[ TCP ]

1. TCP의 기능

- 신뢰성 있는 프로토콜을 지원하려면 IP 프로토콜과 TCP 프로토콜이 짝을 이뤄야지만,

인터넷에서 ‘신뢰성 있는 전송’이 가능하다.

- TCP는 프로세스간 통신(like UDP), 스트림 전달 서비스, 전이중 통신, 연결 지향 서비스, 신뢰성 있는 서비스를 제공한다.

> 연결 지향 서비스이므로 연결설정과 연결해제 과정이 존재한다.

\* 스트림 전달 서비스

: 송신측에서 데이터를 보내면 바이트들의 연속된 내용으로 보낸다.

(UDP는 유닛데이터를 전송하기 때문에, 데이터간의 관련성이 없다, But TCP는 있다)

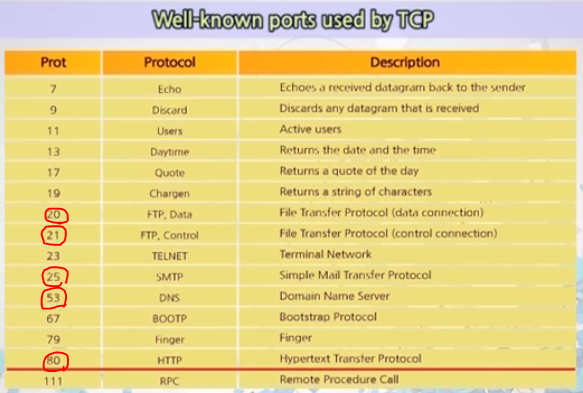
\* 전이중 통신

: 메시지 하나 보내고 끝이아니라, 보내고 받는 채널이 따로있어 보내며 받을 수 있다.

\* 신뢰성 있는 서비스

: 데이터의 안전한 도착을 확인하는 ACK를 사용하여 신뢰성 있는 전송을 제공한다.

- TCP는 프로세스간 통신을 위해 포트번호를 사용한다. (포트번호를 통해 프로세스 구분)



- 스트림 전달 서비스 : TCP는 데이터를 바이트의 나열로 전달한다.

> 효율성을 위해 여러 바이트를 블록으로 구성하여 세그먼트로 만들어 전송한다.

이 것을 TCP Segment라고 한다.

\* TCP의 특징

- TCP는 모든 바이트에 번호를 부여한다.

> 시작번호는 0 ~ (2^32) – 1 사이의 임의의 번호를 갖는다.

> 번호는 흐름제어와 에러제어에 사용한다.

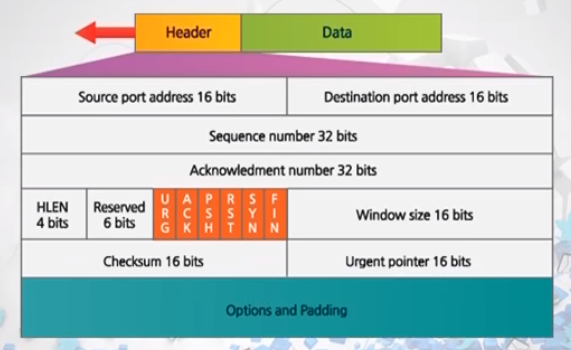
> 세그먼트의 순서번호는 해당 세그먼트가 나르는 데이터의 첫 번째 바이트 번호이다.

- ACK 번호는 수신해야 할 다음 바이트 번호이다.

> ACK 번호는 누적 값으로, 수신자는 해당 번호 이전의 모든 바이트를 안전하게 받았음을 의미

> 예를 들어 1234가 ACK번호라면, 1233바이트까지 모두 받았음을 의미한다.

\* TCP 헤더 : 헤더는 20바이트이며, 옵션이 있으면 60바이트까지 존재할 수 있다.



- Source port address : 송신자 포트번호

- Destination port address : 목적지 포트번호

- sequence number : 바이트 단위의 순서번호, 이 세그먼트가 실어나르는 첫번째 바이트의 번호

- Acknowledgement number : 다음 받을 바이트의 번호

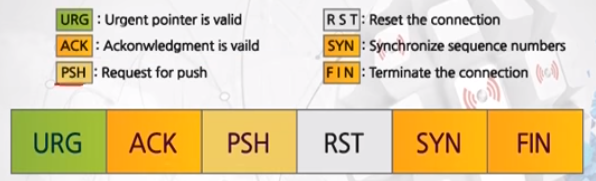
- HLEN : 헤더의 길이 (가변), 4바이트 단위로 표시. (ex. 5라면 20바이트)

- Reserved : 사용되지 않는 비트

- 6비트의 제어필드(URG, ACK, PSH, RST, SYN, FIN)

\* 제어 필드

: TCP 연결 설정과 연결 해제 메시지를 구분하는 것. 6개의 비트로 구성되어 있다.



- URG : 1이면 Urgent field가 유효하다.

\* 긴급 데이터

- 송신자는 수신자가 순서에 관계없이 우선적으로 데이터가 처리되길 원할 수 있다.

- 긴급데이터를 포함하면 제어필드의 URG와 긴급 포인터(Urgent pointer)를 사용

- ACK : 1이면 ACK가 유효하다.

- PSH : Push를 한다.

\* 송신 TCP는 수신 TCP가 버퍼링을 하지 말고 즉시 데이터를 응용에게 전달할 것을

요구할 수 있다. (여러 바이트를 모아서 블록을 만들지 않고 즉시 바로바로 전송)

- 제어 필드의 PSH 비트를 설정한다.

- 지연된 전송은 대화식 응용(Interactive applications)에서는 바람직하지 않다.

- RST : 1이면, 연결 설정을 새로 한다.

> 비정상적인 상황이나 오랫동안 통신이 없으면 현재의 연결을 끊는데, 이 때 사용!

\* 체크섬

- 데이터에 대한 오류를 검사하여 재전송에 의한 복구를 수행할 수 있다.

- TCP에서 체크섬은 **강제 사항**이다.

- UDP에서와 같은 **가상헤더**를 포함하여 계산한다.

- SYN : 연결 설정!!

- FIN : 연결 해제!!

[주의사항] 제어 필드는 부정적 ACK(NACK)가 없다.

- Window Size : 흐름제어에 사용

- Checksum : 에러유무 판단

- Urgent pointer : 긴급 데이터 전송과 관련

- TCP에서 ACK번호의 예

> TCP는 송신자 포트 번호, 수신자 포트 번호를 사용해서 TCP 위에 올라가는 어떤 응용

프로세서가 통신 하는건지 구분하는 식별자로 사용!!

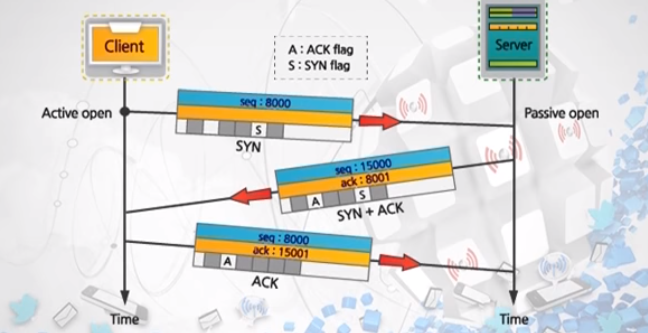
3. 연결 설정과 해제 -> **Three-way handshake**

- TCP는 3단계 메시지 교환(Three-way handshake)을 통해서 연결 설정을 한다.

> 클라이언트는 연결을 요청하는 SYN 세그먼트를 전송한다.

> 서버는 SYN과 ACK를 포함하는 세그먼트로 응답한다.

> 클라이언트는 ACK를 보낸다.



- TCP는 연결해제를 위해서 3단계(Three-way handshaking)와 4단계(four-way handshaking) 메시지 교환을 제공한다.

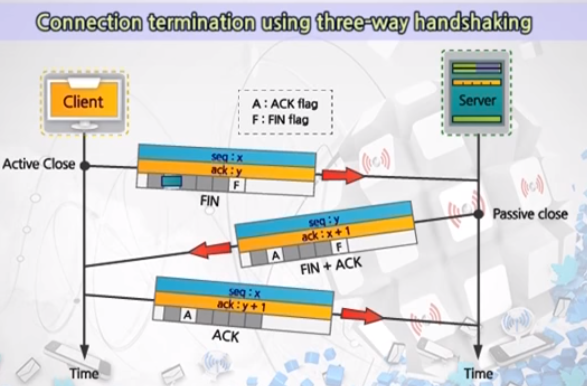
>> Three-way handshaking을 사용하여 연결을 바로 끊을 수 있고

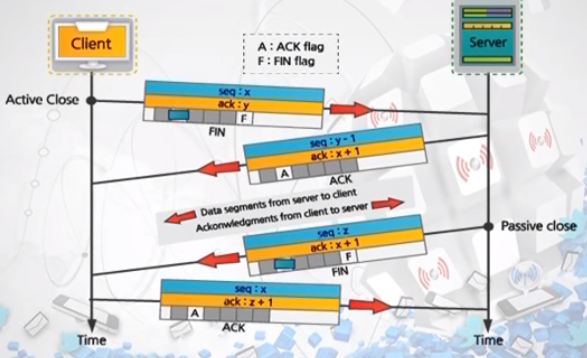
four-way handshaking을 사용해서 지연시간을 두고 연결을 끊을 수 있다.

- 제어필드의 FIN을 사용하여 연결 해제를 한다.

- 4단계 메시지 교환을 통한 연결해제는 half-close 상태를 만들어 수신은 가능하게 할수있다.

- 3단계 메시지 교환을 통한 연결해제는 즉시 연결을 종료시킬 수 있다.





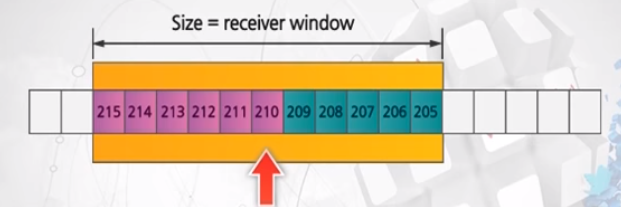
4. 흐름 제어

- 송신 TCP가 목적지로부터 ACK를 수신하기 전에 보낼 수 있는 데이터의 양을 정한다.

> TCP는 슬라이딩 윈도우 프로토콜(sliding window protocol)을 이용한다.

> TCP는 바이트 단위로 윈도우 크기(수신 가능한 데이터 양)을 명시한다.

> 윈도우 크기는 시간에 따라서 변할 수 있다.



> 수신측에서 받을수 있는 데이터 크기를 알려 주는 것을 윈도우 프로콘트롤(window procontrol),

이 크기를 윈도우 사이즈(window size)라고 한다.