

SOFTWARE DEFINED NETWORKS

PROGRAMMABLE DATA PLANE USING P4 LANGUAGE

Midterm report

Team members

- 1. Naga Ganesh Kurapati**
- 2. Rakesh Kiran Musalay**
- 3. Ramanand Shankarling**

Hand in date: March 26, 2016

Due date: March 26, 2016

TABLE OF CONTENTS

1. PROBLEM STATEMENT
2. OBJECTIVES
3. SOFTWARE REQUIREMENTS
4. PLAN OF ACTION
5. CONCLUSION

1. PROBLEM STATEMENT

The network data plane should be programmed to de-capsulate the outer tag/labels to identify and/or trace the packets. This project will use P4, a domain specific programming language to design packet encapsulate/de-capsulate modules that can process packet headers and perform the following tasks

- (a) Identify OpenSSL flows
- (b) Identify video streams from/to a streaming server (assume using Mist Server for streaming)
- (c) Identify YouTube streams
- (d) Record and report statistics regarding the above flows

The team is expected to use P4 reference design as the starting point.

2. OBJECTIVES

Software-Defined Networking (SDN) gives operators programmatic control over their networks. In SDN, the control plane is separate from the forwarding plane. We use P4 language to parse the packets and identify specific type of packets, depending on which further actions can be defined. As given in the problem statement above, the designed software shall be able to identify OpenSSL flows, identify video streams. A record of statistics of packet flows needs to be generated which could be used in further analysis or decision in course of action.

3. SOFTWARE REQUIREMENTS

- Oracle VM VirtualBox
- Mininet
- P4 Compiler
- Streaming server – Mist / Red5

4. PLAN OF ACTION

For this project, we have divided our action plan in to different phases. We believe that we would be able to complete the project following the below plan.

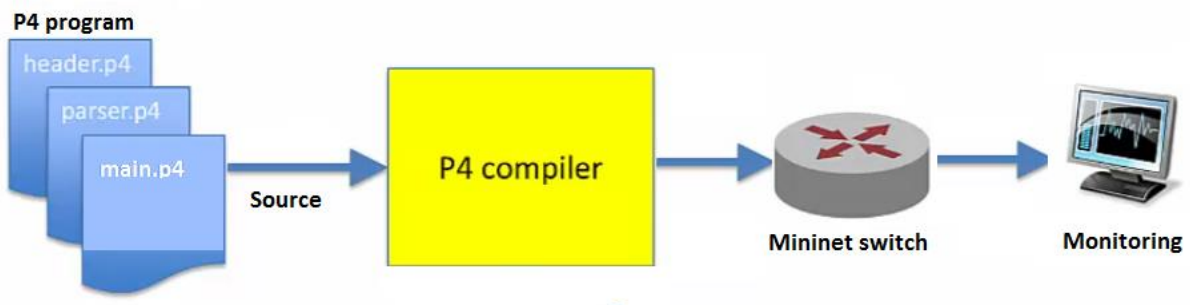
Phase 1

- Literature review
- Learn to program and compile a sample P4 program

Software defined networking is an emerging field in the networking technology. Our team, being new to this, were required to study and learn the basics of SDN and P4 programming. P4 is a high level language for Programming Protocol-Independent Packet Processors. P4 works in conjunction with SDN control protocols like OpenFlow

Phase 2

- Write a program in P4 language to parse the packets
- Perform Encapsulation and Decapsulation
- Identify specific type of packets by using available information after parsing of packets.
- Take intended course of action depending on type of packet.



The program would consist of three files

Header file – Contains definitions for standard header types.

Parser file – Implements parser functionality of the system to perform extraction operation on incoming byte streams.

Main file – Describes the overall flow of the program and action for parsed packets.

A sample parser program we found in our study. We intend to use the same to parse packets and identify the streaming type of packets.

```
parser parse_ethernet {
  extract(ethernet);
  return select(latest.etherType) {
    0x0800 : parse_ipv4;
    default: ingress;
  }}
...
parser parse_ipv4 { ... }

calculated_field ipv4.hdrChecksum {
  verify ipv4_checksum;
  update ipv4_checksum;
}
```

Phase 3

- Build an open source streaming server which would be able to stream content in order generate packets for our designed system.
For this, we plan to either use Mist or Red 5.
- Red 5 is an open source media server for live streaming solutions of all kinds. It is designed to be flexible with a simple plugin architecture that allows for customization of virtually any VOD and live streaming scenario.
- MistServer is a streaming media server that works well in any streaming environment. It works best as a Linux **media** server but is completely dependency-free and runs on Windows or Mac (OSX) systems as well.

Phase 4

- Keep a record of number of type of packets parsed using counter on variables.
- A graph or report could be generated to give the summary of data collected.
- Prepare a project report.

5. CONCLUSION

With our study so far, we have understood the structure, syntax and semantics of P4 language. We are able to write and execute a sample P4 program. Also, we have a fair idea of parser logic, which we need to implement and identify packet types. Then, we will proceed to keeping record and reporting the statistics of data collected.