링크 레이어 계층

호스트와 라우터들을 노드로 칭함

네트워크 계층은 end to end 였다면 링크 계층은 인접 노드간의 데이터 전송을 의미한다.

(진짜 딱 자기 옆에 있는 노드들만 전송을 하면 됨)

EX) 서울에서 교토 가기

서울 – 인천 – 오사카 – 교토

1. 여행객은 데이터그램

2, 여행 수단: 커뮤니케이션 링크

즉, 각 여행지에 도달하기 위해 타는 버스 비행기 기차 이런 것들을 링크 계층프로토콜이라고 생각하면 된다. 그리고 탑승 수단이 다른 것처럼 다른 노드로 갈 때 커뮤니케이션 링크의 종류도 다르다. (와이파이, 이더넷, 광섬유 등)

링크에는 두가지 종류가 존재한다.

1. point to point: 1:1로 혼자 쓰는 링크라고 생각하면 된다.

2. broadcast: 전체에게 데이터를 보내는 것

링크 계층 서비스

\* 링크 계층에서 불리는 데이터 형식은 frame, IP계층의 데이터그램을 감싼것이다.

\* 맥주소를 사용하며(얘도 목적지 소스 주소가 있다) 목적지 주소를 앞에다 둠으로써 목적지 호스트의 맥주소와 다르면 뒤에 정보는 읽어볼 필요도 없이 걍 거른다

\* 맨뒤에 checksum을 둠으로써 해당 프레임에 에러가 있는지 확인한다.

\* 중간에 Packet 데이터가 존재

여기서 중요한건 맥주소는 아이피 주소와는 좀 다르다는 것이다.

링크 계층에서 중요한 것: 어뎁터 혹은 NIC라고 한다.

하는 일: 얘가 패킷을 프레임으로 감싸고 에러 체크 비트를 추가한다. 또는 받는 측이라면 데이터그램을 추출하고 위 계층으로 올린다.

그리고 얘가 맥주소를 가지고 있는데 맥주소는 글로벌하게 유니크한 존재이다. (Ipv4 처럼 ip주소를 동적으로 받지 않음) – IEEE가 맥 주소를 할당해준다.

IP주소와 MAC 주소

맥주소는 평면 구조이다. 컴퓨터가 어느 네트워크에 있던 간에 상관없이 걍 그냥 그대로 값을 갖는다. (주민번호)

맥주소는 다른 노드에 도달을 하게되면 소스와 목적지 주소는 바뀌게 된다. 왜냐하면 노드에서 노드끼리 데이터가 통신 되어야 하니까 주기적으로 바뀜

IP주소는 계층 구조이다. 네트워크에 따라 서브넷도 바뀌고 IP 주소 전체가 바뀌게 된다. 즉 IP 주소로 이 PC가 어느 네트워크를 사용하고 있는 지 알 수 있다. (집 주소)

IP주소는 절대 변하지 않는다. 집 주소로 배송와야되는데 다른 주소로 배달 된다 생각하면…. 절대 안된다.

ARP

\* 라우터의 맥주소를 알아야 외부 subnet으로 갈수 있다. ARP를 통해서 ip address와 MAC address를 맵핑 시킨다. (DNS랑 비슷한 작업을 수행한다.)

\* ARP 테이블: 아이피와 맥주소, TTL이 매핑되어있다. 각 노드마다 ARP 테이블이 존재

\* ARP테이블 채우기

1. a가 데이터그램을 b에게 보내길 원함(하지만 A에게는 B의 맥주소가 없다)

2. a는 b의 아이피 주소를 포함한 ARP 쿼리를 브로드 캐스트한다. (LAN안에 있는 모든 노드들이 받는다. Dest mac address=ff-ff-ff-ff-ff-ff)

3. b는 ARP 패킷을 받고 a에게 b의 맥 주소를 보낸다.

ARP 테이블의 경우 관리자의 간섭없이 자동적으로 채워지고 없어지고 한다.

발신자 end는 DHCP를 통해서 first hop의 아이피 주소를 알기에 이 아이피를 활용하여 라우터를 찾고 라우터는 프레임을 까고 아이피를 봐서 포워딩 시킨다.

브로드 캐스트: 하나의 링크에 여러 노드들이 접근하려고 한다. (이때 당연히 충돌이 생길 가능성이 높고 이를 해결해야할 필요도 있다.)

즉 간섭이 멀티 어세스 프로토콜에서는 문제가 됨.

해결방법

1. channel partitioning

- 채널을 작은 단위(시간, 주파수)로 나누어 사용

- 각 노드에게 작은 단위를 할당하여 접근할 수 있도록 함

2. random access

- 채널을 나누지 않고 충돌을 허락한다. 왜냐면 재전송하면 되니까.

- 충돌로부터 회복할 수 있도록 한다.

3. taking turns

- 차례대로 노드에게 데이터 전송 권한을 부여한다. 충돌이 생기지는 않지만 오래걸린다.

TDMA

시간을 타임 슬랏으로 나눈 후 각 타임 슬랏에 데이터를 전송한다.

하지만 사용하지 않는 타임슬랏이 있어 idle한 슬롯이 발생한다.

FDMA

주파수 구역 별로 슬랏을 나눠서 각 슬랏에 분배

얘도 idle한 경우가 생김

CSMA : 랜덤한 접근을 통해 멀티플 엑세스를 하자

\* 내가 데이터를 보내기전에 들어보고 전한다. 채널이 바쁘지 않다면 보내고 아니면 기다리는 상태

\* 하지만 둘다 눈치보다가 말해야지하고 충돌 날 수 도 있다.

\* 프로파게이션 딜레이에 의해 보냈는 지 모르고 자신의 데이터를 보낼수도 있음

CSMA/CD: 기존 기능에 충돌 감지능력을 준다. (충돌이 발생한 것이 판단 되면 더 이상 말 ㄴㄴ)

충돌이나면 랜덤한 숫자를 고르고 그만 큼 기다린다.

\* 하지만 랜덤한 숫자가 너무 크면 너무 오래기다리고 너무 작으면 다시 충돌이 발생한다.

\* 충돌된 횟수에 따라서 기다리는 범위가 달라진다.