- 주기적 타이머

- 정규 갱신 메시지의 통보를 제어함
    - 라우터들이 동시에 갱신하고자 하는 경우, 발생할 수 있는 과부하 상태를 막기 위해서 사용됨

- 만료 타이머

- 경로의 유효성을 관리함
    - 경로에 대한 갱신 메시지를 수신하면, 타이머를 180초로 지정함
    - 갱신 메시지가 할당된 180초 이내 도착하지 못하면, 경로가 해제되어야 한다고 판단하고 홉 수를 16으로 설정함
    - 각 경로가 만료 타이머를 가짐

- 폐 경로 수집 타이머

- 삭제 전에 이웃들에게 경로가 유효하지 않음을 알리기 위해 사용함
    - 경로에 대한 정보가 무효화되면 즉각적으로 삭제되지 않고 이 값이 0이 되면 테이블에서 삭제됨

# 거리 벡터 라우팅 (23/27)

- RIPv1 (6/6)

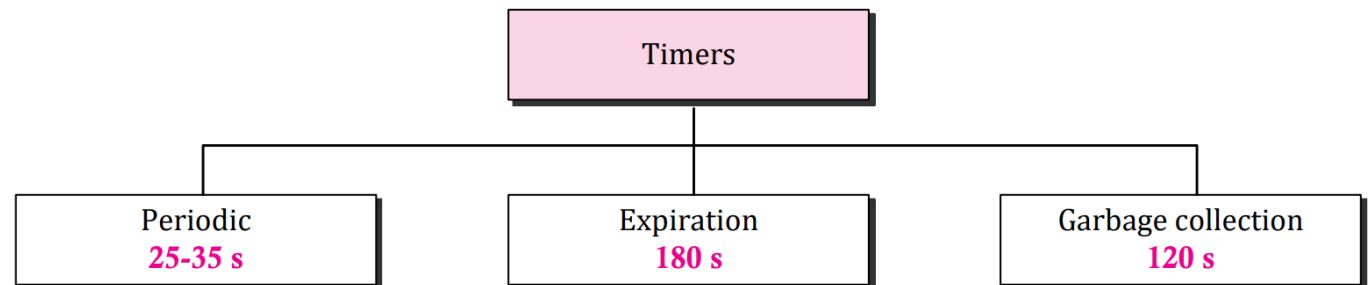
- 예제 11.5

라우팅 테이블이 20개 항목을 가지고 있다. 이 라우터가 200초 동안 5개 경로에 대한 정보를 받지 못했다고 할 때, 이 시간에 가동되고 있는 타이머는 몇 개인가?

- 풀이

- 주기적 타이머: 1개
    - 만료 타이머:  $20 - 5 = 15$ 개
    - 폐 경로 수집 타이머 5개

∴ 총 21개의 타이머가 가동됨

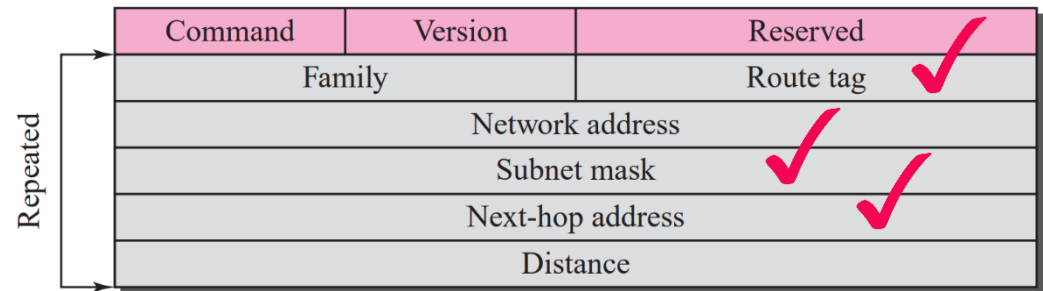
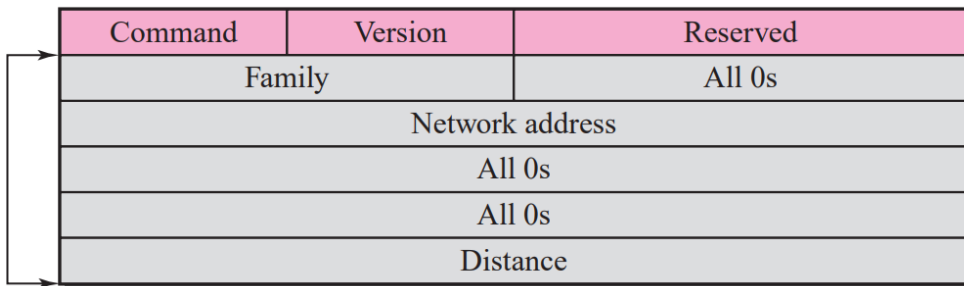


# 거리 벡터 라우팅 (24/27)

- RIPv2 (1/3)
- 새로운 필드

\*CIDR: IP 주소를 유연하게 할당하기 위해, 네트워크 주소와 서브넷 마스크를 슬래시 (/) 표기법으로 나타내는 방식

필드명	설명
경로 태그 (Route tag)	자율 시스템 번호와 같은 정보를 전달함 도메인 간 라우팅 프로토콜로부터의 정보 수신을 가능하게 함
서브넷 마스크 (Subnet mask)	서브넷 마스크를 전달함 RIPv2가 CIDR*을 지원함을 의미함
다음 홉 주소 (Next-ho address)	다음 홉의 주소를 알려줌 두 자율 시스템이 백 본 네트워크와 같은 네트워크를 공유할 때에 유용하게 사용됨



# 거리 벡터 라우팅 (25/27)

- RIPv2 (2/3)

- RIPv1과의 주요 차이점 (1/2)

- 클래스 없는 주소지정

- RIPv1은 클래스 기반 주소지정을 따르지만, RIPv2는 서브넷 마스크를 위해 필드를 추가함으로써 RIPv2에서 클래스 없는 주소 지정 (Classless addressing)을 사용할 수 있음

- 인증

- 메시지의 첫 항목을 인증 정보를 위해 사용함

- 경로 정보가 아닌 인증 정보임을 알리기 위해 계열 필드를 모두 1로 채움
      - 인증 유형 (Authentication type)은 인증에 사용되는 방법을 나타냄
      - 세 번째 필드는 실제 인증 데이터를 포함함

Command	Version	Reserved
0xFFFF		Authentication type
Authentication data 16 bytes		
⋮		

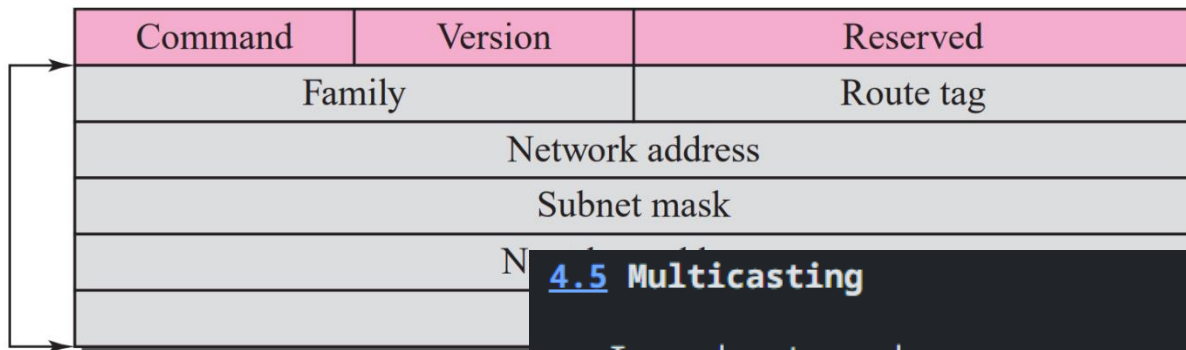
# 거리 벡터 라우팅 (26/27)

- RIPv2 (3/3)

- RIPv1과의 주요 차이점 (2/2)

- 멀티 캐스팅

- RIPv1에서는 모든 이웃에게 메시지를 전송하기 위해 브로드캐스팅을 사용함
  - 호스트도 패킷을 수신한다는 문제점이 존재함
- RIPv2는 네트워크에 있는 라우터들만 RIP 메시지를 수신할 수 있도록 모든 라우터 멀티캐스트 주소 (All-router multicast address) 를 사용함



## 4.5 Multicasting

In order to reduce unnecessary load on those hosts which are not listening to RIP-2 messages, an IP multicast address will be used for periodic broadcasts. The IP multicast address is 224.0.0.9. Note that IGMP is not needed since these are inter-router messages which are not forwarded.

# 거리 벡터 라우팅 (27/27)

---

- RIP 캡슐화

- UDP 데이터그램에 캡슐화 됨

- RIP는 메시지의 길이를 나타내는 필드를 가지지 않고, 이를 UDP에서 결정함
- RIP에 잘 알려진 포트는 UDP 520번 포트를 사용함



# 목 차

---

- 자율 시스템
- 거리 벡터 라우팅
- 링크 상태 라우팅
- 경로 벡터 라우팅

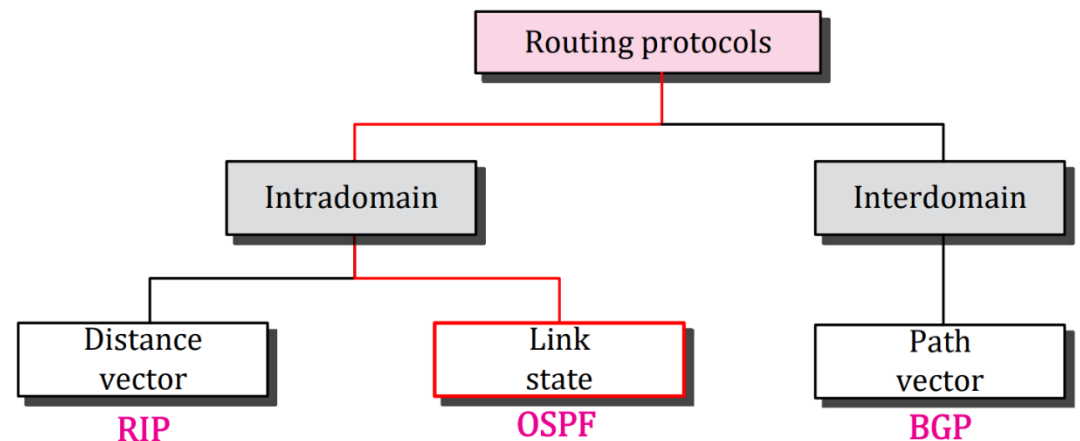
# 링크 상태 라우팅 (1/47)

- 정의

- 링크 상태 정보를 모든 라우터에 전달하여 최단 경로 트리를 구성하는 라우팅 방식

- 특징

- 대규모 네트워크에 적합함
- Dijkstra 알고리즘을 사용함
- OSPF (Open Shortest Path First) 프로토콜로 구현됨



# 링크 상태 라우팅 (2/467)

---

- 라우팅 테이블 생성 (1/7)

- 절차

1. 각 노드에 의해 링크 상태를 생성함
  - 생성된 정보를 링크 상태 패킷 (LSP, Link State Packet)이라 함
2. 모든 다른 라우터로 효과적이며 안전한 방법으로 LSP를 발송함
  - 발송하는 것을 플러딩 (Flooding)이라 함
3. 각 노드에서 최단 경로 트리를 생성함
4. 최단 경로 트리에 기반하여, 라우팅 테이블을 계산함

# 링크 상태 라우팅 (3/47)

---

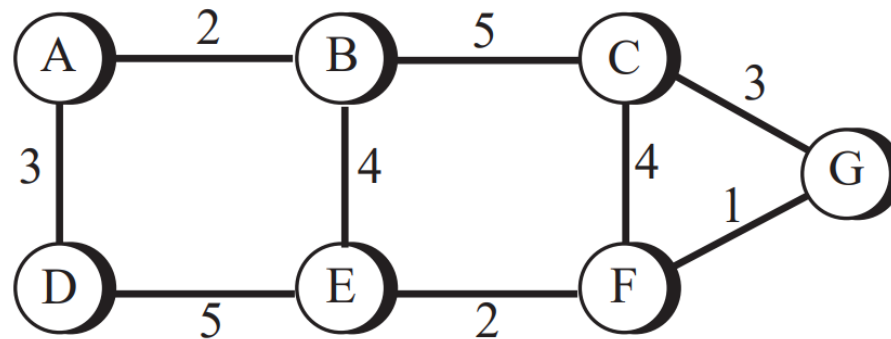
- 라우팅 테이블 생성 (2/7)
  - 절차: LSP 생성
    - LSP가 생성되는 경우
      - 도메인의 토폴로지에 변화가 있는 경우
        - LSP 발송을 시작하는 것이 도메인 내의 임의의 노드에 토폴로지를 갱신하도록 빠르게 알리는 주요 방법임
      - 주기적으로 생성하는 경우
        - 오래된 정보가 제거되도록 보장하기 위해 수행됨
        - 타이머는 구현에 따라 60분에서 2시간 범위로 설정됨

# 링크 상태 라우팅 (4/47)

## • 라우팅 테이블 생성 (3/7)

### • 절차: LSP의 플러딩

- LSP를 생성한 노드는 각 인터페이스를 통해 LSP의 복사본을 전송함
- LSP를 수신한 노드는 이전에 받은 복사본과 비교함
  - 수신한 LSP가 이전 것에 비해 오래된 것이라면, 해당 LSP를 버림
  - 수신한 LSP가 이전 것에 비해 새것이면 다음을 수행함
    - 오래된 LSP를 버리고 새 것을 가짐
    - 복사본을 수신한 인터페이스를 제외한 다른 인터페이스로 전송함
      - 플러딩이 도메인 내에서 중지됨을 보장함 (하나의 인터페이스만 가진 노드로 인해)



Topology



# 링크 상태 라우팅 (6/47)

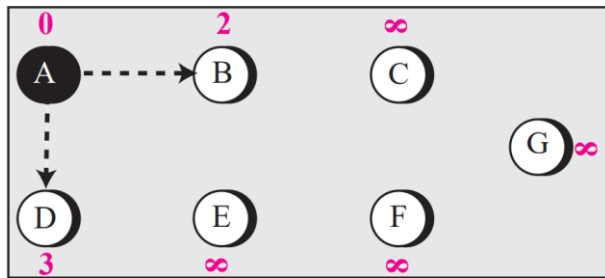
- 라우팅 테이블 생성 (5/7)
  - 절차: 최단 경로 트리 생성 (Dijkstra 알고리즘) (2/3)
    - 의사 코드
      - 남아있는 노드 중에서 현재 가장 짧은 거리  $D[i]$ 를 가지는 노드를 경로에 추가함
      - $i$ 를 선택함으로써 변동 가능한 나머지 노드들에 대한 비용을 갱신함

```
1 Dijkstra ()
2 {
3     // Initialization
4     Path = {s}           // s means self
5     for (i = 1 to N)
6     {
7         if (i is a neighbor of s and i ≠ s) D[i] = c[si]
8         if (i is not a neighbor of s)      D[i] = ∞
9     }
10    D_s = 0
11 } // Dijkstra
12
13 // Iteration
14 Repeat
15 {
16     // Finding the next node to be added
17     if (D[i] is minimum among all remaining nodes)
18         Path = Path u i
19
20     // Update the shortest distance for the rest
21     for (j = 1 to M) // M number of remaining nodes
22     {
23         D[j] = min (D[j], D[i] + c[i][j])
24     }
25 } until (all nodes included in the path, M = 0)
```

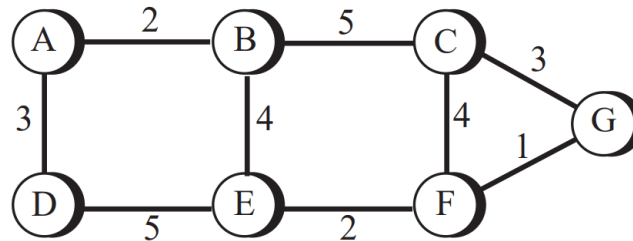
# 링크 상태 라우팅 (7/47)

## • 라우팅 테이블 생성 (6/7)

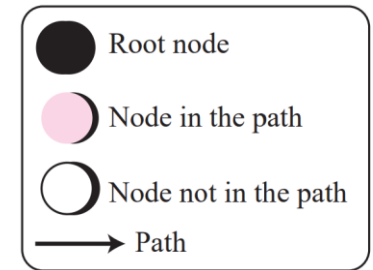
### • 절차: 최단 경로 트리 생성 (Dijkstra 알고리즘) (3/3)



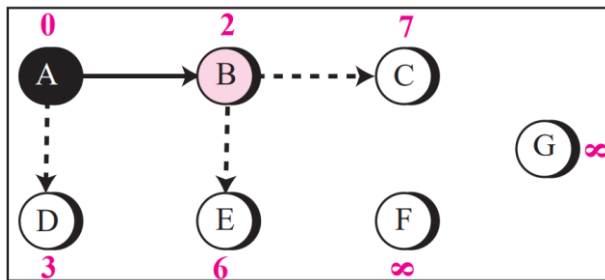
Initialization



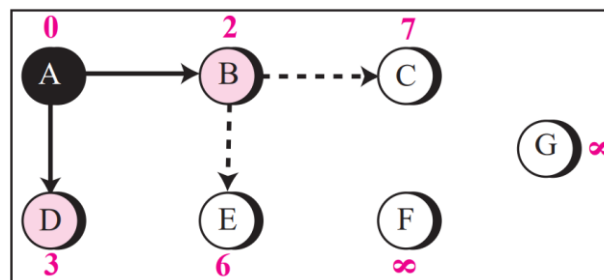
Topology



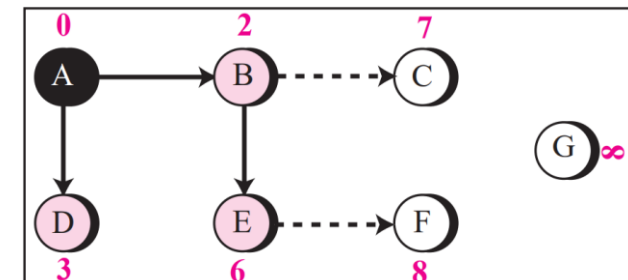
Legend



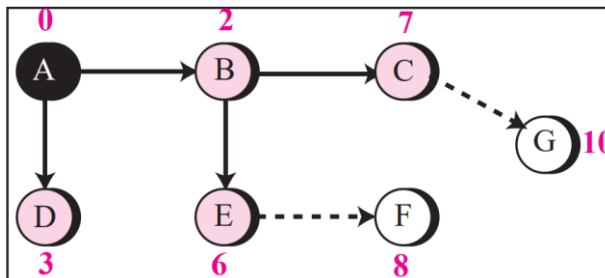
Iteration 1



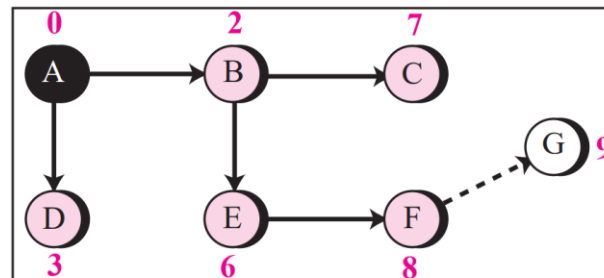
Iteration 2



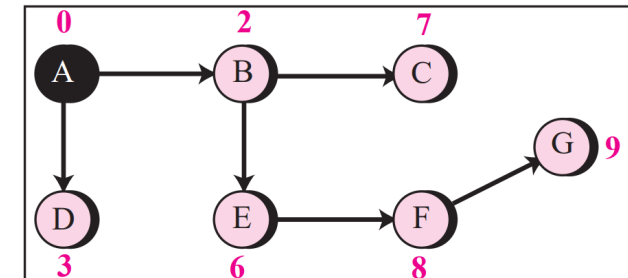
Iteration 3



Iteration 4



Iteration 5



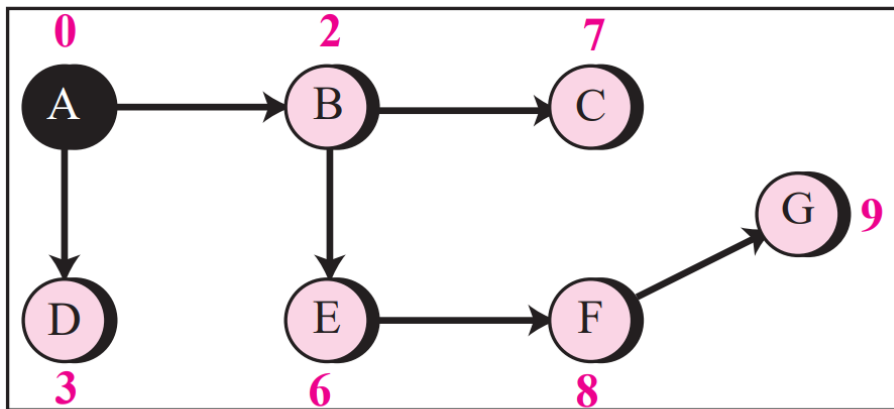
Iteration 6

# 링크 상태 라우팅 (8/47)

- 라우팅 테이블 생성 (7/7)

- 절차: 라우팅 테이블 계산

- 최단 경로 트리 프로토콜을 사용하여 라우팅 테이블을 구성함



Iteration 6

Destination	Cost	Next Router
A	0	-
B	2	-
C	7	B
D	3	-
E	6	B
F	8	B
G	9	B

# 링크 상태 라우팅 (9/47)

---

- OSPF (Open Shortest Path First) (1/39)
  - 정의
    - 링크 상태 라우팅에 근거를 둔 도메인 내 라우팅 프로토콜
  - 특징
    - 라우팅을 적절히 수행하기 위해 자율 시스템을 여러 지역 (Areas)으로 나눔
    - 관리자가 각 경로에 대한 비용을 할당할 수 있음
    - 4가지 유형의 링크가 정의됨
    - 5가지의 서로 다른 패킷 종류를 사용함

# 링크 상태 라우팅 (10/47)

## • OSPF (2/39)

\*백본 (Backbone): 대규모 네트워크의 중심이 되는 고속 통신망

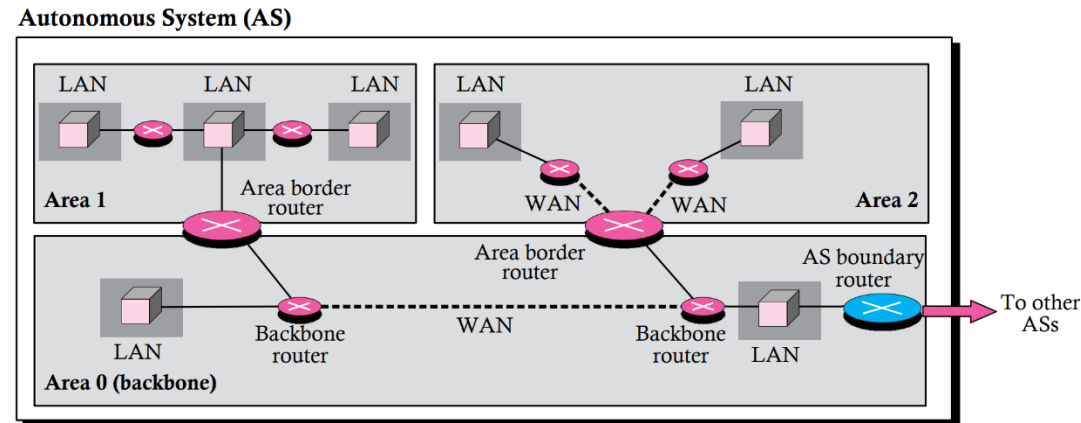
### • 지역

#### • 정의

- 자율 시스템에 포함되는 호스트, 라우터, 네트워크의 모음

#### • 특징

- 하나의 자율 시스템은 여러 개의 다른 지역들로 나뉠 수 있음
- 지역 내 모든 네트워크는 연결되어야만 함
- 지역 내 라우터는 경로 정보를 지역 내로 플러딩함
- 지역 경계 라우터 (Area border router)는 지역에 관한 정보를 요약하여 요약된 정보를 다른 지역으로 전송함
- 자율 시스템 내의 모든 지역은 백본\*에 연결되어야만 함
  - 백본이 1차 지역, 다른 지역이 2차 지역이 됨
- 백본 내 라우터는 백본 라우터라고 지칭함



# 링크 상태 라우팅 (11/47)

---

- OSPF (3/39)

- 메트릭

- 서비스 종류에 기반을 두고 결정됨

- 최소 지연, 최대 성능 등

- 라우터는 서로 다른 서비스 종류에 기반을 두어, 복수 개의 라우팅 테이블을 가질 수 있음

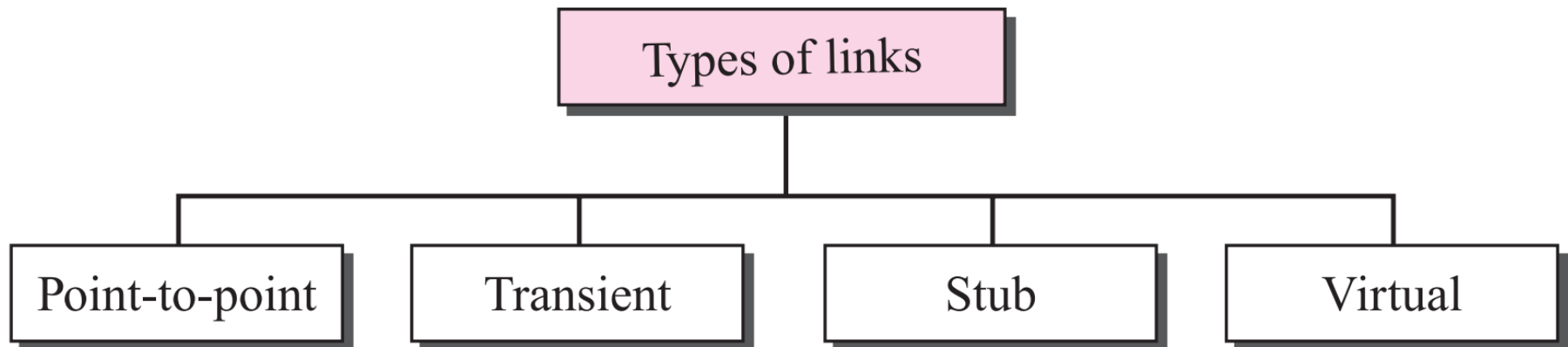
# 링크 상태 라우팅 (12/47)

---

- OSPF (4/39)

- 링크 유형

- 점-대-점 (Point-to-point)
- 경유 (Transient)
- 스템브 (Stub)
- 가상 (Virtual)



# 링크 상태 라우팅 (13/47)

## • OSPF (5/39)

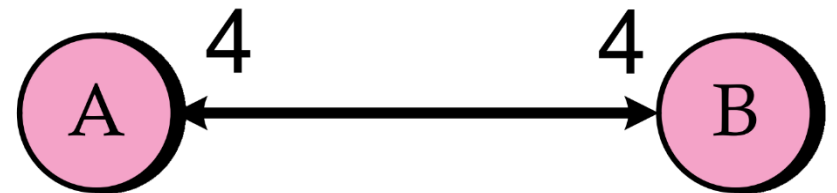
### • 링크 유형: 점-대-점 링크

#### • 정의

- 사이에 어떤 다른 호스트나 라우터 없이 라우터 간에 연결하는 것

#### • 특징

- 라우터 간을 연결하는 것이 목적임
  - e.g., 전화선이나 T-라인을 통해 연결된 두 라우터
- 네트워크 주소를 할당할 필요가 없음
- 라우터는 노드로 표현되고 링크는 노드 간을 연결하는 양방향 화살표로 표현됨
- 같은 값을 가지는 메트릭이 양쪽 노드 끝 점에 각각의 방향마다 표현됨



# 링크 상태 라우팅 (14/47)

- OSPF (6/39)

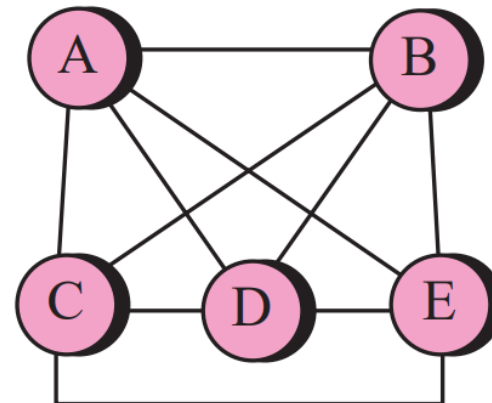
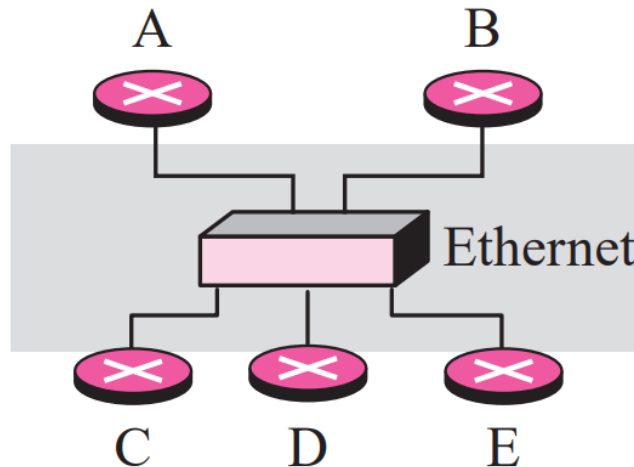
- 링크 유형: 경유 링크 (1/2)

- 정의

- 여러 개의 라우터가 연결되어 있는 네트워크

- 특징

- 데이터는 어떤 라우터로도 들어갈 수 있고 어떤 라우터로부터도 나갈 수 있음
    - 모든 LAN과 2개 이상의 라우터를 가지는 일부 WAN이 경유 링크에 해당함



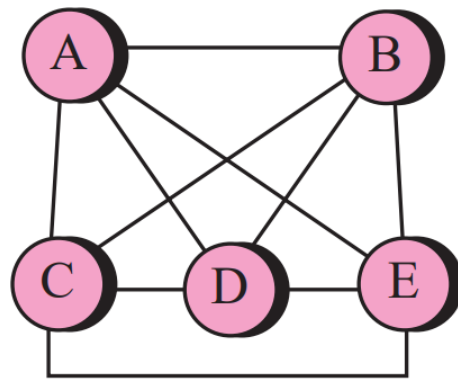
# 링크 상태 라우팅 (15/47)

- OSPF (7/39)

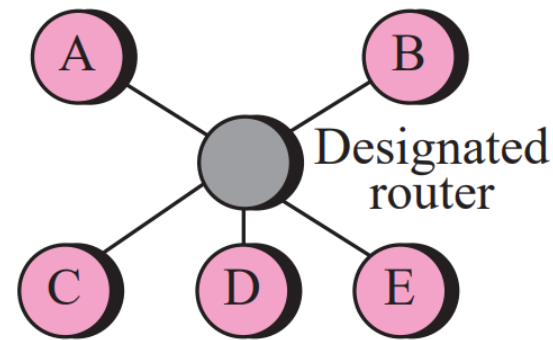
- 링크 유형: 경유 링크 (2/2)

- 효율성과 현실성

- 자신과 이웃하는 4개 라우터들에게 자신의 이웃을 알리려면 총 20개의 통보가 필요하고, 실제로는 각 쌍 간에 단일 네트워크가 존재하지 않음
    - 각 라우터가 하나의 단일 네트워크를 통해 다른 라우터들과 연결되어있다는 것을 보이기 위해, 네트워크 자체가 단일 노드로 표현되어야 함
    - 네트워크에 한 라우터가 지정 (Designated) 라우터로서 책임을 맡음



a. Unrealistic



b. Realistic

# 링크 상태 라우팅 (16/47)

- OSPF (8/39)

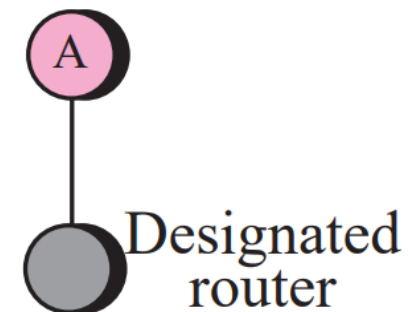
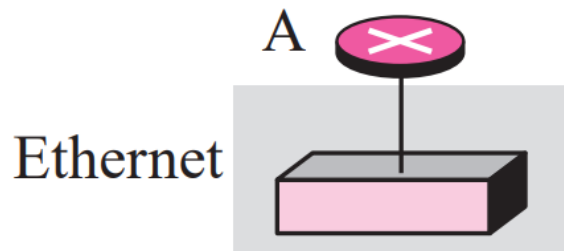
- 링크 유형: 스템브 링크

- 정의

- 하나의 라우터에만 연결된 네트워크

- 특징

- 데이터 패킷은 단일 라우터를 통해서 네트워크에 들어가거나 나갈 수 있음
    - 특별한 형태의 경유 링크로 분류될 수 있음
    - 라우터를 노드로, 네트워크를 지정 라우터로 표현함



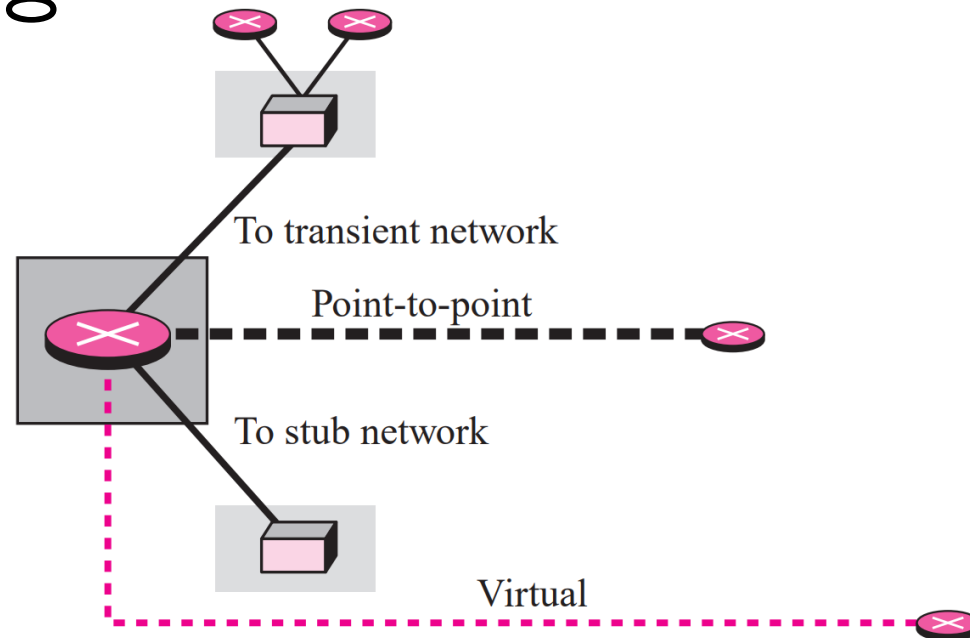
# 링크 상태 라우팅 (17/47)

- OSPF (9/39)

- 링크 유형: 가상 링크

- 두 라우터간의 연결이 끊어지면, 여러 라우터를 거쳐 돌아가는 더 긴 경로를 사용하더라도 둘 사이에 가상 링크를 연결해야 함

- 네 가지 링크 유형

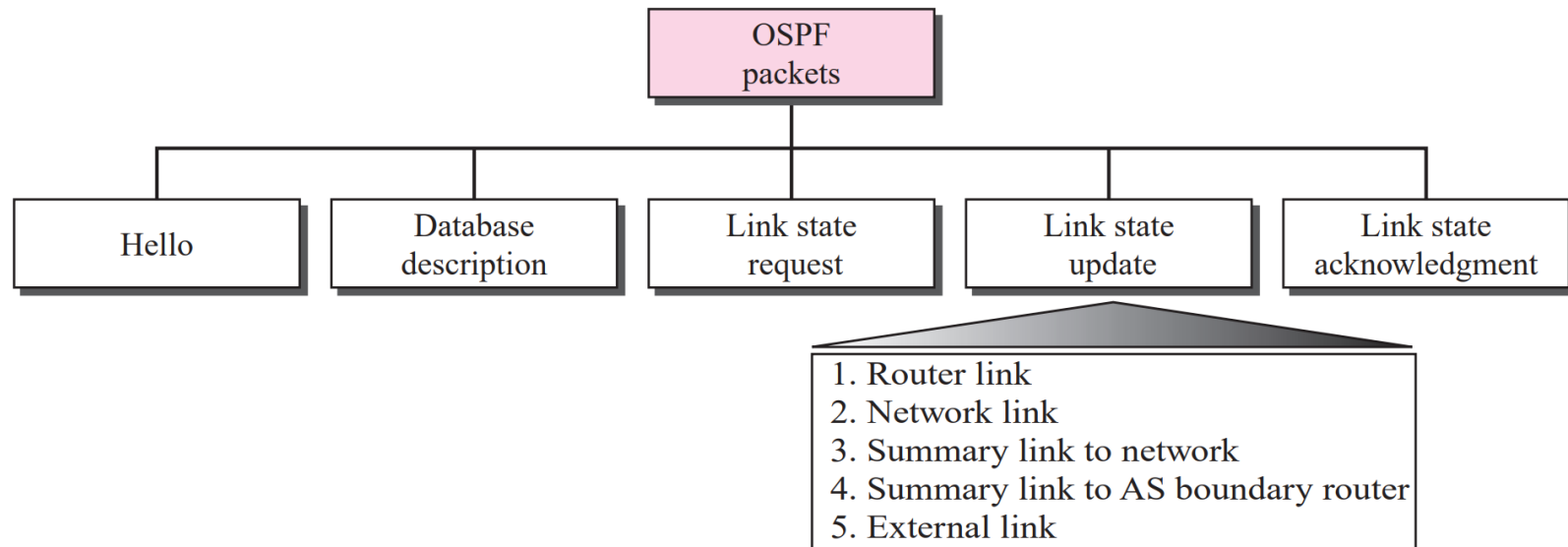


# 링크 상태 라우팅 (18/47)

- OSPF (10/39)

- 패킷 유형

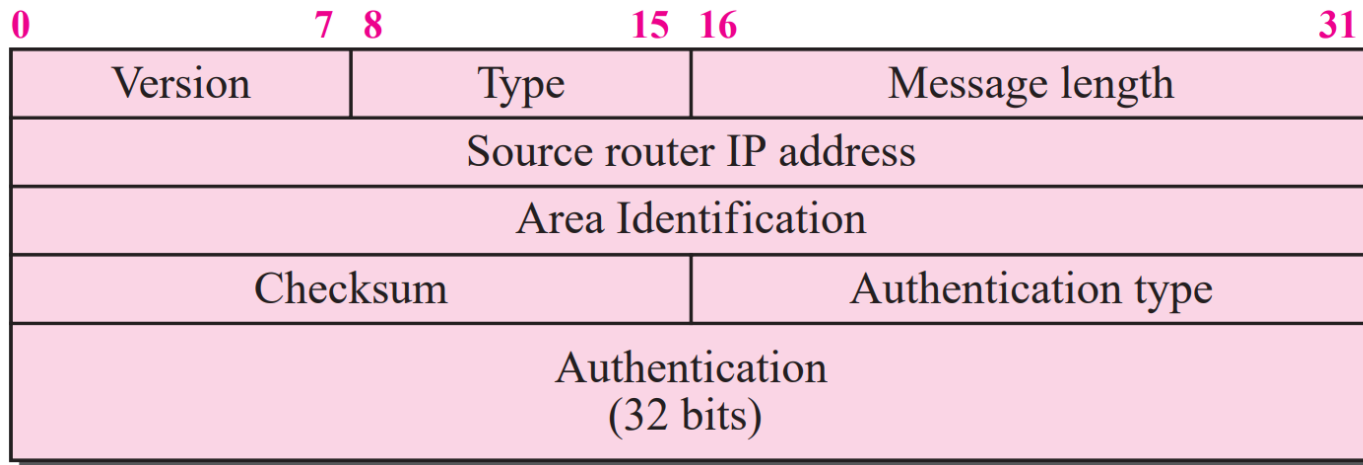
- Hello
- 데이터베이스 기술 (Description)
- 링크 상태 요청
- 링크 상태 갱신
- 링크 상태 확인 응답



# 링크 상태 라우팅 (19/47)

- OSPF (11/39)
  - 공통 헤더 (1/2)

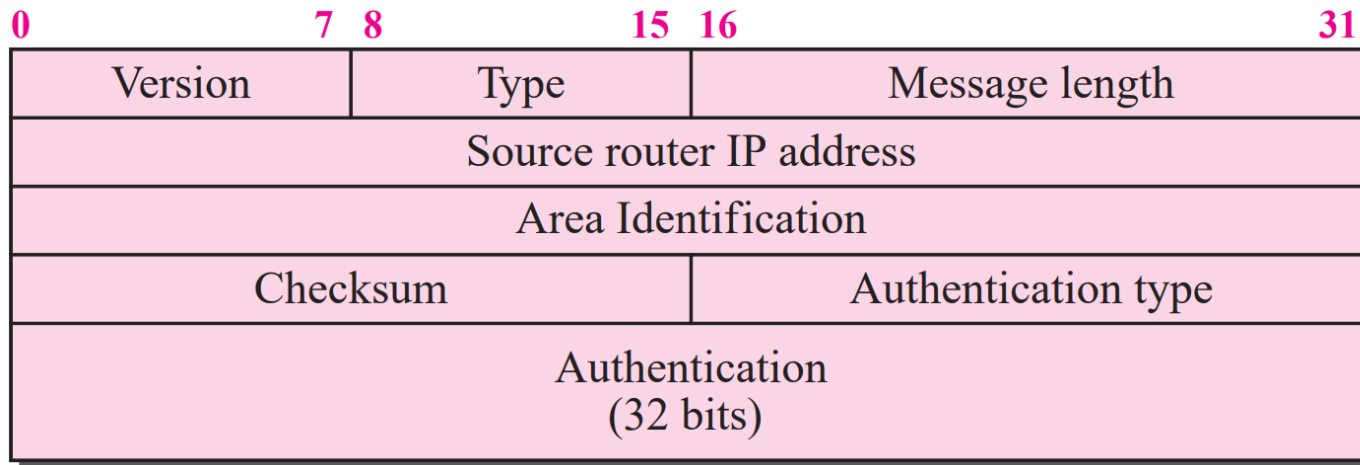
필드명	크기 (bit)	설명
버전 (Version)	8	OSPF 프로토콜 버전을 정의함
유형 (Type)	8	1에서 5의 값으로 패킷의 유형을 나타냄
메시지 길이 (Message length)	16	헤더를 포함한 전체 메시지의 길이를 나타냄
발신지 라우터 IP 주소 (Source router IP address)	32	패킷을 보낸 라우터의 IP 주소를 나타냄
지역 식별자 (Area Identification)	32	라우팅이 일어나는 지역을 나타냄



# 링크 상태 라우팅 (20/47)

- OSPF (12/39)
  - 공통 헤더 (2/2)

필드명	크기 (bit)	설명
검사합 (Checksum)	16	인증 종류와 인증 데이터를 제외한 전체 패킷의 오류 검색에 사용됨
인증 종류 (Authentication Type)	16	지역에서 사용되는 인증 방법을 정의함 아무것도 사용하지 않을 때, 0의 값을 가짐 패스워드를 사용할 때, 1의 값을 가짐
인증 (Authentication)	64	인증 데이터의 실제 값임 인증 종류가 0이면 0으로 채워짐 인증 종류가 1이면 8 문자의 패스워드가 포함됨



# 링크 상태 라우팅 (21/47)

---

- OSPF (13/39)

- Hello 패킷 (1/3)

- 역할

- 이웃 관계를 생성하고 이웃의 도달 가능성을 검사함

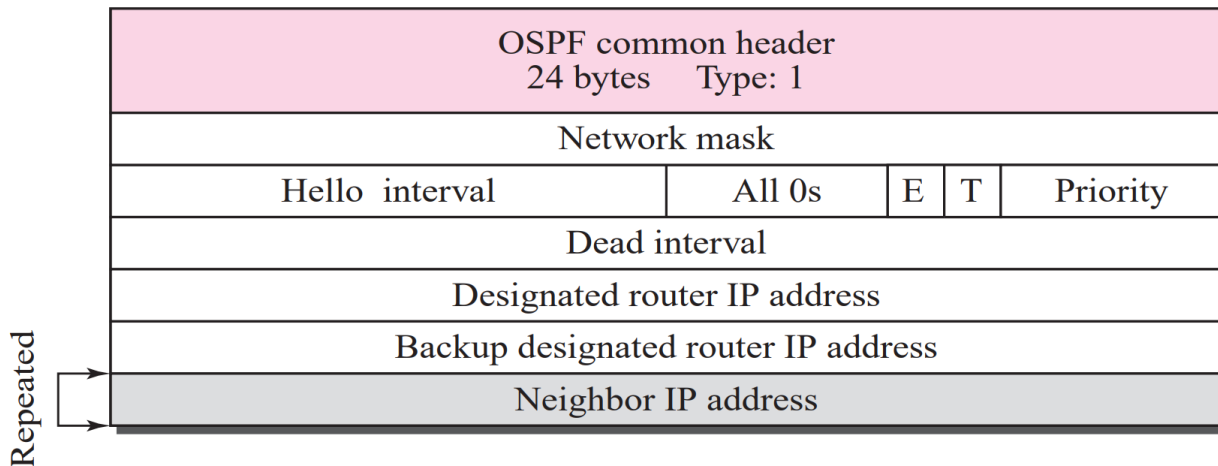
- 특징

- 링크 상태 라우팅의 첫 번째 과정임
      - 라우터가 자신의 이웃에 대한 정보를 다른 라우터로 전송하기 전에 먼저 자신의 이웃과 패킷을 교환 해야함
      - 이웃이 살아있으면 라우터는 이웃에 도달할 수 있는 지 알 수 있음

# 링크 상태 라우팅 (22/47)

- OSPF (14/39)
  - Hello 패킷 (2/3)
    - 형식 (1/2)

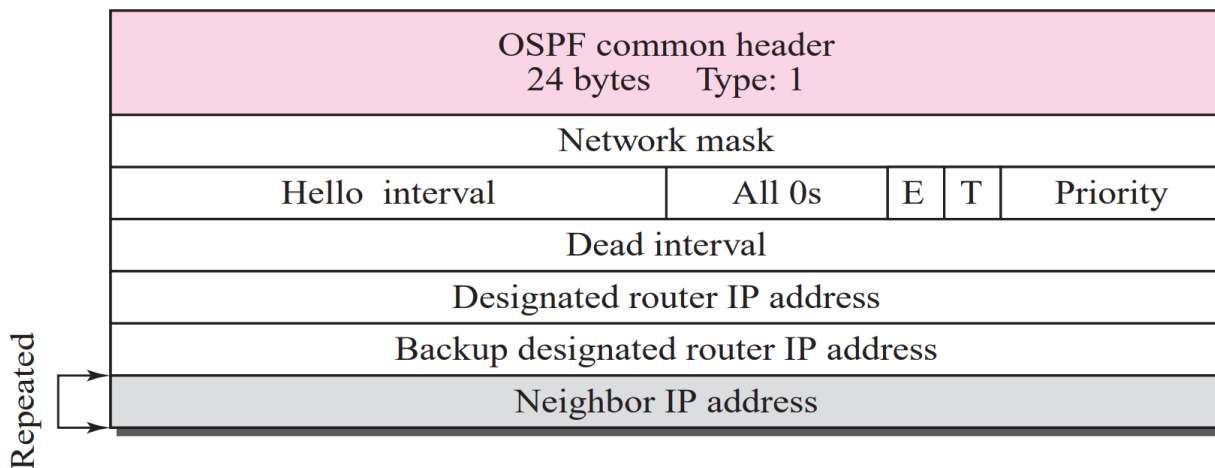
필드명	크기 (bit)	설명
네트워크 마스크 (Network mask)	32	hello 메시지가 전송되어지는 네트워크의 네트워크 마스크를 나타냄
Hello 간격 (interval)	16	hello 메시지 간의 초 단위 시간을 나타냄
E 플래그 (E flag)	1	1로 설정되어있으면 지역이 스텔트 지역이 됨
T 플래그 (T flag)	1	1로 설정되면, 라우터는 다중 메트릭을 지원함
우선순위 (Priority)	8	라우터의 우선순위를 나타냄 우선순위는 지정 라우터와 백업 지정 라우터를 결정함
Dead 간격 (Dead interval)	32	이웃이 다운되었다고 라우터가 가정하기 전까지의 시간을 초 단위로 나타냄



# 링크 상태 라우팅 (23/47)

- OSPF (15/39)
  - Hello 패킷 (3/3)
    - 형식 (2/2)

필드명	크기 (bit)	설명
지정 라우터 IP 주소 (Designated router IP address)	32	메시지가 전송되는 네트워크에 대한 지정 라우터의 IP 주소를 나타냄
백업 지정 라우터 IP 주소 (Backup designated router IP address)	32	메시지가 전송되는 네트워크에 대한 백업 지정 라우터의 IP 주소를 나타냄
이웃 IP 주소 (Neighbor IP address)	32	Dead 간격 내에 Hello 패킷을 송신한 모든 이웃 라우터들의 IP 주소 목록을 나타냄



# 링크 상태 라우팅 (24/47)

---

- OSPF (16/39)

- 데이터베이스 기술 패킷 (1/3)

- 역할

- 링크 상태 데이터베이스의 요약 정보를 전송함

- 특징

- 시스템에 연결된 라우터의 첫 번째 hello 메시지를 받은 이웃은 데이터베이스 기술 패킷을 보냄
    - 두 라우터가 데이터베이스 기술 메시지를 교환하기 원하면 하나가 주 (Master)가 되고 다른 하나는 종 (Slave)이 됨
    - 메시지가 길면, 데이터베이스 내용이 몇 개의 메시지로 나뉠 수 있음

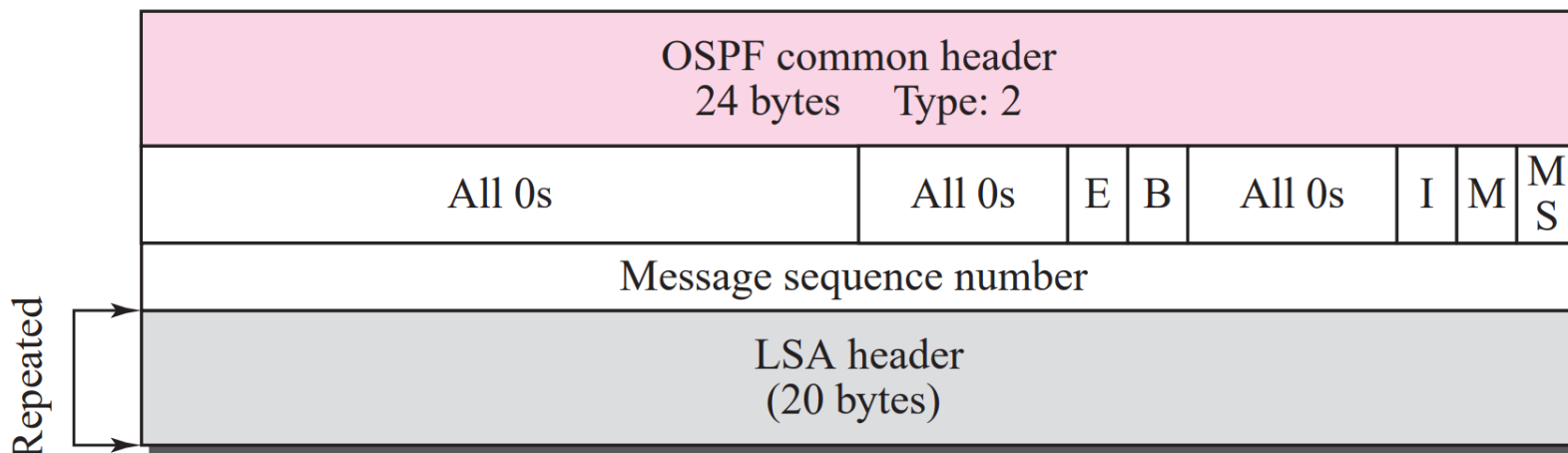
# 링크 상태 라우팅 (25/47)

- OSPF (17/39)

- 데이터베이스 기술 패킷 (2/3)

- 형식

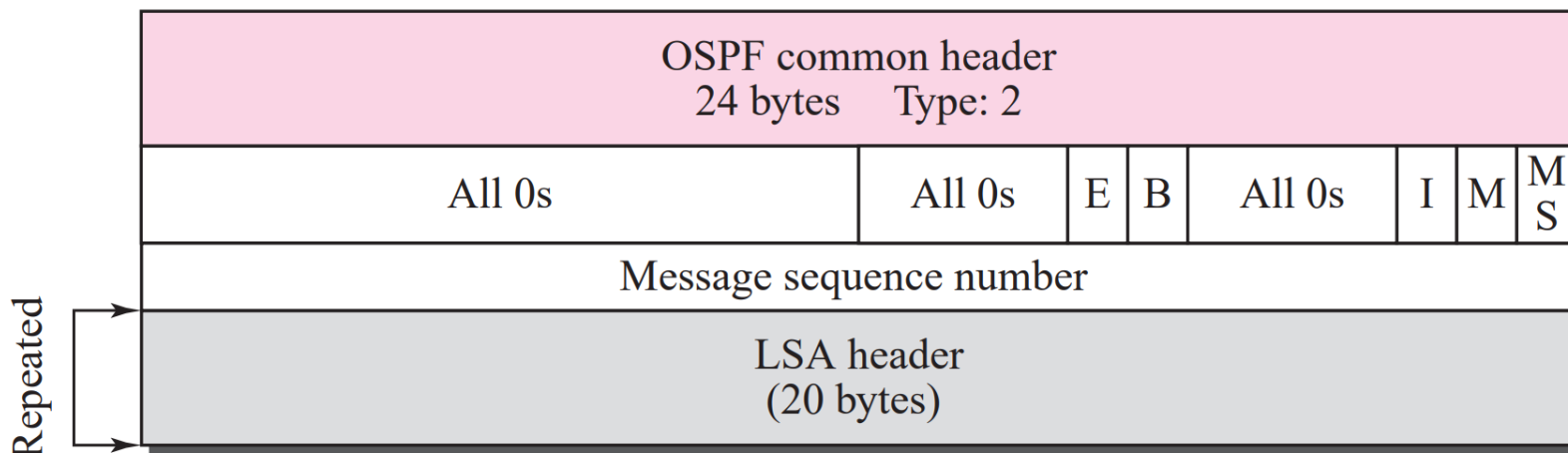
필드명	크기	설명
E(External) 플래그	1bit	광고 라우터가 자율 시스템 경계 라우터이면 1로 지정됨
B(Boundary) 플래그	1bit	광고 라우터가 지역 경계 라우터이면 1로 지정됨
I(Initialization) 플래그	1bit	메시지가 첫 메시지일 때 1로 지정됨
M(More) 플래그	1bit	마지막 메시지가 아닐 경우 1로 지정됨
M/S(Master/Slave) 플래그	1bit	패킷의 시작점을 나타냄 1이면 Master, 0이면 Slave임



# 링크 상태 라우팅 (26/47)

- OSPF (18/39)
  - 데이터베이스 기술 패킷 (3/3)
    - 형식

필드명	크기	설명
메시지 순서 번호	32bits	메시지의 순서 번호를 포함하여, 요청에 대한 응답 비교에 사용됨
LSA 헤더	20bytes	각 LSA에 사용됨 각 링크의 윤곽을 알려주며, 링크 상태 데이터베이스의 각 링크에 대해 반복되어짐



# 링크 상태 라우팅 (27/47)

- OSPF (19/39)

- 링크 상태 요청 패킷

- 역할

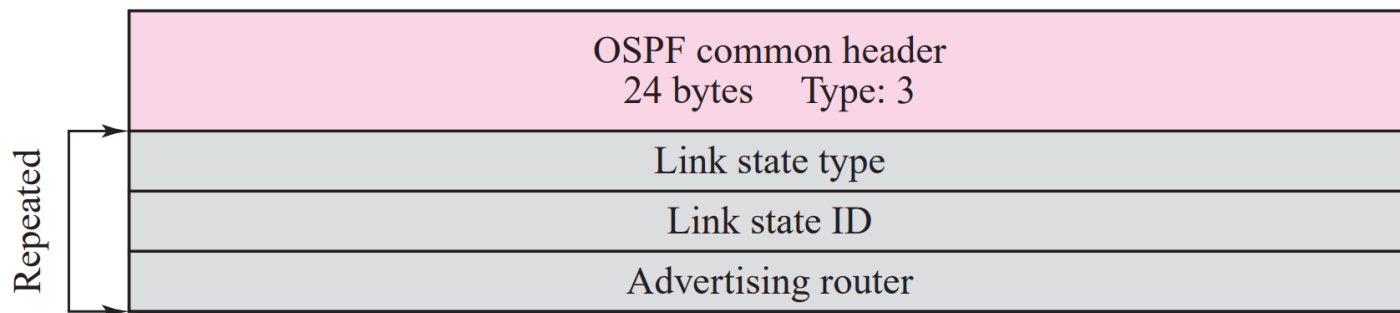
- 특정 경로나 경로들에 대한 정보를 요청함

- 특징

- 링크 상태 갱신 패킷으로 응답됨
    - 새로 연결한 라우터가 데이터베이스 기술 패킷을 수신한 후 몇 개의 경로들에 대한 더 많은 정보를 요청할 때 사용됨

- 형식

- 세 개의 필드로 이루어진 각 세트는 하나의 단일 LSA에 대한 요청임
    - 세트는 하나 이상의 광고가 필요한 경우 반복되어짐



# 링크 상태 라우팅 (28/47)

- OSPF (20/39)

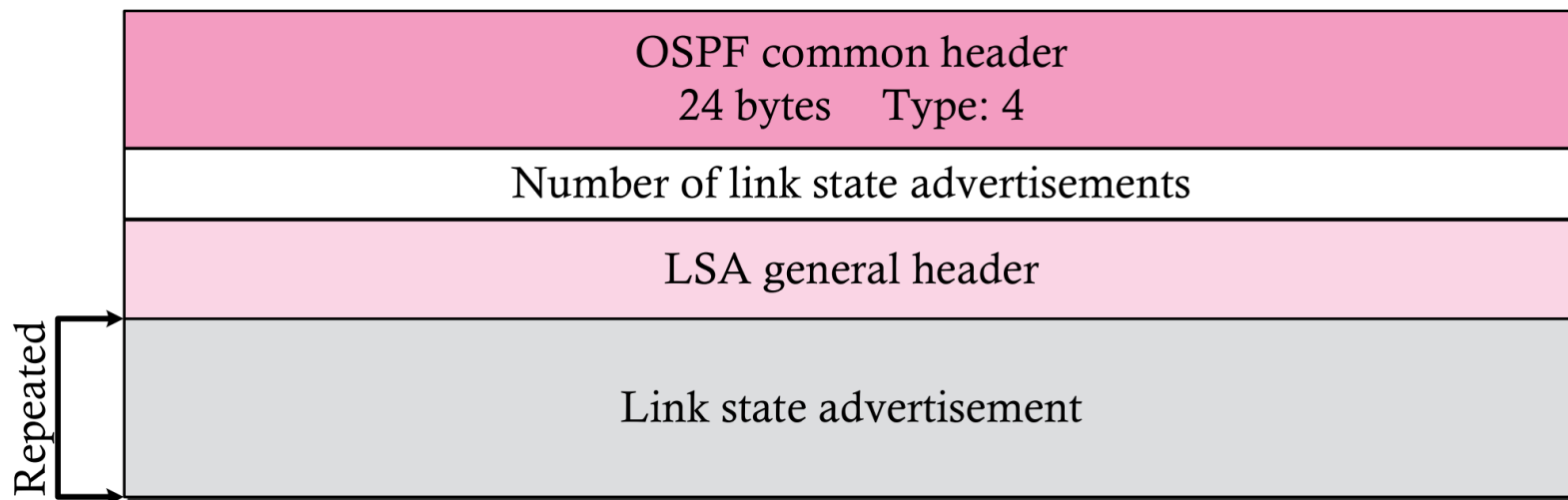
- 링크 상태 갱신 패킷 (1/4)

- 목적

- 라우터가 링크의 상태를 광고하는데 사용됨

- 특징

- 각 갱신 패킷이 몇 개의 서로 다른 LSA (Link State Advertisement)를 포함할 수 있음
    - 모든 5개의 LSA 패킷들은 공통의 일반적인 헤더를 가짐



# 링크 상태 라우팅 (29/47)

- OSPF (21/39)
  - 링크 상태 갱신 패킷 (2/4)
    - 공통 헤더 (1/3)

필드명	설명
링크 상태 시간 (Link state age)	메시지가 처음 만들어지고 난 후부터 경과된 시간을 초 단위로 나타냄 초기 값은 0임 다른 라우터가 메시지를 이어서 전송하게 되면 경유 시간을 추정하고 필드 값에 더함
E 플래그 (E flag)	지역이 스텔브 지역 (하나의 경로에 의해 백본에 연결된 지역)일 경우, 1로 지정됨
T 플래그 (T flag)	라우터가 복수 개의 서비스 유형을 처리할 수 있으면, 1로 지정됨

Link state age	Reserved	E	T	Link state type
Link state ID				
Advertising router				
Link state sequence number				
Link state checksum	Length			

# 링크 상태 라우팅 (30/47)

- OSPF (22/39)
  - 링크 상태 갱신 패킷 (3/4)
    - 공통 헤더 (2/3)

필드명	설명
링크 상태 유형 (Link state type)	LSA 유형을 나타냄 라우터 링크 (1), 네트워크 링크 (2), 네트워크의 요약 링크 (3), 자율 시스템 경계 라우터의 요약 링크 (4), 외부 링크 (5)
링크 상태 식별자 (Link state ID)	링크의 유형에 의존하여 결정됨 유형 1: 라우터의 IP 주소 유형 2: 지정 라우터의 IP 주소 유형 3: 네트워크의 주소 유형 4: 자율 시스템 경계 라우터의 IP 주소 유형 5: 외부 네트워크 주소

Link state age	Reserved	E	T	Link state type
Link state ID				
Advertising router				
Link state sequence number				
Link state checksum	Length			

# 링크 상태 라우팅 (31/47)

- OSPF (23/39)
  - 링크 상태 갱신 패킷 (4/4)
    - 공통 헤더 (3/3)

필드명	설명
광고 라우터 (Advertising router)	메시지를 광고하는 라우터의 IP 주소
링크 상태 순서 번호 (Link state sequence number)	각 링크 상태 갱신 메시지에 할당된 순서 번호
링크 상태 검사합 (Link state checksum)	경과 시간 필드를 제외한 전체 패킷에 플레처의 검사합 (Fletcher's Checksum) 방법을 사용함
길이 (Length)	전체 패킷의 길이를 바이트 단위로 표현함

Link state age	Reserved	E	T	Link state type
Link state ID				
Advertising router				
Link state sequence number				
Link state checksum	Length			

# 링크 상태 라우팅 (32/47)

- OSPF (24/38)

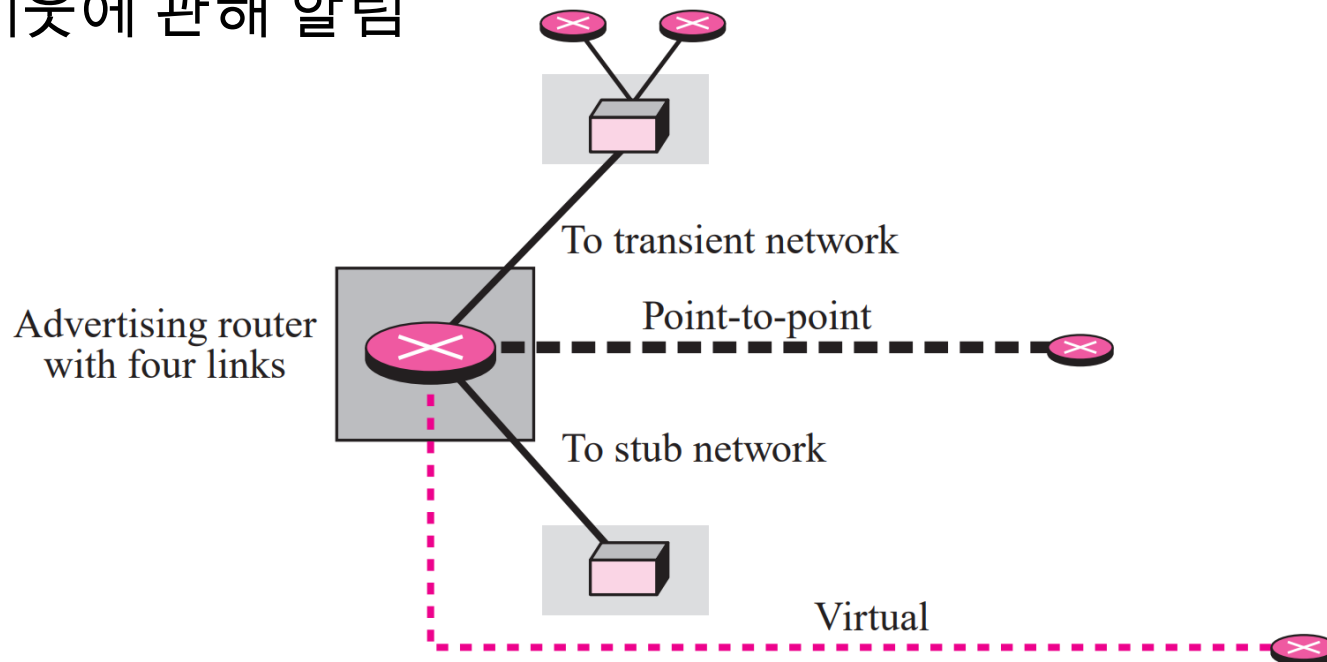
- 링크 상태 갱신 패킷: 라우터 링크 LSA (1/4)

- 역할

- 라우터의 링크들을 정의함

- 특징

- 라우터는 이 LSA를 사용하여 링크의 모든 정보 및 링크의 다른 이웃에 관해 알림



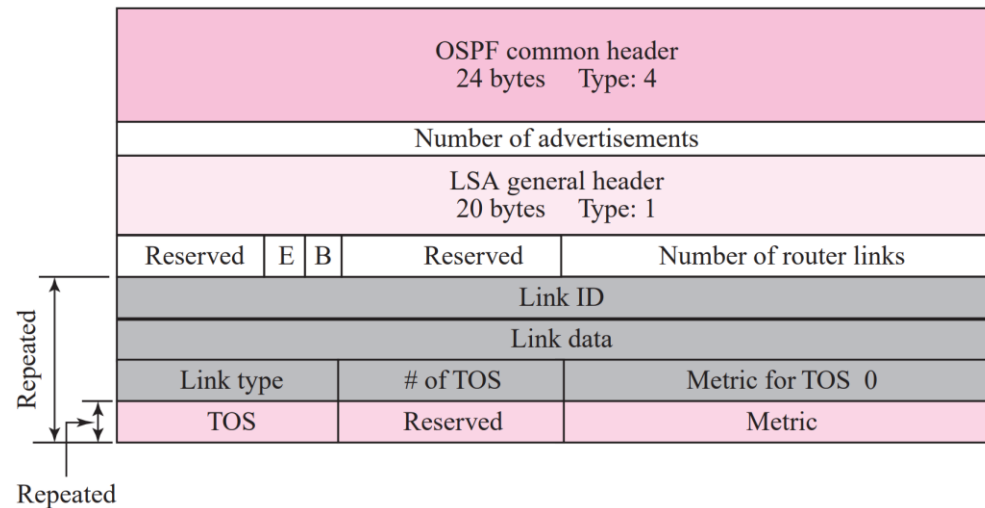
# 링크 상태 라우팅 (33/47)

## • OSPF (25/39)

- 링크 상태 갱신 패킷: 라우터 링크 LSA (2/4)
  - 패킷 형식

필드명	설명
링크 식별자 (Link ID)	링크 유형에 따라 다른 값을 가짐
링크 데이터 (Link data)	링크 유형에 의존한 값을 가짐
링크 유형 (Link type)	라우터가 연결된 네트워크 유형에 따라 4가지의 링크 유형이 정의됨

링크 유형	링크 식별자	링크 데이터
Type 1: 점대점	이웃 라우터의 주소	인터페이스 번호
Type 2: 경유	지정 라우터의 주소	라우터 주소
Type 3: 스텔브	네트워크 주소	네트워크 마스크
Type 4: 가상	이웃 라우터의 주소	라우터 주소

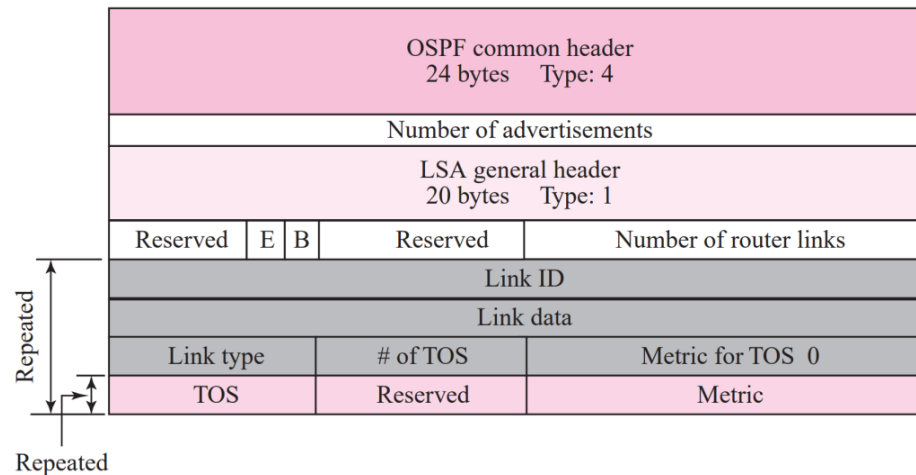


# 링크 상태 라우팅 (34/47)

- OSPF (26/39)

- 링크 상태 갱신 패킷: 라우터 링크 LSA (3/4)
  - 패킷 형식

필드명	설명
서비스 유형의 개수(# of TOS)	각 링크에 알려진 서비스 유형의 개수를 나타냄
TOS 0에 대한 메트릭 (Metric for TOS 0)	기본 서비스인 TOS 0에 대한 메트릭을 나타냄
TOS	서비스 유형을 나타냄
메트릭 (Metric)	해당 TOS에 대한 메트릭을 나타냄



# 링크 상태 라우팅 (35/47)

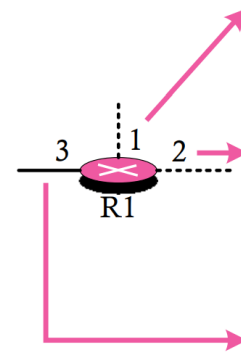
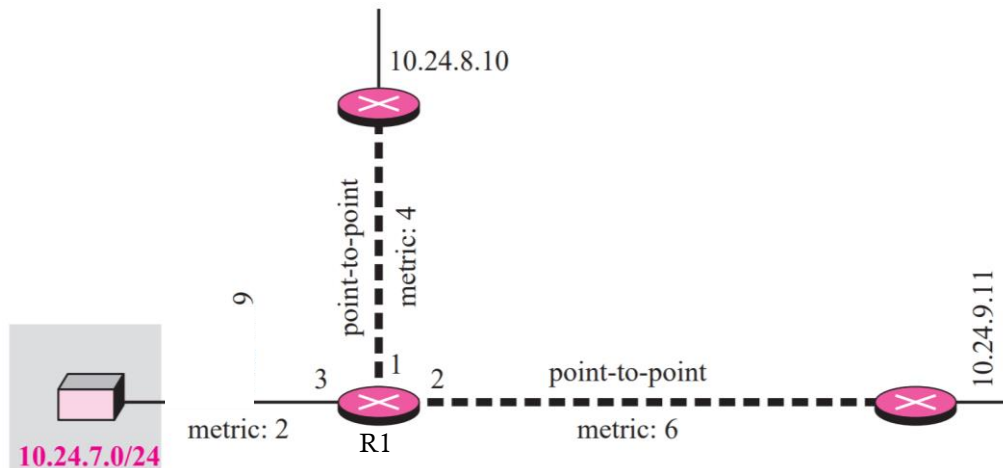
## • OSPF (27/39)

- 링크 상태 갱신 패킷: 라우터 링크 LSA (4/4)
  - 예제 11.7

다음 그림에서 라우터 10.24.7.9에 의해 전송되는 라우터 링크 LSA를 나타내어라.

### • 풀이

- 라우터는 3개의 링크를 가짐
  - 하나는 type 3 (stub)이고, 둘은 type 1 (point-to-point)임
- 오른쪽 그림과 같이 표현될 수 있음



OSPF common header		Type:4
Number of advertisements:		1
LSA general header		Type: 1
Number of links:		3
10.24.8.10		
1		
Link type: 1		
Metric: 4		
10.24.8.10		
1		
Link type: 1		
Metric: 6		
10.24.8.10		
1		
Link type: 3		
Metric: 2		

# 링크 상태 라우팅 (36/47)

- OSPF (28/39)

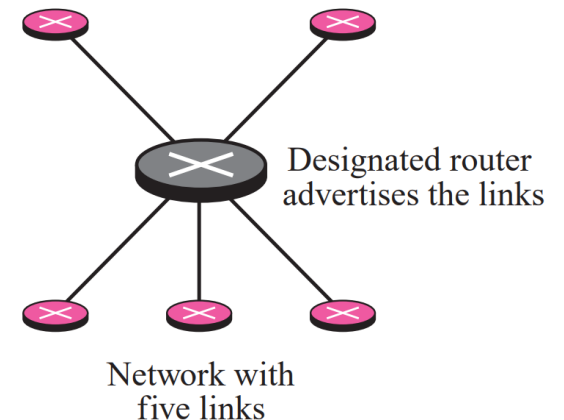
- 링크 상태 갱신 패킷: 네트워크 링크 LSA (1/3)

- 역할

- 네트워크의 링크들을 정의함

- 특징

- 지정 라우터는 경유 망 (Transient network)을 대신해 이런 유형의 LSP 패킷들을 분배함
    - 네트워크에 연결된 모든 라우터들의 존재를 알림

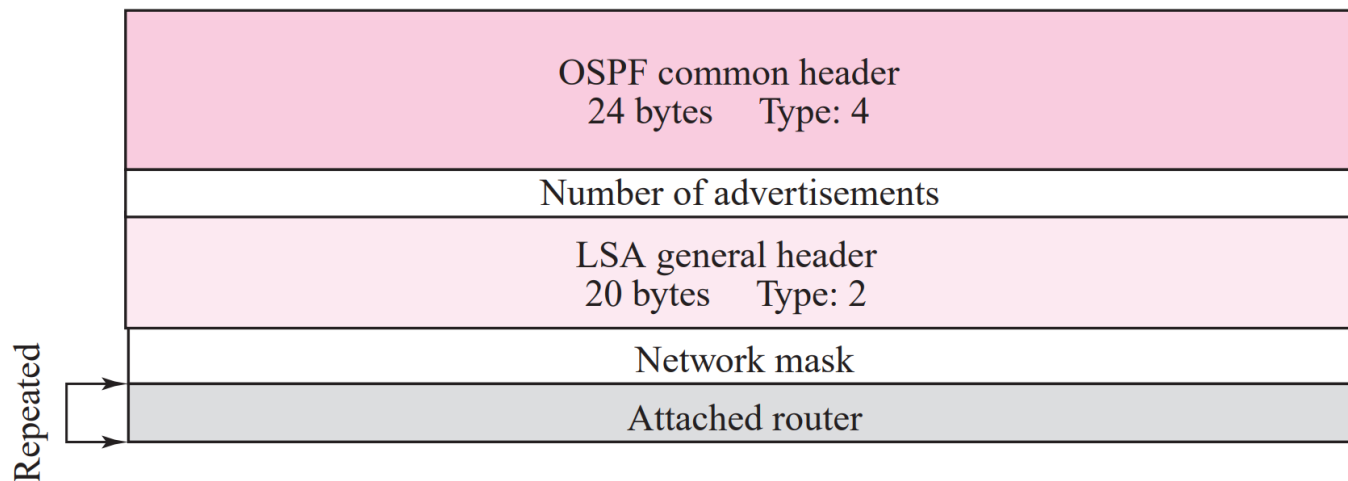


# 링크 상태 라우팅 (37/47)

- OSPF (29/39)

- 링크 상태 갱신 패킷: 네트워크 링크 LSA (2/3)
  - 패킷 형식

필드명	설명
네트워크 마스크 (Network mask)	네트워크 마스크를 나타냄
접속 라우터 (Attached router)	반복되는 필드로, 네트워크에 접속된 모든 라우터의 IP 주소를 나타냄



# 링크 상태 라우팅 (38/47)

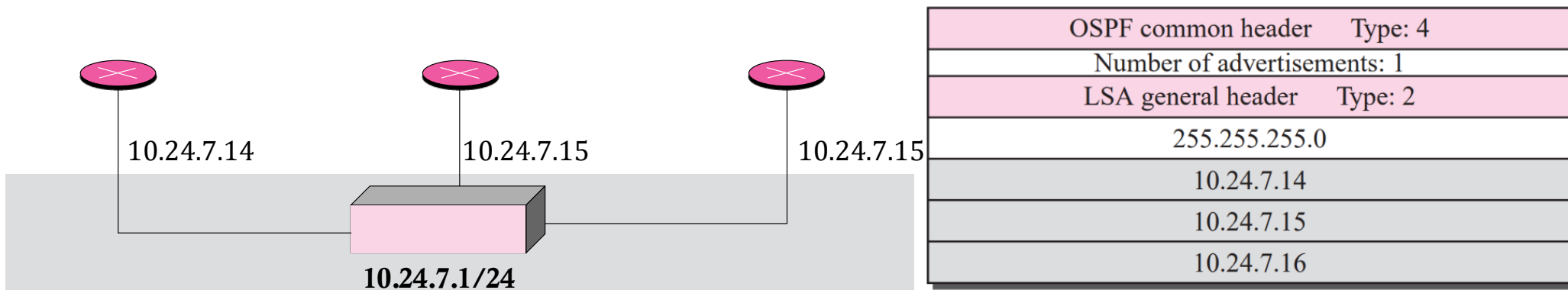
- OSPF (30/39)

- 링크 상태 갱신 패킷: 네트워크 링크 LSA (3/3)
  - 예제 11.8

다음 그림에서 네트워크 링크 LSA를 나타내어라.

- 풀이

- 네트워크 링크를 광고하는 네트워크에는 3개 라우터가 접속되어 있음
- 오른쪽 그림에서 LSA는 마스크와 라우터 주소를 보여줌



# 링크 상태 라우팅 (39/47)

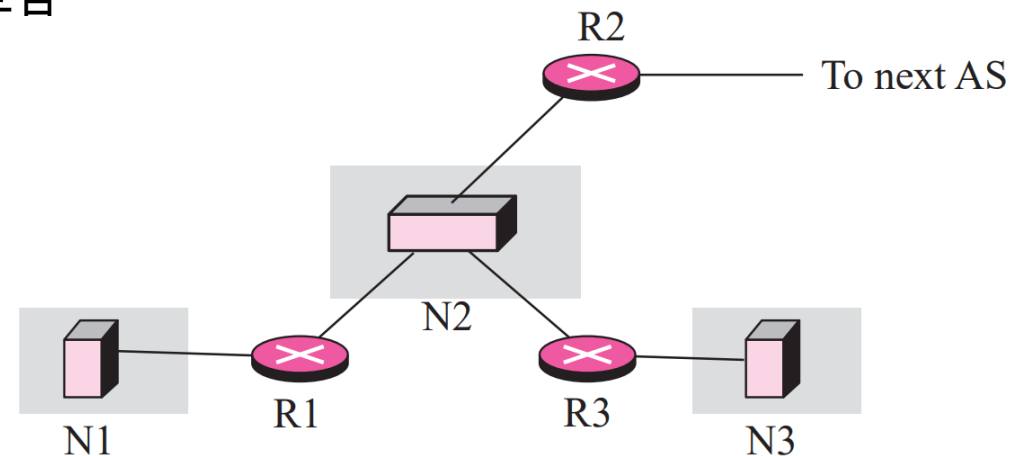
- OSPF (31/39)

- 예제 11.8

다음 그림에서 어느 라우터가 라우터 링크 LSA를 전송하는가?

- 풀이

- R1은 N1과 N2 두 개의 링크를 가짐
- R2는 N2 하나의 링크만을 가짐
- R3은 N2 및 N3 두 개의 링크를 가짐
- 모든 라우터들이 라우터 링크 LSA를 광고함



# 링크 상태 라우팅 (40/47)

- OSPF (32/39)

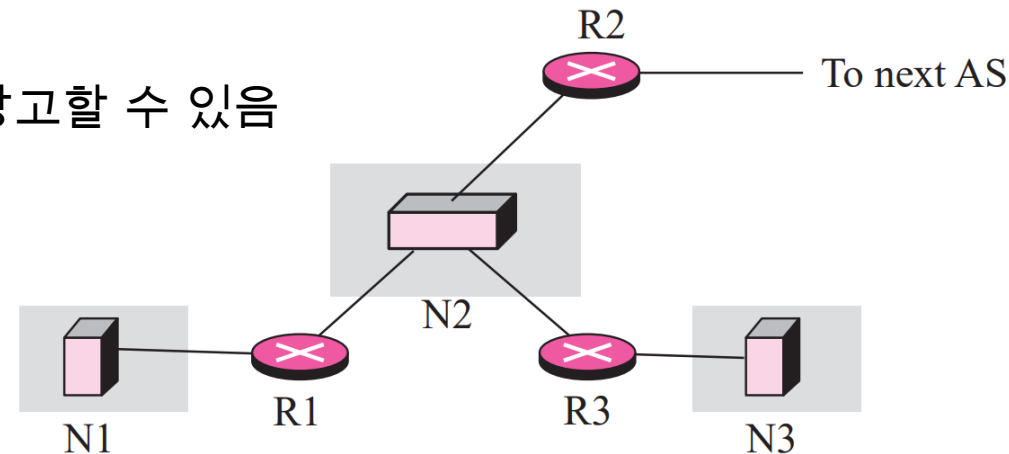
- 예제 11.9

다음 그림에서 어느 라우터가 네트워크 링크 LSA를 전송하는가?

- 풀이

- N1에 대한 광고는 R1이 수행하는데 이는 자신만이 접속 라우터이고 따라서 지정 라우터가 되기 때문임
- N2에 대한 광고는 누가 지정 라우터로 선출되었는지에 따라 R1, R2, R3 모두 가능함
- N3에 대한 광고는 R3이 수행하는데 이는 자신만이 접속 라우터이고 따라서 지정 라우터가 되기 때문임

- 모든 라우터들이 네트워크 링크 LSA를 광고할 수 있음



# 링크 상태 라우팅 (41/47)

## • OSPF (33/39)

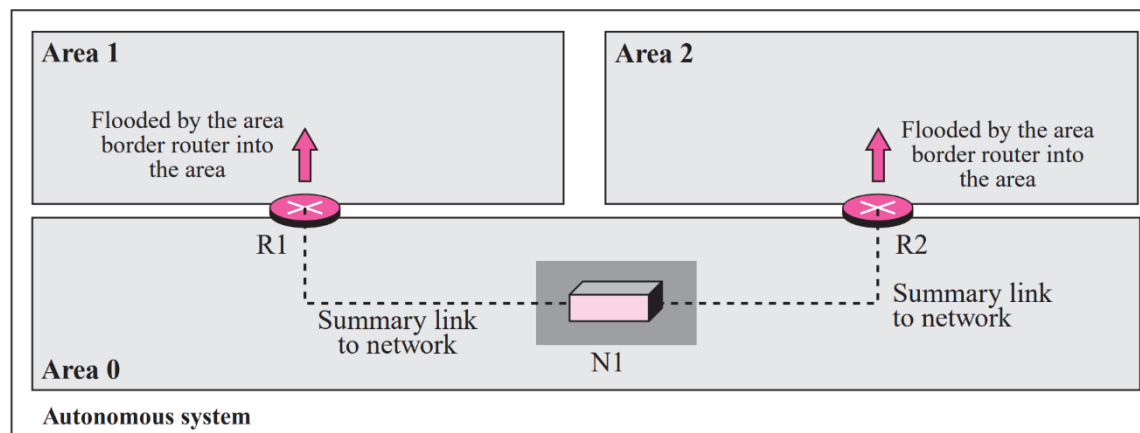
### • 링크 상태 갱신 패킷: 네트워크 요약 링크 LSA (1/2)

#### • 역할

- 지역 바깥의 다른 망 존재를 알리기 위해 사용됨
- 모든 라우터로 하여금 자율 시스템 내의 모든 네트워크로 도달하는 비용을 알도록 함

#### • 특징

- 네트워크 마스크와 각 서비스 유형에 대한 메트릭으로 구성됨
- 각 광고는 하나의 단일 네트워크만을 알림

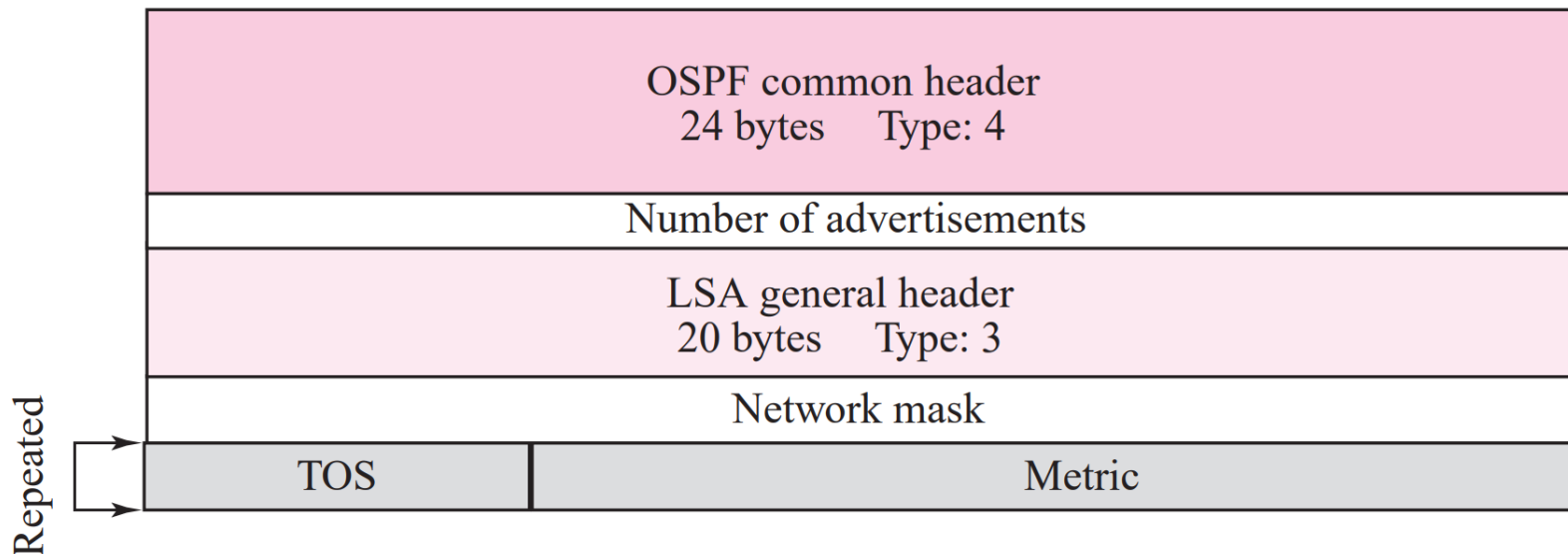


# 링크 상태 라우팅 (42/47)

- OSPF (34/39)

- 링크 상태 갱신 패킷: 네트워크 요약 링크 LSA (2/2)
  - 패킷 형식

필드명	설명
네트워크 마스크 (Network mask)	네트워크 마스크를 나타냄
TOS	서비스 유형을 나타냄
메트릭 (Metric)	TOS 필드에서 정의한 서비스 유형에 대한 메트릭을 나타냄



# 링크 상태 라우팅 (43/47)

## • OSPF (35/39)

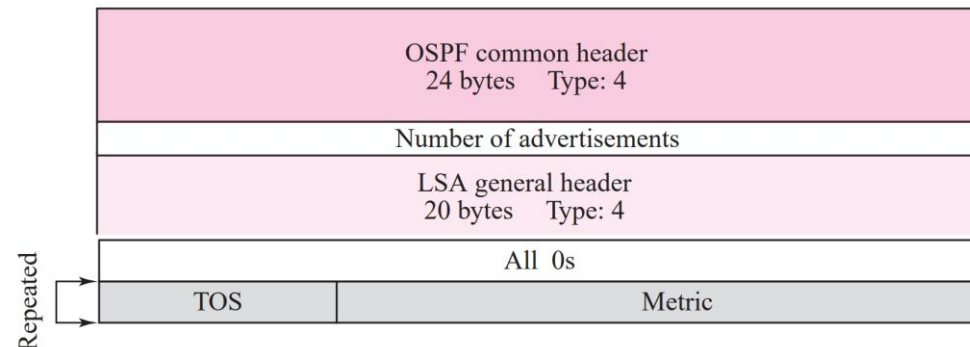
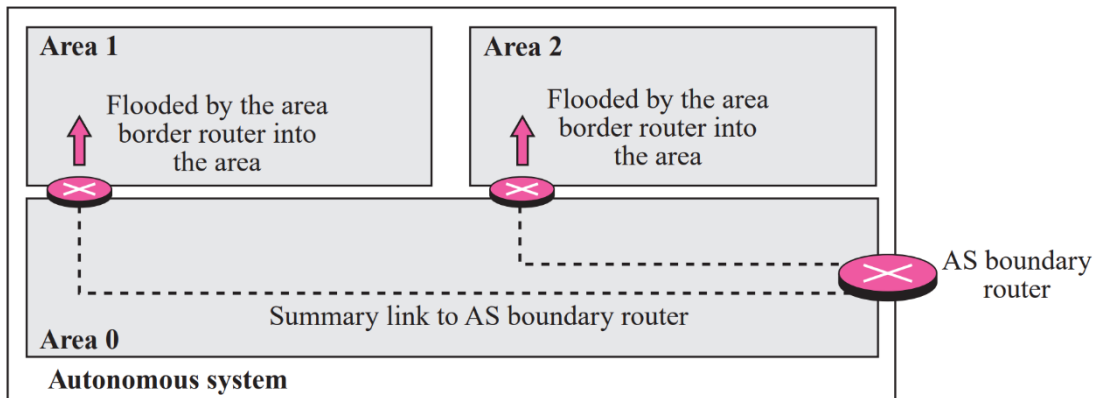
### • 링크 상태 갱신 패킷: 자율 시스템 경계 라우터로의 요약 링크 LSA

#### • 역할

- 자율 시스템 바깥으로 패킷을 전송하고자 할 때, 자율 시스템 경계 라우터로의 경로 정보를 제공함

#### • 특징

- 지역 경계 라우터는 이 정보를 지역에 플러딩함
- 자율 시스템 경계 라우터가 접속되어 있는 네트워크를 정의함
- 형식은 네트워크 요약 링크와 동일함



# 링크 상태 라우팅 (44/47)

- OSPF (36/39)

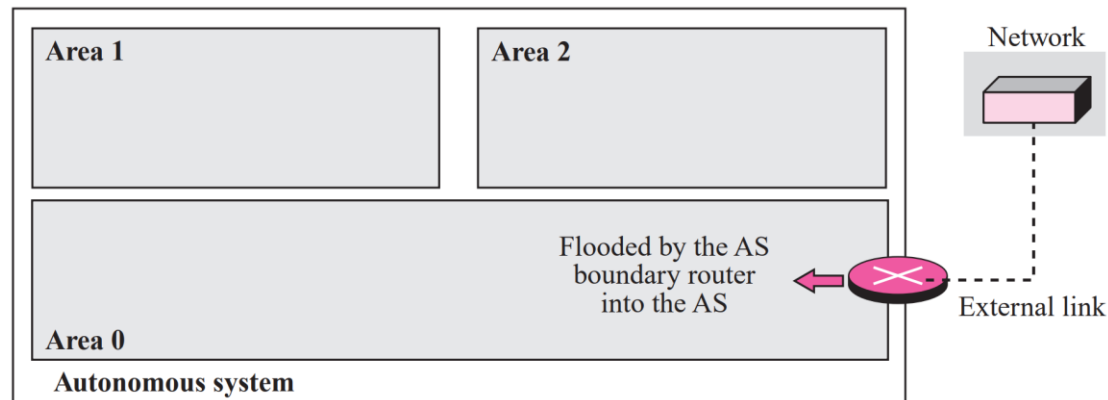
- 링크 상태 갱신 패킷: 외부 링크 LSA (1/2)

- 역할

- 자율 시스템 외부에 이용 가능한 네트워크 정보를 제공함

- 특징

- 도메인 간 라우팅 프로토콜로부터 받은 외부 네트워크 경로 비용을 자율 시스템 내부에 전파하는데 사용됨
    - 각 광고는 하나의 네트워크만을 알림
    - 자율 시스템 경계 라우터로의 요약 링크 LSA에 2개의 필드가 추가된 형식을 가짐

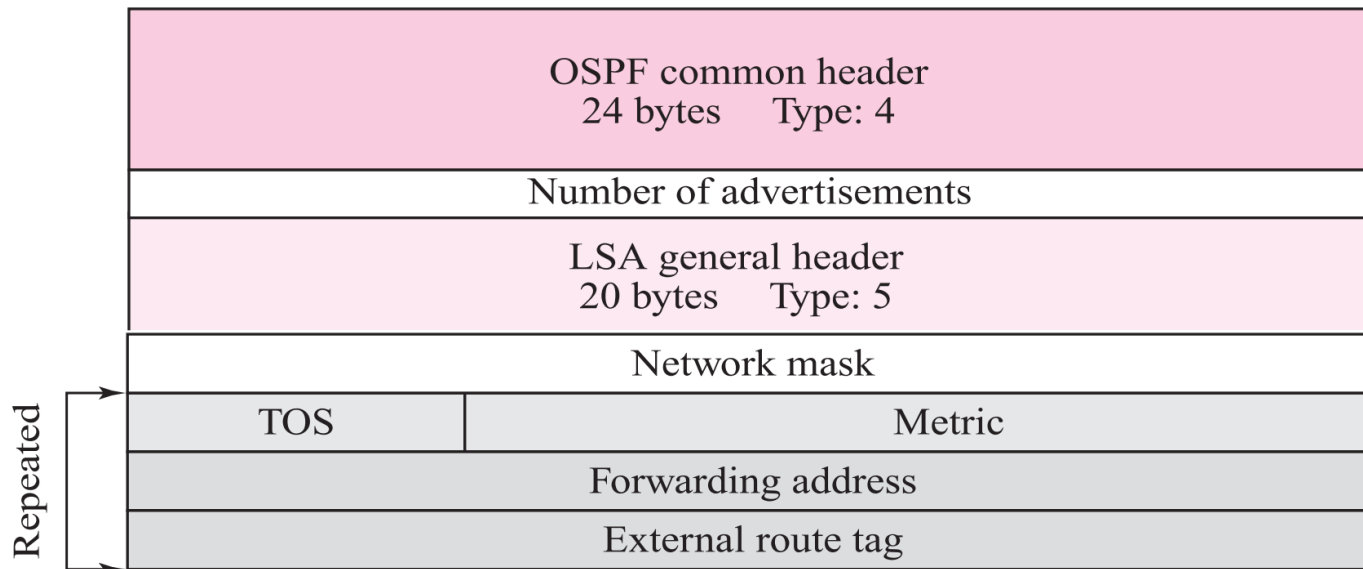


# 링크 상태 라우팅 (45/47)

- OSPF (37/39)

- 링크 상태 갱신 패킷: 외부 링크 LSA (2/2)
  - 패킷 형식

필드명	설명
전달 라우터 (Forwarding address)	외부 경로로 트래픽을 전달할 때 사용하는 다음 홉 IP 주소를 나타냄
외부 경로 태그 (External route tag)	외부 라우팅 프로토콜에서 들어온 경로를 식별하기 위해 사용되는 태그



# 링크 상태 라우팅 (46/47)

- OSPF (38/38)

- 링크 상태 확인응답 패킷

- 역할

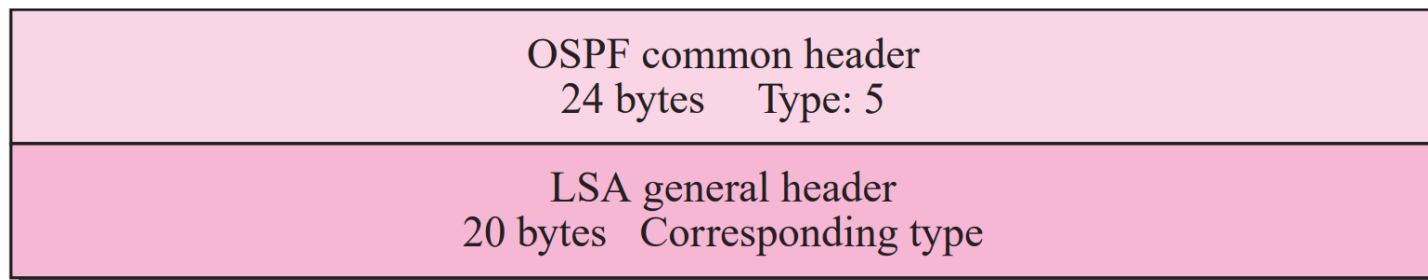
- 모든 라우터들이 확인응답을 하도록 하여, 라우팅을 안정적으로 수행함

- 특징

- LSA에 대한 응답으로 사용됨
    - 전송의 신뢰성을 보장함

- 형식

- 공통의 OSPF 헤더와 일반적인 링크 상태 헤더로 구성됨



# 링크 상태 라우팅 (47/47)

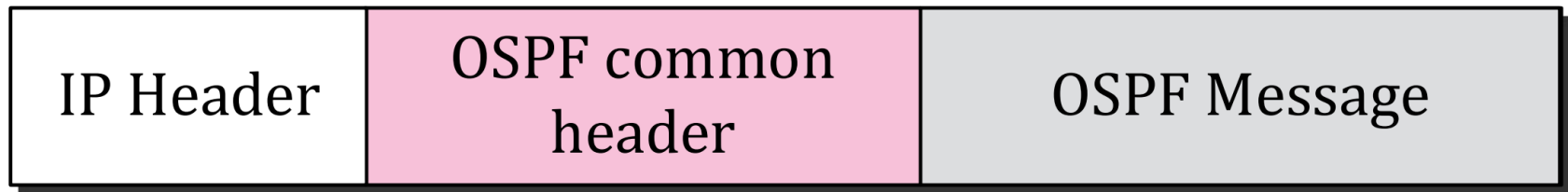
---

- OSPF (39/39)

- 캡슐화

- OSPF는 IP 데이터그램에 캡슐화 됨

- OSPF는 흐름제어와 오류제어에 관한 확인 응답절차를 포함하고 있어, 이에 관한 서비스를 제공하는 전송 계층 프로토콜을 필요로 하지 않음



# 목 차

---

- 자율 시스템
- 거리 벡터 라우팅
- 링크 상태 라우팅
- 경로 벡터 라우팅

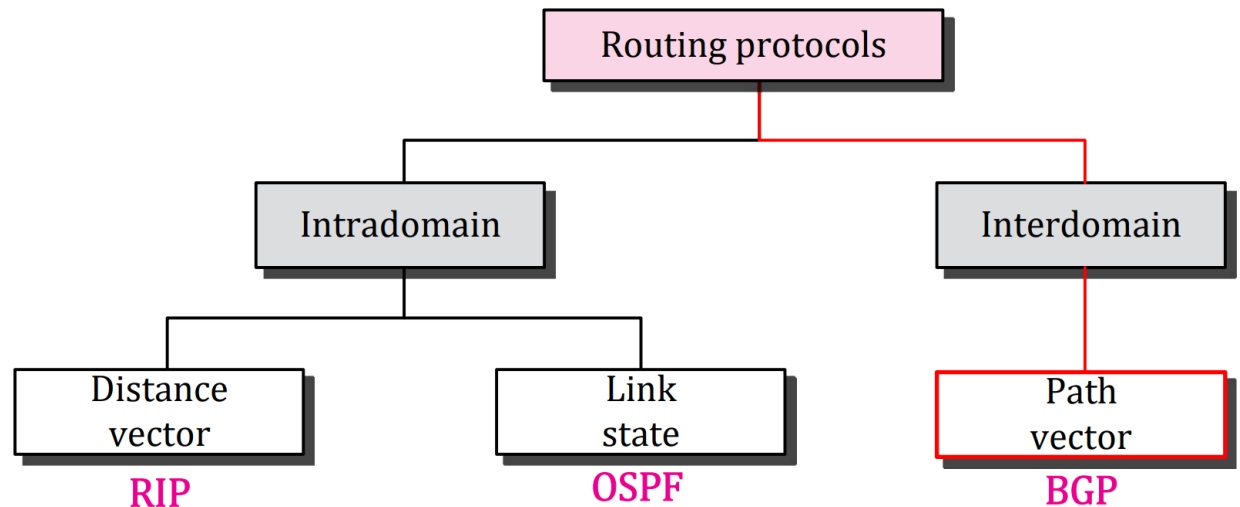
# 경로 벡터 라우팅 (1/21)

- 정의

- 자율 시스템 간 라우팅을 위해 경로에 포함된 자율 시스템들의 목록을 사용하는 방식

- 특징

- 도메인 간 라우팅에 유용함
- 거리 벡터 라우팅과 원리가 유사함
- BGP (Border Gateway Protocol)로 구현됨



# 경로 벡터 라우팅 (2/21)

## • 도달 가능성 정보

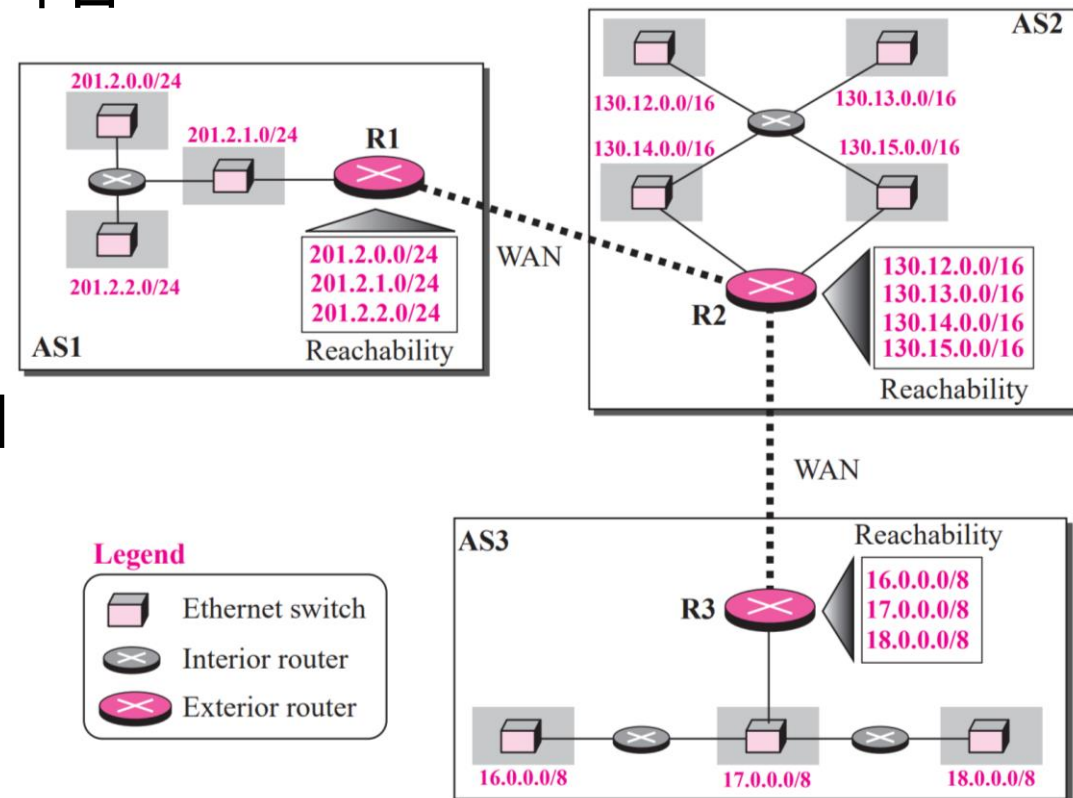
### • 필요성

- 다른 자율 시스템들에 정보를 제공하기 위해서, 각 자율 시스템은 경계 내에서 각 망의 도달 가능성 정보를 모으는 경로 벡터 라우팅을 가져야함

- 도달 가능성 정보들은 어떤 망이 존재하는지를 나타냄

- 네트워크 주소 (CIDR 프리픽스)에 의해 구분됨

- 각 거리 벡터 (외부 라우터)는 어느 망이 그 AS에서 도달 가능한지 보여주는 목록을 생성함



# 경로 벡터 라우팅 (3/21)

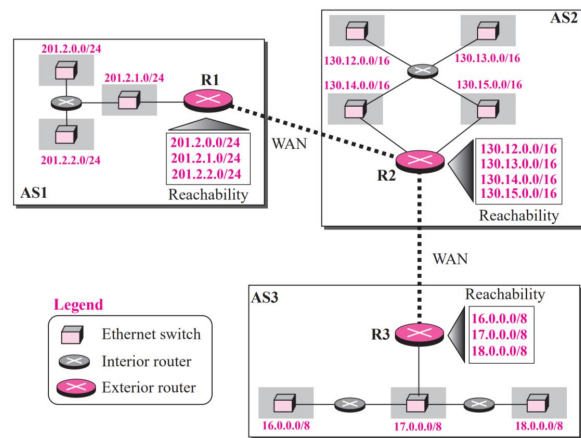
## • 라우팅 테이블 (1/3)

### • 생성

- AS가 도달 가능성 목록을 다른 AS와 공유하면서 생성됨

### • 예시

- R1은 R2에 자신의 도달 가능성 목록을 전송할 수 있음
- R2는 자신의 도달 가능성 정보와 합친 후 R1과 R3에 결과를 전송함
- R3는 도달 가능성 정보를 받아 자신의 라우팅 테이블을 갱신하고, 각 라우터에 해당 정보를 다시 전파함
- R1은 130.14.0.0/16 네트워크로 가는 패킷이 들어오면 목적지 네트워크에 도달하기 위해서 AS1에서 AS2로 가야 함을 알게 됨



Network	Path
201.2.0.0/24	AS1 (This AS)
201.2.1.0/24	AS1 (This AS)
201.2.2.0/24	AS1 (This AS)
130.12.0.0/16	AS1, AS2
130.13.0.0/16	AS1, AS2
130.14.0.0/16	AS1, AS2
130.15.0.0/16	AS1, AS2
16.0.0.0/8	AS1, AS2, AS3
17.0.0.0/8	AS1, AS2, AS3
18.0.0.0/8	AS1, AS2, AS3

Path-Vector Routing Table

Network	Path
201.2.0.0/24	AS2, AS1
201.2.1.0/24	AS2, AS1
201.2.2.0/24	AS2, AS1
130.12.0.0/16	AS2 (This AS)
130.13.0.0/16	AS2 (This AS)
130.14.0.0/16	AS2 (This AS)
130.15.0.0/16	AS2 (This AS)
16.0.0.0/8	AS2, AS3
17.0.0.0/8	AS2, AS3
18.0.0.0/8	AS2, AS3

Path-Vector Routing Table

Network	Path
201.2.0.0/24	AS3, AS2, AS1
201.2.1.0/24	AS3, AS2, AS1
201.2.2.0/24	AS3, AS2, AS1
130.12.0.0/16	AS3, AS2
130.13.0.0/16	AS3, AS2
130.14.0.0/16	AS3, AS2
130.15.0.0/16	AS3, AS2
16.0.0.0/8	AS3 (This AS)
17.0.0.0/8	AS3 (This AS)
18.0.0.0/8	AS3 (This AS)

Path-Vector Routing Table

# 경로 벡터 라우팅 (4/21)

## • 라우팅 테이블 (2/3)

### • 루프 방지

- 거리 벡터 라우팅에서 일어나는 불안정, 루프 생성은 경로 벡터 라우팅에서 해결될 수 있음

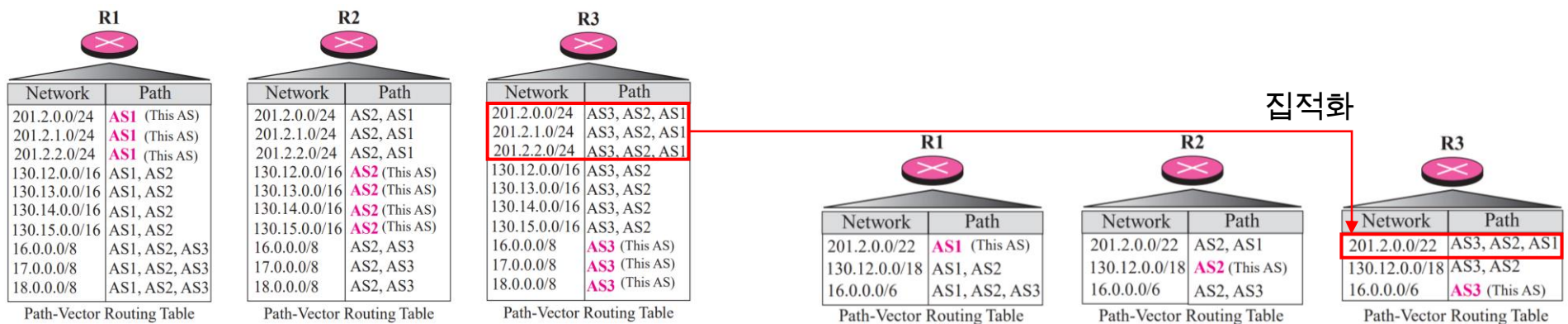
- 라우터가 메시지를 수신하고, 목적지까지의 경로 목록에 해당 자울 시스템이 있는지를 검토함

- 해당 자울 시스템이 있다면, 루프가 일어난 것이므로 메시지를 무시함

### • 집적화 (Aggregation)

- 일반적으로 CIDR 표기가 가능하면 주소의 집적화를 지원함

- 경로 벡터 라우팅을 더 간단하게 만들고 정보 교환을 더 빠르게 함



# 경로 벡터 라우팅 (5/21)

---

- 라우팅 테이블 (3/3)
  - 정책 라우팅 (Policy routing)
    - 라우터가 메시지를 수신하면 경로를 검사하여, 경로 상에 있는 자율 시스템들 중 하나가 정책을 위반하는지 판단함
      - 정책에 위반되면 해당 경로와 목적지를 무시함

# 경로 벡터 라우팅 (6/21)

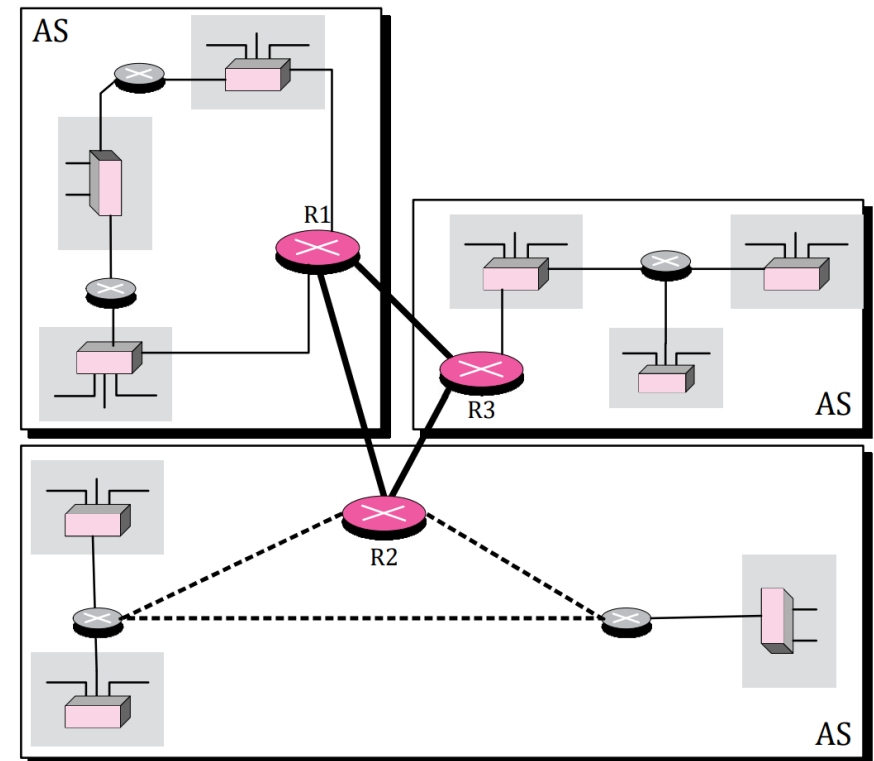
- 경계 게이트웨이 프로토콜 (BGP, Border Gateway Protocol) (1/16)

- 정의

- 자율 시스템 간에 경로 벡터 정보를 교환하여 네트워크 경로를 선택하는 라우팅 프로토콜

- 특징

- 4가지 패킷 종류를 가짐
- CIDR을 사용함
- TCP 서비스를 사용함



# 경로 벡터 라우팅 (7/21)

---

- BGP (2/16)

- 자율 시스템의 유형

- 스템브 AS

- 다른 자율 시스템과 하나의 연결만을 가짐
    - 다른 자율 시스템에 데이터 트래픽 송/수신 가능하나, 거쳐가는 트래픽은 허용되지 않음
    - e.g., 소규모 로컬 ISP (Internet Service Provider)

- 멀티홈 (Multihomed) AS

- 다른 자율 시스템과 하나 이상의 연결을 가짐
    - 하나 이상의 자율 시스템으로 데이터를 송/수신 할 수 있으나, 거쳐가는 트래픽은 허용되지 않음
    - e.g., 국가 AS에 연결된 대규모 조직

- 경유 AS

- 지나가는 트래픽을 허용하는 멀티홈 AS
    - e.g., 국가 혹은 국제 ISP

# 경로 벡터 라우팅 (8/21)

---

- BGP (3/16)

- 경로 속성

- 잘 알려진 속성

- 모든 BGP 라우터가 반드시 인식해야 하는 속성임
    - 필수적인 속성
      - 경로를 기술하는데 반드시 있어야 함
    - 임의 (Discretionary) 속성
      - 모든 갱신 메시지에 포함되지 않아도 됨

- 선택적인 속성

- 모든 라우터에서 인식될 필요는 없는 속성임
    - 선택적 천이 (Transitive) 속성
      - 라우터가 속성을 지원하지 않더라도 다음 라우터로 전달되어야 함
    - 선택적 비천이 (Nontransitive) 속성
      - 수신한 라우터가 속성을 지원하지 않으면 해당 속성은 폐기됨

# 경로 벡터 라우팅 (9/21)

- BGP (4/16)

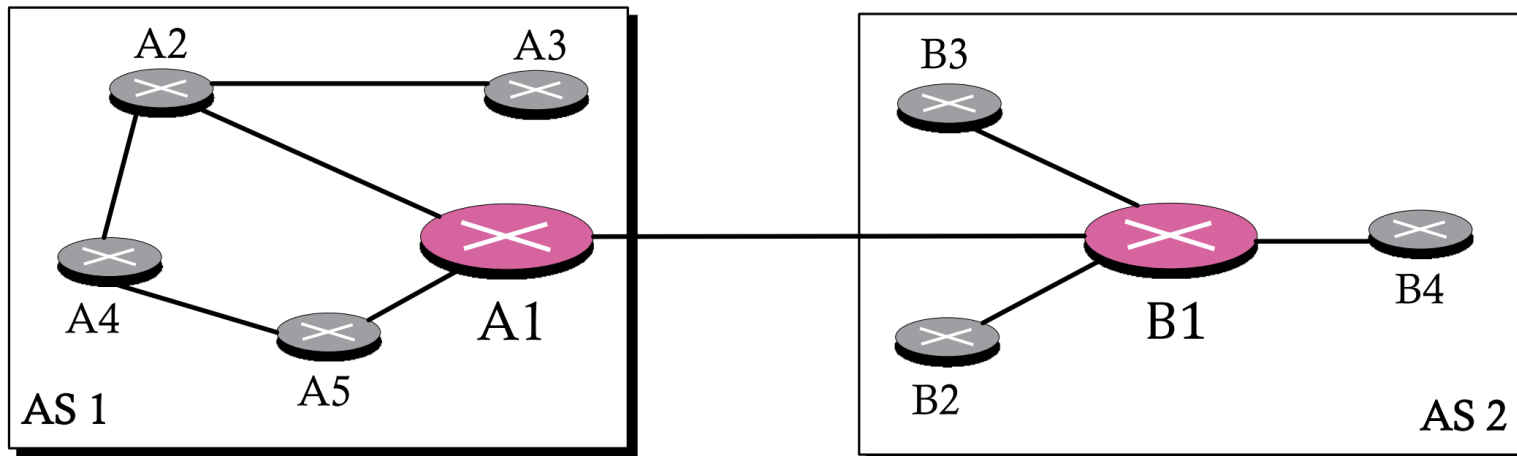
- BGP 세션 (1/2)

- 정의

- 라우팅 정보를 교환하기 위해 두 BGP 라우터 간 설정되는 연결

- 특징

- 신뢰성 있는 환경 구성을 위해 TCP 서비스를 사용함
    - BGP에서 TCP 연결이 수립되면, 예외 사항이 발생하기 전까지 오랜 시간 세션이 유지됨
      - 이러한 이유로, BGP 세션은 반영구적 연결로도 불림



# 경로 벡터 라우팅 (10/21)

- BGP (5/16)

- BGP 세션 (2/2)

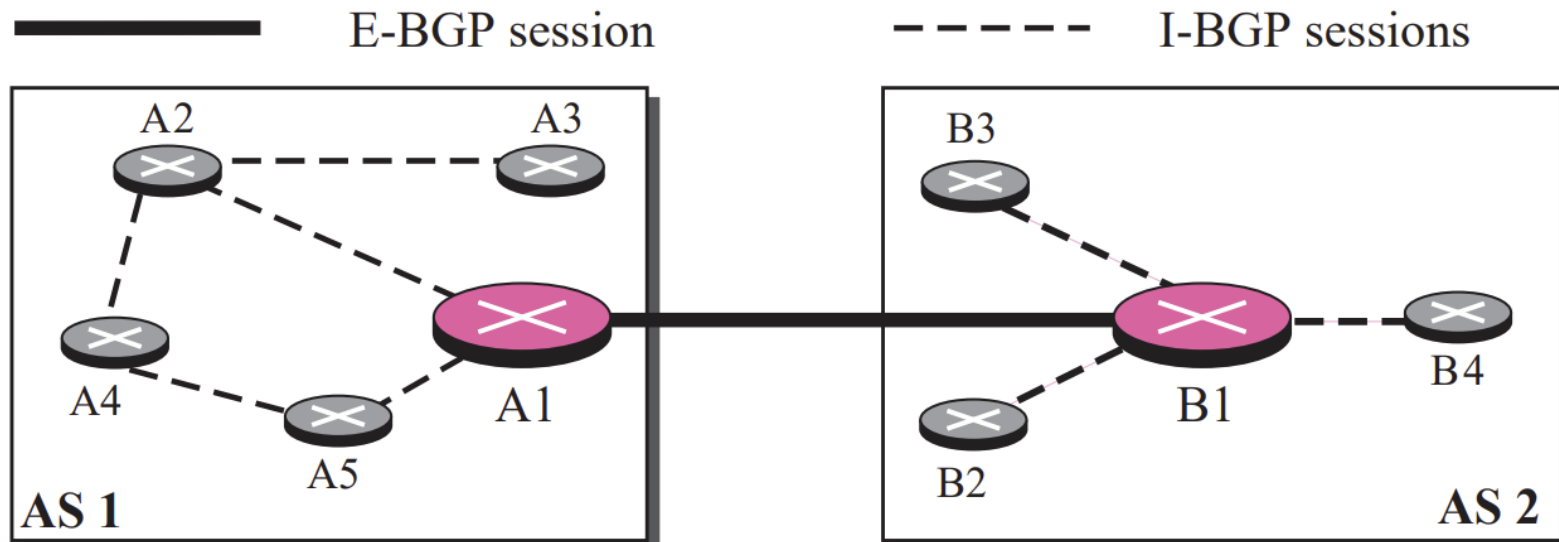
- 외부 BGP (E-BGP, External BGP) 세션

- 서로 다른 자율 시스템에 속하는 두 스피커 노드\*들 간에 정보 교환을 위해 사용됨

- 내부 BGP (I-BGP, Internal BGP) 세션

- 자율 시스템 내 두 라우터 간에 정보 교환을 위해 사용됨

\*스피커 노드 (Speaker node): 경로 벡터 라우팅에서 전체 자율 시스템을 대신하여 활동하는 노드



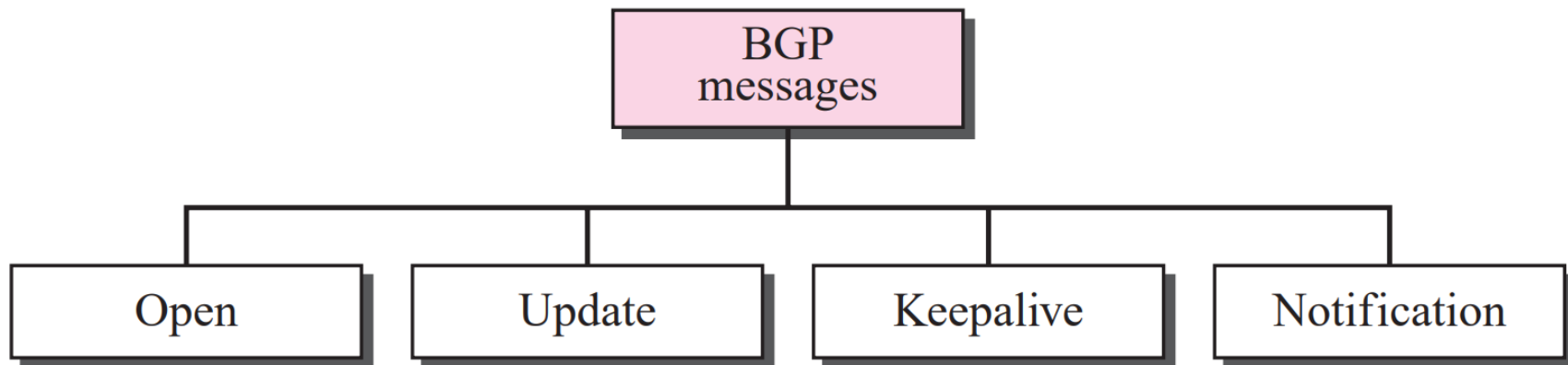
# 경로 벡터 라우팅 (11/21)

---

- BGP (6/16)

- 패킷 종류

- 개방 (Open)
- 킵얼라이브 (Keepalive)
- 갱신 (Update)
- 통지 (Notification)

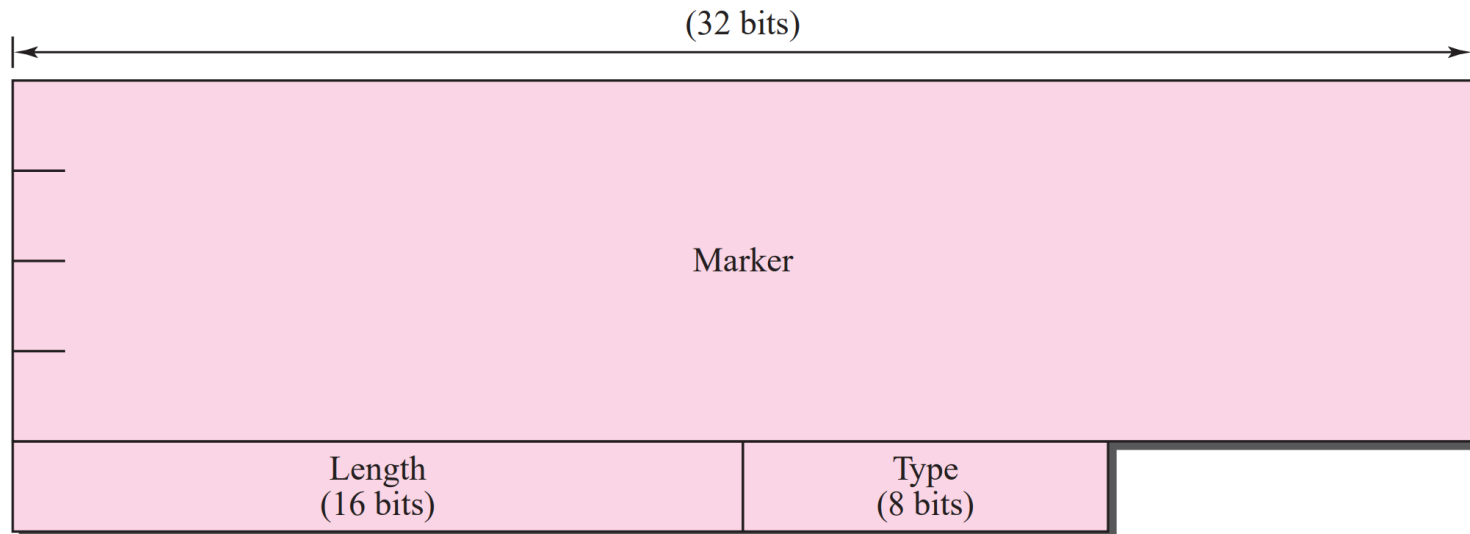


# 경로 벡터 라우팅 (12/21)

- BGP (7/16)

- 패킷 형식: 공통 헤더

필드명	크기 (Byte)	설명
표시기 (Marker)	16	인증을 위해 예약된 필드
길이 (Length)	2	헤더를 포함한 전체 메시지의 길이를 나타냄
유형 (Type)	1	패킷의 유형을 1부터 4까지의 값으로 나타냄



# 경로 벡터 라우팅 (13/21)

---

- BGP (8/16)

- 패킷 형식: 개방 메시지 (1/3)

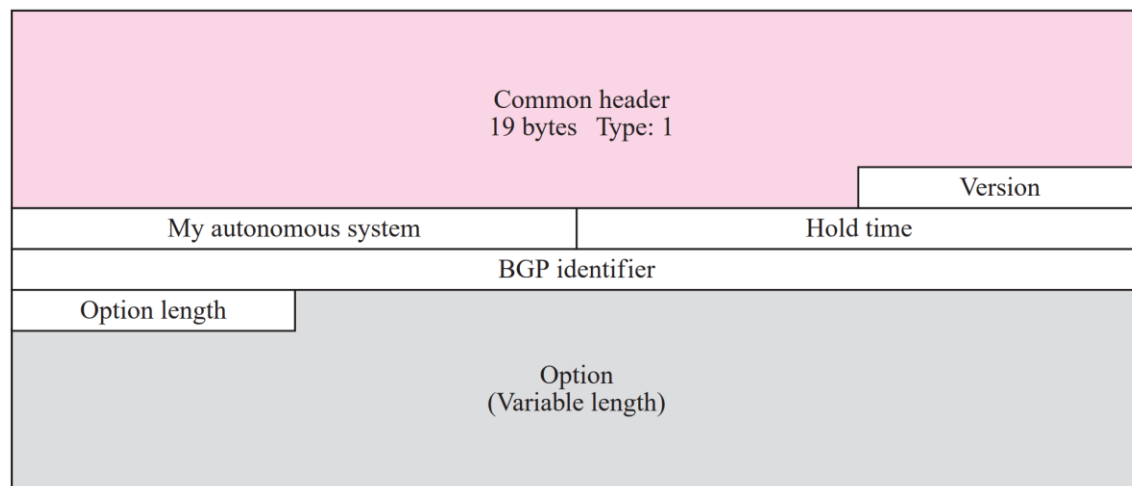
- BGP가 동작중인 라우터가 이웃 관계를 생성하기 위해서 TCP 연결을 열고 개방 메시지를 전송함
- 수신자가 이웃 관계를 받아들이면 두 라우터 간에 이웃 관계가 성립되었다는 의미로, 킵얼라이브 메시지를 보냄

# 경로 벡터 라우팅 (14/21)

- BGP (9/16)

- 패킷 형식: 개방 메시지 (2/3)

필드명	크기 (Byte)	설명
버전 (Version)	1	BGP의 버전을 나타내며, 현재 버전은 4임
자신의 자율 시스템 (My autonomous system)	2	자율 시스템 번호를 나타냄
유지 시간 (Hold time)	2	상대방으로부터 킵얼라이브나 갱신 메시지를 수신하기 전까지 경과될 수 있는 시간을 초 단위로 나타냄

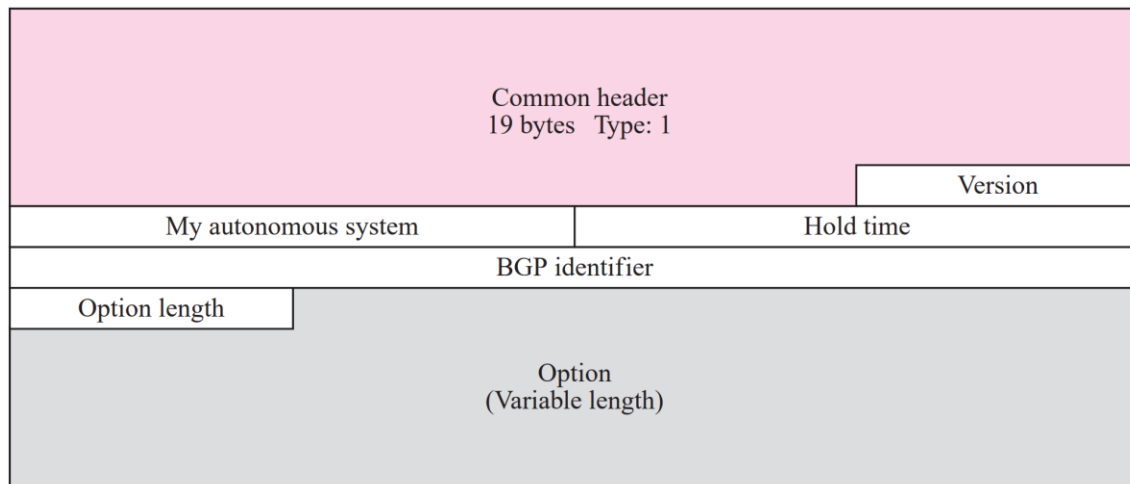


# 경로 벡터 라우팅 (15/21)

- BGP (10/16)

- 패킷 형식: 개방 메시지 (3/3)

필드명	크기 (Byte)	설명
BGP 식별자 (BGP Identifier)	4	개방 메시지를 전송한 라우터의 IP 주소를 나타냄
선택 매개 변수의 길이 (Option length)	1	전체 선택 매개 변수들의 길이를 나타냄
선택 매개 변수들	Variable	각 선택 매개변수들은 매개변수 길이, 매개변수 값의 서브 필드를 가짐 현재까지 정의된 선택 매개변수는 인증이 있음



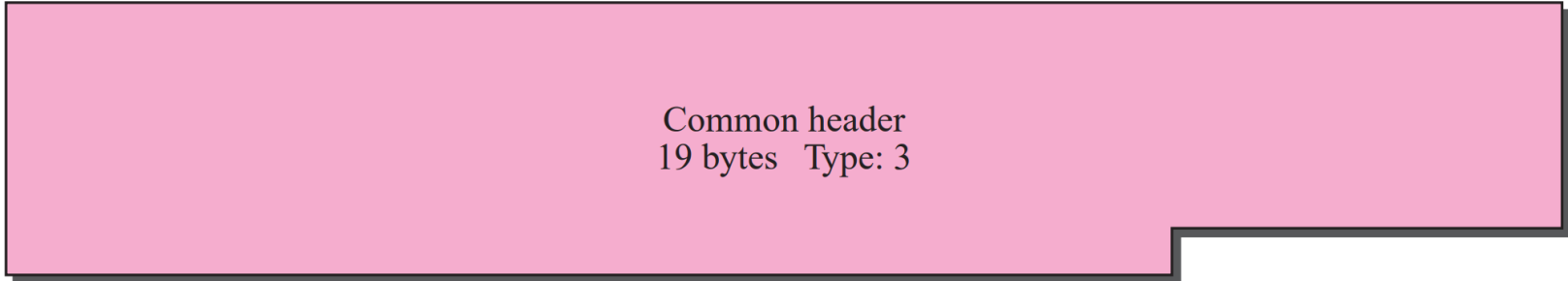
# 경로 벡터 라우팅 (16/21)

---

- BGP (11/16)

- 패킷 형식: 킵얼라이브 메시지

- BGP 라우터들은 자신들이 살아있음을 알리기 위해 유지 시간이 만료되기 전에 정기적으로 킵얼라이브 메시지를 교환함
- 공통 헤더만으로 구성됨



Common header  
19 bytes Type: 3

# 경로 벡터 라우팅 (17/21)

---

- BGP (12/16)

- 패킷 형식: 갱신 메시지 (1/3)

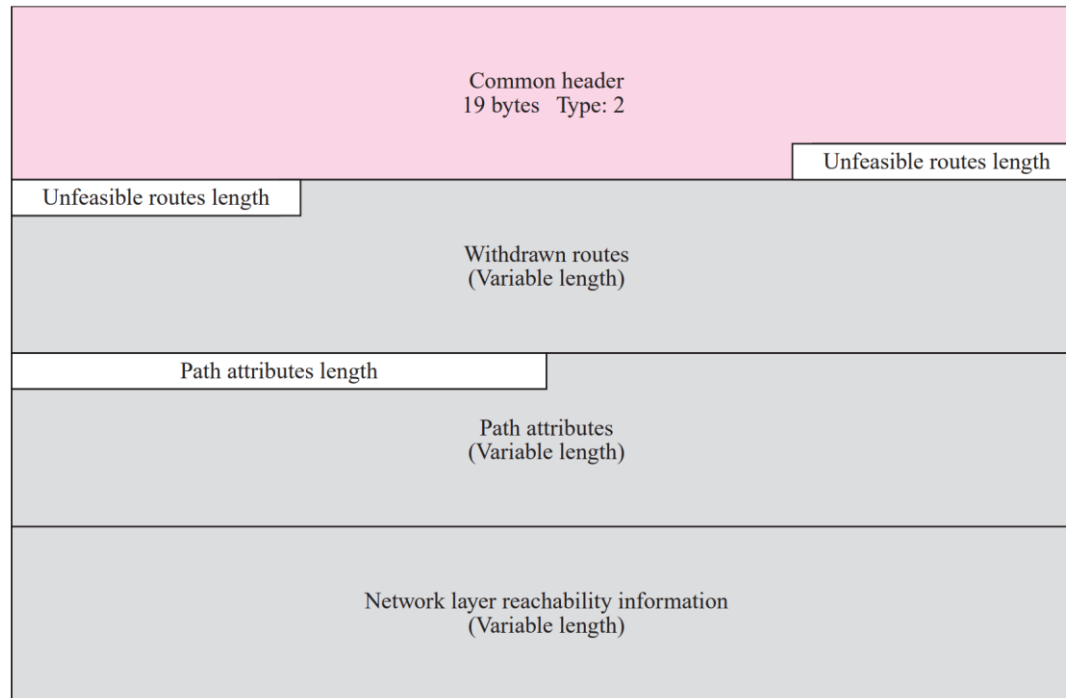
- 라우터로 하여금 전에 광고된 목적지를 취소하거나 새로운 목적지로의 경로를 알리는 데 사용됨
- BGP는 전에 광고된 여러 목적지들을 취소할 수 있으나, 새로운 목적지를 광고할 경우 하나의 갱신 메시지를 통해 한 목적지만 광고 가능함

# 경로 벡터 라우팅 (18/21)

- BGP (13/16)

- 패킷 형식: 갱신 메시지 (2/3)

필드명	크기 (Byte)	설명
불가능 (Unfeasible) 경로 길이	2	다음 필드의 길이를 정의함
취소 (Withdrawn) 된 경로	Variable	전에 광고된 목록 중에서 삭제되어야 하는 모든 경로를 나열함

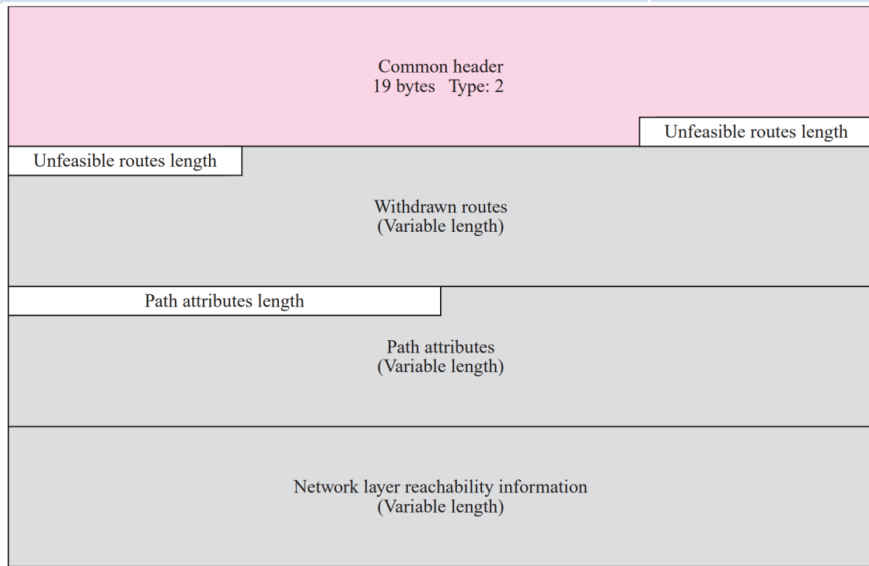


# 경로 벡터 라우팅 (19/21)

## • BGP (14/16)

### • 패킷 형식: 갱신 메시지 (3/3)

필드명	크기 (Byte)	설명
경로 속성 길이 (Path attributes length)	2	다음 필드의 길이를 정의함
경로 속성 (Path Attributes)	Variable	메시지를 통해 광고되는 도달 가능한 네트워크까지의 경로에 대한 속성들을 나타냄
네트워크 계층 도달 가능 정보 (NLRI, Network Layer Reachability Information)	Variable	메시지에 의해 광고되는 네트워크를 나타냄 길이 필드와 IP주소 프리픽스를 가짐



주요 속성	의미
AS-Path	목적지 AS까지 갈 때 경유되는 AS 번호 리스트
Next-Hop	목적지까지 가는 경로에서 반드시 거쳐야 하는 라우터의 주소
Origin	BGP 정보가 내부, 외부 라우팅 프로토콜 중 어디로부터 생성되었는지를 알려줌
Local Preference	목적지까지 여러 경로가 있을 경우, 외부로 나가는 통로들에 대한 선호도

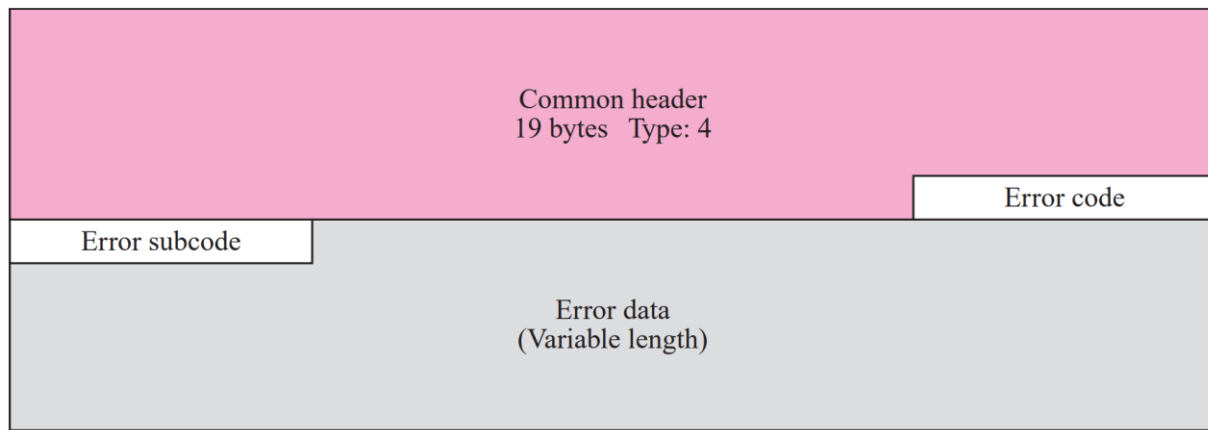
# 경로 벡터 라우팅 (20/21)

## • BGP (15/16)

### • 패킷 형식: 통지 메시지

- 오류 상황이 감지되거나 연결을 닫기 원할 때 라우터에 의해 전송됨

필드명	크기 (Byte)	설명
오류 코드 (Error code)	1	오류의 분류를 나타냄
오류 서브 코드 (Error subcode)	1	각 분류에서의 오류 유형을 나타냄
오류 데이터	Variable	오류에 대한 진단정보를 제공하는데 사용됨



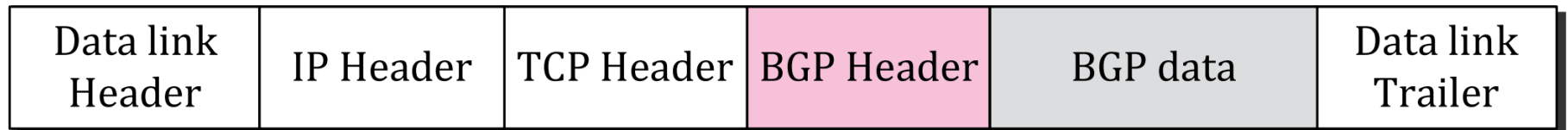
오류 코드	오류 코드 설명
1	Message header error
2	Open message error
3	Update message error
4	Hold timer expired
5	Finite state machine error
6	Cease

# 경로 벡터 라우팅 (21/21)

- BGP (16/16)

- 캡슐화

- 잘 알려진 포트 179를 사용해서 TCP 세그먼트에 캡슐화 됨
  - 오류 제어나 흐름 제어가 필요하지 않음
  - 종료 (Cease) 유형의 통지 메시지가 전송될 때까지 갱신, 킵얼라이브, 통지 메시지의 교환을 지속함



---

# Thanks!

이 정 민(jeongmin@pel.sejong.ac.kr)