Redes Definidas por Software
L3 & L4 Packet Manipulation & P4RunTime

Version: 1

Departamento de Informática
Universidade do Minho
Apr 2024

João Fernandes Pereira

Assignment Overview

Each exercise should build upon the knowledge gained from the previous assignments and gradually introduce new concepts and challenges. By the end of the assignment, students will have a comprehensive understanding of building data planes with P4 and integrating them into SDN environments.

Exercise 1: Simple-Router - ICMP Response

1. **Objective:** Modify the existing P4 simple router to respond to ICMP echo requests (ping) addressed to any of its interfaces.
2. **Steps:**
   - Understand the structure of the existing P4 simple router from the first assignment.
   - Identify an ICMP message: IPv4 Protocol Field Values
   - Identify the necessary modifications to intercept ICMP echo requests. Header fields: *Type* and *Code*.
   - Implement the required changes in the P4 code.
   - Test the modified router using Mininet by sending ICMP echo requests to different interfaces and ensuring appropriate responses.

Exercise 2: Load Balancer with Network Address Translation (NAT)

In this exercise, we'll extend the capabilities of our data plane to incorporate a load balancer with NAT functionality. Consider the company described in assignment 1 with three departments, each requiring access to a common application hosted on three identical servers. However, the servers' addresses are hidden from the departments, and they only know the address of the load balancer.

- **Company Departments:** Each department has access to the load balancer's IPv4 address.
- **Load Balancer:** Responsible for distributing incoming traffic from the departments to the three servers running the same application. It must alter the Layer 3 (L3) and Layer 4 (L4) headers of incoming packets and forward them to one of the servers, aiming to maintain balanced traffic distribution. (keyword: PAT - Port Address Translation)
- **Servers:** Three identical servers hosting the application. They are hidden behind the load balancer and are only accessible through it.

Objectives

1. Implement a load balancer using P4 capable of:
    - Receiving traffic from company departments.
    - Altering the L3 and L4 headers of incoming packets.
    - Distributing traffic to one of the three servers while attempting to maintain balanced load distribution.
2. Incorporate NAT functionality into the load balancer to ensure that:
    - Outgoing packets from servers undergo NAT to conceal their addresses.
    - Responses from servers are properly mapped to the corresponding clients based on TCP sessions.
    - Test the load balancer functionality using Mininet with multiple identical servers behind the NAT.
      `python3 -m http.server`

Note:

The load balancer's ability to dynamically adjust traffic distribution based on load conditions can only be fully realized with the implementation of the control plane using P4Runtime. However, for the purposes of this assignment, manual alteration of load balancing rules on the load balancer is acceptable. Students are encouraged to explore dynamic load balancing as part of optional exercise 3.

Exercise 3: P4RunTime (optional)

1. **Objective:** Implement P4Runtime using a Python library and integrate it with Mininet using the correct switch class.
2. **Steps:**
    - Learn about P4Runtime and its importance in Software-Defined Networking.
    - Choose a Python library for P4Runtime (e.g., P4Utils).
    - Understand the P4Runtime architecture and API.
    - Implement P4Runtime support in the existing Mininet environment.
    - Modify the switch class in Mininet to support P4Runtime.
    - Test the P4Runtime implementation by deploying P4 programs to switches using the Python library and verifying their functionality in Mininet.

Controller Development

-> This section will be released after a two hour class on the subject. <-

Submission Guidelines

- Submit all the P4 code.
- Submit the Mininet topology.
- Submit the rules.
- Include a Report file with:

  - Instructions to run your code on Mininet with the provided topology.
  - Description of ICMP manipulation.
  - Description of load balancing strategy.
  - Description of the developed Controller and P4Runtime arch integration.
  - Explanation of all the test scenarios and their outcomes in the company network context.
  - Any challenges faced and how they were overcome.
- The submission must be a .zip archive, **RDS-G"X"-TP2.zip**. Replace **"X"** with your group number. Your group number contains 2 digits.

Important Notes

- Check all the links in this document.
- Collaboration with peers is allowed for discussions and problem-solving, but each submission must be as a group.
- The deadline for this assignment is 07 June 2024.
- If you have any questions or need clarification, please reach out to d12267@di.uminho.pt.

Useful Links

- ICMP
- PAT
- Official repo of P4Runtime, with tutorials