

<인터넷 프로토콜>

작성자_2018320161_송대선

작성일_05_16

<TCP>

가장 많이 사용되는 프로토콜이다.

95% 이상의 transport 계층이 TCP를 쓴다.

-Service model of TCP-

Connection-Oriented: TCP를 쓰려면 “연결”이 필요하다.

Reliable: 반드시 배달해 준다. 믿고 쓸 수 있다.

->TCP는 반드시 배달한다.

Inorder delivery: 배달의 순서를 지킨다. 먼저 온 데이터를 먼저 받아야 한다.

TCP는 순서가 뒤바뀌면 응용에게 데이터를 주지 않음

Full-duplex: 동시에 양방향으로 흐를 수도 있다.

TCP는 채널을 2개 열어준다.

-MSS-

TCP는 Bytes들을 잘라서 Segment로 만든다. -> 여기서 Path MTU가 필요하다.

IP의 DF=1이면 크기 제한을 넘어가는 패킷을 죽이고, 그에 대한 정보를 host쪽으로 넘김

->TCP가 이를 확인하고, MTU를 재조정해서 다시 보냄

Maximum Segment Size(MSS) = (문제가 된 MTU) - 40B

MSS는 TCP segment에서 TCP Header를 제외한 부분을 의미한다.

MTU = IP Header + TCP Header + MSS

MSS가 최대가 되려면 IP Header, TCP Header가 최소가 되어야 한다.

-> IP Header, TCP Header에 Option이 없어야 한다.

-> Option이 없는 IP Header, TCP Header의 길이가 20B이다.

그래서 MSS = MTU - 40B이다.

Ethernet의 MTU가 1500B이면, MSS는 1460B이다.

-TCP Header format-

Source Port, Destination Port: Multiplex, Demultiplex

Sequence number: TCP의 모든 Byte들 각각에는 Sequence Number가 부여된다.

-> 0, 1같은 문자들이 아니라 이상한 random해 보이는 숫자들을 붙인다

Acknowledge number: 수신 측에서 마지막으로 받은 Segment의 sequence number

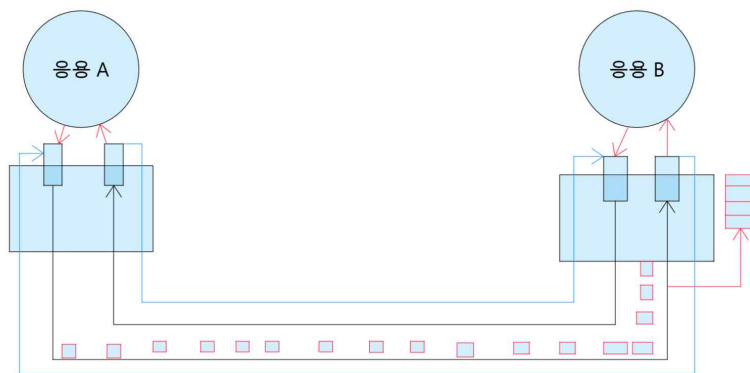
Header length: 4B씩 센다.

15->60byte, 60-20=40, 40Byte가 Option

Flag:

Window Size:

Checksum: 16bit씩 packet을 잘라 더하고 보수로 바꾸면 checksum이다.



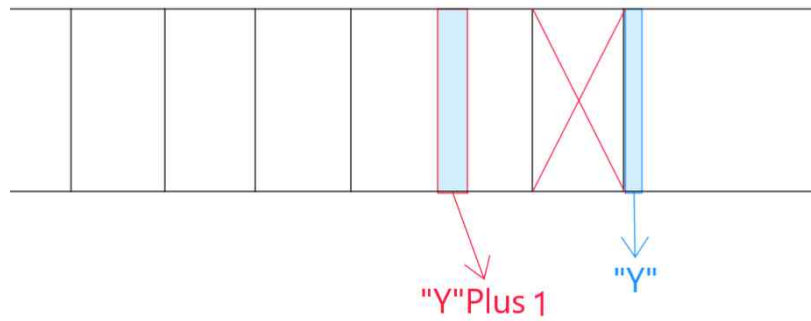
A->B인 채널을 설명해 보겠다.

1. 송신자인 응용 A는 socket buffer에 data를 쏟아붓는다.
(socket buffer는 circular queue이다.)
2. A는 자신의 socket buffer에 있는 data의 copy를 B를 향해 보낸다.
3. 수신자가 data를 받아서 순서에 맞으면 socket에 넘기고 ACK를 한다.
(맨 마지막 data의 sequence number가 "x"이면 "x+1"을 달라고 ACK한다.)
(Socket에 들어가면 데이터 손실의 가능성이 거의 없다.)
4. ACK를 받은 TCP A는 자신의 socket buffer에서 "x"에 대한 copy를 지운다.

reordering buffer: 마지막으로 받았던 data의 sequence 번호가 y인데 y+1이 아닌 다른 번호가 오면 reordering buffer에 그 data를 저장한다.

다른 것들이 reordering buffer에 들어갈 때마다 "y+1"로 ACK한다.

Header에 실리는 Sequence Number는 맨 앞의 Sequence Number이다.



-Synchronize (SYN)-

: TCP 연결에 필요한 3가지 숫자를 알려줌

A는 A->B에 대한 2개, B->A에 대한 1개

B는 B->A에 대한 2개, A->B에 대한 1개

-> total 6개의 숫자에 대한 양측의 이해가 동일해야 한다.

숫자 3개는

1. Maximum Segment Size (MSS)

2. Initial Sequence Number (ISN)

3. Window Size

이 3가지이다.

Flag에서 SYN=1이면 SYN하는 것이다.

SYN=1이면 data는 없고, header만 있다.

처음에는 initial sequence number, 최대 window size가 담겨있다.

최대 window size는 처음에는 socket buffer가 비어있으므로 처음의 window size가 된다.

Window size를 알아야 flow control이 가능하다.

-> 이걸 알아야 내가 얼마를 보냈으니, 얼마를 더 보내도 되겠다 하는 계산이 가능하다.

socket buffer는 왜 필요한가?

-> window process가 항상 가동되는 것이 아니기에 임시 저장공간이 필요해서

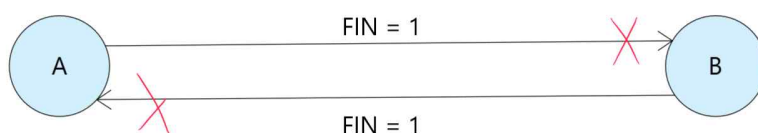
MSS는 왜 필요한가?

그 상대방도 같은 링크를 사용할 가능성이 매우 높기 때문.

A, B 서로 자신의 MSS를 알려주고 그 중 최솟값을 선택한다.

-Finish (FIN)- : 연결을 끝자 (graceful manner-> 순조롭게 끝자)

Flag의 FIN=1이면 끝는 것이다.



-Reset(RST)-

Flag의 RST=1이면 Reset이다. -> kill them all이라는 뜻이다.

이유:

1. 게으른 서버: 다 보낸 것을 서버가 알면, 그냥 reset해버림
2. port에 아무도 안 살면 그냥 reset해 버림

-ACK-

ACK=0은 처음에 SYN Segment뿐이다.

-Push(PSH)-

Flag의 PSH=1이면 push이다.

원래는 “빨리 보내주세요”라는 의미였는데, 요즘은 기능적인 역할이 없다.

(원래 빨리 보낸다.)

PSH=1이면 “떠날 때 보니 뒤에 아무 data가 없더라”라는 의미이다.