KU Leuven

Assignment:

Network Simulation

Report

Authors:
Giuseppe Callari
Xavier Goás Aguililla

Professor:
prof. dr. D. Hughes

May 2nd, 2014

1 Exercise 1

1.1 Question 1

In figure 1 we can clearly see the difference between the scenario with both an uploader and a downloader and that without an uploader. In the scenario with an uploader, around the time the upload starts (3.0s), the throughput for the downloader drops drastically. In the scenario without an uploader, indicated in green, the throughput of the connection remains roughly constant.

1.2 Question 2

In figure 2 we see both the upload and download throughput on the same graph; we can clearly see the aforementioned drop in download speed at the moment the upload over CBR starts.

1.3 Question 3

We expect performance for CBR to be excellent, since a certain amount of bandwidth is always guaranteed for the upload connection. This does imply that the upload connection has a certain privilege over the download connection; to ensure this, we could tell the router to drop downstream packets rather than upstream packets. This would result in increased packet loss for the download connection, especially if we would run multiple upload connections at the same time.

1.4 Question 4

We can see in figures 3 and 4 that both the bridled and unbridled connection show similar patterns but differ at the tail end of the speed drop from 3.0s on. At 6.0s the CBR upload is stopped, but not all packets have been transferred yet due to packet loss.

The connection layer ensures these lost packets still arrive at their destination. To do this, the TCP connection persists while not all packets have arrived at their destination. Looking at figure 4, we see that the TCP upstream connection persists longer in the situation with the limited bandwidth. The packets transferred while the connection persists take up a certain amount of bandwidth. Only around the 9.5 second mark are all the packets transferred, and in figure 3 we can see that the FTP download is negatively affected by this transfer: the throughput is very low until the 9.5 second mark.

1.5 Question 5

When we impose a fixed bitrate on the CBR connection, we ensure that this connection never takes more than its preallocated share of bandwidth. Thus, when we allocate a small percentage of the total connection bandwidth, we ensure that other connections on the same link will not suffer from greatly limited performance, since TCP will not allocate more bandwidth than this fixed rate. TCP can then dynamically allocate bandwidth to other connections.

We set up a simulation in NS/2 and verified our assertion. We observed no packet loss, and only a very slight drop in throughput for the download connection in the period from 3.0s to 6.0s. The speed drop is due to packets from two different origins arriving at a router and being inserted into its buffer. Of course, in this scenario, packets cannot be immediately forwarded to their destination, causing a drop in throughput, which is small because of the limited amount of packets being sent by the CBR application.

1.6 Question 6

1.6.1 Question 6-1

A naive response would be to say that we expect performance to stay the same, since we have ten times the bandwidth for ten times the users. In actuality, this is not true. Consider the output of a simulation we made for this scenario.

We can clearly see that all transfers have very variable levels of throughput when executed simultaneously. The reason for this is that much more packets are being sent to the router simultaneously; the router can only process as much packets at a time as fit in its buffer, so once its buffer is full and more packets arrive, packet loss will occur. This is expressed in the valleys in the graph measuring the throughput.

If we have a smaller amount of nodes emitting packets, the probability of this scenario is also smaller. So, if we have five nodes, packet loss is less likely to occur than with ten nodes, and this will ensure a better throughput.

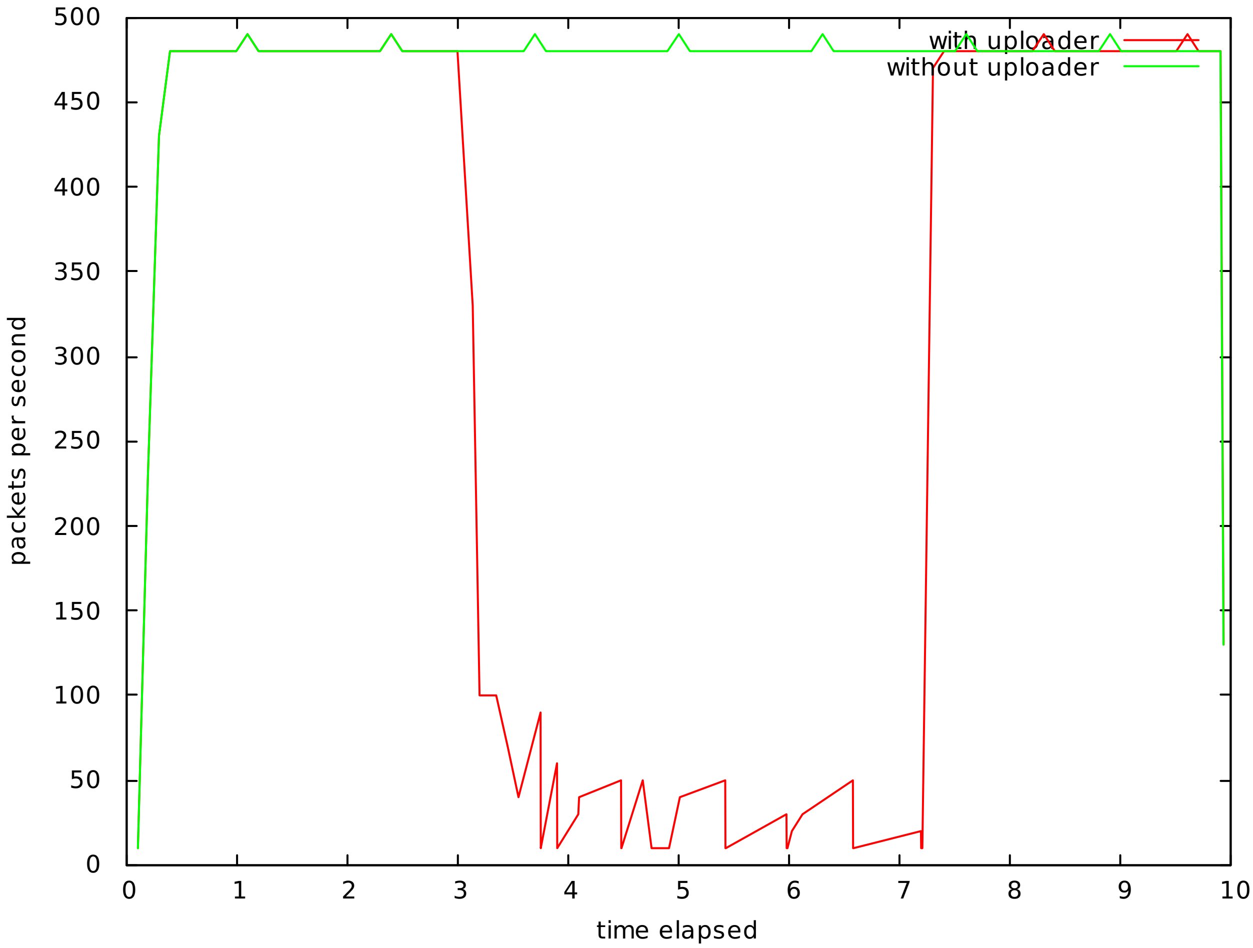

Figure 1: Download throughput compared

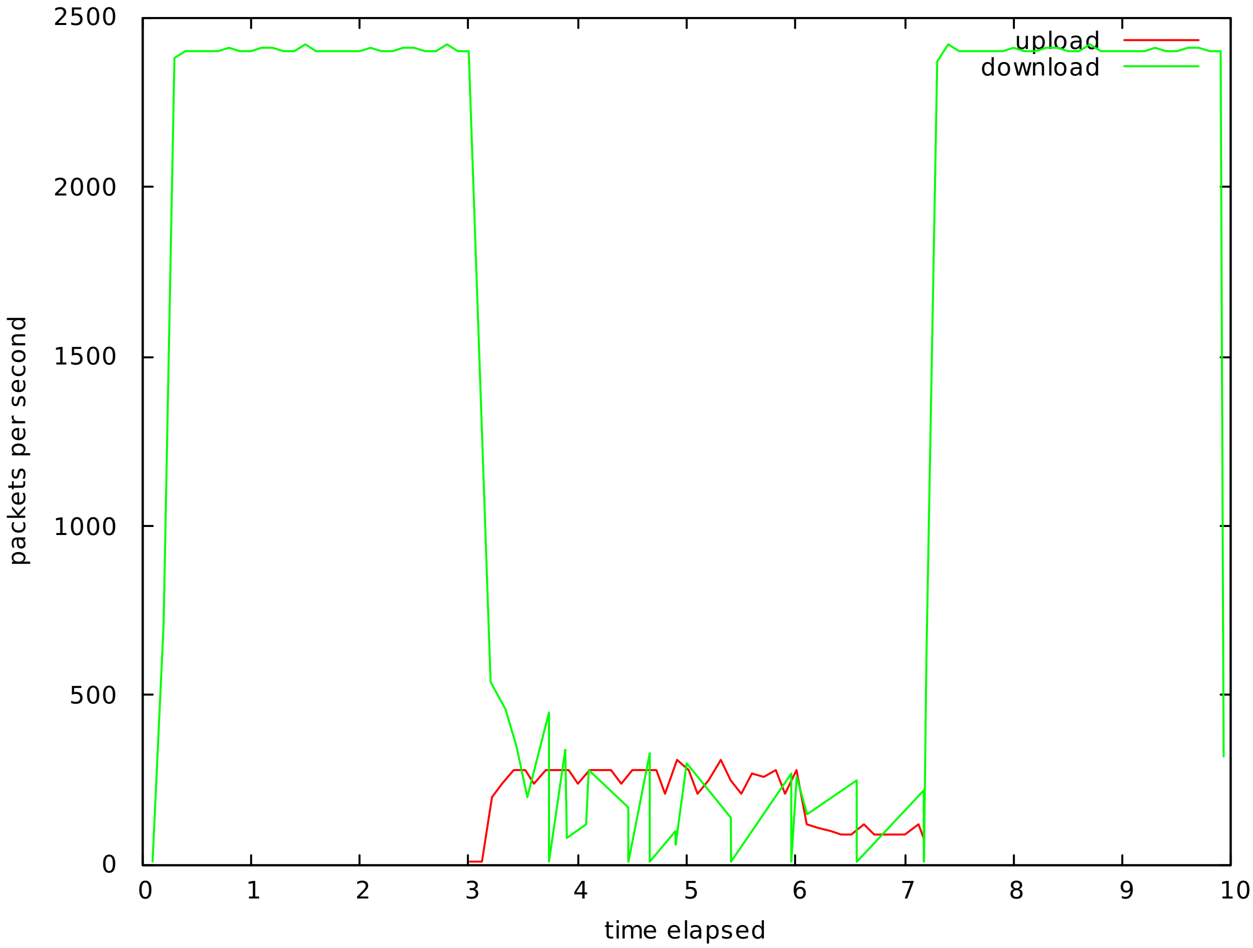

Figure 2: Upload and download throughput

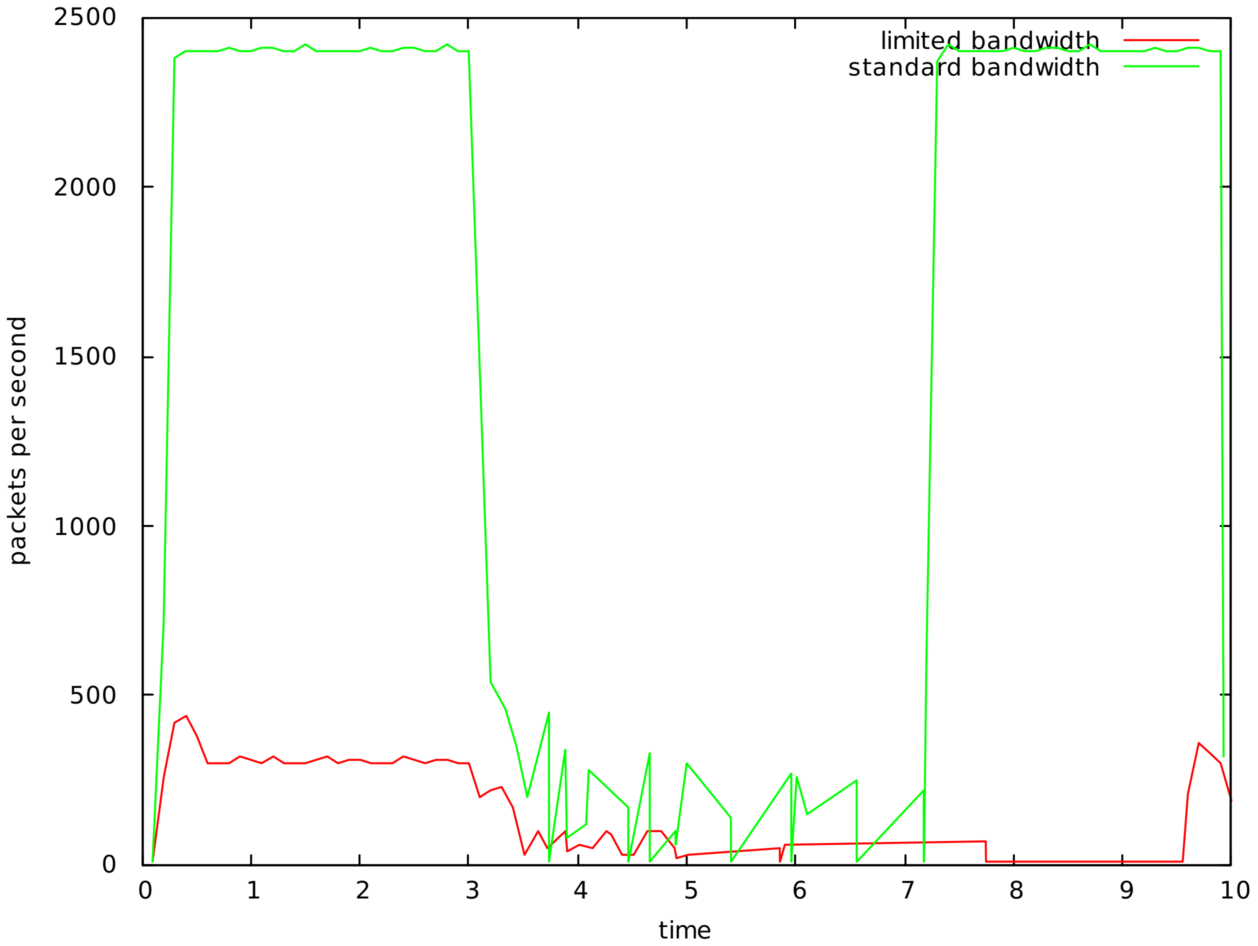

Figure 3: Download throughput

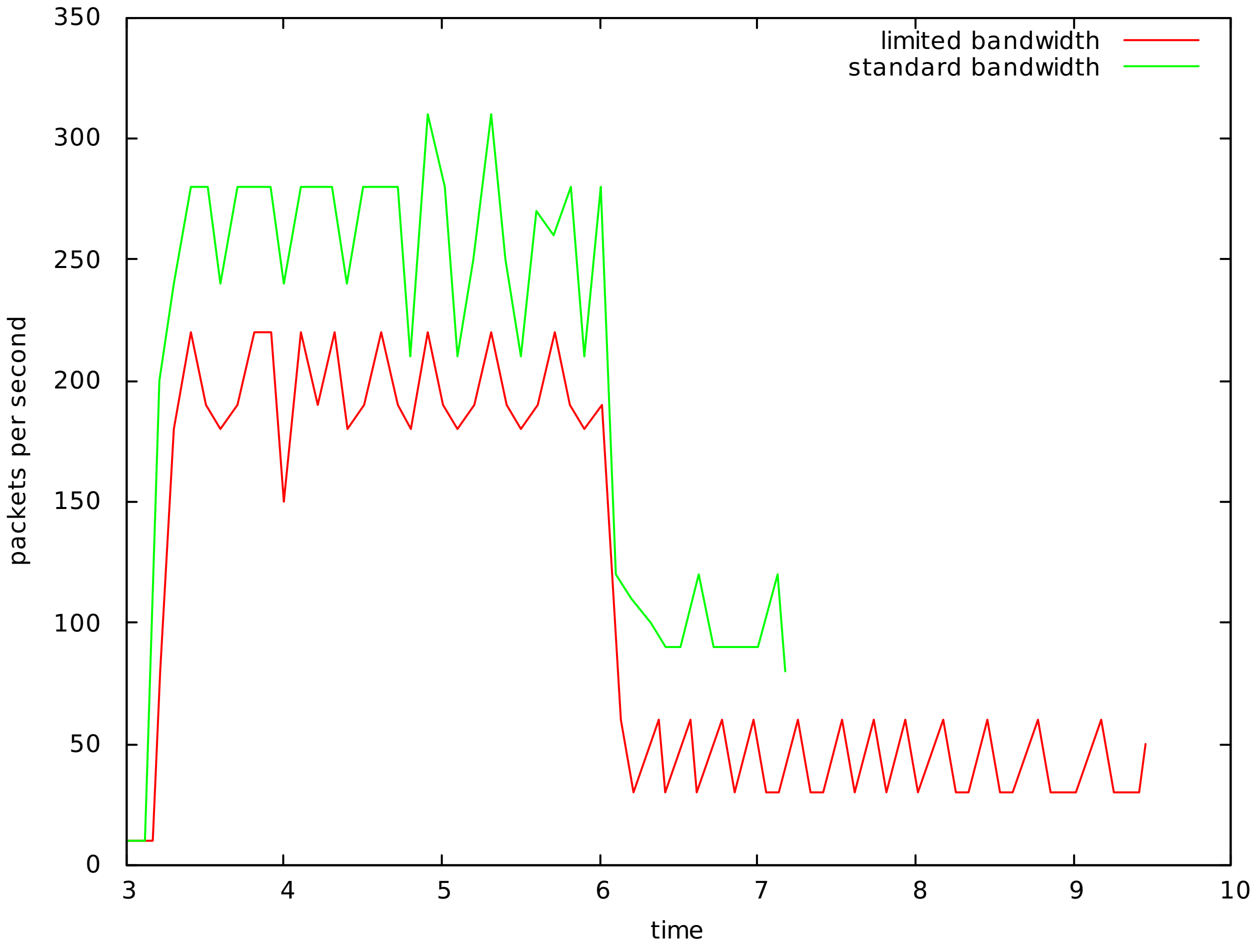

Figure 4: Upload throughput

1.6.2 Question 6-2

If the same activities are performed at random times, performance will be better than if all activities are performed all at the same time. The reason for this is that any given point in time, there will be less than ten users performing the same activity simultaneously (at least, the probability of this scenario is overwhelmingly greater than the converse case).

2 Exercise 2

2.1 Question 1

2.2 Question 2

2.3 Question 3

2.4 Question 4

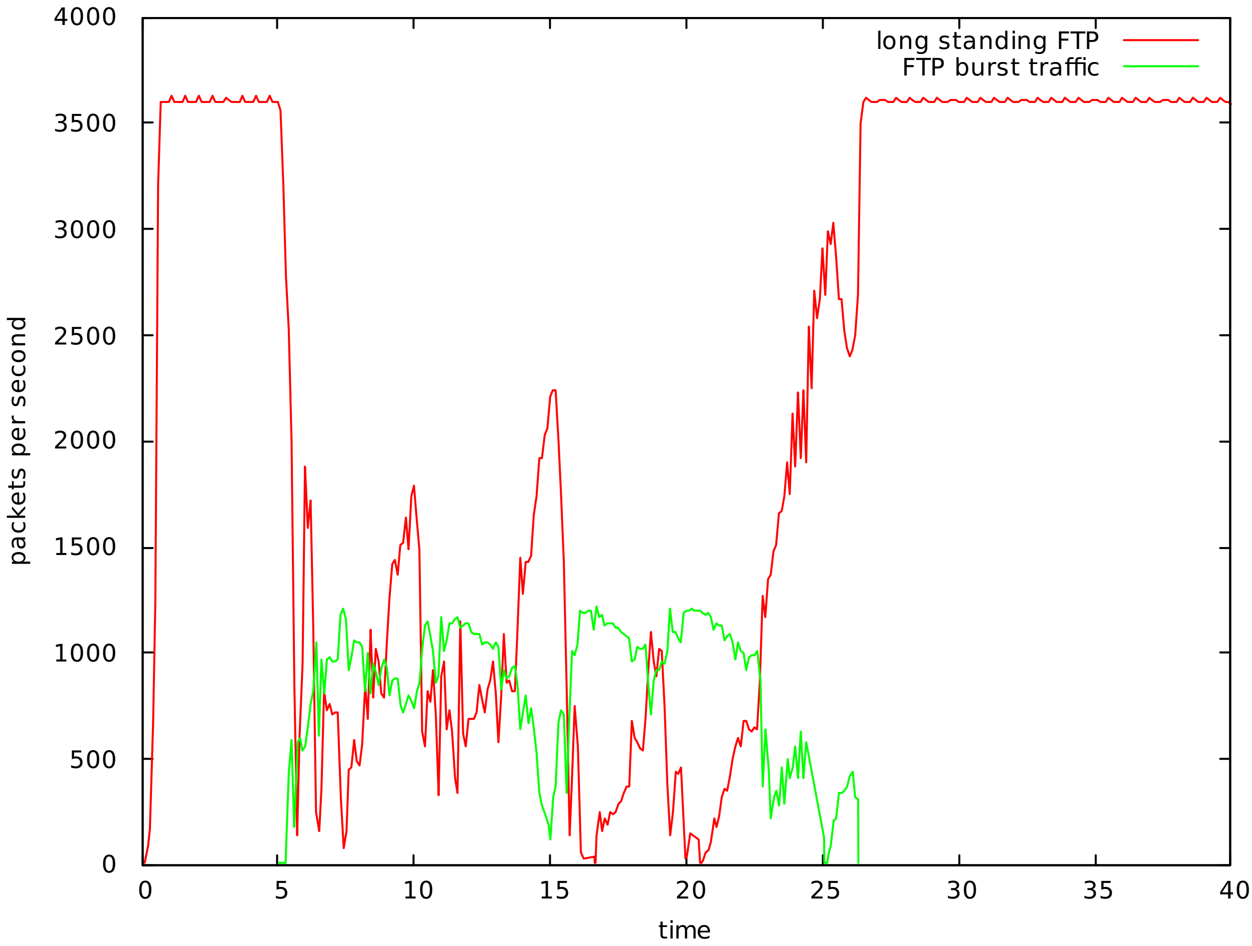

Figure 5: Throughput