KNIT 6: A FABRIC Community Workshop

Unconference Sessions

Overview of P4 Programmable Data Plane Switches

Elie Kfoury, Ali Mazloum, Jorge Crichigno
University of South Carolina
http://ce.sc.edu/cyberinfra

University of South Carolina (USC)

Texas Advanced Computing Center
Austin, TX

April 25th, 2023

Workshop Resources

- Webpage with PowerPoint presentations:

  http://ce.sc.edu/cyberinfra/knit6.html

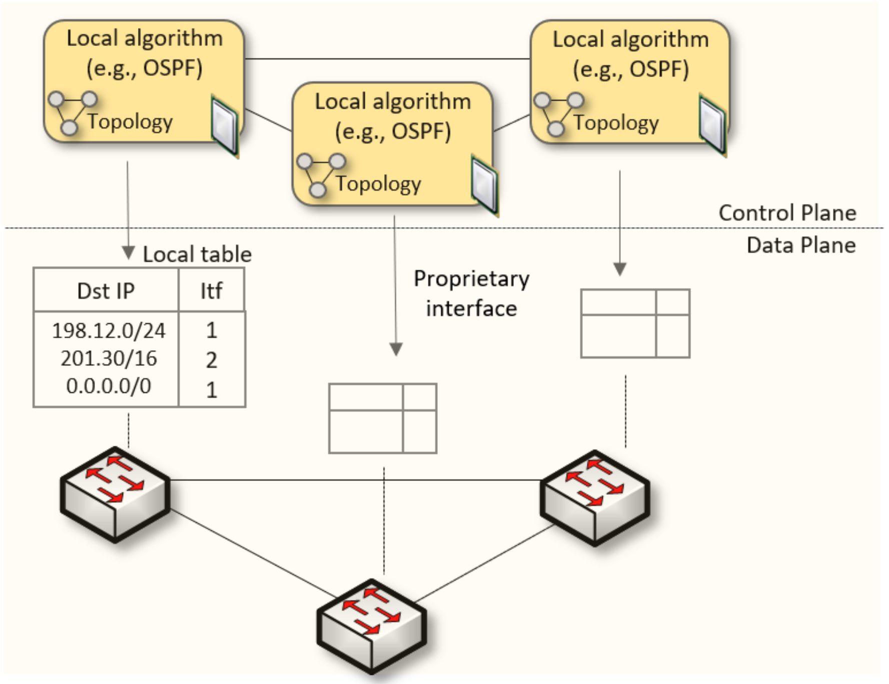
- Hands-on sessions: to access labs for the hands-on sessions, use the following link:

  https://netlab.cec.sc.edu/

- Credentials provided on-site
  - Username: guest1, guest2, …
  - Password: nsf2023 (all lowercase)

Traditional (Legacy) Networking

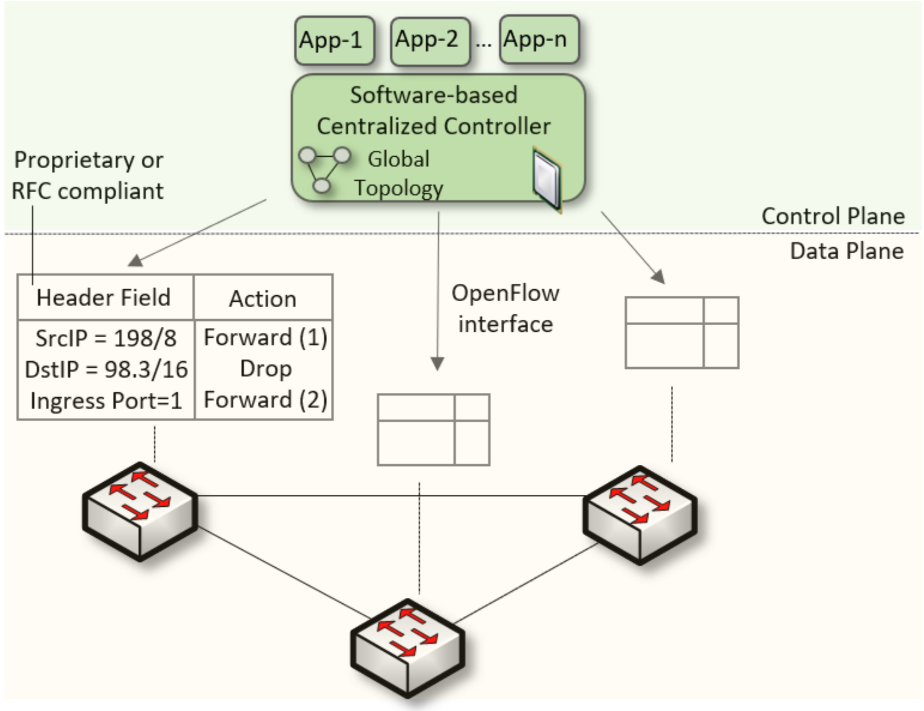
- Since the explosive growth of the Internet in the 1990s, the networking industry has been dominated by closed and proprietary hardware and software
- The interface between control and data planes has been historically proprietary
  - Vendor dependence: slow product cycles of vendor equipment, no innovation from network owners
  - A router is a monolithic unit built and internally accessed by the manufacturer only



Dst IP	Itf
198.12.0/24	1
201.30/16	2
0.0.0.0/0	1

SDN

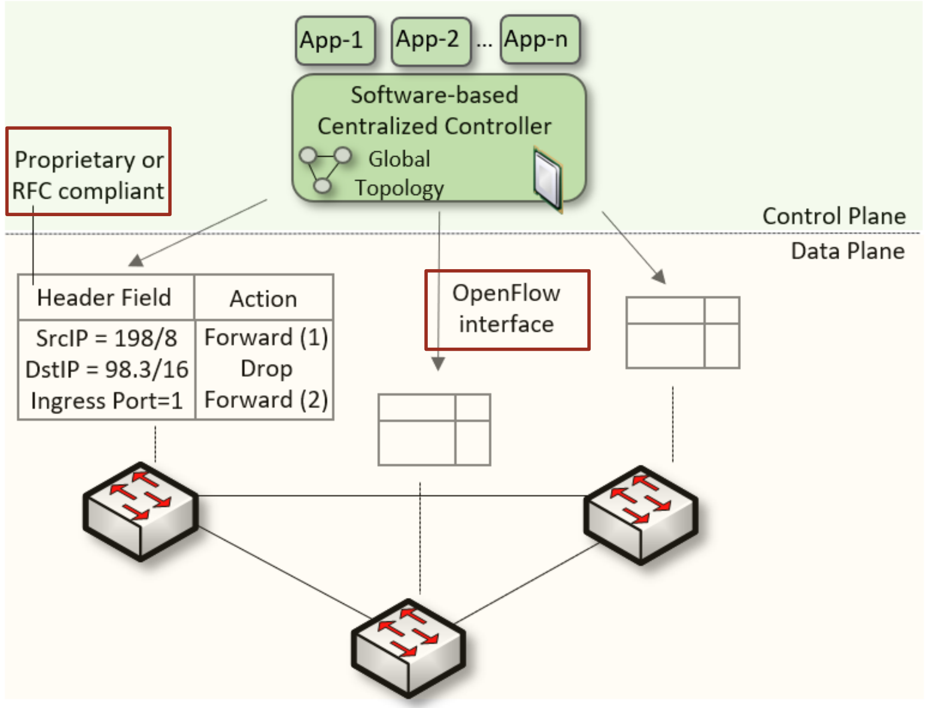
- Protocol ossification has been challenged first by SDN
- SDN (1) explicitly separates the control and data planes, and (2) enables the control plane intelligence to be implemented as a software outside the switches
- The function of populating the forwarding table is now performed by the controller



Header Field	Action
SrcIP = 198/8	Forward (1)
DstIP = 98.3/16	Drop
Ingress Port=1	Forward (2)

SDN Limitation

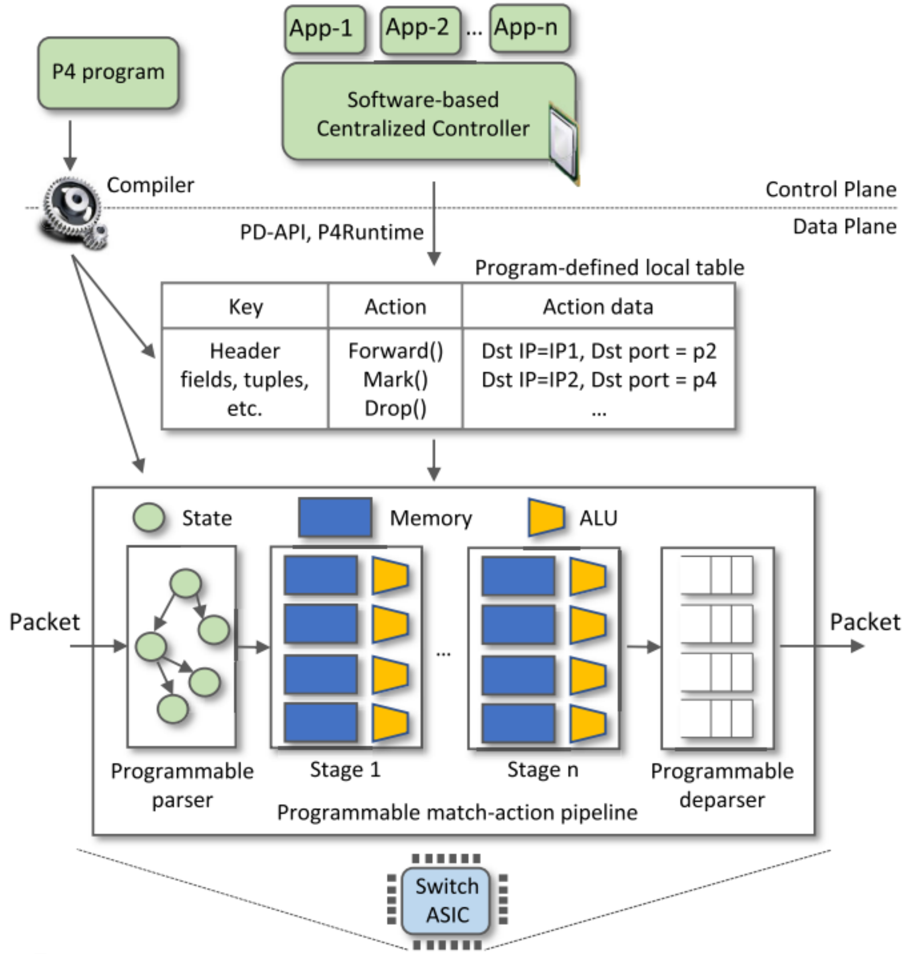
- SDN is limited to the OpenFlow specifications
    - ➢ Forwarding rules are based on a fixed number of protocols / header fields (e.g., IP, Ethernet)
- The data plane is designed with fixed functions (hard-coded)
    - ➢ Functions are implemented by the chip designer



Header Field	Action
SrcIP = 198/8	Forward (1)
DstIP = 98.3/16	Drop
Ingress Port=1	Forward (2)

P4 Programmable Switches

- P4[1] programmable switches permit a programmer to program the data plane
  - ➢ Define and parse new protocols
  - ➢ Customize packet processing functions
  - ➢ Measure events occurring in the data plane with high precision
  - ➢ Offload applications to the data plane



Program-defined local table

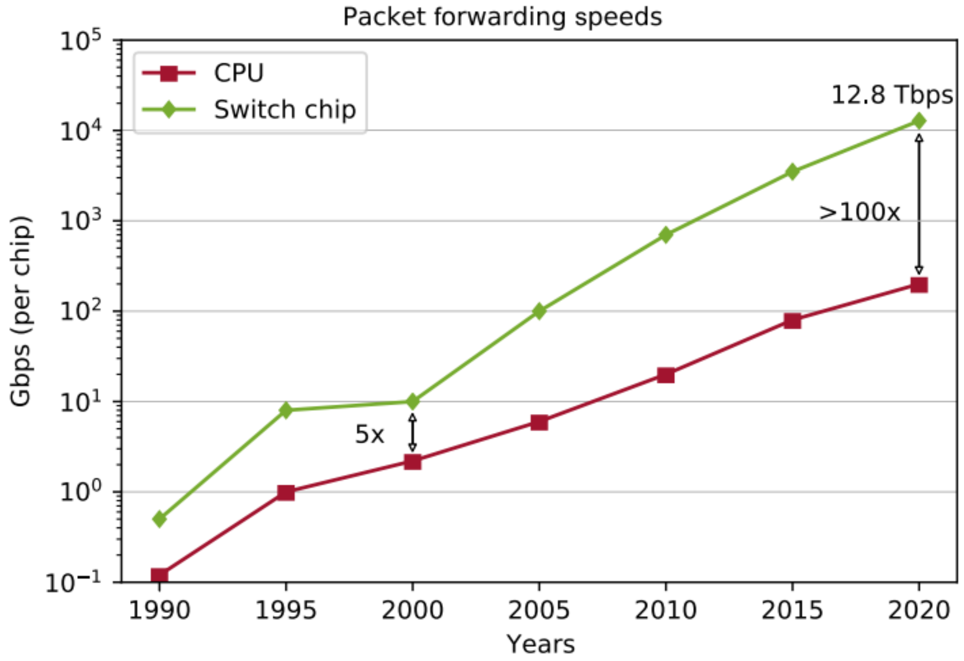
Key	Action	Action data

Header fields, tuples, etc.	Forward()
	Mark()
	Drop()
		Dst IP=IP1, Dst port = p2
		Dst IP=IP2, Dst port = p4
		...

[1] P4 stands for stands for Programming Protocol-independent Packet Processors

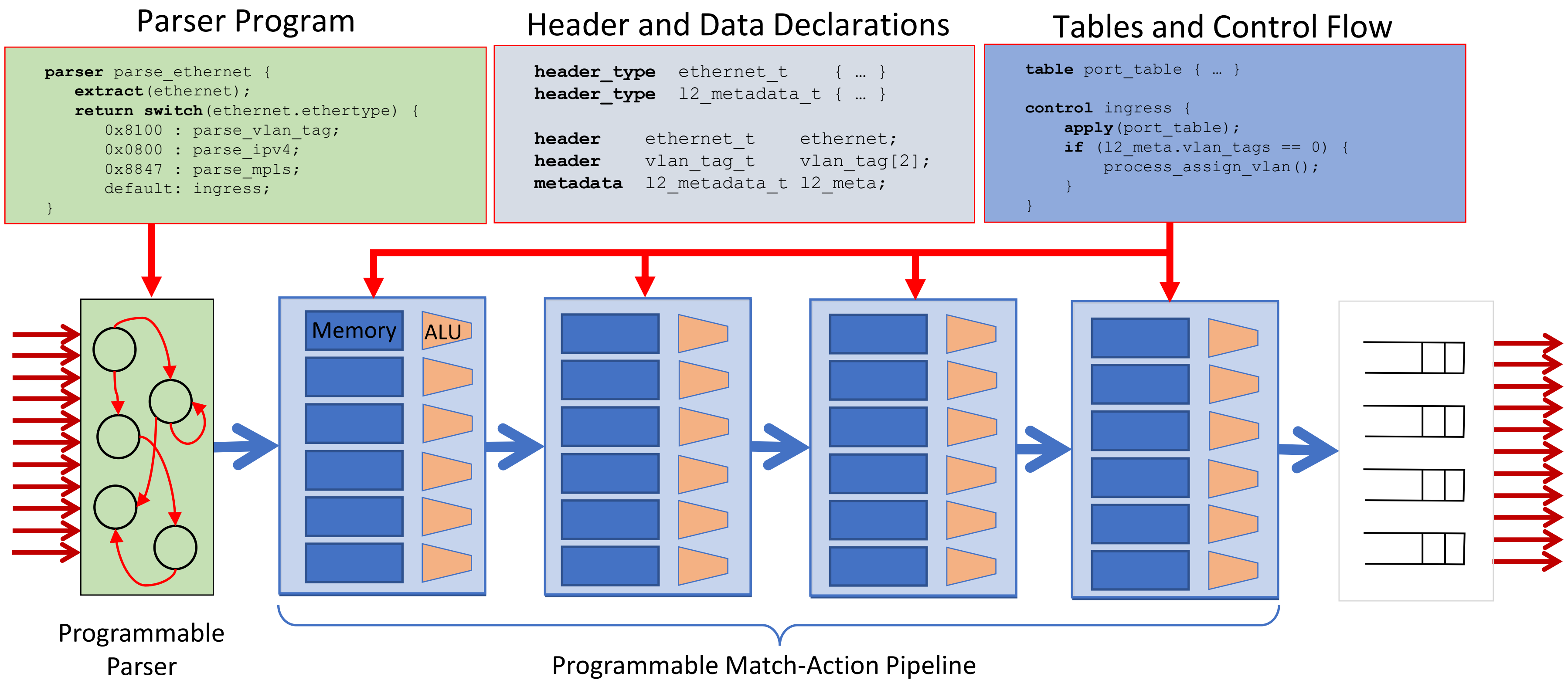
P4 Programmable Switches

- P4[1] programmable switches permit a programmer to program the data plane
  - Define and parse new protocols
  - Customize packet processing functions
  - Measure events occurring in the data plane with high precision
  - Offload applications to the data plane



Reproduced from N. McKeown. Creating an End-to-End Programming Model for Packet Forwarding. Available: **https://www.youtube.com/watch?v=fiBuao6YZI0&t=4216s**

P4 Programmable Switches



Parser Program

```
parser parse_ethernet {
    extract(ethernet);
    return switch(ethernet.ethertype) {
        0x8100 : parse_vlan_tag;
        0x0800 : parse_ipv4;
        0x8847 : parse_mpls;
        default: ingress;
}
```

Header and Data Declarations

```
header_type  ethernet_t     { … }
header_type  l2_metadata_t  { … }

header     ethernet_t     ethernet;
header     vlan_tag_t     vlan_tag[2];
metadata   l2_metadata_t  l2_meta;
```

Tables and Control Flow

```
table port_table { … }

control ingress {
    apply(port_table);
    if (l2_meta.vlan_tags == 0) {
        process_assign_vlan();
    }
}
```

Reproduced from N. McKeown. Creating an End-to-End Programming Model for Packet Forwarding. Available: **https://www.youtube.com/watch?v=fiBuao6YZI0&t=4216s**

Libraries

- A library consists of between 10-20 lab experiments
- Each lab experiment includes a detailed, step by step manual
- Once a learner completes all experiments, the learner acquires significant knowledge and hands-on expertise, and may earn an academic credential or certificate
- Information about libraries are available at http://ce.sc.edu/cyberinfra/cybertraining.html

Library on Introduction to P4 with BMv2

Experiments

- Lab 1: Introduction to Mininet
- Lab 2: Introduction to P4 and BMv2
- Lab 3: P4 Program Building Blocks
- Lab 4: Parser Implementation
- Lab 5: Introduction to Match-action Tables (Part 1)
- Lab 6: Introduction to Match-action Tables (Part 2)
- Lab 7: Populating / Managing Match-action Tables
- Lab 8: Checksum Recalculation and Deparsing

Exercises

- Exercise 1: Building a Basic Topology
- Exercise 2: Compiling and Testing a P4 Program
- Exercise 3: Parsing UDP and RTP
- Exercise 4: Building a Simplified NAT
- Exercise 5: Configuring Tables at Runtime
- Exercise 6: Building a Packet Reflector

Library on P4 Applications, Stateful Elements, and Custom Packet Processing

Experiments

- Lab 1: Introduction to Mininet
- Lab 2: Introduction to P4 and BMv2
- Lab 3: P4 Program Building Blocks
- Lab 4: Defining and processing custom headers
- Lab 5: Monitoring the Switch’s Queue using Standard Metadata
- Lab 6: Collecting Queueing Statistics using a Header Stack
- Lab 7: Measuring Flow Statistics using Direct and Indirect Counters
- Lab 8: Rerouting Traffic using Meters
- Lab 9: Storing Arbitrary Data using Registers
- Lab 10: Calculating Packets Interarrival Time w/ Hashes and Registers
- Lab 11: Generating Notification Messages from the Data Plane

Library on Security Applications with P4

Experiments

- Lab 1: Introduction to Mininet
- Lab 2: Introduction to P4 and BMv2
- Lab 3: P4 Program Building Blocks
- Lab 4: Parser Implementation
- Lab 5: Introduction to Match-action Tables
- Lab 6: Implementing a Stateful Packet Filter for the ICMP protocol
- Lab 7: Implementing a Stateful Packet Filter for the TCP protocol
- Lab 8: Detecting and Mitigating the DNS Amplification Attack
- Lab 9: Identifying Heavy Hitters using Count-min Sketches (CMS)
- Lab 10: Limiting the Impact of SYN Flood by Probabilistically Dropping Packets
- Lab 11: Blocking Application Layer Slow DDoS Attack (Slowloris)
- Lab 12: Implementing URL Filtering through Deep Packet Inspection and String Matching

Library on Network Measurement Apps with P4*

Experiments

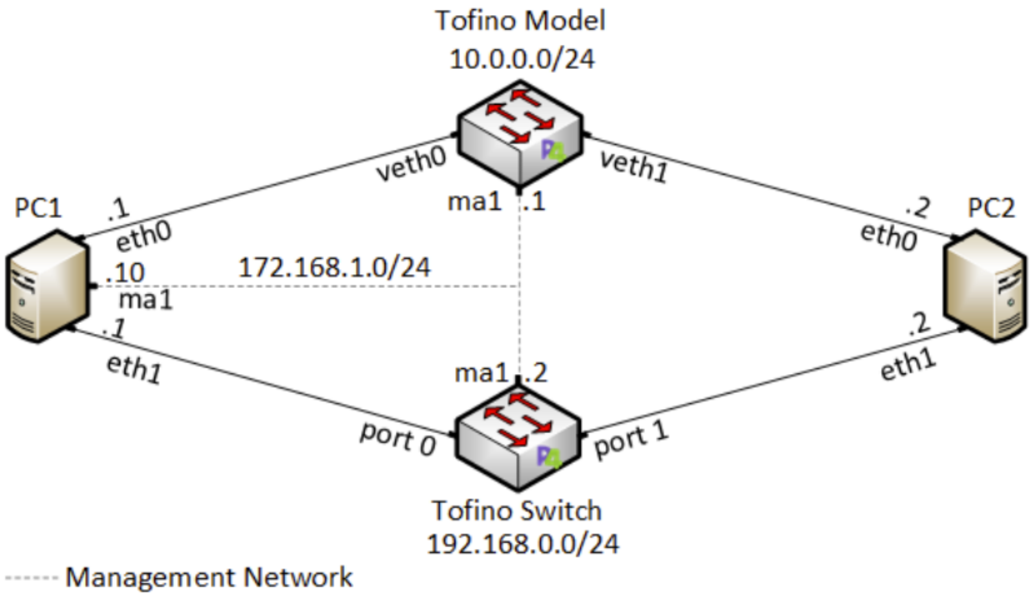
- Lab 1: Introduction to Mininet
- Lab 2: Introduction to P4 and BMv2
- Lab 3: P4 program building blocks
- Lab 4: Defining and processing custom headers
- Lab 5: Monitoring the switch’s queue using standard metadata
- Lab 6: Detecting microbursts and identify flows contributing to the queue
- Lab 7: Measuring flow statistics using direct and indirect counters
- Lab 8: Flow rate measurement and traffic policing using meters
- Lab 9: Identifying elephant flows using count-min sketches (CMS)
- Lab 10: Using In-band Network Telemetry (INT) to track the path taken by packets
- Lab 11: Measuring the round-trip time of flows
- Lab 12: Measuring the packet loss rate
- Lab 13: Measuring packets’ inter-arrival times
- Lab 14: Identifying TCP congestion control type using inter-arrival measurements
- Lab 15: Plotting measurements from P4 on a Grafana dashboard

* Expected by July 2023

Library on P4 Programmable Data Plane with Tofino

Experiments

- Lab 1: Introduction to P4 and Tofino
- Lab 2: Introduction to P4 Tofino Software Development Environment
- Lab 3: Parser Implementation
- Lab 4: Introduction to Match-Action Tables
- Lab 5: Populating and Managing Match-Action Tables at Runtime
- Lab 6: Checksum Recalculation and Packet Deparsing



Organization of the Lab Manuals

Each lab starts with a section *Overview*

- Objectives
- Lab topology
- Lab settings: passwords, device names
- Roadmap: organization of the lab

Section 1

- Background information of the topic being covered (e.g., fundamentals of perfSONAR)
- Section 1 is optional (i.e., the reader can skip this section and move to lab directions)

Section 2… n

- Step-by-step directions

UNIVERSITY OF
South Carolina

This work is supported by NSF award number 2118311

For additional information, please refer to
http://ce.sc.edu/cyberinfra/

Email: jcrichigno@cec.sc.edu, ekfoury@email.sc.edu