Computer Networks & Network Simulation

Prof: Danny Hughes

Tom Stappaerts - Bachelor Informatica
Jeroen Sanders - Bachelor Informatica

April 8, 2014
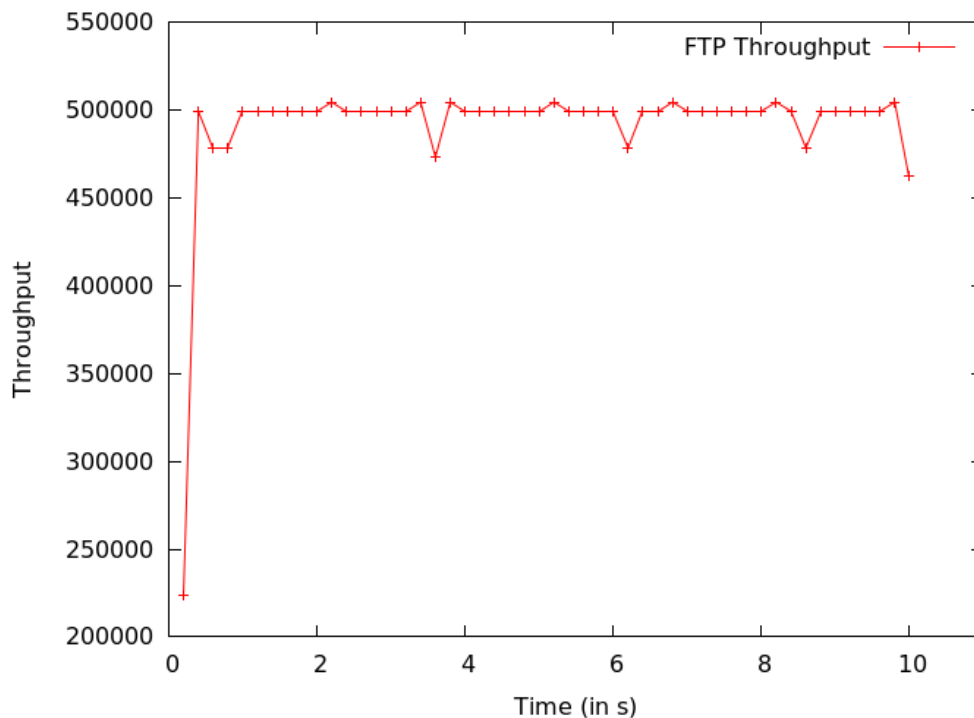
1 Exercise 1

1.1 Question 1

Figure 1: Throughput: bandwidth limitation and download connection

In figure 1 we plotted the throughput of the FTP application. We see that the connection stabilises on 4 Mbps. The download is limited by the strained connection between node 3 and 4.

1.2 Question 2

The FTP application uses all the available bandwidth. As soon as the CBR pplication starts it clogs up the network. It's packages are bigger than the tcp acknowledgement packages and it takes quite a while to upload them through the constrained upload link. Therefore we see a big drop in the graph, because the delivery of the FTP ACK-packets is delayed. In figure 1 we can see a sawtooth behaviour for the FTP throughput in the drop. Between the CBR packets, some ACK's of the FTP appliction get through the strained upload connection.

1.3 Question 3

A reserved amount of upload bandwidth for each application would prevent cannibalisation of the connection. A certain throughput would be guaranteed for both applications. Initially the FTP application would behave just like in question 1. When the CBR application starts, it's bandwidth will decrease because the CBR application has it's own guaranteed bandwidth.

1.4 Question 4

If the bandwidth is even more constrained, the saturation of the network is even higher as you can see in figure 3. It's worse because it takes longer for the ACK's to arrive. The throughput of the FTP application crashes to zero.
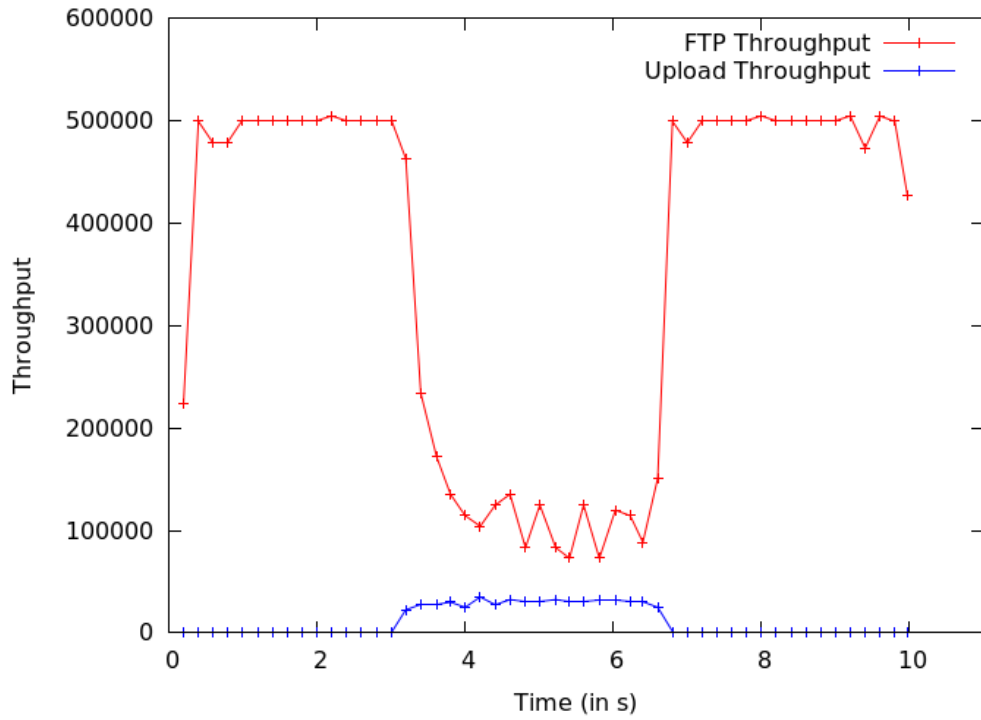
Figure 2: Throughput: bandwidth limitation, download and upload connection
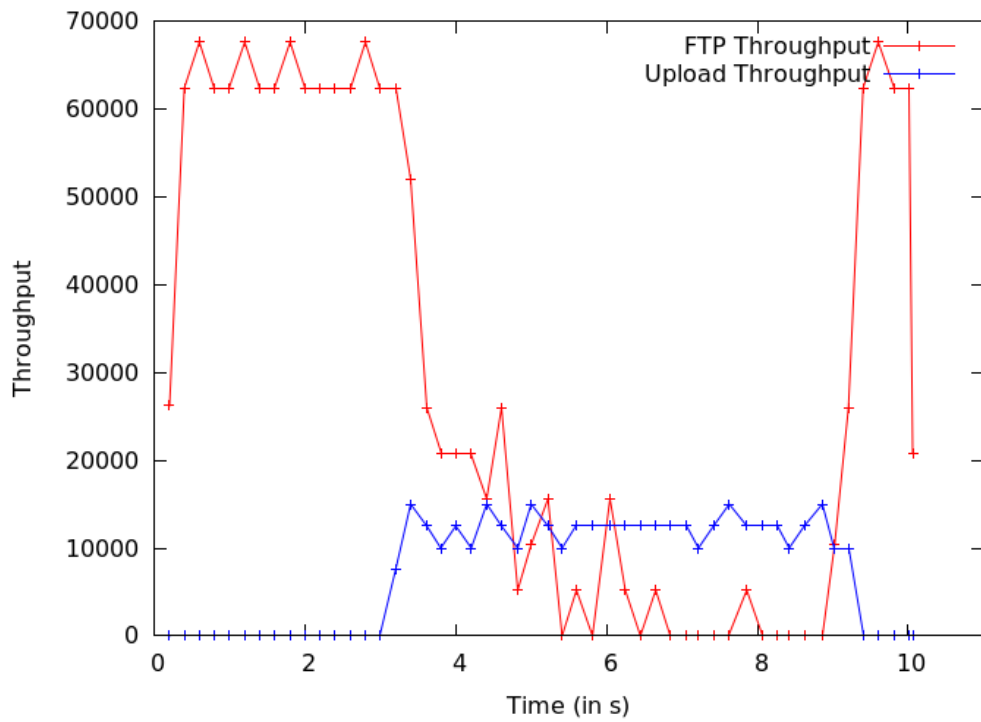

Figure 3: Throughput: bandwidth limitation, download and upload: even more restricted

1.5 Question 5

When the CBR upload connection is fixed to a rate of 30 kbps the FTP connection is not hindered by it as you can see in figure 4. To prevent hosts from interfering with each other, you could implement some sort of load balancing between hosts. In practice this would result in a (weighted) round-robin queue instead of a FIFO queue.
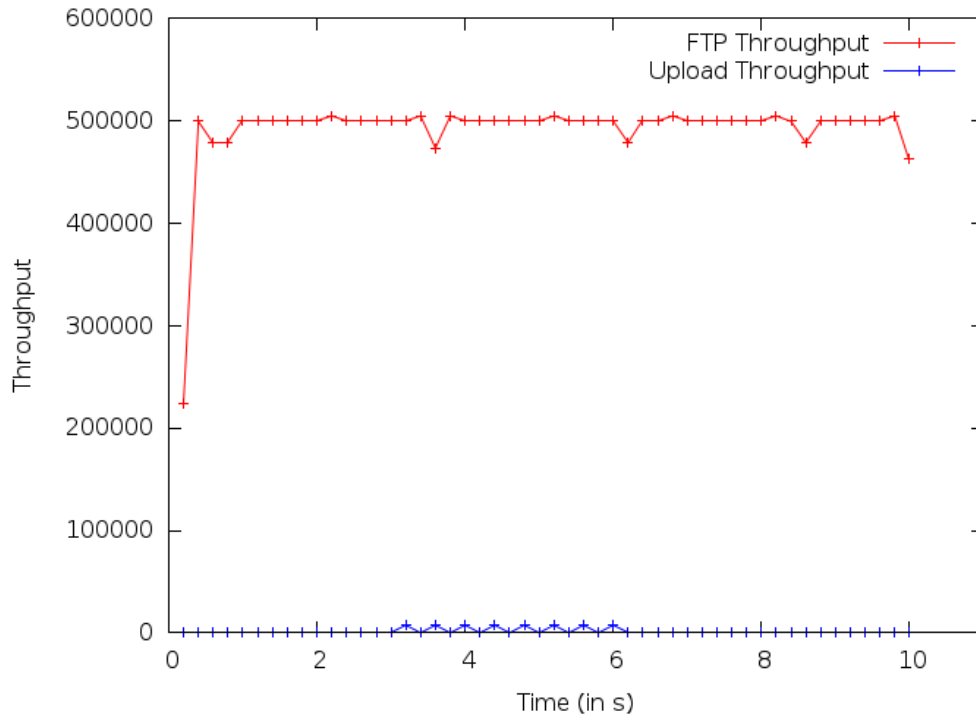
Figure 4: Throughput: upload rate at 30 000

1.6 Question 6

1.6.1 Question 6.1

Since the capacity has a linear increase relative to the load, we see a result similar to question 1 of this exercise. TCP will be responsible for equally distributing the bandwidth amongst the clients. Since big downloaders put more data into the network, they'll have more packets dropped. These big downloaders will then restrict their dataflow.

In the case where only 5 clients are connected with the modem we see a doubling in bandwidth used by each client. There are fewer connections competing for the same bandwidth.

1.6.2 Question 6.2

When the users perform their activities at random times, we expect varying performance. This is because not all the nodes will be active at the same time. In the worst case we get the performance like in the non-random case with 10 users.

If some users always send data when the DropTail queue is full, their packets will always be dropped. A different queue design (like round-robin) could aleviate this situation.

2 Exercise 2

2.1 Question 1

At 5s the throughput of the main FTP application drops, because of the burst connections to the webserver. In figure 5 and 6 we see that the throughput does not drop immediately at 5s. This is because the different connections to the webserver do not start immediately at 5s but at randomly from 5s to 7s.

2.2 Question 2

The different phases of the TCP algorithm are annotated on the graph in figure 7. At first we see the slow start. After we reach the slow start threshold we reach the additive increase phase. When the
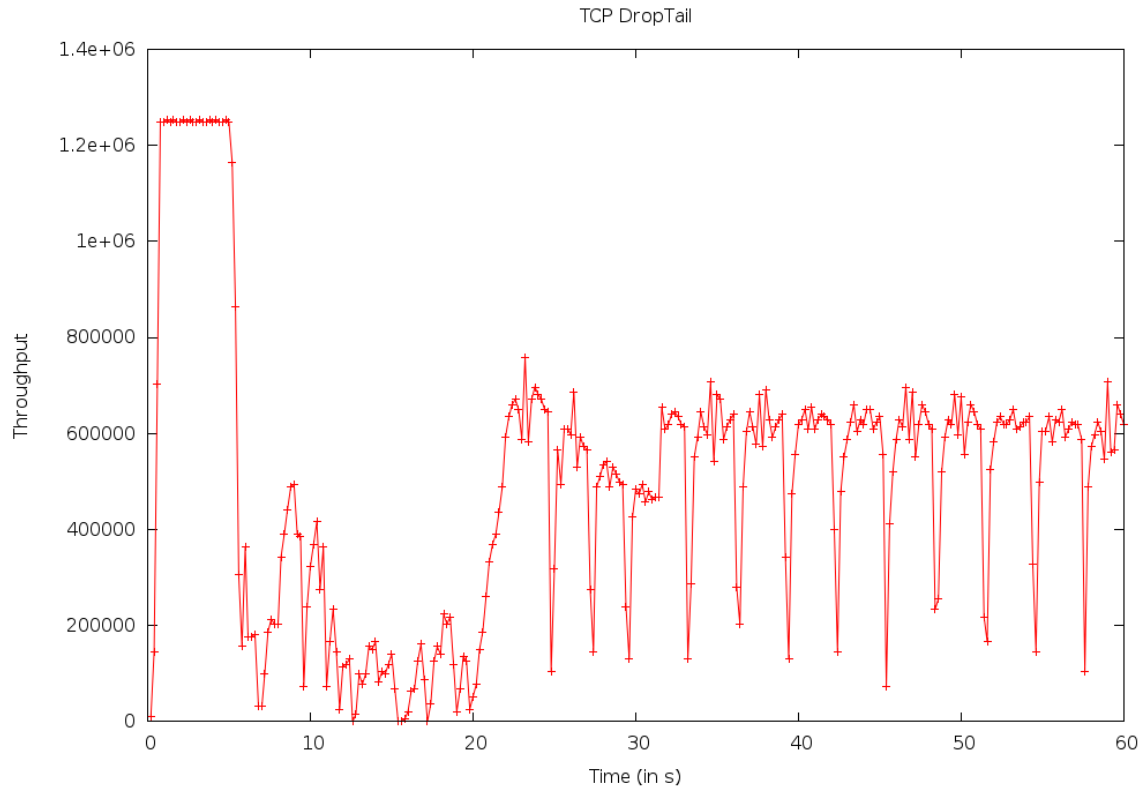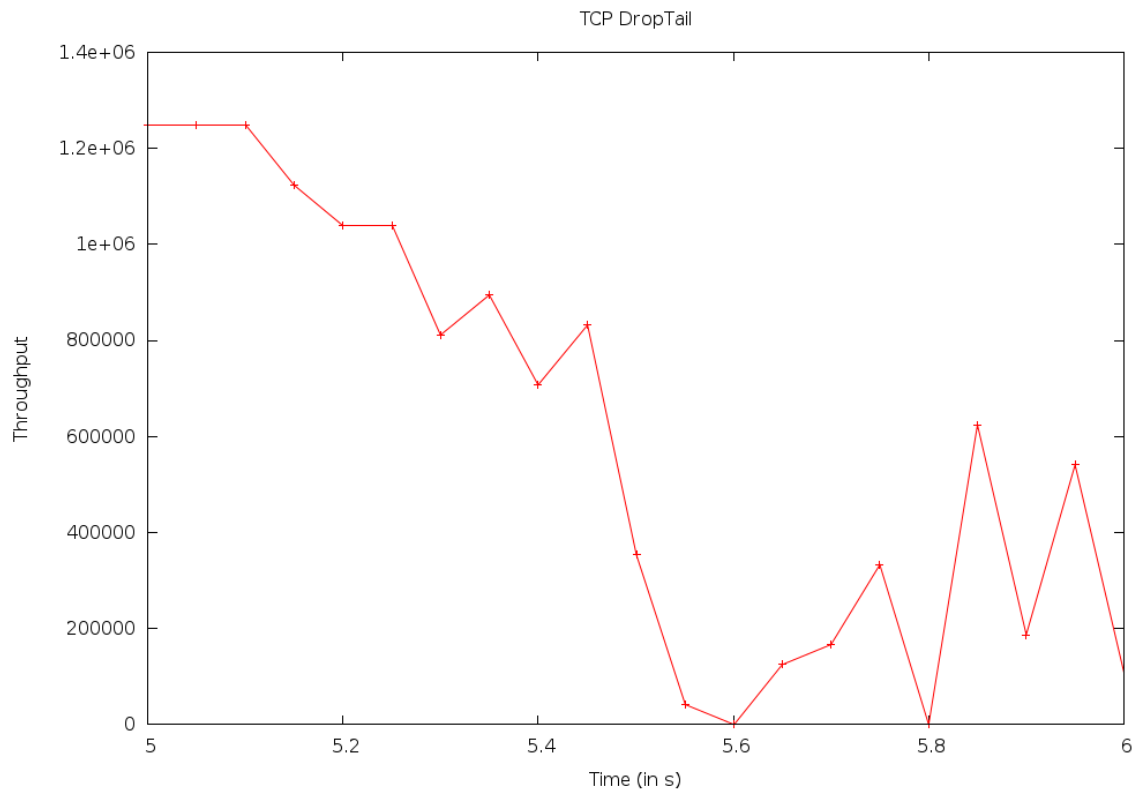
Figure 5: Throughput main FTP connection

Figure 6: Throughput main FTP connection (zoomed in)

connection times out/ to many packets are lost, the congestion window is set to 1 and the slow start threshold is set at half of the reached window size.
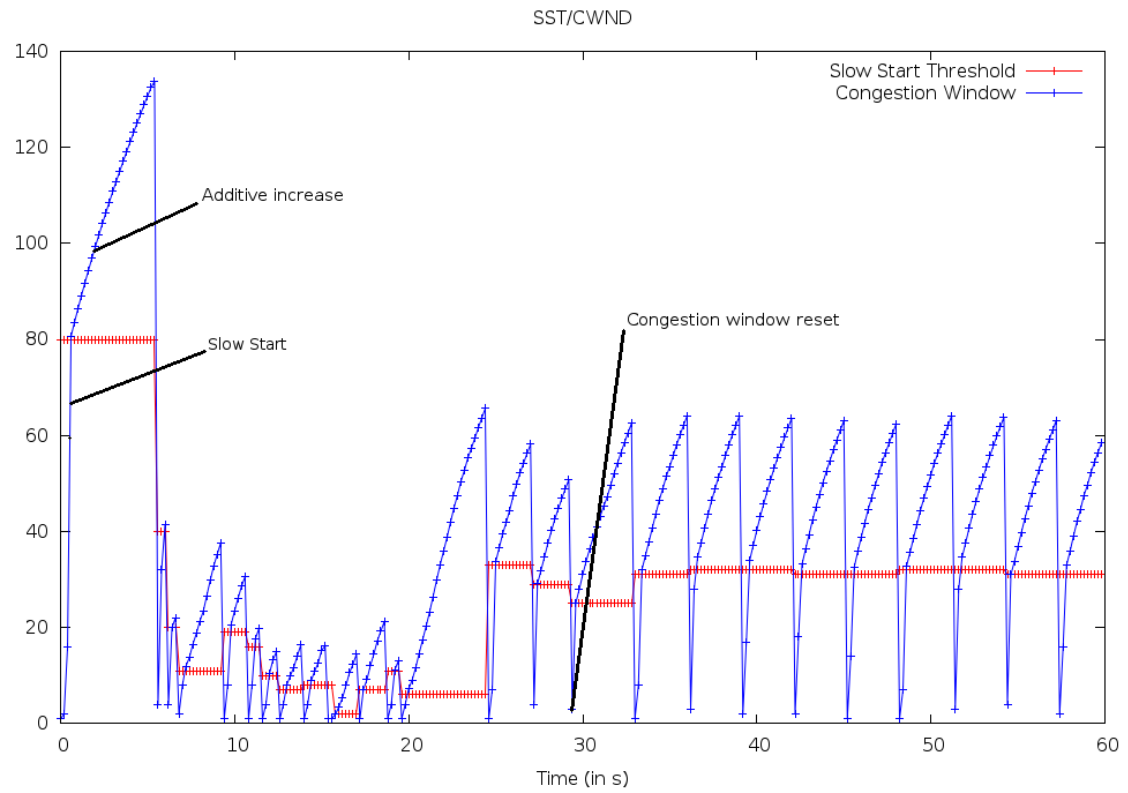
Figure 7: Throughput main FTP connection

2.3 Question 3

2.4 Question 4