CS스터디 네트워크계층 3, 4

날짜	@2022년 11월 8일

IPv4의 문제점

1. 보안성이 약함
2. 공간이 부족함

그렇지만 바꿀순 없음 WHY?

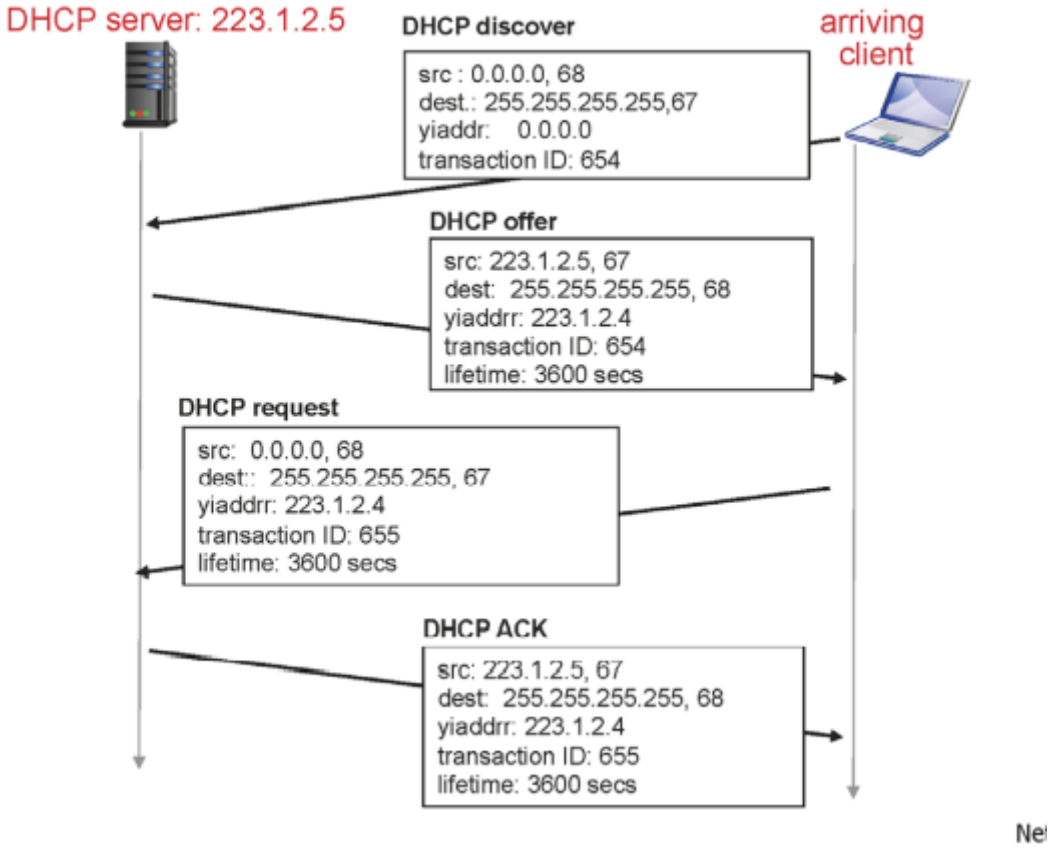
⇒ 이미 라우터들이 ipv4에 맞추어 설계되어있어서 생태계를 유지중임 바꾸려면 전세계가 동시에 바꿔야하지만 각 회사별로 가지고있는 라우터가 달라 힘듬.

DHCP(Dynamic Host Configuration Protocol)

ex) 대학은 고정 IP 10000개 → 이용자(학생)가 요청하면 DHCP를 통해 IP를 렌탈해주고 일정시간 사용 후 반납

DHCP 에서 client는 address / network 정보를 어떻게 알 수 있을까?

⇒ DHCP server message



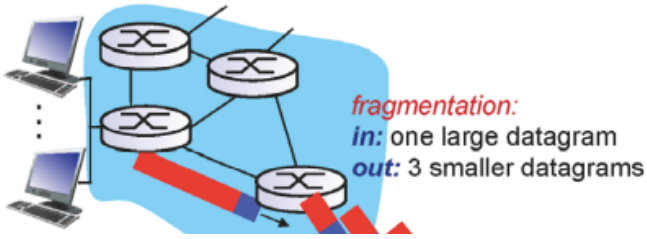
1. 클라이언트가 67번 주소를 DHCP 서버에 요구한다.(broadcast로)
2. 서버는 Clinet에게 67번 주소가 남아 있음을 확인하고 67번 사용할 것 인지 offer를 보낸다.
3. 요청자만 DHCP offer를 의미있게 받아 들이고 DHCP 서버로 request를 보냄
4. 서버가 최종 승인

DHCP를 사용하고 있으니까 내가 실제로 사용하는 정보랑 외부로 보내는 IP정보는 다르다!.

MTU

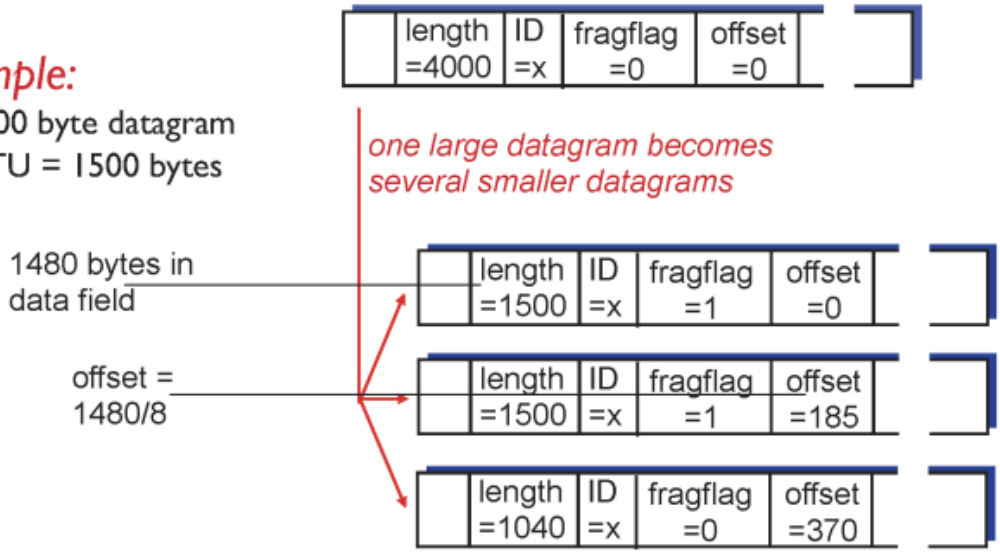
각 장비마다 다 다른 사이즈를 가지고있음(네트워크를 받을 수 있는 사이즈가 다르다). → 그래서 사이즈를 분할하여서 독립적으로 패킷을 나눠 보냄 → 다시 받을때 합쳐져서 정보를 받음.

- network links have MTU (max.transfer size) - largest possible link-level frame
  - different link types, different MTUs



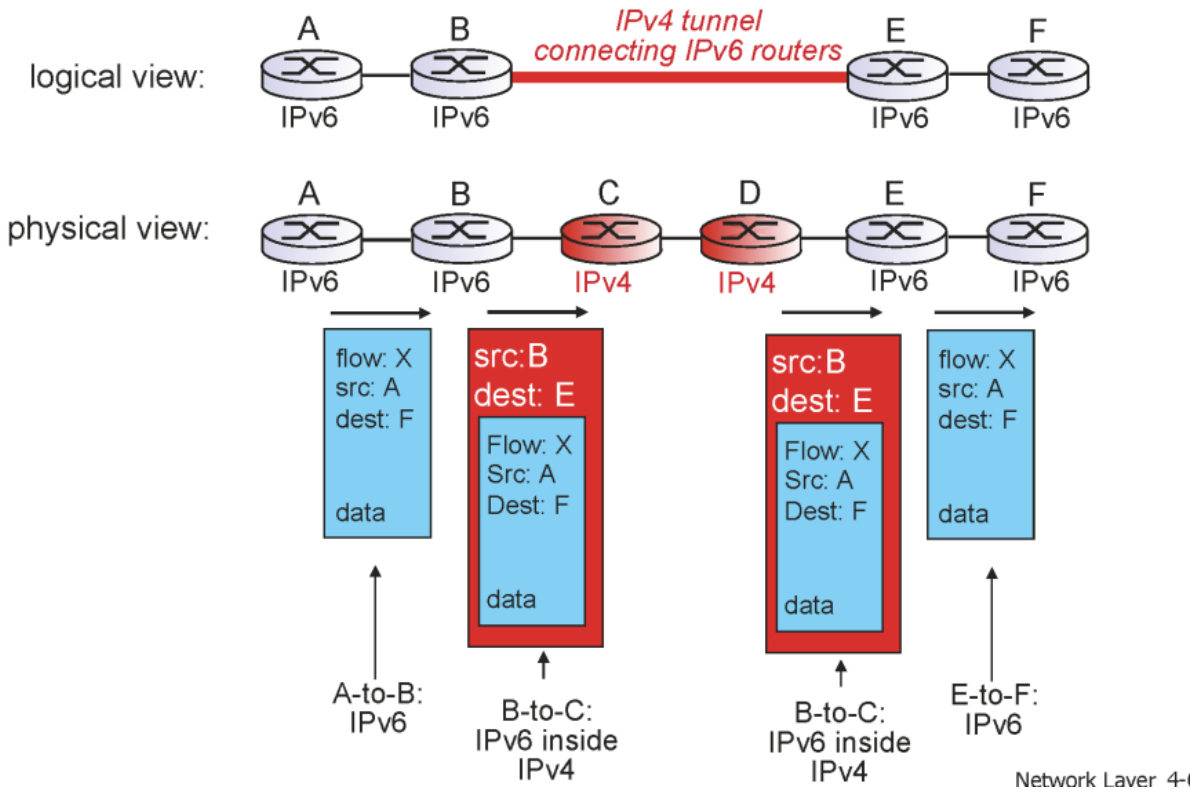
example:

- 4000 byte datagram
- MTU = 1500 bytes



Tunneling

과도기 같은시기에 IPv4와 IPv6 를 둘다 해석가능한 라우터가 필요하다. 연결을 위해!



routing algorithm의 목적

목적지 까지 최단경로를 찾기 위해

<방법>

1. 모든 네트워크 정보를 알고 있는 경우(link state)

   → 각 노드들이 자기의 링크 state를 전체 네트워크에 정보를 제공한다.

   → 다익스트라를 이용하여 최단경로 탐색 가능!!

2. 이웃을 통해 정보를 공유해서 알아내는 경우(distance vector)

다익스트라 heapq로 구현한 코드 올리기

```python
def dijkstra():
    global D
    INF = 0xffffff
    L = [0] * (V+1)          #prim기준 MST
    D = [INF] * (V+1)        #prim에서 key 같은 존재
    start = 1
    D[start] = 0

    for v, w in adjL[start]:    # 각 간선별 가중치 먼저 저장.
        D[v] = w

    for _ in range(V):
        minV = INF
        for i in range(V+1):
            if L[i] == 0 and D[i] < minV:
                u = i
                minV = D[i]
        L[u] = 1
        for v, w in adjL[u]:
            D[v] = min(D[v], D[v]+w)



V, E = map(int ,input().split())
adjL = [[] for _ in range(V+1)]
for _ in range(E):
    u, v, w = map(int, input().split())
    adjL[u].append([v,w])

dijkstra()
```