

RDT TRANSMITTER

Scott Jin

Usage

The Program was a simple implementation of RDT(Reliable Data Transfer) 3.0 MODEL. It works by using finite state machine model combined with timeout on transmitter and receiver model in application layer to add security to transport layer UDP.

```
USAGE: make test
TEST: Plz wait two time[unix command] result, then use make diff
INPUT: /
FUNCTION :reliable data transfer over a man-made noisy channel which can drop, random
ize, swap order on packets.
OUTPUT : rendered in output.txt should match the sent file [default to "file_1MB.txt"
] .
```

Screenshot

```
~/Desktop/ece303/2018 ➤ master ● ? make test
time python2 receiver.py > ./output.txt & time python2 sender.py < ./file_1MB.txt &
File sent, sender exited. Plz wait the receiver timeout to make diff. Sent, took 0.0576989650726 s.
real    0m0.214s
user    0m0.066s
sys     0m0.065s
~/Desktop/ece303/2018 ➤ master ● ? Timeouted, plz check use make diff, if any output, wait till the process end[ps -a]. Took 15.0633189678 s.
real    0m15.221s
user    0m0.073s
sys     0m0.070s
~/Desktop/ece303/2018 ➤ master ● ? make diff
diff ./file_1MB.txt ./output.txt
~/Desktop/ece303/2018 ➤ master ● ? make test
time python2 receiver.py > ./output.txt & time python2 sender.py < ./file_1MB.txt &
~/Desktop/ece303/2018 ➤ master ● ? Sender Timeouted. if this messege loops, receiver timeout need to be longer File sent, sender exited. Plz wait the receiver timeout to make diff. Sent, took 10.0664660931 s.
real    0m10.217s
user    0m0.068s
sys     0m0.068s
Timeouted, plz check use make diff, if any output, wait till the process end[ps -a]. Took 25.0700891018 s.
real    0m25.228s
user    0m0.079s
sys     0m0.071s
~/Desktop/ece303/2018 ➤ master ● ? make diff
diff ./file_1MB.txt ./output.txt
~/Desktop/ece303/2018 ➤ master ● ? make test
time python2 receiver.py > ./output.txt & time python2 sender.py < ./file_1MB.txt &
~/Desktop/ece303/2018 ➤ master ● ? Sender Timeouted. if this messege loops, receiver timeout need to be longer File sent, sender exited. Plz wait the receiver timeout to make diff. Sent, took 10.0894458294 s.
real    0m10.249s
user    0m0.079s
sys     0m0.072s
Timeouted, plz check use make diff, if any output, wait till the process end[ps -a]. Took 25.088709116 s.
real    0m25.253s
user    0m0.088s
sys     0m0.075s
~/Desktop/ece303/2018 ➤ master ● ? make diff
diff ./file_1MB.txt ./output.txt
~/Desktop/ece303/2018 ➤ master ● ?
```

Notes

make test give a simple description for prompting, but several scenarios may appear and worth noting:

1. Sender exited prompt suggest the time that sender used to transmit all the file contents. Because the receiver wait a full timeout to quit(15 secs), the whole process performance depends on the first time result `time python2 sender.py < ./file_1MB.txt` Note the makefile is modified from initial kickstart implementation, user should wait two unix `time` command to use make diff to see the output.txt [make diff].
2. If sender timeout keeps prompting, and a timeout appeared before the loop, it means receiver exited because timeout, and the timeout window in receiver [default to 15 secs] should be modified to a larger value to allow successful transmittion.
3. default port number was used, it the port is already used in your host, plz change the API settting in the channelsimulator.py.

Diffuculties

The sender should be able to kill the receiver to reduce the whole process time, but the receiver seems not be able to write the last message to file before it gets killed, so I instead still use a receiver timeout to quit. Still, sender time result should be used for evaluation.

Implementation

1. The transmitter model was coded to mimic the action of FSM as following:

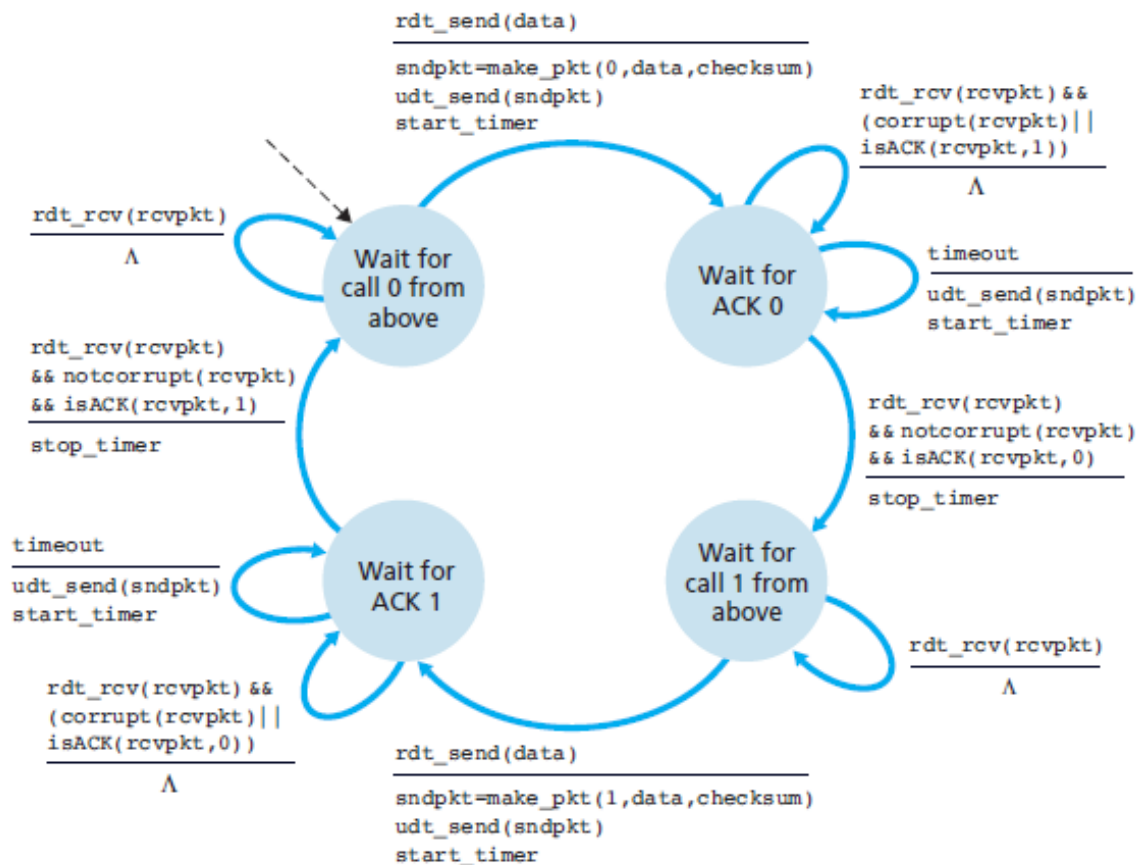
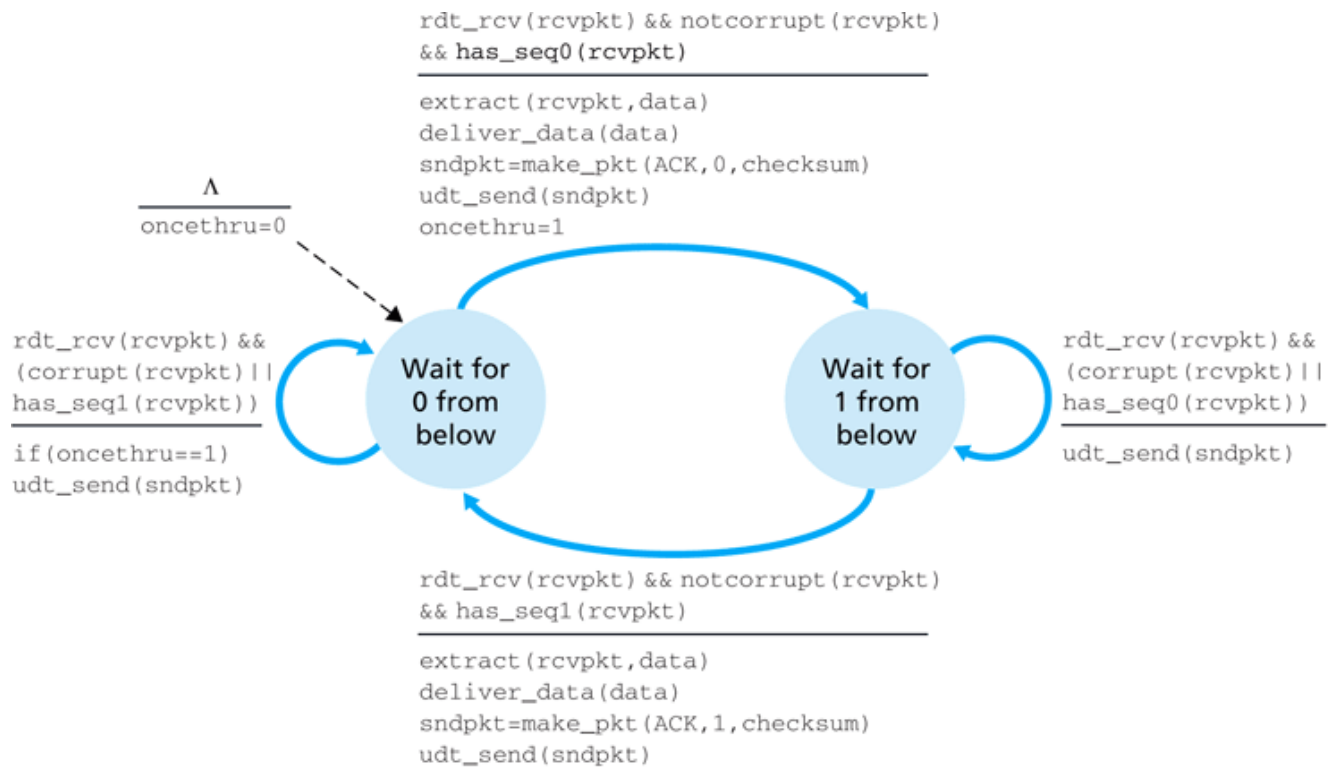
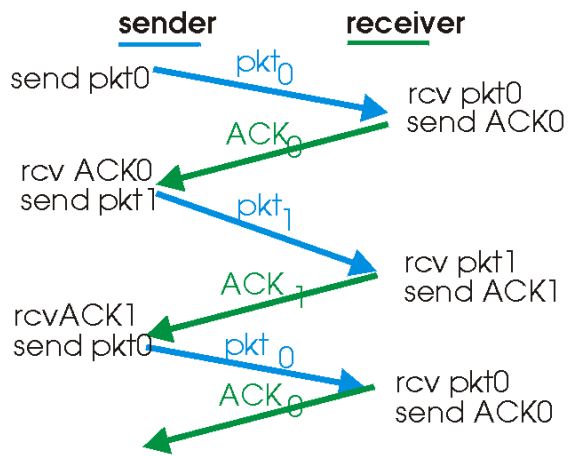


Figure 3.15 ♦ `rdt3.0` sender

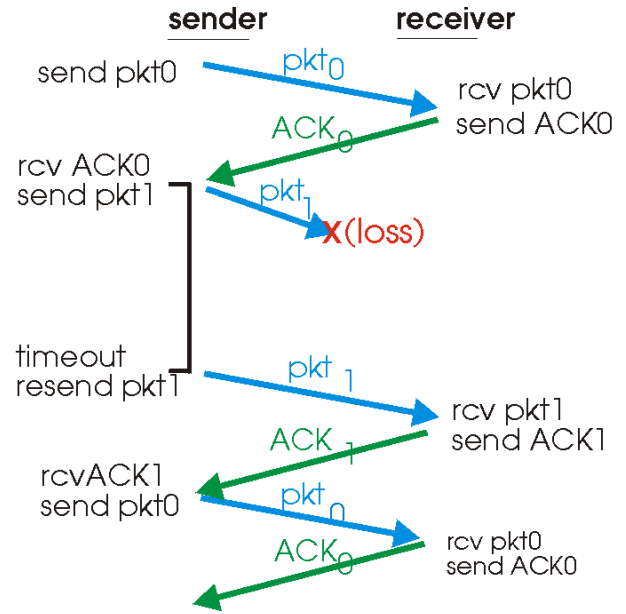
2. The receiver mode FSM is depicted as following:



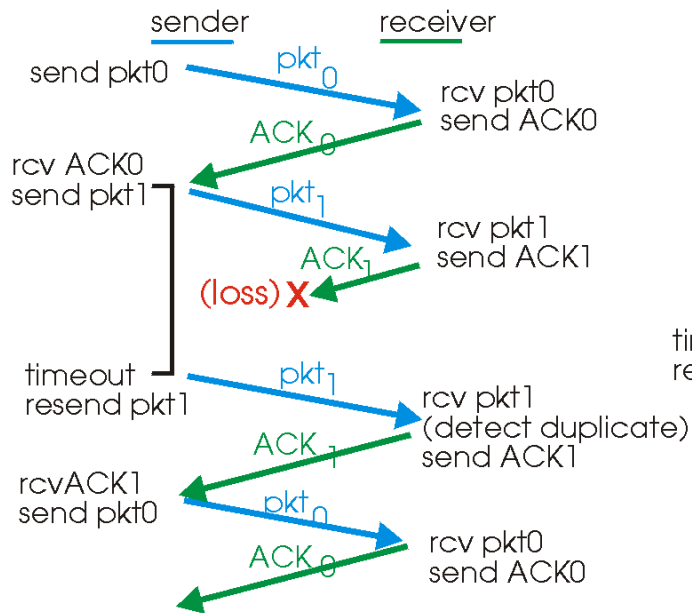
3. transmission in action:



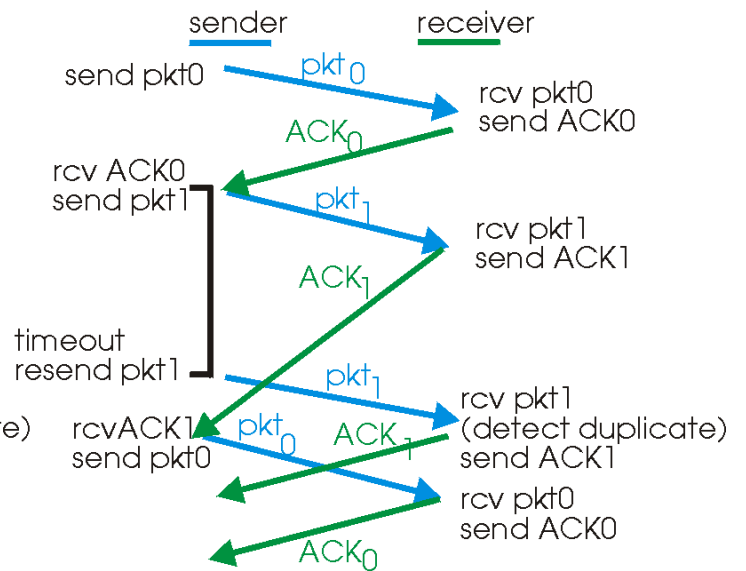
(a) operation with no loss



(b) lost packet



(c) lost ACK



(d) premature timeout