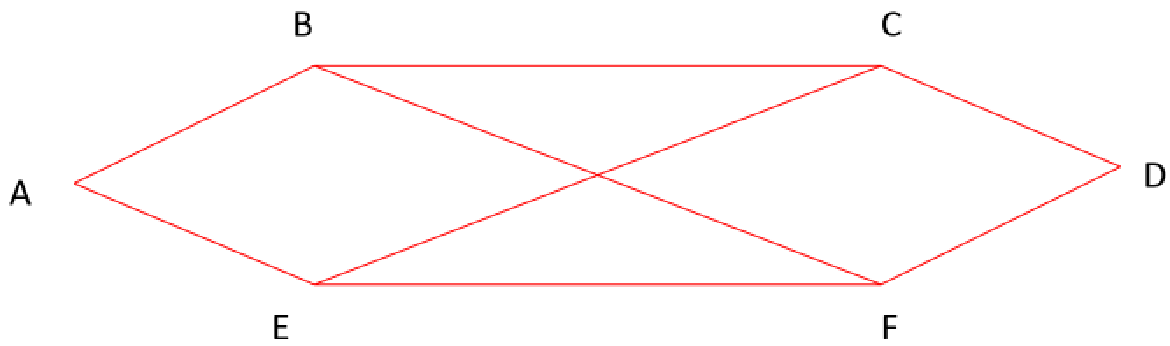
EE460 Network Design Project

6-Node Network

Renato Vilca Valderrama



6 - node network

Initial Calculations

Traffic Matrix (MPackets/second) & Routing Matrix (Symmetric)

	A	B	C	D	E	F
A		AB 9	ABC 4	ABFD 1	AE 7	AEF 4
B	BA 9		BC 8	BFD 3	BFE 2	BF 4
C	CBA 4	CB 8		CD 3	CE 3	CEF 2
D	DFBA 1	DFB 3	DC 3		DCE 3	DF 4
E	EA 7	EFB 2	EC 3	ECD 3		EF 5
F	FEA 4	FB 4	FEC 2	FD 4	FE 5	

I begin by calculating node link Capacity ("C" in Mbits/s):

Using the given Traffic Matrix and given value for mean packet size (800 bits per packet) a Capacity Table is created.

λ_{AB} = 9+4+1 = 14 Mpackets/s

$\mu C_{AB} = 2*\lambda_{AB}$ = 28 Mpackets/s

C_{AB} = 28 / (1/800) = 22400 Mbits/s

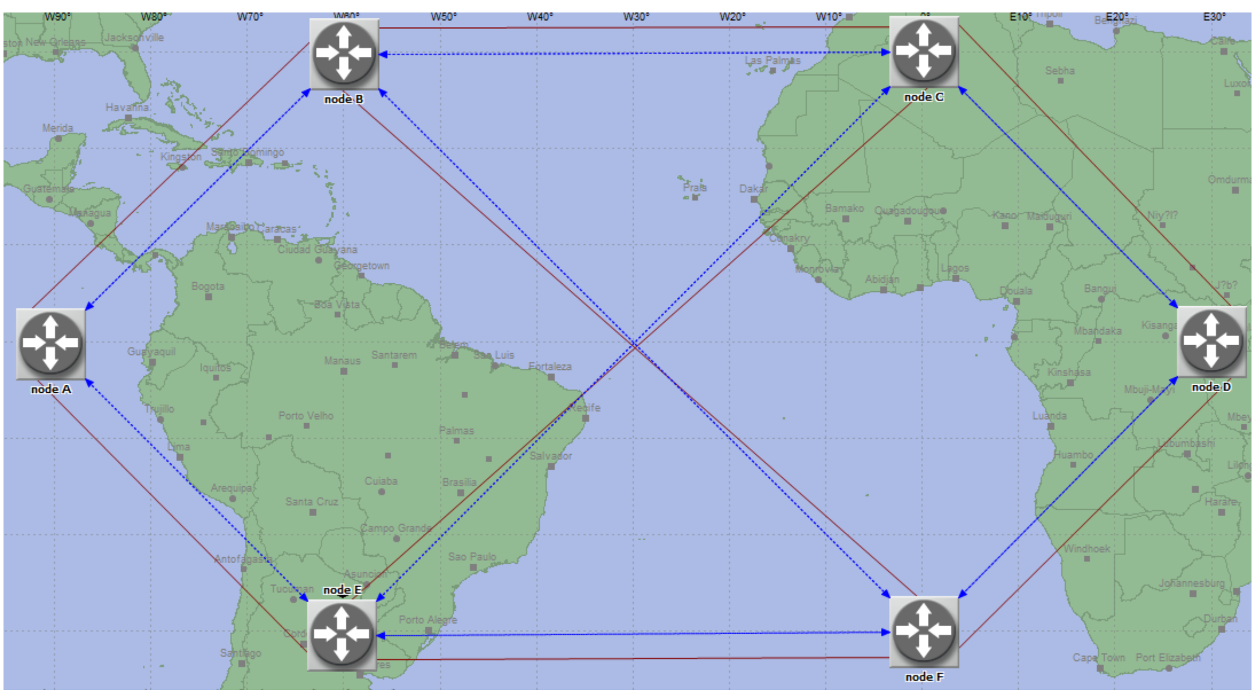
Following this procedure, the Capacity Table below can be filled out.

Capacity Table (Mbits/second)

	λ (Mpackets/s)	μC (Mpackets/s)	C (Mbits/s)
AB	14	28	22400
BC	12	24	19200
AE	11	22	17600
EF	13	26	20800
FD	8	16	12800
BF	10	20	16000
EC	8	16	12800
CD	6	12	9600

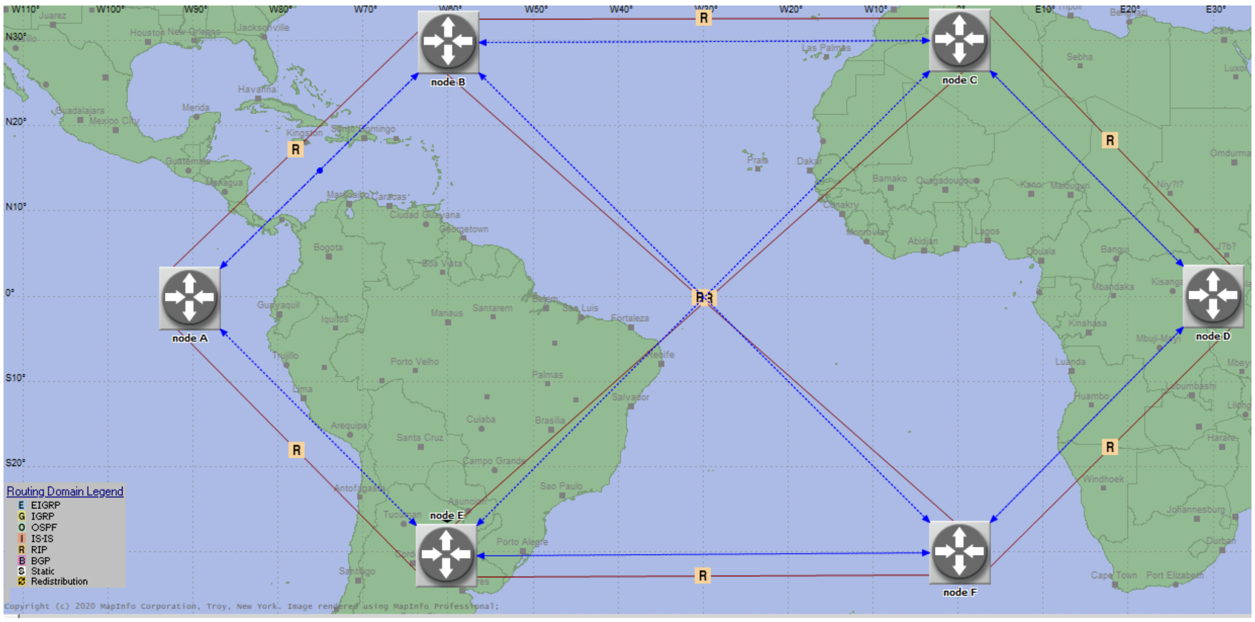
6-Node Network Creation in Riverbed

Next, I create a 6-node network in the Riverbed application using 6 Ethernet4_slip8_gtwy routers, connected by several 100BaseT Links with duplex IP_traffic_flow in between nodes being used to monitor traffic.

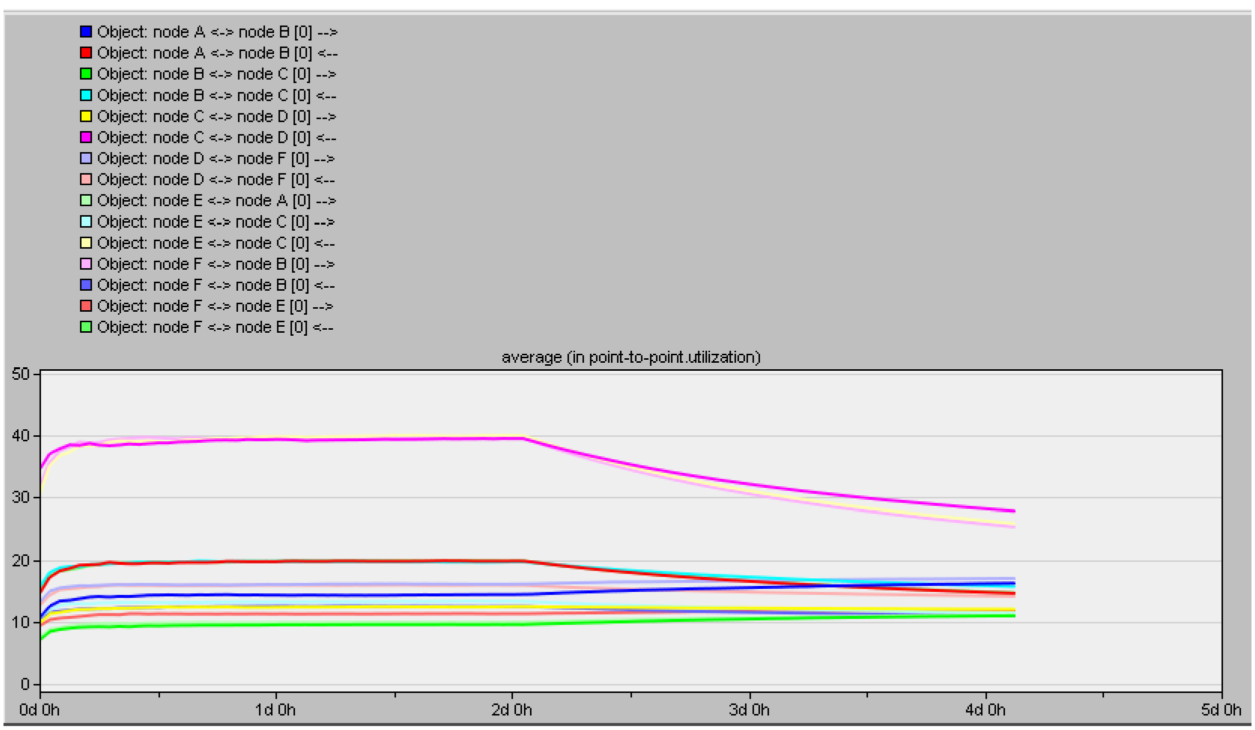


RIP Routing Algorithm

The first routing algorithm used is the RIP routing algorithm. I first edit the IP_traffic_flow link's attributes. I edited all the Traffic (bits/second) and Average Packet Size (Bytes) to contain the Capacity Table and the given 800 bits/packet values (equal to 100 bytes/packet), respectively. For all links, I used 10 min/step intervals.

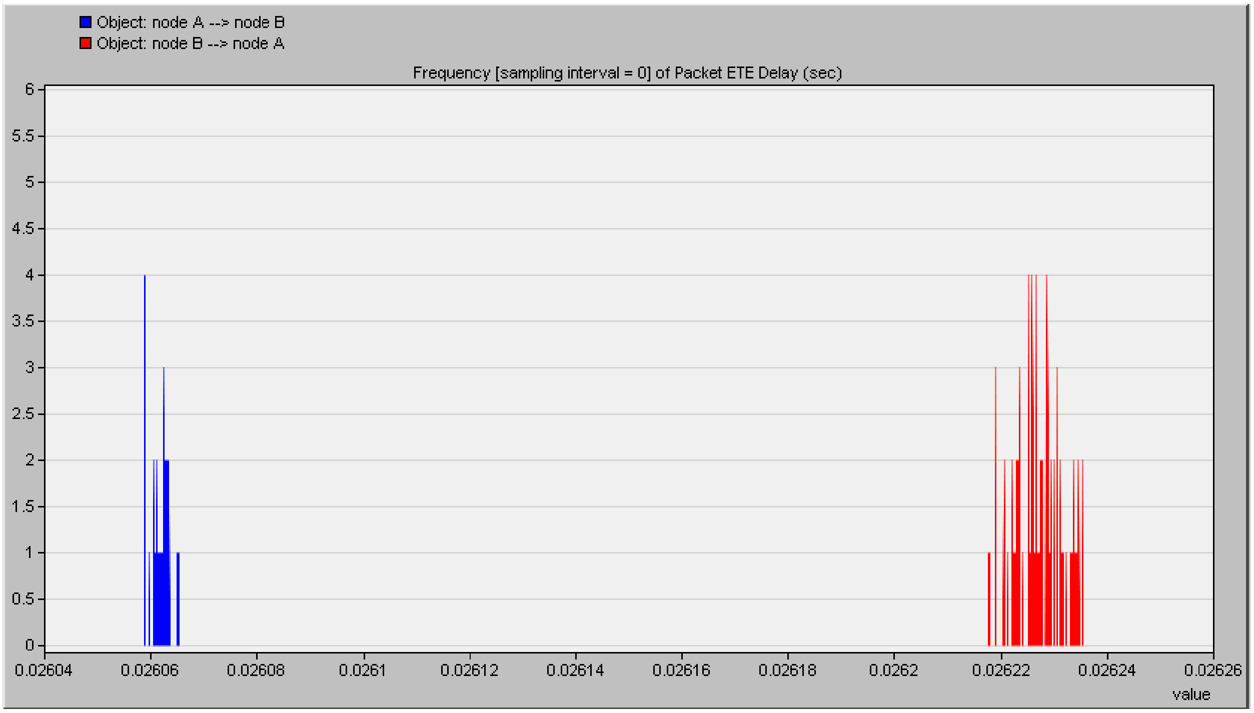


The first simulation I ran contained the forward and backward average utilization of every link over 100 hours. Our utilizations were all <=50%.

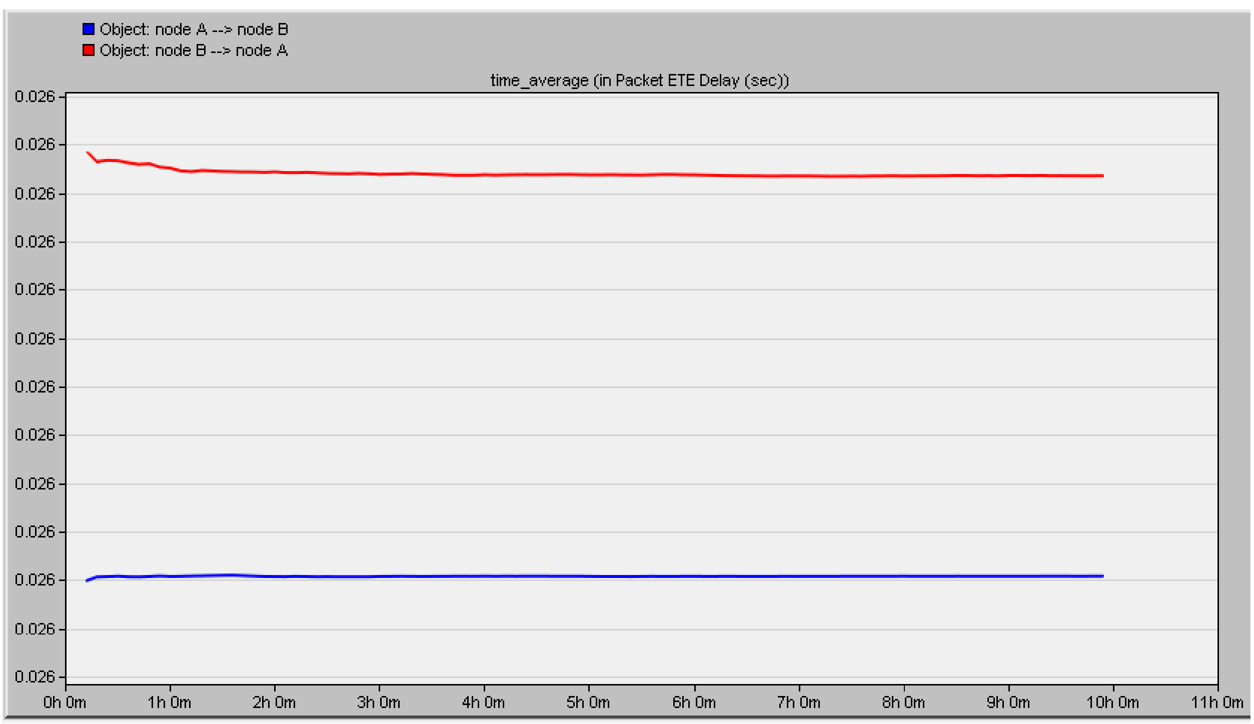


Next, I attained the histograms for link delay and found the average delay for two links. I set the next simulation to show Packet ETE Delay for Node A to B and Node B to A.

First the histogram

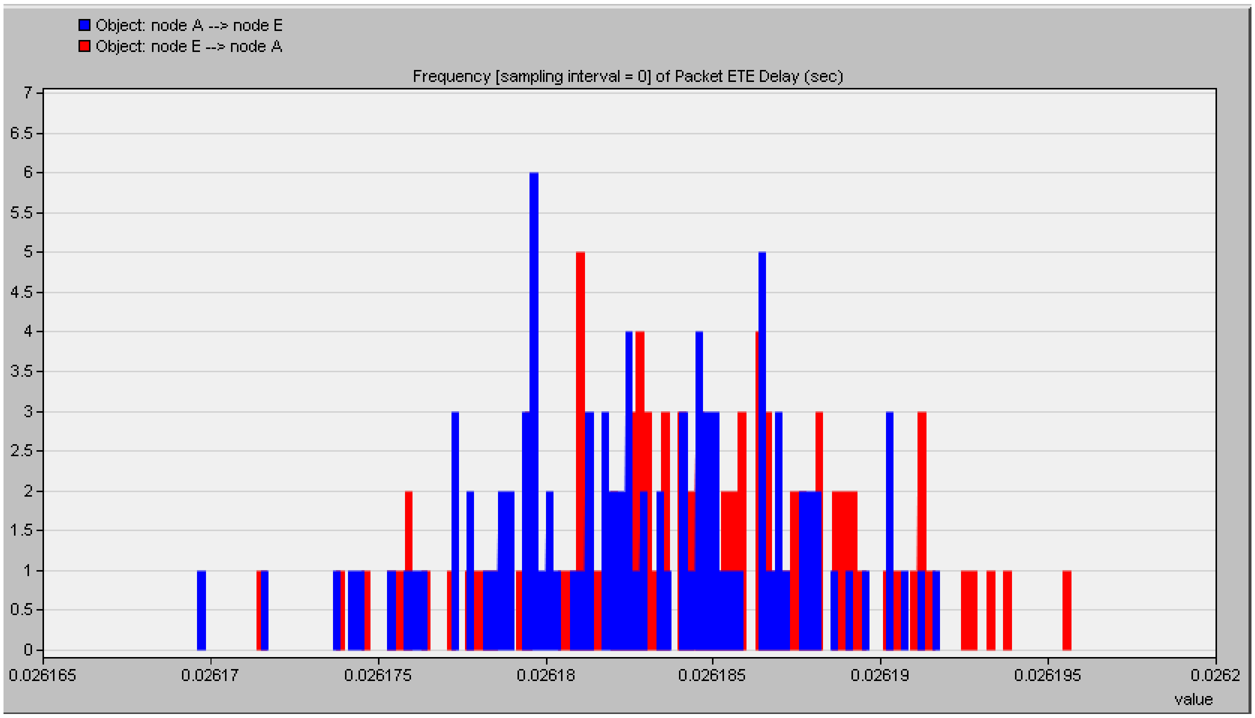


Then the time average

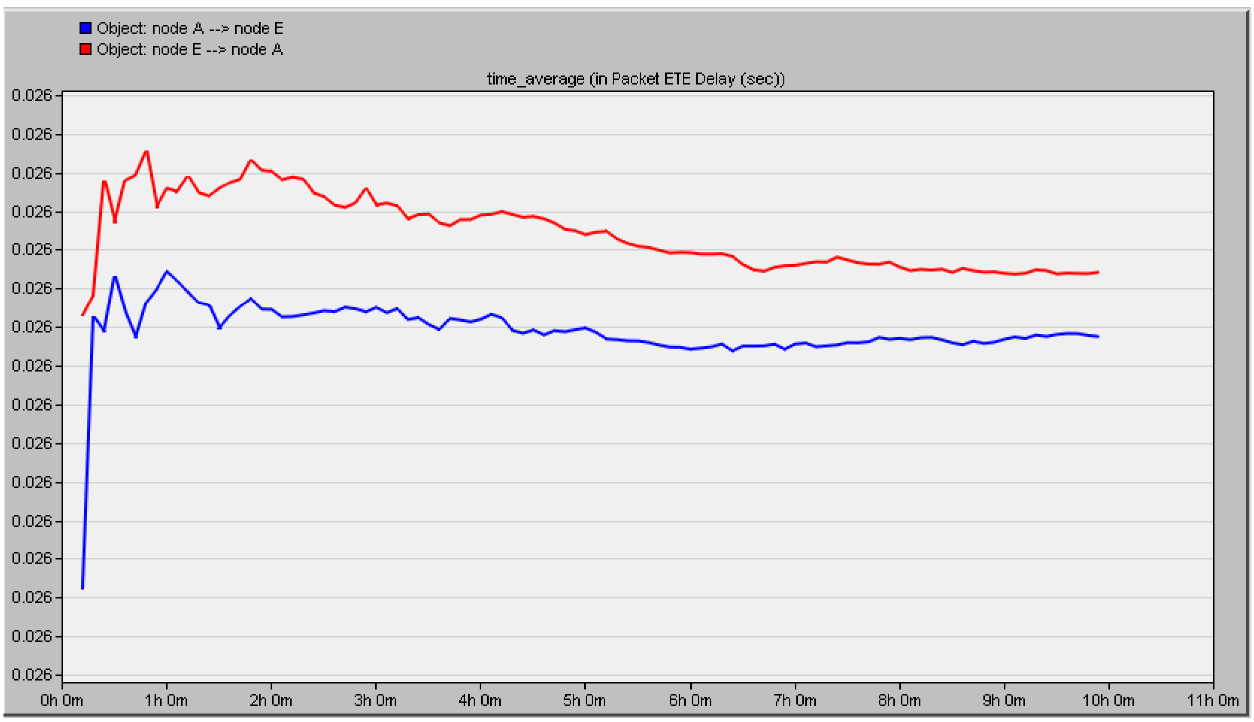


I then repeated the process for Node A to E and Node E to A.

First the histogram



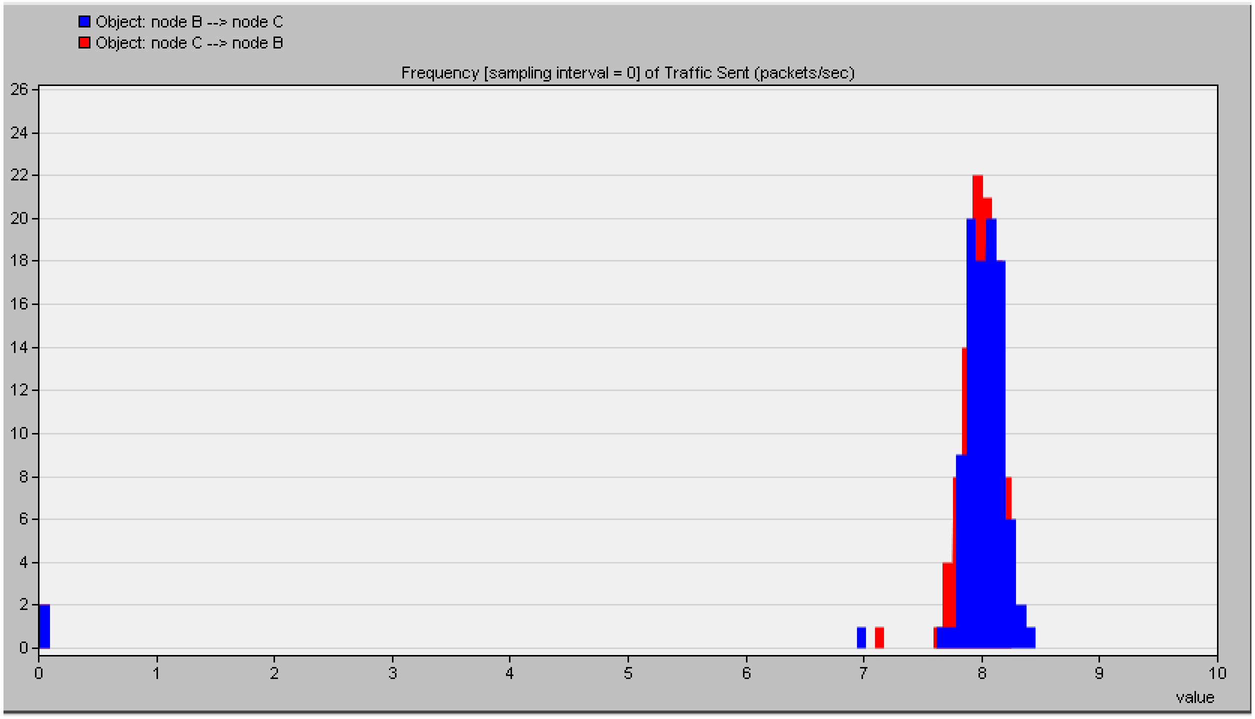
Then the time average



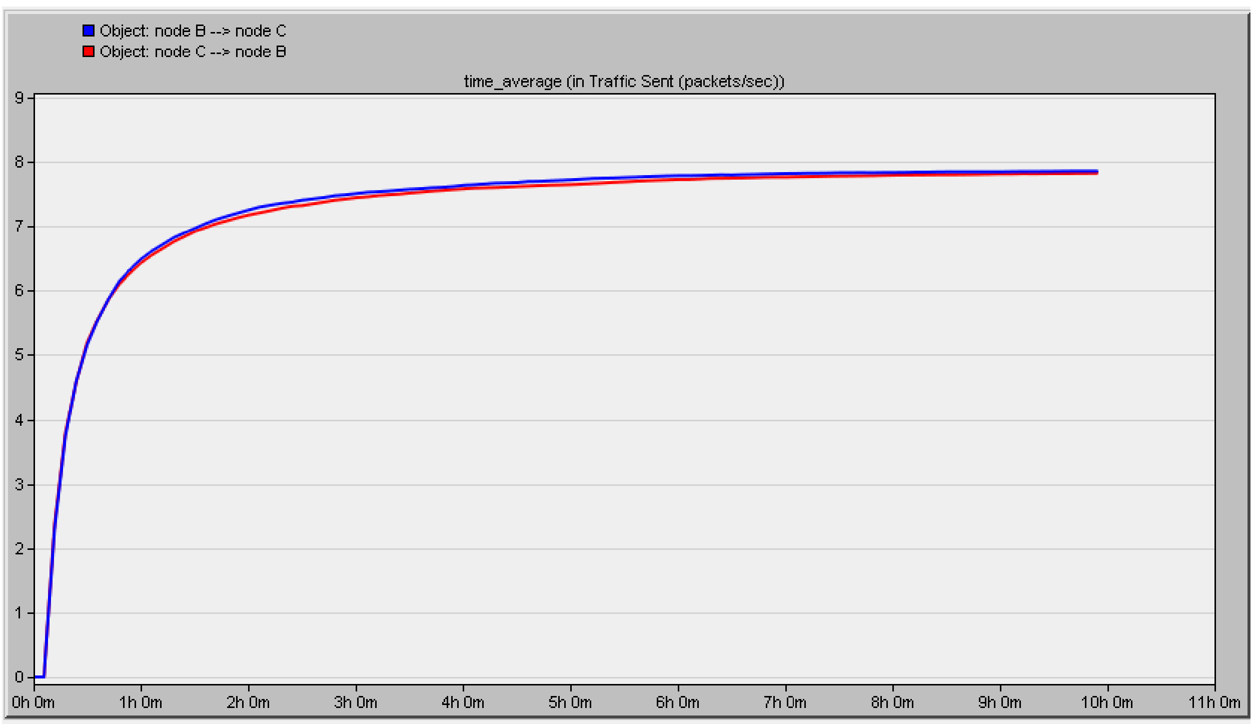
The average delay for link A to B is 0.026s and the average delay for link A to E is also 0.026s.

Next, I attained the histograms for link flows and found the average flows for two links. I set the next simulation to show Traffic Sent for Node B to C and Node C to B.

First the histogram

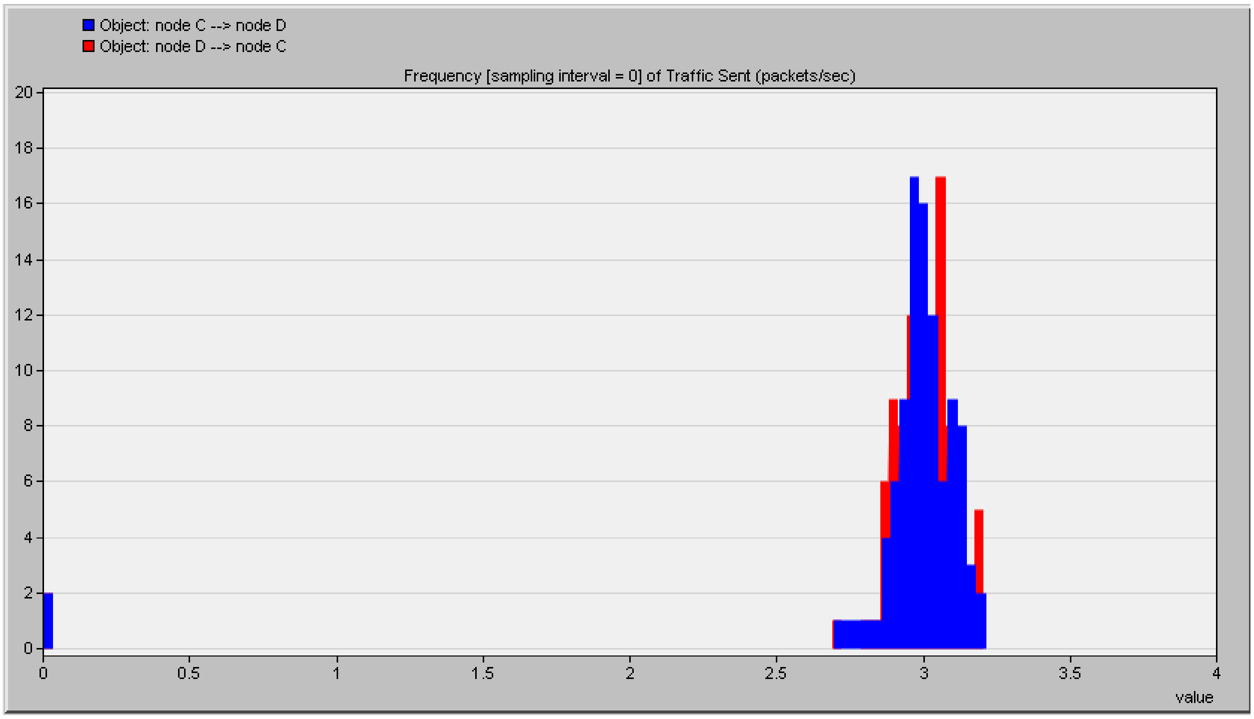


Then the time average

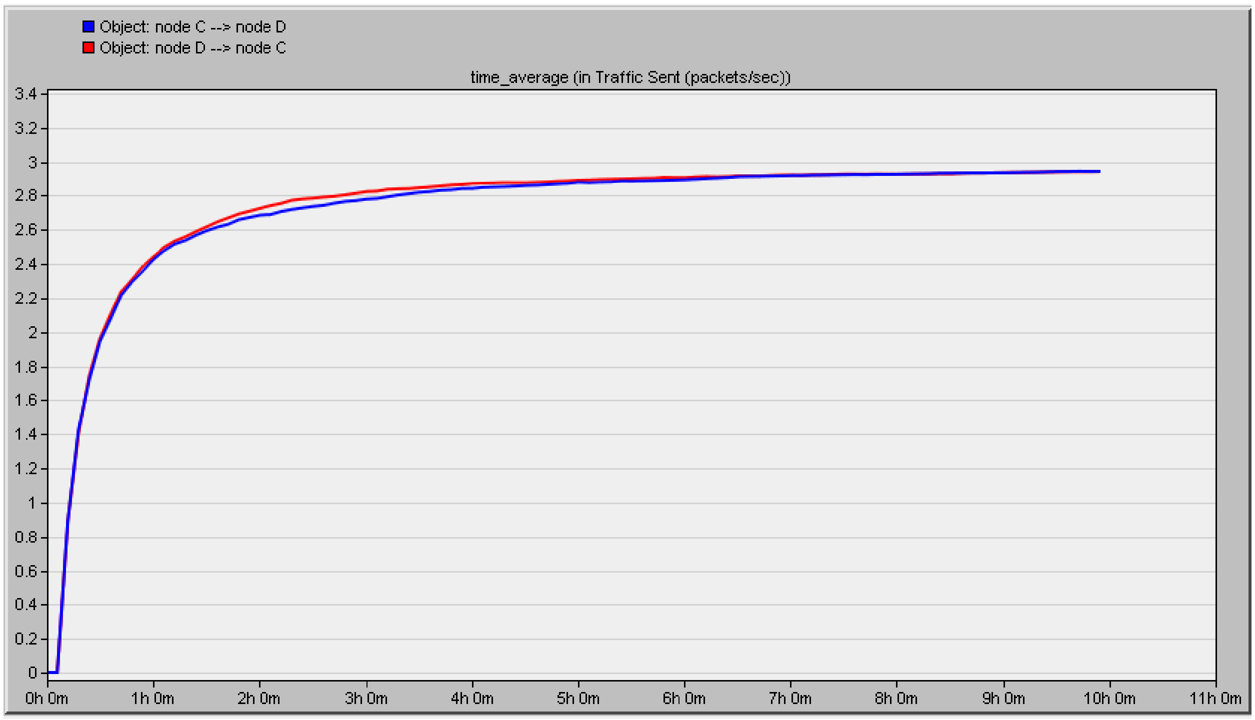


I then repeated the process for Node C to D and Node D to C.

First the histogram



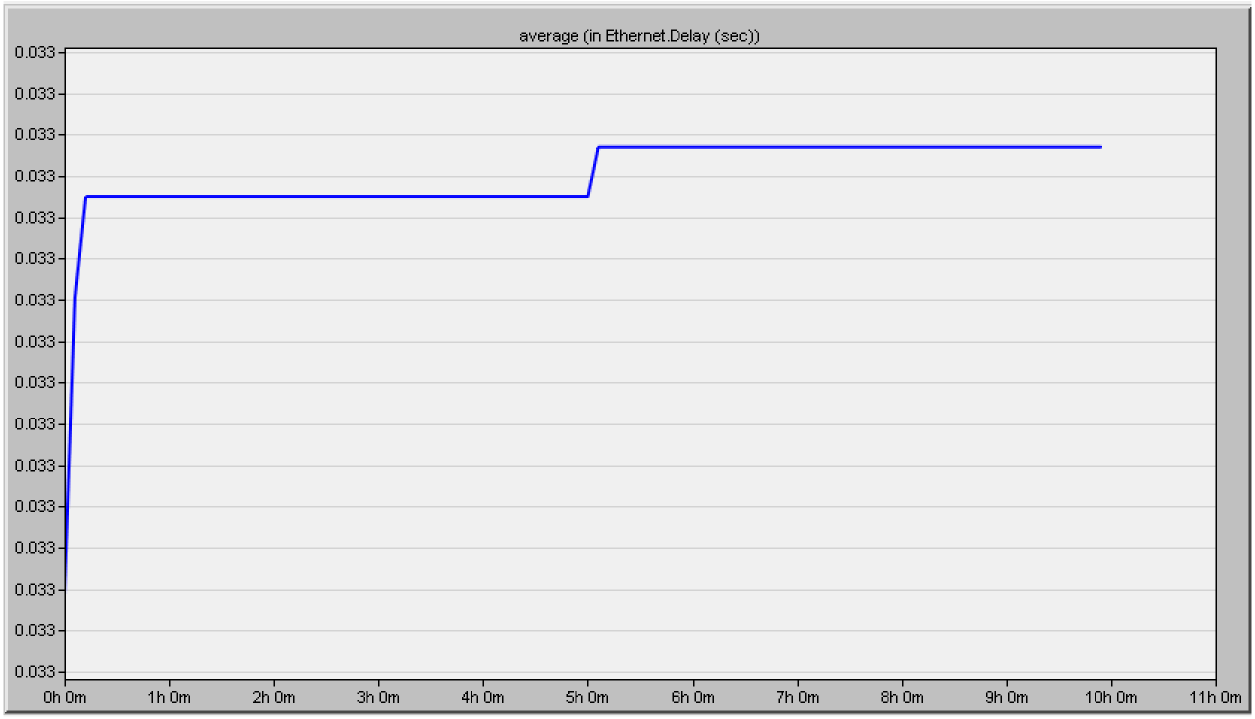
Then the time average



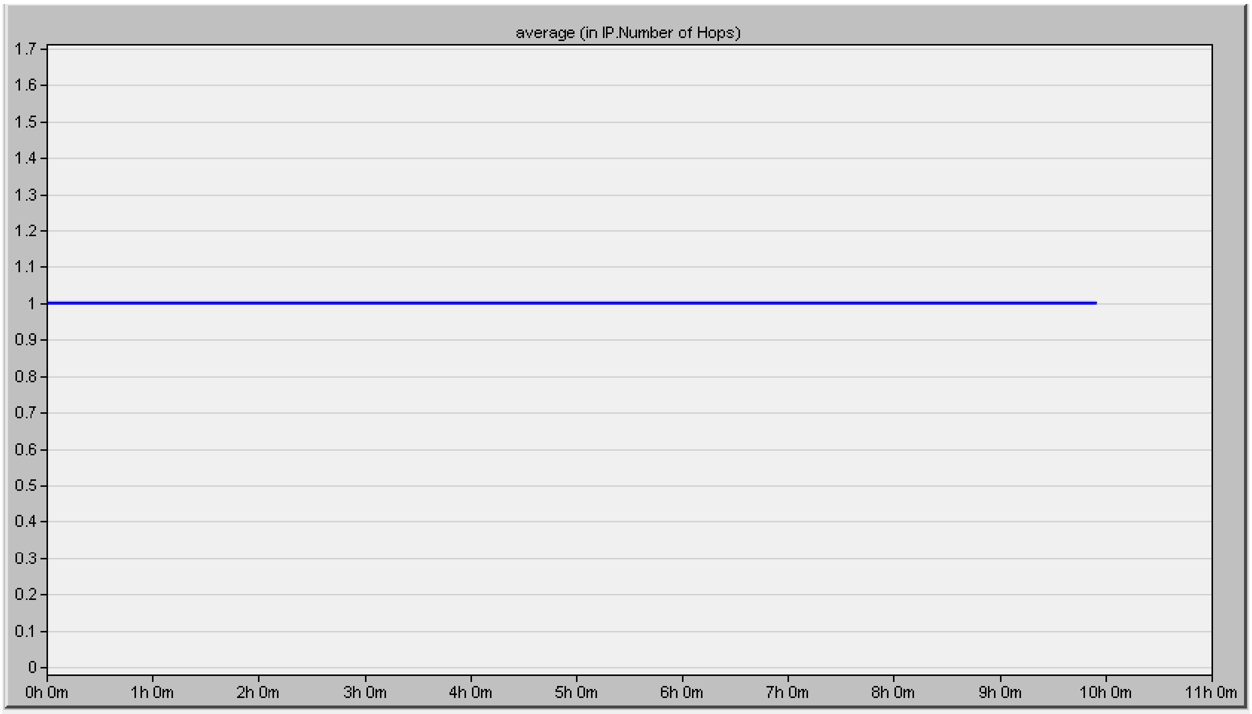
The values from the simulation exactly match the traffic matrix table. Average link flows for

Nodes B to C and C to B is 8 packets/s and average link flows for Nodes C to D and D to C is 3 packets per second.

Next, I set the simulation to display the end-to-end delay and the average number of hops.



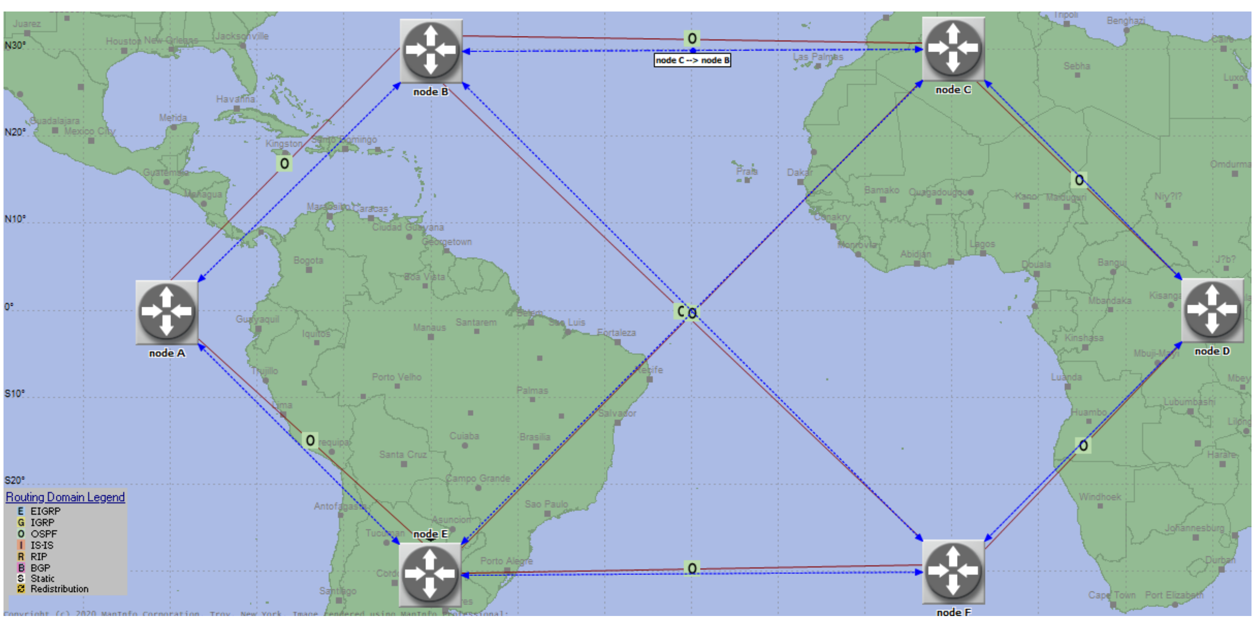
The average end-to-end delay is 0.033s.



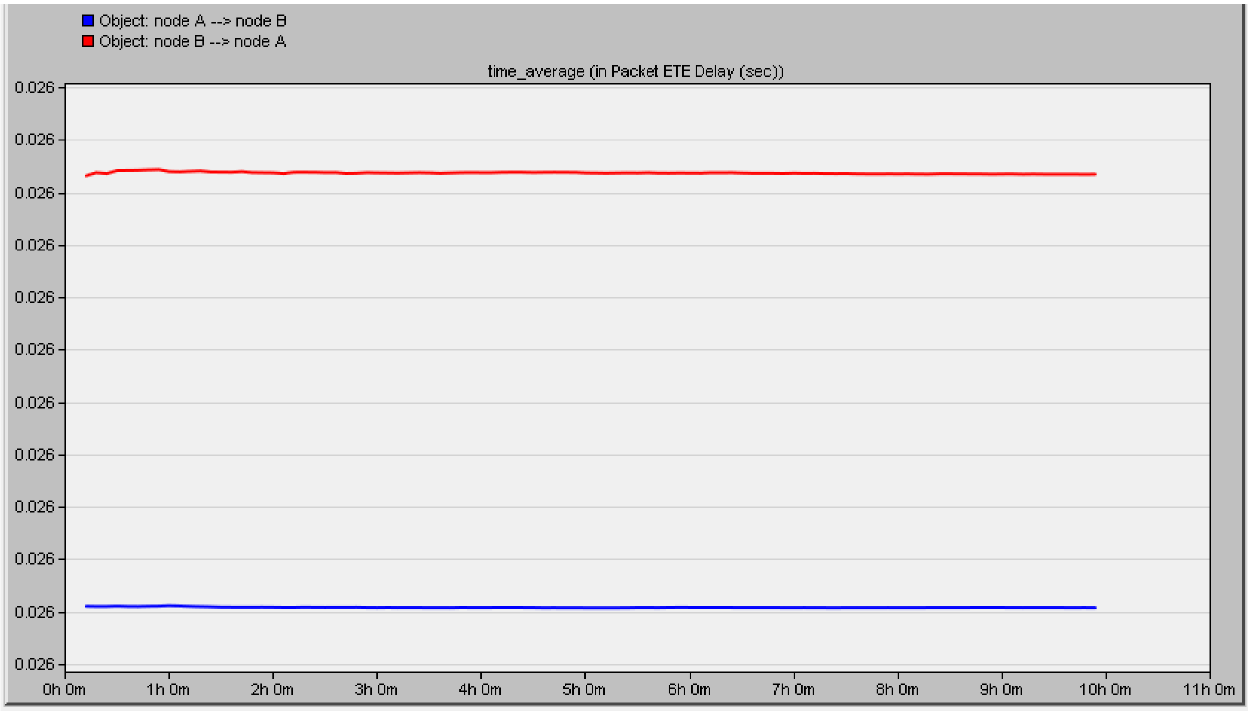
The average number of hops is 1.

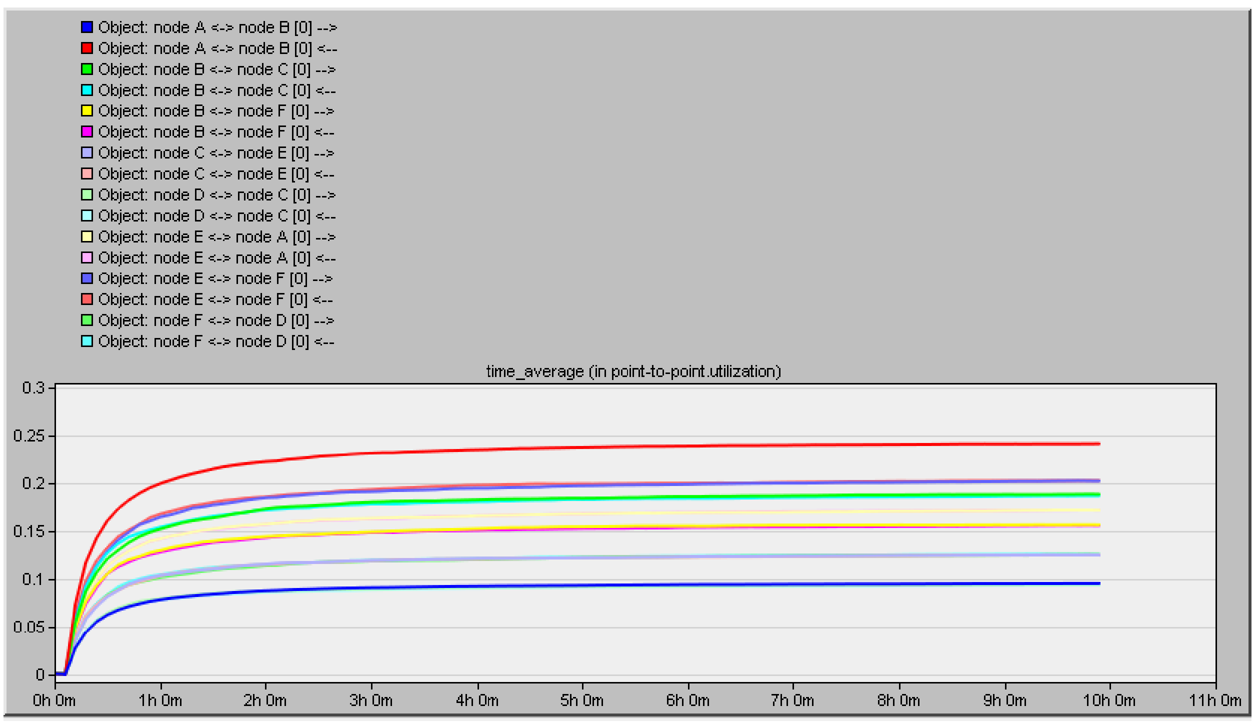
OSPF Routing Algorithm

The second routing algorithm used is the OSPF routing algorithm.



I ran the link delay simulation again to obtain the OSPF average values below for Nodes A to B and B to A. I also ran a simulation for the utilization averages for all the links.





Comparing the two methods, RIP and OSPF had about the same average delay but OSPF had much more uniform utilization values across the board and converged quicker. Overall, RIP seems to be more in line with what we are trying to achieve in this particular project as the values were much closer to the desired 0.5 utilization.