

네트워크 계층 (3~4)

🔗 URL

▼ 목차

Network Address Translation

NAT의 동작과정

NAT의 문제점

IPv4의 문제점

해결책

Dynamic Host Configuration Protocol (DHCP)

DHCP

DHCP의 동작 과정

1. DHCP discover

2. DHCP offer

3. DHCP request

4. DHCP ACK

Router의 프로세스들

IP datagram의 헤더

IP:ICMP, IPv6

ICMP: Internet Control Message Protocol

IPv6

IPv6 datagram

Routing Algorithm : Link State

그래프로 나타낸 라우터

Link State Algorithm

Dijkstra Algorithm

Network Address Translation

NAT의 동작과정

- 로컬 네트워크의 호스트가 데이터를 보낼 때
 - 라우터의 ip/포트로 변환해서 보낸다

NAT의 문제점

1. 디자인 문제점 - 레이어 디자인이 무너짐 (패킷의 ip수정, 데이터 수정)

2. 현실적인 문제점 - NAT에서 Port를 이용하여 내부 주소를 찾을

- 결국 NAT도 완전한 해결책은 아니다.

IPv4의 문제점

1. 주소 공간의 문제
2. security 문제

해결책

- IPv6도 20여년이 지났기 때문에 완전하지가 않기 때문에 쉽게 넘어갈 수 없는 것이 현실
- 미래의 요구사항에 맞출 수 있는 동적인 IP protocol
- 아직 정해진 것은 없다

Dynamic Host Configuration Protocol (DHCP)

DHCP

- 서버로부터 동적으로 주소 (IP, mask, router, DNS)를 받아오는 프로토콜
- host가 서버에 ip주소를 요청 (DHCP request) 하면 dhcp가 host에게 일정시간 동안 ip주소를 대여 (DHCP ack) 해주는 개념
- 고정 IP를 사용하는 경우는 필요 없다
- 그러나, 고정 IP를 사용할 경우 IP 주소의 낭비가 발생할 수 있음

DHCP의 동작 과정

1. DHCP discover

- 초기에 DHCP 서버를 찾아내는 과정
- src ip: 아직 없음
- dest 255.255.255.255: subnet나의 모든 멤버에게 메시지를 보냄 (broadcast)

2. DHCP offer

- 67번 포트를 열어두고 있는 DHCP 서버만 응답
- dest: 255.255.255.255: 아직 요청 host가 ip가 없는 상태이므로 broadcast로 보냄
- host는 포트 번호 (68)이랑 transaction id로 해당 응답을 받음

3. DHCP request

- offer를 받아들인다는 의미로 request 보냄
- 아직 ip주소를 받은 건 아니므로 src ip 가 없다
- transaction ID: offer를 받아들인다는 의미로 +1

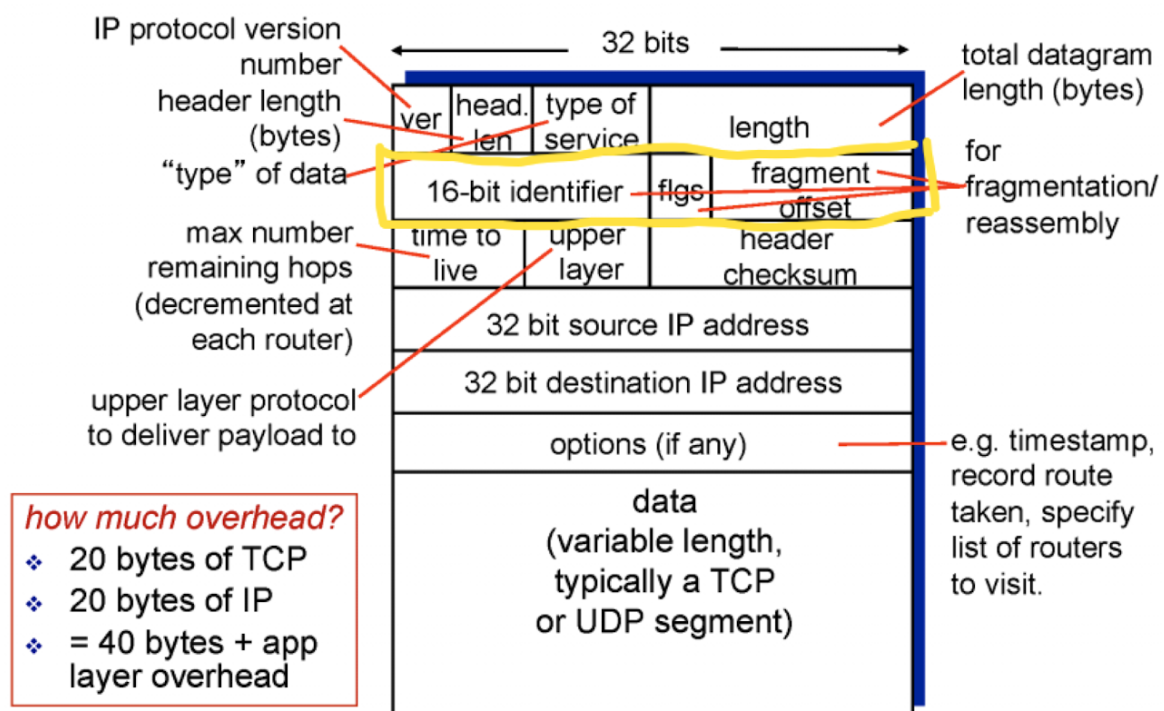
4. DHCP ACK

-

Router의 프로세스들

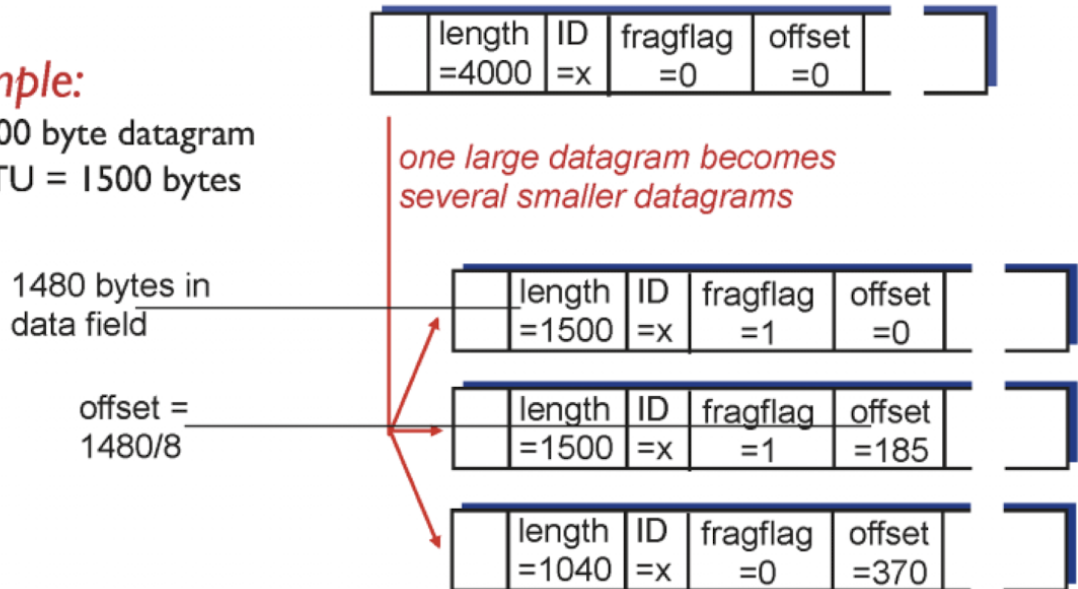
- forwarding, DNS, DHCP, NAT, firewall 등

IP datagram의 헤더



example:

- ❖ 4000 byte datagram
- ❖ MTU = 1500 bytes



- Max Transfer Unit(MTU): 각 링크 레이어의 (wifi, 3G, LTE...) 처리 가능한 최대 유닛 크기
- 처리할 수 있는 MTU보다 더 큰 패킷이 들어오면 MTU사이즈에 맞는 독립적인 Frame 으로 나뉘어져 진행된다
- 20byte의 헤더가 각각 들어간다

IP:ICMP, IPv6

ICMP: Internet Control Message Protocol

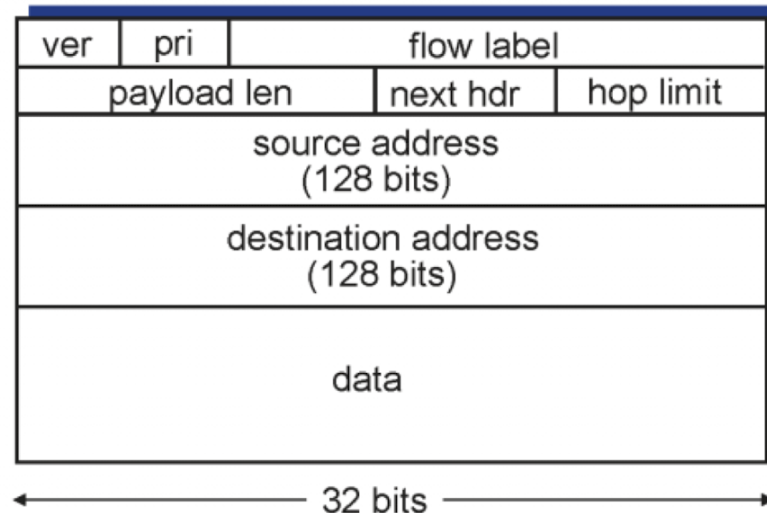
- 사용자 데이터가 아닌 네트워크 상의 증상 (데이터 유실, TTL expire 등등)을 나타내는 control message를 운반하는 프로토콜
- ip 패킷의 데이터 부분에는 발생한 증상에 대한 정보를 담는다
- 네트워크 상태 진단 등에 이용

IPv6

- 우선, 본격적으로 도입 될 가능성이 높지는 않다

IPv6 datagram

priority: identify priority among datagrams in flow
flow Label: identify datagrams in same “flow.”
 (concept of “flow” not well defined).
next header: identify upper layer protocol for data

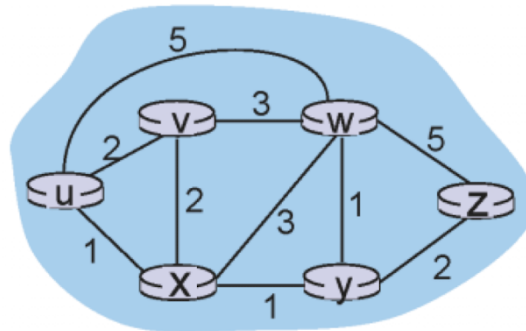


- 128bit의 주소 공간
- flow label 등의 헤더는 정의만 되었고 구체적 사용 방법은 논의되지 않았다.

Routing Algorithm : Link State

그래프로 나타낸 라우터

Graph abstraction



graph: $G = (N, E)$

N = set of routers = $\{ u, v, w, x, y, z \}$

E = set of links = $\{ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) \}$

- 결국은 최단경로 구하기
 1. 전체 네트워크의 연결 상태를 아는 경우
 - Link State Algorithm
 2. 연결된 인접 라우터만 아는 경우
 - Distance Vector Algorithm

Link State Algorithm

- 모든 노드가 자신의 연결 상태 (Link state)을 전체 네트워크에 broadcast로 전파한다

Dijkstra Algorithm

Example

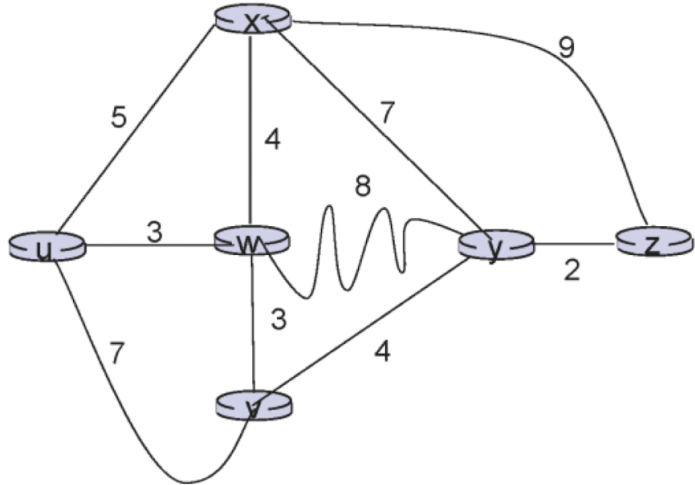
1 **Initialization:**

2 $N' = \{u\}$
 3 for all nodes v
 4 if v adjacent to u
 5 then $D(v) = c(u,v)$
 6 else $D(v) = \infty$

7

8 **Loop**

9 find w not in N' such that $D(w)$ is a minimum
 10 add w to N'
 11 update $D(v)$ for all v adjacent to w and not in N' :
 12 **$D(v) = \min(D(v), D(w) + c(w,v))$**
 13 /* new cost to v is either old cost to v or known
 14 shortest path cost to w plus cost from w to v */
 15 **until all nodes in N'**



Network Layer 4-78