Ch3. Internet Protocol

Created	@Jun 21, 2020 1:04 AM
Tags	

Network Layer

송신 측 → 수신 측으로 Segment를 **Datagram/Packet으로 Encapsulate해서 전달**해 준다.

NL의 핵심 기능 두 가지

- **Forwarding** : 라우터 내에서 패킷 이동
- **Routing** : Source에서 Dest까지 길찾기

Network Layer: Data plane, Control plane

- **Data plane** : 실제로 데이터를 어떻게 보내는가를 관리.
- **Control plane** : 데이터를 제대로 보내기 위해서 통신 장비들끼리 제어해야 하는데, 어떤 제어를 해야 하는가? 관리.

Buffering

평균 버퍼링 : $RTT \times C$ (RTT=0.25s=250ms, C: Link capacity)

Flow가 N개라면, $(RTT \times C)/\sqrt{N}$

Scheduling mechanisms

스케쥴링 : 패킷이 버퍼링되고 있는 상황에서 (buffer라는 큐에 패킷이 쌓여있는 거), 새 패킷이 도착하면? → **어떤 패킷을 먼저 보내야 하는가?**

- **FIFO Scheduling : 먼저 온 애가 먼저 나간다.**

    - Discard policy : 버퍼 큐가 가득 찼거나, 가득 찰 위기에 처했을 때, 새 패킷이 들어오면 뭘 버려야 하는가? → Tail drop(들어오는 거 버리기), Priority(우선순위 기반), Random(랜덤)
- **Priority Scheduling : 우선순위가 높은 애를 먼저 소화.**

> 💡 **Starving : 항상 우선순위가 높은 애만 들어오면, 우선순위가 낮은 애는 전송되지 않고 버퍼에 계속 남아있게 된다.**

    - **Round-Robin Scheduling (RR)**
        - 큐가 여러 개 있으면, **큐 자체를 위에부터 순서대로 돌아가면서 하나씩 빼서 전송하자.**
    - **Weighted RR Scheduling (WRR)**
        - **RR의 각 큐에 Weight 부여**
        - 가중치는 **서비스될 패킷 개수 비율**을 나타냄
        - **패킷 길이가 가변이면 부적합.**
    - **Weighted Fair Queueing (WFQ)**
        - 가중치는 **서비스될 비트 개수 비율**을 나타냄

IP Services

인터넷은 Best Effort Service이다. Out-of-order, Packet loss 등이 발생할 수 있다.

The Internet network layer

레이어들을 Remind.

- **Physical Layer** : 약속된 신호를 보낸다. 0101...1010... 그걸 약속된 내용으로 검증 및 해석하는 것.
- **Link Layer** : 에러 없는지 체크 및 에러 있으면 재전송 신호 (**Error control**), 수신측의 상황 고려하여 속도 동기화 (**Flow control**) (왜? 수신측의 상황을 고려하지 않고 막 보내면 오버플로우 나니까)
- **Network Layer** : 라우팅, Forwarding table, IP protocol, ICMP protocol

IP Datagram format

- 32bits
  - **ver (4bit)** : v4/v6
  - **header length (4bit)** : 어디서부터 어디까지가 헤더인지.
    - **헤더 길이는 가변이다.** Dest IP 주소까지는 20바이트로 고정이다. 가변적으로 변하는 부분은 Options 부분.
  - **Type of Service (8bit)** : 데이터그램이 요청하는 서비스는 다양하다.
    - **Delay** : 최대한 **딜레이 없이** 보내라.
    - **Throughput** : 1초당 처리 양. **처리양 높이는 쪽으로** 보내라.
    - **Reliability** : **신뢰성 높이는 쪽으로** 보내라. 이건 에러가 있거나 잃어버리면 안 된다.
    - **Monetary** : **비용이 저렴한 경로로** 보내라.
  - **length (16bit)** : 헤더 포함해서 **전체 데이터그램의 길이**
    - → **데이터그램의 최대 길이= 2^16.** TL → NL 넘어올 때 전달해줄 수 있는 데이터 크기에는 한계가 있다.
- 32bits (Fragmentation/Reassembly 관련)
  - **MTU : Maximum Transmission Unit**. 패킷(데이터그램)이 너무 커서 Framing을 할 수 있는 크기가 안 되는 상황이면 → **데이터그램을 MTU 크기만큼 잘라서 보내야 한다.**
    - MTU는 네트워크마다 다르다.
  - **패킷의 데이터 부분을 토막내서 → 자른 토막들을 다시 패킷화시키는 것.**
  - 조각에는 각각 헤더들을 또 붙여서, 조각난 이 패킷들을 보내는 것. 패킷들이 독립적으로 날아가니까, 받는 쪽에서 다 받아서 순서대로 Reassembly해 줘야 한다.
  - 각 조각들의 헤더는 거의 비슷하게 붙는다. (Source, Dest, option 등...) **그럼 헤더에서 각 조각마다 달라져야 할 부분은?**
  - **16-bit identifier (16bit)**
    - 이 조각이 같은 데이터그램에 속하는지 확인.
  - **flags (3bit)**

        - 안 씀 / **Don't** / **More**
        - **Don't=1** → fragmentation 하지 마라. 그럼 못 보내잖아? 드랍시켜야 한다. 라우터에게 이벤트 발생해서 송신측에 알려줘야 한다. 이것도 ICMP가 받는다.
        - **Don't=0** → fragmentation 해도 된다.
        - **More=1** → 아 얘 말고도 원래 패킷이 토막난 애들이 더 있구나
        - **More=0** → 아 얘가 마지막이구나
    - **fragment offset (13bit)**
        - 각 조각들의 순서 맞춰주기 위한 정보
        - 처음 토막이 0이라면, 이 토막은 그 다음으로 얼마나 떨어져있는 데에서 시작하는 데이터인지.
- 32bits
    - **TTL (8bit) : 남은 hop 수**
    - **upper layer (8bit)** : 상위 레이어 타입
    - **header checksum (16bit)** : 헤더에 대해서 에러가 있는지 없는지.
        - 계산은 32비트를 16비트씩 잘라서 두개 xor 후 bit flip.
- **Source IP 주소 (32bit)**
- **Dest IP 주소 (32bit)** : Dest IP를 routing table에서 lookup해서, 어디로 보낼지 결정하는 것. (Routing)
- **options (가변)** :
    - **Record Route** : 패킷이 어느 라우트를 거쳐 왔는지 기록
    - **Source Route** : Strict / Loose
        - **Strict** : R1 R2 R3 R4면 딱 거기거기 갔다오는 거 얘기
        - **Loose** : 다른 데 중간에 거쳤다와도 됨
    - 옵션은 또 많다.

IP addressing: introduction

라우터에는 인터페이스가 여러 개 꽂힌다. 인터페이스에는 네트워크(호스트)가 여러 개 연결되어 있다. 컴퓨터에는 인터페이스 카드 보통 한 개 있다. 그래서 컴퓨터마다 주소 한 개 할당받는 것.

- IP주소 : Network address, Host address로 나뉜다.
- *Classful : 네프 필기 참고*
  - Class D : Multicast
  - Class E : Future Use

1:1 → **Unicast**
1:ALL → **Broadcast**

Address 부족 → 해결?

1. Class 단위 addresssing (Classful addressing) → **Classless addresssing**
   필요한 host 개수만 고려해서 뒤에서부터 2의 몇 승으로 끊어서, 그 나머지 앞부분은 다 class 부분으로 쓰자
2. **CIDR : Classless InterDomain Routing**
   0.0.0.0**/00** → **슬래시 뒤는 서브넷 파트의 비트를 나타냄. 앞의 이 비트만큼이 네트워크 부분이다.**

DHCP : Dynamic Host Configuration Protocol

- = Plug-and-play
- 서버로부터 동적으로 IP주소를 받아온다. 영구적인 IP주소가 아니다.
- Address 부족 해결. (컴퓨터 안 쓰는 애들은 할당 안 하고, 현재 쓰는 애들만 할당하니까)