

ABSTRACT

In the network-based Software Defined Network (SDN), the number of users will boost the data rate needs to access various services. On the other hand, SDN network with many users will lead to a link over load which will result in a collision or even packet loss. Therefore we need a load balancing method on SDN to set the load on the path from the source node to the destination node. On this load balancing method is expected to have a good quality of service and can anticipate their excessive strain on one of the links.

In this Final Project, simulated the implementation of load balancing on the SDN. The tools that will be used is Mininet as data plane, Floodlight as control plane along with iperf and D-ITG as tools for analysis. The topology used is the Abilene topology with the data flowed for testing in the form of data and video streaming packet.

The results of SDN network performance testing using load balancer, QoS values including delay, jitter, throughput, and packet loss in the scenario without link termination meet the ITU-T standard G.1010 while for packet loss in the link termination scenario the data packet testing does not meet the standards ITU-T G.1010. The average delay generated by testing the data packet with load balancer has a value of 26.55% better than without the load balancer, in the link termination scenario, the delay value with load balancer has a 40.79% better, as well as 9.97% better on video streaming and 39.91% better in the link termination scenario. The average throughput generated by testing data and video streaming packet with load balancing is 17.55 Mbps and 15.68 Mbps, and 17.47 Mbps and 15.47 Mbps in link termination scenarios, throughput without load balancer 17.5 4Mbps and 15.69 Mbps, and 17.41 Mbps and 15.33 Mbps in the link termination scenario. The average link recovery time in testing data and video streaming packet with load balancer is 18.22 ms and 16.76 ms, while without load balancer, it is 20.16 ms and 19.42 ms.

Keywords: Load Balancing, SDN, Floodlight, Mininet