데이터통신과 네트워킹

Data Communication & Networking Ch. 11

동아대학교 컴퓨터AI 공학부
S03 301-01호실 (이양민 교수 연구실)
2024.11 (Kakao ID: yanwenry)

CHAPTER

11

인터넷 프로토콜(IP)

Section

SECTION 01

인터넷 프로토콜(IP)의 작업

1. 네트워크 계층과 IP의 이해

- IP의 작업은 매우 많음.
  - 네트워크 계층의 가장 중요한 작업은 라우팅.
  - IP 주소 체계와 도메인 주소를 관리.
  - IP – MAC 주소변환.
  - 네트워크를 여러개의 서브넷으로 분할하고 다른 기종의 LAN을 연결하는 역할.

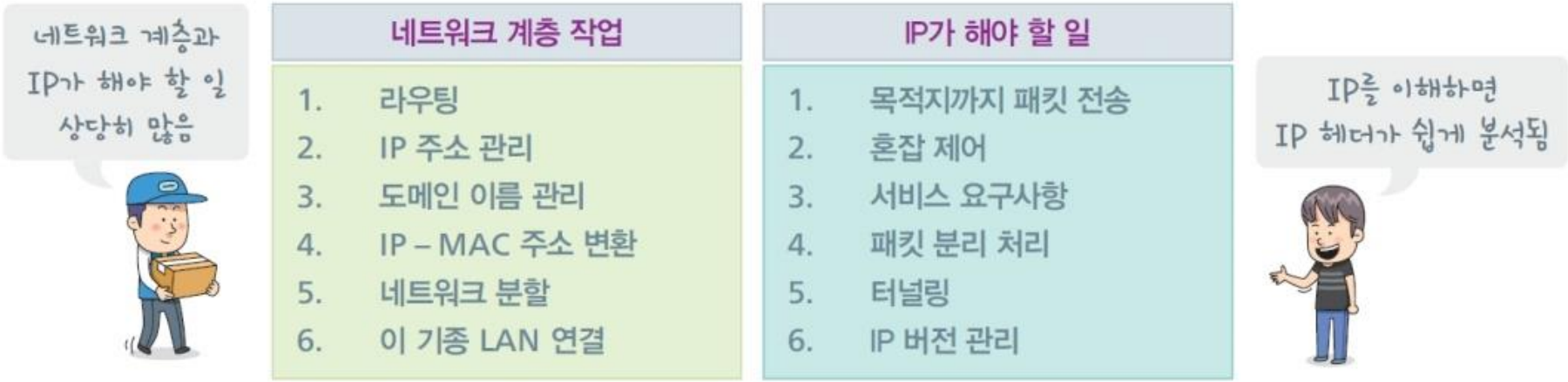


그림 11-1 네트워크 계층과 IP가 해야 할 일

인터넷 프로토콜(IP)의 작업

2. 서비스 요구사항(Quality of Services; QoS)

- 네트워크에게 요구되는 사양은 신뢰성reliability, 지연delay, 지터jitter, 대역폭bandwidth
- 각 응용프로그램 마다 요구사항은 다름.
  - 이메일 서비스: 높은 신뢰성을 필요, 지연, 지터, 대역폭은 낮아도 상관없음.
  - 화상회의: 지연, 지터, 대역폭 모두 높음을 필요로 함. 몇몇 패킷은 잃어 버려도(신뢰성이 낮아도) 됨.
- 서비스 요구사항(QoS)을 표시하도록 되어 있으나 무의미해짐.

표 11-1 응용 프로그램별 요구사항

응용 프로그램	신뢰성	지연	지터	대역폭
이메일	높음	낮음	낮음	낮음
파일 전송	높음	낮음	낮음	중간
웹 접속	높음	중간	낮음	중간
인터넷전화	낮음	높음	높음	낮음
화상회의	낮음	높음	높음	높음

인터넷 프로토콜(IP)의 작업

3. 혼잡제어(congestion control)

- 혼잡제어congestion control란 혼잡이 발생하는 곳의 흐름을 변화시켜 혼잡을 완화시키는 작업.
  - 호스트의 타임아웃 시간을 늘리는 것.
  - 타임아웃보다 더 효과적인 방법은 패킷을 천천히 전송하는 것 -> 슬라이딩 윈도우 크기 줄이기.
  - 패킷과 패킷의 전송간격을 조정.

표 11-2 혼잡 완화 방법

혼잡 완화 방법	관련 주제
타임아웃 시간 늘리기	
슬라이딩 윈도우 크기 줄이기	초크 패킷
패킷 전송 간격 조정	토큰 버킷 알고리즘

인터넷 프로토콜(IP)의 작업

- 초크 패킷chock packet은 윈도우의 크기를 줄일 때는 사용되는 패킷 -> 분필을 뜻하는 초크는 혼잡제어에 사용되는 빈 패킷을 의미.
- 윈도우의 크기를 줄이자는 의미로 상대방 호스트에게 보내는 빈 패킷이 초크 패킷 -> 초크 패킷을 받은 호스트는 윈도우를 줄임.
- IP 헤더에서 초크 패킷의 역할을 하는 필드가 ECN.

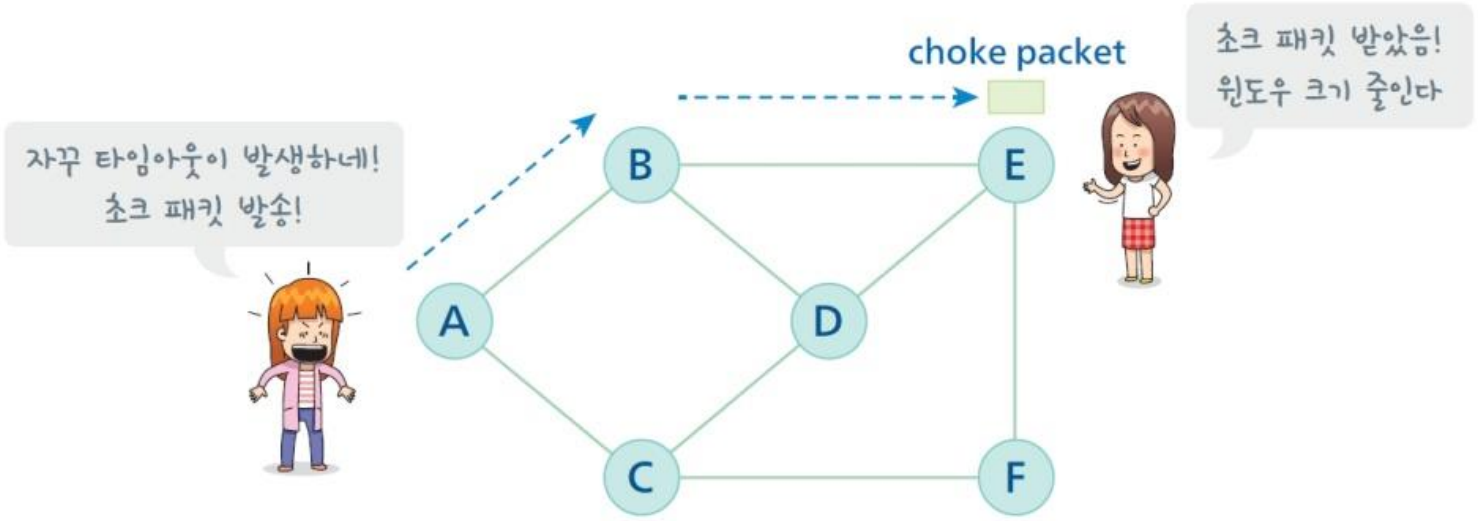


그림 11-2 초크 패킷(chock packet)

인터넷 프로토콜(IP)의 작업

- 패킷의 흐름을 일정간격으로 조절하면 혼잡이 완화됨 -> 윈도우 크기가 10인 경우 10개의 패킷을 0.1초 간격으로 보내는 것과 1초 간격으로 보내는 것은 혼잡에 미치는 결과가 다름.
- 버퍼를 이용한 혼잡제어 알고리즘이 토큰 버킷token bucket 알고리즘
  - 일정하지 않은 간격으로 도착하는 패킷을 버킷에 넣고, 보내는 쪽에서는 일정간격을 맞추어 천천히 패킷을 처리하는 알고리즘.

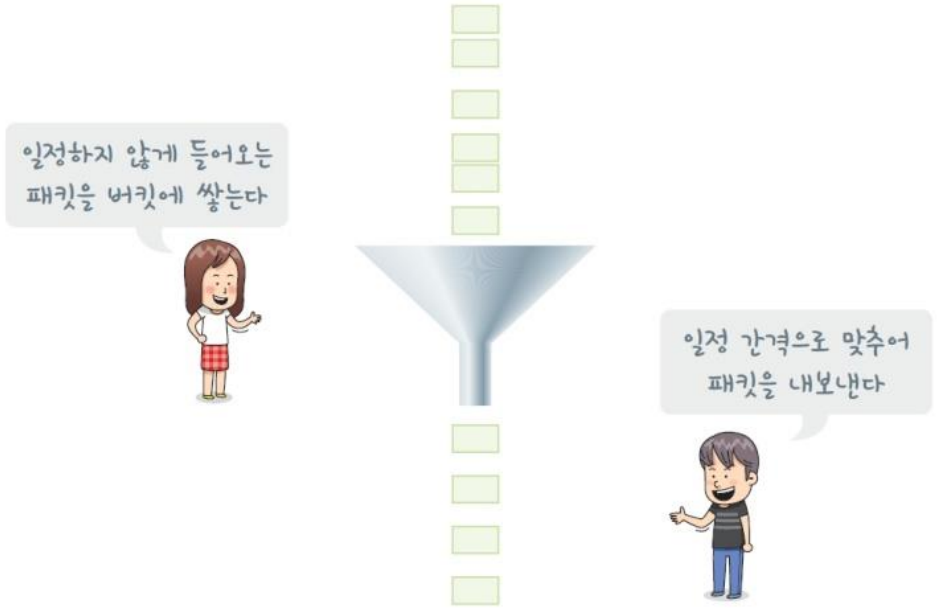
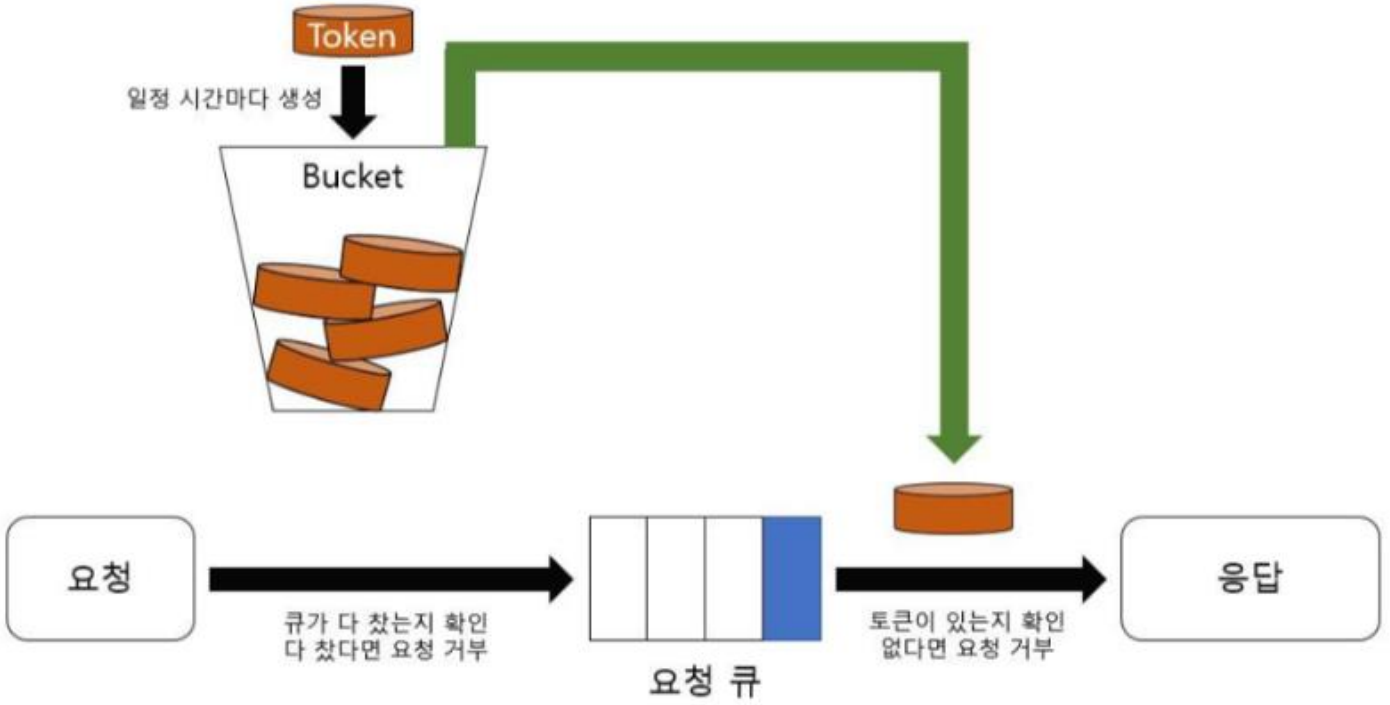


그림 11-3 토큰 버킷(token bucket) 알고리즘

인터넷 프로토콜(IP)의 작업

- 토큰 버킷(Token Bucket)이란
  - 양동이 안에 토큰을 넣는다는 뜻으로, 양동이에 토큰이 있을 때만 요청을 처리하는 알고리즘
  - 일정한 시간마다 양동이에 제어용 토큰이 생성되고, 요청이 들어온다면 양동이에 토큰이 있는지 확인하고 응답하는 방식
- 토큰 버킷 구조
  - 예를 들어, 양동이에 담을 수 있는 최대 토큰이 100개이고, 1초 당 넣는 토큰이 1개라면 1초 당 처리할 수 있는 요청의 수는 1개가
  - 만약 100초 동안 요청이 들어오지 않았다면 양동이에는 100개의 토큰이 쌓이게 되고 이후에 들어오는 100개의 요청을 연속 처리할 수 있음 (양동이가 가득 찼다면 다음으로 생성되는 토큰은 폐기)
  - 요청 큐가 가득 차거나 양동이에 토큰이 없어서 요청을 거부할 땐 429 Too Many Requests를 반환



인터넷 프로토콜(IP)의 작업

4. 패킷 단편화

- LAN 마다 패킷의 처리능력이 다를 뿐 아니라 다룰 수 있는 패킷의 크기도 다름 -> 처음 보낸 패킷이 여러개로 조각나서 배달되는 경우가 발생.
- LAN이 끝나는 지점에서 원래의 패킷으로 합쳐서 보내주면 큰 문제는 없음 -> 문제는 그림의 왼쪽과 같이 3개로 분리된 채로 인터넷을 돌아다니는 경우.

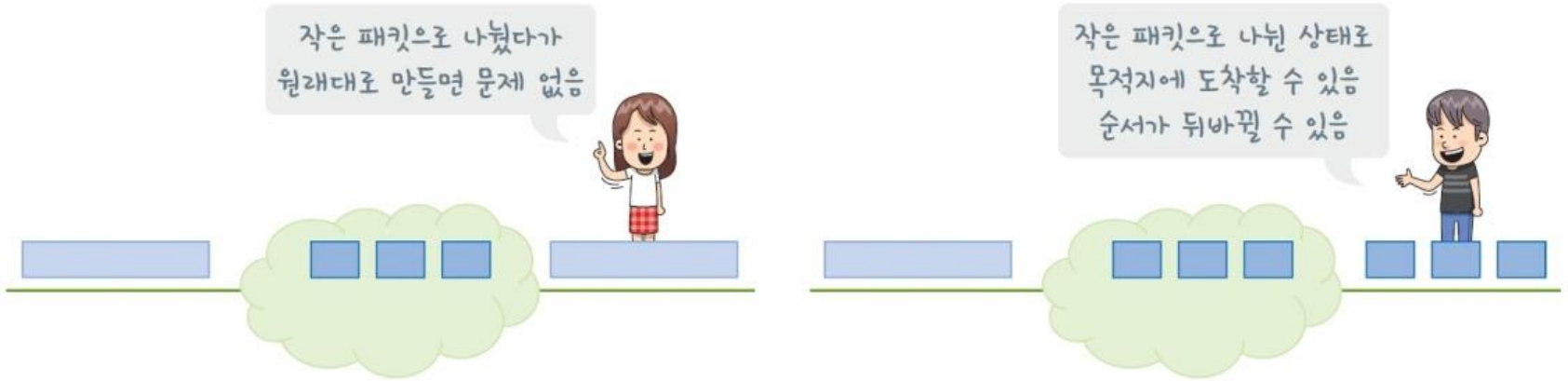
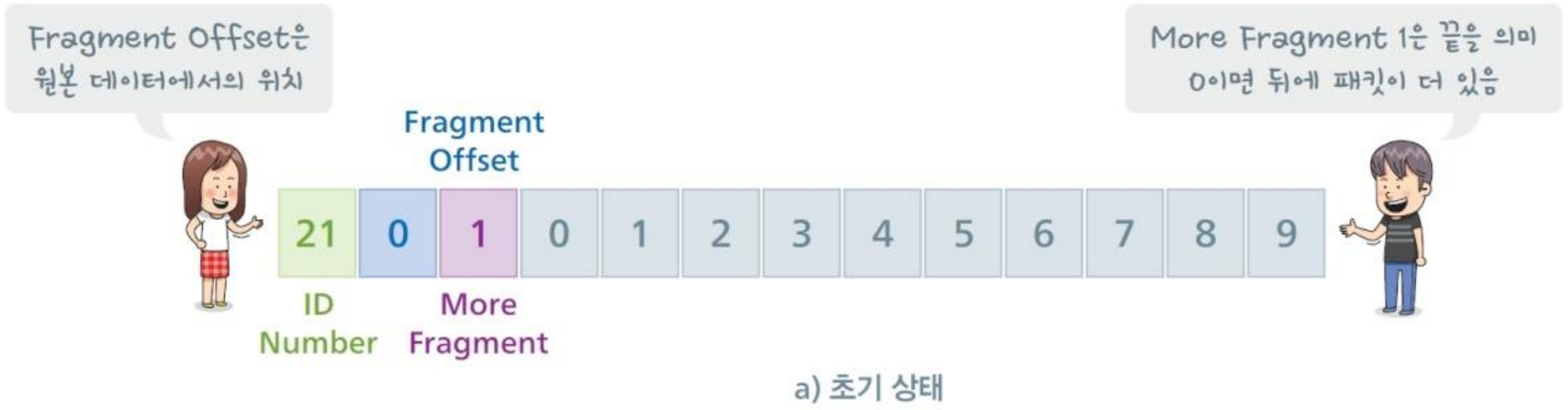


그림 11-4 패킷이 여러 조각으로 나뉘는 문제

인터넷 프로토콜(IP)의 작업

- 패킷의 맨 앞의 3개의 필드는 패킷 분할과 관련하여 헤더에 있는 값.
  - 가장 맨 앞이 Identification Number(ID Number) 필드: 그림에서는 21 -> 잘려나간 패킷이 여러 개 있는 경우, 같은 패킷에 있는 데이터인지를 확이 할 수 있는 필드.
  - 두 번째 필드는 Fragment Offset: 옵셋offset은 기준위치로부터 얼마큼 떨어졌는지를 나타냄 -> Fragment Offset은 분할되기 전 패킷 데이터로 부터 얼마큼 떨어진 위치에 있는 데이터인지를 알려줌.
  - 3번째 필드는 More Fragment이다. More Fragment가 1이면 맨 마지막 패킷이라는 의미, 0이면 패킷이 분할되어 뒤따르는 패킷이 더 있다는 의미.



a) 초기 상태

인터넷 프로토콜(IP)의 작업

- b) 2개로 분할 그림은 원본이 2개로 분할된 경우
  - ID Number 21로 두 개의 패킷이 하나의 패킷에서 분할 -> 앞의 패킷의 Fragment Offset 0 -> 이는 원래 데이터의 0번째 위치라는 의미.
  - 뒤의 패킷의 7은 원래 데이터의 7번째 위치에 있었다는 것을 알려줌.
  - 앞 패킷의 More Fragment 0이 뒤로 패킷이 더 있다는 것을 알려주고, 뒷 패킷의 More Fragment 1은 패킷의 끝을 알려줌.

b) 2개로 분할

인터넷 프로토콜(IP)의 작업

- c) 3개로 분할, 순서가 바뀌어 도착한 경우
  - 가장 앞 패킷은 Fragment offset이 0인 맨 앞의 패킷, 중간의 Fragment offset이 7이 원본의 마지막 패킷, Fragment offset이 3이 원본의 두번째 패킷.
  - 받는 쪽에서는 패킷이 3개로 분할되었는지 혹은 4개나 5개로 분할되었는지 알지 못함 -> 중간 패킷의 More fragment 1을 보면 더 이상 분할된 패킷이 없다는 것을 알게 됨.

c) 3개로 분할(순서가 뒤바뀌어 도착)

그림 11-5 패킷의 분할에 사용하는 필드 및 의미

인터넷 프로토콜(IP)의 작업

5. 터널링(tunneling)

- 서로 다른 종류의 통신망을 사용하여 인터넷에 접속하는 경우 -> 집에 있는 무선 공유기 고장으로 스마트폰의 핫스팟 기능을 이용하여 노트북을 인터넷에 연결하는 경우.
- 무선 전화통신망은 IP 패킷과는 다른 종류의 헤더를 사용 -> 패킷이 인터넷 망이 아닌 구간을 통과할 때 어떻게 해야 하는가의 문제가 발생.
- 터널링 혹은 IP 터널링 -> 기존의 IP 패킷을 무선 전화망에서 사용하는 패킷에 집어넣고, 무선 전화망과 인터넷이 연결되는 곳까지 보냄. 이후 IP 패킷으로 전송.

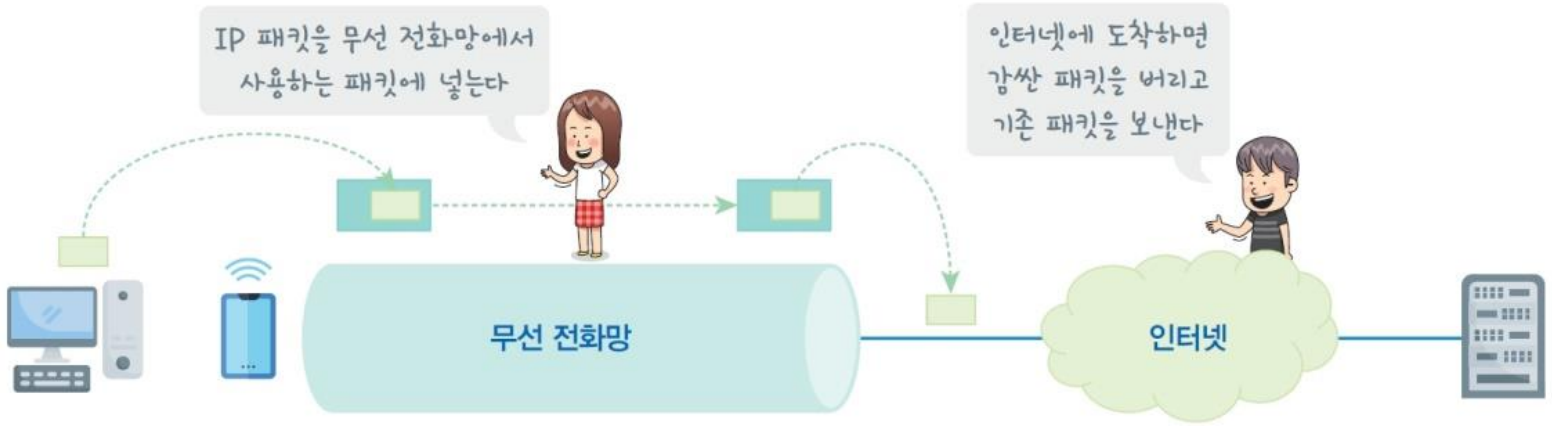


그림 11-6 터널링

SECTION 02

IP 헤더 분석

1. IPv4 헤더

- IPv4의 헤더 : 32비트(4바이트)를 기준으로 나눔.
- 위에서부터 다섯 번째 줄까지 필수부분이고, 나머지는 옵션부분, IP 헤더의 필수 부분은 5 x 4바이트 = 20바이트이고, 옵션은 없을 수 있으며, 있더라도 4바이트씩 증가.

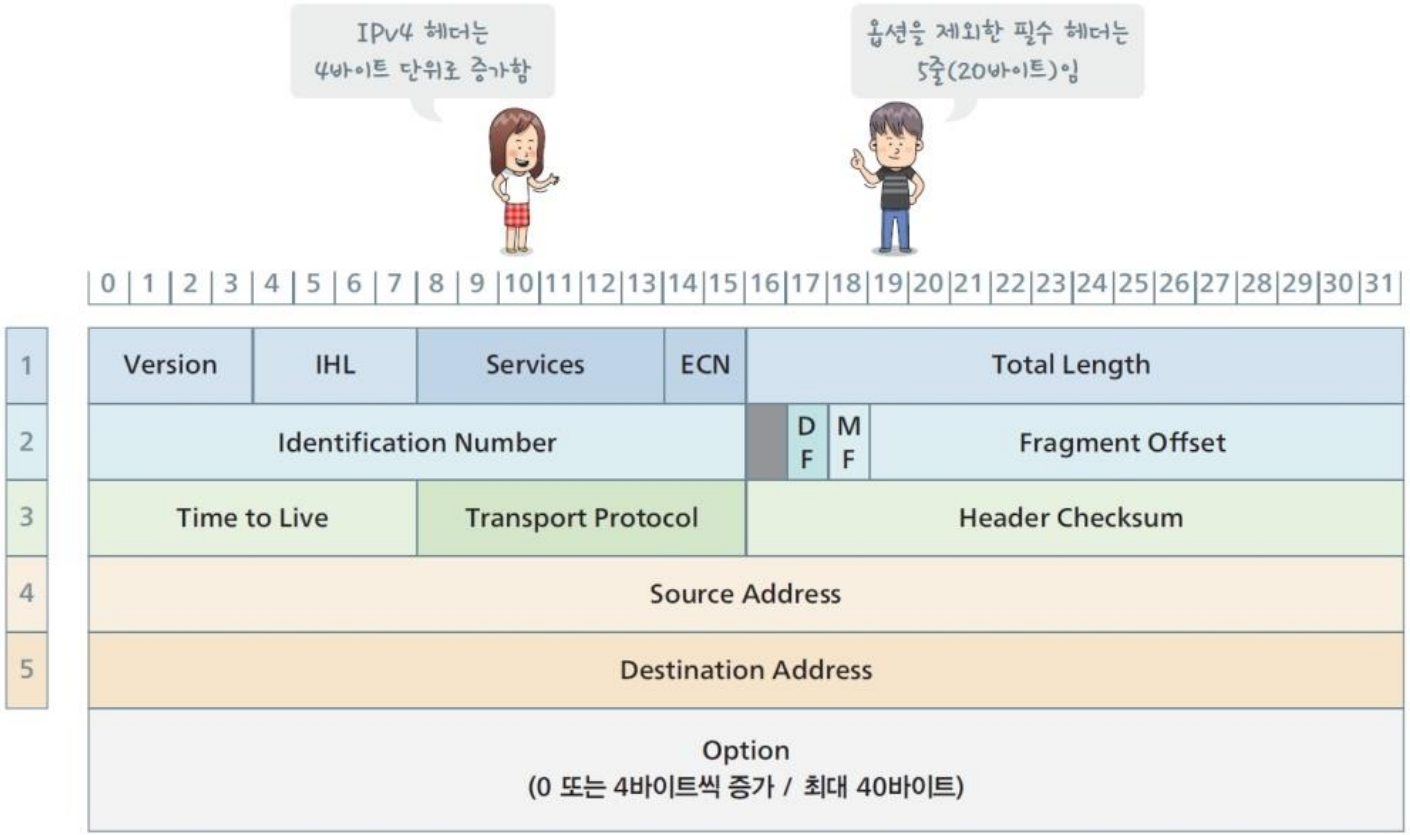


그림 11-7 IPv4 헤더

IP 헤더 분석

- Version : IP 버전 번호, IPv4임으로 4가 들어가 있음.
- IHL : IHL은 IP Header Length의 약자. 옵션을 포함한 IP 헤더 길이를 나타냄. 언제나 4바이트 기준. 옵션이 없는 경우 IHL에는 5가 들어가 있음 -> 1줄이 4바이트임으로 20바이트 크기의 헤더가 됨.
- Services : 1번 줄의 8번에서 15번 비트까지는 원래 Quality of Service를 명시. 일반적인 인터넷에서는 무시.
- ECN : 14번과 15번 비트는 ECN Explicit Congestion Notification 필드이다. ECN은 혼잡제어와 관련된 필드. ECN은 ECT ECN Capable Transport 비트와 CE Congestion Experienced 비트 두 개로 구성. ECN는 TCP의 요청에 의해 사용되는 필드이며 14장 TCP와 소켓 프로그래밍에서 자세히 설명.
- Total Length : Total Length는 헤더를 포함하여 전체 패킷의 크기를 나타냄 -> IHL과 달리 Total Length에 있는 값은 바이트를 의미. 200이면 헤더를 포함하여 전체 크기가 200바이트라는 의미.

IP 헤더 분석

- Identification Number: 패킷 번호를 의미하며, 분할 된 패킷 중 원본 패킷이 무엇인지 확인하는데 사용. 같은 Identification Number를 가진 패킷은 원본에서 분할된 패킷.
- DF : DF는 Don't Fragment의 약자이며, 목적지의 호스트가 나눠진 패킷을 하나로 만들 수 없는 경우 패킷을 분할하지 못하도록 만든 필드. 이 필드가 1이 되면 패킷을 분할하지 못함.
- MF : MF는 More Fragment의 약자이며 앞서 패킷 단편화에서 설명한 필드. 0이면 뒤로 패킷이 더 있다는 의미이며, 1이면 마지막 패킷이라는 의미.
- Fragment Offset : Fragment Offset은 앞서 패킷 단편화에서 설명한 필드. 현재 가지고 있는 데이터가 원본 데이터에서 어느 위치인지를 나타낸다. 바이트로 표시되며, 0이면 맨 앞의 패킷.

IP 헤더 분석

3	Time to Live	Transport Protocol	Header Checksum

- Time to Live: 패킷이 목적지에 도착하지 못하고 인터넷에서 계속 살아 있는 경우가 좀비패킷 -> 좀비패킷이 많아 질 경우 네트워크에 부담이 커짐. 패킷이 살아 있을 수 있는 시간을 기록한 것이 Time to Live. TTL에 명시된 시간을 넘어서 네트워크를 돌아다니다 라우터를 만나면 해당 패킷을 폐기. 라우터 마다 시간의 오차가 있을 수 있어 홉Hop으로 표시하는 추세.
- Transport Protocol: 전송 계층에는 TCP 뿐 아니라 UDP와 같은 여러 종류의 프로토콜들이 있음. 해당 패킷을 전송계층에 있는 어떤 프로토콜에게 전달해야하는지 명시.
- Header Checksum: 헤더에 에러가 있는지 없는지를 검사하는 16비트 코드가 Header Checksum에 들어 있다. 에러검사 코드 중 체크썸Checksum 방식을 사용. 헤더만 검사.

- IP헤더 4번 줄의 Source Address는 보내는 호스트의 IP 주소를 나타내며, 5번 줄 Destination Address는 받는 호스트의 IP 주소를 의미.

IP 헤더 분석

2. IPv6 헤더

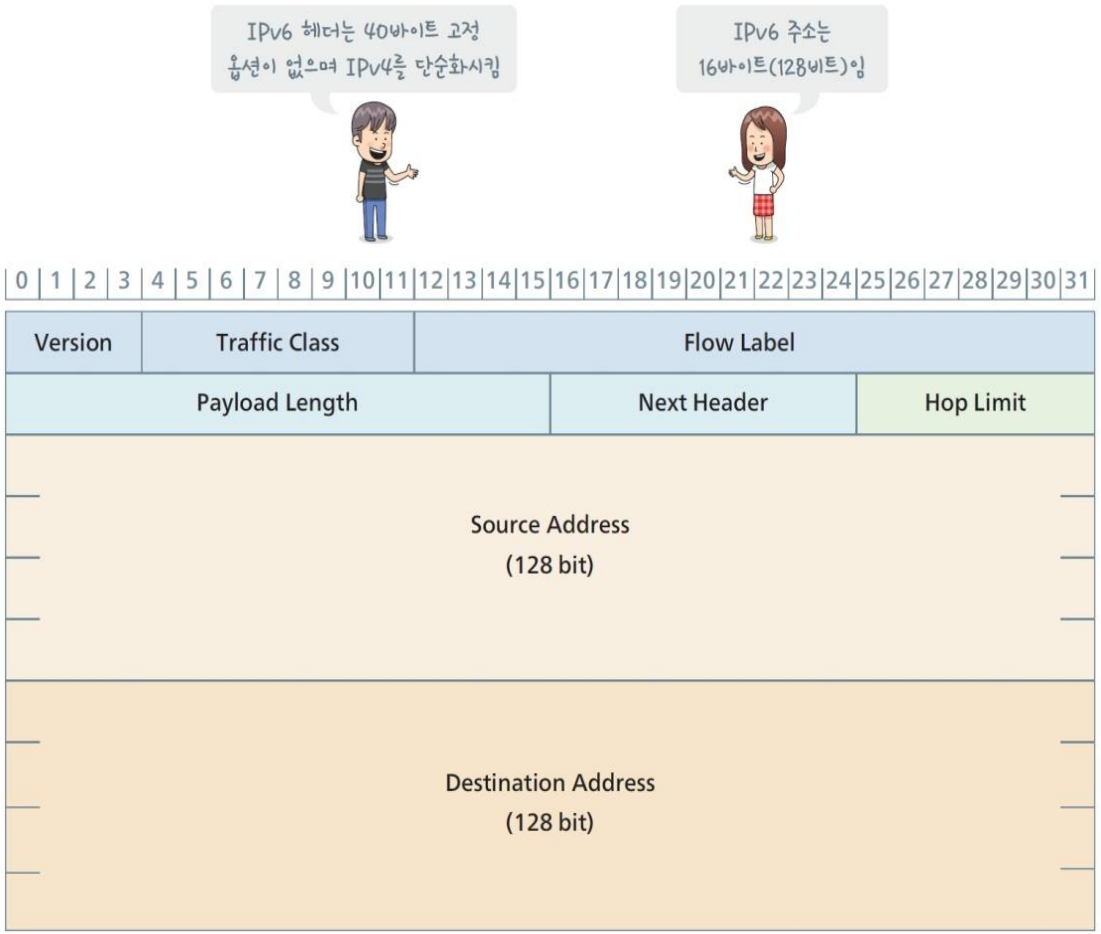


그림 11-8 IPv6 헤더

IP 헤더 분석

- IPv6의 크기는 40바이트 고정이다. IPv4와 같은 옵션은 없음.
- IPv6 헤더에서 Source Address 16바이트(128비트)와 Destination Address 16바이트.
- 1번 줄 -> Version 필드에는 버전 번호 6이 들어감.
  - Traffic Class에는 Quality of Service가 들어감. Traffic Class라고 이름을 바꾼 이유는 사물 인터넷의 경우, 응용 프로그램에 따라 급히 처리해야 하는 패킷이 있을 수 있기 때문.
  - Flow Label은 IPv6가 설계될 당시에 다양한 용도가 제안 되었지만 아직까지는 뚜렷한 표준이 적립되지 않은 영역. 일반적으로 0으로 설정.

- 2번 줄 -> Payload Length는 헤더를 제외한 데이터의 크기를 바이트로 표시. IPv6는 헤더가 40바이트로 고정 크기이기 때문에 IPv4에 있던 IHL 필드가 필요 없으며, 데이터의 크기만 명시하면 됨.
  - Next Header는 IPv4의 Transport Protocol의 이름이 변경된 것. 따라서 상위계층에 전달될 프로토콜의 헤더를 가리킨다.
  - Hop Limit는 IPv4의 Time to Live가 변경된 것이다. 따라서 패킷이 살아 있을 수 있는 최대의 홉 개수가 적혀 있음.

IP 헤더 분석

- IPv6의 주소표기 문제 -> 128비트(16바이트)의 주소를 10진수로 표현하면 길이가 매우 길어짐. 그래서 4개 16진수(2바이트)를 하나로 묶어서 8개의 덩어리로 표시. 각 덩어리(16진수)를 콜론(:)으로 구분

  9000:0000:0000:0000:1F23:2D5C:2323:34FF

- IPv6의 경우, 주소의 중간에 0이 많이 나타남. 이 경우 생략을 할 수 있음.

  9000::1F23:2D5C:2323:34FF

- 인터넷은 IPv4가 대부분이고 IPv6는 나중에 만들어졌기 때문에 IPv6와 IPv4를 같이 표시하는 경우가 있음. 따라서 IPv6에 연결된 IPv4주소는 콜론(:) 두 개와 10진수로 표시.

  9000::1F23:2D5C:2323:34FF**::258.231.20.12**