컴망 9주차

TCP:

신뢰성

Inform 을 보내면 ack 이 옴.

Bidirection

Ack 과 함께 자기가 보내고 싶은 inform 이랑 보낼 수 도 있음 : piggyback

재전송의 time out

Data 를 보내면 ack 을 기다리는 timer 설정

Retransmission time out : RTO

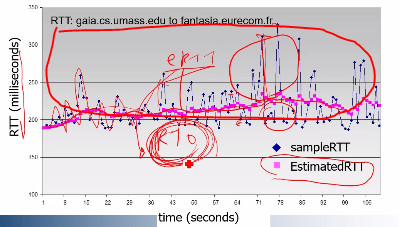
RTT: round trip time

을 측정해서 평균값을 정해서 rto 값을 구한다.

근데 통신을 안 기다리는데 다음 무슨 값이 필요한지 ack 보내는게 가능한가 저번에 설명한 거를 더해서

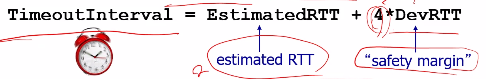
가중치 Alpha (a): 현재 측정된 RTT 값에 가중치를 둘 것인가, 이전의 평균에 가중치를 두고 계산할 것인가?

Typical value :a = 0.125 경험적인 값



과연 효율적인 RTO 값인가?

수정 => estimaition moving average DEVRTT 구함, average 와 sample 의 폭의 평균을 구하기.



상수 4 도 경험적으로 얻은거임

IP 의 위에서 어떻게 TCP 가

Reliable data transfer 을 제공 하는가?

Pipelined segment 가 뭐였는지 기억안남.

Cumulative acks 🡺 ack 은 축적해서 쌓여야됨 5 ->7로 스킵x

그럼 5까지는 잘 쌓였구나(왓구나) 확인

Fast retransmission 이 있긴함.

Ack 이 중복되서 날아올 경우.. 지금 4번 필요한데 왜 안와~~ \*5번

이런느낌.

ACK traffic 양을 줄여보자… delayed ack. : inform 을 뭉쳐서 ack 을 보냄

67 P 오른쪽거 간단하게 보기. (delayed ack)

Fast retrainsmit

아직 내가 ack 안보내서 내가 저쪽에다 계속요청해서, Ack 이 중복되서 날아오기 때문.. 지금 4번 필요한데 왜 안와~~ \*5번

이런느낌.

한 3개정도 계속 요청하면 retrains.

Buffer 이야기

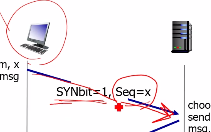
Recv window 에 대해 이야기함

Socket buffer 의 사이즈는 어느정도가 적당함? ( 근데 이거 kernel buffer 아닌가? Application buffer 가 아니라?)

통신 시작할 때

각 buffer 같은 거 정해주고 생성.

내가 줄 Sequence 번호도,



Seq = x 로 시작하는 통신좀 합시다.

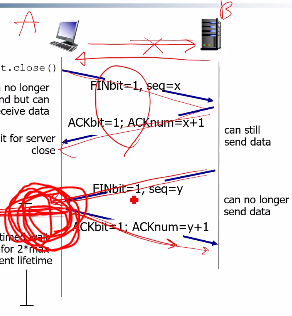
X를 잘 받고 나면 x+1 를 acknum 으로 보내줌

Finite state machine

Plant == 코넥트 합시다 하고 먼저 이니시에이팅 하는 사람.

통신 끝날 때

FIN 해서 보내기



저기서 A를 닫아버리면

B 가 ack 을 받았는지 (RTO ) 확인이 안되서 다시 보내는 FIN 을 처리 해줄 수 없기 때문에 좀더 기다렸다가 닫는다.

Shutdown

Congestion : 혼잡하담.

Loss packet : overflow

Long delay : queing 에서 대기

Loss , delay 가 생겼다 -> congestion 때문

1. Network 에서 Source 한테 알려줌
2. Source 가 congestion 을 detection
   1. Loss or delay 가 발생되면 congestion 을 detect
      1. Data 를 보냈는데 ack 않옴 => loss
      2. Delay 는 RTT 가 자꾸 올라감 => Delay

Congestion control 방법은 ?

Multiplicative decrease

점점 transmission rate 를 올리다가 loss 가 발생하면 rate 를 떨어뜨린다.

Cwnd : congestion window size (window size 는 buffer size 아닌가?)

연속으로 얼마만큼 보낼 수 있는가?

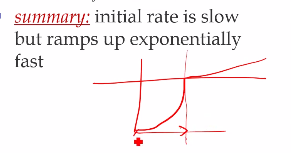
Cwnd 를 올리다가 loss 발생, =>훅 감소

TCP sending rate : cwnd/RTT

Initially cwnd == 1MSS

MSS == maximum segmentation size

?예시에는 두 배씩 커지는데?



앞쪽은 exponential : slow starter

뒤에 linear 은 congestion avoidance

Thresh hold? 저기 linear 로 바뀌는 선: Slow Stater Thresh hold

Loss by timeout

Cwnd to 1MSS, 그리고 다시 반복하며 올린다.

3개 중복 하는 거 나오면 또 처리하는 방법있음.

Tcp 종류 여려개 : congestion control 여러 개

End to end 만 종류 통일해주면 되기 때문에 괜찮음.

타호는 : event 구분 x

Reno : event 구분 O

둘 다 loss 는 생겼지만 ack dup 은 그래도 loss 에 대한 반응은 왔기 때문에 congestion 의 환경 상태가 그나마 낫다고 판단한 것이 reno

=== 총평 : 소 잃고 외양간 고침===

미리 예측해서 외양간 고치는 건 어떻게할까?

🡺Delay 🡺 RTT 계산하여 미리 LOSS 차단 하자!

TCP Vegas 의 방법