Software Defined Networking Project

Project Report

Group F
Project 3 – Access Control

Authors:

Giuseppe Gattulli - giuseppe.gattulli@mail.polimi.it

Manuele Montrasio - manuele.montrasio@mail.polimi.it

Borroni Luigi - luigi2.borroni@mail.polimi.it

Date: 19/07/2021

Specification

The controller can manage all network topologies, where it must be aware of all host addresses, otherwise packets will be dropped. Some features of the project are useless for network topologies that do not have rings.
No new switches (static network) should be added or removed to the network to be managed.
Managed traffic is TCP/UDP between client and server.

The primary goal is to route new incoming packets to the switches, through the insertion of rules, using the lowest cost path search and block any flows on links where the number of active flows is too high, by setting a limit on them on the link. The only links that don't have any boundaries are those that connect switches and hosts.

The secondary goal is to delete inactive flows on links when the rule timeout expires, making the link available again in the virtual graph.

Architecture

The controller intercepts the following events:

- **Switch Features**: The controller asks the switch to send all packets to it;
- **Packet In (TCP/UDP)**: The controller builds the graph for the path search at minimum cost by checking whether or not the link has reached the maximum number of flows (3), in this case the link is not inserted in the graph construction and sends to the switch 2 rules containing a timeout: one to search for the match on the flow table related to the link on which you want to route the packet; the other on the flow table for the link for routing the packet. In addition, the counter for the link on which the packet is routed is incremented;
- **Packet In (ARP):** The controller executes the ARP proxy that generates an ARP reply, based on the ARP tables known to the controller;
- **Flow Removed**: The controller decrements the flow counter on the link for the Flow Removed event.

Since the *flow removed function* manages the table_id parameter ，we decided to create rules for the construction of flow tables specific to each connection, so as to be able to manage the counters on the various flows.

Description of the demonstrator

In the demonstration the operation of the primary objective is shown first and then that of the secondary objective, it can therefore be considered divided into two parts:

Primary objective: TCP/UDP connections are generated between client and server, in order to reach the flow limit on a certain link (in our case 3). The link is then blocked to other incoming streams, deleting it from the virtual graph. At this point, the new flows will be routed on paths other than the primary minimum path because it is blocked. Connections between switches and hosts are excluded from these operations.

Secondary objective: once the flows on the primary path are completed, the packet routing rules are eliminated from the switches at the end of the timeout so as to make the link virtually available by decreasing the value of the counter. A new TCP/UDP flow is then generated to demonstrate that the link has been reactivated.

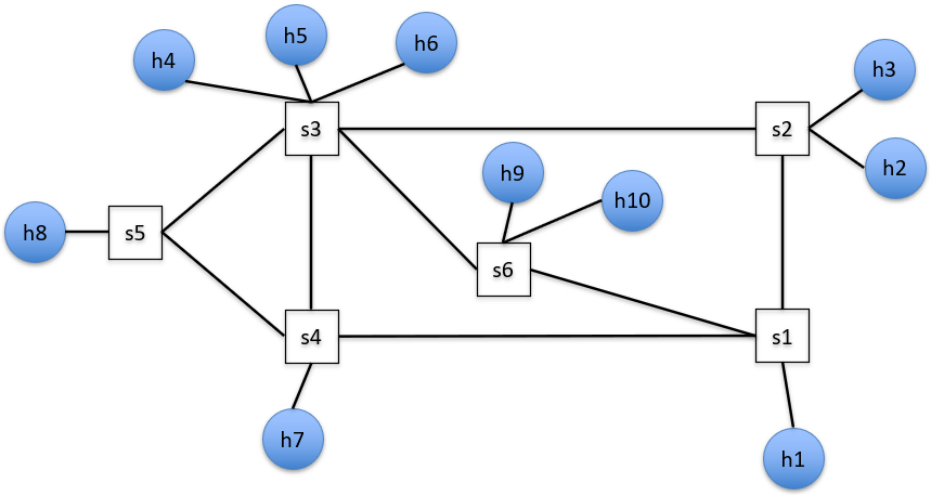
Automated testing: An initial pingall is done to allow the controller to know all the MAC addresses of the hosts and the hosts to populate their own ARP tables. After that, the two servers are created, the first of which is UDP h2 and the second is TCP h3.
Three connections are then established:

- h5 to h2 UDP type 100 seconds long
- h6 to h3 of TCP type 100 seconds long and consequent ACK response flow
- h4 to h2 UDP type 100 seconds long

in this way it is possible to see that the link s2-s3 reaches its limit of active flows (3) so the fourth stream (the UDP flow from h4 to h2) follows the path s3-s4-s1-s2 or s3-s6-s1-s2. A sleep action is then performed that lasts for 20 s and a new UDP stream from h4 to h2 is created to prove that the s2-s3 link is active again since the previous flows were terminated.

The bandwidth of the hyperf executed is limited only to obtain more readable graphs on the InMon mininet dashboard (Sflow), but they have no other utility for the operation of the project.

Topology

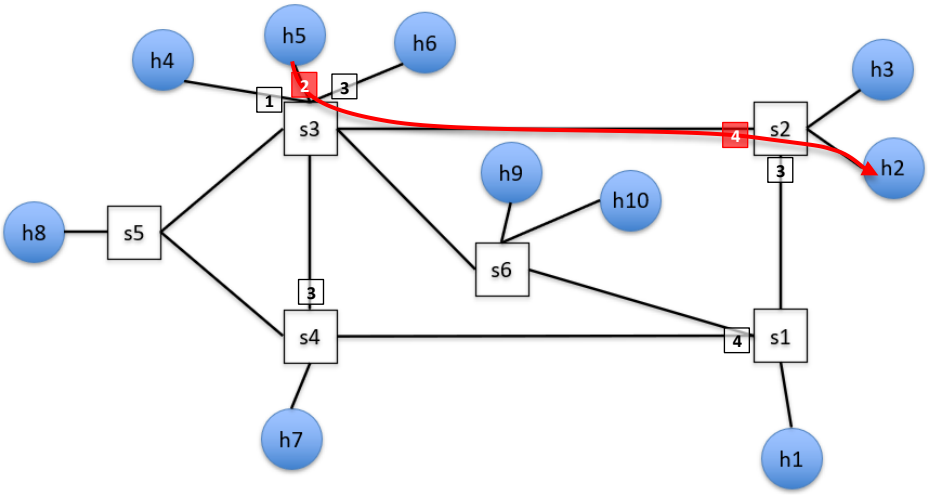


Performance Measurement

Thanks to the graph provided by the inMon dashboard mininet (sflow) we can observe the interfaces of the switches with the highest bitrate values entering the interface at that instant.

1st Stream Generation (UDP):

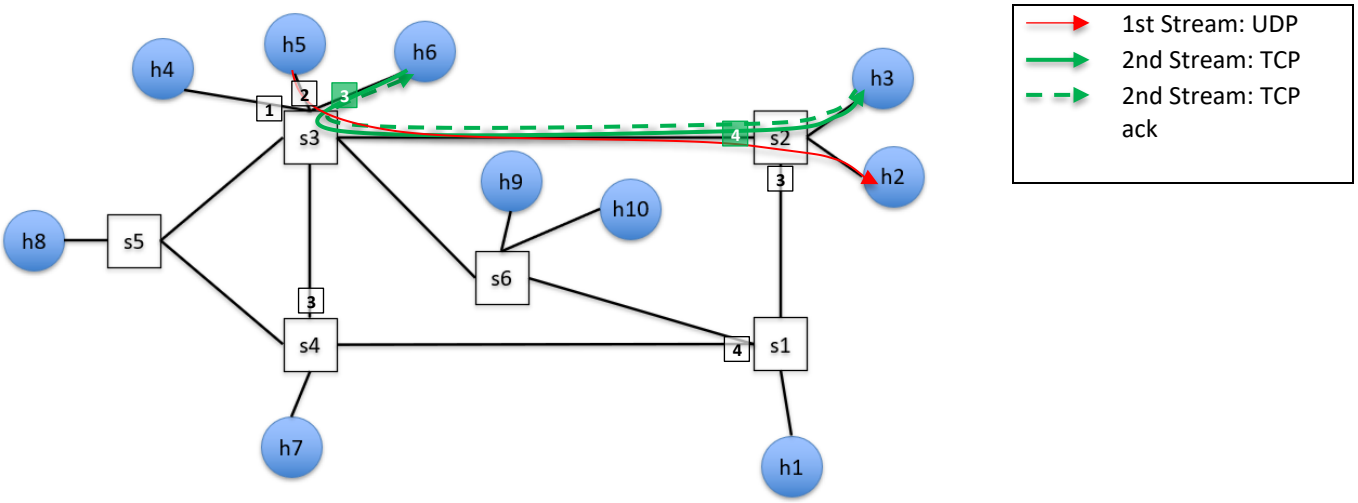
UDP flow from h5 to h2 is generated



Generation of the 2nd stream (TCP):

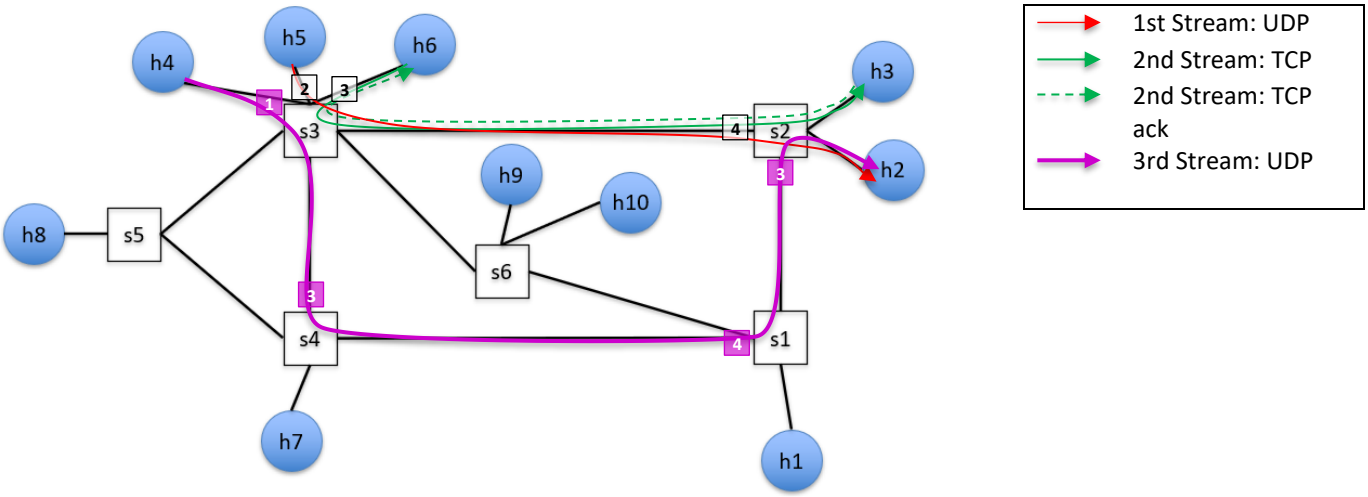
A TCP stream is generated from h6 to h3 and a response stream from the server at h3.

The TCP response is low bandwidth usage because the acknowledgment acks have fewer bytes than the forward TCP stream, so this stream is not clearly visible from the graph.



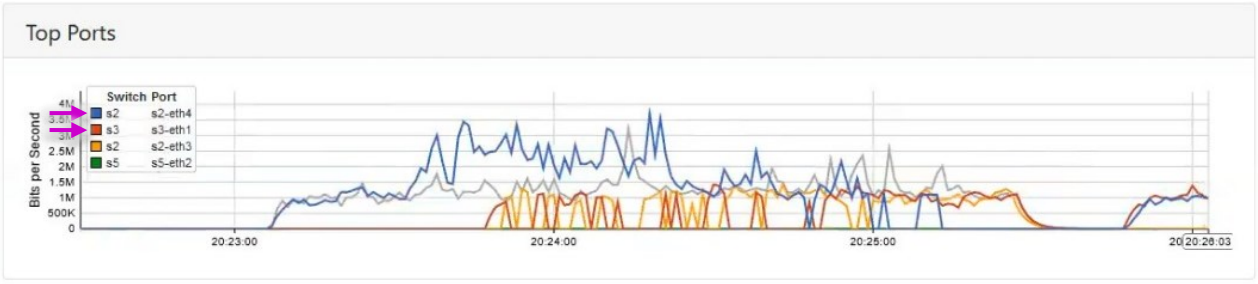
3rd Stream Generation (UDP):

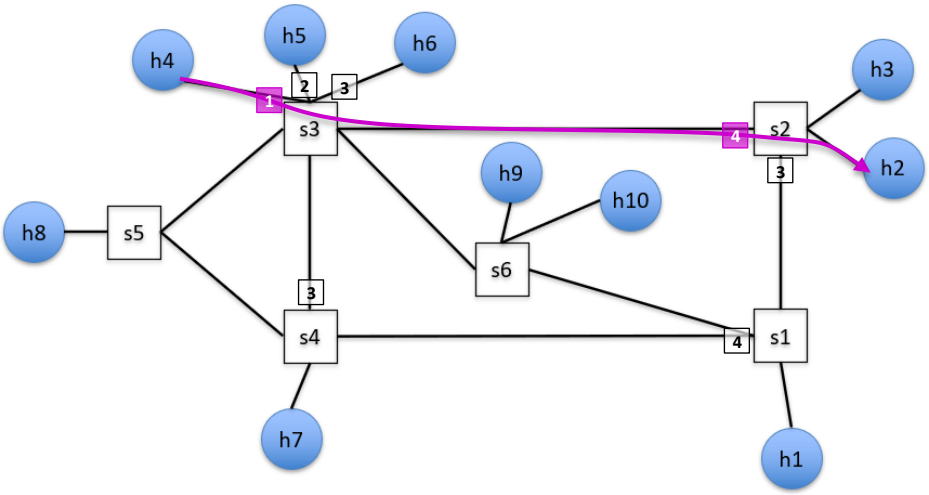
A UDP stream from h4 to h2 is generated.



4th Stream Generation (UDP):

We generate a UDP stream with the same characteristics as the previous stream, but since the timeouts of the other flows occupying the s3-s2 link have expired, the rules inserted in the switches have been removed and therefore the UDP flow is routed to the s3-s2 link (minimum path).





Installation Manual and User Manual

Start mininet with the code for the presented topology:

- **$** *sudo mn --mac --custom /*percorso_file*/Topologia_P3.py --topo Project3 --remote controller*

In a new terminal window, launch the designed controller:

- **$** *ryu-manager --observe-links /*percorso_file*/Test_Azioni.py*

Then, within the mininet, type the source command followed by the path where you saved the automated test to reproduce the demonstrator results:

- **>** *source /*percorso_file*/Azioni_host_P3.sh*

To be able to observe the results obtained on FlowManager, you also need to start the flowmanager.py script; The total command is then as follows:

- **$** *ryu-manager --observe-links /*percorso_file*/Test_Azioni.py flowmanager/flowmanager.py*

To be able to observe the results obtained on the InMon dashboard (sflow) you need to start sflow with the command:

- **$** *sflow-rt/start.sh*

And then start mininet with the --custom sflow-rt/extras/sflow.py option; The total command is then as follows:

- **$** *sudo mn --mac --custom /*percorso_file*/Topologia_P3.py --custom sflow-rt/extras/sflow.py --topo Project3 --controller remote*