Sponsored By

Google intel STORDIS The Open Networking Expert

PBT-on-Demand on Mellanox P4-Capable Hybrid Switch

Itzik Ashkenazi
Networking Lab Chief Engineer
CS Faculty, Technion, Israel

Agenda

- Introduction
- Enhanced Network Monitoring
- PBT-on-Demand
- Mellanox Spectrum-2
  - P4 Architecture
  - Extensibility Mechanisms
  - Telemetry Header
- PoC Implementation

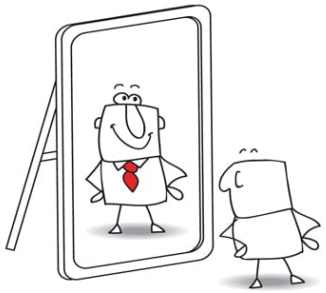
Introduction

- Joint Project

- Students: Sami Zreik, Abdallah Yassin
- Instructors: Alan Lo (Mellanox), Matty Kadosh (Mellanox), Itzik Ashkenazi (LCCN)

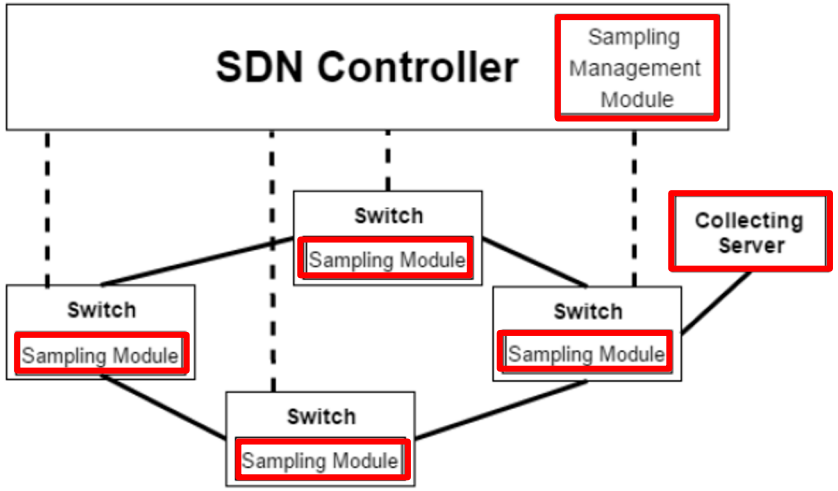
Enhanced Network Monitoring

- Needed for:
  - Traffic Engineering
  - Security
  - Anomaly Detection
  - Online Troubleshooting
- OAM Protocols: IEEE 802.1ag , ITU-T Y.1731 are useful , but..
- Flow statistics (NetFlow) is not enough…
- Need the real packet level information
- Mirroring all packets is not a good option..
- So.. mirror sampled packets



Sampling-on-Demand

- Mirroring and Sampling is an expensive resource
- sampling-on-demand monitoring framework proposed by [1]
- Sampling Management Module (SMM)
  - An SDN controller application.
  - Determines the sampling rate of each flow at each Switch according to the monitoring goals of the network operator, while taking into account the monitoring capabilities of each switch.
- Sampling Module
  - Added to some or all network switches/routers.
  - Encapsulates each sampled packet and sends it to a collecting server.
- Collecting Server
  - One or more are located in the network in order to collect and process the sampled packets.



[1] R. Cohen, E. Moroshko: Sampling-on-Demand in SDN. IEEE/ACM Transactions on Networking, Vol. 26, No. 6, Dec. 2018

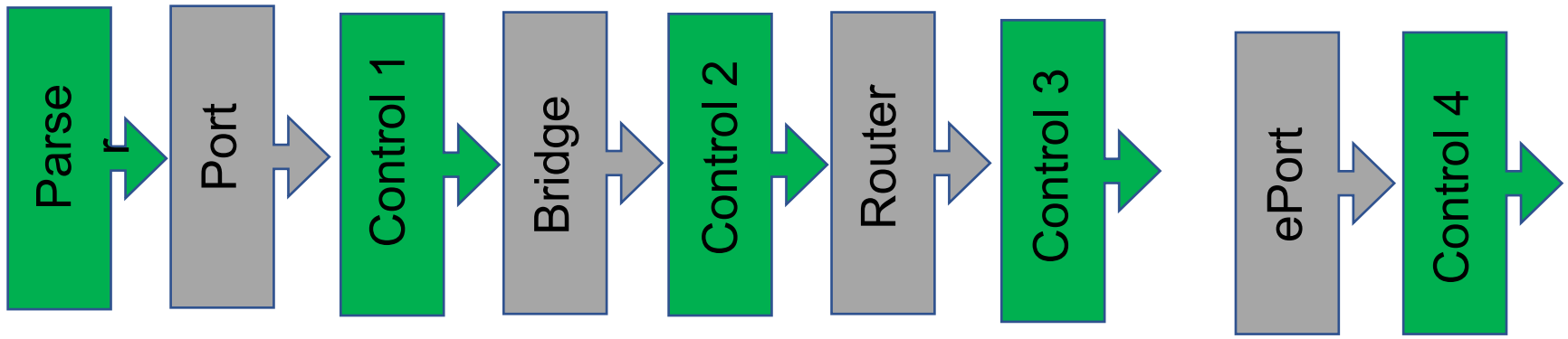
PBT-on-Demand

- Postcard-Based Telemetry (PBT) exports the packet with on-line meta-data like:
  - Switch ID
  - Time Stamp
  - Latency
  - Ingress/Egress queues occupancy
- PBT-on-Demand allows the network controller to tell the network switches how to use the expensive PBT resources in most optimal way

Mellanox Ethernet Switch

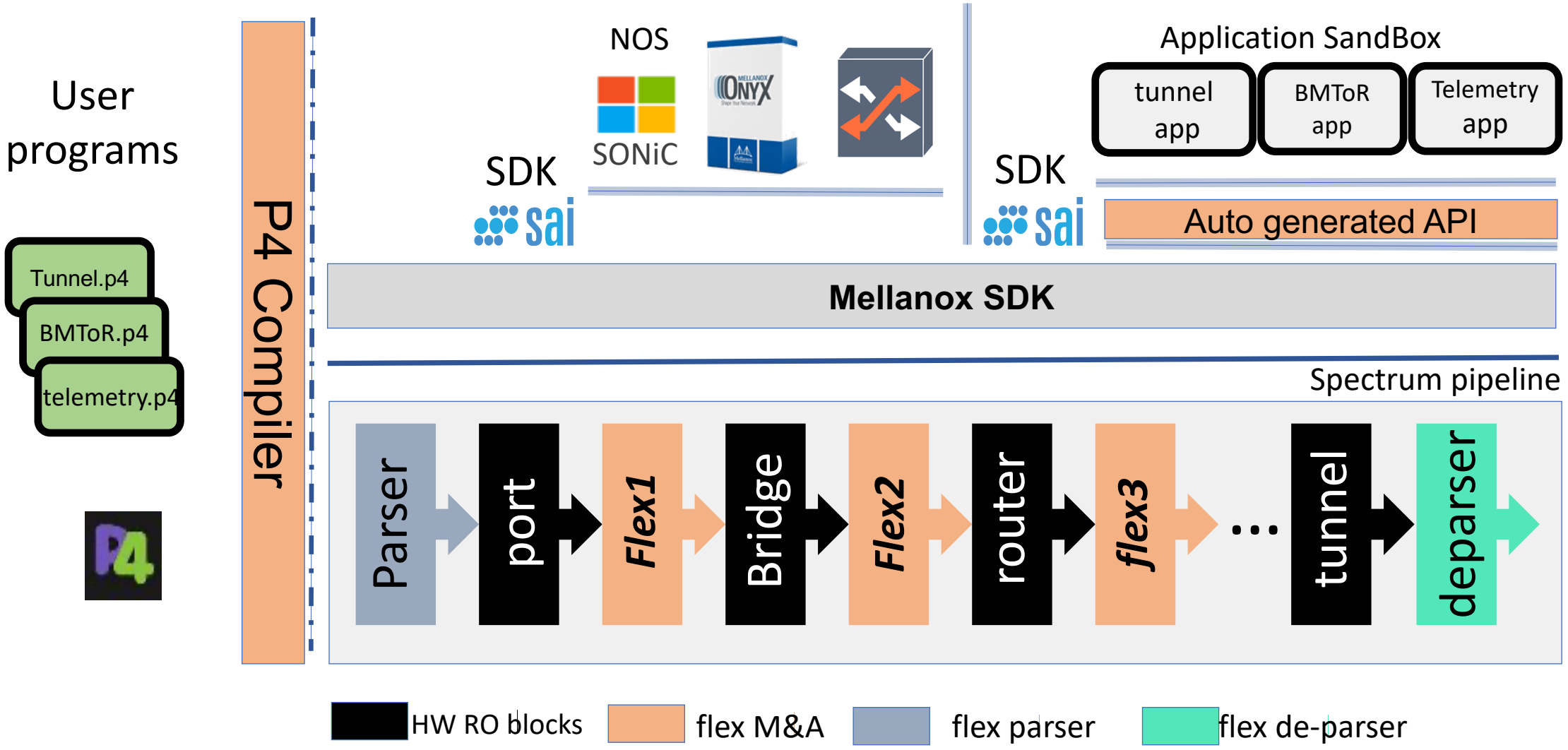
- Flexible form-factors with 16 to 128 physical ports.
- Supporting 1GbE through 400GbE.
- Based on Mellanox Spectrum-2 silicon.
- Hybrid packet forwarding concurrent capability: Legacy and Programmable.

Spectrum P4 target Architecture



- **Programmable block 1: parser**
  - Mellanox provides parsing graph base line - User will be able to add new headers to the packet-parsing graph.
- **Programmable block 2: ingress port**
  - Ability to define chain of multiple match action table. Supported actions: drop, forward to port , mirror, packet modification, tunnel encap ,tunnel decap , 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) ,tunnel encap ,tunnel decap , 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.

Hybrid Programmability



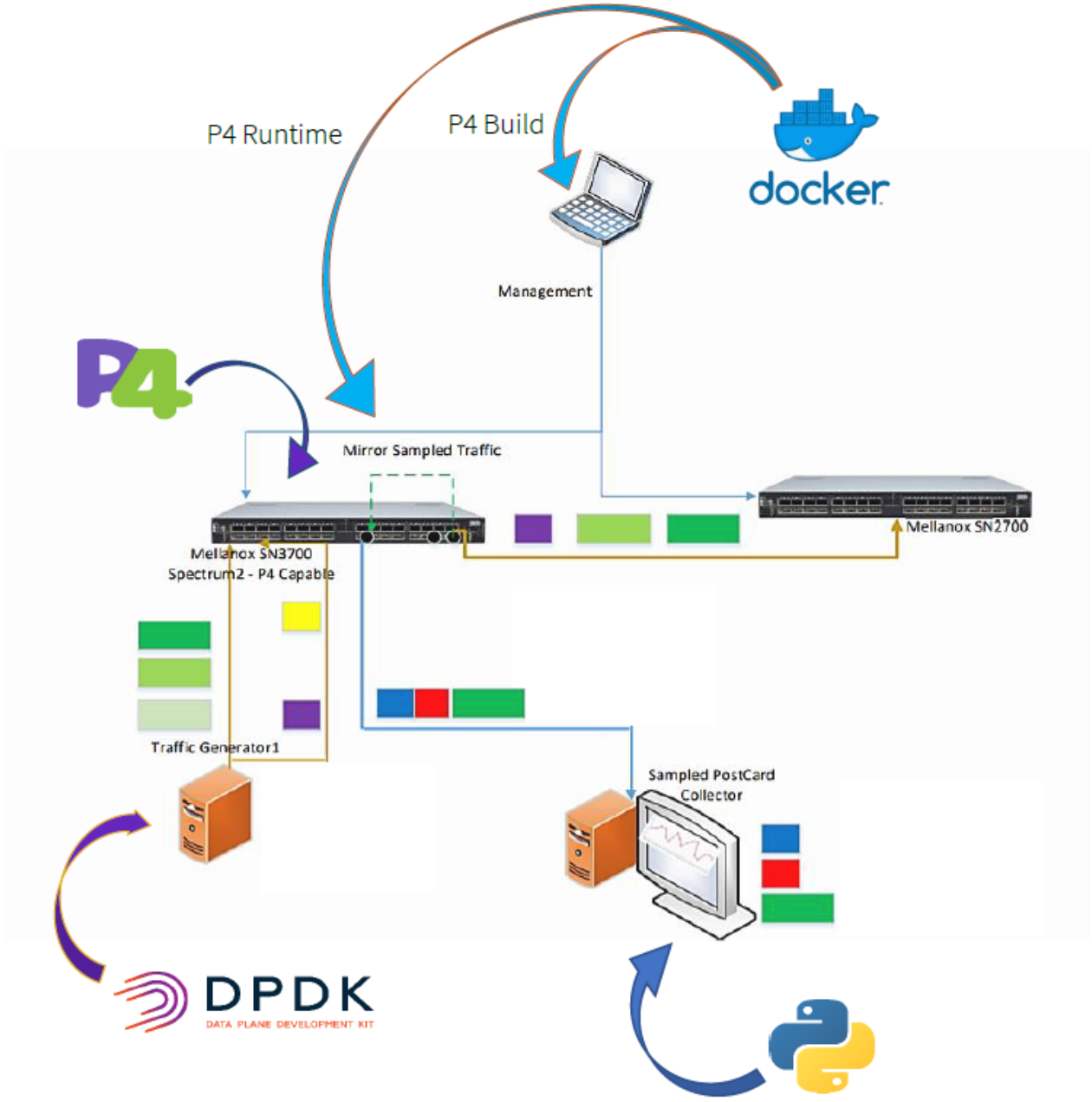
P4 Extensibility Mechanisms

- Spectrum architecture extends on P4 by providing access to hybrid mode actions as externs
  - Policy based switching
    - extern void set_pbs_port(in label_port_t pbs_port);
  - Trapping packet and send to CPU
    - extern void trap(in bit<8> trap_type, in bit<32> trap_id);
  - Mirroring packets to a remote controller using GRE
    - extern void mirror_to_remote_l3(in bit<8> session_id);
  - Setting QOS and shapers
    - extern void set_policer(in bit<64> policer_id);
- User P4 code can combine these primitives using a standard action
- A rich set of Spectrum pipeline metadata (via standard_metadata_t) is provided at various control points in the pipeline

Spectrum-2 Telemetry Header Implementation

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Offset
mirror_ethertype																ingress_label_port																00h
	raw_opcode '2'							pad_count								flags								pack-et_-type								04h
timestamp[80:48]																																08h
timestamp[47:16]																																0Ch
timestamp[15:0]																original_packet_size																10h
egress_label_port																psn																14h
mirror_header_tlvs																																
[Internal] For Spectrum-2 the TLVs are hard coded as folows:																																
'0'								ing_buff_occupancy																								18h
'1'								egr_buff_occupancy																								1Ch
'2'								latency																								20h
'3'								flags_ext																								24h
'4'								mirror_reason								'5'								tclass								28h
'6'								mirror_agent								'135' [Internal] 128+7								pg								2Ch

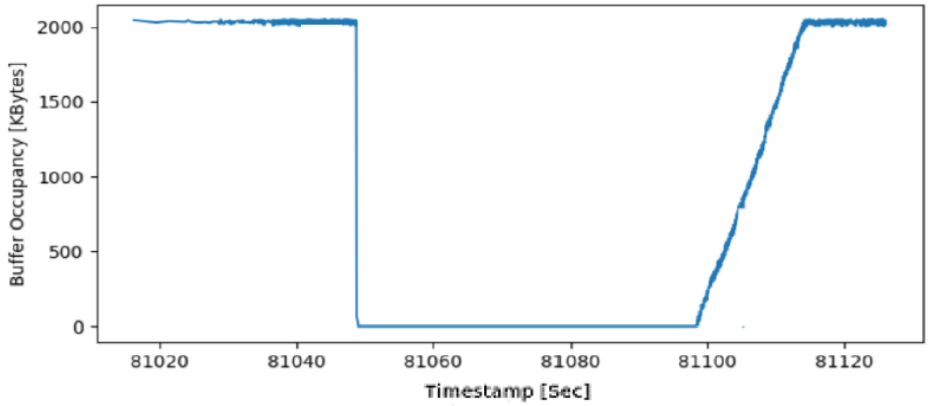
Our PoC


P4 Runtime
P4 Build
docker
Management
Mirror Sampled Traffic
Mellanox SN2700
Mellanox SN3700
Spectrum2 - P4 Capable
Traffic Generator1
Sampled PostCard
Collector
DPDK
DATA PLANE DEVELOPMENT KIT

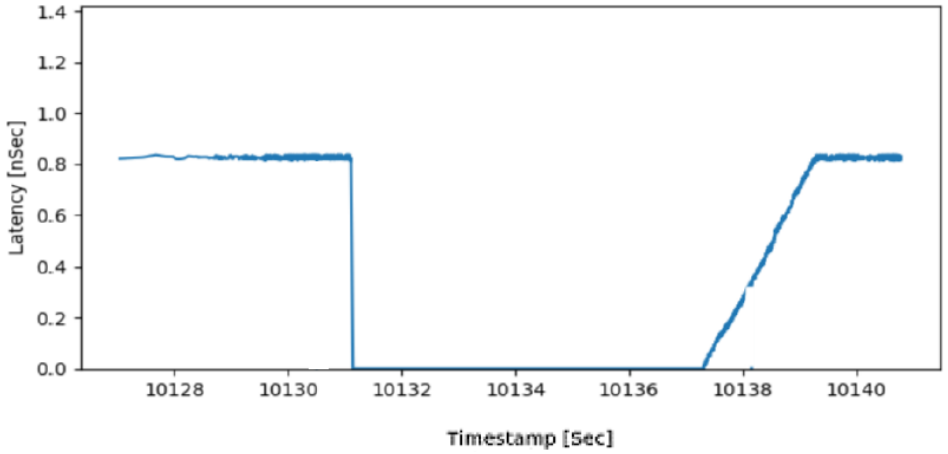
Postcard Based Telemetry on Mellanox Switch

Real-Time Graphs

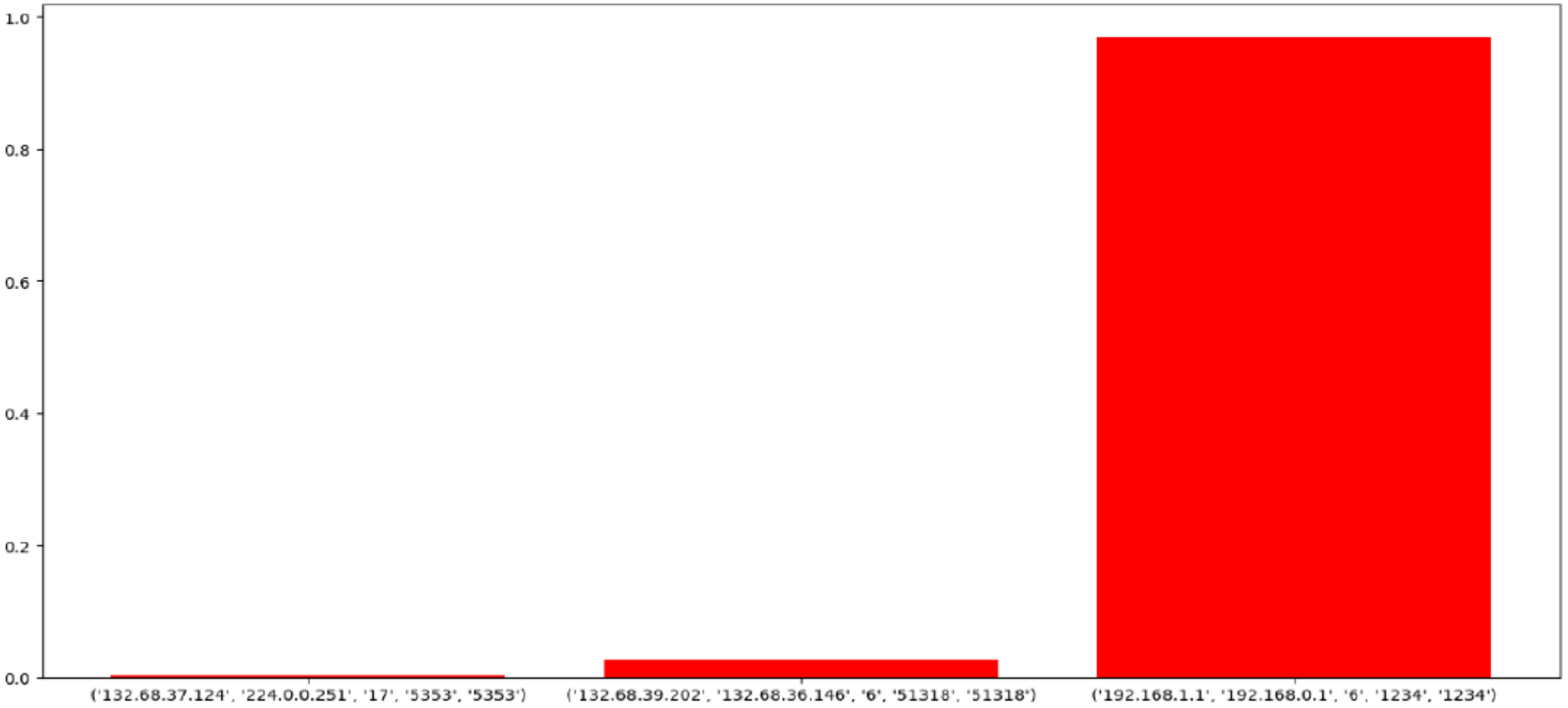
- Egress Buffer Occupancy



- Latency



Real-Time Heavy-Heater Distribution



Sponsored By

Google intel STORDIS The Open Networking Expert

Thank You

iashken@cs.technion.ac.il
Lccn.cs.technion.ac.il