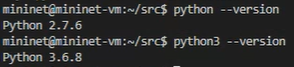
Assignment 3

2016310932 배현웅

* Environment



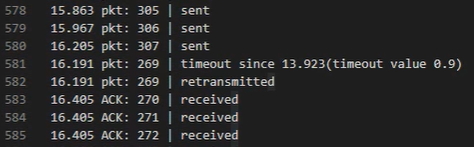
Evaluation

* Evaluation

1. Sender side
   1. Sending – OK

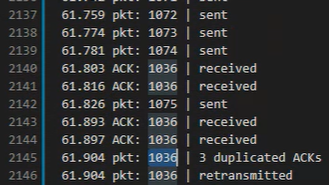


* 1. Logging – OK





* 1. 3 duplicated detect and retransmit – OK



* 1. Time-out retransmit – OK

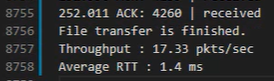


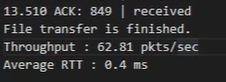






* 1. Write goodput and average RTT at the end of transmit - OK





테스트를 위해서 작은 파일을 사용하였습니다. 테스트에 쓰인 전송 패킷 개수는 총 849, 0번째 패킷은 수신자쪽에 시작을 알리며 dst filename을 알려주는 패킷, 마지막 패킷인 848번째로 보내고 마지막으로 받는 ack 849번을 보낸후 전송자 쪽은 프로그램을 종료합니다.

패킷을 보내는 영역은 pos 부터 pos+windowsize입니다. Pkt\_sender에서 패킷을 전송할 수 있는 지 확인하는데, 그 함수에서 보내고 싶어하는 패킷 번호는 변수 i를 통해 결정됩니다. Ack 를 받을때마다 pos를 이동시켜 packet window를 변경합니다.

Logging : 모든 로그 작성은 수신자에게 패킷을 전송한 이후에 이루어집니다. Threading 모듈을 이용하여 독립적으로 로그를 기록할 수 있게 하였습니다.

3 duplicated detect and retransmit : ack\_recevier 함수/스레드에서 관리합니다. 매번 ack 를 받을 때마다 seq\_num를 확인하여서 data\_ack\_info 에 저장을 합니다. 누적되어서 3번 중복되어서 ack가 온다면 data\_ack\_info 에서 해당영역이 4 를 넘어가므로 그 것을 확인하여 패킷을 보낸이후 로그를 기록합니다.

Time-out retransmit : window에서 제일 첫번째 패킷 번호인 pos가 보내진 시간을 pkt\_t\_info 에 기록해놓고, 그것을 확인합니다. 만약 현재 시간과 보내진 시간의 차이가 average RTT 의 두배라면, timeout이라 간주하고 retransmit한 이후에 로그에 기록합니다.

1. Receiver side

메인에서 Packet\_process 함수를 불러와서 패킷을 받고, 파일에 저장하고 로그에 남기는 일을 합니다.

While 루프를 통해서 sender로부터 받을 패킷을 처리하고, 0번이라면 파일을 미리 열고 닫아 copy와 logging을 할 준비를 합니다. 그후 expected 라는 변수를 이용해 순서대로 packet sequence number를 받습니다. 그리고 로그를 기록합니다. 만약 원하지 않는 것이라면, data\_info라는 dictionary 변수를 통해서 data를 저장해둡니다. 그후 순서대로 들어오는 대로 파일을 write합니다.

그 이후 ACK sequence number 를 sender에게 보내고 Ack 보냈다고 로그를 기록합니다.

얼마나 보낼지 packet\_number 에 대한 정보를 보내고 있으므로 packet\_num 보다 하나 큰 것을 seq\_num 으로 받았다면 수신을 종료합니다.

Sender side 결과

