

네트워크 개론 Part 19

: TCP Transmission Control Protocol의 이해 7

조인준

KBS 미디어기술연구소

차장

TCP에 관한 설명을 이어오면서 안정적 전송을 보장하는 TCP의 특징으로 흐름제어(Flow Control), 오류제어(Error Control), 혼잡제어(Congestion Control)가 있음을 알려드렸고, 이 중 흐름제어와 오류제어는 Go-Back-N ARQ와 Selective Repeat ARQ 관련 내용을 통해 설명해 드렸습니다. 이번 편에서는 위 세 가지 특징 중 마지막인 혼잡제어에 대해 다뤄보겠습니다.

혼잡제어에 관한 설명을 위해서는 우선 네트워크의 '혼잡'이란 무엇인가에 관해 간략히 정리할 필요가 있을 것 같습니다. 네트워크의 혼잡이란 '네트워크를 통해 전송되는 패킷의 양이 과중하게 되어 패킷 전송의 실패로 이어질 수 있는 네트워크의 속도 저하 상태' 정도로 이해하면 매우 전문적 경우가 아닌 실용적 측면에서는 적당할 것 같습니다. 네트워크를 통해 전송되는 과도한 패킷들로 인한 네트워크 혼잡은 피할 수 없는 문제이고, 이는 제어를 통해 해결해야 합니다. 이 때문에 TCP에 안정적 전송을 위해 혼잡제어 기술이 적용된 것입니다.

TCP의 혼잡제어는 네트워크의 혼잡을 예방하거나 발생 후 해결하기 위한 일련의 기술적 조치이며 상황에 따라 데이터를 송신하는 디바이스인 Sender의 윈도우 크기(window size)를 조절하는 방식으로 동작합니다. 송신 디바이스의 송신 윈도우 사이즈(Sender window size)는 수신 디바이스의 수신 버퍼 크기에 따른 수신 윈도우 사이즈(Receiver window size)와 Sender의 판단에 따라 설정된 혼잡 윈도우 사이즈(Congestion window size)를 비교하여 둘 중 작은 값을 취하게 됩니다.

Case : 수신 윈도우 사이즈 $\leq$ 혼잡 윈도우 사이즈
송신 윈도우 사이즈 = 수신 윈도우 사이즈

Case : 혼잡 윈도우 사이즈 < 수신 윈도우 사이즈
송신 윈도우 사이즈 = 혼잡 윈도우 사이즈

$\therefore$ 송신 윈도우 사이즈 = MINIMUM(수신 윈도우 사이즈, 혼잡 윈도우 사이즈)
 MINIMUM(x, y) = x, y 중에 작은 값

표 1. 혼잡제어에서의 송신 윈도우 사이즈

앞선 편들에서 말씀드렸듯이, 수신 윈도우 사이즈는 ACK 메시지 없이 송신 디바이스가 수신 디바이스에 보낼 수 있는 데이터의 바이트 수를 의미합니다. 수신 디바이스가 데이터 수신 후 일시적으로 저장할 수 있는 데이터의 최대량이기 때문에 송신 디바이스는 이보다 더 큰 데이터를 보내면 안 됩니다. 그리고 이번 편에서 소개되는 혼잡 윈도우 사이즈는 송신 디바이스가 수신

측의 ACK 메시지 없이 보내도 네트워크에 혼잡을 일으키지 않을 것으로 예상되는 데이터의 최대량 정도로 이해하시면 될 것 같습니다. 그래서 [표 1]과 같이 네트워크 혼잡을 일으키지 않는 범위에서 수신 디바이스에서 문제없이 수신 가능한 양의 데이터를 보내야 하기 때문에 수신 윈도우 사이즈와 혼잡 윈도우 사이즈 둘을 비교하여 둘 중에 작은 윈도우 사이즈를 송신 윈도우 사이즈로 설정하는 것입니다. 그러나 혼잡 윈도우 사이즈는 수신 윈도우 사이즈처럼 누군가 알려주는 것이 아니고 블랙박스 처럼 알 수 없는 복잡한 네트워크 상황을 추정하며 혼잡을 예방하거나 해결해야 하므로 IP 네트워크 기술의 발전이나 상황에 맞게 혼잡 윈도우 사이즈를 조절하는 기법들이 개발되었습니다. 또한, 혼잡 윈도우 사이즈는 이를 추정한 송신 디바이스만 알고 있는 수치이며 수신 측에 전달되거나 하지 않습니다.

말씀드렸듯이 혼잡제어에는 몇몇 버전들이 존재하는데 자료 등을 통해 일반적으로 소개되는 혼잡제어 기술은 다음과 같은 세 단계로 나누어 볼 수 있습니다.

- ① Slow Start ② Congestion Avoidance (혼잡 회피) ③ Congestion Detection (혼잡 검출)

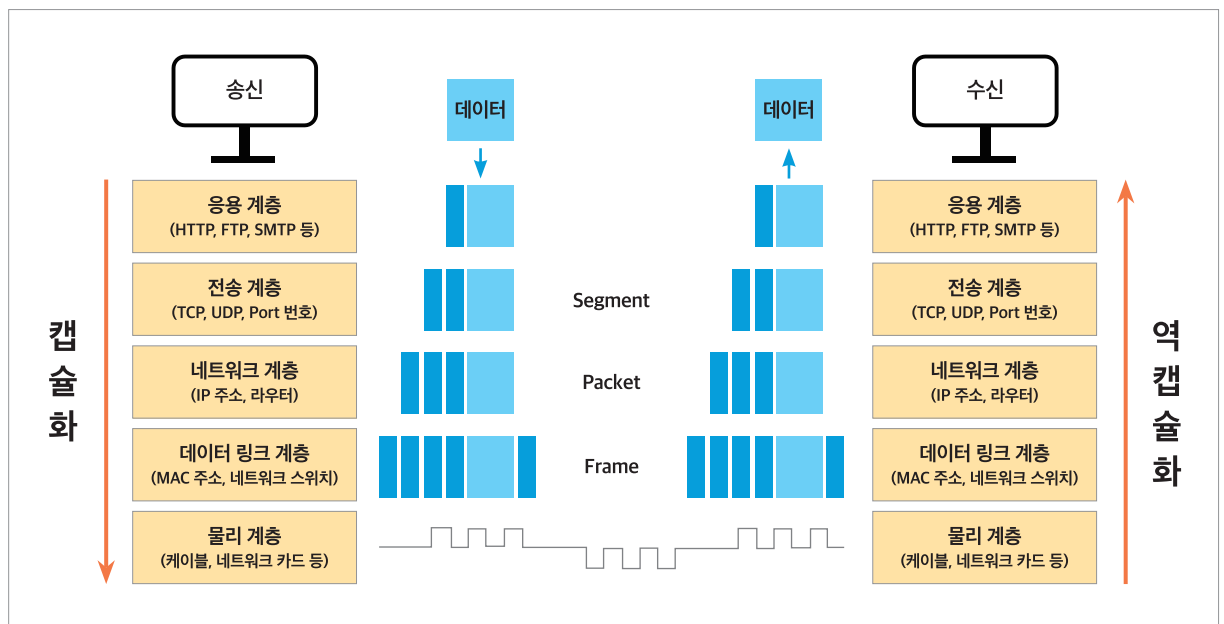


그림 1. 전송계층과 세그먼트(Segment)

혼잡제어에서는 MSS(Maximum Segment Size)를 이용하여 송신 윈도우 사이즈를 정합니다. MSS는 MTU(Maximum Transmission Unit)에서 IP 헤더와 TCP 헤더를 제외한 TCP 프로토콜에 실리는 데이터의 양입니다. 우선 세그먼트(Segment)라는 용어는 이전 편에서 잠시 설명 드린 적이 있는데 [그림 1]과 같이 캡슐화 과정에서 응용계층의 헤더+데이터에 전송계층 헤더를 붙인 것을 일컫는 용어입니다.

또한 MTU는 한 번에 보낼 수 있는 데이터의 최대량을 가리키는 용어이며, 윈도우즈의 경우 명령 프롬프트를 실행한 후 'netsh interface ipv4 show subinterfaces' 명령어를 입력하면 [그림 2]와 같이 Loopback 인터페이스가 아닌 실제 통신을 하는 인터페이스의 경우 1500바이트로 기본설정이 되어 있음을 독자 여러분께서도 확인할 수 있습니다. 참고로 MTU는 사용자가 변경 가능합니다. 이렇게 시스템의 MTU를 알면 [표 2]와 같은 계산 방법으로 MSS를 구할 수 있습니다.

```
C:\W>netsh interface ipv4 show subinterfaces
```

MTU	미디어	감지	상태	수신	바이트	수	전송	바이트	수	인터페이스
4294967295			1	0	736486	Loopback	Pseudo-Interface	1		
1500		5		0	0	로컬	영역	연결		
1500		1		141595769640	3524513131	로컬	영역	연결	2	

그림 2. 윈도우즈 시스템의 MTU 확인

MSS = MTU - IP 헤더 바이트 수 - TCP 헤더 바이트 수

표 2. MTU로 부터 MSS 계산

Slow Start의 경우 처음 시작은 네트워크의 혼잡 상태를 알 수 없으므로 [그림 3]과 같이 우선 혼잡 윈도우 사이즈를 1 MSS로 하여 데이터를 보내고 수신 측의 ACK 메시지를 기다립니다. 만약 ACK 메시지가 정상적으로 수신되면 다음 단계에서는 혼잡 윈도우 사이즈를 2배 증가시킨 2 MSS로 데이터를 보낸 후 ACK 메시지를 기다립니다. 2 MSS에 대한 ACK 메시지가 정상적으로 수신되면 또다시 혼잡 윈도우 사이즈를 2배 증가시켜 4 MSS로 데이터를 보낸 후 ACK 메시지를 기다리는 패턴을 반복합니다. 이렇게 ACK 메시지가 정상적으로 수신될 때마다 혼잡 윈도우 사이즈를 이전 사이즈의 2배씩 지수적으로 증가시켜 송신 속도를 빨리 높여가는 것이 Slow Start입니다.

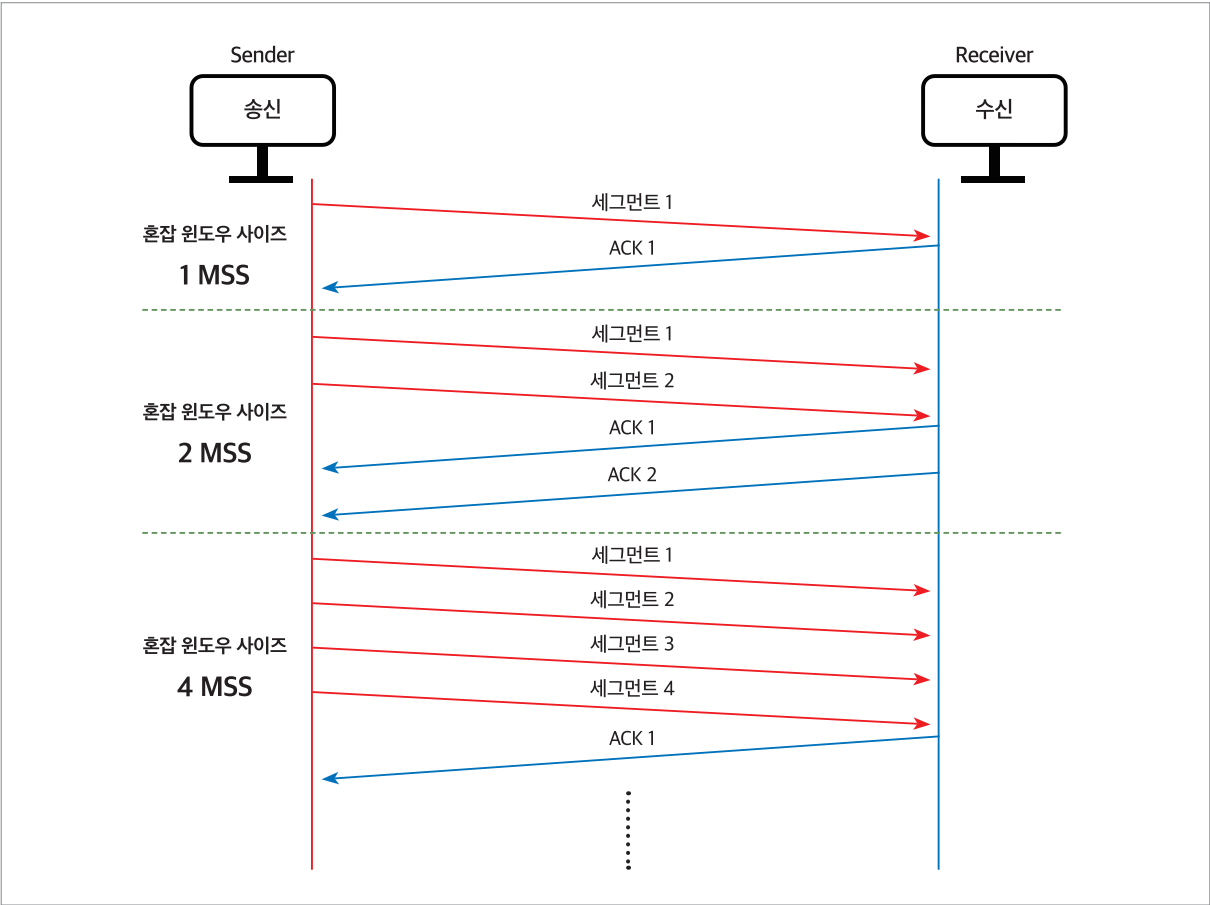


그림 3. Slow start에서의 혼잡 윈도우 사이즈 증가

하지만 무한정 혼잡 윈도우 사이즈를 2배씩 증가시키지는 않습니다. 일정한 임계값(Threshold)까지만 혼잡 윈도우 사이즈를 2배씩 증가시키며 이 임계값을 기점으로 Slow start와 Congestion avoidance 단계가 구분됩니다. 임계값 계산 방법은 [표 3]과 같습니다.

$$\begin{aligned}\text{임계값(Threshold)} &= \text{수신 윈도우가 수용할 수 있는 TCP 세그먼트 수} / 2 \\ &= \text{수신 윈도우 사이즈(바이트)} / \text{MSS(바이트)} / 2\end{aligned}$$

표 3. Slow start / Congestion avoidance 단계 임계값 계산 방법

Slow start를 통해 혼잡 윈도우 사이즈가 임계값에 도달하면 [그림 4]와 같이 ACK 메시지 정상 수신 시 지수적 증가를 하던 혼잡 윈도우 사이즈가 [표 4]와 같이 1 MSS씩만 증가하는 선형적 증가로 바뀌며 이 단계가 Congestion avoidance에 해당합니다.

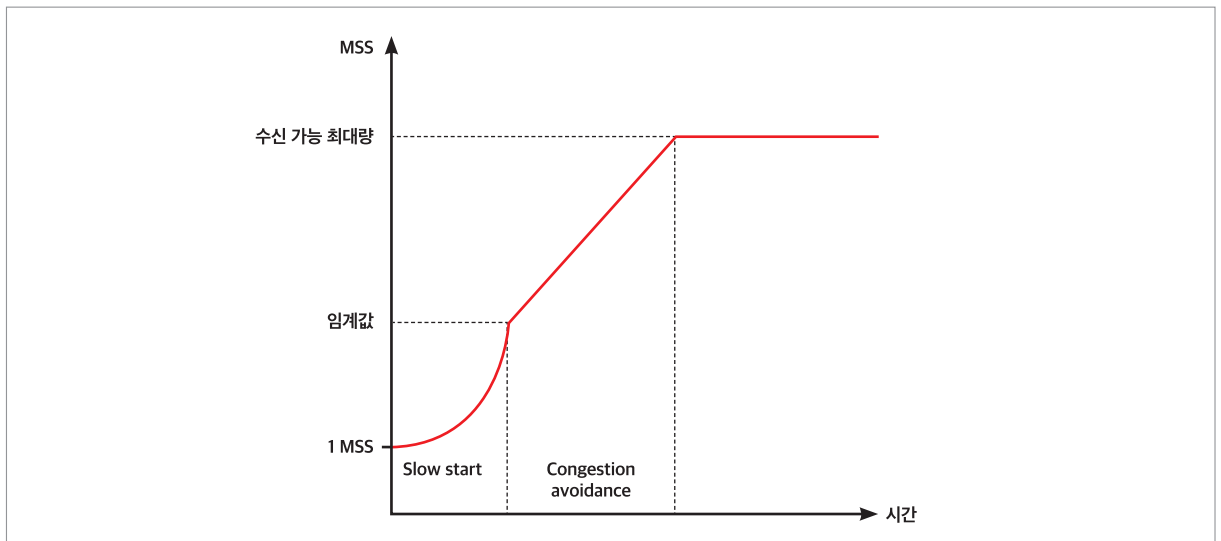


그림 4. 혼잡 윈도우 사이즈 증가 방식

$$\text{새 혼잡 윈도우 사이즈} = \text{기존 혼잡 윈도우 사이즈} + 1 \text{ MSS}$$

표 4. Congestion avoidance 단계의 혼잡 윈도우 사이즈

정리하면 수신 가능한 최대 데이터양의 절반을 임계값으로 잡아 Slow start를 통해 임계값에 빠르게 접근한 후 임계값 이후에는 1 MSS씩 증가시켜 혼잡에 조심스럽게 대응하는 방식으로 이해하시면 될 것 같습니다. 지금까지 Slow Start, Congestion Avoidance까지 설명 드렸습니다. 남은 Congestion Detection 관련 내용은 다음 편에서 다루도록 하겠습니다.

P.S.

C군이 여러분께 전하는 내용 중 전문적 성격이 짙은 것은 엄밀한 언어를 사용하여 설명하기에는 한계가 있습니다. 본 내용은 설명하는 대상에 대한 전체적 맥락의 이해에만 이용하시고, 그 이상은 권위 있는 전문자료를 참고하시기 바랍니다. 🐼