CS 349:
Using a Packet Sniffer

Shrinivas Acharya
10010164
Computer Science and Engineering

Note: The assignment has been done using ns-3.16. Please use this version since some of the functions used which are available in this version might not available in older versions. Place all the scripts submitted in the scratch folder located at /path/to/ns-3.16/scratch

Sol1:

Part a: Created a point-to-point link in two nodes and started with a UdpClient on node 1 and UdpServer on node 2 and measured the throughput whilst varying the latency of the link from 0ms to 3000ms. It outputs the delay and throughput values.

Assumptions:

1.) Link Data Rate: 8Kbps
2.) Link Latency: Varying from 0 to 3000 ms
3.) Packet Size: 1024 bytes
4.) App Start time: 0 s
5.) App Stop Time: 20 s
6.) Delay is increased in step size of 500 ms at a time

Output The C++ script q1_a submitted computes the throughput for various delays and prints them on the terminal.

To run the script

Place the script in the folder /path/to/ns-3.16/scratch
$ cd /path/to/ns-3..16
$ ./waf --run q1_a

Observation: As the delay increases from 0ms to 3000ms the throughput decreases from 7.37 Kbps to 6.55 Kbps which is as per expected. As the delay increases the throughput decreases.

Part b: This part is same as the last part with minor modifications. In this part, two clients and two servers were installed on the nodes and data was exchanged between each client server pair.

The servers were configured with different port numbers and data was sent to corresponding ports to avoid any conflicts in the traffic.

Assumptions are same as the earlier part.

Output The script q1_b submitted computes the throughput of the two connections (averaged over the period of connection) against the delay in the link which is shown as output on the terminal. It also creates two pcap files by the name of q1_b-0-0.pcap and q1_b-0-0.pcap in the scratch folder as per the requirements of the question.

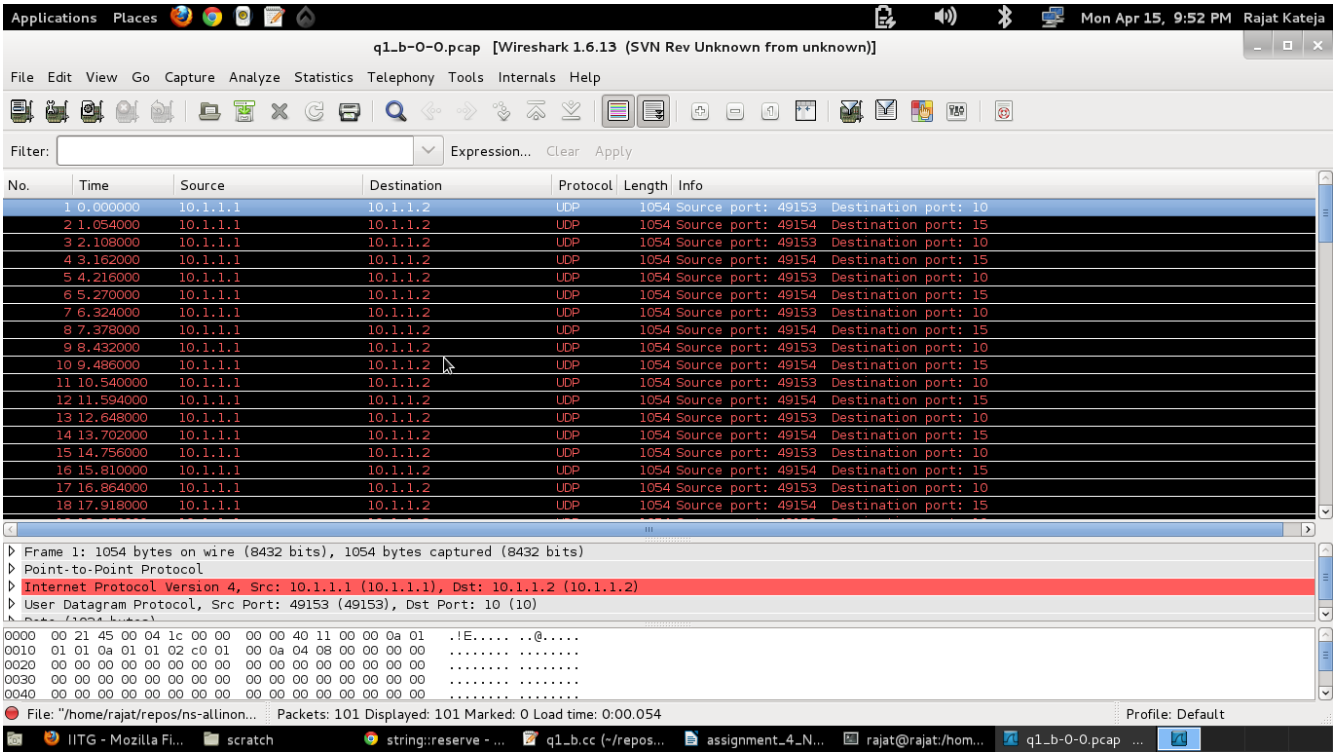
To run the script

Place the script in the folder /path/to/ns-3.16/scratch
$ cd /path/to/ns-3..16
$ ./waf --run q1_b

Observations:

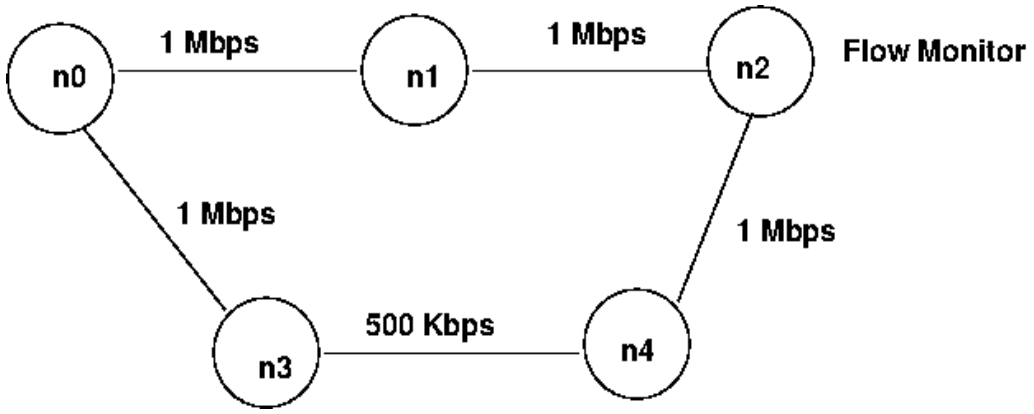
1.) Now in case of two client server pairs, both of the connections receive a lower throughput which changes from 3.6 to 3.2 as the delay is successively increased from 0ms to 3000ms.
2.) Both the connections receive approximately equal throughput as is expected due to the symmetry of the topology and connection and both the throughput go down in similar fashion.

Screen shot to confirm no conflict of traffic



As is clear from the screen shot, each packets is alternatively sent to port 10 and port 15 from the same source IP address to same destination IP client. This confirms that the traffic is not conflicting. The screen shot has also been submitted.

Sol2: The topology was built as per the diagram provided in the assignment sheet with the link capacities.



Assumption:

1. All the point to point links have a delay of 2ms.
2. IP address are assignmed with different subets to each point to point link. That is there are 5 different subnets with addresses 10.1.1.0, 10.1.2.0, 10.1.3.0, 10.1.4.0 and 10.1.5.0
3. The global routing tables are computed after 1 second of the link going down or coming up again. This is done from practicality otherwise there would be no packet losses.
4. Packet Size: 128 bytes. This choice was decided upon after trying various other packet sizes so as to get a good graph of the percentage loss.
5. Apps starting and ending times are as mentioned in the question.

Output: The C++ script q2.cc submitted computes the percentage of packet lossed against time for the flows generated in the simulation and outputs them on the terminal. Along with this it also generates a file "q2_plot.txt" in the scratch folder which can be used to generate the required graph in the file "q2.gif" using the gnuplot script submitted "q2.plt".

To run the script

Place the script in the folder /path/to/ns-3.16/scratch
$ cd /path/to/ns-3..16
$ ./waf --run q2
$ cd scratch/
$ gnuplot q2.plt

Observations / Explanations:

1. There are no flows till t = 1s, as per the start and stop times set.
2. From t = 1 to 1.5 s, there is only one flow from node 0 to node 2.
3. At t = 1.5s second flow from node 0 to node 1 starts.
4. There are very few losses till t = 2s at which time the link between node0-node 1 goes down.
5. Between the time of link failure at t = 2s to the time of recomputing routing tables at t = 2.1s, the losses in both the flows increase significantly since packets are being dropped on the node0 – node 1 link.
6. After this there are two new flows between the same pair of nodes, but now between different

IP addresses on the nodes since the path has changed, and hence the interface and IP address has also changed.

7.) At time t = 2.7 the link is again put up, but still the packets follow the same path since the tables are not recomputed till t = 2.8s. At t = 2.8 the tables are recomputed and now again the original paths are followed.

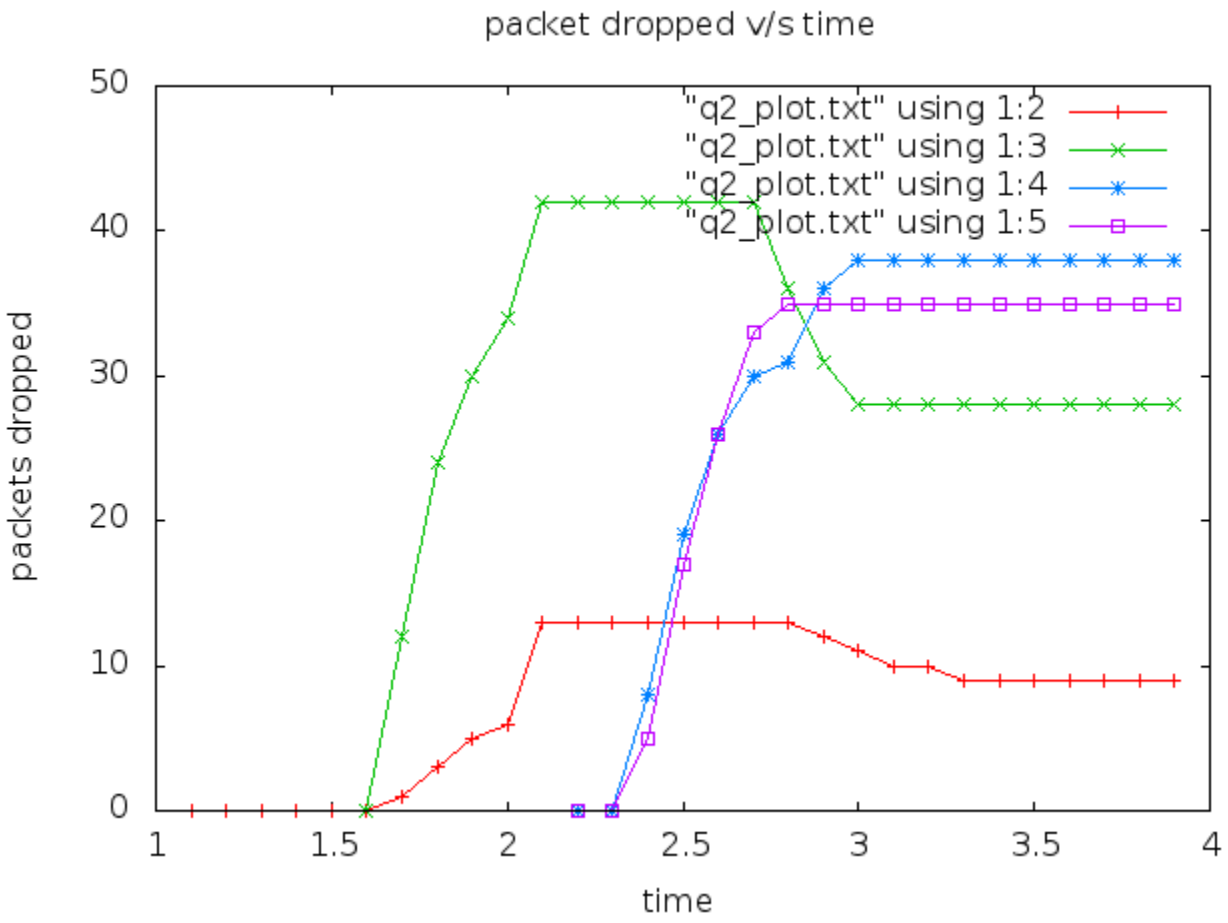
8.) The percentage loss for the first two deceases a bit after some time because more packets are successfully received and hence the percentage (#pcaketsDropped/#packetsReceived) decreases. For the other two flows, the percentage loss remains constant as no more packets are received on those flows after re computation of the paths.

Graph:

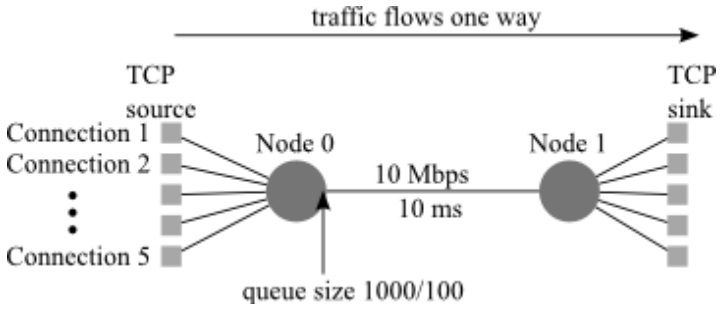
In the graph that follows, the following color code is in effect:

First Flow (before link goes down and after it comes up):
node0-node2 : Red Line
Second Flow (before link goes down and after it comes up):
node0-node1 : Green Line
First Flow (after link goes down):
node0-node2 : Blue Line
Second Flow (after link goes down):
node0-node1 : Purple Line

The graph follows the explanations as given above.



Sol 3: The topology was set up according to the given specifications and five TCP connections were set up as required.



*Part a and Part b:*Part a required the computation and plotting of the throughput of the five connections over the period of time for which the connections. Part b required the computation and plotting of queue size against time.

Output: In the script q3_a.cc throughputs are computed and printed onto the terminal. Along with this, it also creates 5 files, one for each connection named q3_con<i>.txt (i = 1, 2, 3, 4, 5) in the scratch folder which has the throughput values. These files are used by the gnuplot script “q3.plt” submitted to generate the required graph in “q3_throughput.gif”.
In the script q3_b.cc the queue size is computed and printed on the terminal. Along with this it also creates a file “q3_qsize.txt” which is used by the gnuplot script “q3.plt” to generate the required graph in “q3_queue_size.gif”

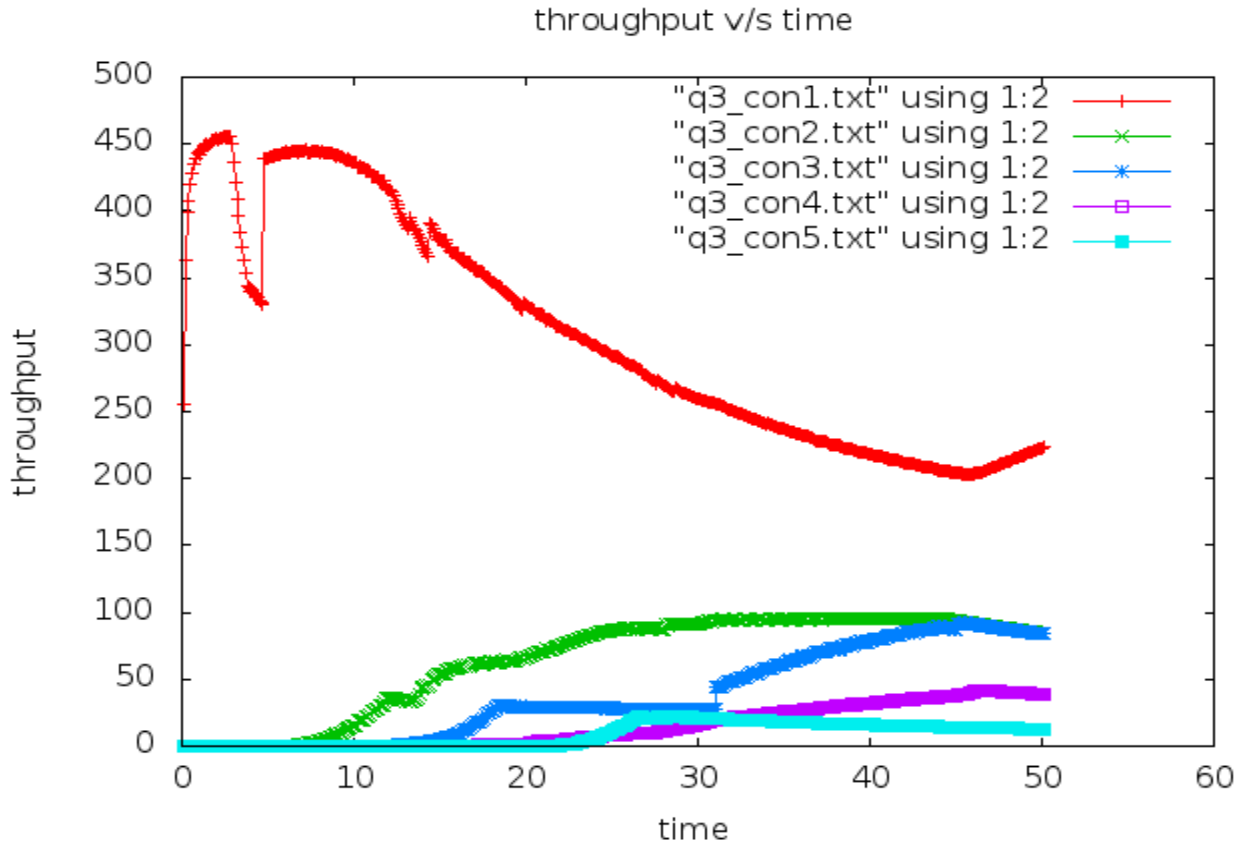
To run the script:

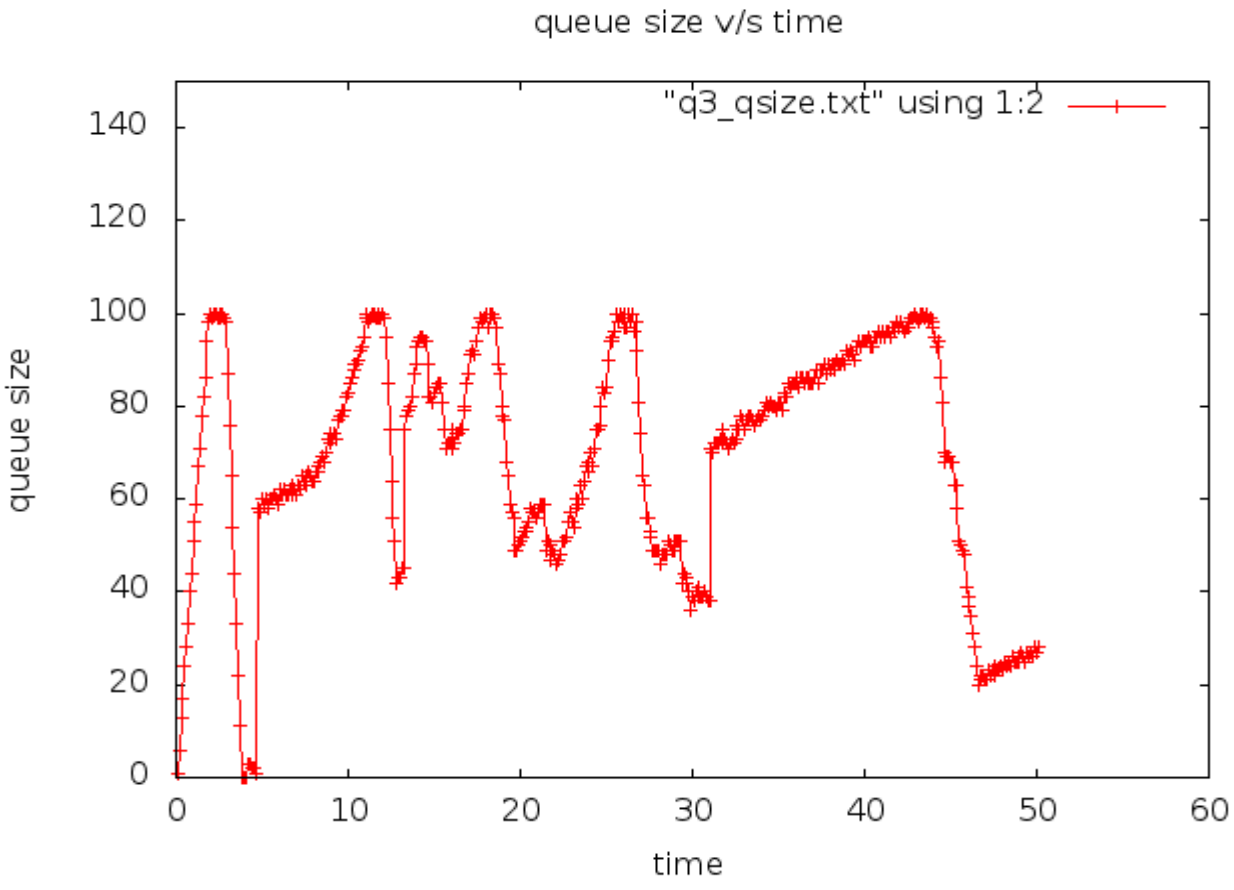
Place the script in the folder /path/to/ns-3.16/scratch
$ cd /path/to/ns-3..16
$ ./waf --run q3_a
$ ./waf --run q3_b
$ cd scratch/
$ gnuplot q2.plt

Answers:

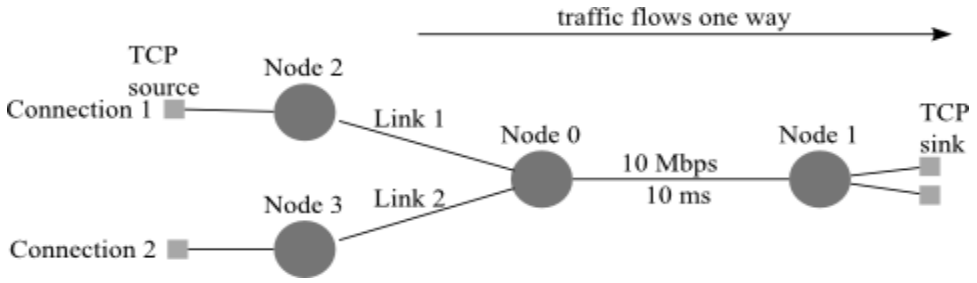
1.) TCP is quite fair in its allocation of resources as can be seen from the distribution of bandwidth in the throughput graph. It starts in a skewed fashion but slowly all the connection's throughput start converging towards a bandwidth which is equable amongst the five connections.
2.) Queue size rarely goes below 40% usage and infact most of the time remains above 60% use as is clear from the graphs attached below.

Graphs:


throughput v/s time
"q3_con1.txt" using 1:2
"q3_con2.txt" using 1:2
"q3_con3.txt" using 1:2
"q3_con4.txt" using 1:2
"q3_con5.txt" using 1:2
throughput
time
0
50
100
150
200
250
300
350
400
450
500
0
10
20
30
40
50
60


queue size v/s time
"q3_qsize.txt" using 1:2
queue size
time
0
20
40
60
80
100
120
140
0
10
20
30
40
50
60

Sol 4: The connection is set up as per the the specification of the question



Specifications:

1.) Link 1 and Link 2 capacity: 1.5Mbps
2.) node0-node1 link capacity: 10Mbps
3.) Delay on all links: 10ms. Delay of link 2 is varied for part b of the question.
4.) Both connections are run from time t = 0s to t = 5s.

Part a: In this part the throughputs were measured as a funtion of time to observe how the bandwidth is divided amongst the two flows

Output: The script q4_a submitted computes the throughput of the two connections over the period of time for which the connections are functional and prints them on screen. It also generates a q4_a_plot.txt file in the scratch folder to be used with the gnuplot gile q4.plt

Part b: In this part the delay of second link was increased and the throughput over the period of time was observed.

Output: The script q4_b submitted computes the throughput of the two connections (averaged over the period of connection) against the delay in link 2 which is shown as output on the terminal. Along with this it also creates a file “q4_plot.txt” in the scratch folder from which the required graph by the name of “q4_gif” can be made using the gnuplot script “q4.plt” submitted.

To run the script:

Place the script in the folder /path/to/ns-3.16/scratch
$ cd /path/to/ns-3..16
$ ./waf --run q4_a
$ ./waf --run q4_b
$ cd scratch/
$ gnuplot q4.plt

Observations:

1.) The throughput increases with time and converges towards a fixed maximum value which is governed by the specifications of the links.
2.) The throughput of the two connections is more or less equal, which is as per the expectations from a symmetric topology and connection setup.
3.) The throughput of the first link goes up and that of the second link goes down as the delay in the second link is increased from 0ms to 2500ms using time steps of 100ms.
4.) As the RTT of link 2 increases (due to increasing delay), the network becomes increasingly unfair in its allocation of the resources, and link 1 gets hold of a major part of the bandwidth available.

Graph: