

What is TCP ?

Network

CONTENTS

01 TCP

02 흐름제어

03 혼잡제어

04 오류제어

01

TCP

01. CONTENTS

TCP

패킷을 추적 관리하는 역할

신뢰성 있는 데이터 전송을 지원 **"연결 지향형 프로토콜"**

가상회선 패킷 교환 방식을 사용하는 **연결형 서비스**

흐름 제어, 혼잡 제어, 오류 제어를 통해 신뢰성을 보장

데이터의 **전송 순서를 보장**하며 수신 여부를 확인할 수 있음

HTTP 통신 , 이메일 , 파일 전송(FTP)

01. CONTENTS

TCP

Q. 패킷 (Packet) 이란 ?

A. 라우팅을 효율적으로 하기 위해 조각 낸 데이터의 단위

Q. TCP는 패킷을 어떻게 추적 관리하는가 ?

A. 패킷 번호를 붙여 같은 목적지(IP계층) 보내며,
또한 도착지에서 패킷을 재조립하기 때문

01. CONTENTS

TCP

TCP Header

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset	Reserved 0 0 0			N S	C W R	E C E	U R G	A C K	P S H	R S T	S S T	F Y N	Window Size																		
16	128	Checksum																Urgent pointer (if URG set)															
20	160	Options (if <i>data offset</i> > 5. Padded at the end with "0" bytes if necessary.)																															
...																															

* IP 주소의 경우 IP 패킷에 정의되어 있음

01. CONTENTS

TCP

TCP Header

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset	Reserved 0 0 0			N S	C W R	E C R	U R G	A C K	P S H	R S T	S S Y N	F I N	Window Size																		
16	128	Checksum																Urgent pointer (if URG set)															
20	160	Options (if <i>data offset</i> > 5. Padded at the end with "0" bytes if necessary.)																															
...																															

Sequence Number : 세그먼트의 순서

01. CONTENTS

TCP

TCP Header

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64																	Acknowledgment number (if ACK set)															
12	96	Data offset				Reserved 0 0 0			N S	C W R	E C R	U R G	A C K	P S H	R S T	S S T	F I N	Window Size															
16	128	Checksum																Urgent pointer (if URG set)															
20	160	Options (if <i>data offset</i> > 5. Padded at the end with "0" bytes if necessary.)																															
...																															

Acknowledgment Number (승인번호) : 데이터를 받은 수신자가 예상하고 있는 다음 시퀀스 번호

01. CONTENTS

TCP

TCP Header

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 0 0 0			N S	C W R	E C E	U R G	A C K	P S H	R S T	S S T	F I N	Window Size															
16	128	Checksum																Urgent pointer (if URG set)															
20	160	Options (if <i>data offset</i> > 5. Padded at the end with "0" bytes if necessary.)																															
...																															

Data Offset : 헤더가 아닌 데이터가 시작되는 위치

01. CONTENTS

TCP

TCP Header

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset	Reserved 0 0 0		N C E U A P R S F S W C R C S S Y I R E G K H T N N								Window Size																				
16	128	Checksum																Urgent pointer (if URG set)															
20	160	Options (if <i>data offset</i> > 5. Padded at the end with "0" bytes if necessary.)																															
...																															

세그먼트의 속성을 표현하기 위한 플래그, 수신자 측에 전달하는 과정 중에서 발생할 수 있는 상황들에 대한 정의를 내려 어떤 행동을 해야하는지 알려주는 역할을 함

01. CONTENTS

TCP

TCP Header

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset	Reserved 0 0 0			N S	C W R	E C E	U R G	A K	P H	R T	S S	F Y N	Window Size																		
16	128	Checksum																Urgent pointer (if URG set)															
20	160	Options (if <i>data offset</i> > 5. Padded at the end with "0" bytes if necessary.)																															
...																															

Window Size : 한번에 전송할 수 있는 데이터의 양

01. CONTENTS

TCP

TCP Header

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset	Reserved 0 0 0			N S	C W R	E C R	U R G	A C K	P S H	R S T	S S Y	F I N	Window Size																		
16	128	Checksum																Urgent pointer (if URG set)															
20	160	Options (if <i>data offset</i> > 5. Padded at the end with "0" bytes if necessary.)																															
...																															

CheckSum : 데이터를 송신하는 중 발생할 수 있는 오류검출을 위한 값

02

흐름제어

02. CONTENTS

흐름제어

수신측이 송신측 보다 데이터처리 속도가 느릴 경우 문제가 발생

송신측의 전송량 > 수신측의 처리량일 경우 전송된 패킷은 수신측의 큐를 넘어서 손실될 수 있기 때문에 송신측의 패킷 전송량을 제어해야 함

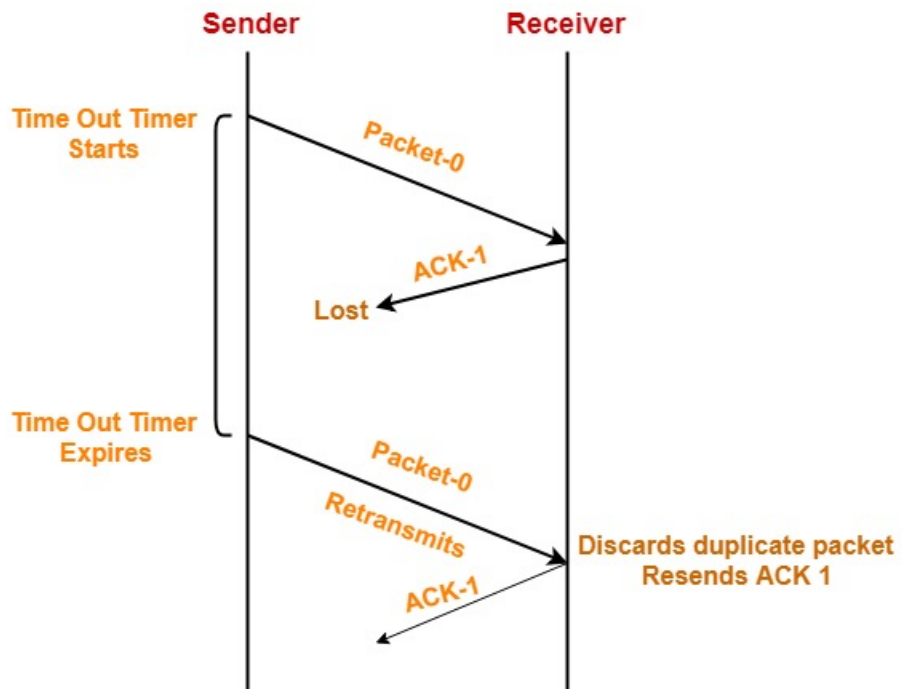
송신측과 수신측 사이의 데이터 처리 속도 차이를 해결하기 위한 기법 "흐름제어"

1. Stop and Wait (정지 - 대기)
2. Sliding Window (슬라이딩 윈도우)

02. CONTENTS

Stop and Wait

데이터를 1개 보내고 ACK가 오면 그 다음 데이터를 보내는 방식



속도가 느리고 비효율 적

02. CONTENTS

Sliding Window

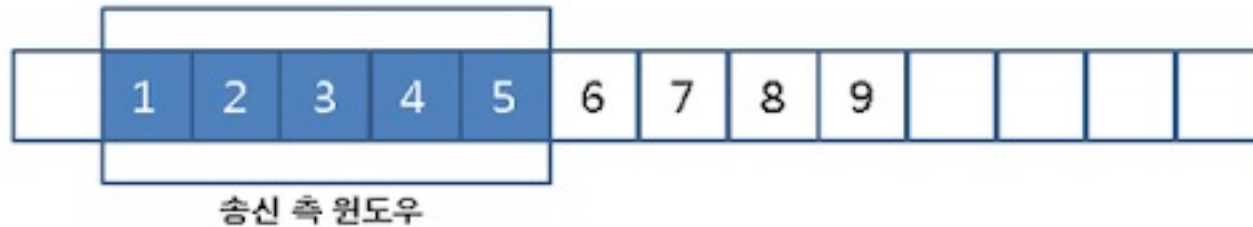
수신측에서 설정한 윈도우 크기만큼 송신측에서 확인응답 없이 세그먼트를 전송할 수 있게 하여 데이터 흐름을 동적으로 조절

수신측이 한번에 처리할 수 없는 양의 데이터를 송신측이 전송하게 되면 Drop 패킷이 발생하기 때문에 전송하는 패킷의 양을 슬라이딩 윈도우를 통해 조절

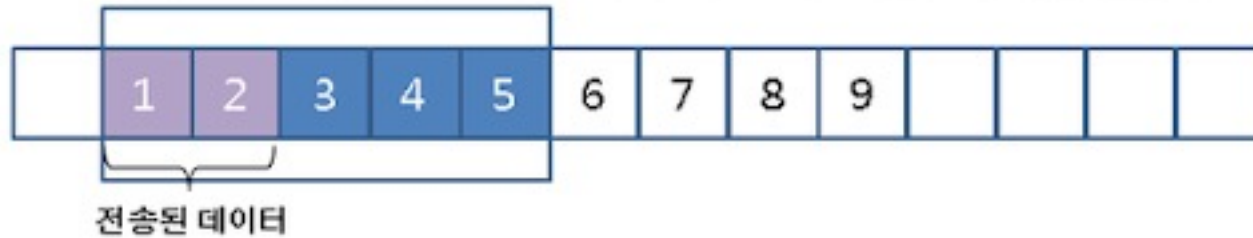
02. CONTENTS

Sliding Window

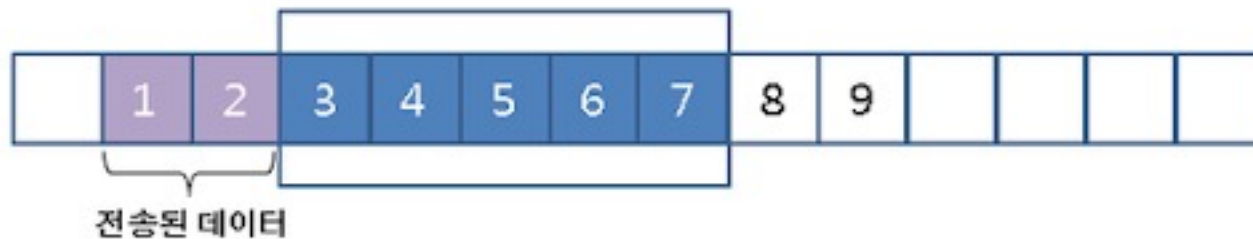
1. 송신 측에서는 1~5까지의 프레임을 전송 가능



2. 데이터 1,2를 전송 하고 아직 데이터 3,4,5는 전송하지 않은 상태



3. 데이터 1,2에 대한 ACK 프레임 수신 후, ACK된 프레임 만큼 윈도우 이동



03

흔잡제어

03. CONTENTS

혼잡제어

데이터가 몰려 혼잡할 경우 처리하지 못한 데이터가 발생할 수 있음

그럴 경우 다시 데이터를 처리해달라고 요청이 와 오버 플로우 혹은 데이터 손실이 발생하게 됨

이러한 경우 송신측에서 보내는 데이터의 전송속도를 강제로 줄이는 것을 "혼잡 제어"라고 함

1. 합 증가 / 곱 감소
2. Slow start
3. 혼잡 회피, 빠른 재전송, 빠른 회복

03. CONTENTS

합증가/곱감소

처음 패킷을 하나씩 보내고 이것이 문제 없이 도착하면 Window를 1씩 증가시켜 전송함

단, 일정시간을 넘기거나 패킷전송을 실패하면 Window Size를 절반으로 감소시킴

초기 네트워크의 높은 대역폭을 사용하지 못하고,
네트워크가 혼잡해지는 상황을 미리 감지하지 못하여 혼잡해지고 나서야 대역폭을 줄이는 방식이라는 문제가 있음

03. CONTENTS

Slow start

합 증가 / 곱 감소 방식과 비슷하게 패킷을 하나씩 보내는 방식
패킷이 문제 없이 도착하면 ACK 패킷마다 window 크기를 1씩 늘림

하나의 주기가 끝나면 Window의 크기가 2배가 된다!

단, 혼잡현상이 발생하면 window 크기를 1로 줄여버림

초기 네트워크 수용량을 예상하긴 어렵지만 혼잡현상을 경험하고 나
선 네트워크 수용량을 네트워크 수용량을 예상할 수 있음

03. CONTENTS

혼잡제어

혼잡 회피 : 혼잡이 감지되면 가산 적인 증가 방식을 채택한 방식

빠른 재전송 : 정상적인 재전송 큐 과정을 따르지 않고, 중간 누락된 세그먼트를 빠르게 재전송하는 방식

빠른 회복 : 혼잡 상태가 되면 Window 크기를 반으로 줄이고 선형으로 증가시키는 방법.

해당 기법을 적용하면 혼잡 상황을 한번 겪고 나서부터는 순수한 합 증가/곱 감소 방식으로 동작하게 됨

04

오류제어

04. CONTENTS

오류제어

오류검출과 재전송을 포함

Automatic Repeat Request (ARQ) 기법을 사용해 프레임이 손상되거나 손실되었을 경우, 재전송을 통해 오류를 복구

ARQ 기법은 흐름 제어 기법과 관련되어 있음

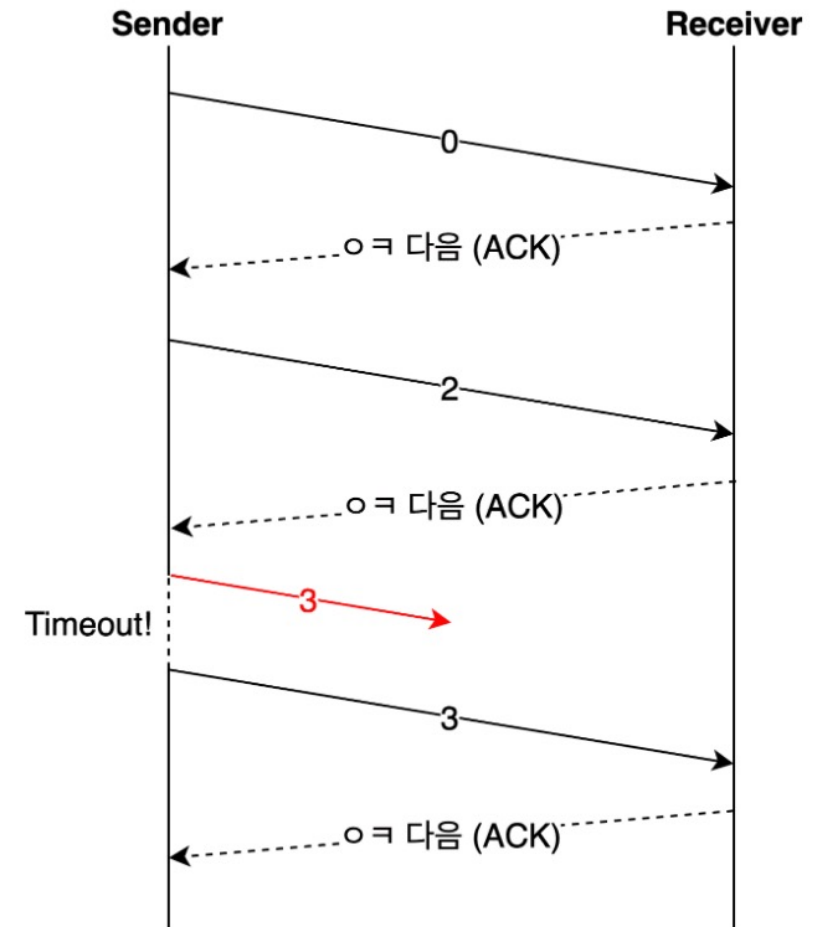
1. Stop and Wait ARQ
2. Go - Back - n ARQ (슬라이딩 윈도우)

04. CONTENTS

Stop and Wait ARQ

ACK를 받고나서 다음 데이터를 보내는 방식

일정 시간을 지나 타임아웃이 발생하면
이전 데이터를 재전송함

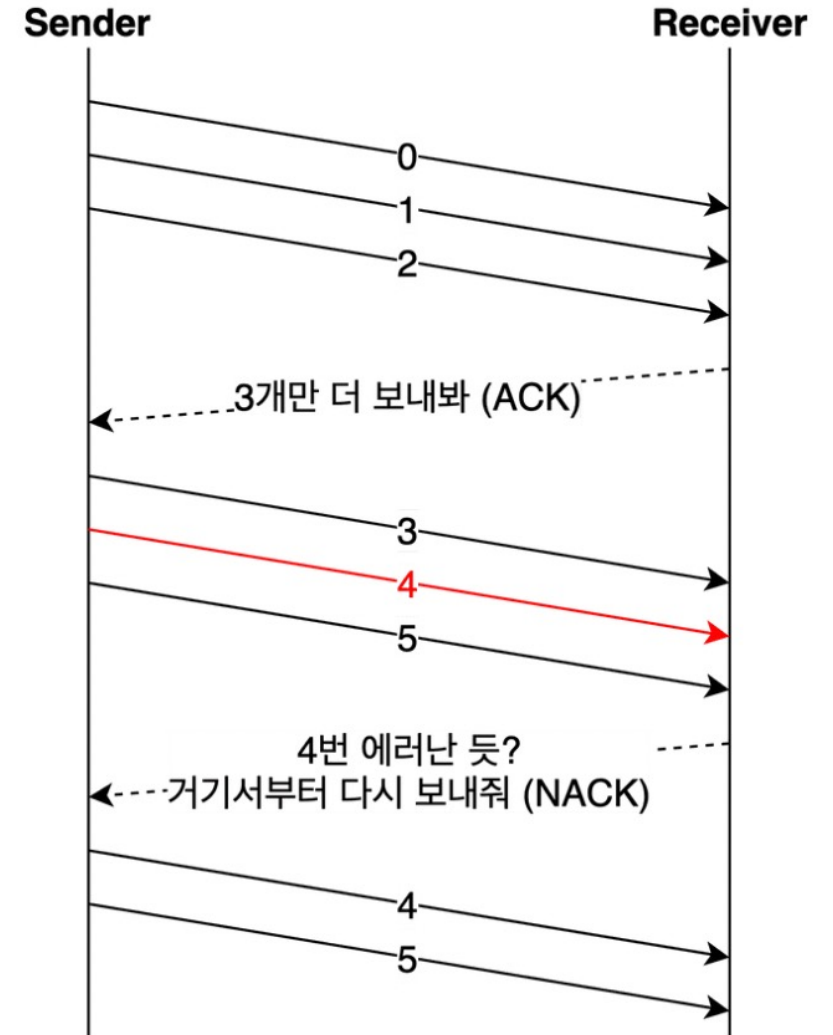


04. CONTENTS

Go - Back - n ARQ (슬라이딩 윈도우)

연속으로 데이터를 보내다가 오류가 발생한 지점
부터 재전송하는 방식

에러가 발생한 부분 이후의 데이터를 모두 삭제한
뒤 재전송함

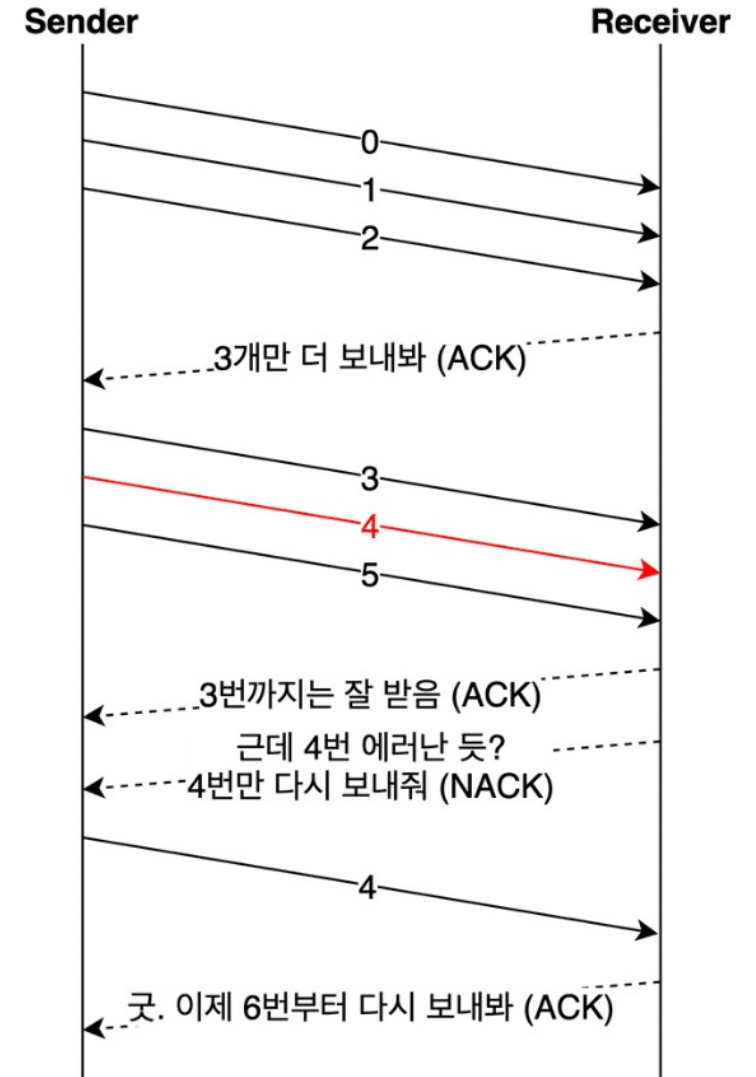


04. CONTENTS

Selective Repeat

오류가 발생한 데이터만 재전송하는 방식

수신 측 버퍼의 데이터가 순차적이지 않기 때문에
정렬과정이 추가로 필요하고 별도의 버퍼가 필요함





Thank you

TCP

