Mice and Elephants

Ioannis Giannoualtos

Master in System and Network Engineering

UNIVERSITY OF AMSTERDAM

July 3, 2013

Why?

- Google showed high and stable utilisation of the links in their G-network.

However :

- Google has full control over the network
- In order to achieve that kind of utilisation the ability to create stable,limited bandwidth flows is needed.

Goals

- Configure the parameters of the Linux Kernel to achieve highest throughput.
- Create a constant flow of limited bandwidth using traffic control mechanisms
- Evaluate possible advantages to using traffic control in order to limit the throughput of flows

Definition

Definition

- *Elephant flow* is an extremely large (in total bytes) continuous flow set up by a TCP (or other protocol)flow over a network link.
- *Mice flow* is a flow that is short(in total bytes).

These flows took their name back in 2001 after noticing that a small amount of flows carried the majority of Internet traffic. Even though, the rest of the traffic consisted of large amount of flows,these carried very little Internet traffic

Research Questions

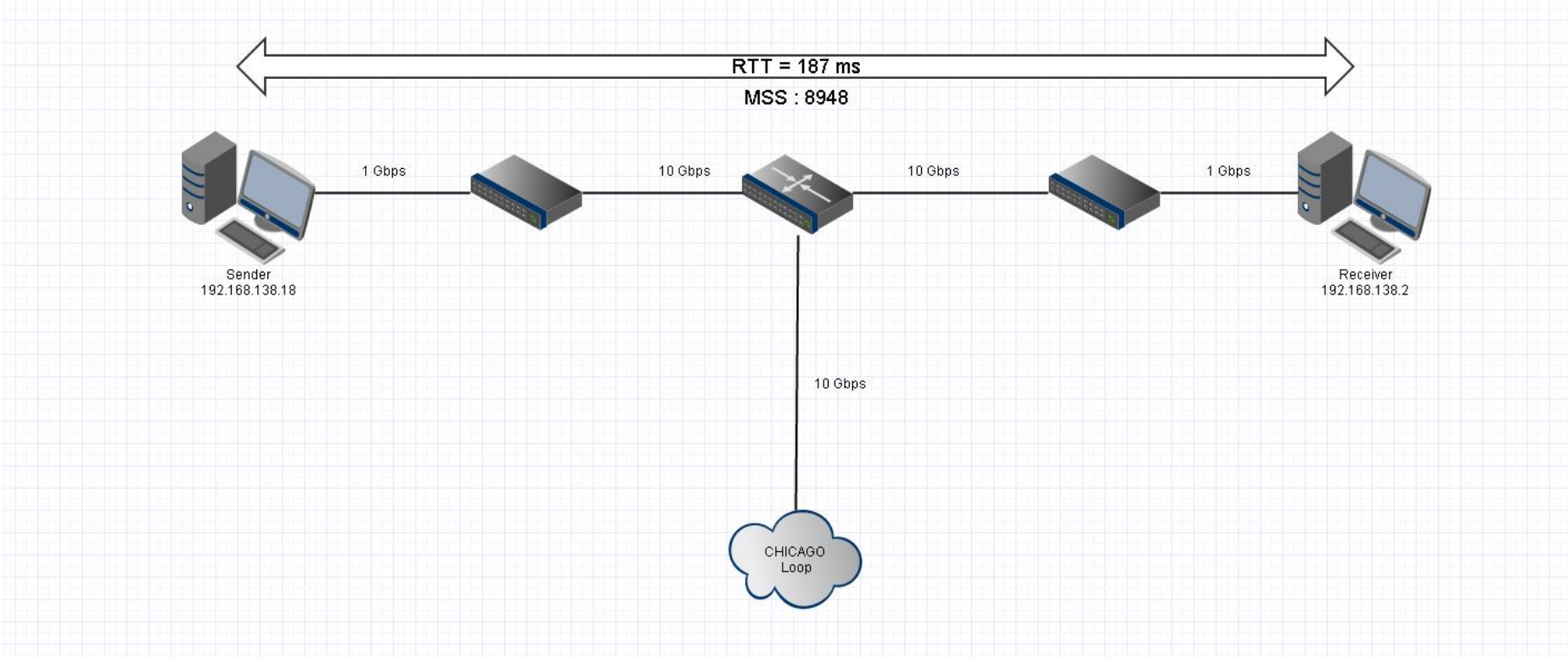
How can we achieve constant throughput and high utilisation of the link, while intermixing small and large TCP flows?

- What changes should be made in the configuration of the Linux TCP network stack to achieve the highest throughput?
- What effect does packet loss have on throughput?
- How can already existing traffic shaping techniques be used in order to provide a better throughput on the link with less packet loss?

Restrictions

- Knowledge of the kind of flows that go through the network.
- Traffic Control tools that already exist in the Linux kernel
- Focusing on Long Fat Networks (LFN)

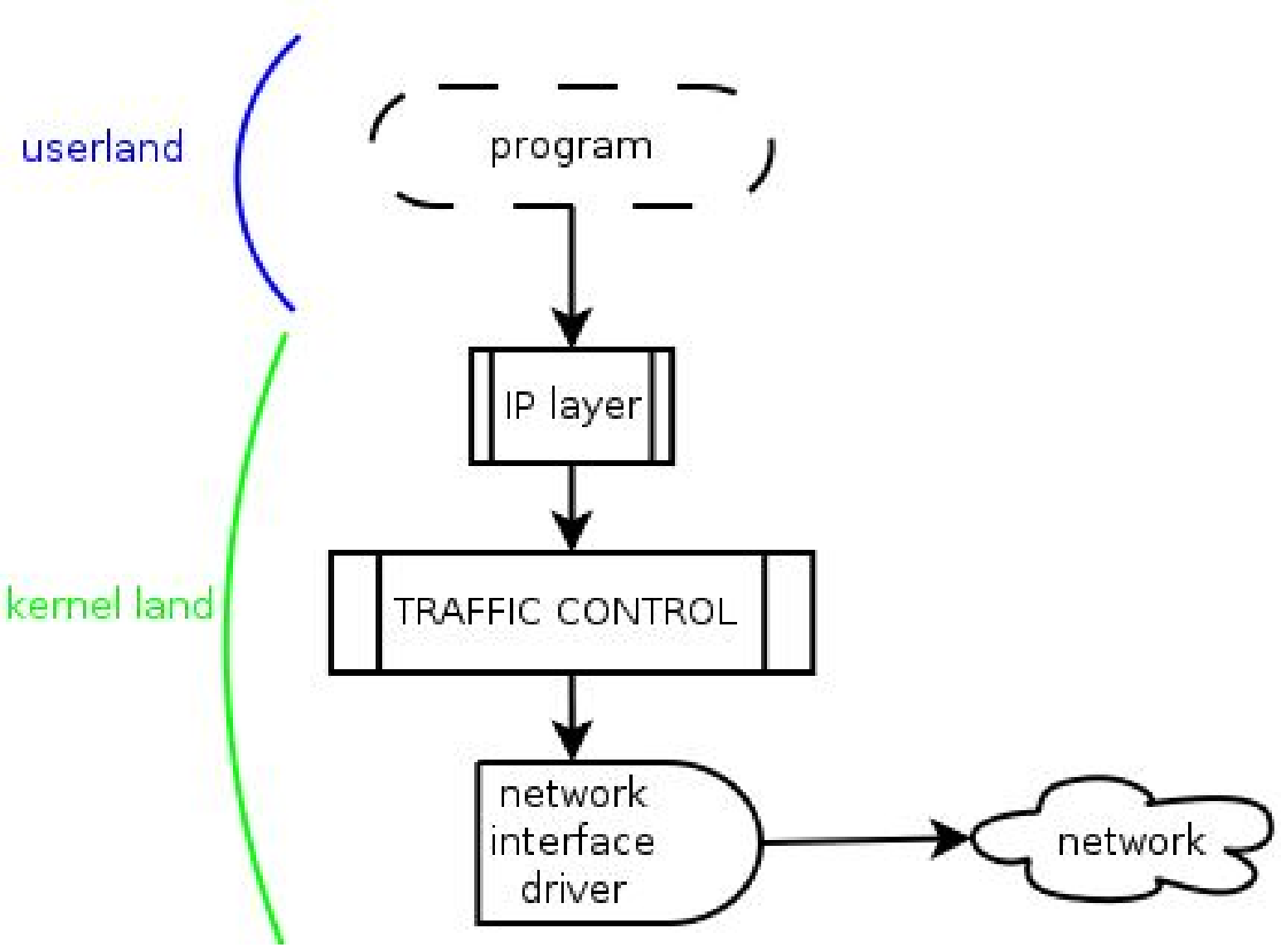
Testbed


RTT = 187 ms
MSS : 8948
1 Gbps
10 Gbps
10 Gbps
1 Gbps
Sender
192.168.138.18
Receiver
192.168.138.2
10 Gbps
CHICAGO
Loop

Tools Used

- Iperf
- Wireshark
- Traffic Control(TC)
- Tcp probe

Linux Traffic Control(TC)



Linux Traffic Control(TC)

- Already enabled in the kernel by default
- Queueing disciplines, Classful-Classless
- Classes in order to simulate sublinks
- Filters to distinguish traffic and assign it to classes

Why HTB?

- Best documented among the classfull disciplines
- More understandable and intuitive

TCP probe

- Kernel module that records the state of a TCP connection
- One line for each packet captured
- Captures Congestion window, Slow start threshold, Sequence numbers and many more

Theoretical background

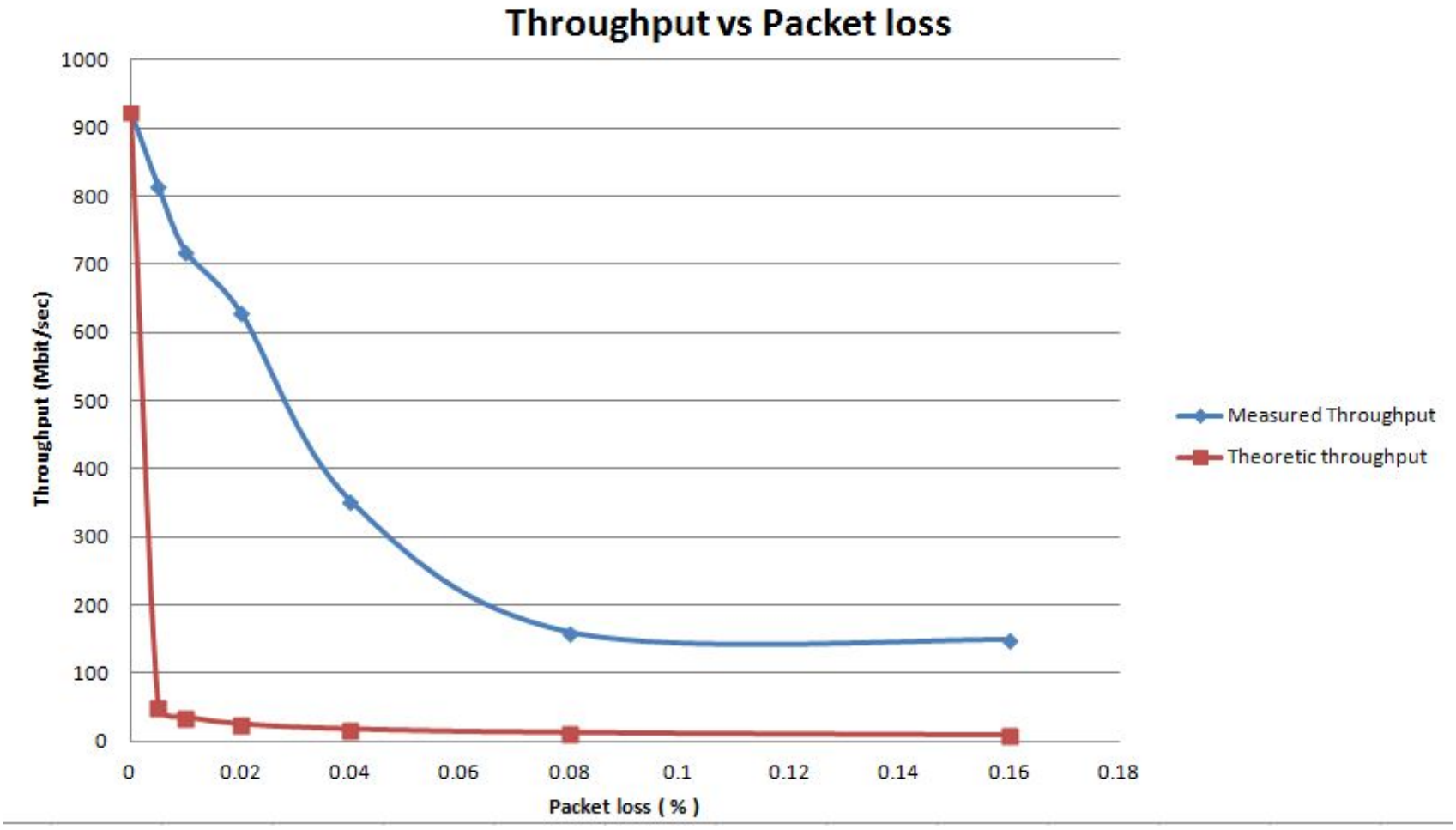
Mathis et. al. formula[1]:

$$Rate <= (MSS/RTT) * (1/\sqrt{p}) \quad (1)$$

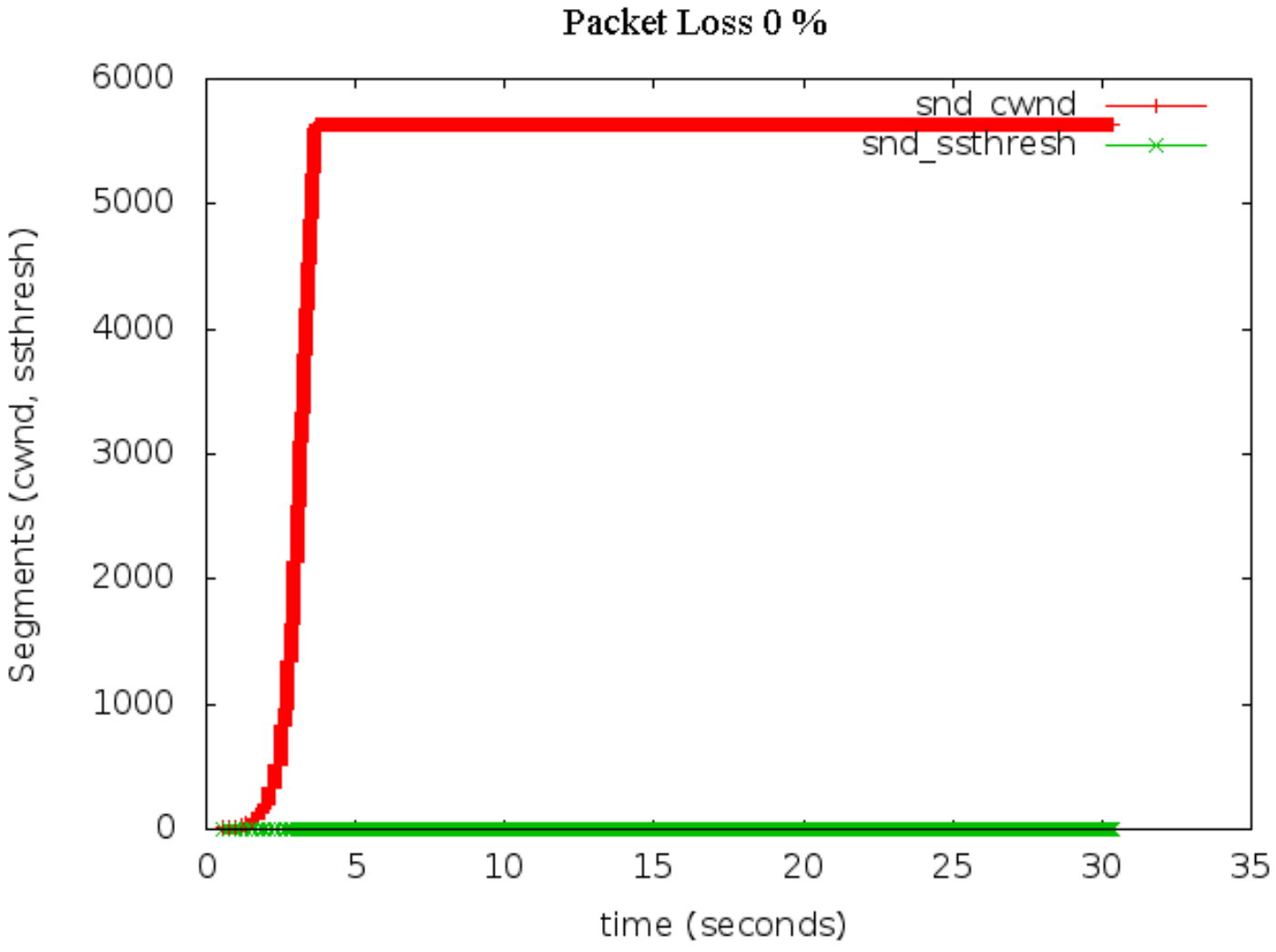
- MSS: Maximum Segment Size
- RTT: Round Trip Time
- p: packet loss

[1] The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm (1997)

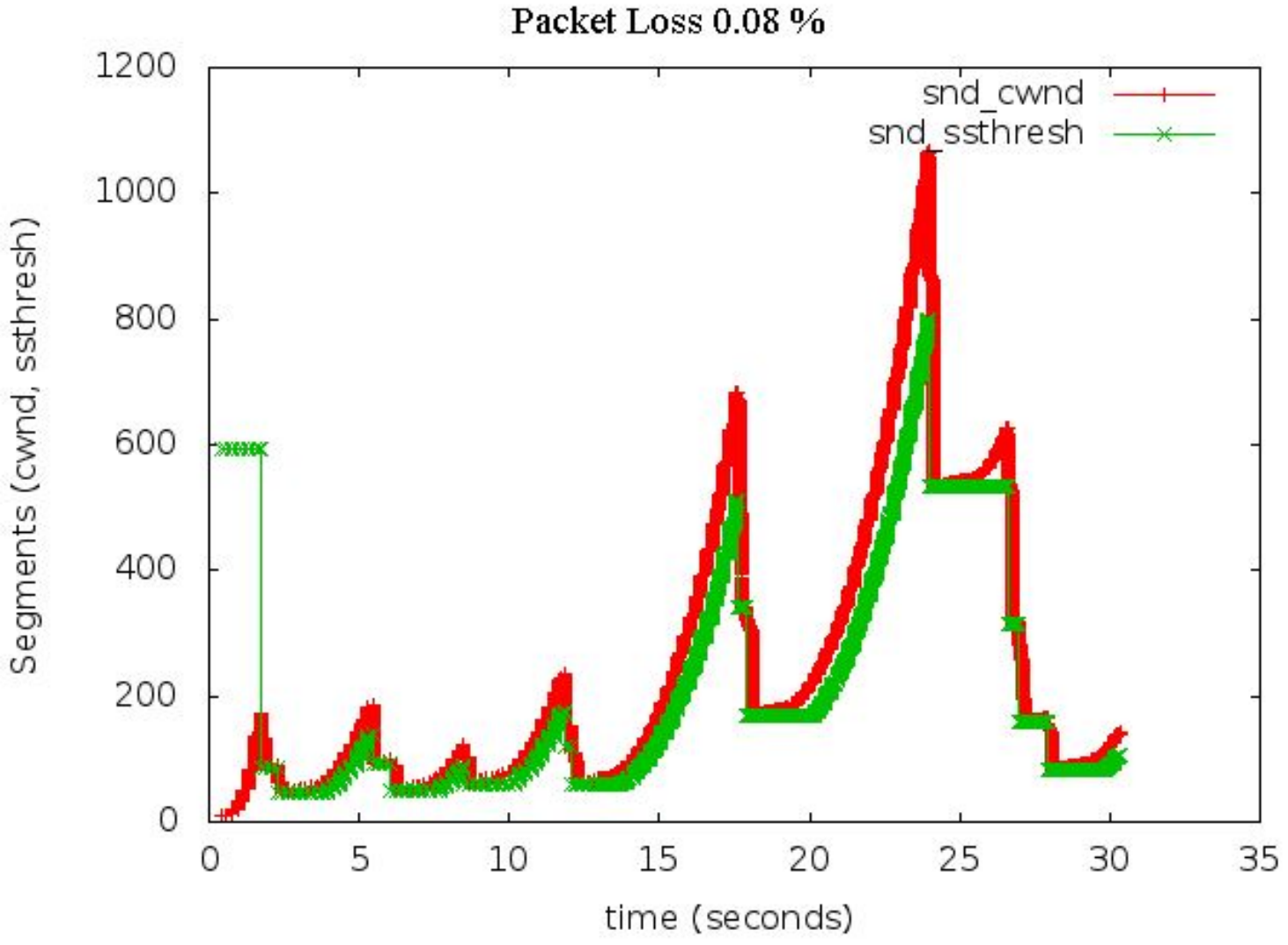
Packet Loss effect measured



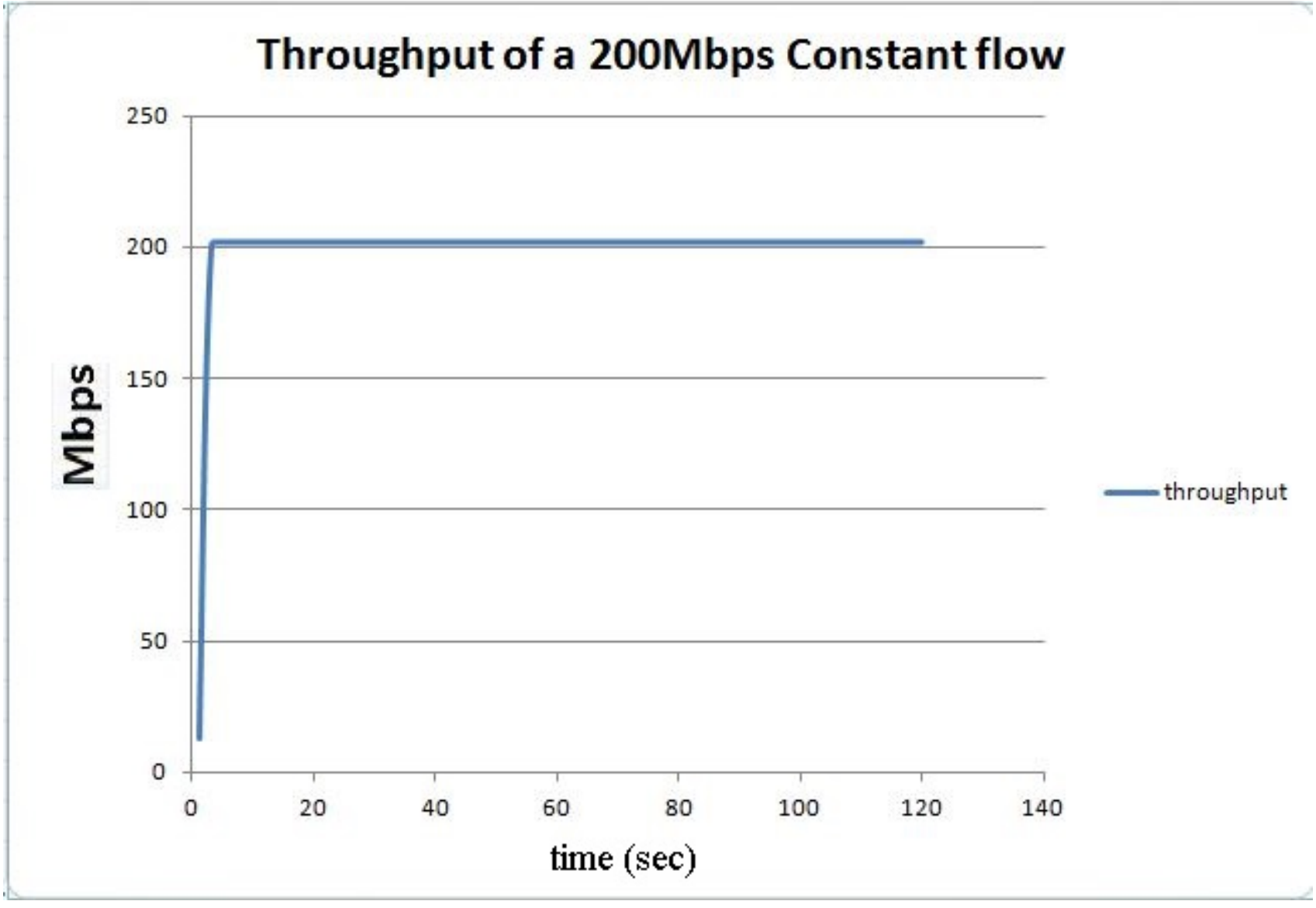
Packet Loss effect measured cont.



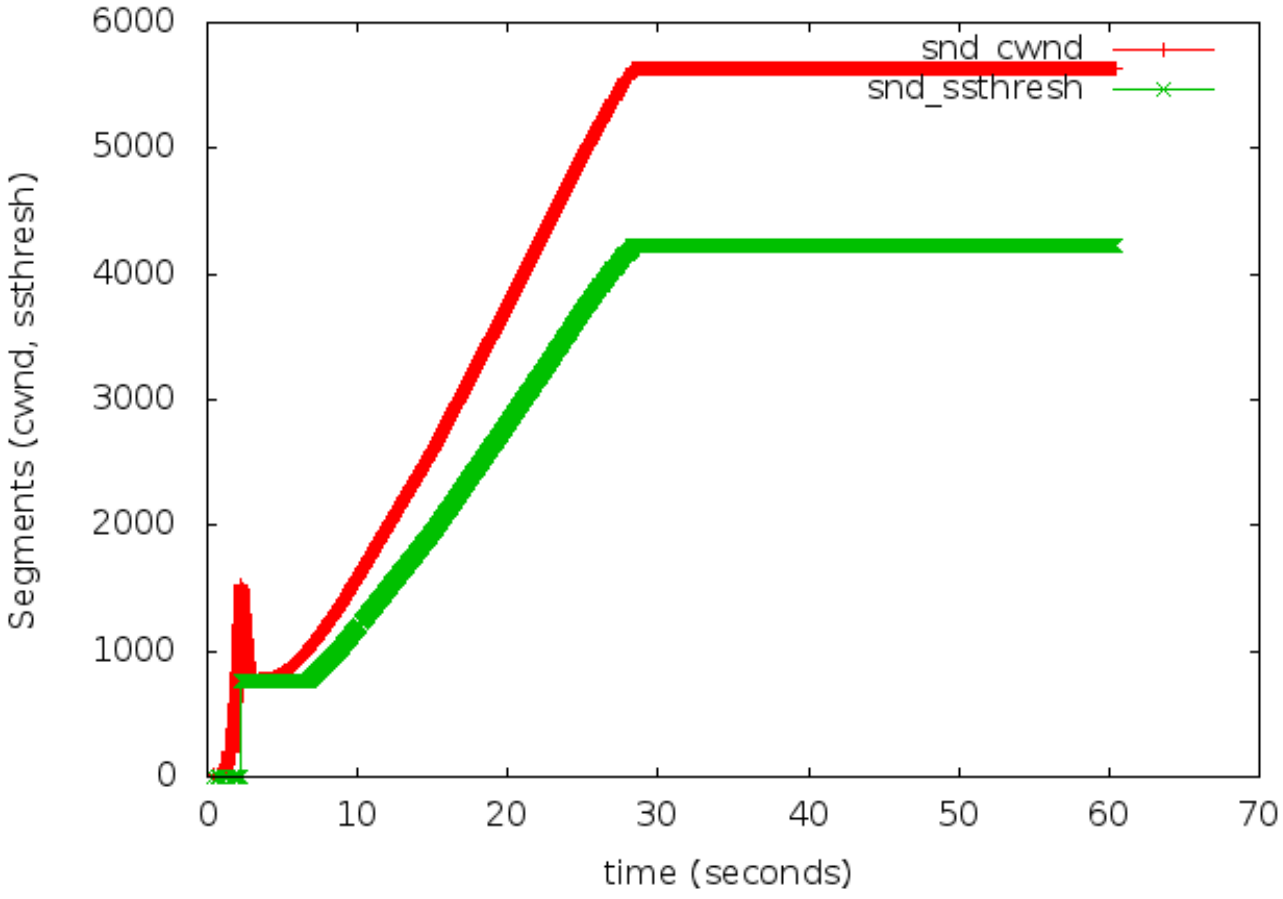
Packet Loss effect measured cont.



Creating a constant TCP Stream cont.



Creating a constant TCP Stream cont.

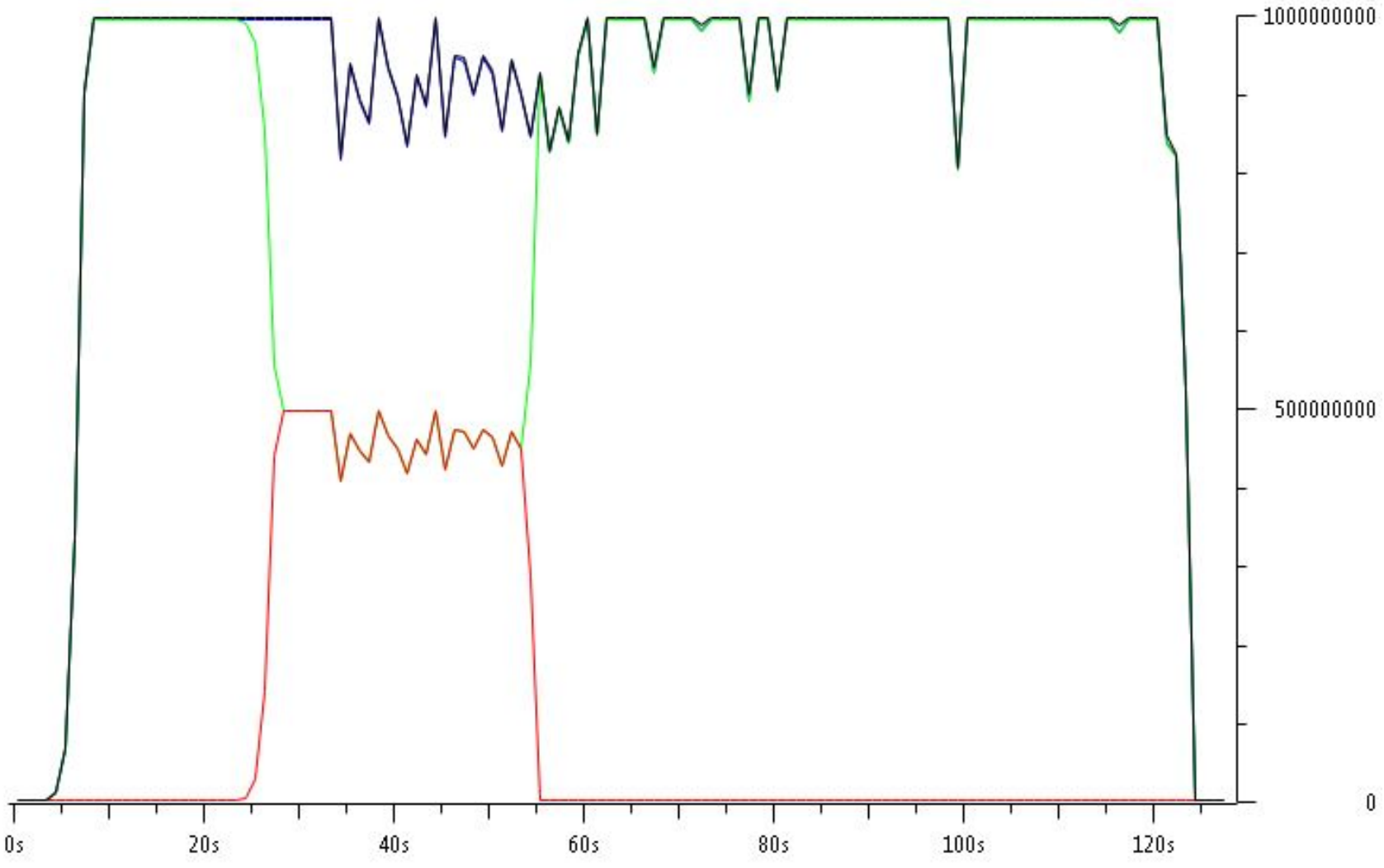


2 Flow Experiments

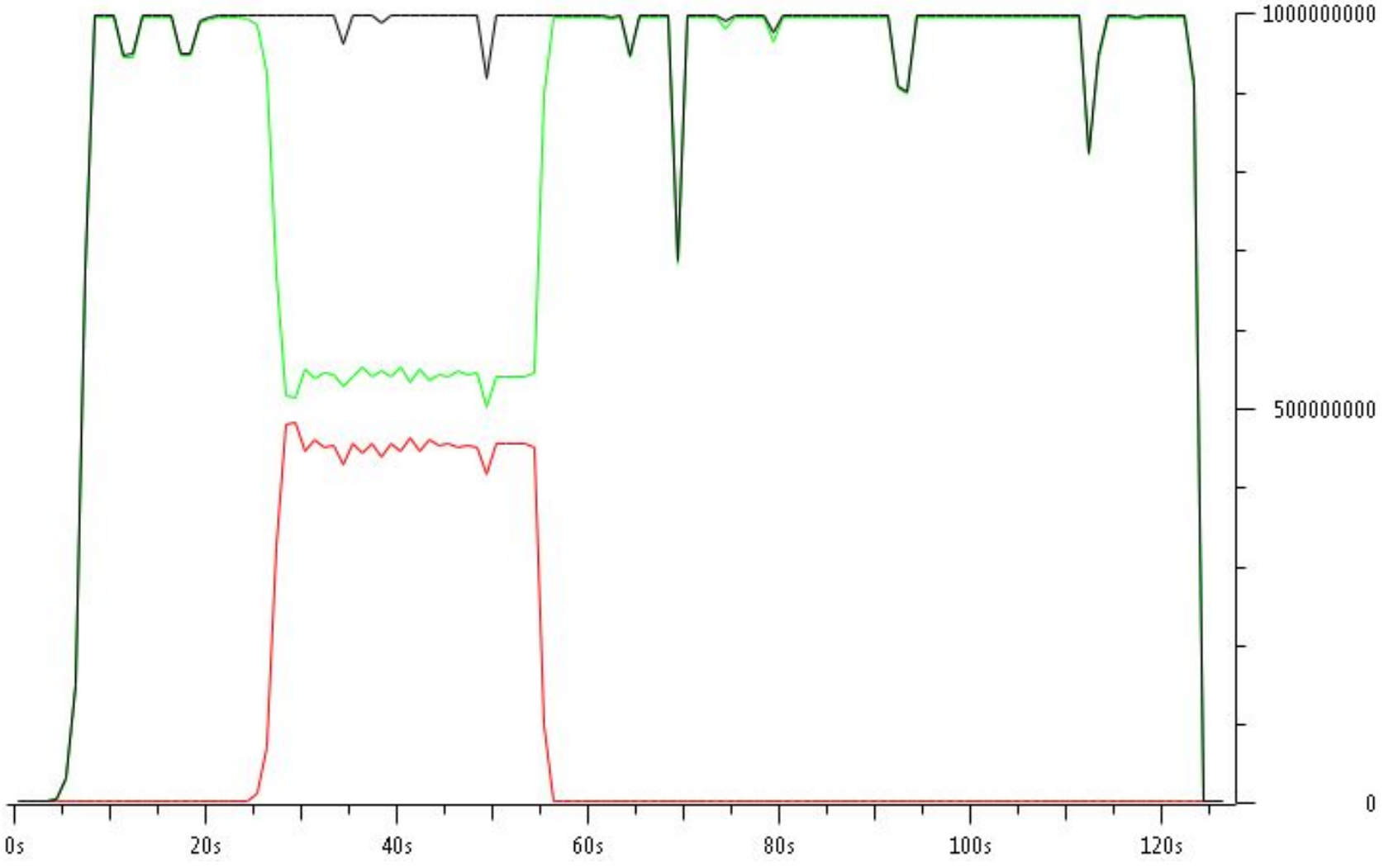
- Full link,no traffic control, 2 Flows competing for the bandwidth
- Full link,divided in half,with priorities borrowing
- Link limited to 400Mbps,no further traffic control,2 flows competing.
- Link limited to 400Mbps,divided in half,with priorities borrowing

Many more experiments were conducted with different bandwidth allocations and can be reviewed in the report.

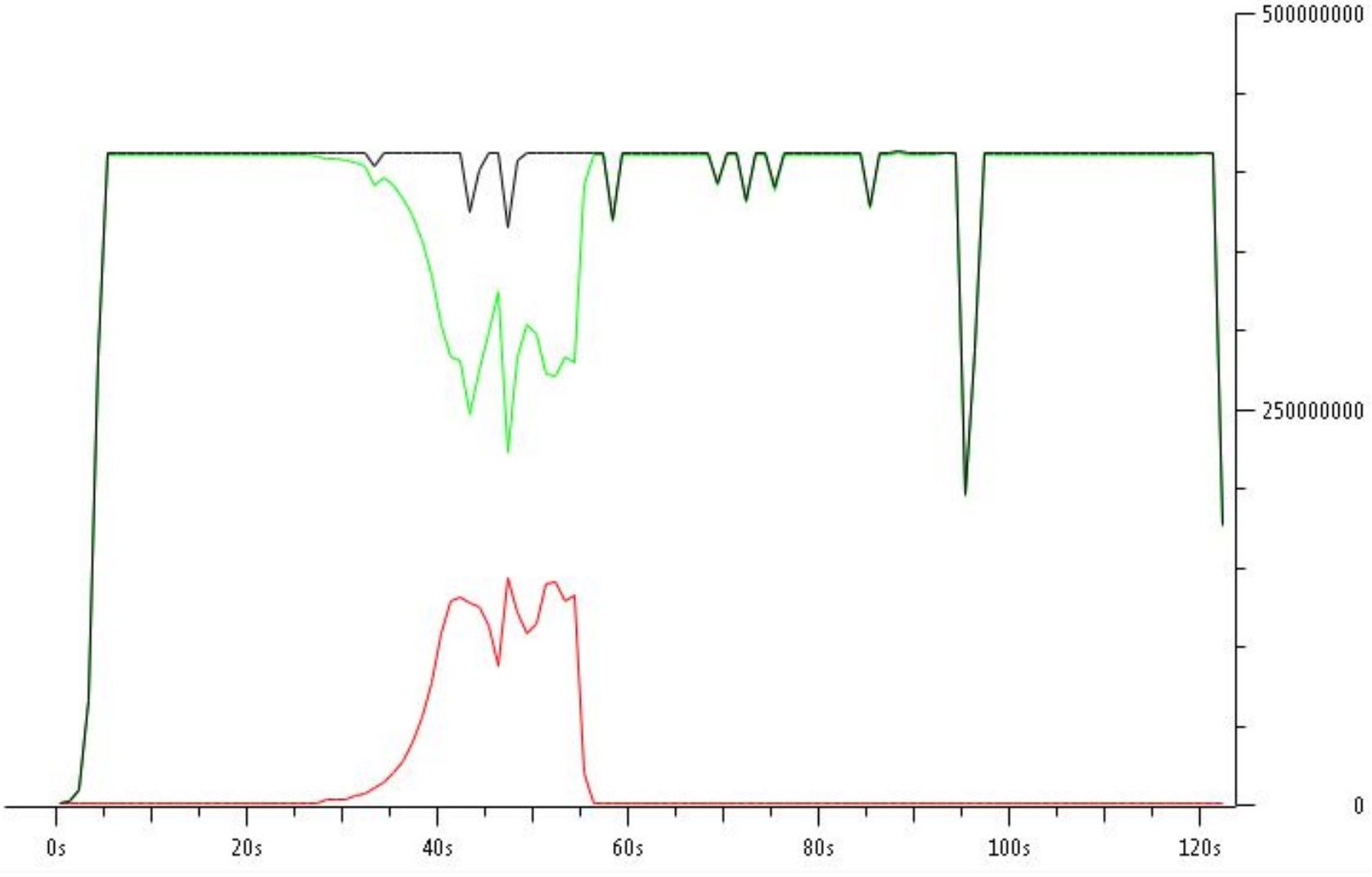
Full link,no traffic control, 2 Flows competing for the bandwidth-Throughput(Bits) vs Time(sec)



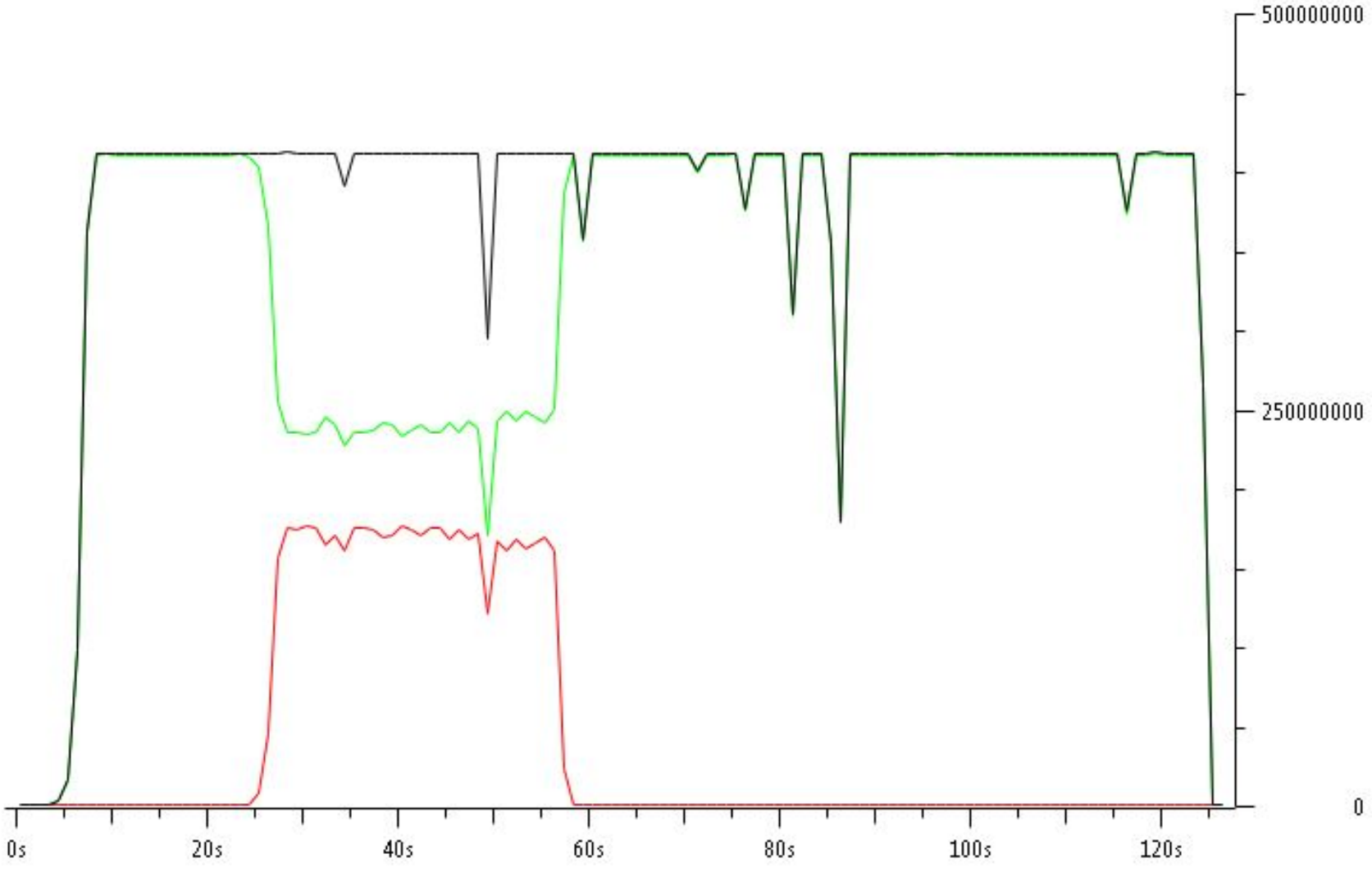
Full link,divided in half,with priorities borrowing-Throughput(Bits) vs Time(sec)



Link limited to 400Mbps,no further traffic control,2 flows competing- Throughput(Bits) vs Time(sec)



Link limited to 400Mbps,divided in half,with priorities borrowing- Throughput(Bits) vs Time(sec)



Conclusions

- Better utilisation of the link when full link is used
- There is a small deviation on the bandwidth allocation when using tc.
- The throughput is more stable using traffic control
- Less throughput reduction due to packet loss in contrast to theory

Future Research

- Create an Openflow monitored testbed and create the forwarding rules needed in order utilise the links fully.
- Changing the TCP implementation instead of only altering parameters in order for it to cooperate better with the Traffic control policies or even avoid them altogether.

Questions?