IT 304: Computer Networks

Lab # 10: Evaluation of Throughput and Friendliness of TCP *

For the Week of October 21, 2013

Pre-lab preparation:

- Read the relevant portions of text book on the basic operation of TCP and about its variants.

1. **Aim:** The purpose of this lab is to compare the throughput of TCP evaluated by simulations with that of the corresponding approximate expressions given in the literature. We also explore friendliness of TCP flows towards each other and the unfairness that TCP is subjected when competing with UDP.

2. **Expressions for TCP throughput:** Two (approximate) expressions exist for TCP throughput Q in the presence of random losses. The first one known as the *square root* formula does not account for timeouts and receiver window size and is given as

$$Q = \frac{S}{RTT}\sqrt{\frac{3}{4p}}\,,$$

where S denotes the packet size in bits, RTT denotes the round trip time and p denotes the loss probability.

The second expression derived in the paper (Modeling TCP Reno Performance: A Simple Model and Its Empirical Validation, by J. Padhye et al, IEEE/ACM Transactions on Networking, vol. 8, no. 2, April, 2000) is given as

$$Q = \min\left(\frac{rwnd}{RTT}, \frac{S}{RTT\left(\sqrt{\frac{4p}{3}} + 4\min\left(1, 3\sqrt{\frac{6p}{8}}\right)p(1+32p^2)\right)}\right)\,,$$

where $rwnd$ denotes the receive window size and S, RTT and p are defined as above.

Note that in both the expressions we assume that the delay acknowledgments are enabled. The second expression accounts for the timeouts and receiver window size that were ignored in the first expression.

3. **Lab Scenarios:** In this part we validate the two expressions given above via NS2 simulations. To this end consider a simple network which consists of a source and destination that are connected together by a link of rate 1 Mbps, propagation delay 100 milliseconds and a drop tail buffer at its input of very large size (ensure this by setting the queue limit to a very large value.)

(a) Create a TCP Newreno agent and connect it to the source. Next create a TCP sink in which delay acknowledgments are enabled and connect it to the destination. Set the TCP packet size to 500 bytes. Ensure that an FTP agent is connected to TCP.

(b) Compute the delay-bandwidth product of the network given above and set the receiver window and slow start threshold to the delay-bandwidth product.

(c) Run the TCP connection for 1000 seconds and answer the following questions:

- What should be the link utilization?
- What should be the queue size at the input of the link? If the receiver window is set to a very large value, how should we set the buffer at the input of the link in order to fully utilize the available bandwidth? Validate with the simulation your answer about the utilization of the link and the queue size.
- Plot the window size versus time.

(d) We now insert on the link an object that randomly drops TCP packets with probability p. This object belongs to the class `ErrorModel`. The code is as follows:

```
set lossmodel [new ErrorModel]
$lossmodel set rate_ p
$lossmodel unit packet
$lossmodel drop-target [new Agent/Null]
set lossylink [$ns link $S $D]
$lossyLink install-error $lossmodel
```

Next do the following:

- Plot the congestion window of the TCP connection for two values of p: 10^{-4} and 10^{-2} and interpret the results.
- For each probability, compute the utilization of the link, the utilization given by the first expression, and the utilization given by the second expression.
- How does the performance of TCP vary as p increases? Which one of the two formulas approximate better the throughput of TCP?
- Where can we consider the square root formula as a good approximation of TCP throughput, and why it does not work in the other regimes?

(e) We next explore the friendliness of TCP connections to each other. For this purpose consider the same network as given above. Run on this network a New Reno TCP connection of packet size 500 bytes, and of very large receiver window and slow start threshold. Keep this connection running for 1500 seconds. At time 500s, run another TCP connection between the source and destination for 500 seconds. Next read the number of bytes of each flow that cross the link, divide this number by 10 seconds, compute the throughput, save it in a file, then plot it. What do you conclude? Do both flows share fairly the available bandwidth?

(f) Repeat now the simulation but this time with a CBR UDP flow of 800 kbps instead of the second TCP flow. CBR packets are of 500 bytes each (Note: For this you must first create the UDP agent then the CBR traffic generator). Do both flows share fairly the available bandwidth? Who is favored in this scenario?

(g) In the two sub-parts immediately above the flows can be monitored using the flowmonitor object (to facilitate this one has to associate flow IDs to both the flows. This can be accomplished as follows: `$tcp1 set fid_ 1` and `$tcp2 set fid_ 2`) as follows:

- The flowmonitor can be created and attached to a particular link with the commands: `set fmon [$ns makeflowmon Fid]` and `$ns attach-fmon [$ns link $S $D] $fmon` .

- To track the statistics for a particular flow, a classifier must be defined so that it selects the flow based on its flow id, this can be done as follows: `set fclassifier [$fmon classifier]`. The flow statistics for flows with ID 1 and 2 can be done as follows: `set flowstats1 [$fclassifier lookup auto 0 0 1]` and `set flowstats2 [$fclassifier lookup auto 0 0 2]`. These two commands must be issued a couple of seconds after the simulation to ensure that one packet of each flow has been monitored. For example, the command for flow with flow ID 1 (i.e. `set flowstats1 [$fclassifier lookup auto 0 0 1]`) can be issued at 1 second and the command for flow with ID 2 (i.e. `set flowstats2 [$fclassifier lookup auto 0 0 2]`) can be issued at 501 second.
- Next one can probe the two objects `flowstats1` and `flowstats2` every 10 seconds and read the number of bytes of each flow that cross the link. This can be done as follows for `$flowstats1`: `set bytes [$flowstats1 set barrivals_]`.