Assignment 3 – Congestion Control

Team Members:

Lawrence (Aohui) Lin

Justin (Zihao) Zhang

Steve (Siyang) Wang

Total Time Put-in:

5 days

**Part A:**

Implemented sudo-tcp style flow control where there is no real buffering, and sender and receiver has the same window size. When window size is one for both sender and receiver, the stop-and-wait principle is employed.

We passed all the tests with various tester options, as well as without tester options:

A description...

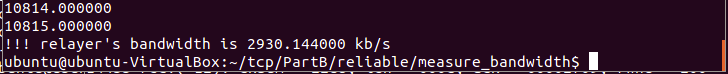
**Part B:**

Implemented TCP style flow control, and congestion control including mechanisms like slow start, fast retransmission. The basic algorithm we came up with matches with the sudo-code provided in the lecture slide, as is appended in the end of the README. Also, we have realized the sender window adjustment based on the congestion window, receiver's window size, and buffer window.

Using the configuration provided in the original config.xml file and the receiver window size specified in the instruction handout (-w 200), and running in the Ubuntu VM environment, we get the following benchmark performance:

* One pair of sender-receiver, file size = 10MB (10^7 bytes)

A description...



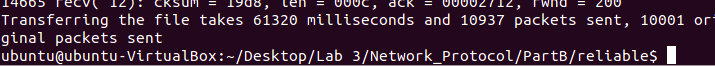
Note:

1. The last original packet sent is an EOF packet
2. Size of each data packet is 16+1000=1016 bytes. Size of EOF packet is 16 bytes.
3. At the time of benchmarking, bandwidth, measured by the tool provided on the assignment page, is 2930.144 kb/s.
4. The packet sent due to re-transmission is not included in the total data transferred

We ran the experiment multiple times and the results all ended up being somewhere between 80-90%.

* Two pairs of sender-receivers, file size = 10MB for both pairs

**Pair 1**:



**Pair 2**:

A description...

We started the two senders at roughly the same time. Notice that the time taken for transmission is very similar between the two pairs. We take the maximum of the two timings in our calculation. Also notice that there are a lot more packets being retransmitted this time, which is expected since the two pairs are competing for the same shared resource (the relayer).

29/

We ran the same experiment with the setup several times, and the utilization rate fluctuated around 90%.

**Part 3 – extra credit:**

In part 3, we are looking for the fairness index of greater 80%. The fairness index is calculated by the following formula:

A description...

Using the two pair case, we have from PartB the following data:

Pair 1: utilization = (10000\*1016+16)\*8/61320/2930 = 0.452

Pair 2: utilization = (10000\*1016+16)\*8/60601/2930 = 0.457

Fairness for two pairs = (0.452 + 0.457)^2 / 2 / (0.452^2+0.457^2) = 99%

Pair 1: utilization = 0.301

Pair 2: utilization = 0.371

Pair 3: utilization = 0.392

Fairness for three pairs = (0.301+0.371+0.392)^2 / 3 / (0.302^2 + 0.371^2 + 0.392^2) = 95%

Situation with four pairs and above with different file size, after tested, the fairness index remaining above 80%, which passes the evaluation for Part3 extra credit.

APPENDIX:

