ECEN602 Network Simulation Report

tiandi

April 2021

1 Simulation

1.1 Network Topology

In the experiment, a network topology is constructed as follows:

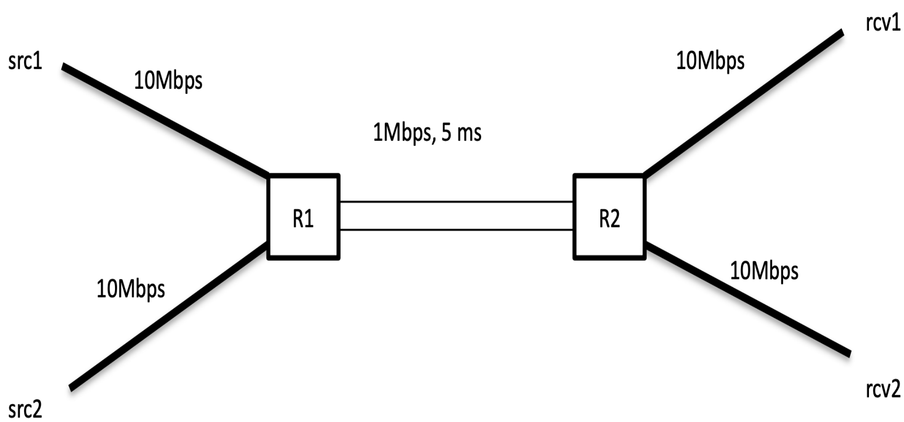

Figure 1: Network topology

In the topology graph, circles on the left represent two TCP senders, circles on the right represent two TCP receivers. The squares sit on the middle represent routers.

The TCP connection 1 is established between the top left node and the top right node, the TCP connection 2 is established between the down left node and the down right node. The application runs over TCP is FTP.

Basically, the topology represents a network where two TCP connections share a bottleneck physical link at low level.

1.2 Simulation Setup

Consider that the original 6 experiments might not provide enough information for observing any general trends, we extend 14 more experiments.

The first set of experiments applies TCP SACK and the second one applies TCP VEGAS. Both sets uses 10 RTT ration setup: 1:2, 1:3, 1:4, 1:5, 1:6, 1:7, 1:8, 1:9, 1:10 and 1:20.

Exp No	TCP Flavor	RTT Ratio (TCP1 to TCP2)
1-1	SACK	1:2
1-2	SACK	1:3
1-3	SACK	1:4
1-4	SACK	1:5
1-5	SACK	1:6
1-6	SACK	1:7
1-7	SACK	1:8
1-8	SACK	1:9
1-9	SACK	1:10
1-10	SACK	1:20
2-1	VEGAS	1:2
2-2	VEGAS	1:3
2-3	VEGAS	1:4
2-4	VEGAS	1:5
2-5	VEGAS	1:6
2-6	VEGAS	1:7
2-7	VEGAS	1:8
2-8	VEGAS	1:9
2-9	VEGAS	1:10
2-10	VEGAS	1:20

Table 1: Simulation setup

1.3 Results

This section shows part of the simulation results(the rest can be found in appendix).

In the figures, the red line represents TCP connection 1 and the green line represents TCP connection 2.

1.3.1 Network topology

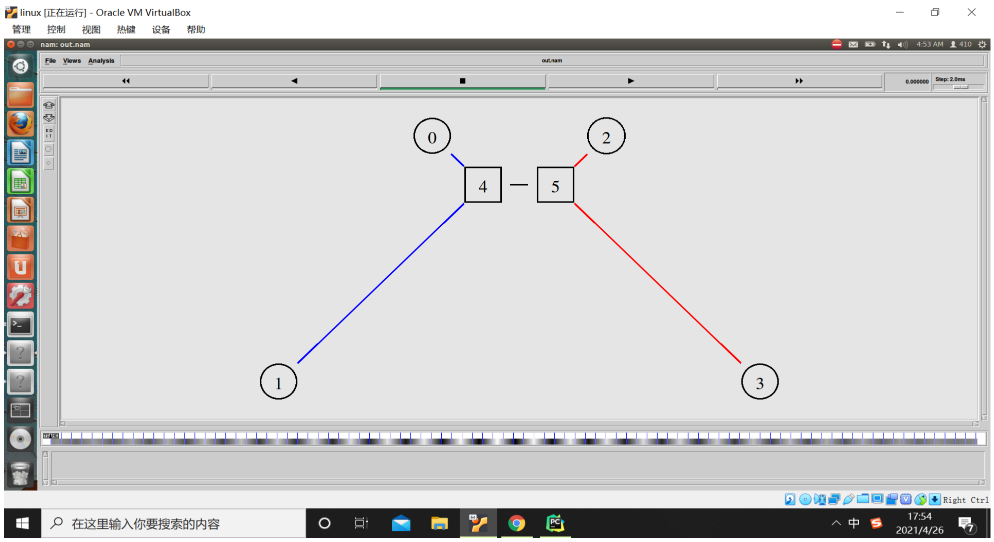

Figure 2: Network topology

1.3.2 Exp 1-1 (SACK + RTT Ratio 1:2)

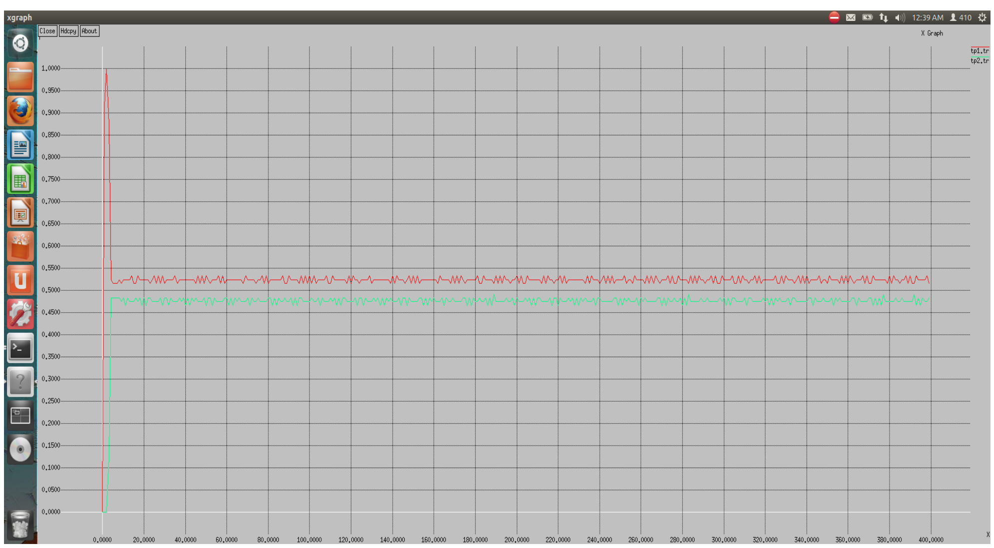

Figure 3: SACK, 1:2

1.3.3 Exp 2-1 (VEGAS + RTT Ratio 1:2)

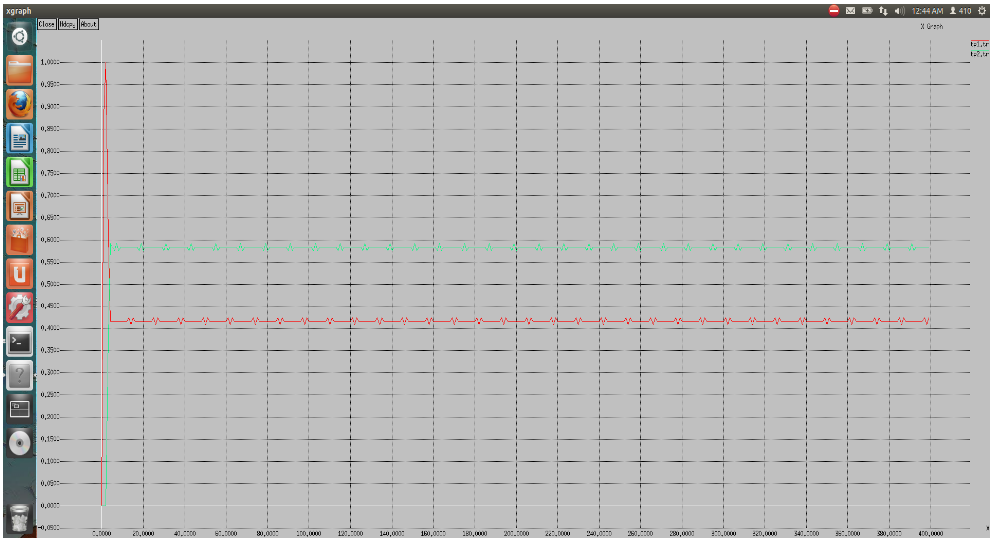

Figure 4: VEGAS, 1:2

2 Throughput Ratio(Question(i))

This section shows the throughput ratio of the two TCP connections. Section 2.1 shows the throughput ratio table of TCP SACK. Section 2.2 shows the throughput ratio table of TCP VEGAS. Section 2.3 plots the results in one figure.

2.1 Throughput Ratio for TCP SACK

Exp No	RTT Ratio (TCP1 to TCP2)	Throughput Ratio (TCP1 to TCP2)
1-1	1:2	1.105
1-2	1:3	1.222
1-3	1:4	1.273
1-4	1:5	1.443
1-5	1:6	1.547
1-6	1:7	1.667
1-7	1:8	1.849
1-8	1:9	1.990
1-9	1:10	2.077
1-10	1:20	3.545

Table 2: Throughput ratio, TCP SACK

2.2 Throughput Ratio for TCP VEGAS

Exp No	RTT Ratio (TCP1 to TCP2)	Throughput Ratio (TCP1 to TCP2)
2-1	1:2	0.724
2-2	1:3	0.818
2-3	1:4	1.105
2-4	1:5	1.101
2-5	1:6	1.439
2-6	1:7	1.004
2-7	1:8	0.835
2-8	1:9	1.011
2-9	1:10	1.174
2-10	1:20	3.012

Table 3: Throughput ratio, TCP VEGAS

2.3 TCP SACK vs TCP VEGAS

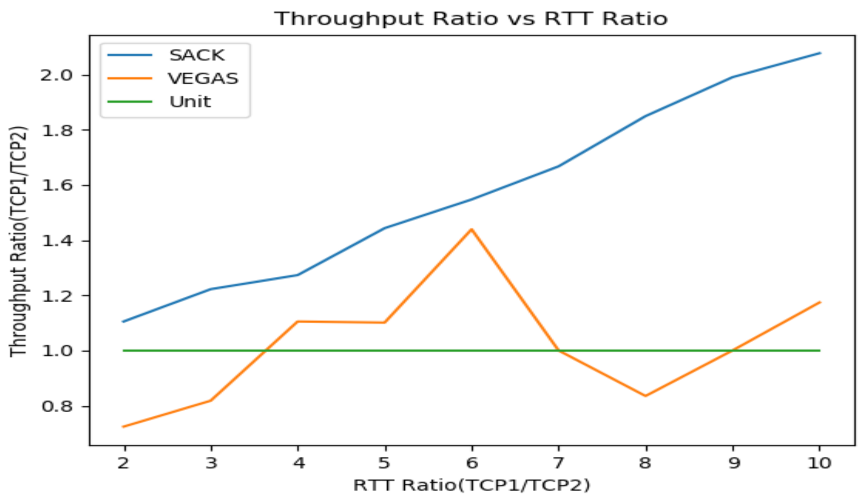

Figure 5: Throughput ratio, SACK vs VEGAS

3 Result Evaluation(Question(ii))

This section, we evaluate the results from three aspects: we first evaluate the performance of TCP SACK in section 3.1, then we evaluate TCP VEGAS in section 3.2, finally we compare the two TCP flavors.

3.1 TCP SACK

3.1.1 Bandwidth Utilization

The results show that under the given network topology in this report, TCP SACK takes full utilization of the bandwidth(see figures in appendix, the total throughput of the two TCP connection is equal to the bandwidth of the bottleneck link, which is 1Mbps).

3.1.2 Throughput Ratio

From the figure in section 2.3, we see that for TCP SACK, the throughput ratio is linearly positively correlated with RTT Ratio.

For two connections share a common bottleneck link, the one with larger latency get less bandwidth. This implies that TCP SACK is unfair to slow connections.

3.2 TCP VEGAS

3.2.1 Bandwidth Utilization

The results show that under the given network topology in this report, TCP VEGAS takes full utilization of the bandwidth(see figures in appendix, the total throughput of the two TCP connection is equal to the bandwidth of the bottleneck link, which is 1Mbps).

3.2.2 Throughput Ratio

From the figure in section 2.3, we can see that for TCP VEGAS, there's no obvious correlation between throughput ratio and RTT ratio when RTT ratio(TCP2/TCP1) is lower than 10. The throughput ratio fluctuate around 1. However, with the table from section 2.2, we see that when RTT ratio reaches 20, the throughput ratio of TCP VEGAS is 3.012.

The results imply that when the RTT of connections are not too far away from each other(within 10 times), each connection get approximately fair share of bandwidth.

3.3 VEGAS vs SACK

From the earlier discussion, we see that under the network topology in this report, the most significant difference in performance between VEGAS and SACK is bandwidth fairness. VEGAS treats connections with different latency more fair than SACK.

The possible cause is that they use different congestion control mechanism.

SACK estimates the network condition by monitoring the loss of packets. SACK increases its congestion window each time an ACK is received until there are packets lost, it then reduces the congestion window by half.

Imagine that the two TCP SACK connections are established simultaneously, they then increase their congestion windows and send more and more packets. When the routers' buffer is full, the packets from both connections tend to be discarded simultaneously, causing the two connections halving their congestion window. However, since the two connections have different latency, the fast connection always receives more ACKs and increases its congestion window faster, therefore ends up with more bandwidth than the slower one.

For connections that are not established at the same time, the conclusion still applies. Because when network is congested, all connections are expected to reduce their congestion window at the same time, but fast connections always recover faster, end up with more bandwidth.

VEGAS estimates the available bandwidth by observing the changes of RTT. A variable ExpectedRate is first computed by dividing congestion window by BaseRTT, which is usually the minimum of all measured RTT. Then the congestion window is adjusted according to the actual rate measured. This mechanism estimate the network condition more accurate than the one used by SACK.

For a given bandwidth, connections with higher RTT also have higher number of in-flight packets, i.e. packets that have not be ACKed, which leads to higher congestion window. Therefore, the ExpectedRate of connections in the experiments should be closed. Because all connections adjust their sending rate according to their ExpectedRate, which is close, they hence also end up with a close share of bandwidth.

4 Conclusion

In this network simulation assignment, we conducted two sets of experiments, one applies TCP SACK and another applies TCP VEGAS. Each set of experiments exploit ten different RTT ratio. Results show that SACK and VEGAS both take full utilization of the bandwidth. However, VEGAS is more fair than

SACK when connections have different latency. The possible reason is that VEGAS estimates the available bandwidth more accurate than SACK.

5 Appendix

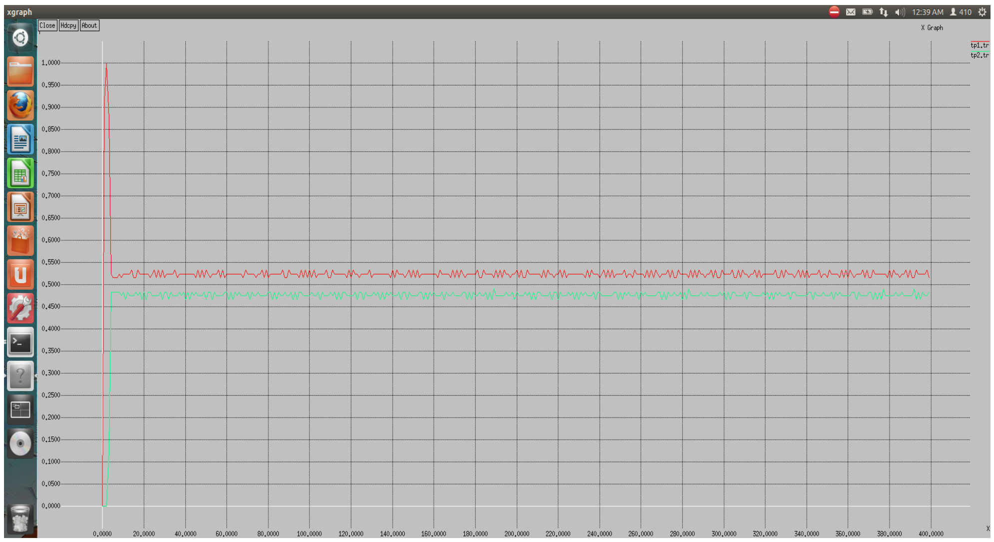

Figure 6: SACK, 1:2

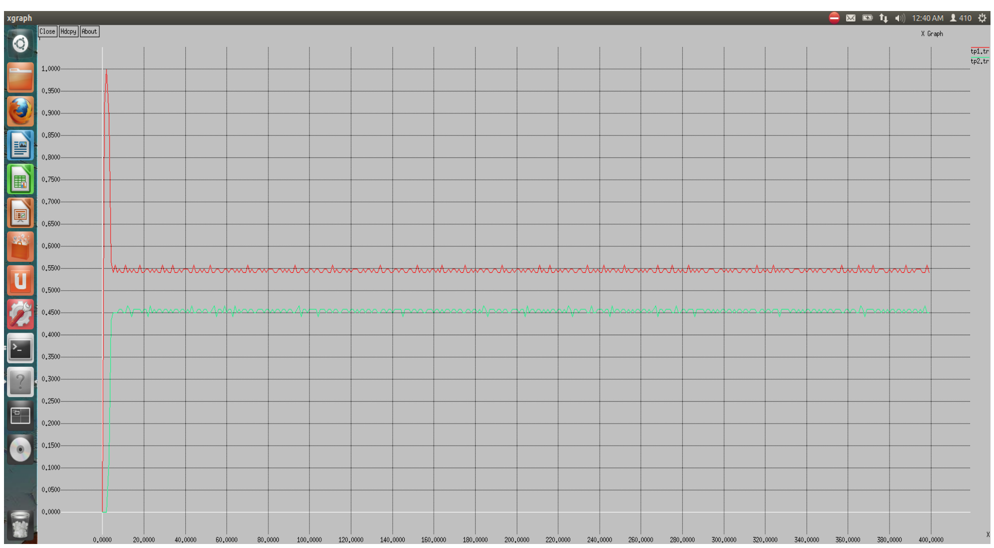

Figure 7: SACK, 1:3

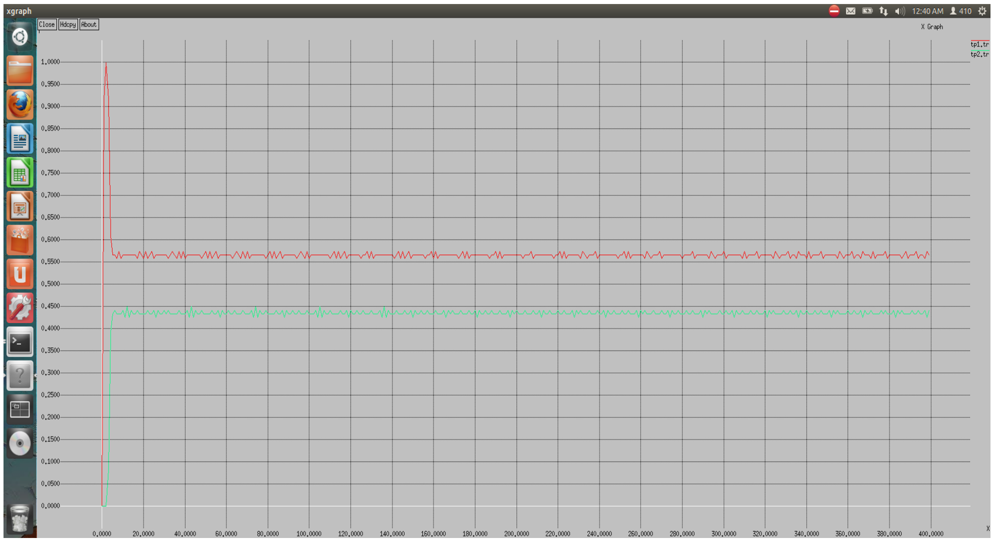

Figure 8: SACK, 1:4

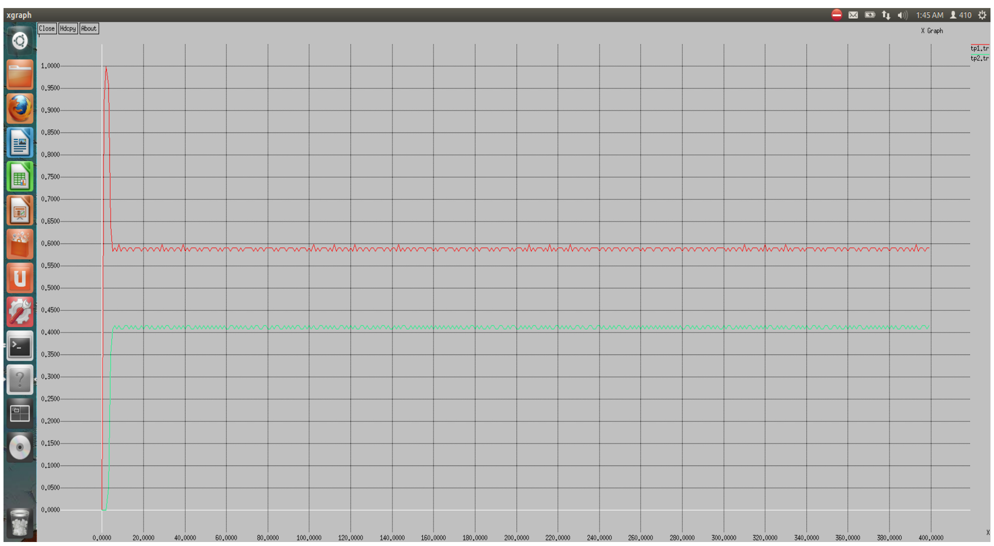

Figure 9: SACK, 1:5

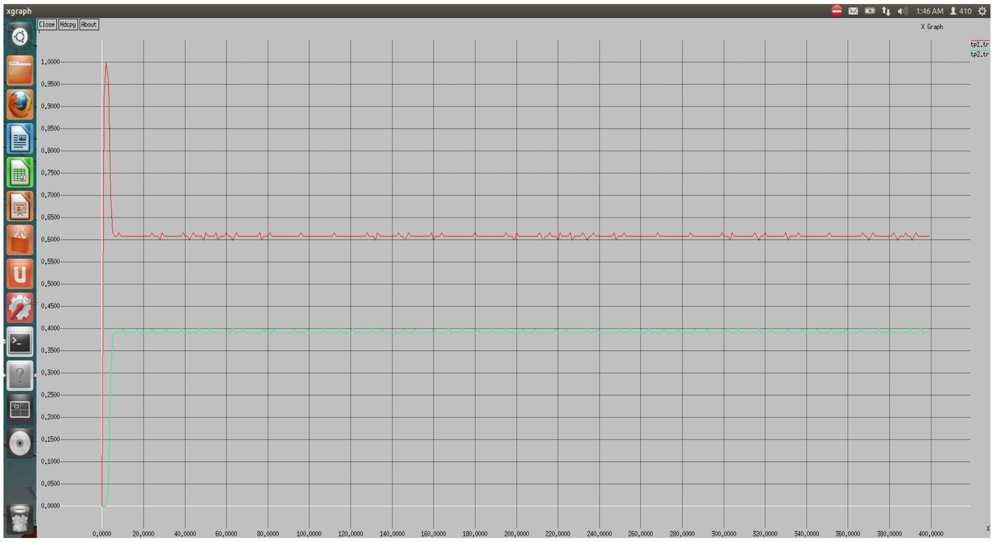

Figure 10: SACK, 1:6

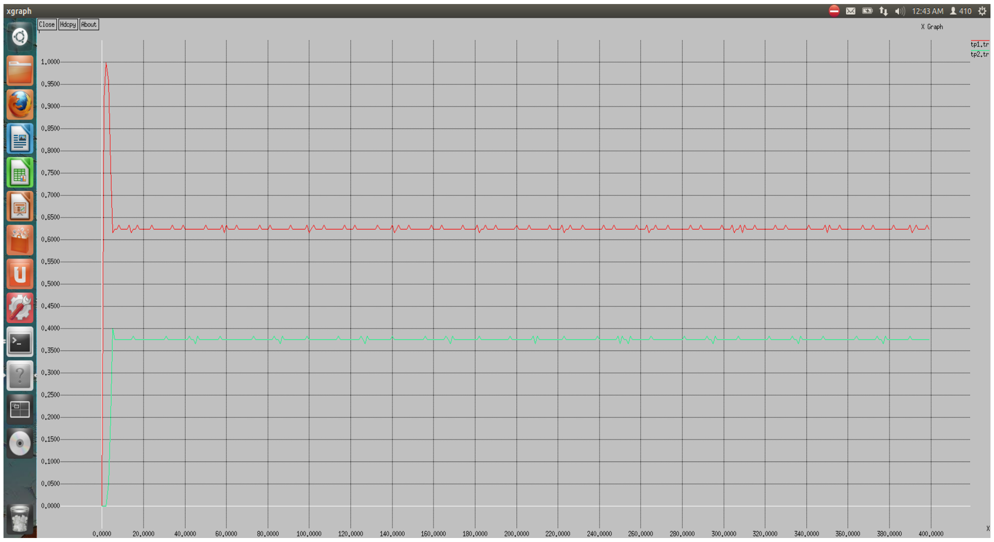

Figure 11: SACK, 1:7

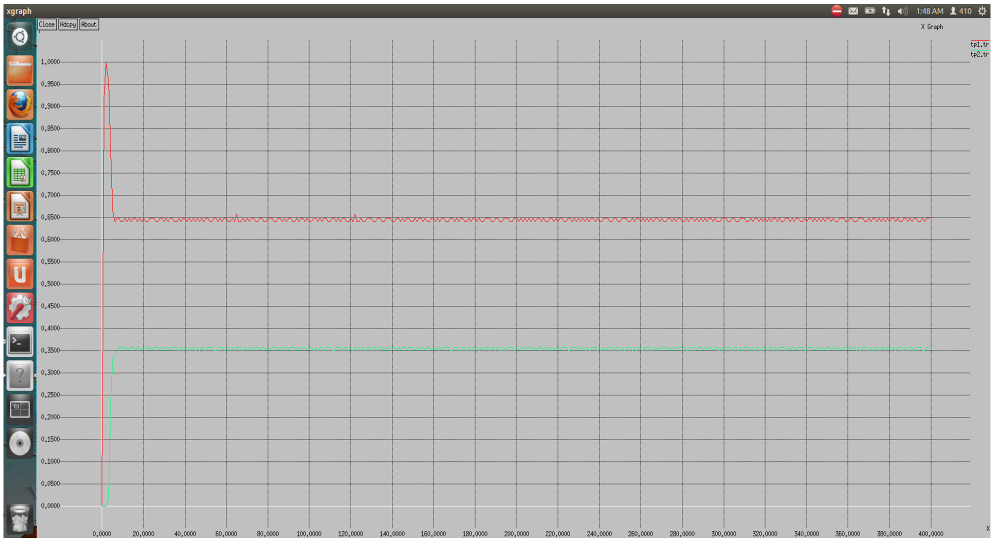

Figure 12: SACK, 1:8

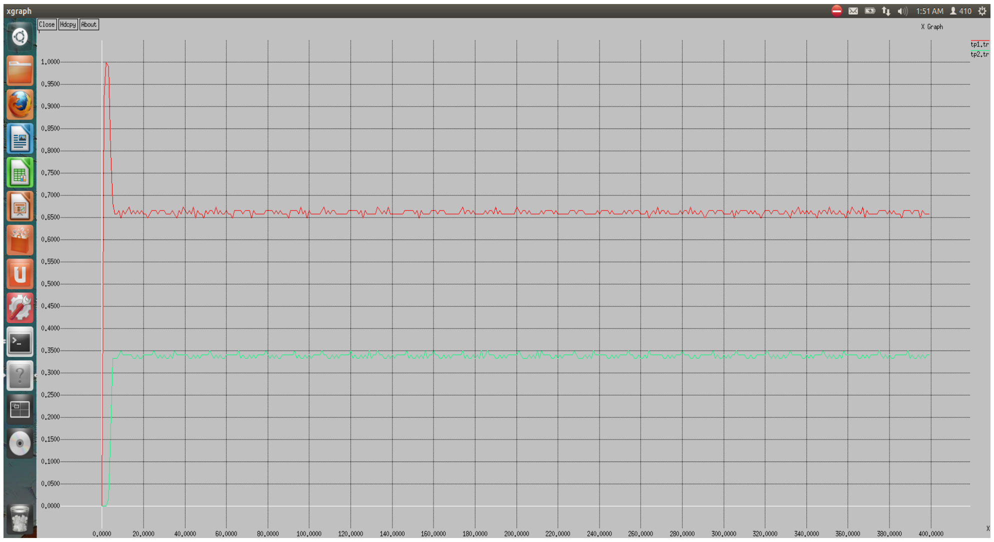

Figure 13: SACK, 1:9

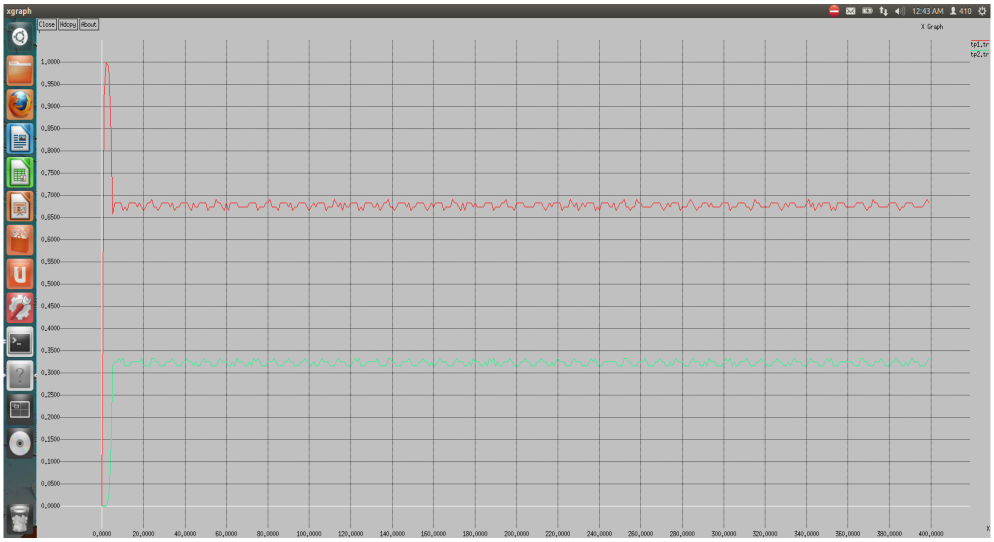

Figure 14: SACK, 1:10

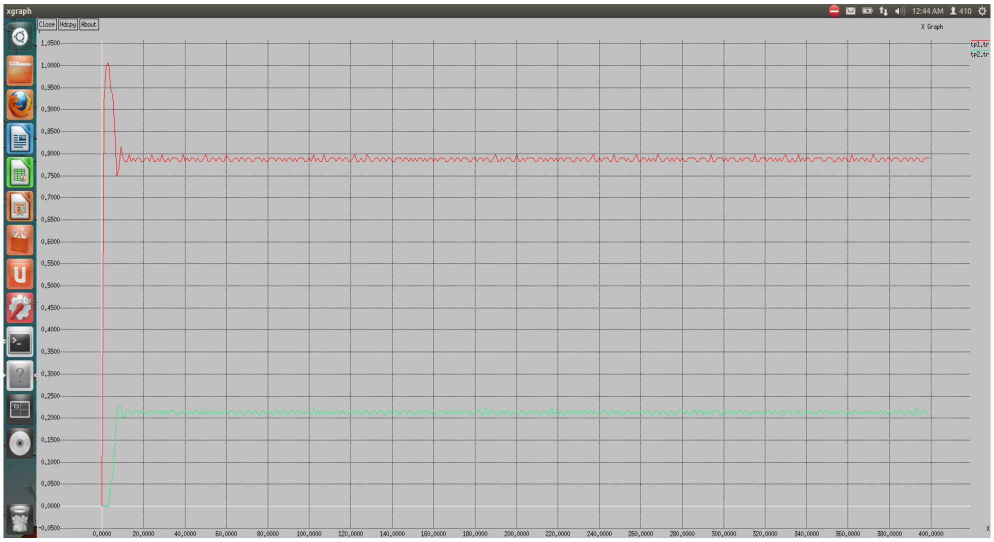

Figure 15: SACK, 1:20

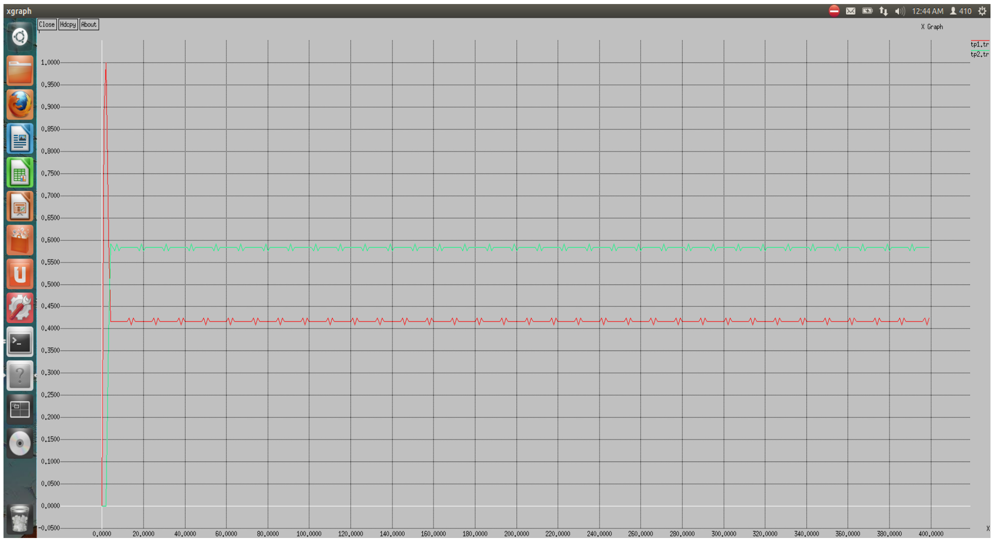

Figure 16: VEGAS, 1:2

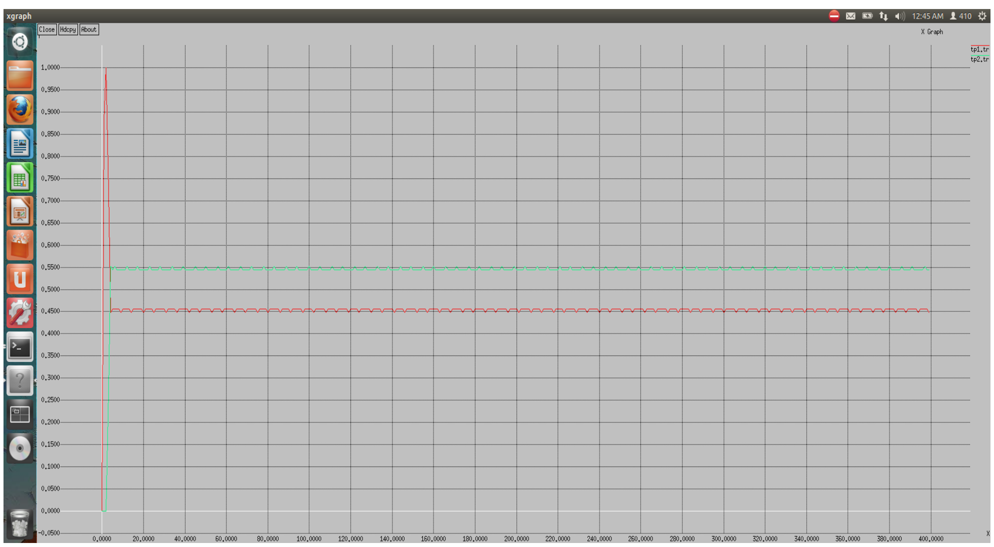

Figure 17: VEGAS, 1:3

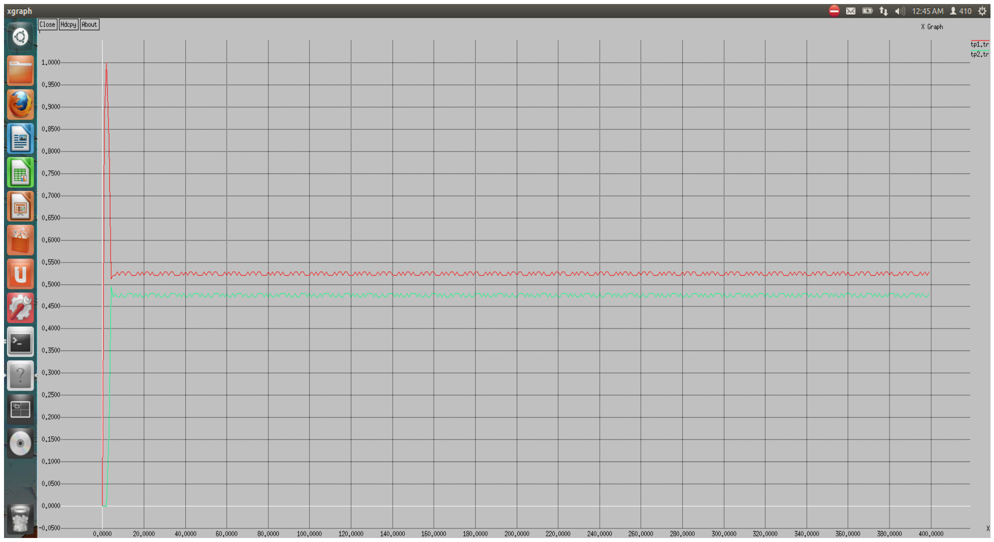

Figure 18: VEGAS, 1:4

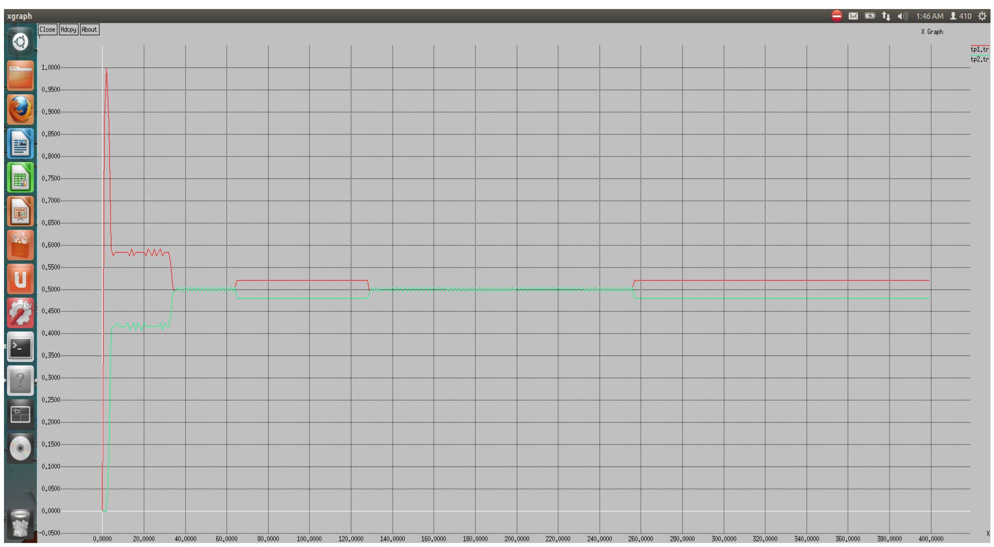

Figure 19: VEGAS, 1:5

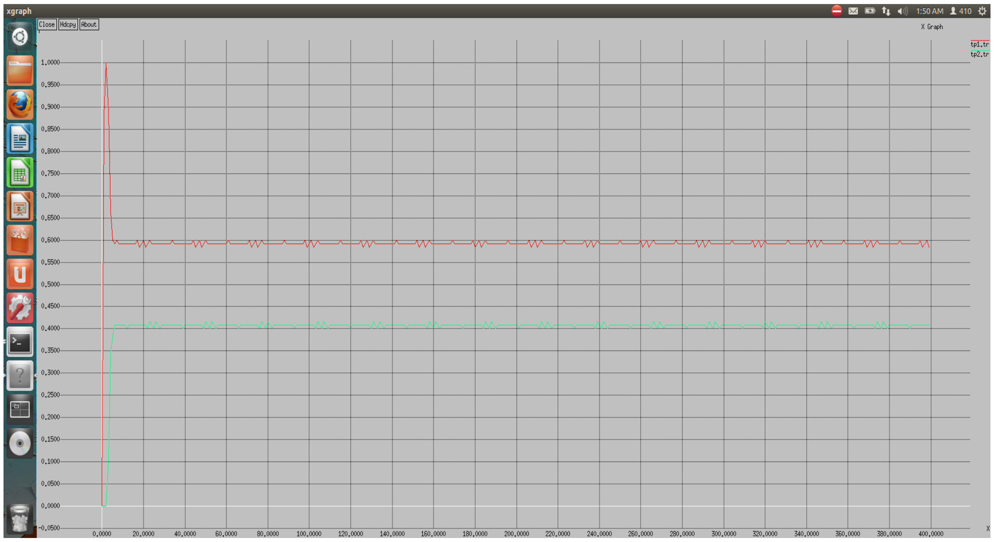

Figure 20: VEGAS, 1:6

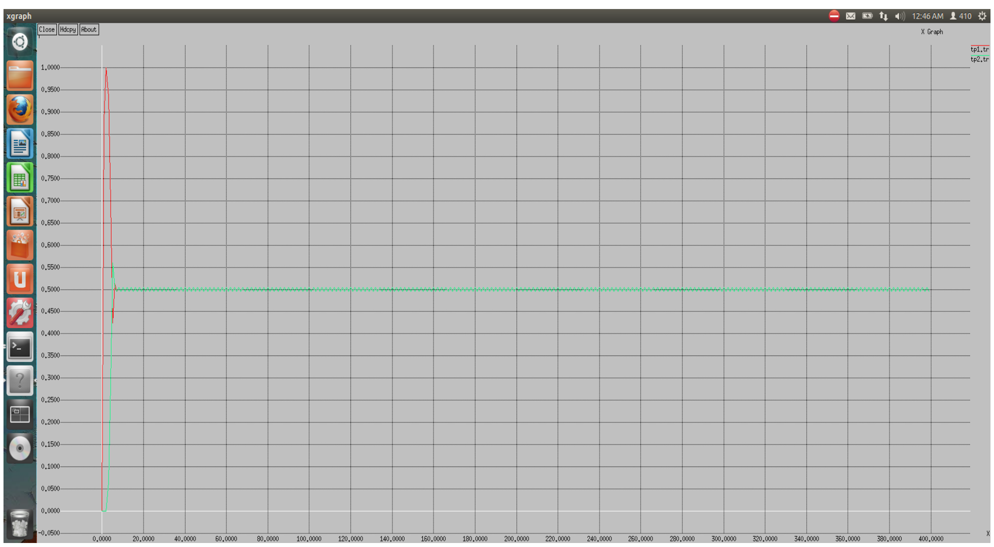

Figure 21: VEGAS, 1:7

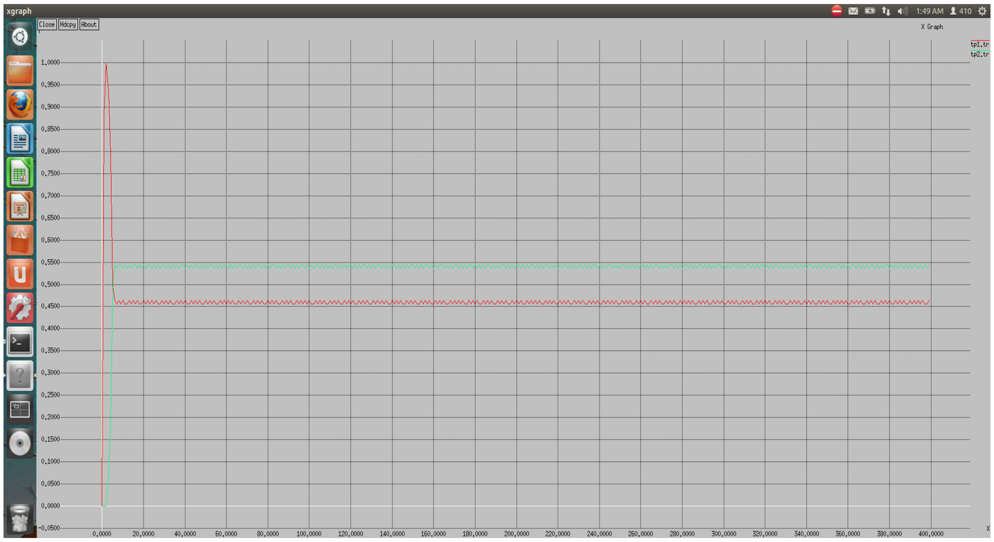

Figure 22: VEGAS, 1:8

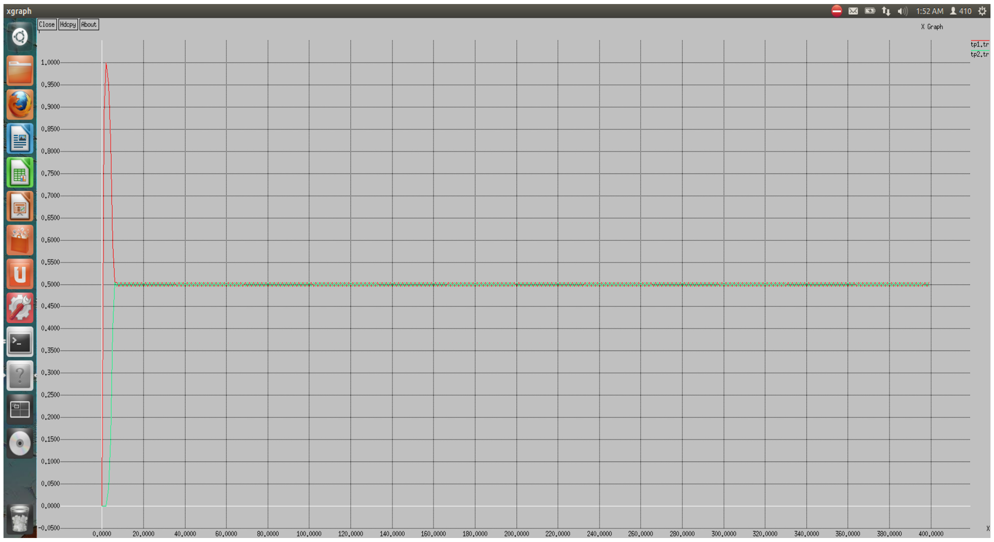

Figure 23: VEGAS, 1:9

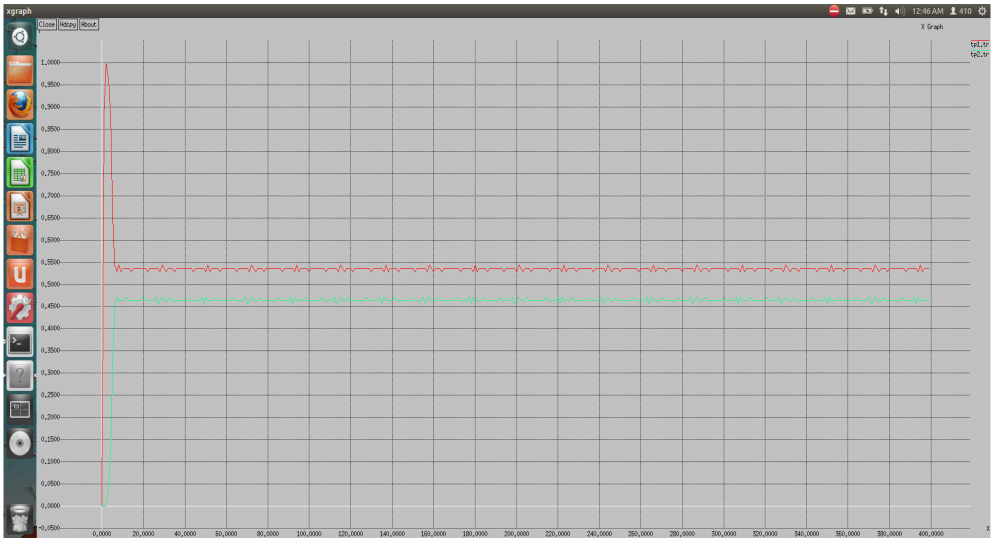

Figure 24: VEGAS, 1:10

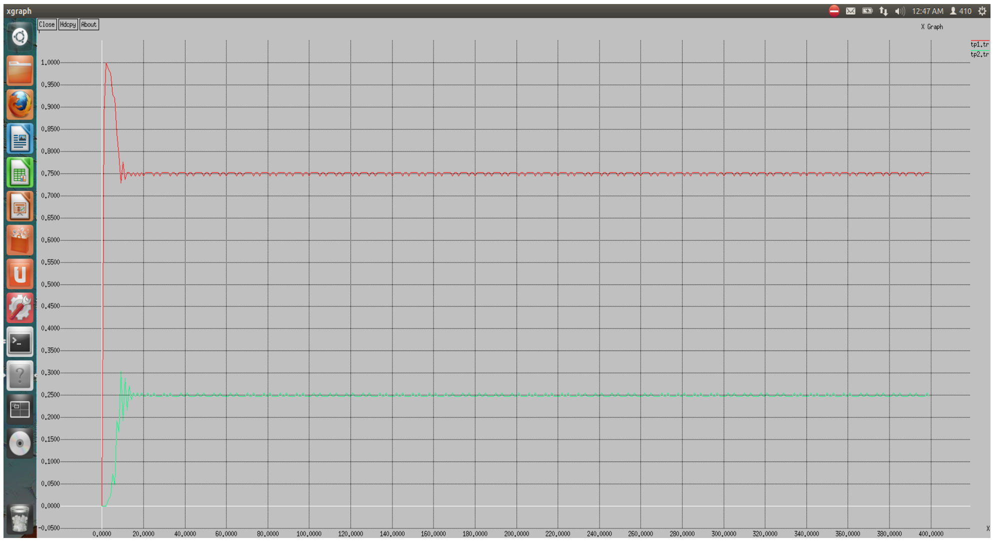

Figure 25: VEGAS, 1:20