Advanced Computer Networks (2021)

Lab4: Video Interception

Type: Group of 2
Maximum points: 15
Submission: report and code submission on Canvas
Deadline: Friday December 17, 2021

The goals of this lab are

- Implement an IPv4 router in P4
- Intercept RTP video streaming with P4
- Write a report to document your methodology and results

Note: Please include your group number and all group member names at the beginning of your report to facilitate our grading. Also, please rename the folder to be submitted with name “lab4-groupx” where “x” is your group number and then compress the folder into a “.zip” file. Please do the folder renaming **before** the compression so that the folder name is correct when we decompress the folder. Please submit the zipped folder on Canvas to the assignment “Lab4”.

Before we start with the tasks in this lab, you should run `install_p4_env.sh` which is located under the `lab4` directory. This is to install the necessary software environment for P4 compilation, runtime, and other related packages. Please make sure no errors occur during the installation process. The installation process may take around 30 minutes. Please be patient.

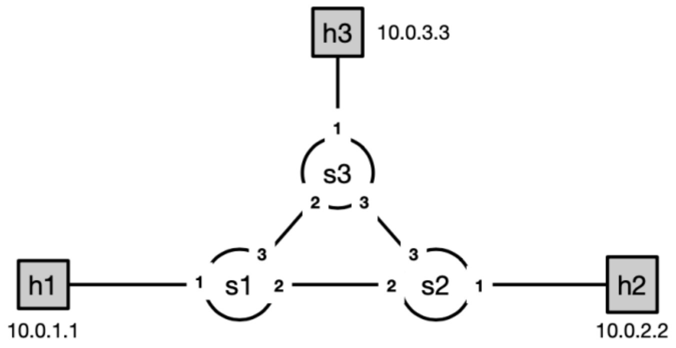
1. Implement an IPv4 router in P4

In this part of the lab, you are supposed to implement an IPv4 router with the P4 language. You should work in the directory `lab4/routing`. Please do not change the directory structure; otherwise, your code may not compile correctly.

Your first task is to implement an IPv4 router with P4 in the code template `routing.p4`. As we learned in the lecture, you can define packet headers, specify packet header parsers, and define match-action tables with P4. Here, you need to properly define the packet headers including Ethernet (already given in the code template) and IPv4 (check RFC791 for details: https://tools.ietf.org/html/rfc791) headers, define parsers to properly parse the packet headers you have just defined, and define a match-action table that matches on the destination IP address (or prefix) and forwards the packet to the output port based on a forwarding table that will be supplied by the control plane. You may want to refresh what you have learned in Lecture 10 on a programmable data plane with P4. Alternatively, the following tutorial may help you understand the basics of P4: https://p4.org/assets/P4_tutorial_01_basics.gslide.pdf.

For IPv4 routing, please pay attention to what operations you need to perform on a router. You can refer to the following document if you want to refresh your knowledge about IPv4 routing: https://people.cs.umass.edu/~arun/cs491g/lectures/IPForwarding-Lab3.pdf.

Once you finish your code you can compile it by running `make` in the `lab4` directory. You will see a Mininet prompt if the compilation is successful. If so, an instance of the following network is now built in Mininet. (The tiny numbers on the routers denote the port numbers.)



Your second task is to populate the table entries for the routers you have just built with a control plane. The routers (s1, s2, and s3) in this network work exactly in the way you have specified them with P4 in `routing.p4`. However, the routers have empty routing tables and you need to tell them how to forward packets by specifying table entries. This is similar to what you have done with Ryu where you install the flow entries by writing a control script for Ryu. However, here we take a simplified approach where we specify the table entries manually for the routers.

To specify table entries for the routers in the network, you can navigate to directory `lab4/routing/topo/` where you can find three files with name in the form `s*-runtime.json`, each corresponding to one of the routers matching their names. In these files, you can find a segment called `table_entries` and this is where you can specify the table entries that you want to install for your routers. Note that the format for the table entry is dictated by the way you define the match-action table in `routing.p4`. You may want to check the IP/MAC addresses that have been assigned to the hosts and you can find such information in the file `lab4/topo/topology.json`. The hosts have already been configured with default gateways and ARP tables, so you do not need to do anything to them.

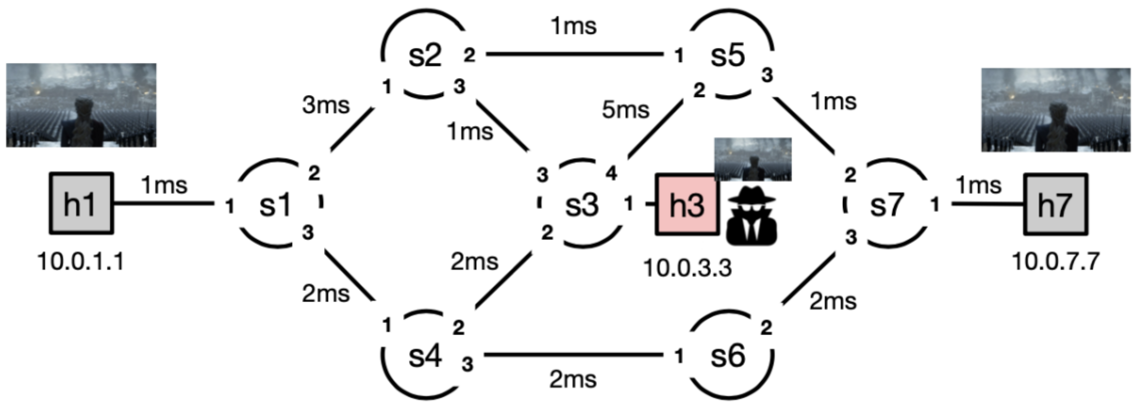
Once you have completed your router specification and inserted table entries, you can test if the routers are correctly implemented for IPv4 routing with the following command:

```
$make
mininet> pingall
```

2. Intercept RTP video streaming with P4

In this part of the lab, you should work in the `lab4/streaming` directory. Please do not change the directory structure to make sure that your code will compile correctly.

Imagine an ISP has a P4-enabled network with a topology depicted in the following figure.



There is a customer (`h7`) of the ISP who is watching a new TV series streamed from server `h1` over the network using the RTP protocol. Now assume you are also interested in this TV series but do not have access to it. As a network engineer you found a magic way to take control of the whole network and intercept the RTP connection `h1-h7` to get access to the TV series.

Your first task is to configure the network such that it works as it should. Let us assume the ISP was running a shortest-path routing protocol. The goal is to establish a connection between `h1` and `h7` with the lowest delay. Following the experience you gained from the previous tasks, please configure the routers in the network such that this goal can be achieved. (It is okay if you do manual calculations for the shortest path.) Here, we expect you to parse the packet all the way to the RTP packet header, meaning that you need to specify the packet header formats for Ethernet, IPv4, UDP (see RFC 768: https://tools.ietf.org/html/rfc768), and RTP (see RFC 3550: https://tools.ietf.org/html/rfc3550).

Once you have set up forwarding correctly on all the routers, you can run `vlc-server.sh` on `h1` and `vlc-client.sh` on `h7` to see if the RTP streaming works as expected. You can use

```
mininet> xterm h1 h7
```

to open terminals for `h1` and `h7` and run the scripts from these terminals. These scripts are located under directory `lab4/streaming`.

Now comes to the more interesting part: You want to intercept the traffic between `h1` and `h7` from `h3`. With P4 this can be achieved by either multicast (check the `mcast_grp` metadata at https://github.com/p4lang/behavioral-model/blob/main/docs/simple_switch.md) or cloning on routers. You are free to choose either. The goal is to be able to receive a copy of all the packets from `h1` to `h7` on `h3`. Once you have implemented your solution, you can test it by running

```
$make
minint> h1 ping h7
```

Use `tcpdump` to listen on `h3-eth0` and check what you see from there. You can also test if you can receive and play the RTP stream on `h3` by running `vlc-client.sh` on `h3` and `vlc-server.sh` on `h1`. You might notice that `h3` is not able to play the video even though the packets have been forwarded to it. Think about why and find a way to handle it.

3. Write a report to document your methodology and results

Once you have finished all the above two parts, congratulations! You have completed all the programming assignments for Advanced Computer Networks. Yet, you have one last task to finish here. Please write a report to document your methodology and basic setup for the above implementations, as well as your findings, just as what you have done for the previous assignments.

4. Assessment criteria

This lab will be assessed based on the following criteria:

- **Correctly implementing the IPv4 router (5 points).** Your code should correctly implement the IPv4 router in P4 and the `pingall` test should succeed.
- **Correctly implementing the RTP streaming interception.** Your code should correctly implement the RTP streaming connection from `h1` to `h7` where you parse the packets up to the RTP protocol on all the routers and you should be able to play the video at `h7` **(3 points)**. Also, your code should allow you to play the streaming at `h3` **(4 points)**.
- **Code and report quality (3 points).** Your report should document the necessary details for your implementation, including your methodology, system setups, and your observations, or other interesting findings/thoughts.

5. Bonus

If you want to challenge yourself further, the following bonus worth **5 points** is available for you. You have to complete the bonus task in full, no partial points can be claimed for incomplete attempts. Please make a copy of the `streaming` folder and rename it as `bonus` before you work on it.

The bonus task is to enable multi-path packet interception from `h1` to `h3`, i.e., route the intercepted packets to `h3` via more than one path and balance the load on these paths. You are free to choose the paths but you need to explain why you choose them. Also, since out of order delivery is probable with multi-path routing, you need to find a way to quantify the severity of out-of-order delivery on `s3`. Explain in the report how you do it, why, and what you observe.