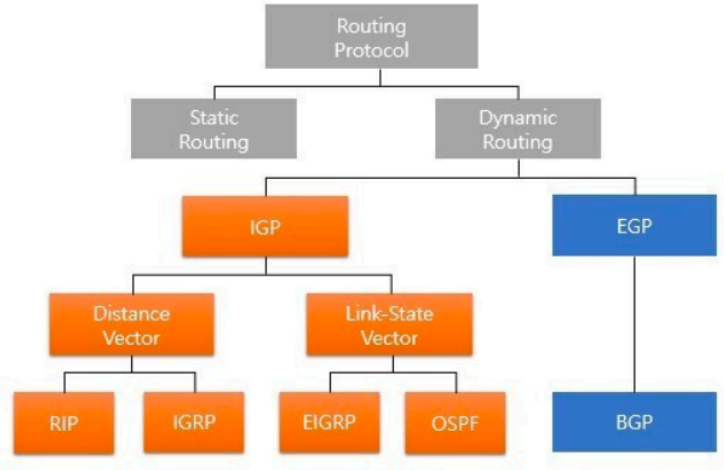
1.네트워크_라우팅 알고리즘



- 라우터는 들어온 패킷을 전송하는 Forwarding 기능과 어디로 패킷을 보낼지 결정하는 Rounting 기능이 있음, Forwarding을 담당하는 부분을 data plane, Routing을 담당하는 부분을 control plane이라고 함
- 정적 라우팅(Static Routing) : 관리자에 의해 Routing Table이 유지/관리 되는 기법
- 동적 라우팅(Dynamic Routing) : 라우팅 프로토콜에 의해 자동으로 라우팅 테이블을 구성하는 기법
  - 내부 라우팅(IGP) : 같은 AS 내부의 라우팅 정보를 고환하는 프로토콜
    - Distance Vector Routing : router 간, 특정 네트워크까지의 거리와 방향을 교환항 라우팅
    - Link state Routing : router 간, 전체 network topologt 연결 상태를 교환하여 라우팅
  - 외부 라우팅(EGP) : 다른 AS 간의 라우팅 정보를 교환하는 프로토콜
  - AS(Autonomius System) : 하나의 그룹/기관/회사와 같이 동일한 라우팅 정책으로 하나의 관리자에 의해 운영되는 네트워크

1. 라우팅 알고리즘

- IP가 출발지에서 목적지까지 패킷을 전달하는 과정에서 라우터가 패킷을 어디로 보낼지 결정하는 알고리즘

1. Link-State Routing Algorithm
    - 중앙집중형 라우팅 알고리즘
    - 과정
        - 각 라우터는 자신과 직접 연결된 링크 상태 정보를 수집
            - 예) 나는 A고 B 까지 비용이 2, C까지는 비용이 5
        - 이 정보를 LSP(링크 상태 패킷 : Link State Packet)이라고 부름
        - 이 LSP를 모든 다른 라우터들에게 브로드 캐스트
        - 각 라우터는 모든 라우터의 LSP를 모아서 전체 네트워크 토폴로지를 구성
            - 이걸 기반으로 Dijkastra 알고리즘을 돌려서 최단 경로 테이블을 만듦
    - 작동 방식
        - 출발 노드를 선정
        - 출발 노드를 기준으로 각 노드의 최소 비용을 저장
        - 방문하지 않은 노드 중에서 가장 비용이 적은 노드를 선택
        - 해당 노드를 겨쳐서 특정한 노드로 가는 경우를 고려하여, 최소 비용을 갱신
        - 3 ~ 4번을 반복
    - 라우터의 브로드 캐스트, 모든 라우터가 연결 상태와 링크 비용을 알고있음
        - 각 라우터가 자신과 직접 연결된 이웃 라우터들과의 상태 정보를 네트워크 전체에 퍼뜨리는 행위, 즉 네트워크의 모든 라우터가 이 정보를 공유
    - 모든 라우터가 모든 링크의 비용(*)을 알고 있기 때문에 다익스트라 알고리즘(Dijkstra Alogrithm)(*)을 이용해 최적의 경로를 계산
    - 대규모 네크워크에 적합
    - 모든 라우팅 정보관리로 메모리 소모
    - SPF(최단 경로 거리 계산) 계산 등 CPU 로드 소요
    - 라우팅 프로토콜
        - OSFP : 기본적으로 ip를 라우팅하도록 구축 되었고 3계층의 프로토콜
        - IS-IS : 2계층 프로토콜로 확장성이 뛰어나 OSFP보다 대규모 네트워크에서 사용

- (*)Dijkstra Alogrithm : 그래프 상에서 시작 정점부터 나머지 각 정점까지의 최단 거리를 계산하는 알고리즘
    - 시간복잡도가 빠름
- (*)비용 : 네트워크에서 패킷이 전달 될 때의 부하, 거리, 지연 시간, 대역폭 소모, 혼잡도 등을 숫자로 추상화한 값
    - A ↔ B 1ms 지연 시간 1
    - B ↔ C 혼잡한 회선이라 느림 5
    - A ↔ C 빠른 광케이블 2
    - A → C → B 경로 (2 + 5 = 7)
    - A → B 경로 (1)
    - 따라서 아래가 더 적은 비용이므로 저 좋은 경로로 선택 됨
- LSD(Link State Database) : 전체 네트워크에 대한 토폴로지와 관련된 모든 정보 이를 통해 SPF 트리를 만들고 최적 경로에 대한 라우팅 테이블을 구축
- SPF(Shortest Path First) : 가장 빠른 경로를 찾는 알고리즘

2. Distance-Vector Routing Algorithm
    - 분산형 라우팅 알고리즘
    - 과정
        - 각 라우터는 자신이 직접 연결된 이웃까지의 거리만 알고 있음
            - 예) A는 B까지 1, C는 모름(무한대)
        - 각 라우터는 주기적으로 자신이 알고 있는 목적지까지의 거리 정보를 이웃에게 보냄
        - 라우터는 이웃에게 받은 정보를 바바탕으로 그 이웃을 통해 목적지까지 가는 거리를 계산함
        - 더 짧은 경로가 있으면 테이블을 업데이트함
        - 전체 네트워크의 라우팅 테이블이 더 이상 바뀌지 않을 때까지 반복
    - 작동 방식
        - 출발 노드를 선정
        - 최단 거리 테이블을 초기화
        - 모든 간선 E개를 하나씩 확인

        - 각 간선을 거쳐 다른 노드로 가는 비용을 계산하여 최단 거리 테이블 갱신
        - 3,4번 반복
        - 위의 동작을 노드 개수 -1 번 반복
    - Count To Infinity(무한 계산 문제)가 생김
        - 링크가 고장나거나 비용이 증가할 때 발생
        - 이웃 라우터가 자신을 경유하는 최소 경로 비용 전송
        - 2개의 이웃 라우터 간에 순환 경로 발생
        - Poison Reverse로 해결
    - 가장 가까운 경로로 설정(가장 가까운 경로는 Hop Count가 최소인 경로, Hop Count는 라우터를 하나 거칠 때마다 1 증가)
    - 라우팅 프로토콜
        - RIP : 라우터와 목적지의 서브넷의 홉 수 (홉 카운트), 가장 오래된 프로토콜이지만 비효율적
        - EIGRP : 하이브리드(거리 벡터 + 링크 상태의 장점 혼합)
    - 벨만-포드 알고리즘(Bellman-Ford)(*)을 이용해 최적의 경로를 계산 할 수 있음
        - 시간복잡도가 느림
    - (*)Bellman-Ford Rounting Algorithm : 한 노드에서 다른 노드까지의 최단 거리를 구하는 알고리즘
3. 벨만-포드 VS 다익스트라
    - 다익스트라와 달리 간선의 가중치가 음수인 경우에도 최단 거리를 구할 수 있음 (비용이 5가 아니라 -5일 경우)
    - 다익스트라 알고리즘에서는 매번 방문하지 않은 노드 중에서 최단 거리가 가장 짧은 노트를 선택해서 1단계씩 최단 거리를 구함, 벨만포드 알고리즘에서는 매 단계마다 모든 간선을 전부 확인하면서 모든 노드 간의 최단 거리를 구함

2. 라우팅 알고리즘 동작 방식

라우팅 알고리즘 동작방식_발표용.pdf