

Assignment 3

2016310932 배현웅

- Environment

```
mininet@mininet-vm:~/src$ python --version
Python 2.7.6
mininet@mininet-vm:~/src$ python3 --version
Python 3.6.8
```

```
mininet@mininet-vm:~/src$ cat /etc/issue
Ubuntu 14.04.6 LTS \n \l
```

Evaluation

- Evaluation

1. Sender side

A. Sending – OK

```
mininet@mininet-vm:~$ make diff
diff ./src/EMMA.word ./dst/tmp1.txt
mininet@mininet-vm:~$
```

B. Logging – OK

```
578 15.863 pkt: 305 | sent
579 15.967 pkt: 306 | sent
580 16.205 pkt: 307 | sent
581 16.191 pkt: 269 | timeout since 13.923(timeout value 0.9)
582 16.191 pkt: 269 | retransmitted
583 16.405 ACK: 270 | received
584 16.405 ACK: 271 | received
585 16.405 ACK: 272 | received
```

```
1221 File transfer is finished.
1222 Throughput : 18.89 pkts/sec
1223 Average RTT : 1.3 ms
```

C. 3 duplicated detect and retransmit – OK

```
2137 61.759 pkt: 1072 | sent
2138 61.774 pkt: 1073 | sent
2139 61.781 pkt: 1074 | sent
2140 61.803 ACK: 1036 | received
2141 61.816 ACK: 1036 | received
2142 61.826 pkt: 1075 | sent
2143 61.893 ACK: 1036 | received
2144 61.897 ACK: 1036 | received
2145 61.904 pkt: 1036 | 3 duplicated ACKs
2146 61.904 pkt: 1036 | retransmitted
```

D. Time-out retransmit – OK

```
53 2.283 pkt: 31 | sent
97 3.523 pkt: 31 | timeout since 2.283(timeout value 0.6)
98 3.523 pkt: 31 | retransmitted
7661 224.363 pkt: 3731 | sent
7737 227.383 pkt: 3731 | timeout since 224.363(timeout value 1.4)
7738 227.383 pkt: 3731 | retransmitted
```

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

```
13.510 ACK: 849 | received
File transfer is finished.
Throughput : 62.81 pkts/sec
Average RTT : 0.4 ms
8755 252.011 ACK: 4260 | received
8756 File transfer is finished.
8757 Throughput : 17.33 pkts/sec
8758 Average RTT : 1.4 ms
```

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

패킷을 보내는 영역은 pos 부터 pos+window size 입니다. Pkt_sender 에서 패킷을 전송할 수 있는 지 확인하는데, 그 함수에서 보내고 싶어하는 패킷 번호는 변수 i 를 통해 결정됩니다. Ack 를 받을때마다 pos 를 이동시켜 packet window 를 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_num 를 확인하여서 data_ack_info 에 저장할 합니다. 누적되어서 3 번 중복되어서 ack 가 온다면 data_ack_info 에서 해당영역이 4 를 넘어가므로 그 것을 확인하여 패킷을 보낸이후 로그를 기록합니다.

Time-out retransmit : window 에서 제일 첫번째 패킷 번호인 pos 가 보내진 시간을 pkt_t_info 에 기록해놓고, 그것을 확인합니다. 만약 현재 시간과 보내진 시간의 차이가 average RTT 의 두배라면, timeout 이라 간주하고 retransmit 한 이후에 로그에 기록합니다.

2. Receiver side

메인에서 Packet_process 함수를 불러와서 패킷을 받고, 파일에 저장하고 로그에 남기는 일을 합니다.

While 루프를 통해서 sender 로부터 받을 패킷을 처리하고, 0 번이라면 파일을 미리 열고 닫아 copy 와 logging 을 할 준비를 합니다. 그후 expected 라는 변수를 이용해 순서대로 packet sequence number 를

받습니다. 그리고 로그를 기록합니다. 만약 원하지 않는 것이라면, data_info 라는 dictionary 변수를 통해서 data 를 저장해둡니다. 그후 순서대로 들어오는 대로 파일을 write 합니다.

그 이후 ACK sequence number 를 sender 에게 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

```
17 0.249 ACK: 1 | sent
18 0.258 pkt: 40 | received
19 0.267 ACK: 1 | sent
20 0.369 pkt: 1 | received
21 0.423 ACK: 2 | sent
```

```
1216 31.412 pkt: 606 | received
1217 31.418 ACK: 607 | sent
1218 File transfer is finished.
1219 Throughput : 19.36 pkts/sec
```

```
8544 226.132 pkt: 4259 | received
8545 226.186 ACK: 4260 | sent
8546 File transfer is finished.
8547 Throughput : 18.87 pkts/sec
```

```
mininet@mininet-vm:~$ make test
sudo python execute_mn.py 40 ./src/n.pdf ./dst/n.pdf
Starting test...
```

```
mininet@mininet-vm:~$ ll ./dst/n.pdf ./src/n.pdf
-rw-r--r-- 1 root root 5961964 Nov 22 21:26 ./dst/n.pdf
-rw-rw-r-- 1 mininet mininet 5961964 Nov 22 18:27 ./src/n.pdf
```