9주차 1차시 라우팅 개요

【학습목표】

1. 라우팅의 기본원칙들을 설명할 수 있다.
2. 서브넷과 라우팅, CIDR 개념에 대해 설명할 수 있다.

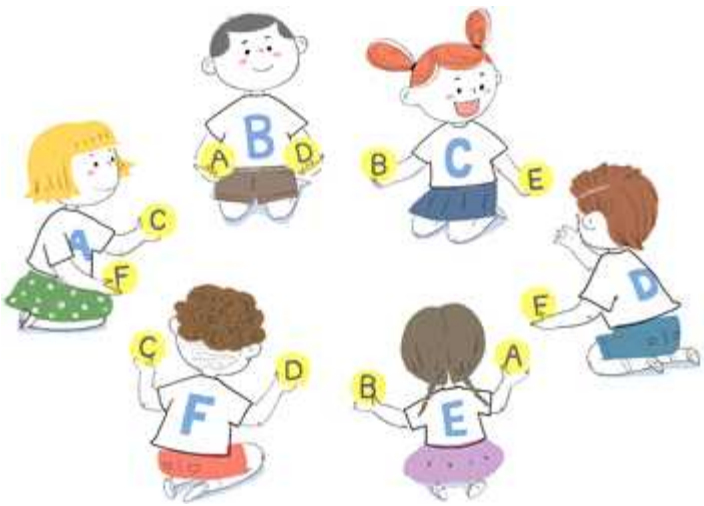
학습내용1 : 라우팅의 개요

1.라우팅 정의

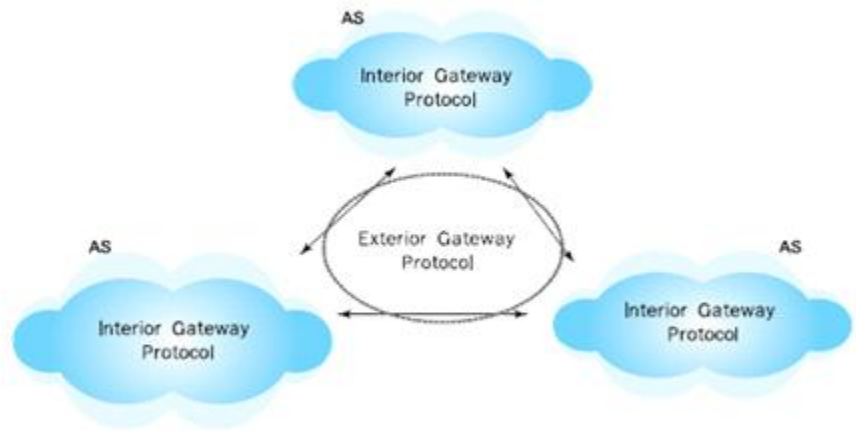
* 라우팅
- 패킷을 전송하기 위해 송신측에서 목적지까지의 경로를 정하고 정해진 경로를 따라 패킷을 전달하는 일련의 과정

* 라우팅 알고리즘
- 최적의 경로를 찾는 방법
- 정적 라우팅 알고리즘 : 관리자가 직접 라우팅 테이블 설정
- 동적 라우팅 알고리즘 : 라우팅 정보 변화에 능동적으로 대처

* 라우팅 프로토콜
- 네트워크 정보의 생성, 교환 제어하는 프로토콜



* AS (Autonomous System)
- 하나의 관리 도메인에 속해 있는 라우터들의 집합
- IGP (Interior Gateway Protocol) : AS내에서 라우팅 정보 교환 (ex. RIP, OSPF, …)
- EGP (Exterior Gateway Protocol) : AS간의 라우팅 정보 교환 (ex. BGP, …)

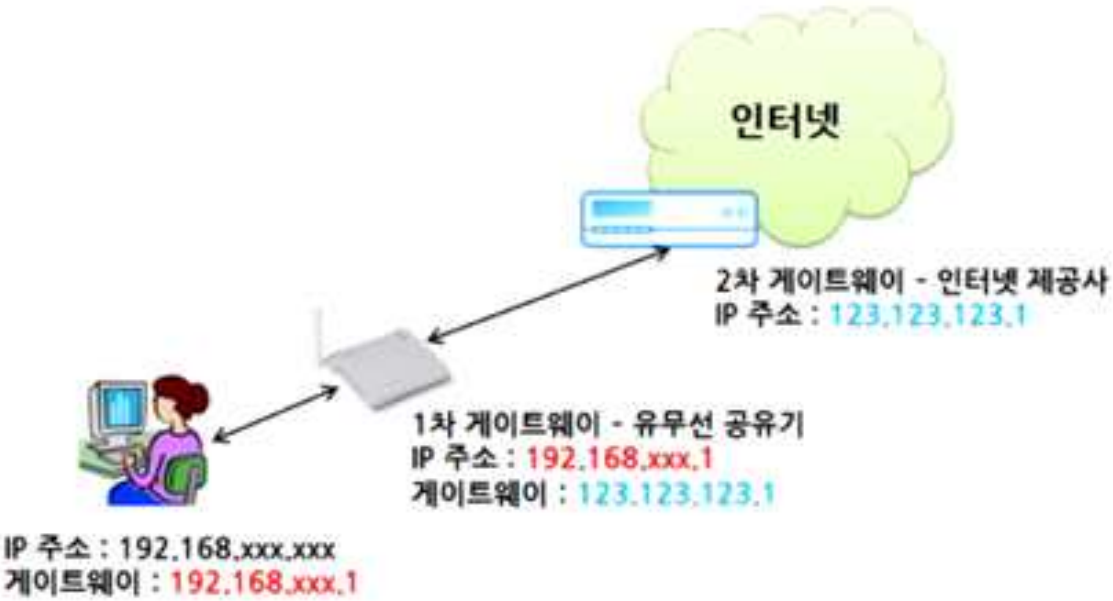


2. 라우팅 기본 원칙

- 라우터는 다른 네트워크의 경로를 나타내는 네트워크 IP 주소와 지역 네트워크에 대한 호스트 IP 주소를 나열하고 있는 라우팅 테이블을 가지고 있음
- 라우터에서 IP 패킷의 처리 되는 과정
① IP 패킷이 도착하면 목적지 주소를 찾기 위해 라우팅 테이블을 검색
② 패킷이 다른 네트워크로 전달되어야 하는 것이면 테이블의 인터페이스에 있는 다음 네트워크로 전달
③ IP 패킷이 LAN에 있는 호스트와 같이 라우터의 지역네트워크에 있다면 직접 보냄
④ 위와 같은 검색 후에도 라우팅 테이블에 경로가 검색되지 않은 경우 디폴트 라우터로 보내짐
- 라우터들이 다른 네트워크들과 지역 호스트에 도달할 수 있는 경로만을 유지 하면 되기 때문에 라우팅 테이블의 크기를 크게 줄일 수 있게 함

3. 서브넷과 라우팅

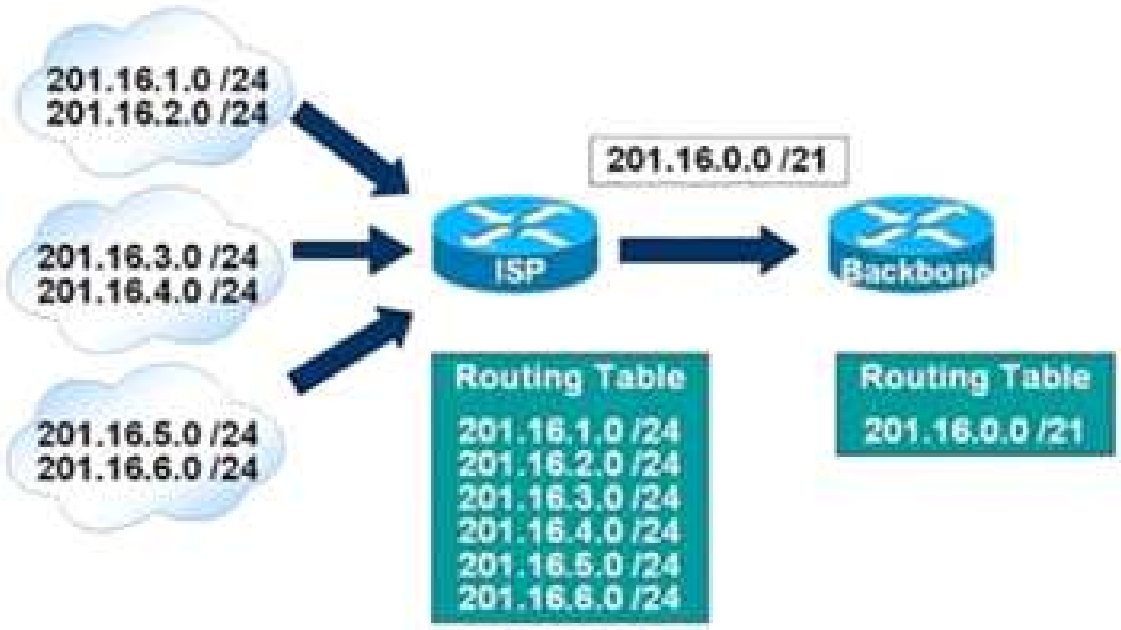
- 라우팅에서 호스트의 네트워크를 결정하기 위해 네트워크의 서브넷 마스크와 호스트 IP의 boolean AND로 IP 주소에서 네트워크 주소를 식별하는 마스킹을 수행
- 서브넷팅이 이루어진 IP의 라우팅은 네트워크로 전달된 다음 해당 서브네트워크로 전달되고 서브네트워크에서 최종 호스트로 전달
- 서브넷은 IP주소 지정을 네트워크, 서브넷, 호스트의 3단계로 나눔으로써 라우팅 테이블의 크기를 줄일 수 있음



4. CIDR (Classless Inter-Domain Routing)

- 대부분의 조직에서 1,600만 개의 주소가 있는 A 클래스 네트워크는 너무 크고, 256개의 주소를 가지는 C클래스는 너무 작음
- 라우터들은 모든 호스트들을 알 필요는 없지만 모든 네트워크들은 알아야 함
- CIDR은 인터넷 라우팅 테이블 크기가 폭증하는 것을 막는 하나의 방법

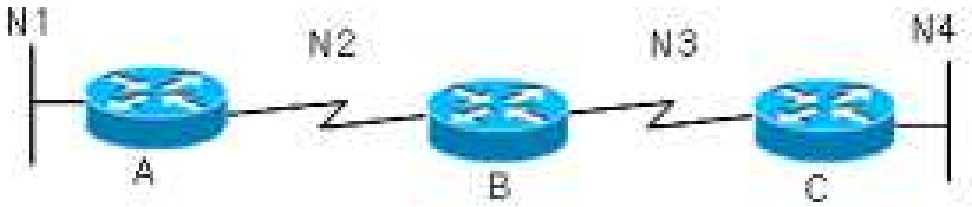
① CIDR는 여러 IP 주소를 보다 작은 라우팅 테이블 엔트리로 나타낼 수 있음
② 단일 사이트에 10의 C 클래스 주소가 할당되었다고 하면 이를 합쳐서 인터넷에서 한개의 라우팅 테이블 엔트리로 찾아갈 수 있도록 할 수 있음
③ CIDR는 라우팅을 위해 합쳐진 동일한 상위 비트의 32비트 마스크를 IP 주소와 함께 전송
④ 212.122.0.0에서 212.122.255.0 범위의 모든 C 주소는 하나의 212.122.0.0/16으로 표현



학습내용2 : 라우팅의 대표적인 알고리즘

1. 거리벡터 알고리즘

1) 개요
- 자신의 라우팅 테이블을 주기적으로 이웃 라우터에게 전송
- 이웃 라우터로부터 라우팅 정보를 수신하여 자신의 라우팅 테이블을 갱신하고 이를 통하여 경로 선택
- RIP 네트워크 모델



Net	D.V	Metric
N1	←	0
N2	→	0
N3		
N4		

↔

Net	D.V	Metric
N1		
N2	←	0
N3	→	0
N4		

↔

Net	D.V	Metric
N1		
N2		
N3	←	0
N4	→	0

Net	D.V	Metric
N1	←	0
N2	→	0
N3	→	1
N4		

↔

Net	D.V	Metric
N1	←	1
N2	←	0
N3	→	0
N4	→	1

↔

Net	D.V	Metric
N1		
N2	←	1
N3	←	0
N4	→	0

Net	D.V	Metric
N1	←	0
N2	→	0
N3	→	1
N4	→	2

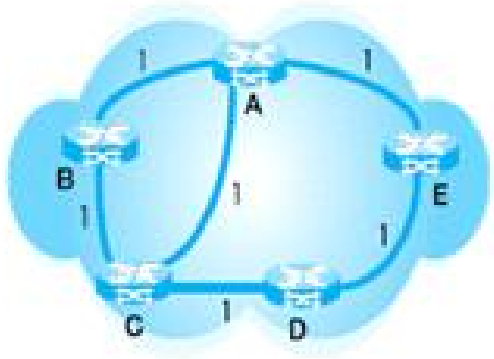
↔

Net	D.V	Metric
N1	←	1
N2	←	0
N3	→	0
N4	→	1

↔

Net	D.V	Metric
N1	←	2
N2	←	1
N3	←	0
N4	→	0

2) 라우팅 테이블 생성
- 목적지 주소와 비용, 이웃 라우터의 주소 저장



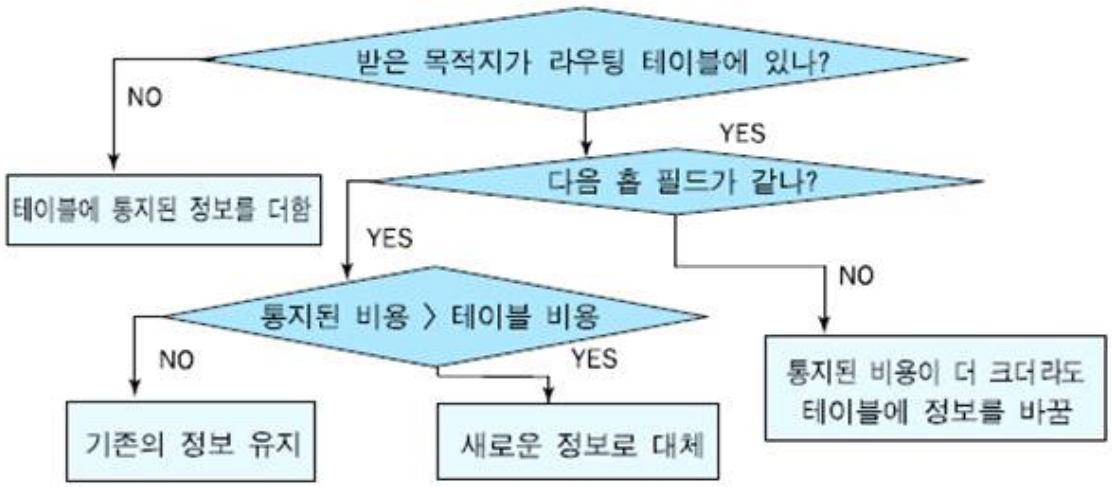
- 라우터의 초기 라우팅 테이블은 자신의 이웃 정보로 구성
- 각 라우터는 자신의 라우팅 테이블을 모든 이웃 라우터와 교환
- 라우팅 테이블 교환 과정을 반복하여 각 라우터는 전체 네트워크의 정보를 얻음

A 라우터			B 라우터			C 라우터			D 라우터			E 라우터		
목적지	비용	다음 홉	목적지	비용	다음 홉	목적지	비용	다음 홉	목적지	비용	다음 홉	목적지	비용	다음 홉
B	1	B	A	1	A	A	1	A	A	∞	-	A	1	A
C	1	C	C	1	C	B	1	B	B	∞	-	B	∞	-
D	∞	-	D	∞	-	D	1	D	C	1	C	C	∞	-
E	1	E	E	∞	-	E	∞	-	E	1	E	D	1	D

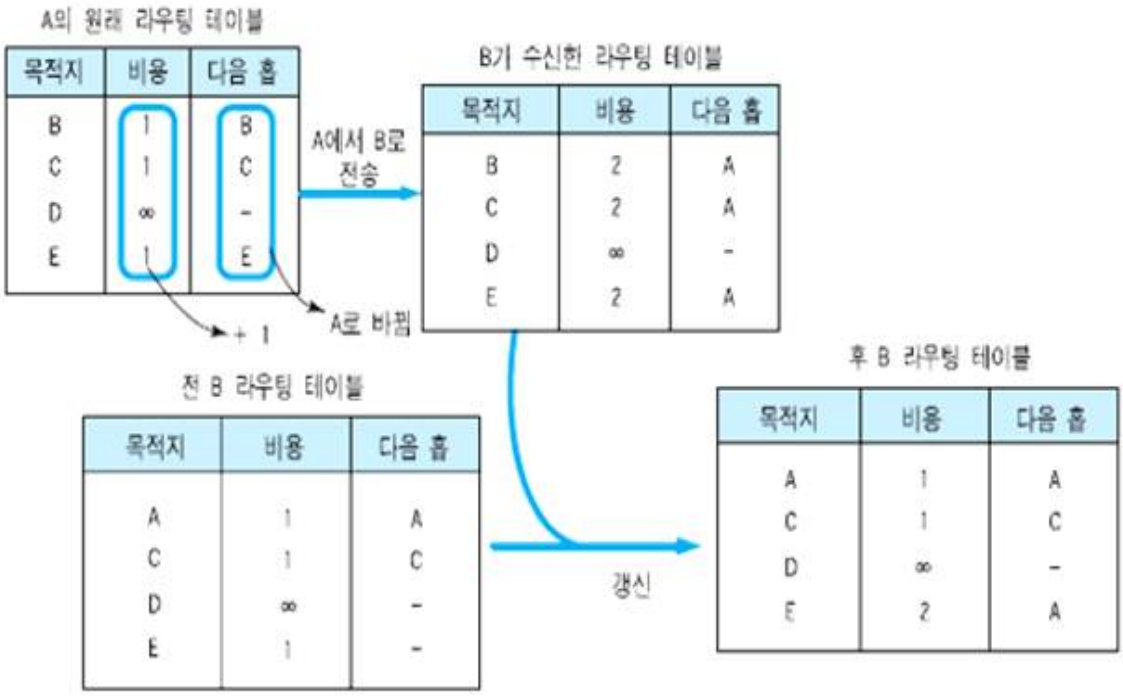
* 라우터 B의 최종 라우팅 테이블

라우터B Routing Table		
목적지	비용	다음 홉
A	1	A
C	1	C
D	2	C
E	2	A

* 라우팅 테이블 갱신
- 이웃 라우터로부터 라우팅 테이블을 수신하면 라우터는 자신의 것과 비교하여 라우팅 테이블을 갱신



* 라우팅 테이블 동작 예 (B가 A로부터 라우팅 테이블 수신)

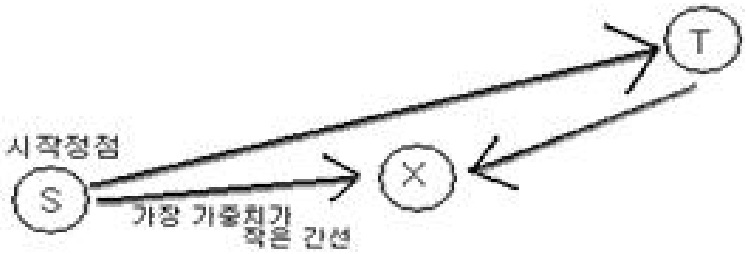


참고) Dijkstra Algorithm

1959년에 만들어진 Dijkstra 알고리즘은 오래된 알고리즘으로서 간선(Edge)을 가진 유향 그래프(Directed Graph)로써 정점들간의 간선에 가중치(Weight)를 가진 그래프에서 최단 경로를 찾을 수 있게 해준다. 또한 이 알고리즘은 음수 값을 갖지 않는 간선을 가지는 가중치 유향 그래프에 사용되는데, 이때 그래프는 서로 연결되어 있어야만 한다.

[원리]

시작 정점으로부터 임의의 정점 X까지의 거리가 현재 다른 정점까지 가는 것보다 작다고 가정하자. 그렇다면 다른 정점을 경유해서 갈 경우, 그냥 임의의 정점 X까지 가는 것이 무조건 더 작을 것이다. (가정에 의해, 그리고 모든 가중치는 양수이다)



위에 그림에서 보면, S->X는 가장 짧은 간선이다. 그러므로 S->T라는 간선은 S->X보다 클 것이다. 또한 T->X는 물론 양수이므로 결국, (S->T + T->X) >(S->X)임을 알 수 있다. 물론, 가장 가중치가 작은 간선이 경로로 바뀐다 해도 이 원리는 변하지 않는다. 다익스트라는 이 원리를 이용한 알고리즘이다.

2.링크상태 알고리즘 (Link State Algorithm)

1) 개요

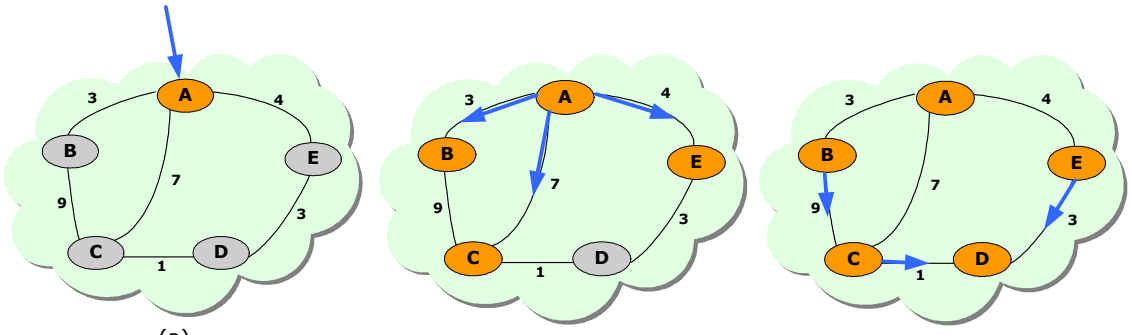
- 라우터는 이웃에 대한 연결정보를 다른 모든 라우터에 전달
- 네트워크 전체 토폴로지에 대한 정보를 얻고 이를 바탕으로 최적의 경로 선택

2) 플러딩 (Flooding)

- 링크 상태 프로토콜을 사용하고 있는 모든 라우터에 링크 상태 정보를 전송 과정
- 링크 상태 패킷(LSP: Link State Packet)을 사용하여 정보 전송
- LSP 구성

LSP 생성 라우터 ID	목적지 주소	비용	이웃 라우터 ID

- LSP 플러딩 과정

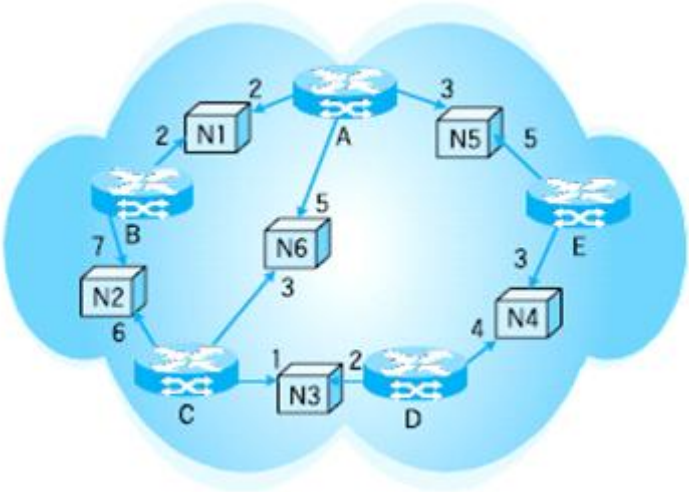


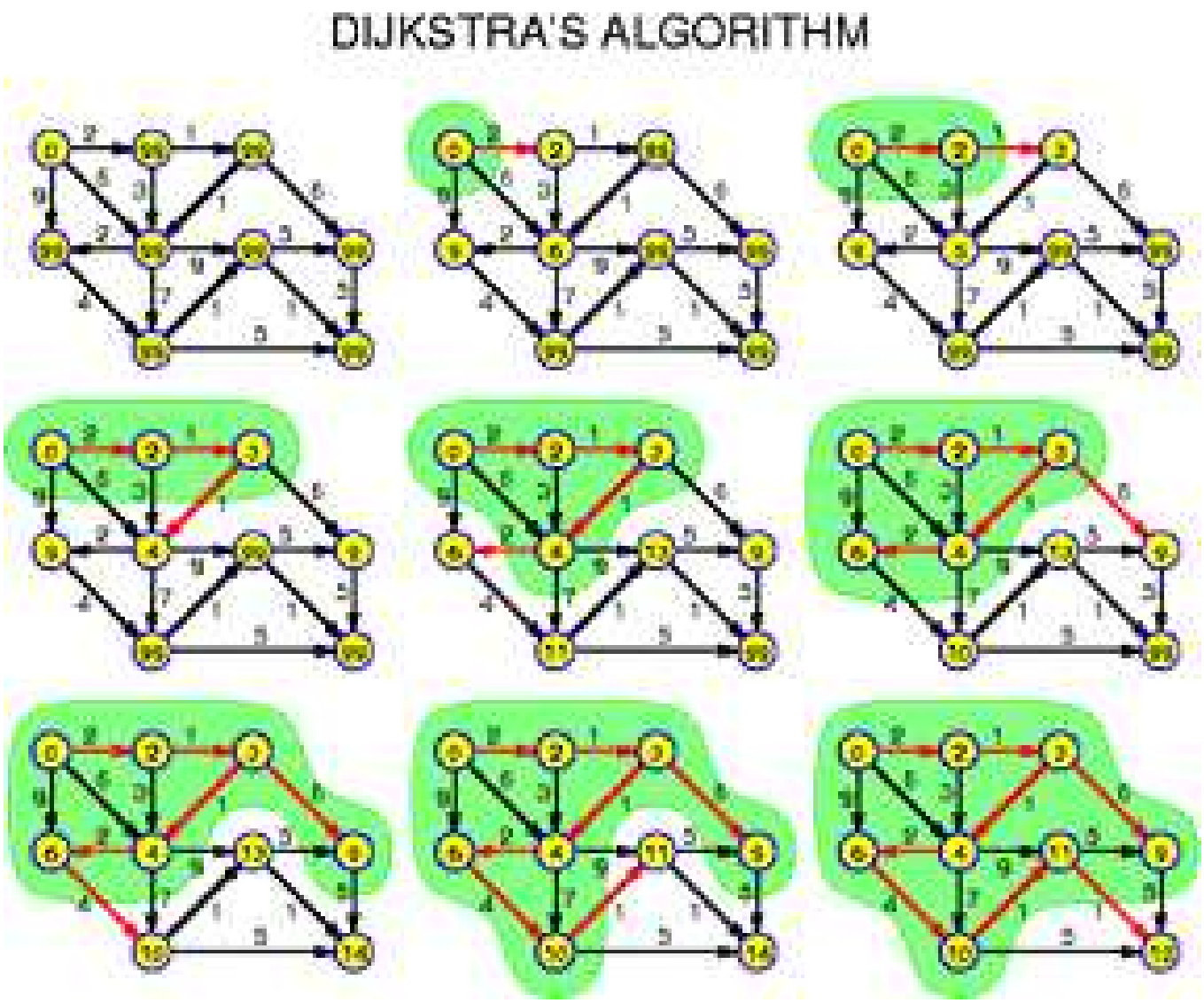
(a) (b) (c)

* 링크 상태 데이터베이스 (Link State Database)
- 모든 라우터는 동일한 네트워크 맵정보를 보유하며 이것으로 최적의 경로를 계산
- 공통의 데이터 베이스 유지
- 링크 상태 데이터베이스 예

광고자	목적지	비용	이웃 노드
A	광고자와 이웃 노드 사이의 네트워크 주소	3	B
A		7	C
A		4	E
B		3	A
B		9	C
C		7	A
C		9	B
C		1	D
D		1	C
D		3	E
E		4	A
E		3	D

3) 최단 거리 트리 - Dijkstra 알고리즘
- 라우터는 자신을 루트로 하여 목적지까지의 최단 거리 트리 구성
- Dijkstra 알고리즘 적용 모델

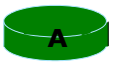




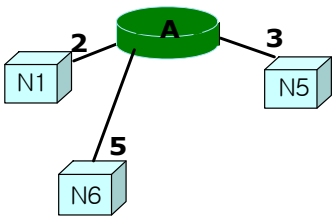
- Dijkstra알고리즘은 라우터와 네트워크를 노드로, 그 연결을 아크로 한 그래프를 이용하여 네트워크에 있는 노드간의 최단경로를 계산
- 하나의 노드를 루트로 하여 아크에 연결된 노드를 임시 노드에 두고 최소 비용을 가지는 노드를 찾는 검사를 수행하여 최단거리 트리의 영구 노드를 결정

① 트리의 루트가 될 하나의 노드를 정한다.
② 1의 노드를 영구노드로 결정한다.
③ 가장 최근에 영구 노드가 된 노드의 이웃 노드를 검사한다.
④ 각 노드에 누적합의 비용을 계산하고 임시 노드로 만든다
⑤ 임시 노드들에 대해서 가장 비용이 적은 노드를 찾아 영구 노드로 만든다. 하나이상의 경로가 존재 할 때는 누적합이 가장 작은 경로를 선택한다.
⑥ 3.에서 5.의 과정을 모든 노드가 영구노드가 될 때까지 반복한다.

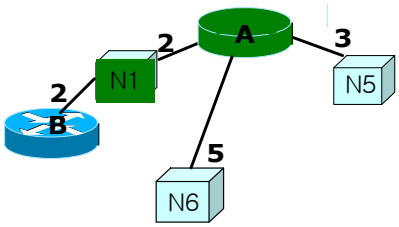
4) A를 루트로 최단 경로 계산



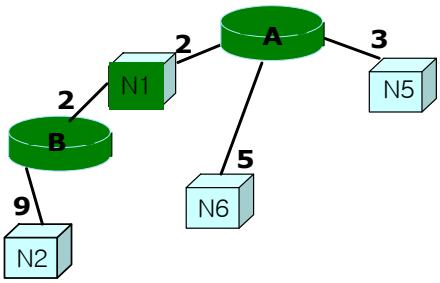
(a) A가 root로 지정



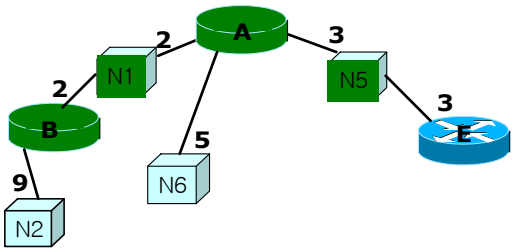
(b) N1, N6, N5가 임시 노드로 추가



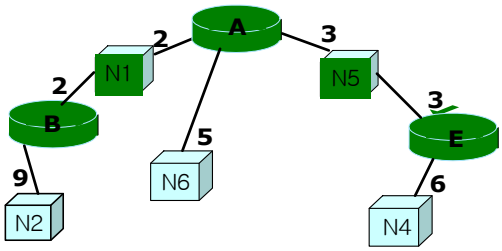
(c) N1이 영구노드, B 트리 추가



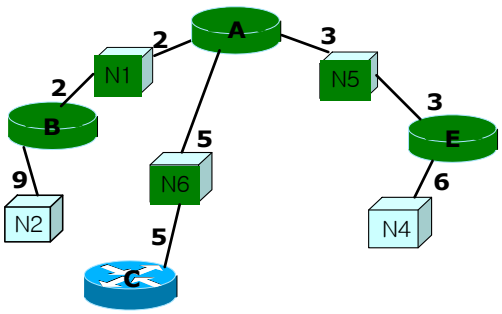
(d) B이 영구노드, N2 추가



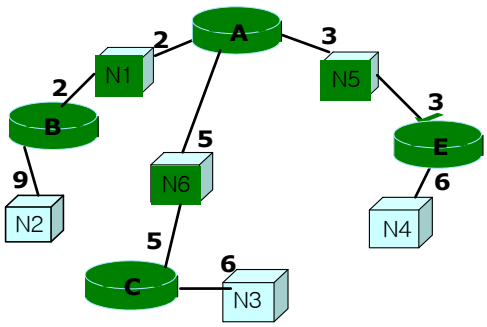
(e) N5이 영구노드, E 트리 추가



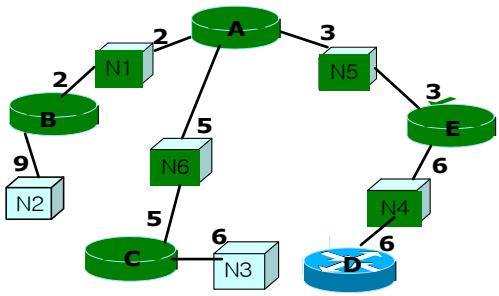
(f) E이 영구노드, N4 트리 추가



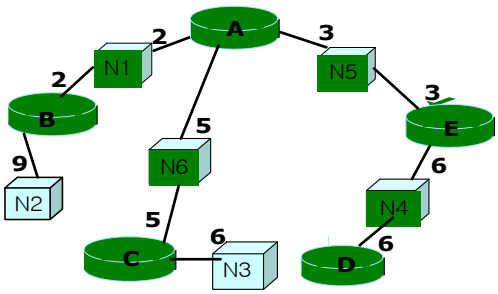
(g) N60이 영구노드, C 트리 추가



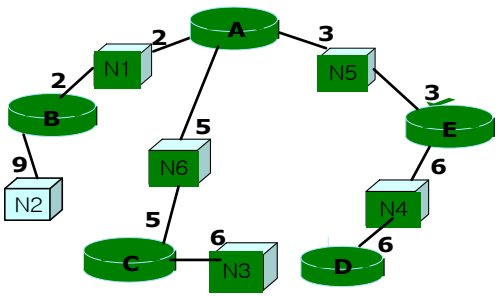
(h) C이 영구노드, N3 트리 추가



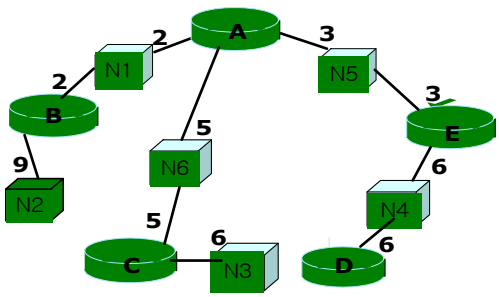
(i) N40이 영구노드, D 트리 추가



(j) D0이 영구노드, N3는 8>6므로 추가안함



(k) N30이 영구노드



(l) N2 영구노드

* 라우터 A의 링크상태 라우팅 테이블

목적지 네트워크	비 용	다음 홉
N1	2	-
N6	5	-
N5	3	-
N2	9	B
N3	6	C
N4	6	E

【학습정리】

1. 라우팅은 패킷을 전송하기 위해 송신측에서 목적지까지의 경로를 정하고 정해진 경로를 따라 패킷을 전달하는 일련의 과정이다.

2. 라우팅을 위한 대표적인 알고리즘으로는 거리벡터 알고리즘과 링크상태 알고리즘을 들 수 있다.