

SDN Performance Analysis^{*}

Vincenzo Imperati[†]

September 6, 2022

^{*}github.com/VincenzoImp/SDN-Performance-Analysis

[†]imperati.1834930@studenti.uniroma1.it

Contents

1	Used Technologies	3
2	Remote Controller Used	3
3	Topology v1	3
3.1	Observations	4
3.2	Results	4
4	Topology v2	6
4.1	Observations	6
4.2	Results	7
	References	11

1 Used Technologies

Mininet APIs[1] were used to create custom SDN topologies, Ryu[2] was used for topology management, D-ITG[3] was used to create network flows. The data observed for the performance evaluation were extrapolated thanks to ryu (bandwidth utilization) and D-ITG (end to end delay and packet drop).

2 Remote Controller Used

The remote controller used (for both analyzed topologies) redirects the packets on the shortest path, assuming all links have the same weight. This assumption is true due to the fact that all links of the examined topologies have the same performance.

3 Topology v1

The Topology v1 is characterized by four hosts and three switches connected as shown in **Figure 1**. All the links have the same performance (bandwidth = 20 Mbits, delay = 20 ms, max_queue_size = 10). The flows generated (UDP packets) in the network for the performance study are unidirectional and are the following: (H_1, H_3) , (H_1, H_4) , (H_2, H_3) and (H_2, H_4) . The streams transmit packets with the same lambda rate. The performance measurements were made with lambda rate starting from 2 pkt/s and increasing by 2 pkt/s with each new measurement until reaching a lambda rate of 200 pkt/s. The lambda rate had a Poissonian distribution and the observation time was set at 20 s.

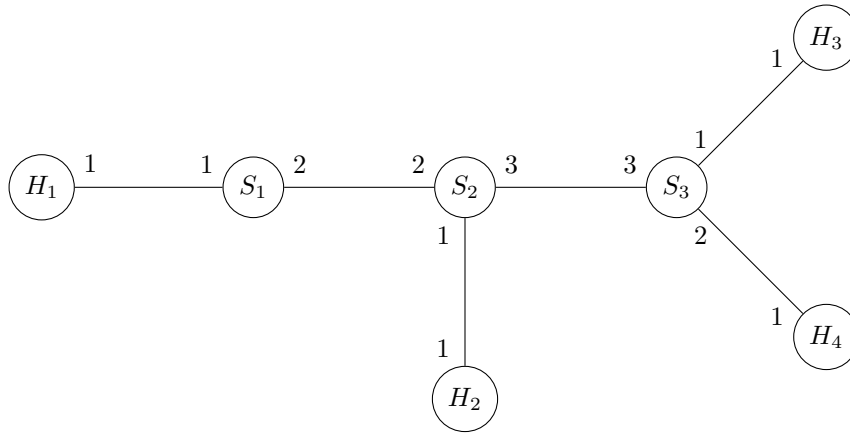


Figure 1: Topology v1

3.1 Observations

Note that packet streams follow the following paths:

- $path(H_1, H_3)$: $H_1 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow H_3$
- $path(H_1, H_4)$: $H_1 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow H_4$
- $path(H_2, H_3)$: $H_2 \rightarrow S_2 \rightarrow S_3 \rightarrow H_3$
- $path(H_2, H_4)$: $H_2 \rightarrow S_2 \rightarrow S_3 \rightarrow H_4$

Since the flows have the same lambda rate and transmit the same packets we can calculate how many distinct flows pass on a specific link and therefore also the unit cost that passes on each link:

- $cost(H_1, S_1)$: $|\{(H_1, H_3), (H_1, H_4)\}| = 2$
- $cost(H_2, S_2)$: $|\{(H_2, H_3), (H_2, H_4)\}| = 2$
- $cost(H_3, S_3)$: $|\{(H_1, H_3), (H_2, H_3)\}| = 2$
- $cost(H_4, S_3)$: $|\{(H_1, H_4), (H_2, H_4)\}| = 2$
- $cost(S_1, S_2)$: $|\{(H_1, H_3), (H_1, H_4)\}| = 2$
- $cost(S_2, S_3)$: $|\{(H_1, H_3), (H_1, H_4), (H_2, H_3), (H_2, H_4)\}| = 4$

It is possible to believe that H_1 packets take a third longer than H_2 packets due to the longer path they travel. It is also possible to foresee that the link (S_2, S_3) increases its utilization doubly compared to the other links.

3.2 Results

As lambda increases, three graphs have been produced showing correspondingly: bandwidth utilization ([Figure 2](#)), end-to-end latency on average ([Figure 3](#)), and packet drop ([Figure 4](#)).

From the [Figure 2](#) we notice that the utilization of the link (S_2, S_3) grows twice as fast as the others. It is also noted that the links (H_3, S_3) and (H_4, S_3) reach their maximum utilization more slowly due to the packets dropped by the link (S_2, S_3) , which is the one subject to lose more packages than all. We also note that with lambda rate 60 pkt/s we saturate the bottleneck utilization, that is link (S_2, S_3) .

From the [Figure 3](#) we notice that packets sent from H_1 take a third longer than those sent from H_2 .

Finally, from the [Figure 4](#) we note that for lambda rates greater than 100 pkt/s the packet drop grows linearly.

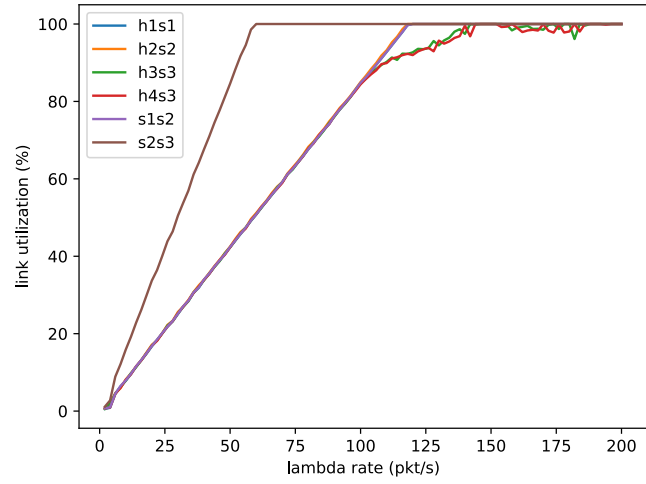


Figure 2: Links Utilization Topology v1

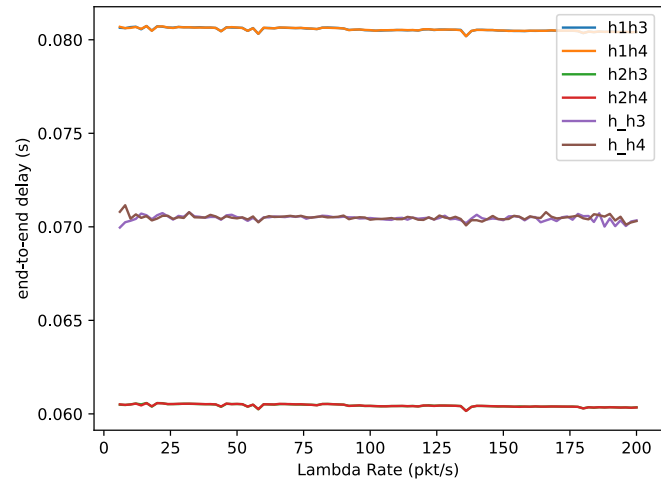


Figure 3: End-to-End Delay Topology v1

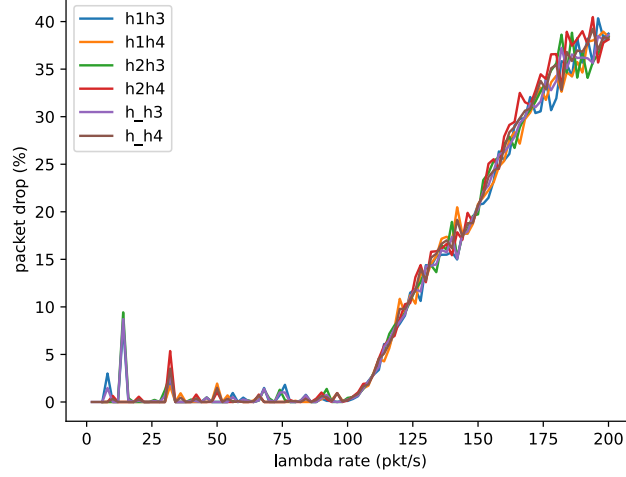


Figure 4: Packet Drop Topology v1

4 Topology v2

A new topology (Topology v2) has been studied with the aim of starting from the Topology v1, adding a new link, re-studying the direction of flows and finally studying its performance in order to be able to make a comparison with the Topology v1. So it was decided to update the Topology v1 with the addition of the link (S_1, S_3) , keeping the performance of all the links (including the new one) equal to those of the Topology v1. The exact same misurations made on Topology v1 were then carried out. Topology v2 is shown in [Figure 5](#).

4.1 Observations

The packet streams follow the shortest path, so the paths are the following:

- $path(H_1, H_3): H_1 \rightarrow S_1 \rightarrow S_3 \rightarrow H_3$
- $path(H_1, H_4): H_1 \rightarrow S_1 \rightarrow S_3 \rightarrow H_4$
- $path(H_2, H_3): H_2 \rightarrow S_2 \rightarrow S_3 \rightarrow H_3$
- $path(H_2, H_4): H_2 \rightarrow S_2 \rightarrow S_3 \rightarrow H_4$

Also for the Topology v2 we can calculate how many distinct flows pass on a specific link and therefore also the unit cost that passes on each link:

- $cost(H_1, S_1): |\{(H_1, H_3), (H_1, H_4)\}| = 2$
- $cost(H_2, S_2): |\{(H_2, H_3), (H_2, H_4)\}| = 2$

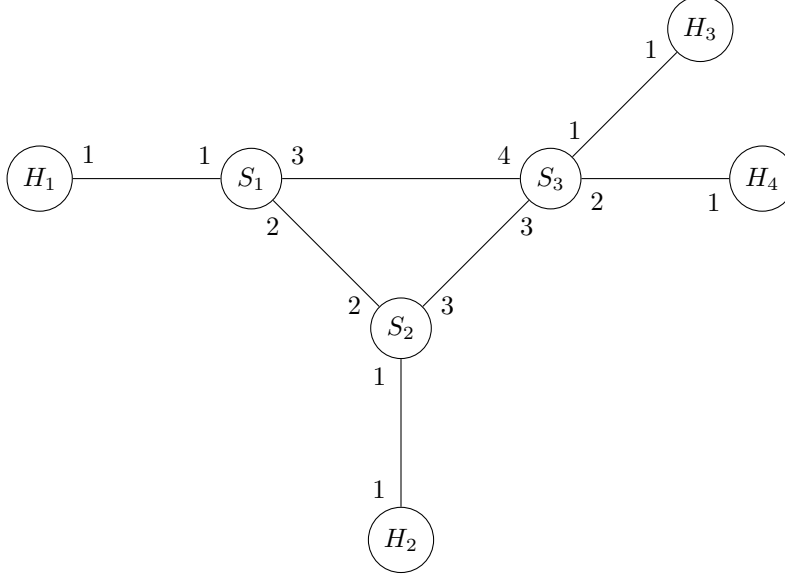


Figure 5: Topology v2

- $cost(H_3, S_3)$: $|\{(H_1, H_3), (H_2, H_3)\}| = 2$
- $cost(H_4, S_3)$: $|\{(H_1, H_4), (H_2, H_4)\}| = 2$
- $cost(S_1, S_2)$: $|\{\}| = 0$
- $cost(S_2, S_3)$: $|\{(H_2, H_3), (H_2, H_4)\}| = 2$
- $cost(S_1, S_3)$: $|\{(H_1, H_3), (H_1, H_4)\}| = 2$

By observing that all paths are equal in length, it is possible to predict that all packets transmitted by H_1 will take the same time as those transmitted by H_2 . Note that $cost(S_2, S_3)$ has been reduced to 2, and therefore since for each link a maximum of two distinct flows pass through, it is safe to believe that there is no link that is used more than the others causing bottleneck.

4.2 Results

Also for the Topology v2, the following three graphs were generated as lambda rate increased: bandwidth utilization (Figure 6), end-to-end latency on average (Figure 7), and packet drop (Figure 8).

From the Figure 6 we note that the use of all links grows linearly in the same way, except for the link S_1, S_2 which is not used for packet routing. We also note that with lambda rate 120 pkt/s we saturate the utilization of all the useful links.

From the [Figure 7](#) we notice that all packets are the same latency, while from the [Figure 8](#) we note that in the range of lambda rates studied, there aren't relevant packet drop.

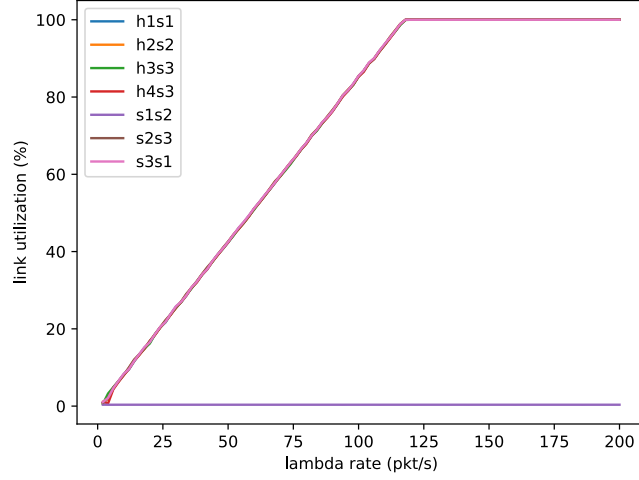


Figure 6: Links Utilization Topology v2

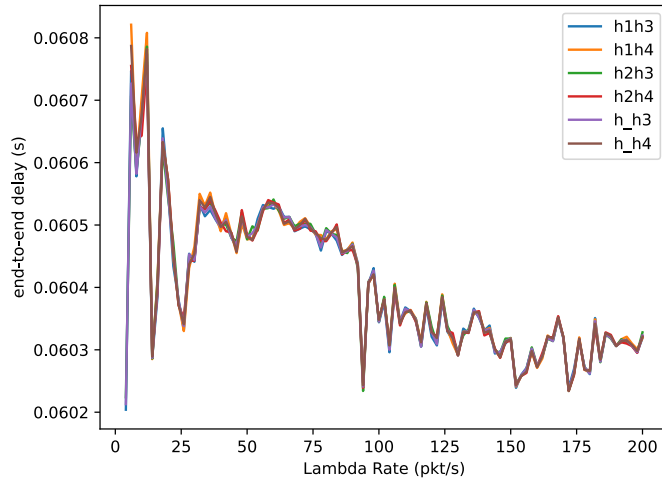


Figure 7: End-to-End Delay Topology v2

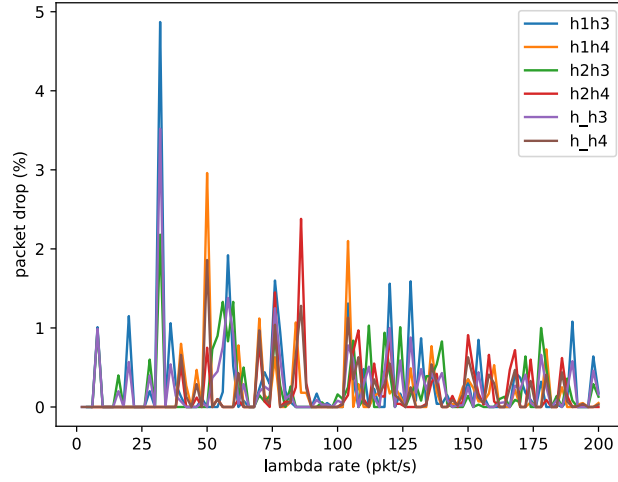


Figure 8: Packet Drop Topology v2

At the end we extended the lambda rate range for the Topology v2 simulation to see when we obtain an increase of packet drop. So new graph was created: bandwidth utilization (Figure 9), end-to-end latency on average (Figure 10), and packet drop (Figure 11). For lambda rates greater than 240 pkt/s the packet drop grows linearly.

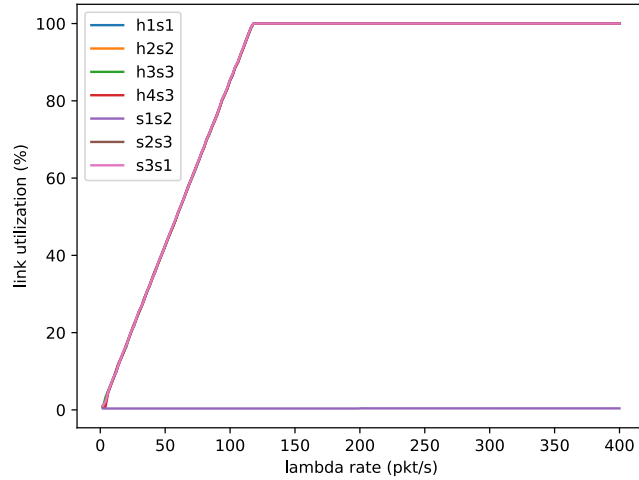


Figure 9: Links Utilization Topology v2

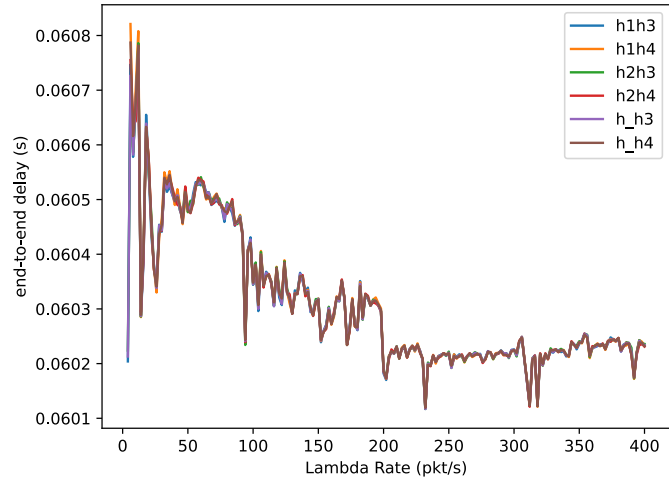


Figure 10: End-to-End Delay Topology v2

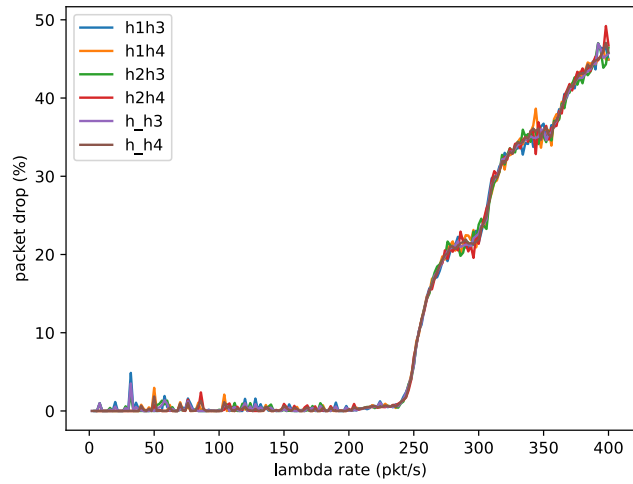


Figure 11: Packet Drop Topology v2

References

- [1] *Mininet: Rapid Prototyping for Software Defined Networks*. URL: <https://mininet.org/>.
- [2] *Ryu: SDN Framework*. URL: <https://ryu.readthedocs.io/>.
- [3] *D-ITG: Distributed Internet Traffic Generator*. URL: <https://traffic.comics.unina.it/software/ITG/>.