Real-Time Traffic Classifier on Mellanox Switch– using P4

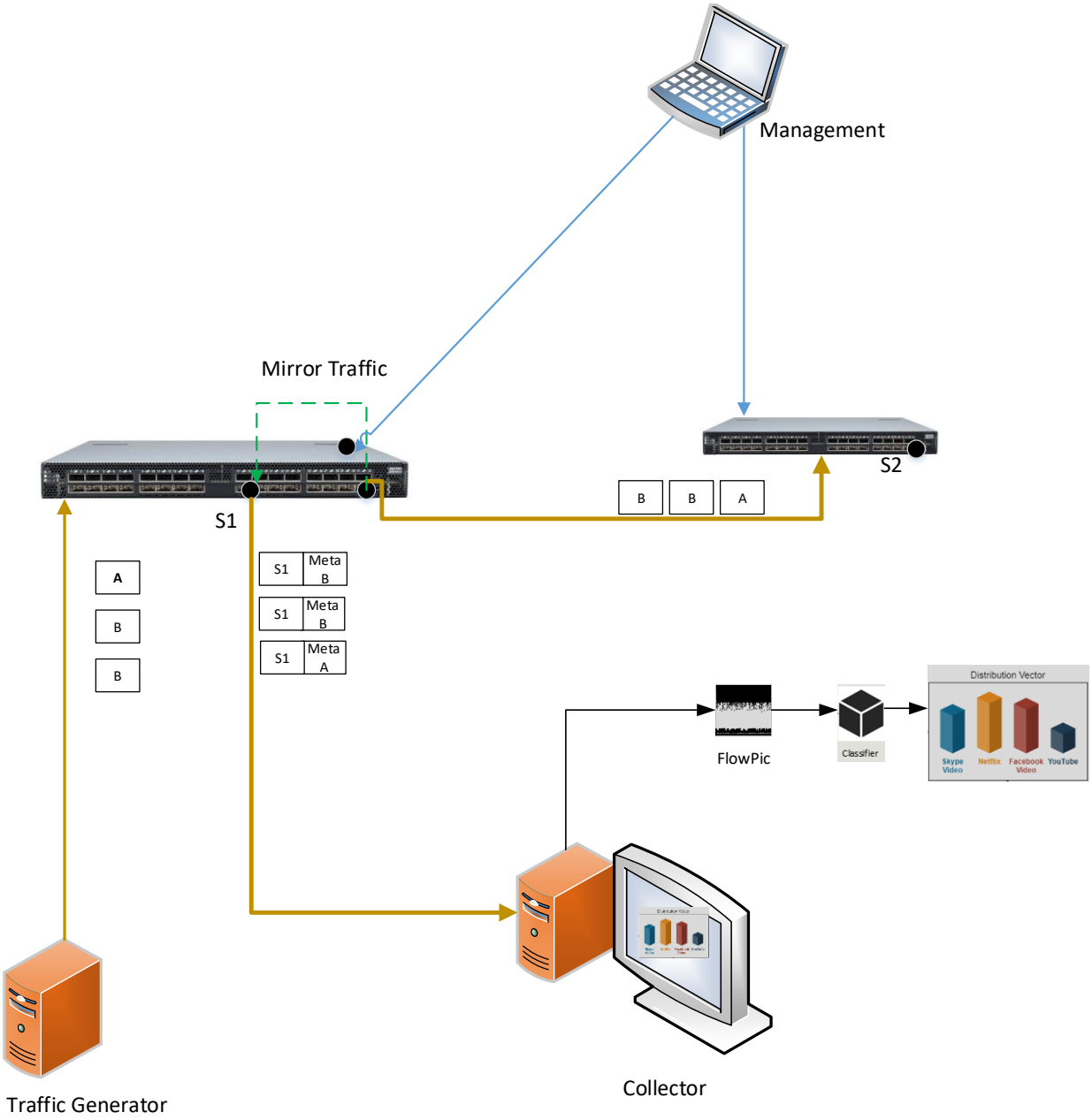
Abstract:

Traffic classification, the categorization of network traffic into appropriate classes, is important to many applications, such as quality of service (QoS) control, pricing, resource usage planning, malware detection, and intrusion detection. Because of its importance, many different approaches have been developed over years to accommodate the diverse and changing needs of different application scenarios. In particular, the growing trends of Internet traffic encryption and an increase Virtual Private Networks (VPNs) and The Onion Router (ToR) usage, raise additional challenges to network traffic classification.

Using Deep Learning (DL) methods, we managed to train a Convolutional Neural Network (CNN) model that can classify encrypted internet traffic. The model is trained with labeled FlowPic images build from the flow packets metadata (packet arrival time, packet length). This technology enables real-time traffic classification, once the model receives FlowPic image of a certain flow.

Programming Protocol-independent Packet Processor (P4) is a high-level language that can be deployed in the future into Software Defined Networks (SDN) and can actually serve as an alternative to OpenFlow that is currently used – due to its flexibility and ability program the data plane and support emerging new protocols.

In this project we will use Mellanox SN3700 P4-capable Spectrum-2 based switch. The P4 programmed switch will send to the collector, for a certain flow, its packet metadata. The collector will build FlowPic image and send it to the trained model for real-time flow classification.


Management
Mirror Traffic
S1
S2
B
B
A
A
B
B
S1
Meta B
S1
Meta B
S1
Meta A
FlowPic
Classifier
Distribution Vector
Skype Video
Netflix
Facebook Video
YouTube
Traffic Generator
Collector

Goals:

The project's objective is to learn P4 programming language and use it on Mellanox P4 Capable switch (SN37000) in order to be able to build per flow its FlowPic image and send it to a trained image CNN model for traffic classification . The project will include the following phases:

- Learn the P4-16 language
  - Refer to http://p4.org/
  - Read the paper **The P416 Programming Language:** https://dl.acm.org/citation.cfm?id=3139648
  - Perform basic P4 exercise on Mininet - https://github.com/p4lang/tutorials/tree/master/exercises/basic
  - **Read** https://p4.org/p4/prototyping-in-p4.html

- Learn the Mellanox p4 target architecture (See Appendix A)
- Learn the Mellanox p4 Architecture Schema (See Appendix B)
- Refer to previous student Mellanox-P4 project – https://gitlab.cs.technion.ac.il/lccn/s2020-intelligent-traffic-debugger-using-p4
- **Read the FlowPic: Encrypted Internet Traffic Classification is as Easy as Image Recognition -** https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8845315&tag=1
- Refer to previous student Deep Learning Traffic Classification project – https://gitlab.cs.technion.ac.il/lccn/s2020-deep-learning-based-traffic-classification-new
- Implement the following steps:
  - Using P4RT, Initialize the P4 program tables in Mellanox switch mirror to a GRE port certain flow, with additional metadata per packet:
    - Packet arrival time
    - Packet length
  - Play on the Traffic Generator pre-recorded traffic pcap file that includes several flows
  - Once received on the collector – build the FlowPic image and send it to the model for flow classification.
- Stretch goal: Check accuracy when sending traffic samples (for example – 1:32).

Appendix A: Mellanox p4 target architecture

The current Mellanox p4 target architecture compress from 5 programmable blocks (1 parser block, and 4 control - match action).

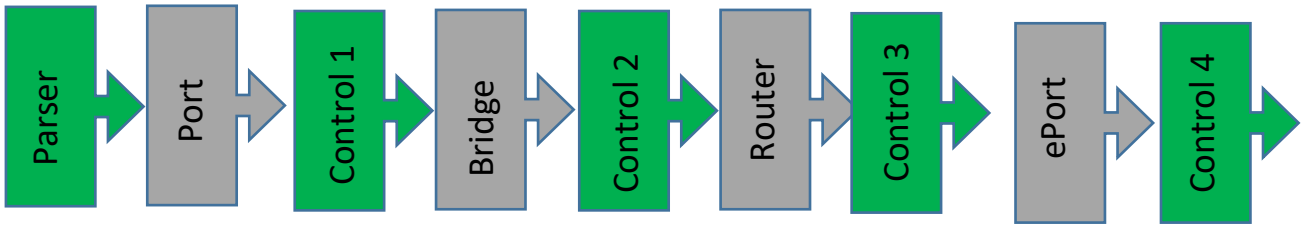
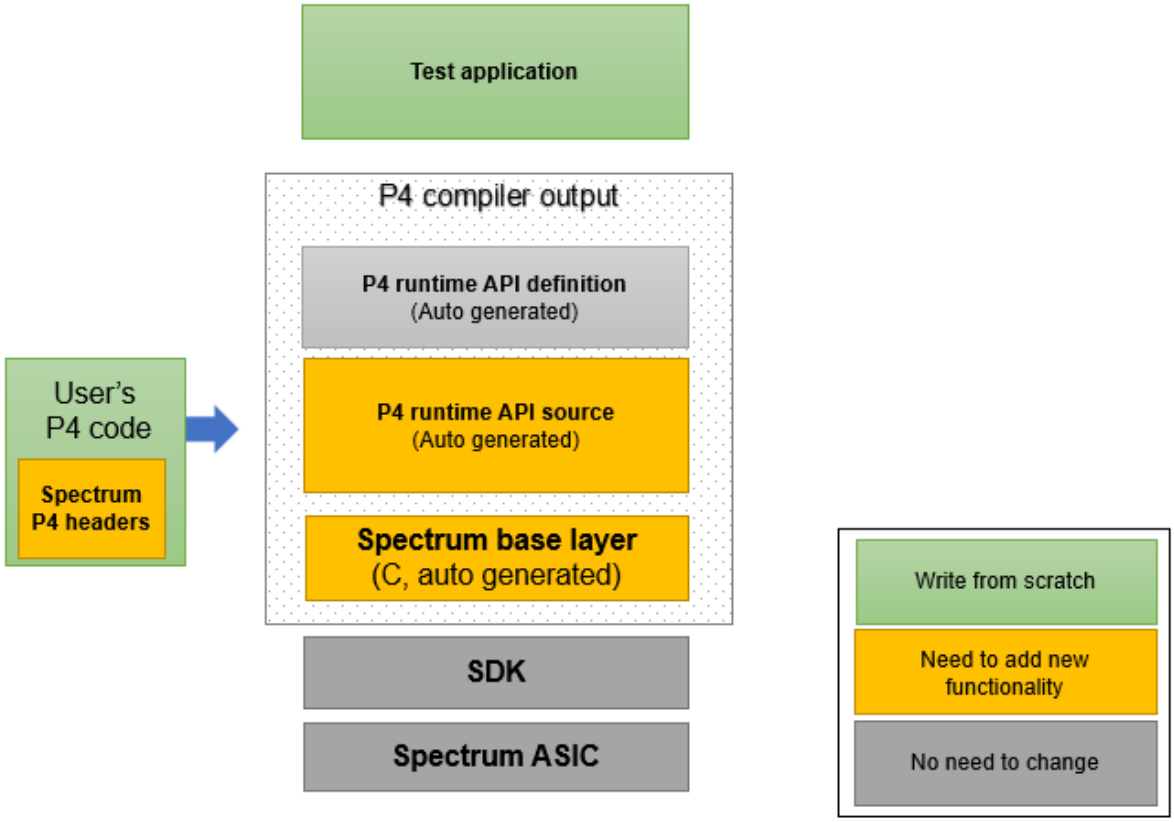


Figure 1. Target architecture**.**

Programmable block 1: parser
Mellanox provides parsing graph base line user will be able to add up to 4 new nodes to the packet-parsing graph.
Programmable block 2: ingress port
Ability to define chain of multiple match action tables supported actions – drop, forward to port , mirror, packet modification, routing(including ECMP) ,tunnels encap ,tunnel decp , set QoS, counters, meters ,go to table.
Programmable block 3: ingress router
Ability to define chain of multiple match action tables supported actions – drop, mirror, packet modification, routing(including ECMP) ,tunnels encap ,tunnel decp , set QoS, counters, meters ,go to table.
Programmable block 4: egress router
Ability to define chain of multiple match action tables supported actions – drop, mirror, packet ,forward to port , packet modification, set QoS, counters, meters ,go to table
Programmable block 5: egress port
Ability to define chain of multiple match action tables supported actions – drop, egress mirror, packet modification, set QoS, counters, meters ,go to table “

Appendix B: Architectural schema



Requirements:

Introduction to Networking (Must), Internet Networking (Optional)

Introduction to Artificial Intelligence (Must)

or

Introduction to Machine Learning (Must)

Guided by:

Matty Kadosh & Alan Lo

Resources:

1. P4 tutorials on GitHub (see readme for install instructions):

https://github.com/p4lang/tutorials

2. P4 mailing list:

http://mail.p4.org/pipermail/p4-dev_p4.org/

3. P4 runtime:

https://p4.org/p4-runtime/

4. Mellanox SDK API:

http://www.mellanox.com/page/products_dyn?product_family=124&mtag=switchx_sdk

5. Mellanox P4 compiler:

Code repository will be shared with the students.