**물리 계층**

실제 장치들을 연결하기 위해 필요한 전기적, 물리적 세부 사항들을 정의

**데이터링크 계층**

Point to Point 간 신뢰성 있는 전송을 보장하기 위한 계층으로 CRC 기반의 오류 제어와 흐름 제어를 요구한다.

주소 값은 물리적으로 할당 받는데, 이는 네트워크 카드가 제작될 때부터 LAN카드 주소(MAC address)가 정해져 있으며, 해당 주소를 통해 데이터링크 계층에서 통신이 가능

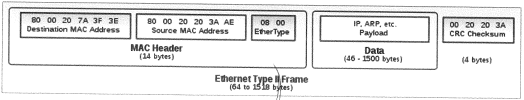
Mac 주소는 데이터링크 계층에서 사용되는 네트워크 카드의 48비트 하드웨어 주소

**이더넷의 구조**

다수의 PC를 연결한 환경인 LAN상에서 동작

1. 송신중인 다른 PC가 있는지 확인
2. MAC주소 B에게 전송
3. 자신의 MAC주소가 아닌 경우에는 폐기
4. 자신의 MAC주소인 경우만 수신
5. 다른 PC와 동시 전송하는 경우 충돌 발생 (충돌이 발생한 경우 일정시간 동안 기다린 후 다시 데이터 전송)

* 이와 같은 통신방식을 CSMA/CD라고 한다.



**ARP Protocol**

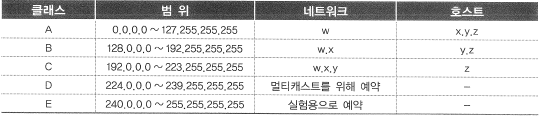
서버의 MAC 주소를 알지 못하는 경우 ARP 프로토콜을 구동하여 서버의 MAC주소를 얻는 과정을 먼저 수행하게 된다.

1. 모든 시스템에 ARP Request 프로토콜을 전달하기 위해 목적지 MAC주소를 ff:ff:ff:ff:ff:ff를 세팅하여 전송 (**브로드캐스팅 방식**<->유니캐스트), 목적지 IP주소는 있다.
2. 이러한 요청을 받은 수신자는 자신의 MAC주소를 채워 보냄 (ARP Reply)

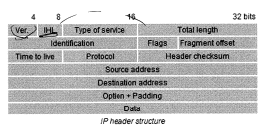
**네트워크 계층**

목적지 위치를 알려주는 고유한 32비트 (현재 4버전, 차세대는 6버전), 이를 IP주소라고 한다.

IP주소는 네트워크 부분과 호스트로 구분된다.



IP헤더는 이더넷 헤더(목적지 MAC주소 + 발신지 MAC주소 + 패킷유형) 다음에 따라붙는 형태



Ver : IP 버전 저장 / IHL : 헤더 길이정보 / Type of Service : 서비스 종류, 왼쪽 3비트는 우선권 필드, 다음 3비트는 플래그 비트 / Total Length : 헤더와 데이터의 길이를 합한 값

Identification : 전송할 최대 사이즈를 초과하여 분할된 경우, 분할되기 전, 어떤 패킷에 속한 것인지를 구분하기위한 고유번호 할당 / Flags : 분할된 추가 패킷이 있다는 것을 알려줌 (0비트 예약필드, 1비트 분할된 패킷이 없는 경우, 2비트 마지막 조각을 제외하고 해당 비트를 설정하여 분할된 패킷이 더 있음을 알림) / Flagment Offset : 수신지에서 재배열하는 과정에서 각 조각의 순서를 파악하는데 사용

Time to Live : 패킷수명을 제한하기 위해 데이터그램이 통과하는 최대흡수를 지정할 수 있으며, 패킷 전달 시 통과되는 흡수마다 TTL값은 감소한다. 만약 0인 경우에는 라우터에서 폐기하여 불필요한 패킷이 네트워크 상에 방치되는 것을 제한한다. / Protocol : IP헤더에 따라올 상위 프로토콜을 지정하는 것으로 TCP, UDP, ICMP 등을 확인할 수 있다. / Header Checksum : 헤더의 오류를 검증하기 위해 사용

Source Address : 송신자 IP주소 값

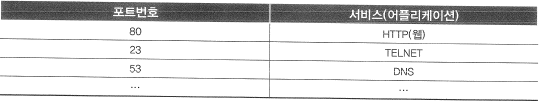
Destination Address : 수신자 IP주소 값

Options : 새로운 실험 혹은 헤더정보에 추가정보를 표시하기 위해 설계

**트랜스포트 계층**

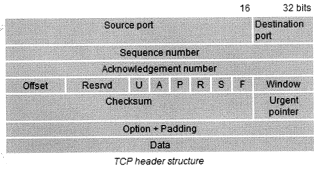
전송 계층은 양 끝단의 사용자들이 신뢰성 있는 데이터를 주고받을 수 있도록 해주어, 상위 계층들이 데이터 전달의 유효성이나 효율성을 보장해준다.

포트(Port)라고 지칭하는 서비스의 출입구가 존재하며, 고유번호를 할당하여 어떤 서비스를 사용할지를 결정하게 된다. 사용되는 포트범위는 1~65535번 까지이며, 이중에서 어플리케이션에 미리 할당된 포트번호를 Well Known 포트(1~1023)라고 정의하고 있다.



**TCP** – 상대방에게 정확하게 데이터 전달, 상대방이 수신이 준비된 상태에서 데이터를 전달하는 세션연결 과정을 거친다.

세션을 맺은 이후 데이터 전송, 데이터 전송 중 발생하는 데이터 훼손의 문제는 재전송을 통해 보완 (ex. 인터넷뱅킹, 메일 ,,,)



**3 Way HandShaking**

통신시작 (세션 연결) - TCP 통신에서 가장 핵심적인 부분

* SYN, SYN+ACK, ACK의 조합을 통해 상대방의 요청 확인 및 승낙

데이터 송수신 - 사용자는 홈 체이지 초기 파일을 요청하고, 웹서버는 요청한 초기파일을 만들어 다시 사용자에게 전달하는 구조

* PUSH를 통해 요청한 데이터에 대한 빠른 처리 요청

통신종료 - 통신을 시작한 쪽에서 먼저 통신해제 요청

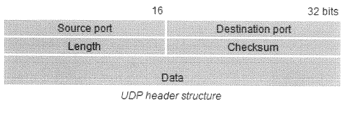
* 송수신자가 서로 FIN+ACK을 보내 상호간 합의하는 식으로 통신종료

**UDP** – 데이터 전달에만 목적이 있기 때문에 상대방의 준비상태와 정상수신 여부에 관계없이 전송한다. (ex. 인터넷 전화, 영상전송,,,)

헤더정보도 단순하고, 속도 또한 빠르다. 그러나 TCP 프로토콜이 가지고 있는 신뢰성은 보장되지 않는다.

데이터 전달 중 손실이 발생된 경우, 해당 데이터를 폐기할 뿐 다시 요청하지는 않는다.

대상 서버가 제공하지 않는 포트로 접속 시도 시 UDP 프로토콜에는 에러처리기능이 없기 때문에 ICMP 프로토콜이 대신 처리한다.



**어플리케이션 계층**

다른 계층의 서비스를 위한 다양한 어플리케이션은 제공하고, 어플리케이션 상호간데이터를 교환하기 위한 프로토콜을 정의한다.

* HTTP : WWW 서비스 제공
* FTP : 상호 파일 전송
* SMTP : 메일 메시지와 첨부파일 전송
* Telnet : 네트워크 호스트에 원격 접속하기 위해 사용
* DNS : 호스트 이름을 IP주소로 변환하기 위해 사용
* RIP : IP 네트워크상에서 라우팅 정보를 교환하기 위해 라우터가 사용
* SNMP : 네트워크 관리 콘솔과 네트워크 장비간에 네트워크 관리 정보를 수집 및 교환하기 위해 사용