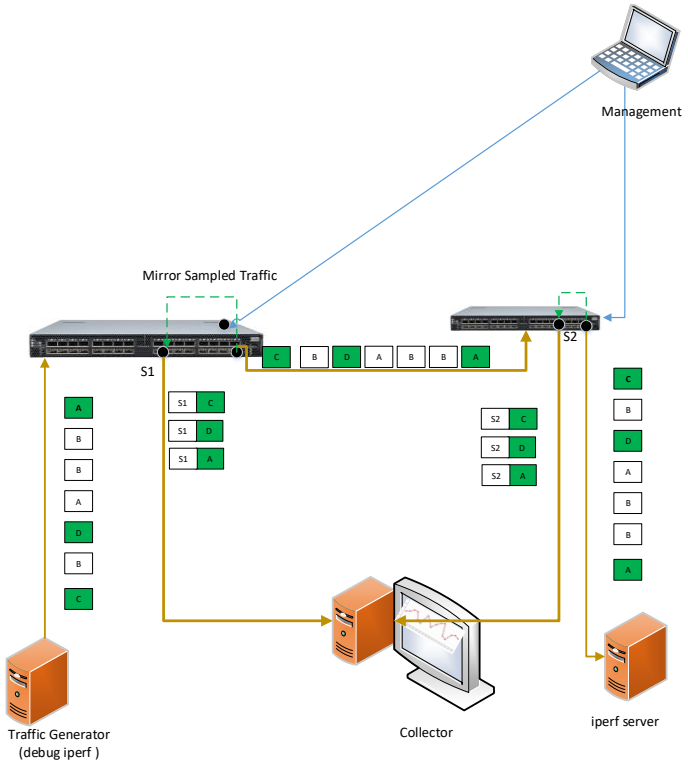
Intelligent End-To-End Traffic Congestion Trouble Shooting – using P4

Abstract:

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.

Debugging End-To-End traffic problems and finding the root cause for packets congestion or un-expected high End-To-End latency plays a significant role in network management. Trouble shooting such issues in order to find the root cause, especially in a Data Center where traffic volume is massive, can be a difficult task.

In this project we will use Mellanox SN3700 P4-capable Spectrum-2 based switches and implement an online intelligent method, based on postcard telemetry, that will debug end-to-end traffic congestion or high latency events and will pinpoint its root cause.



Goals:

The project's objective is to learn P4 programming language and deploy it on Mellanox P4 Capable switch (SN37000). In addition, we will implement in this project on line machine learning cardinality estimation framework. 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

- Learn the Mellanox p4 target architecture (See Appendix A)
- Learn the Mellanox p4 Architecture Schema (See Appendix B)
- Refer to previous student project – https://gitlab.cs.technion.ac.il/lccn/w2019-postcard-with-p4.
- Implement the following steps:
  - Initialize the P4 program tables in Mellanox switches to send Postcard telemetry to the collector on a match of DSCP value (for example: 0x0F). The postcard will include:
    - switch ID
    - marked packet 5-tuple
    - switch latency
    - egress queue size
  - Run on the Traffic Generator in the above topology iperf TCP/UDP "good user" end-to-end traffic with multiple flows
  - Create a congestion event by adding iperf TCP/UDP "bad user" end-to-end traffic

    Assuming an external traffic loss notification is received from the "good user"…

  - Run on the Traffic Generator "DebugTool ipef" that will:
    - Find all sockets (connections) opened by iperf
    - Per each socket: find the flow 5-tupple
    - For each found flow: Update the ip-tables to match every X-th flow packet and mark the packet with a DSCP value (for example: 0x0F)

- On the collector side – the analyze tool will start receiving postcards from both switches and needs to detect for each flow:
  - Path
  - End-To-End latency (note: latency on the cable between to switches is known)
  - Egress queue sizes
  - Drop Rate

  Note: matching certain flow packet traversing between switches requires hash calculations – for example – on IP header (without TTL..)

The target: Detect the "bad flow(s)"

- Stretch Goal: "Zoom-In" on the "bad flows" by updating the traffic generator to stop marking good flows and marking the "bad flows" in higher frequency (2*X for example). In addition, modify the P4 tables to match on the DSCP and only on the bad flow(s) 5-tupple

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).

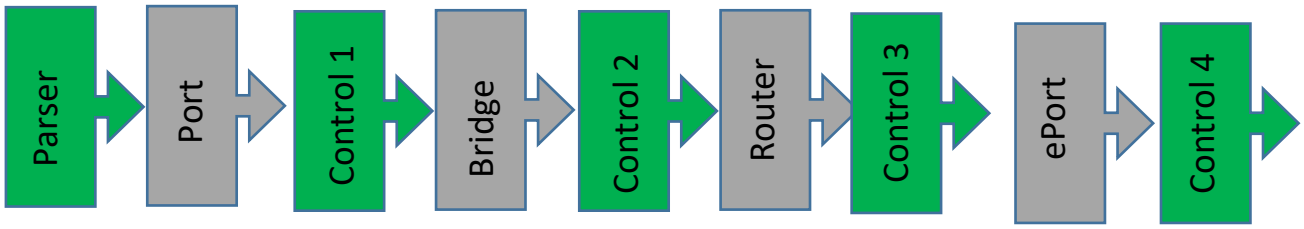
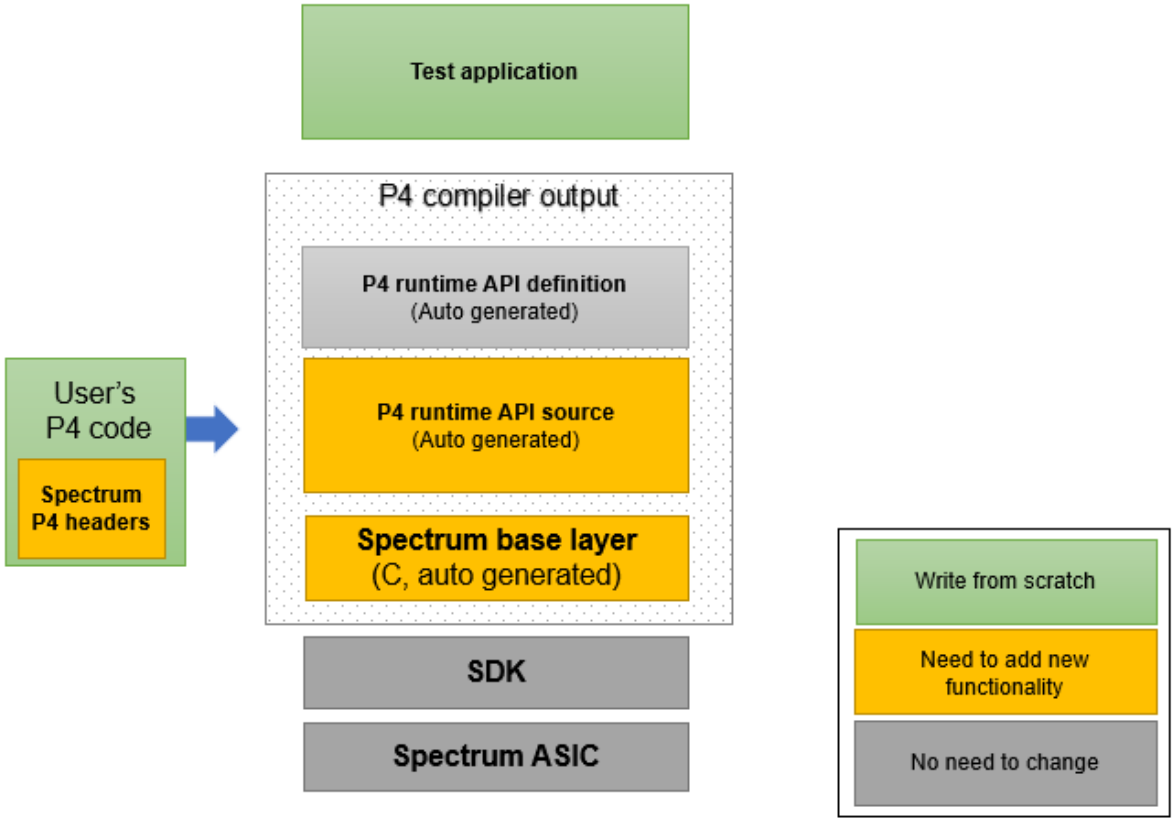


Figure 1. Target architecture**.**

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 Course (236334)

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.