

# 4/20 IP

이전까지는 네트워크 이하의 레이어를 블랙박스라고 보고 있었음

세그먼트 전송 과정 자체에 대해서 이야기하지는 않았는데 이제 그 프로토콜을 이야기할 것

## IP

Internet Protocol

네트워크 레이어에는 이거 하나만 있음. 세그먼트 배송에 대한 일을 관장함

각 라우터에는 Network, Link level까지의 stack이 있다.

## Network Layer

소스부터 도착점까지 패킷을 보냄

패킷 전송에 관한 이야기

### 라우터 : 패킷이 들어오면 그 패킷을 목적지 방향의 다른 라우터로 전달

1. 목적지를 확인해야 함
2. 목적지로 가는 outgoing link를 확인해야 함 → 라우터 안에 테이블(forwarding table)이 있어서 dest에 따른 다음 링크가 정해져 있음 (forwarding)
  - routing: forwarding table을 만드는 알고리즘

실질적인 라우팅 테이블은 정확한 주소보다는 주소 범위로 관리되어 있음

Longest Prefix Matching : 주소가 bit로 되어 있을 때 가장 구체적으로 일치하는 prefix를 설치함

## IP Packet Header

- ip 주소는 32비트 체계 사용.
- time to live: router을 하나 거칠 때마다 -1됨. 0이 되면 파기됨.  
table이 잘못되었을 때 무한루프를 돌 수 있는데, 이게 있어서 그런 일을 좀 막을 수 있음
- upper layer: data 안에 들어가는 것. tcp/udp segment? receiver 측에서 이것을 사용해  
서 응답을 보낼 것

- options

패킷의 헤더는 20바이트, tcp 헤더도 20바이트. 그래서 tcp 패킷은 결국 ip 헤더, tcp 헤더 해서 40바이트임. 실제로 패킷을 조사해보면 상당수의 패킷이 달랑 40바이트인데, data가 없다는 내용임. 이건 tcp ack임!!!

## IP Address (IPv4) - unique 32bit num

$2^{32}$ 개의 IP주소를 가질 수 있음. dotted-quad notation을 쓰고, 8비트씩 4개로 끊어서 읽고 이걸 10진수로 읽는다는 뜻임. 예를 들어 12.34.158.5고 최댓값은 255.255.255.255임.

IP 주소는 host보다는 거기 들어 있는 network interface를 지칭한다. 그래서 랜카드 여러개 꽂으면 ip주소가 여러개일 수 있음. 이런 컴퓨터의 예시가 바로 라우터. 즉 라우터 하나로 통하는 ip주소는 여러개일 수가 있다.

- 배정 방법:
- forwarding table을 통한 계층화: hierarchical addressing: IP Prefixes
- a.b.c.d에서 a, b, c는 네트워크 id, d는 host. 같은 network 안에 있는 host는 같은 network id를 가진다. == subnet id == prefix == network id.
- ip 주소 밑에 subnet mask가 있고 이거는 11111111.11111111.11111111.0이어서, subnet id가 24비트임을 알려주는 것임.
- scalability improved: 같은 network = 같은 prefix., easy to add new hosts.

## Classful Addressing: 과거에 어떤 식으로 ip주소를 배분했는지

인터넷의 구성 요소는 네트워크임. 즉 인터넷은 네트워크의 네트워크임.

prefix를 받더라도 네트워크 사이즈에 맞는 prefix를 줘야하지 않니?

- 예전에는 class를 두고 배정했음.
- class A: 0\* - prefix가 /8bit, host를  $2^{24}$ 개 보유 가능, class A는 0으로 시작해야 해서  $2^7$ 개만 있을 수 있음. 근데 개발한 사람들이 선점함.
- class B : 10\* prefix가 /16bit (대충 비슷, 선점당함)
- class C: 110\* prefix가 /24bit - 호스트를 255개까지만 보유할 수 없음.
- class D: 11110\*
- class E: 111110\*

그래서 클래스를 없애고 90년대 중반쯤에 CIDR가 나옴

## CIDR: Classless Inter-Domain Routing

위에서 말한 Mask를 도입해 자릿수를 비트 단위로 끊을 수 있게 됨 → 어느 정도 필요한 갯수만큼 할당이 가능해짐. forwarding table이 작아진다. (원래 class로 해서 여러개 받다가...)

## Longest Prefix Match Forwarding

forwarding table이

4.0.0.0/8

201.10.6.0/23

201.10.0.0/21 이고 패킷 dest.add가 201.10.6.17이면, 더 긴 곳까지 일치하는 걸로 골라서 보낸다. 라우터는 이 matching을 계속해서 하게된다.

## Creating Forwarding Table

Destination-based forwarding

routing algorithm으로 채운다.(다음 시간)

## Subnet

라우터를 거치지 않고 접근이 가능한 호스트의 집합

같은 서브넷 파트를 가진 장치의 인터페이스의 집합

물리적인 라우터 하나는 여러 개의 ip주소를 가진다. 즉 여러 개의 서브넷에 속해있다.

그래서 라우터가 없다면 다른 서브넷으로 갈 수 없게 된다.

## IPv6, 128bit (1996)

$2^{128}$  hosts를 표시할 수 있어서...대박임.

근데 2015년 현재 IPv4를 아직도 사용한다. 이거 고갈됐을텐데? 왜? ip주소를 공유하기 때문이다. 그 트릭이 바로 NAT다.

## NAT - Network Address Translation

내부에서는 유일하지만 다른 네트워크까지 포함하면 그렇지 않은 식으로 주소를 할당하고, NAT기능이 있는 라우터에만 유일한 ip를 할당한다.

근데 돌아올땐 어떡함? 포트번호를 보고 찾는다. 즉 segment header를 참조해야 한다. 근데 이거 지금 우편배달부가 송수신 주소를 바꾸고, 봉투 까서 내용을 바꾼것임. (layer violation)

포트번호를 호스트 찾는데 썼으면 호스트의 포트번호는 어떻게 찾아야함? 그래서 이걸 쓰면 그 내부의 네트워크에서는 서버를 운영할 수가 없게됨. 이 경우에는 포트가 여러개일 수 없기 때문임.