

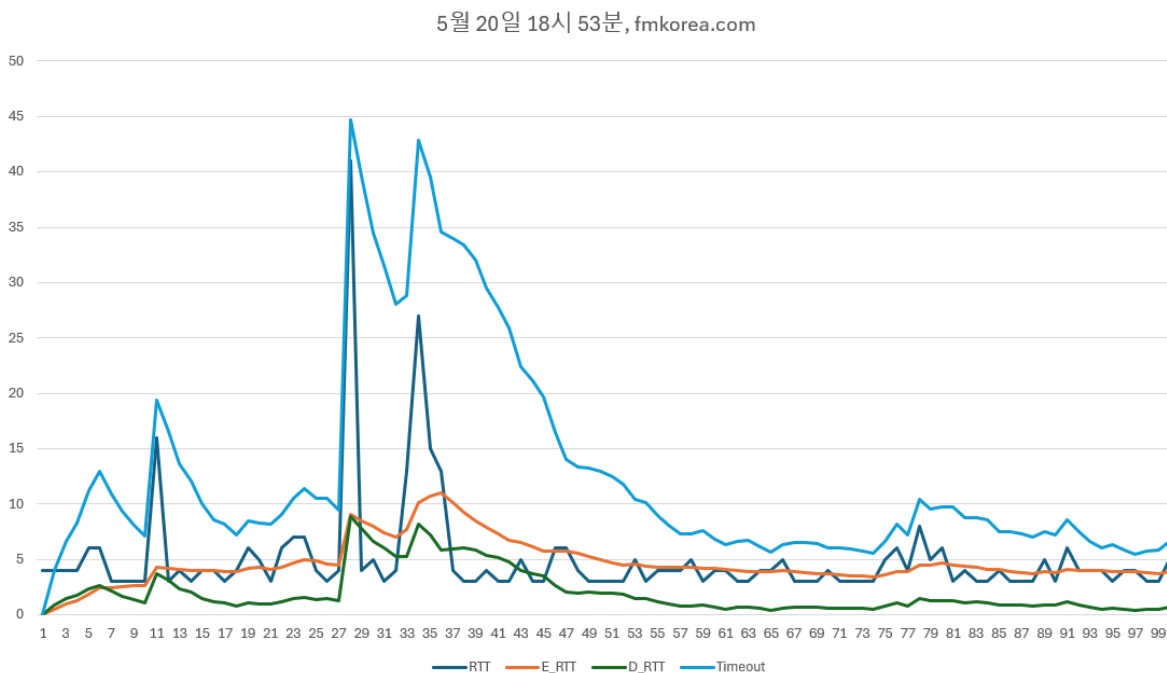
Homework #2 – TCP 동작 프로그래밍

김재학

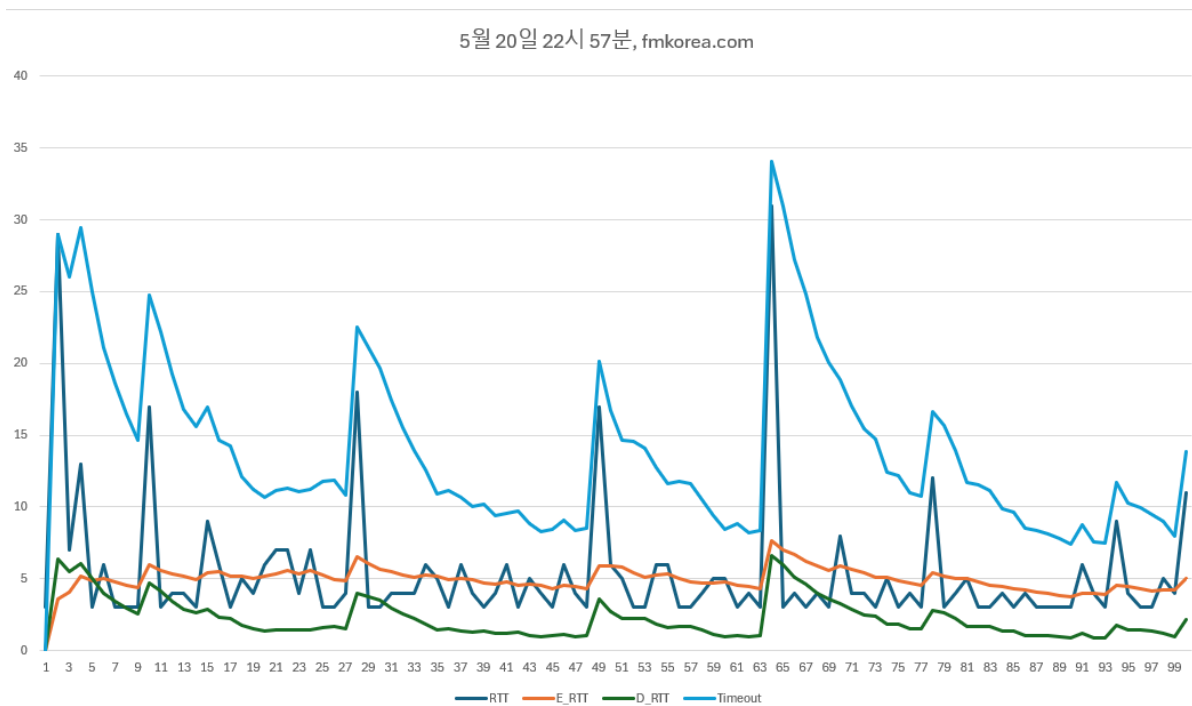
(소프트웨어학과, 202220757)

1. TCP timeout 계산 프로그램

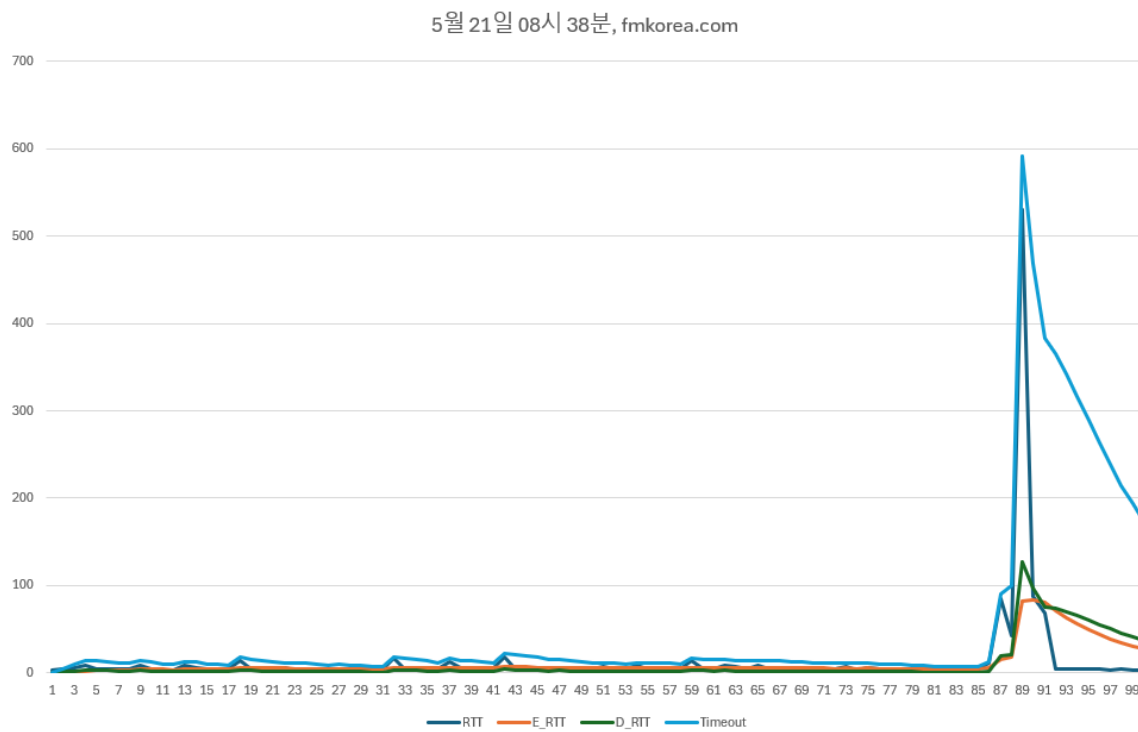
본 프로그램의 목적은 특정 사이트에 ping을 하여서 구한 RTT를 통해 TCP timeout 값을 구하여 오전, 오후, 저녁의 시간대에 따라 어떤 변화가 있는지 관찰하는 것을 목적으로 한다. 이에 따라 특정 사이트로 ping을 1회 보내 RTT 값을 측정, 이를 통해 Estimated RTT, Dev RTT를 계산하여 최종적으로 timeout 값을 구하는 코드를 작성하였고, 이를 100회 반복하였다. 최종적으로 모든 값을 csv파일로 출력하여 이 결과를 분석하였다. 시간대에 따른 변화를 쉽게 관찰하기 위해 국내에서 ping이 가능한 사이트 중 사용자 수가 많은 “fmkorea.com”으로 아침, 점심, 저녁, 밤 시간에 ping을 100회 실시하였다.



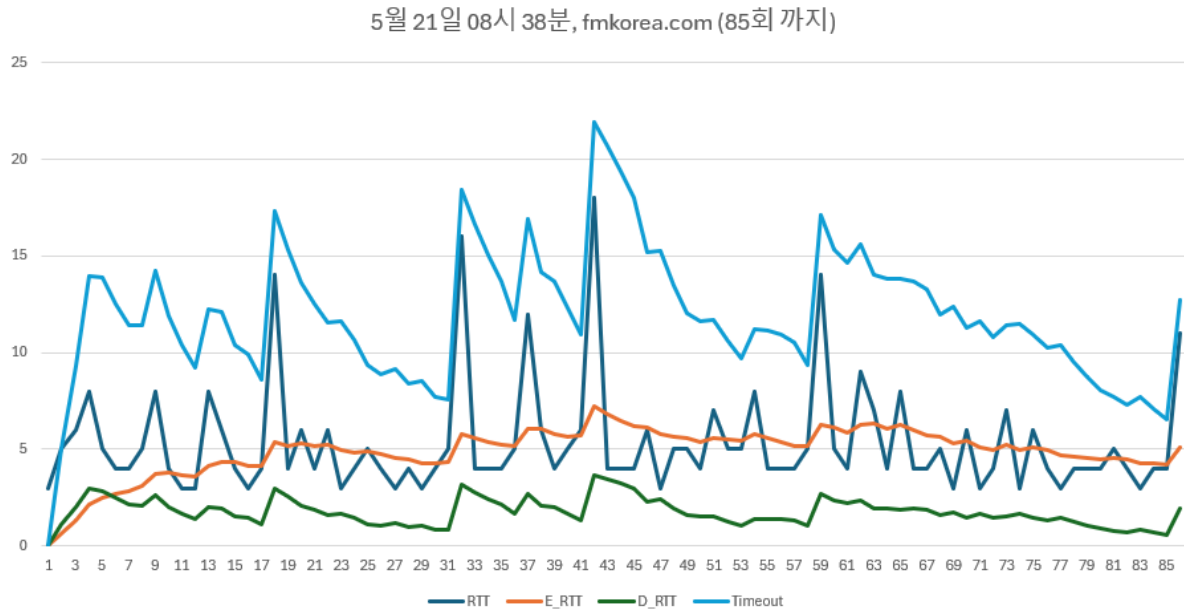
[그림 1] 5월 20일 18시 53분



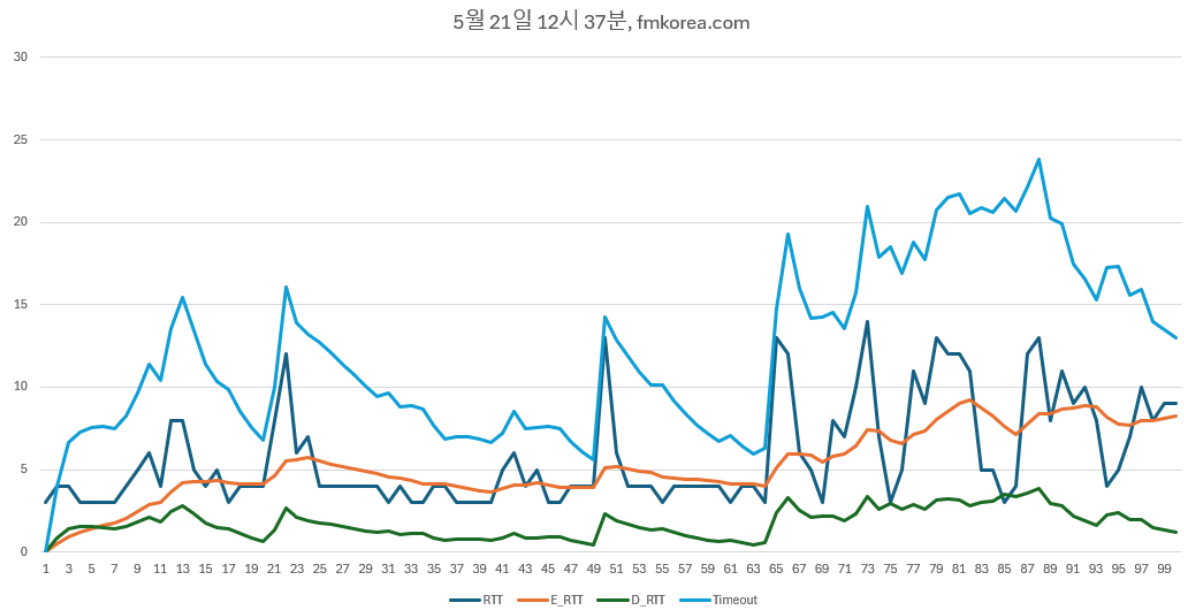
[그림 2] 5월 20일 22시 57분



[그림 3] 5월 21일 08시 38분 - 1



[그림 4] 5월 21일 08시 38분 - 2



[그림 5] 5월 21일 12시 37분

실제 데이터 측정은 위 그림의 순서대로 저녁, 밤, 아침, 점심 순으로 측정되었다. 또한, [그림 3]과 [그림 4]의 경우 같은 값의 데이터로 가독성을 위해 [그림 4]에서 [그림 3]의 일부 데이터만 표시하였다.

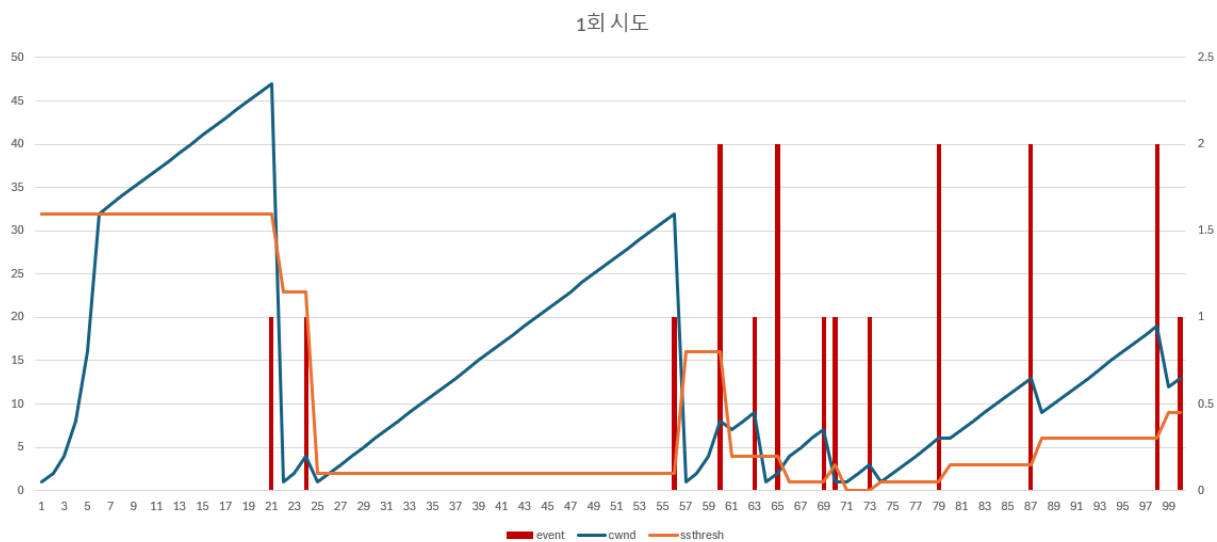
데이터를 측정하여 그래프를 그렸을 때, timeout 값의 규칙성을 찾기는 어려웠다. 하지만 시간대별로 사용자의 수가 달라짐을 고려하였을 때, timeout 값이 어떻게 변하는가를 확인할 수는 있었다.

국내 커뮤니티 사이트의 경우 대체로 저녁에서 새벽까지의 시간대에 접속자의 수가 늘어난다. Timeout을 계산한 그래프를 보면, 해당 시간대에 측정된 결과를 나타내는 [그림 1]과 [그림 2]에서 timeout 값이 30을 초과하는 경우가 나타남을 알 수 있다. [그림 3]에서 급격하게 발생한 timeout 값이 예외적인 경우라 가정하면 다른 시간대와 비교하였을 때 저녁에서 밤의 시간대에 사용자 수가 급증하면서 RTT 값이 증가하여 timeout 값이 늘어났다고 추측할 수 있다.

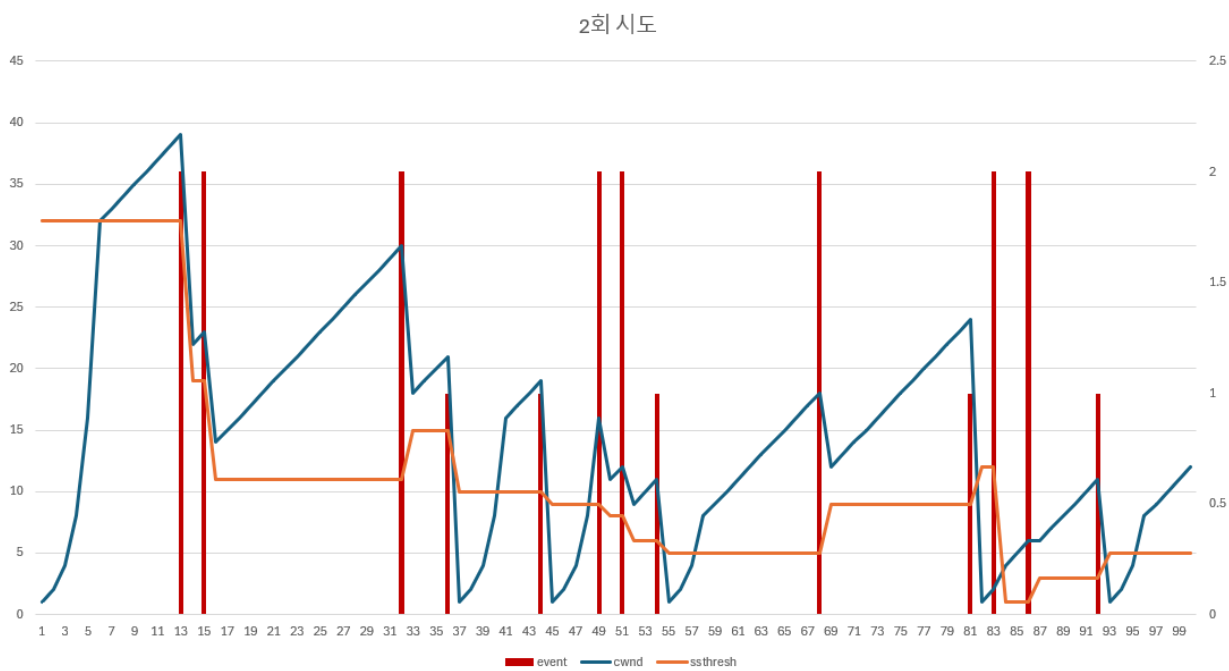
그와 반대로 아침, 점심 시간대의 결과인 [그림 4]와 [그림 5]에서는 timeout 값이 25를 넘지 않는다. [그림 3]에서는 [그림 4]의 이후 부분이 표시되어 있는데, timeout 값이 급격하게 상승한 부분이 있다. 이는 모종의 이유로 네트워크 환경이 안 좋아진 순간이라고 추측된다.

2. TCP congestion control 프로그램

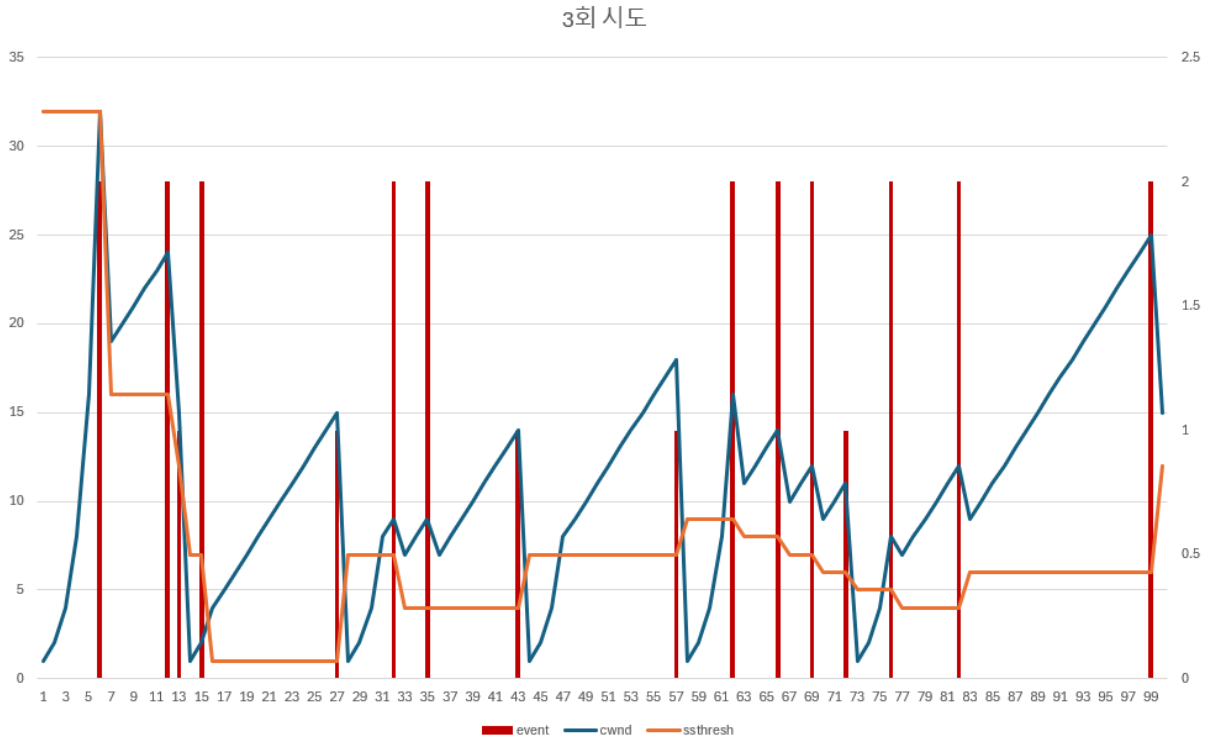
본 프로그램 작성을 통해 Slow start, Congestion Avoidance, Fast Recovery를 결합한 TCP congestion control 과정을 시뮬레이션 하였다. 해당 프로그램에선 문제에서 주어진 상황처럼 초기값들을 설정하고, 일정 확률로 이벤트를 발생시켜서 cwnd 와 ssthresh 값의 변화를 측정하였다. 해당 코드 또한 100회의 라운드를 실시, 각 라운드마다 cwnd, ssthresh의 값과 event 발생여부(0: 정상 수신, 1: timeout, 2: 3-duplicate-ACKs)를 csv 파일로 기록하였다. 이를 3회 시도한 결과는 아래와 같다.



[그림 6] 시도 1회차



[그림 7] 시도 2회차



[그림 8] 시도 3회차

각 시행 결과를 나타낸 그래프 [그림 6, 7, 8]에서 적색의 막대는 event 발생 시점을 의미한다. 막대가 1을 나타내는 경우 timeout, 2를 나타내는 경우 3 dup. ACK이 발생했음을 나타낸다. 막대가 없는 곳은 정성적으로 데이터를 수신한 상황이다. 또한, cumulative ACK을 수신하는 상황을 가정하였다.

해당 그래프들을 통해 볼 수 있듯이 threshold 값보다 cwnd 값이 작은 경우엔 cwnd가 두배씩 증가하여 곡선의 형태를 띄는 것을 볼 수 있다. 그리고 threshold 값을 초과한 이후부터는 cwnd가 1씩 증가하여 선형으로 증가하는 모습을 볼 수 있다.

각 시도에서 event 발생횟수는 다음과 같다. 1회차 시도에서는 timeout이 8회, 3 dup. ACK이 5회 발생되었고, 2회차 시도에서는 timeout이 5회, 3 dup. ACK이 8회 발생하였으며, 3회차 시도에서는 timeout이 5회, 3 dup. ACK이 11회 발생하였다. 이 결과를 통해 설정한 확률대로 event가 잘 발생하였다고 볼 수 있다.

Event 발생 시 threshold 값과 cwnd 값은 fast recovery의 규칙에 따라서 감소하는 모습을 볼 수 있다. Timeout 발생시 threshold 값은 cwnd의 절반으로, cwnd는 1로 설정되며, 3 dup. ACK 발생시 threshold 값은 cwnd의 절반으로, cwnd는 기존 cwnd의 절반에 3을 더한 값으로 설정된다.