네트워크 컴퓨팅 1주차 과제

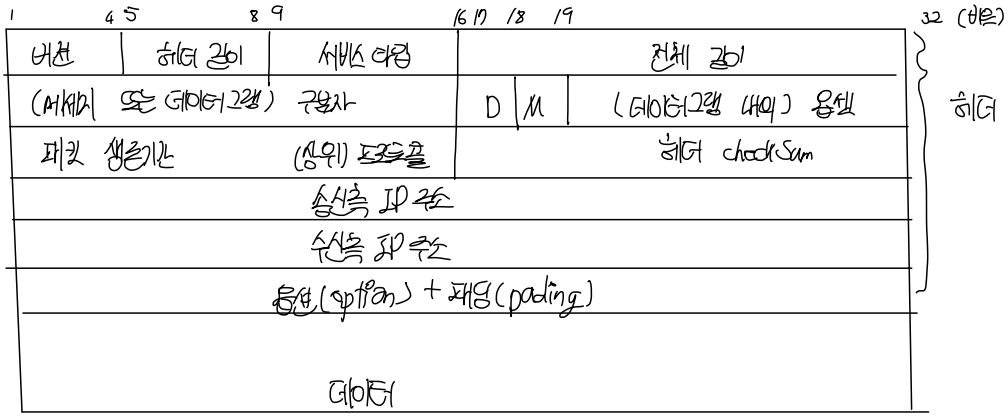
1603019 박세현

1. 이더넷 프레임 구조를 기술하라

Preamble (7byte), SFD (1byte)	DA (6byte)	SA (6byte)	Type (2byte)	Info (46~1500byte)	FCS (4byte)

- Preamble : 수신측에서 하드웨어 비트 동기를 맞추기 위한 준비 신호
- SFD (Starting Frame Delimiter) : 다음 바이트부터 프레임의 시작을 알림.
- DA (Destination Address) : 목적지 MAC 주소, 앞 3바이트는 제조회사 식별코드이며 나머지 3바이트는 제조회사의 목적지 식별을 위한 코드이다.
- SA (Source Address) : 송신측 MAC주소, 자신의 주소를 ROM에 기억, 전원이 ROM에서 읽어 레지스터에 저장. 프레임 송신 시 이 레지스터를 읽어 프레임의 SA에 삽입.
- EtherType : MAC 프레임 다음의 데이터 부분에 상위 프로토콜의 종류를 표시
- DATA : 상위 프로토콜 데이터가 위치, 최대 허용길이 (MTU)는 1500byte, DA부터 FCS까지 전체길이가 64byte 이상이어야 한다는 규정을 준수
- PAD : 최소길이 규정을 만족하지 못할 경우 '0'으로 채운다.
- FCS (Frame Check Sequence) : 프리앰블과 SFD를 제외한 MAC 프레임의 비트열의 이상을 검사.

[현재 IP 데이터그램 구조]

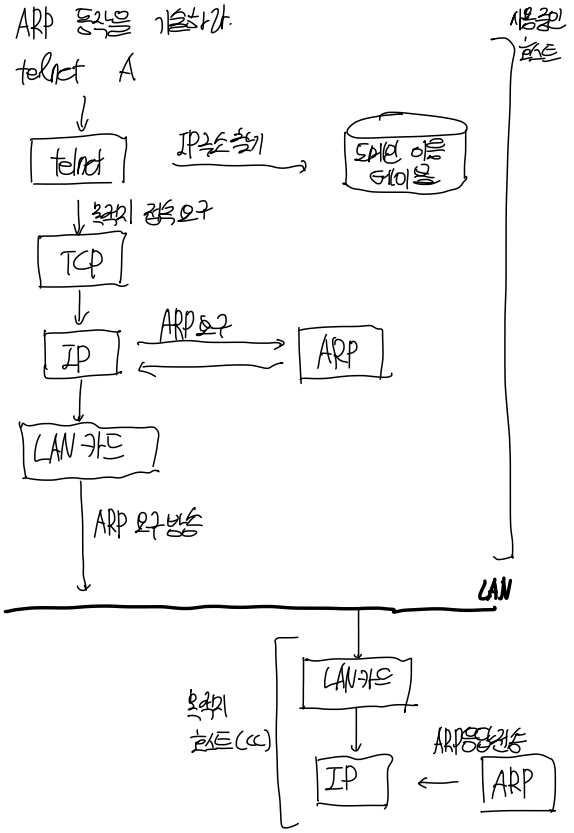


- Version : IP 버전 값을 표시하며 현재는 4의 값을 가짐.
- IHL (Internet Header Length) : 헤더 길이를 4바이트 단위로 표시, 최소값은 5 (헤더의 최소 크기는 20byte)
- Service Type : 서비스 클래스 지정, 보통 0으로 지정
- Total length : 헤더를 포함한 IP 데이터그램의 전체 크기를 바이트 단위로 나타냄. 16비트로 표현하므로 최대 값은 65535
- Identification : 상위 계층이 전송한 큰 메시지가 여러 데이터그램으로 단편화 되어 전송되었을 때 수신 측에서 원래 메시지를 재구성할 때 사용하는 번호. (한 메시지를 구성하는 데이터그램들은 모두 같은 Identification 값을 가진다.)
- Flag
  - 첫번째 비트 : 사용되지 않음, 항상 0
  - 두번째 비트 : 단편화 금지 플래그 (Don't Fragment) 비트
    DF=0 : 허용 DF=1 허용 X
  - 세번째 비트 : More 비트
    More=1 : 이 데이터그램과 다음에 전송되는 데이터그램이 단편화되어 전송됨을 나타냄.
    More=0 : 한 메시지의 마지막을 구성하는 데이터그램 또는 메시지가 한 데이터그램.
- Protocol : 데이터그램에 실려 있는 데이터를 처리할 상위 계층 프로토콜을 지정 (ex TCP: 6, UDP: 17, ICMP: 1)
- TTL (Time to Live)
  - 데이터그램들이 인터넷 내에서 계속 돌아다니는 것을 방지하기 위하여 사용
  - 헤더를 16비트 단위 크기로 나누고 각 16비트를 숫자로 취급하여 차례로 더한 결과에 1의 보수를 취해 HeaderChecksum에 넣어 송신
  - 수신 측 컴퓨터는 같은 방식으로 데이터그램 수신
- Source IP Address : 송신측 IP 주소
- Destination IP Address : 수신지 IP 주소
- IP Options : 옵션 선택, 보통 사용되지 않음.
- Padding : 32비트 단위로 헤더의 길이를 맞춤. 보통 사용되지 않음.

2. IP 프로토콜의 특징을 기술하라.

- OSI 7계층의 관점에서 보면 IP 계층은 주소와 라우팅을 담당하는 OSI 계층 3에 해당.
- IP계층은 서브네트워크를 이용해 IP 데이터그램을 임의의 호스트 사이에 전달하는 기능
- 임의의 호스트를 4바이트의 IP주소만으로 찾고 데이터그램을 전달.
- 비연결 방식으로 데이터그램을 전달.
- 데이터그램의 분실, 중복, 전달 순서 바뀜, 혹은 에러 등이 발생해도 IP 계층에서는 데이터그램 에러 회복이나 재전송 등을 하지 않는다.
- 데이터 그램을 안정적 목적지로 전달한다고 전송 결과 확인은 하지 않는다.

3. ARP 동작을 기술하라.



- 응용 프로그램 telnet은 먼저 A호스트가 어떤 IP 일을 찾음 (네임 서비스)
- telnet은 이 IP주소를 TCP에게 알려주고 이것으로 연결하도록 요청
- IP계층은 목적지 호스트의 net 번호와 자신의 net번호가 같은지 비교.
  - 같은 LAN에 있으면 LAN을 통해 바로 telnet 서버로 요청을 전송.
  - 만약 같은 LAN에 있지 않으면 telnet 서비스 요청을 default gateway를 통해 타국로 요청.
- 목적지 MAC 주소를 알아내기 위해 ARP를 구동
- ARP에서 목적 호스트의 MAC주소를 찾기 위한 ARP request 패킷을 LAN 내의 모든 장비에 방송.
- 목적지 호스트만 ARP request에 자신의 MAC 주소를 응답.

4. TCP의 특징을 기술하라.

- 연결형 서비스 제공
  - 종점호스트 사이에 1대 1연결을 제공
  - 연결설정, 데이터 송수신, 연결종료 과정이 필요.
  - 두 호스트 사이에 TCP 연결 설정이 이루어지면 두 개의 스트림이 생성되어 양방향 통신
  - 호스트 A가 호스트 B로 연결요청을 한 경우에도 호스트 B가 호스트 A로 데이터 전송이 가능.
- 신뢰성 있는 데이터 전달.
  - 신뢰성 있는 종점간 데이터 송수신을 보장.
  - 흐름제어(ACK), checksum, 재전송, 흐름제어 등을 사용.

↳ 흐름제어(ACK)

= 상대방 TCP에게 본 컴퓨터로부터를 확인하기 위해 사용.

= 데이터를 수신한 상대방은 데이터에 이상이 없고 순서에 올바게 왔을 시 송신 측에 Ack를 보냄.

= 에러 확인은 checksum, 순서확인은 순서번호를 사용함.

= Piggy-back 기능 : 데이터를 보낼때 마다 Ack를 전송하면 불필요한 트래픽이 발생

수신측은 즉시 Ack를 보내지 않고 약간의 시간지연을 둔다.

지연시간 동안 보낼 데이터가 생기면 데이터에 Ack를 포함시켜 전송.

↳ Checksum

= 데이터 전송 중에 발생한 에러를 검출.

= Checksum 에러 검출 시 데이터를 버리고 Ack를 보내지 않는다.

= 목적로 에러 상황을 송신 측에 알려주지는 않는다.

↳ 재전송

= 일정 시간 동안 Ack가 오지 않을 경우 전송했던 데이터를 재전송

= 재전송을 위해 송신측은 Data를 사용. (데이터 손실로 Ack가 오지 않을 경우 재전송)

= 전송 데이터 값은 RTT(Round Trip Time)에 비례하여 점점 증가하여 혼잡이도 제한. (일정 횟수 이상 Ack가 왔을 시 종결된 연결을 종료.)

↳ 흐름제어

= 수신측의 사정에 의한 흐름제어 : 상대방이 너무 많은 데이터를 보내지 못하게 하는 흐름제어.

수신 버퍼의 여유 크기를 고려하여 상대방이 전송할 수 있는 데이터의 양을 바이트 단위로 알려준다.

수신할 수 있는 데이터 양을 Window 필드(16bits)를 통해 송신 측에 알려준다.

= 송신측의 사정에 의한 흐름제어 : 송신 측에서는 송신버퍼가 일정크기면 write(), send() 같은 쓰기 함수가 블록된다.

↳ 스트림형 서비스 제공

=송신측에서 전송된 데이터는 바이트 단위로 처리되어 수신 측에 전달.

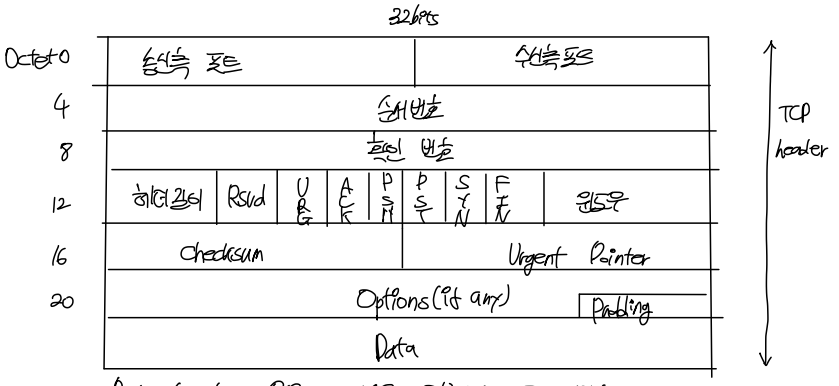
=스트림 서비스 : 송신된 데이터를 수신 측에서 차례대로 읽을 수 있게 제공되는 통신 채널 서비스

수신측에서는 송신된 데이터를 한 번에 읽거나 여러 번으로 나누어 읽을 수 있다.

송신측에서 write()들을 호출하여 전송한 데이터들의 경계를 수신 측에서는 알 수 없다.

송신측이 전송한 데이터 경계를 수신 측이 알도록 전송 데이터 크기를 따로 보낸다.

5. TCP헤더 구조를 기술하라.



- Source Port : 송신 측의 응용 프로세스를 구분하는 포트 번호
- Destination Port: 수신 측의 응용 프로세스를 구분하는 포트 번호
- Sequence Number 필드
  - 세그먼트에 실려 있는 데이터의 첫 번째 바이트의 순서번호 지정.
  - 수신 측에서 데이터가 정상적으로 도착하는지 확인하는데 사용
- Ack (Acknowledgement) Number : Ack 플래그가 1일 경우에만 의미가 있다.
- Header Length
  - 헤더의 크기를 4byte 단위로 나타낸다.
  - TCP에서 옵션을 사용하지 않는 경우 TCP 헤더의 크기가 20byte 이므로 5로 set.
- Rsvd : 현재 사용되지 않는 값으로 0으로 set.
- Code Bits
  - ↳ URG: Urgent Pointer가 유효한 값임을 나타낸다.
  - ↳ ACK: Ack Number에 set 되어 있는 값이 의미 있는 값임을 나타낸다.
  - ↳ PSH: 이 데이터를 가능한 신속히 응용에 전달하도록 한다.
  - ↳ RST: 연결을 reset할 때 사용.
  - ↳ SYN: TCP 연결을 시작할 때 사용.
  - ↳ FIN: TCP 연결을 종료할 때 사용.

- Window : 수신측이 현재 수신 가능한 데이터 버퍼 크기를 바이트단위로 나타낸다.
- checksum
  - pseudo 헤더 : TCP 세그먼트 전체 + IP 헤더의 후반부 12byte
  - pseudo 헤더에 대한 에러 검출 코드
- Urgent Pointer
  - URG 비트가 1일 때 의미 있다.
  - 현재 세그먼트에 포함된 긴급 데이터의 마지막 위치를 가리키는 offset을 저장.
  - 긴급데이터는 Sequence Number 위치부터 시작하여 Urgent Pointer 바이트 만큼 범위에 속한다.

6. TCP에서 TIME-WAIT 상태에 대해 기술하라.

- Active close를 한 호스트가 FIN을 받고 이 FIN에 대한 ACK을 보내기 까지 잠시 머문 상태
- Time-wait 상태에 있는 동안 지금까지 사용했던 포트번호를 재사용할 수 없도록 제한.
- Time-wait 상태를 두는 이유1.
  - 연결종료를 신청한 측에서 ACK(L+1)까지 안전하게 전송해야 양방향 채널 종료.
  - 만약 ACK(L+1) 메시지가 사라지면 상대방(Passive close)는 FIN(L)을 재전송.
  - 이때 Time-wait 상태에 머물지 않았다면 FIN(L)에 대한 ACK(L+1)을 보낼 수 없다.
  - Time-wait 시간은 종료를 모두 완료하기 위해서 기다리는 시간.
- Time-wait 상태를 두는 이유2
  - 연결 종료 직후에 같은 IP 주소와 포트 주소로 새로운 연결 요청을 허용하는 경우 : 과거의 연결에 사용되었던 세그먼트가 뒤늦게 도착하는 것을 잘못 수신할 수 있다.
  - 고의적인 해킹 공격 방지를 위해서 필요.
- 2MSL
  - Time-wait의 다른이름.
  - Time-wait 상태에서 기다리는 시간이 2*MSL(maximum segment lifetime) 시간
  - MSL은 네트워크 내에서 세그먼트가 남아있을 수 있는 최대 예상 시간
  - 대부분의 시스템은 30초에서 2분사이의 MSL 값을 가진다.
  - Time-wait 상태에 있는 동안 과거에 전송된 세그먼트가 도착 시 : 이 세그먼트는 버려지고 2MSL 타이머는 재시작.
  - Time-wait 시간 이후에 세그먼트 도착 시 : 세그먼트는 무시되고 송신 측으로 리셋(RST) 비트를 set시켜 전송 .