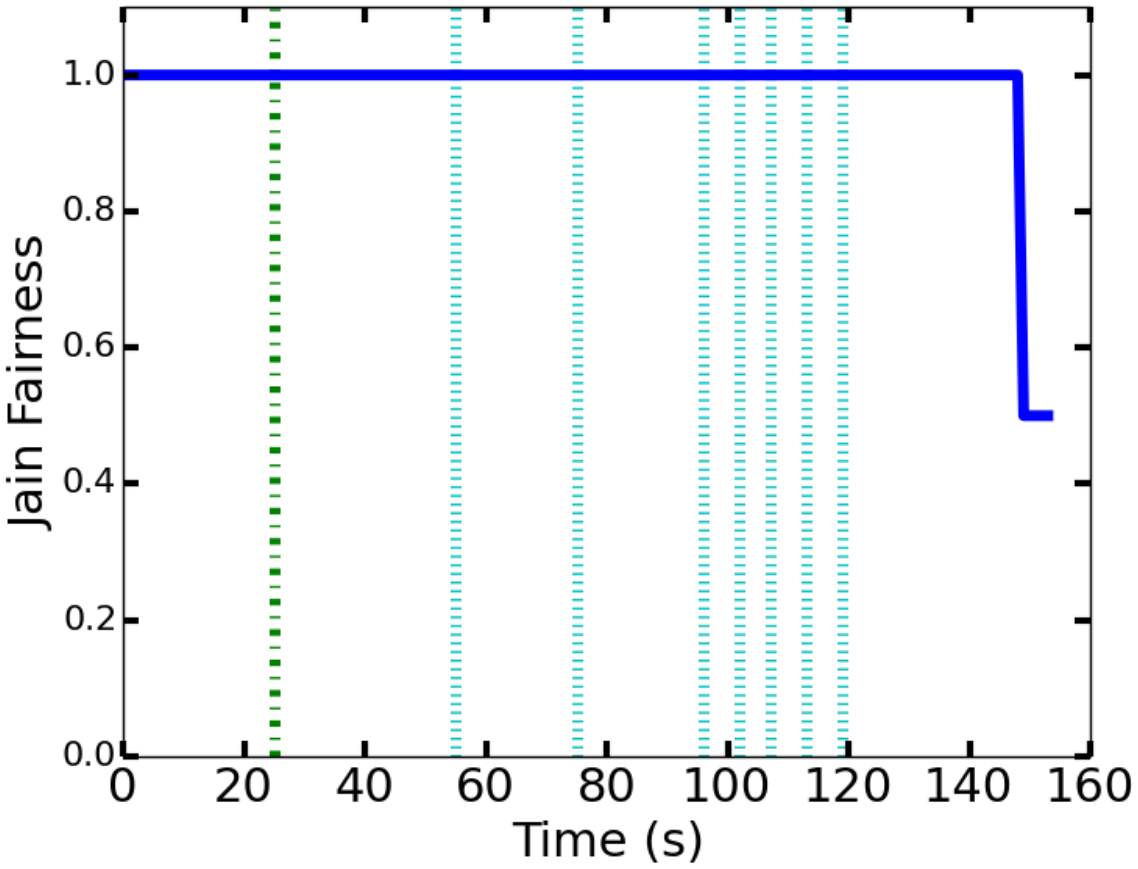
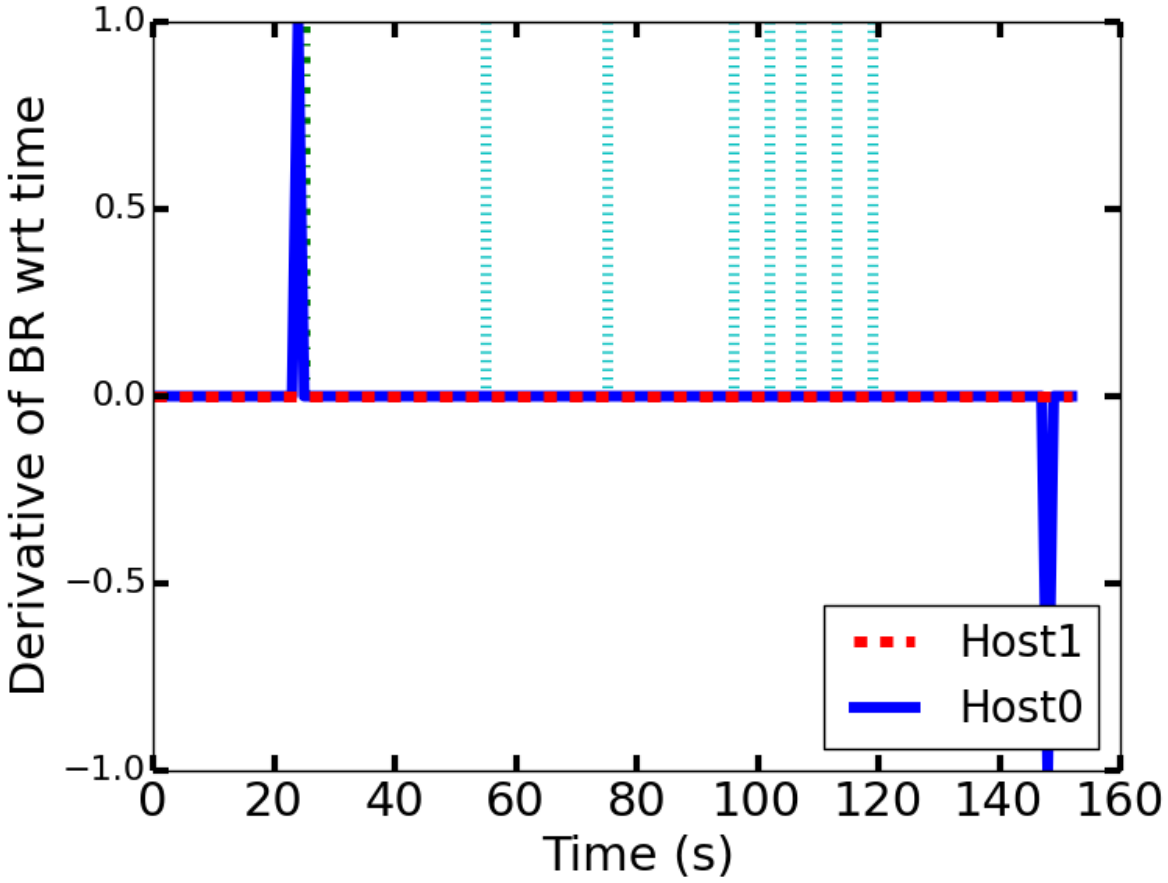
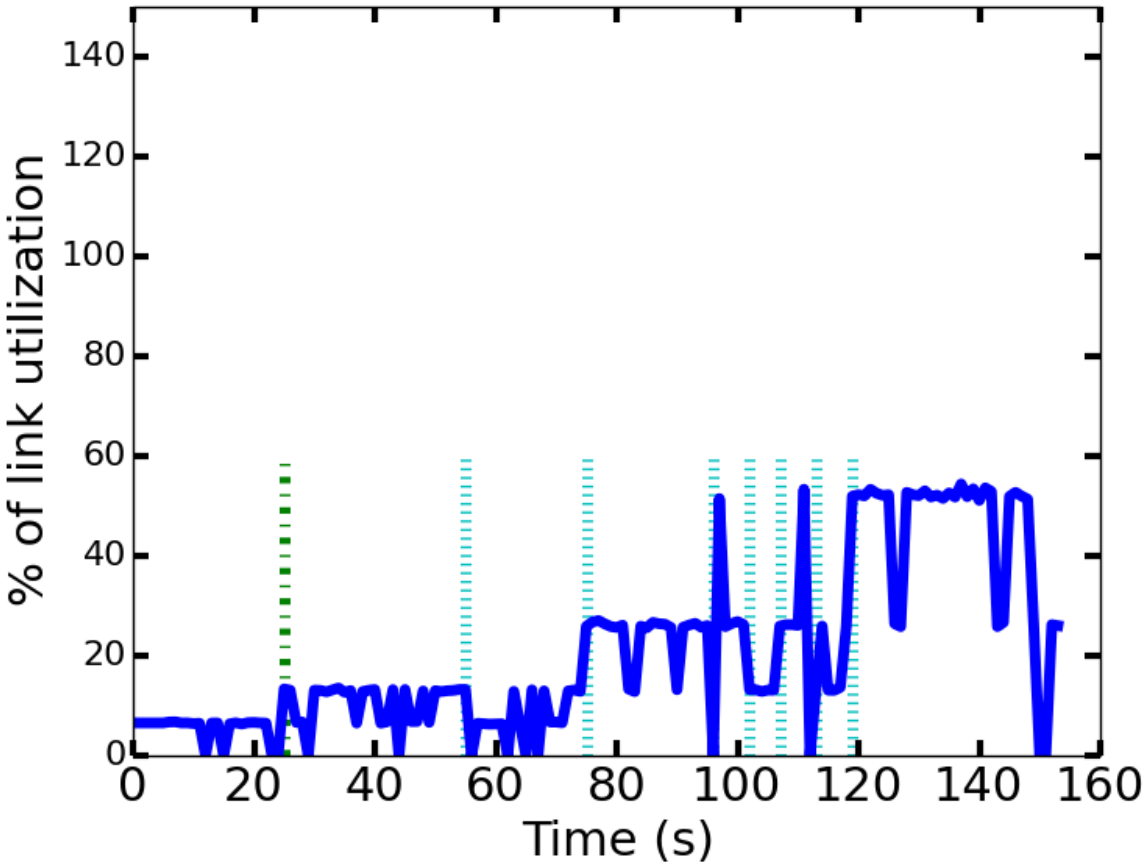
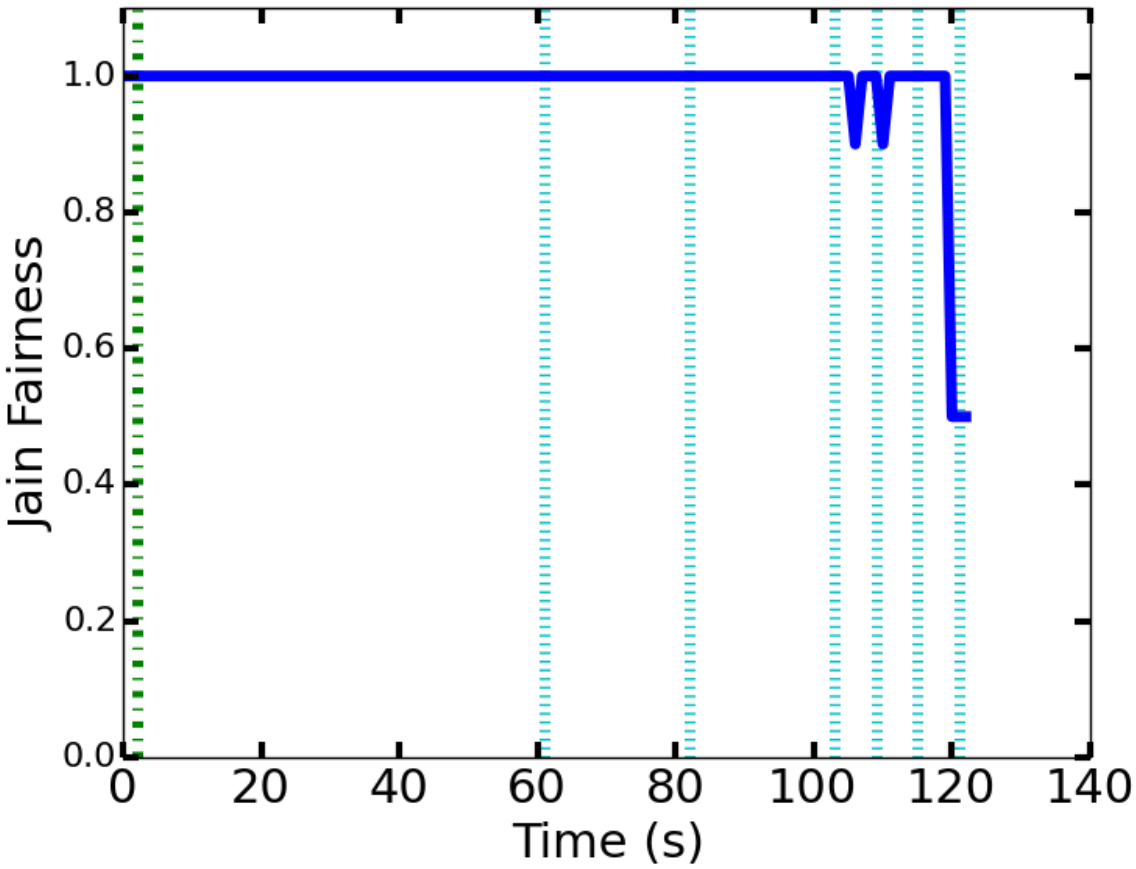
For alpha = 0.1:

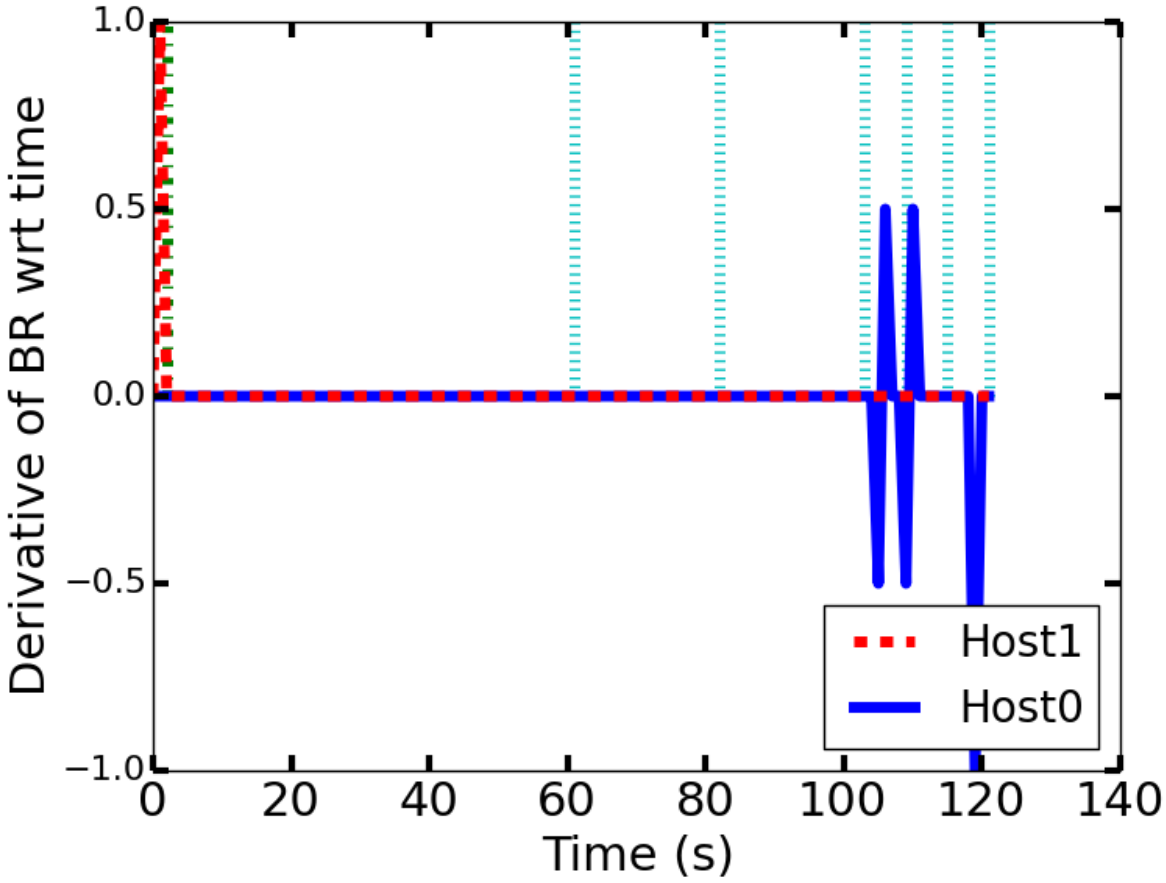

Jain Fairness
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80
100
120
140
160
Time (s)

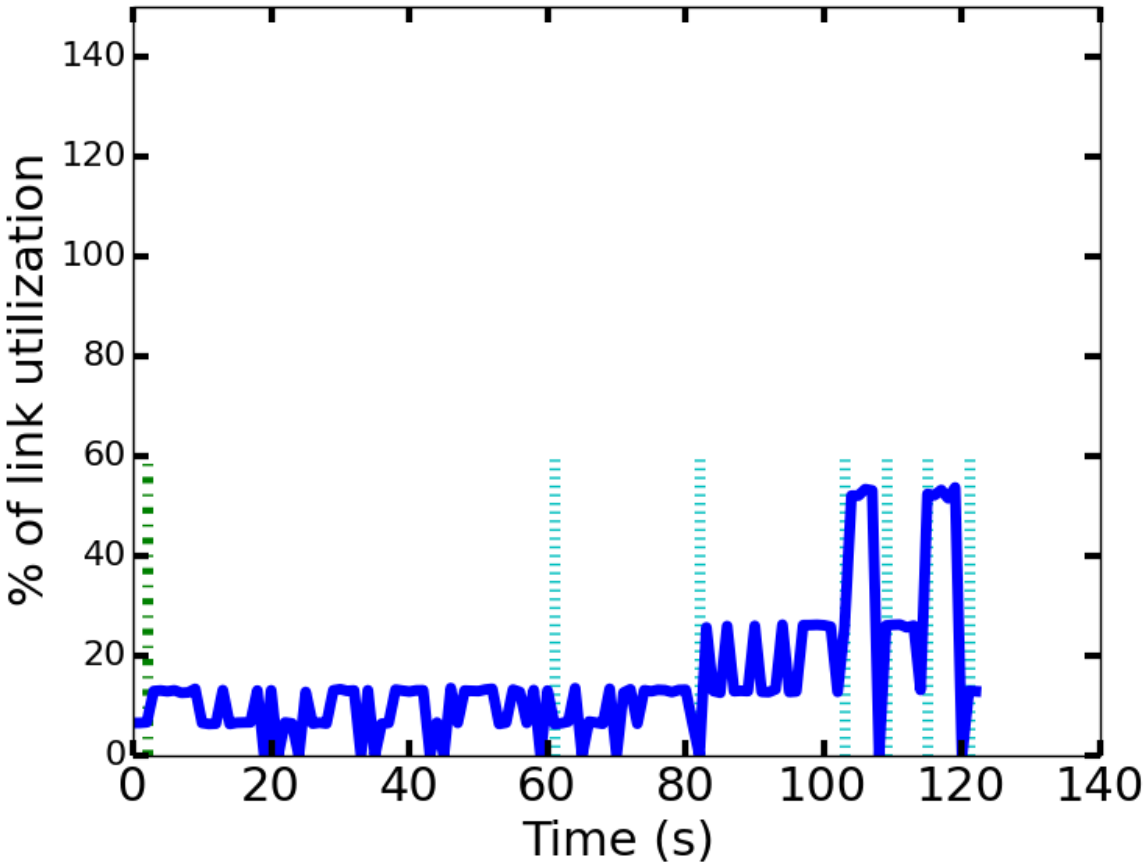

Derivative of BR wrt time
1.0
0.5
0.0
−0.5
−1.0
0
20
40
60
80
100
120
140
160
Time (s)
Host1
Host0


% of link utilization
Time (s)
0
20
40
60
80
100
120
140
160

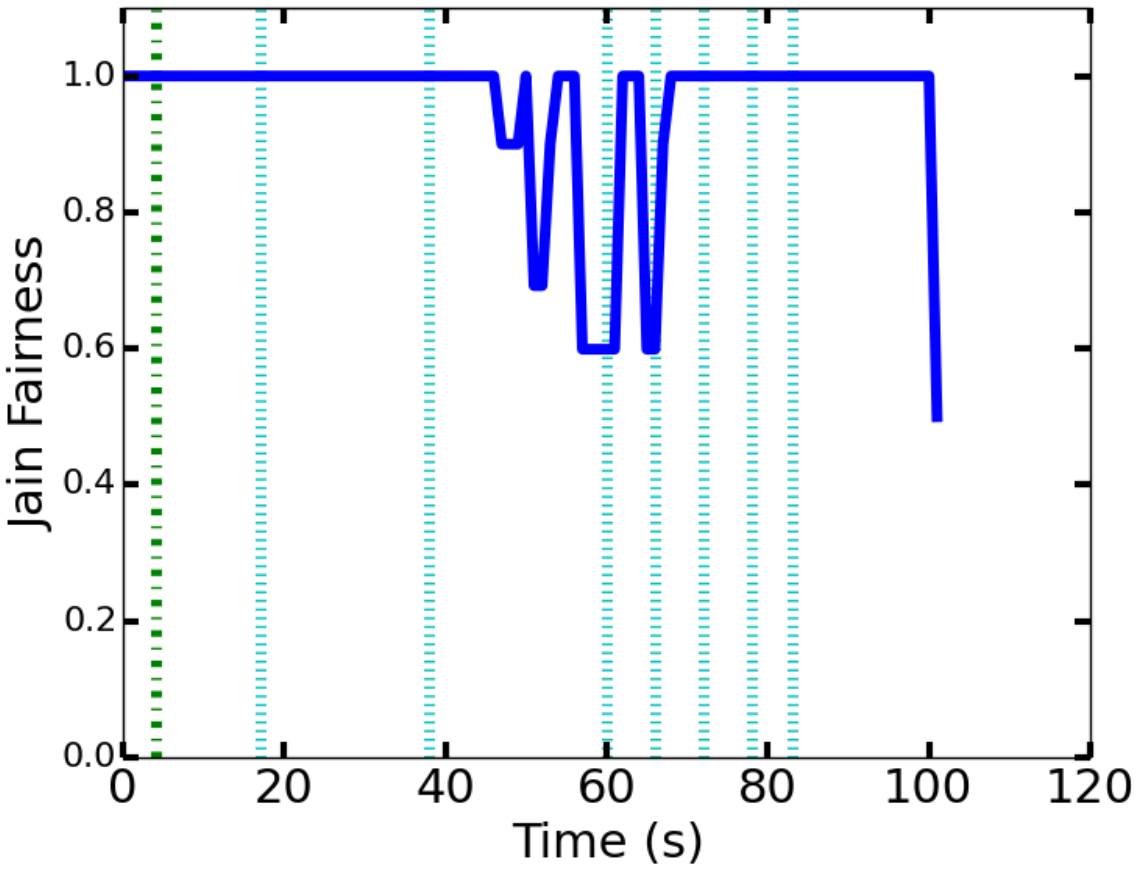
For alpha = 0.5

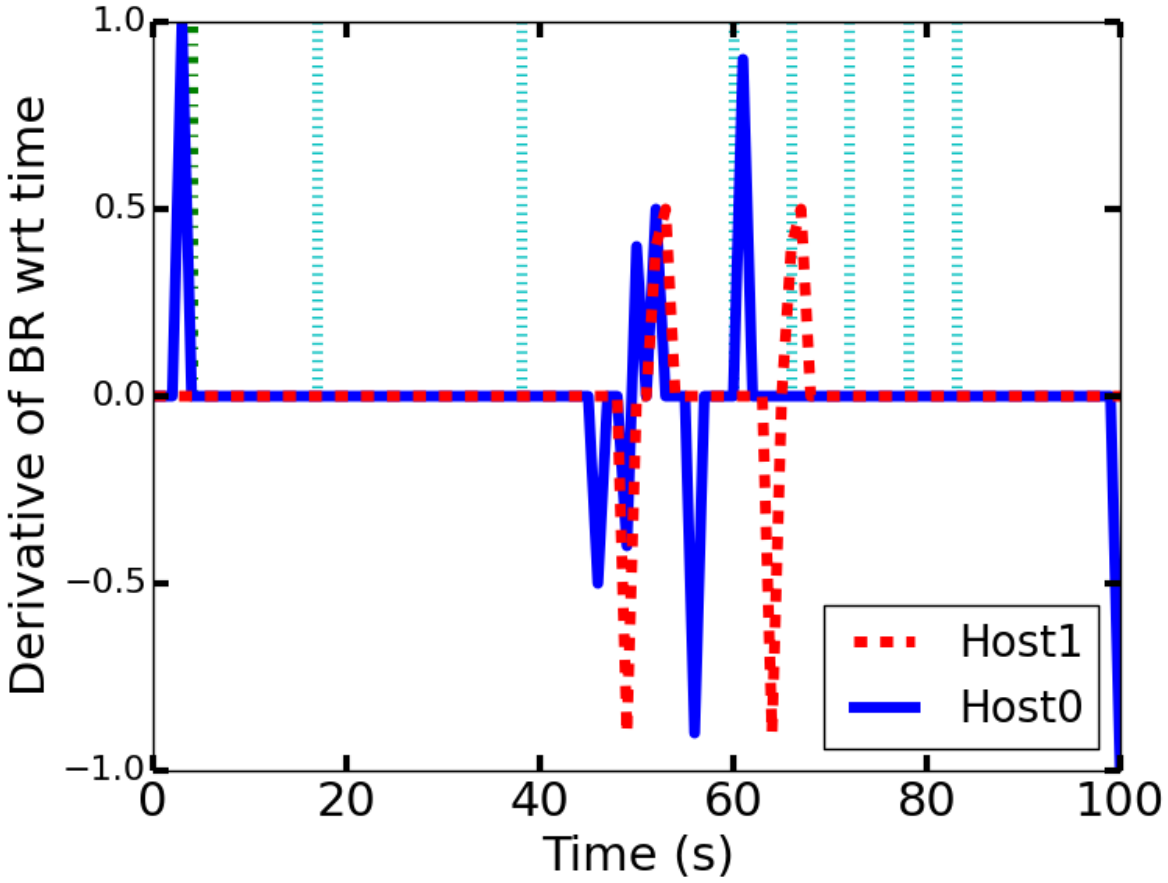


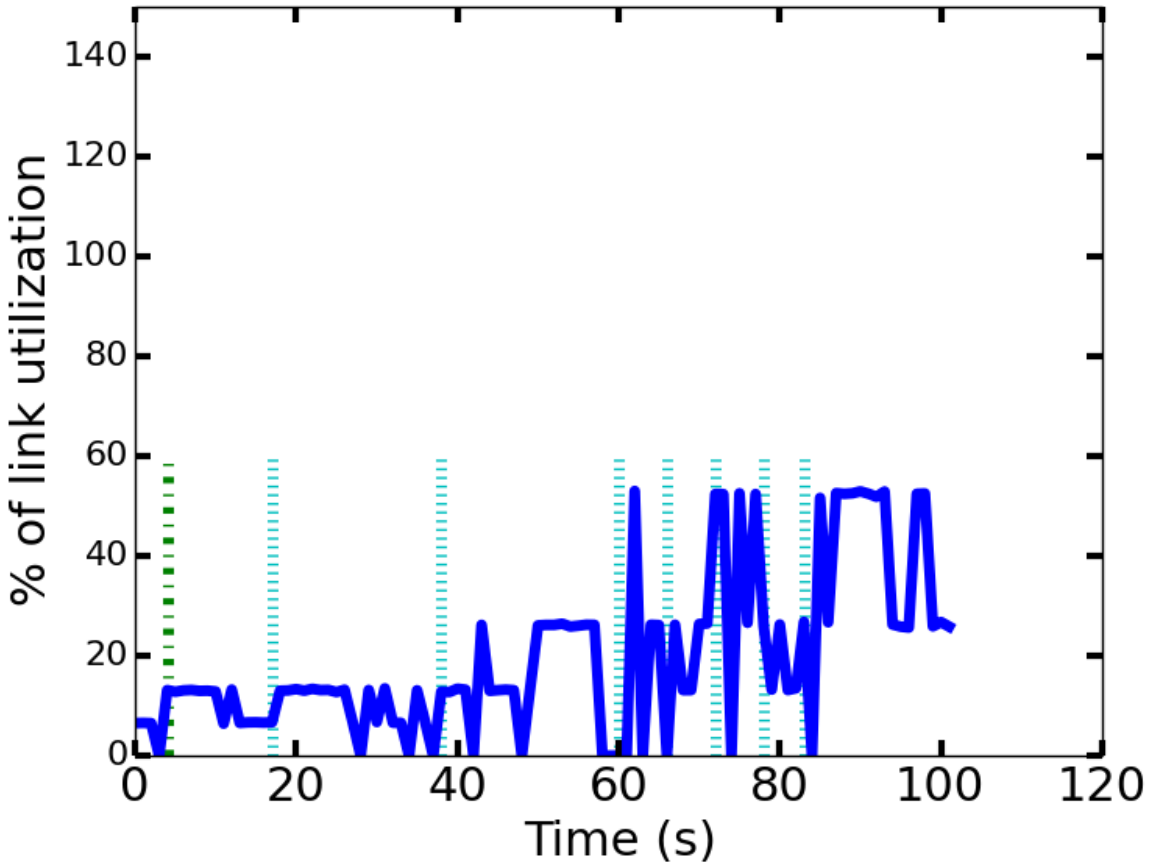

Derivative of BR wrt time
Time (s)
1.0
0.5
0.0
−0.5
−1.0
0
20
40
60
80
100
120
140
Host1
Host0


% of link utilization
Time (s)
0
20
40
60
80
100
120
140

For alpha = 0.9




Derivative of BR wrt time
Time (s)
Host1
Host0
1.0
0.5
0.0
−0.5
−1.0
0
20
40
60
80
100



There are several conditions we must consider when choosing the appropriate alpha value. If we expect to a generally steady connection with very few “spikes” or “drops” in throughput, then we would pick a low alpha value, like 0.1. That way, we avoid unnecessarily dropping the quality of the video. Of course, it would take a long time for the server to adjust if our throughput drops and stays low for a long time. It takes a very long time for the shapes of the graph to change, as seen above.

An alpha value like 0.5 gives equal importance to both previous values of the throughput as well the newly calculated one. A long-lasting drop in throughput will be noticed and adjusted fairly quickly; at the same time, it isn’t adjusted quickly enough to completely eliminate all previous throughput values. Thus, such an alpha value is a sort of “middle-of-the-road” solution. We can see that the graphs change more frequently than those of alpha = 0.1.

Lastly, an alpha value like 0.9 gives much more importance to current throughput values compared to old ones. The advantages and disadvantages, of course, are the exact opposite of that of when alpha is 0.1: we notice drops in throughput much earlier on, but we might be unnecessarily dropping the quality of the video because of a brief spike. We can that in the first

graph, the shape is much more inconsistent compared to the first two alpha values, due to the reasons just discussed.