IN THE NAME OF ALLAH

Implementation of “TCP over Large Buffers When Adding Traffic Improves Latency” paper on Testbed

Professor
Dr. Mohammad Nassiri

Authors
Parisa Abdolmaleki, Ali Ahmadi Dehmand

Nov. 2017

Contents

Testbed topology

```
| (Client) |    Ethernet    | (Router) |    Ethernet    | (Server) |
|  Linux   |---------------|   Unix   |---------------|  Linux   |
|  Debian  |               |  FreeBSD |               |  Debian  |
```

Figure 1. Testbed

You can see the topology of our Testbed in figure 1. It consists of three nodes that connected with Ethernet interfaces and Cat6 cable. The client and server run Debian Linux 8 and the router runs FreeBSD. We have installed some required packages on every node that explained in bellow.

Client and server

- ipmt : Traffic generator tool
- tcpdump : Packet capture tool
- tcptrace : Packet analyzer tool
- gnuplot : Plot drawing tool
- ssh, sshd : remote login shell daemon and client

Router

- ipfw : dummynet tool
- ssh, sshd : remote login shell daemon and client

Testbed configuration

You can see in figure 2 some configurations of Testbed nodes such as IP address's , user names. Note that the router has two interfaces also we have done some configurations related to ssh that explained bellow and enabled IP forwarding in router.

Ssh configs

You must enable root login permission on every machine this is necessary to run some scripts, to do this change "PermitRootLogin without-password yes" line to "PermitRootLogin yes" in ssh config file in /etc/ssh/sshd_config and also you must generate ssh key on every machine and copy the key to the others, this allows ssh to login a machine without asking for password.You can do this with bellow commands. Run this command on every machine in Testbed with another IP's

ssh-keygen -t ras   # generate ssh key
ssh-copy-id root@<IP Address>   # copy ssh key to other machine's

```
|      (Client)      |            |      (Router)      |
| IP:192.168.10.10 |------------| IP-1:192.168.10.1 |            |      (Server)      |
| User:tstclient    |            | IP-2:192.168.11.1 |------------| IP:192.168.11.10 |
                                 | User:tstrouter     |            | User:tstserver    |
```

Figure 2

Scripts for automation

We have written many shell scripts and thair config files to run experiments. Those automate doing configurations, generating traffics, analyzing packets and drawing plots as experiments results. We listed them bellow acoreding to every node on our Testbed.

Client : scripts and config files for client side

- validation_run.sh script for run validation experiment in client side
- experiment_run.sh script for run two experiment in client side
- dnld_clnt.sh script for run downlaod response time experiment in client side
- tbl_luncher.sh an auxiliary script for display result in a table
- config this directory contains config file`s
- plot this directory contains plot`s in png format
- rawdata this directory contains raw information that gathered during run

Server : scripts and config files for server side

- ipmt_srvr.sh script for prepare server for run validation experiment
- dnld_srvr.sh script for run download response time in server side
- config this directory contains config files
- distributions this directory contains tools and scripts for generate distributions numbers
- log scripts will generate some logs that stored in this directory
- plot this directory contains plots in .png format
- rawdata this directory contains raw information that gathered during run

Router : script and config file's for router side

- ipfw_luncher.sh this script will run dummynet for config firewall and adjust link's
- config this directory contains config file`s

Results of experiments

Validation results

We have validated the testbed in a simple experiment.With Dummynet, the router emulates an asymmetric link: 5Mbit/s bandwidth, 42 ms delay with a buffer of 120 packets for the downlink and 800 Kbit/s bandwidth, 52 ms delay, and a buffer of 120 packets for the uplink. We have used a TCP upload concurrently with a 250 pkt/s UDP stream and compare to the same TCP flow without UDP. The results appear in figure 3.

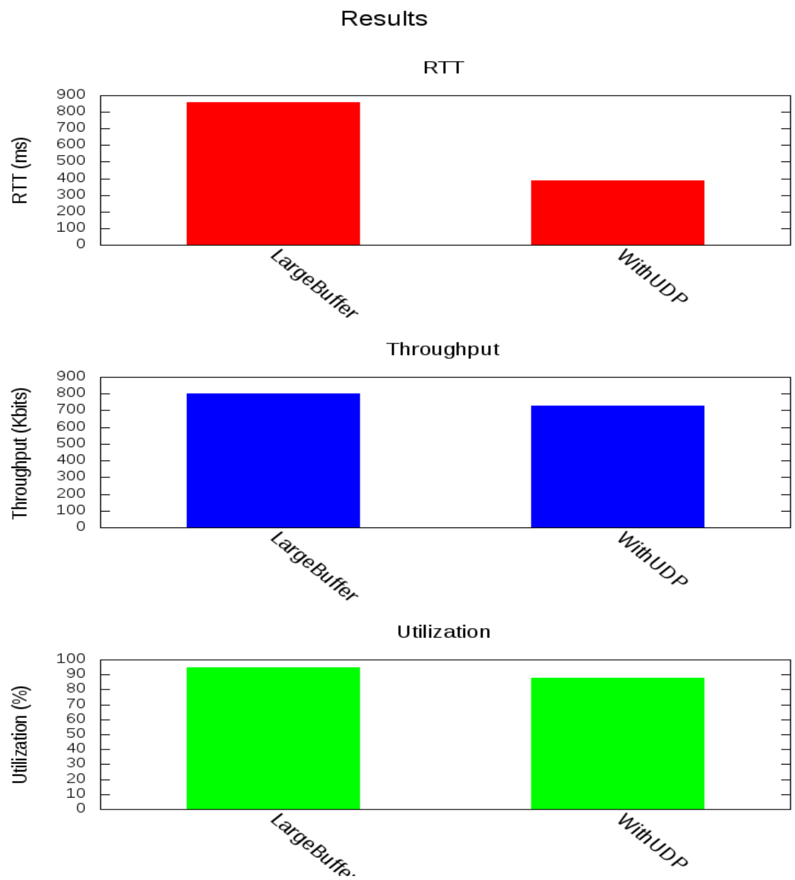

Figure 3

Validation Results

Gauge	Without UDP Stream	With UDP Stream
TCP goodput	801 Kbit/s	729 Kbit/s
Link utilization	95 %	88 %
RTT	859.9 ms	387.8 ms

As you can see in Figure 3 and Table 1, RTT has significant decrease when we add UDP traffic it's about 800 ms to 300 ms and of course as we expected we have a few decrease in throughput and Utilization. This decrease is inevitable because we impose some overload to the network The result of this experiment clearly shows the impact of adding UDP traffic on long delay caused by large buffers.

Table 1

Long lasting connections

In our second experiment, we have two instances of runs under the situation that described in Table 2 and each run in this table consists of three instances of runs, first one with large buffer as you can see in the table which it means 60 and 120 packets for first and second run respectively, next one with UDP and the third one with Small buffer.

Experiment Parameters

Parameter	First Run	Second Run
Download Delay	42 ms	42 ms
Upload Delay	52 ms	52 ms
Downlaod Capacity	288 KB/s	600 KB/s
Uplink Capacity	48 KB/s	96 KB/s
Uplink Buffer	10 and 60 packets	20 and 120 packets
Downlink Buffer	60 packets	12 packets
TCP Variant	New Reno	New Reno

Table 2

The results of this experiments are showed in Figure 4 and Figure 5 in bellow. For more details about experiments you can refer to the main paper.

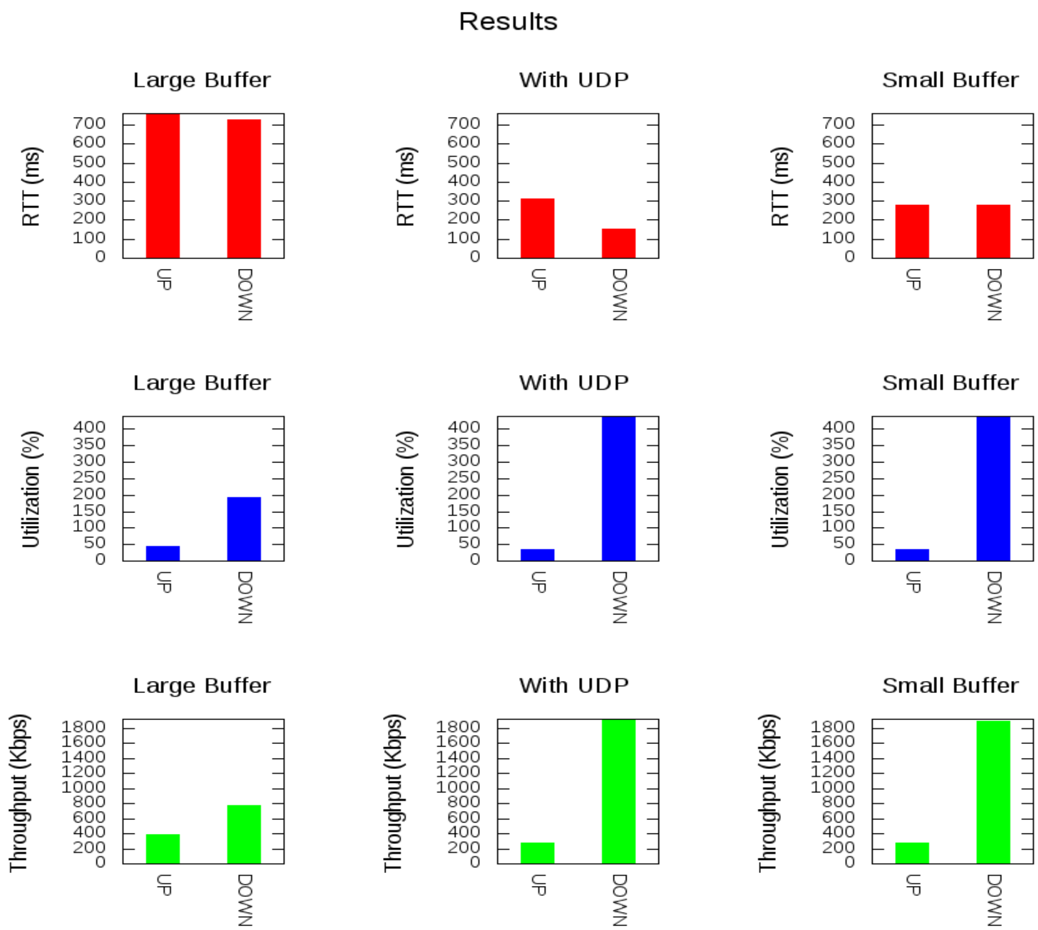
Results
Large Buffer
With UDP
Small Buffer
RTT (ms)
Utilization (%)
Throughput (Kbps)
UP
DOWN

Figure 4

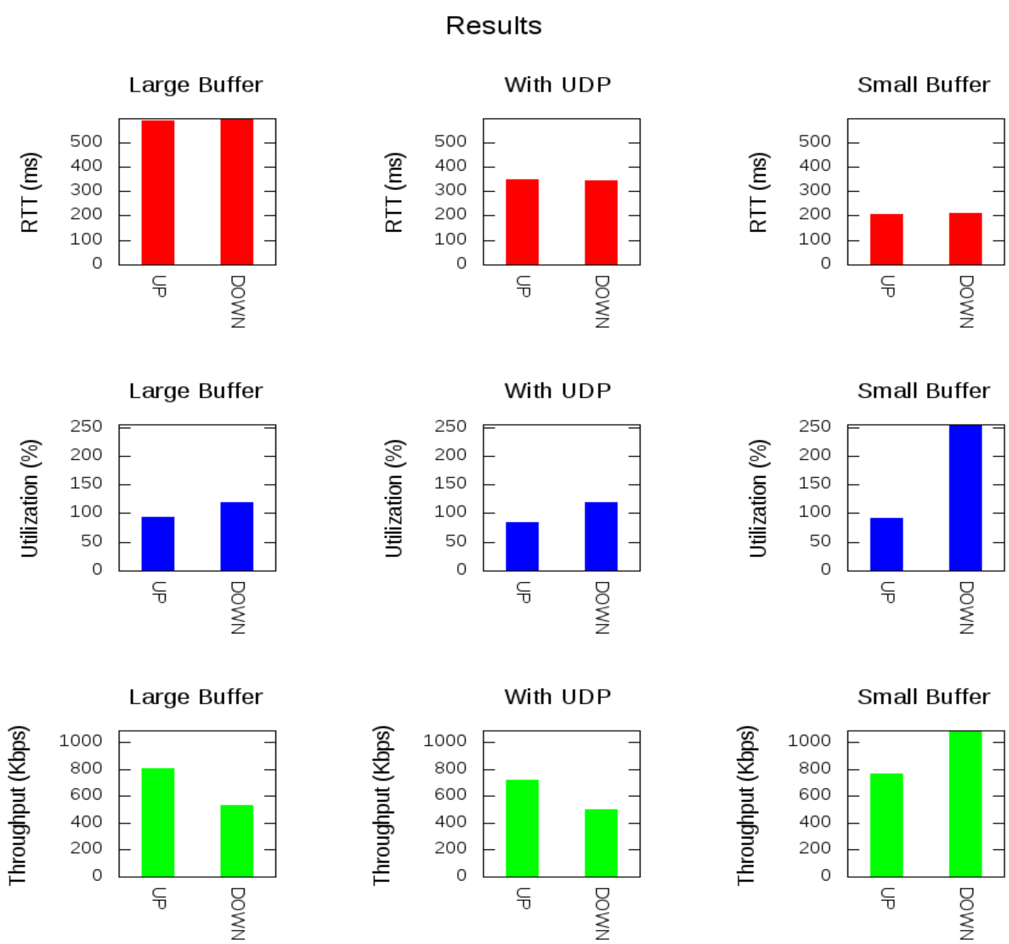
Results
Large Buffer
With UDP
Small Buffer
RTT (ms)
Utilization (%)
Throughput (Kbps)
UP
DOWN

Figure 5

We can compare RTT, Throughput and Utilization by Figure 4 and 5 when buffer of router is large and small also when adding UDP traffic. As you can see in top of Figure 4 when buffer is large RTT almost reaches 700 ms in download and upload link but when buffer is small, RTT is about 300 ms. This means large buffer causes long delay and decreasing the size of buffer improves this latency so our purpose is decrease this size of buffer without Dummynet in Testbed i.e. without changing in hardware in real networks. Figure 4 and 5 show that the RTT when we adding UDP traffic is almost equal with when we use small buffer.

Because this is an ADSL asymmetric network we always have less RTT in downlink than uplink.

References

[1] Braud, T., M. Heusse, and A. Duda. TCP over large buffers: When adding traffic improves latency.in Teletraffic Congress (ITC), 2014 26th International. 2014. IEEE.