P4 - PA1: Custom Function on a Programable Switch

Description:

One of the main advantages of programmable data planes is the fact that you can implement whatever you need in your switch instead of being restricted to a fixed set of functions according to the corresponding nature of your specific device (e.g. switch, router, firewall, etc).

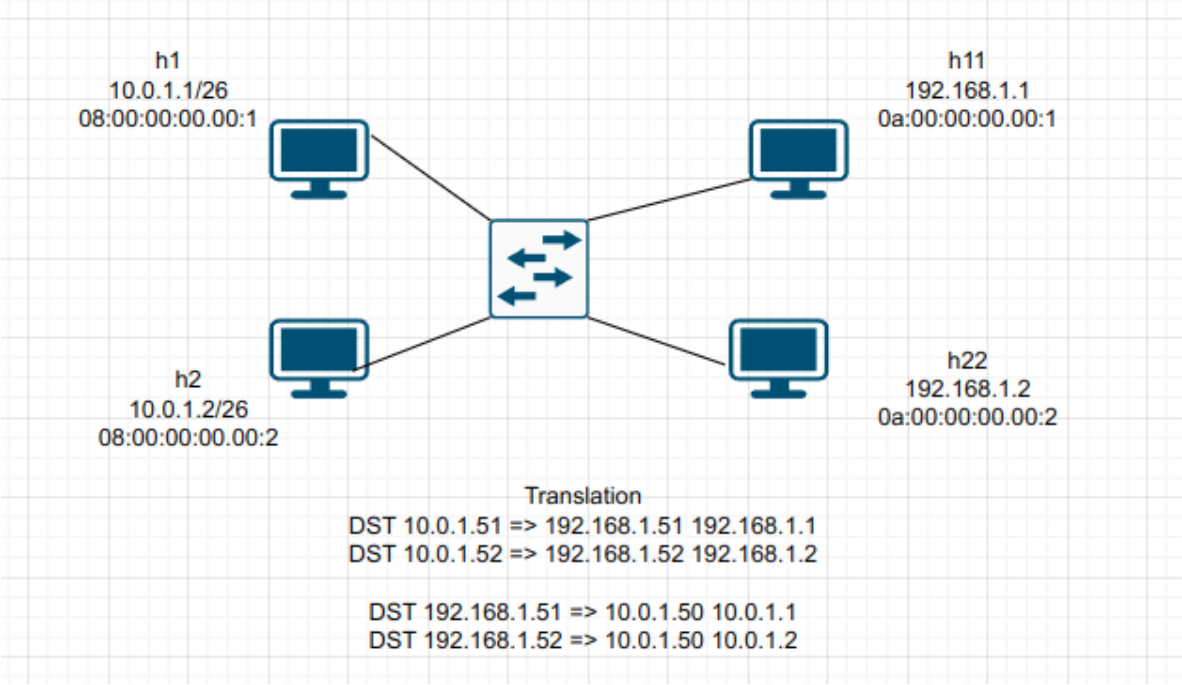
In the present exercise, you will implement a switch capable to i) respond to ARP queries, ii) forward traffic based on IP addresses and iii) perform NAT to increase the isolation of a network segment.

Rationale:

In some cases, instead of implementing routing between two network segments, you might want to provide connectivity between specific pairs of hosts in a network, as if they were located in the same segment. This can be done through NAT, by modifying source or destination IP addresses. In this exercise, you will implement NAT by using P4.

Operation Principle

In the code that you will be provided (See instructions later), you will have the following topology created in mininet. You will have to implement the corresponding translations.



- From host h1, you will be able to ping to the destination IP 10.0.1.51. The switch will change the destination address to become 192.168.1.1, so that the ping can arrive at h11. In order to have the ping working well in the back address, the source IP address of the packet will be changed also to 192.168.1.51. Then, the echo reply will have the source address 192.168.1.1 and the destination address 192.168.1.51. When the packet arrives at the switch, the destination address will be restored to 10.0.1.1, and the source address will be 10.0.1.51.
- From host h2, you will be able to ping to the destination IP 10.0.1.52. The switch will change the destination address to 192.168.1.2, so that the ping can arrive at h22. To have the ping working well in the back address, the source IP address of the packet will be changed also to 192.168.1.52. Then, the echo reply will have the source address 192.168.1.2 and the destination address 192.168.1.52. When the packet arrives at the switch, the destination address will be restored to be 10.0.1.2, and the source address will be 10.0.1.52

Implementation Tip 1: Remember that your switch will know how to respond to ARP requests. To implement the translation, you must ensure that the packets targeting the corresponding destination IPs (10.0.1.51 from h1 and 10.0.1.52 from h2) be received at the switch. The destination mac address and the destination ip address within these packets will trigger the NAT operation.

Implementation Tip 2: Since you are modifying parts of the IP header, you must ensure the checksum is correctly updated.

To have this implementation working successfully, complete all the tasks marked with TO-DO in the source code of the p4 program template provided to you.

Installation and execution

- Unpack the file PA1.tgz available at this link at the /home/p4/tutorials folder of your virtual machine instance (available in the same folder).
- Edit the simple_switch.p4 file and perform the modifications indicated with the TO-DO comments.
- You can test your configurations by running your topology with the make run command. From the mininet prompt, you can execute the corresponding ping commands like this: XX ping YY where XX is the source node (either h1 or h2) and YY is the destination ip address (either 10.0.1.51 or 10.0.1.52). Keep in mind that ping will work only between the specified pair of hosts. That is, h1 should not be able to ping 10.0.1.52 and h2 should not be able to ping 10.0.1.51.

The file README.md tells you how to connect to the switch. The file rules.sh contained in the source file will provide you with examples about how to insert entries in the tables. Check the structure of your tables and your actions in the source code of the P4 program.

Bonus: Suppose you have to use as destination addresses 10.0.1.151 and 10.0.1.152 instead of the corresponding provider addresses. What would you have to keep in mind for this case?

Contact in case of questions: sergio.gutierrezb@udea.edu.co