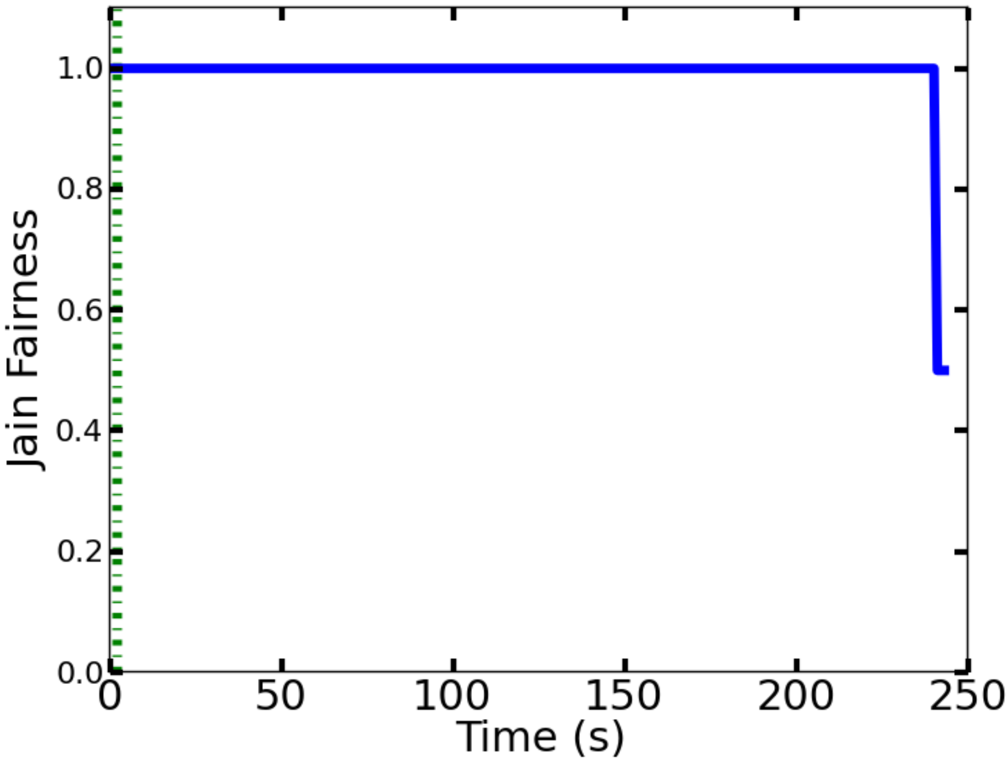
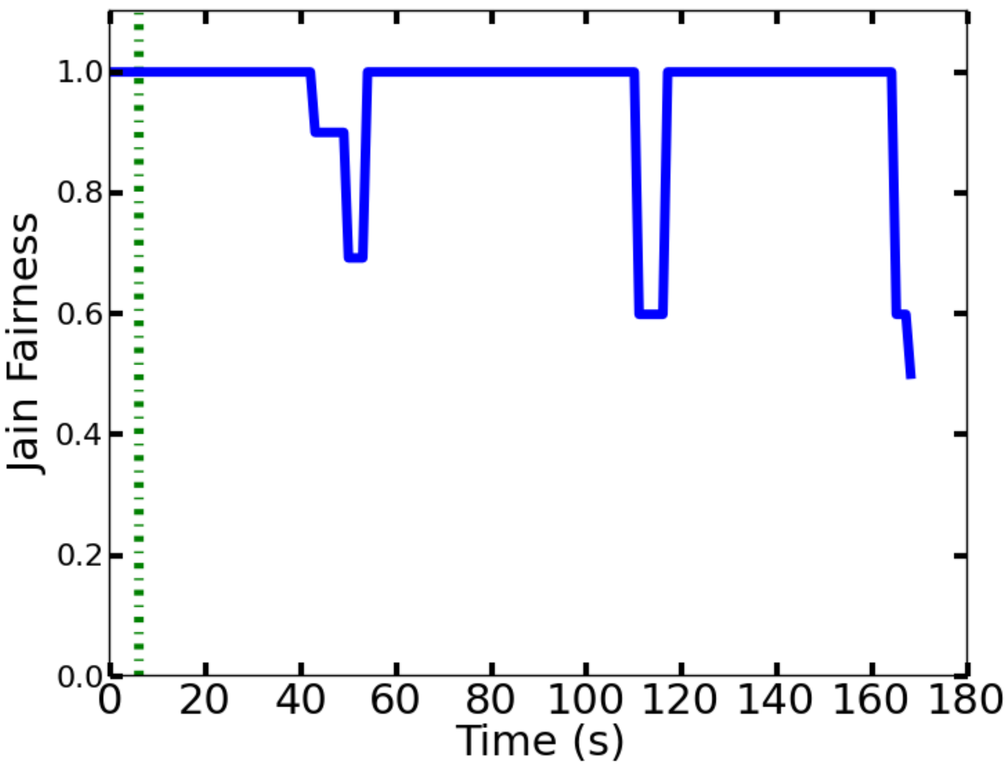
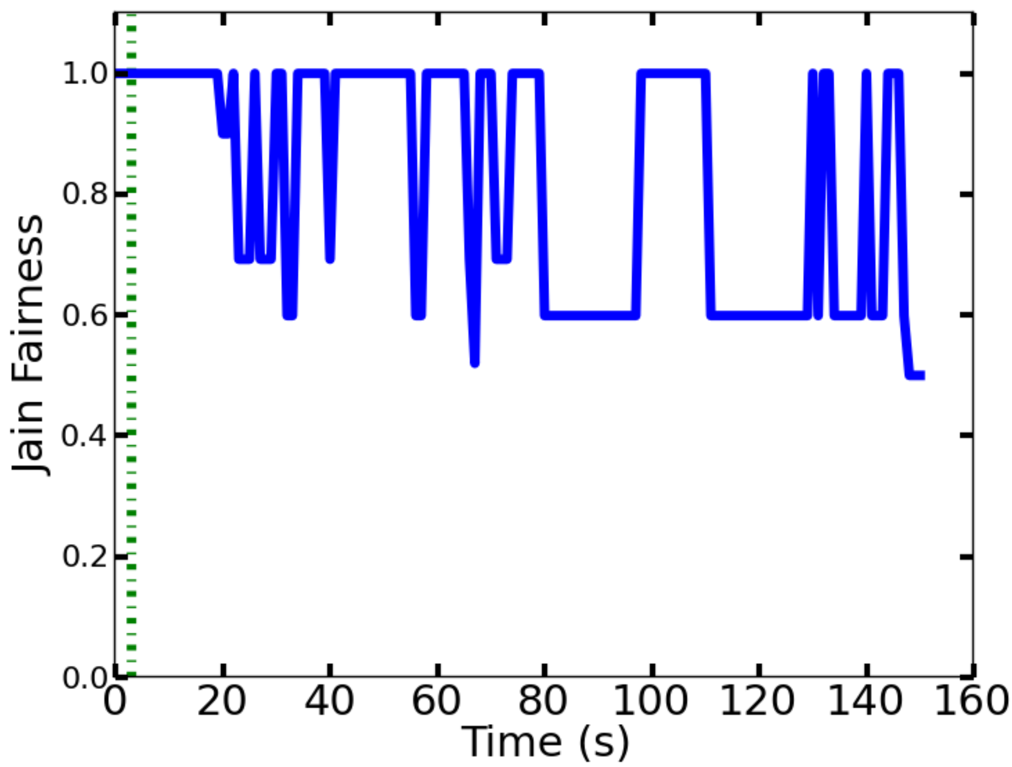
Jain Fairness:



Alpha = 0.1



Alpha = 0.5



Alpha = 0.9

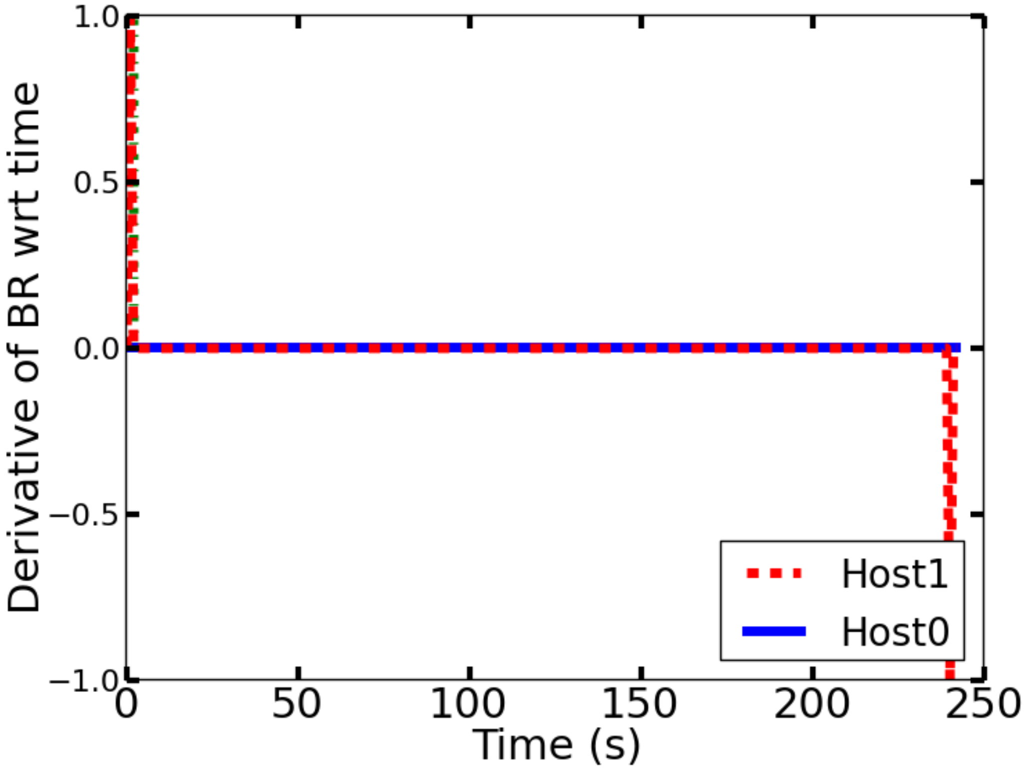
Jain Fairness is a number which reflects whether two clients are receiving a fair share of the resource from the server. The value is proportional to the difference of throughput of two clients, say if two clients are receiving the data by same or similar bitrate, the fairness will equal or close to 1. On the contrary, if two clients are receiving the data at different bitrates, the fairness will go down depends on how much the difference is.

The value of alpha does not contribute to the value fairness directly, instead, it donates the fluctuation of the fairness. For example, assume two browsers start to play the video at the same time, if the difference of throughputs of two servers are huge at the beginning (low fairness), with a small alpha, the fairness may continuous to keep a low value, in other words, host receiving fast will keep receiving and host receiving slow will keep receiving slow. But if both two hosts are fairly receiving at the beginning (high fairness), both of host may highly possible to keep a fair receiving in the following time, no matter events occurred or not (low alpha makes two host event-resist, both hosts would like to keep their former pace).

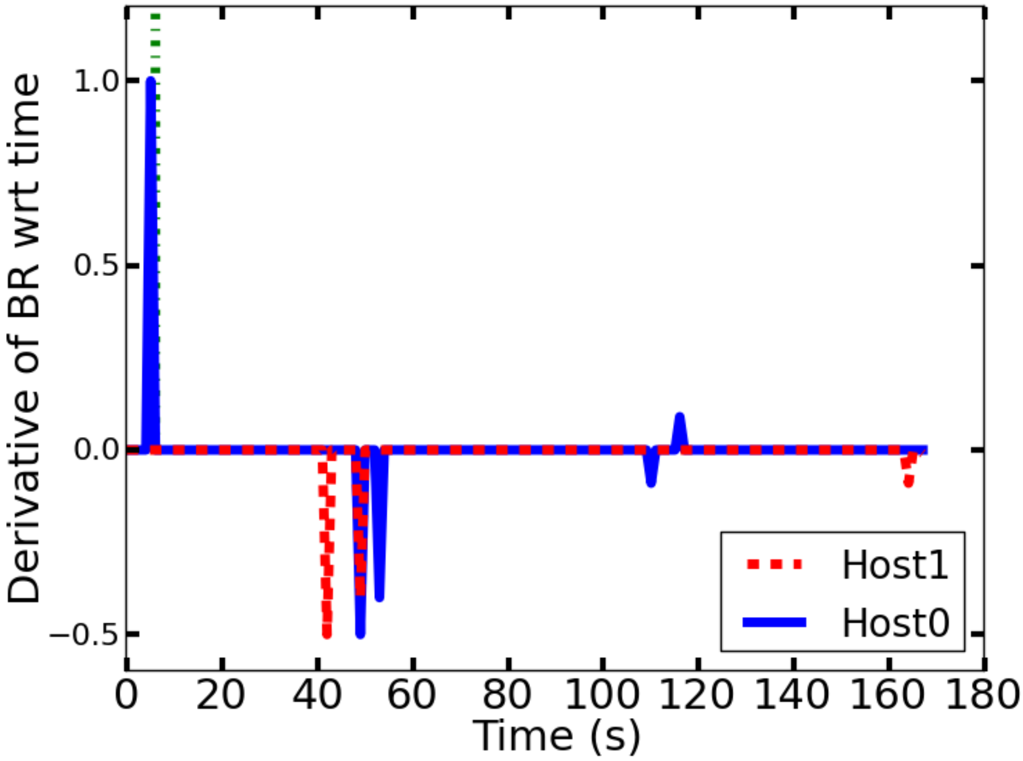
On the contrary, with a big alpha, the fairness may fluctuate a lot. Back to the previous example, if there is a big difference of throughput at the beginning (low fairness), the curve is expected to fluctuate a lot since high alpha makes two hosts more sensitive to events, both two hosts cannot keep their former pace when events occurred.

Therefore, low alpha makes the fairness curve smoother, no matter what the start-points is. High alpha makes more fluctuation, two hosts may shorten or broaden the throughput difference when events occurred.

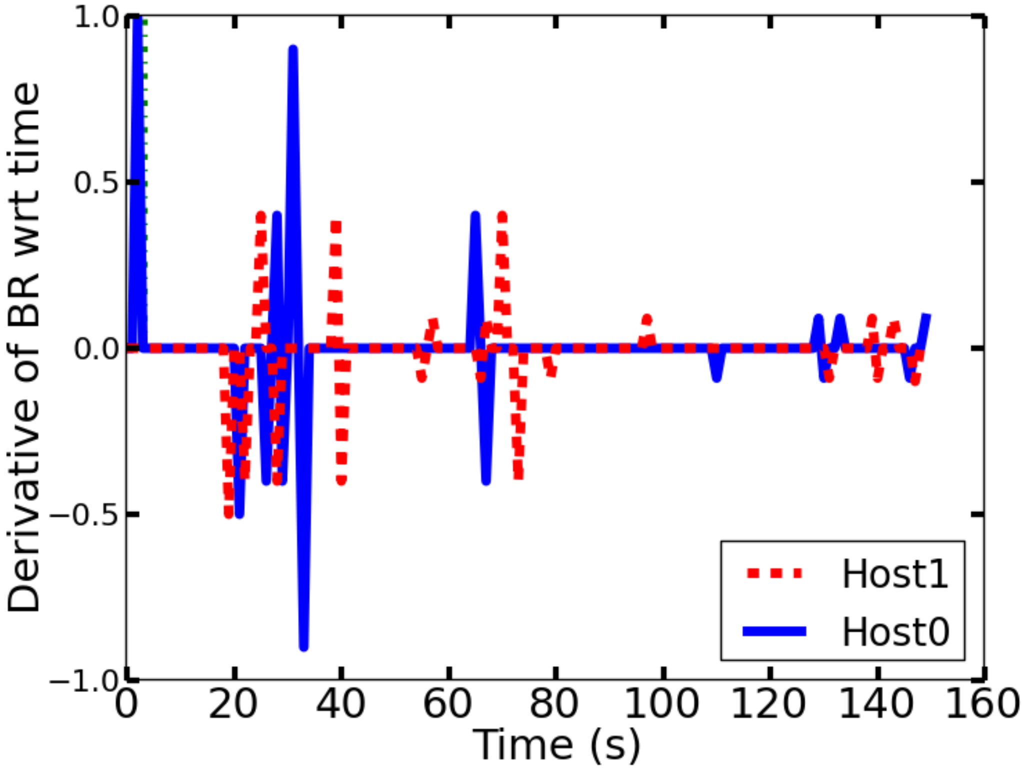
Smoothness:



Alpha = 0.1



Alpha = 0.5

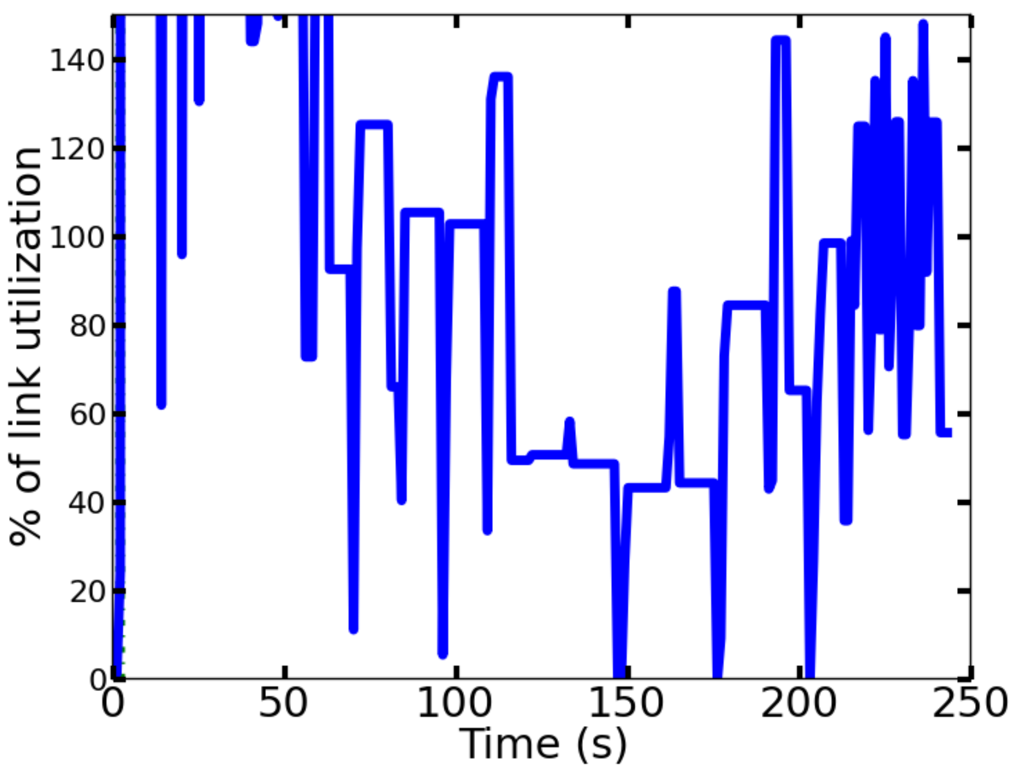


Alpha = 0.9

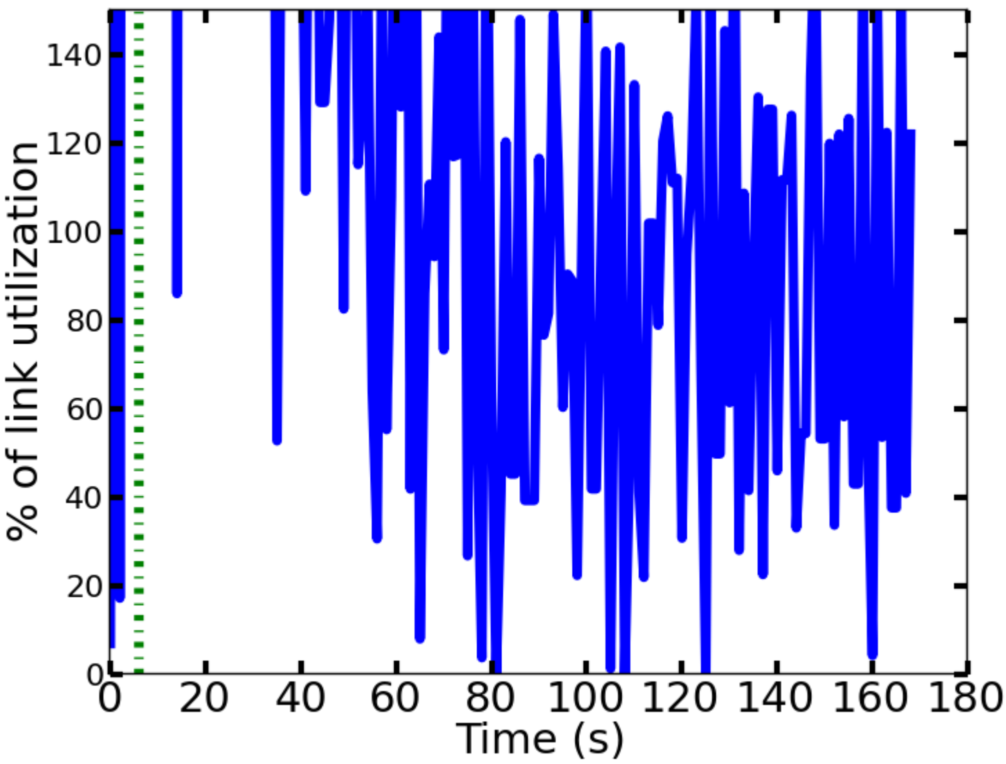
The above three graphs represent the smoothness where alpha = 0.1, 0.5 and 0.9 respectively. When alpha = 0.1, the graph is the most smoothness one and alpha = 0.9 results the least smoothness one.

The smoothness of a graph depends on the variation of throughput. When throughput varies heavily, the graph is expected to fluctuate dramatically with sharp spikes over a short time, and vice versa. Based on the definition of the throughput, when alpha is close to zero, the overall average throughput donates the current throughput. On the contrary, when alpha is close to one, the bitrate of last received chunk donates the current throughput. Therefore, when events occurred, the curve (alpha = 0.9) is excepted to fluctuated more and the curve (alpha = 0.1) is excepted to fluctuate less. The results match the expectation.

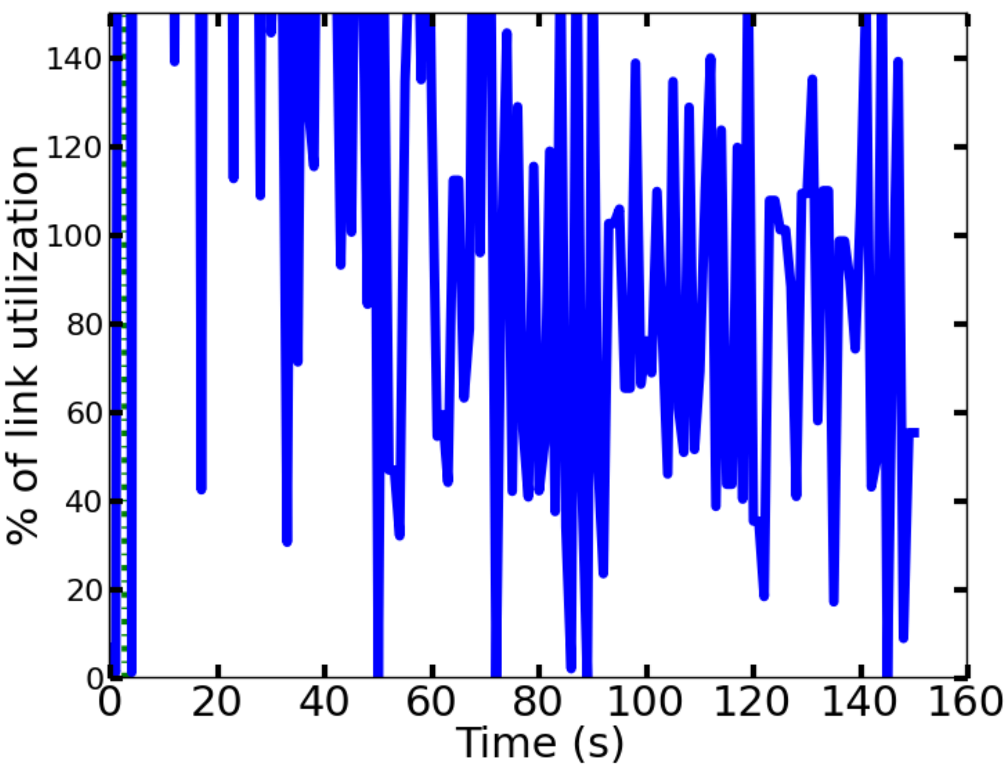
Utilization:



Alpha = 0.1



alpha = 0.5



Alpha = 0.9

The above three graphs represent the percentage of utilization where alpha = 0.1, 0.5 and 0.9 respectively.

The utilization is proportional to the value of alpha, that is, higher alpha has larger utilization and vice versa. The main reason is that higher alpha can adjust the throughput bitrate better than lower alpha because higher alpha makes throughput_new donates the throughput_current so that the proxy has more flexibility to adjust the throughput.