Ch5. Routing

Created	@Jun 16, 2020 2:56 PM
Tags	컴망

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/680cd85d-fc71-4451-8e12-3f47e902d9b5/Chapter_5_Routing.pptx

Introduction

IP계층(Network Layer)을 두 Plane으로 나눌 수 있다.

- **Control plane** - 라우팅 테이블을 어떻게 만들 것인지와 관련. **Routing.**

- **Data plane** - 목적지 주소를 보고, 라우팅 테이블을 참조해서, 어느 곳으로 보낼지 **Forwarding.**

라우팅이란?

- Source → Destination 길 찾아가기.
- 어떤 라우터를 거쳐서 갈 것인가?
- 어떻게 길을 찾는가? **Routing table**을 참조한다. 그럼 이 라우팅 테이블을 어떻게 만들까?

라우팅 프로토콜(라우팅 알고리즘) : 라우팅 테이블을 만드는 방법

라우팅 테이블을 만드는 방법에는 세 가지가 있다.

> 💡 프로토콜 자체는 복잡한데, 개념을 알아둬라.

Link state 프로토콜

> 💡 Link : 노드 간 직접적으로 연결된 것
> **Link state : 이 라우터와 연결된 다른 라우터들의 Link 정보. (cost(거리, 시간 돈 등), ...)**

라우터 R1 기준의 **Link state를 Flooding**한다.
= **링크 정보를 나머지 라우터들에게 다 보낸다. 모든 라우터들이 이를 수신한다.**

R2 기준의 Link state도 Flooding한다. R3도, R4도, ...

→ 시간이 지나면, 어느 라우터에서든 모든 라우터의 연결 관계를 그래프로 그릴 수 있다. 그럼 이제 임의의 두 노드 간에 데이터를 전달할 때, 어느 경로가 제일 빠를지 바로 알 수 있다.

Distance vector 프로토콜

사거리 A, B가 있는 도로.
A의 이정표를 보고, B의 이정표를 업데이트한다. B에는 B의 원래 내용에 추가로 A에만 있고 B에는 없는 내용까지 추가된다.
A의 입장에서도, 업데이트된 B의 이정표를 보고 자신 걸 다시 업뎃할 수 있다.

R1이 **주변의 라우터에게 라우팅 테이블을 전달**하면, **주변 라우터의 라우팅 테이블이 업데이트**된다. 라우팅 테이블이 변경됐다면 전달해준 라우터에게 변경된 라우팅 테이블을 다시 넘겨준다.

→ 이런 식으로 진행하면 모든 정보로 업데이트된 라우팅 테이블을 모든 라우터가 갖게 된다.

Path vector 프로토콜

R1을 거치면 X까지 도달할 수 있다. R2를 거치면 X까지 도달할 수 있다.

→ 이러한 **경로 정보를 라우터끼리 전달**하면서 라우팅 테이블을 완성한다.

Control Plane의 라우팅: 두 접근

Per-router control (전통적 방법)

- 라우터들끼리 서로 동작해서 테이블을 만드는 것
- 라우팅 알고리즘 (라우팅 프로토콜)을 통해 **라우터 자체에서 테이블을 만듦**.
- **Control plane, Data plane 자체가 라우터에 모두 구현**됨

Logically centralized control

- **라우팅 테이블 만드는 Remote Controller가 따로 있다.**
- 외부 프로그램에서 테이블을 만들어주고, 라우터에게 전달해준다.
- **라우터는 데이터 처리만** 한다.
- **Control plane은 외부**에 있고, **Data plane만 라우터**에 있다.

라우팅 알고리즘의 분류

라우팅 알고리즘은 결국 그래프에서의 최소 비용 문제.

- Global한 정보를 사용하는가, 탈중앙화된 정보를 사용하는가?
  - **Link state 알고리즘** : Global
    - 모든 라우터가 완전한 연결 정보를 갖는다.
  - **Distance vector 알고리즘** : Decentralized
    - 서로 인접한 라우터끼리의 정보만 알고 있다.
- 경로 정보가 Static한가, Dynamic한가?
  - Dynamic하다면 주기적 업데이트 필요.

Link state 알고리즘 중 하나 : 다익스트라

문제가 생길 수 있다. **Cost값이 계속해서 바뀌는 경우, 최단경로 계산이 이리저리 왔다갔다한다. (oscillations)**

- ex) Cost값이 Carried traffic인 경우, 최단경로가 계산할 때마다 계속 바뀐다.

Distance vector 알고리즘 중 하나 : 벨만-포드

Cost값이 바뀔 경우, 이 링크와 연결된 노드는 이를 알아차리고 라우팅 정보를 업데이트 한다.

- Good news travels first.
  - Cost가 낮아진 경우, 주변 라우터들에게 계속해서 정보가 전달되면서 업데이트 된다. **단, Distance vector의 변화가 없을 때는 전해준 라우터에게 다시 알리지 않는다.**
- Bad news travels slow
  - → **Count to infinity** 문제가 생긴다! 서로 Cost가 무한대까지 증가하고 나서야 못 간다는 것을 안다.

💡 **RX—A—R1—B** 연결.
* A가 가지는 정보 : X까지 A를통해 1번만에.
* B가 가지는 정보 : X까지 A를통해 2번만에.

연결이 끊어져 **RX A—R1—B** 가 된 경우, 진행을 잘 보자.
* A 정보 업뎃 : X까지 A를통해 16번만에. (inf)
그런데, B는 아직 업뎃되지 않았다. A는 다시 B의 정보를 보고는, X에 3번만에 도달할 수 있다고 생각한다.
* A 정보 업뎃 : X까지 B를통해 3번만에.
그럼 B는 A의 정보를 보고 자신 걸 업뎃한다.
* B 정보 업뎃 : X까지 A를통해 4번만에.
* A 정보 업뎃 : X까지 B를통해 5번만에.
* B 정보 업뎃 : X까지 A를통해 6번만에.
...
이렇게 가면 둘 다 inf까지 가고, 그제서야 못 간다는 걸 안다.

- 해결책 : Poisoned reverse, Split horizon, Triggered update

LS와 DV 알고리즘의 비교

수렴 속도

- **LS** : O(n^2). **진동(Oscillations) 발생 가능.**
- **DV** : 수렴 속도 다양함. 루프 발생 가능, **Count-to-infinity 문제.**

Scalable Routing

실제 네트워크에 이러한 LS, DV 방법을 적용할 수 있는가? 그렇지 않다.

전 세계의 모든 라우터들이 동일한 알고리즘을 사용하는가? 그렇지 않다.

- Network is "flat" - 모든 네트워크는 평등하다.
- **Scalable하지 않다.**
  - Destination의 숫자는 너무 많다. **라우팅 테이블에 모든 걸 저장할 수 없다.**
- **Administrative autonomy**

- 인터넷 = 네트워크들끼리 연결해 둔 또 다른 네트워크.
- 각각의 네트워크는 운영 주체를 갖는다.
- **AS (Autonomous Systems)** : 하나의 단위. "Domains"
- **Intra-AS 라우팅** : 한 AS 내부의 호스트끼리의 라우팅.
  - **게이트웨이 라우터** : 서로 다른 AS로의 엣지가 있는 라우터.
- **Inter-AS 라우팅** : 서로 다른 AS간의 라우팅
  - BGP 방법이 있다.
- **한 AS 내의 모든 호스트들은 동일한 Intra-domain 프로토콜**을 돌린다.
  AS가 다르면 Intra-domain 프로토콜도 다를 수 있다.
- Intra-AS 라우팅 알고리즘, Inter-AS 라우팅 알고리즘 두 개를 복합적으로 사용해서 라우팅 테이블을 만든다.

Intra-AS : AS 내부에서 어느 라우터를 거쳐야 하는지만 알면 된다.
Inter-AS : 어느 AS를 거쳐야 하는지만 알면 된다.

Inter-AS : 서로 다른 AS간 라우팅

AS1의 라우터에서 AS2의 라우터로 데이터를 보내기 위해서는?

1. **어느 AS를 통해야** AS1 → AS2로 갈 수 있는지 확인.
2. 어느 AS로 갈지 결정했으면, AS1 안에 있는 **어느 게이트웨이 라우터를 통해야** 그 AS로 갈 수 있는지 확인.
3. **모두 확인됐으면 그 게이트웨이 라우터로 보내면 된다.**

BGP (Border Gateway Protocol)

AS간 프로토콜로 가장 많이 사용됨.

자신의 존재를 subnet (서브 네트워크)에 알려주는 역할을 한다.

- **eBGP** (exterior BGP)
  - 서로 다른 AS간 통신

- **iBGP** (interior BGP)
  - 한 AS 내부의 호스트 간 통신

BGP 프로토콜들은 서로 **BGP 세션**이라는 것을 만든다.

- **BGP 세션** : 두 BGP 라우터 사이의 반영구적인 TCP 연결

BGP는 Path vector 프로토콜을 사용한다.

- 라우터끼리 경로 정보를 전달한다.

BGP 경로 선택 옵션

1. Policy에 따라서
2. 최단 AS경로
3. **Hot potato routing** : 최대한 지금 있는 AS에서 빨리 벗어나도록. 가장 가까운 게이트웨이 라우터로 전달.
4. ...

Intra-AS = IGP (Interior Gateway Protocols)

- **RIP** : Routing Information Protocol
  - DV 알고리즘 사용
- **OSPF** : Open Shortest Path First
  - LS 알고리즘 사용
- **IGRP** : Interior Gateway Routing Protocol

SDN : Software Defined Networking

> 💡 Remind: 기존의 방식은 라우터 내에 라우팅 프로토콜이 내장되어 있다. 라우터 내의 라우팅 프로토콜은 우리 임의대로 수정할 수 없다.

항상 같이 등장하는 기술이 **NFV (Network Function Virtualization)**이다.

- **NFV** : 네트워크 노드들의 함수들을 가상화해둔 것

Control plane, Data plane이 분리되어 있다.

Control plane도 두 부분으로 나누어볼 수 있다.

- **Network-control applications**
  - 라우팅, 로드 밸런싱, ... 등을 수행
- **SDN Controller**
  - 인터페이스 레이어, 상태 관리 레이어, 커뮤니케이션 레이어 등 존재.
  - ODL(OpenDayLight) 컨트롤러, ONOS 컨트롤러 등이 있다.

ICMP : Internet Control Message Protocol

호스트와 라우터들이 **네트워크 레벨의 정보를 전달**할 수 있도록 해 준다.

- **Error reporting : 문제가 생겼을 때 정보 전달.**

> 💡 라우터를 거칠 때마다 TTL이 감소한다. TTL이 0이 되면? 패킷을 drop한다. 그러면 Source에다가 라우터가 **TTL이 0 돼서 패킷 드랍했다고 알려야 한다.**
> → 이러한 이벤트를 알리는 프로토콜이 ICMP.

- **Echo request/reply** : 네트워크에 대한 정보를 알고 싶을 때, **ping 보내고 받는 프로토콜.**

ICMP도 일종의 **Network layer(IP계층)에 포함되는 프로토콜**이다.

- ICMP 메시지를 전달할 때, IP계층의 도움을 받아 이 패킷에 패킷 헤더가 붙어서 전달된다.

ICMP를 이용하는 예시 1 : ping

서버-클라 만드는 게 아니라, ICMP 기능을 활용한다.

ping은 그냥 나만 가지고 있고, 상대에는 ping 없어도 된다. 이럴 때, 상대 컴퓨터가 살아있는지 어떻게 확인하는가?

- A에서 B로 ping 실행 → A의 ICMP 모듈 호출 → IP계층을 통해 echo_request 메시지 전달 - B의 IP계층에서 수신 - B의 IP계층에서 ICMP 프로토콜 메시지를 뜯어봤더니 echo_request → A에게 다시 echo_reply를 전달 → ...

ICMP를 이용하는 예시 2 : traceroute

tracert B 하면 B로 갈 때 어느 호스트를 거쳐가는지 다 뜬다.

이것도 마찬가지로, tracert 프로그램은 나만 가지고 있다.

- Remind: TTL 0 되면 라우터는 패킷 버리고 원래 Source에 메시지를 전달한다.
- 이를 이용한다.
- TTL 1인 패킷 보내고, 2인 패킷 보내고, 3인 패킷 보내고, … → 이렇게 하면 경로에 있는 라우터들의 정보가 순서대로 ICMP 메시지로 들어오게 된다.
    - Timeout은? TTL이 0이 됐다는 정보가 안 들어온 것

SNMP : 네트워크 관리에 사용

여러 기기의 정보를 MIB(Management Information Base)로 저장한다.

MIB에 저장된 정보를 가지고 오는 프로토콜 : SNMP