Hands-on Workshop on IPv6 and CI Training

Hands-on session 5: Configuring perfSONAR
Lab 11 – perfSONAR 5 Lab Series

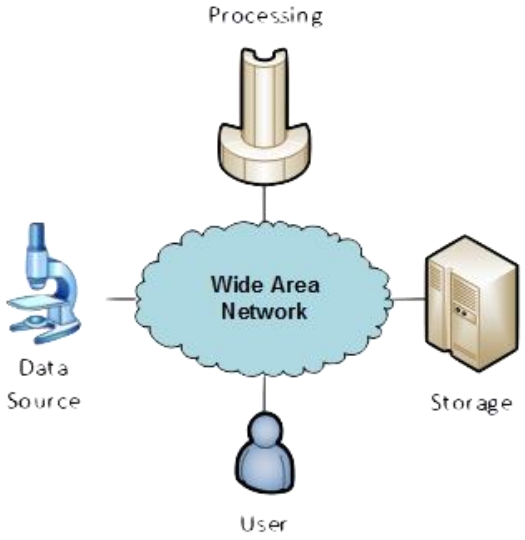
Jorge Crichigno
University of South Carolina
https://research.cec.sc.edu/cyberinfra/

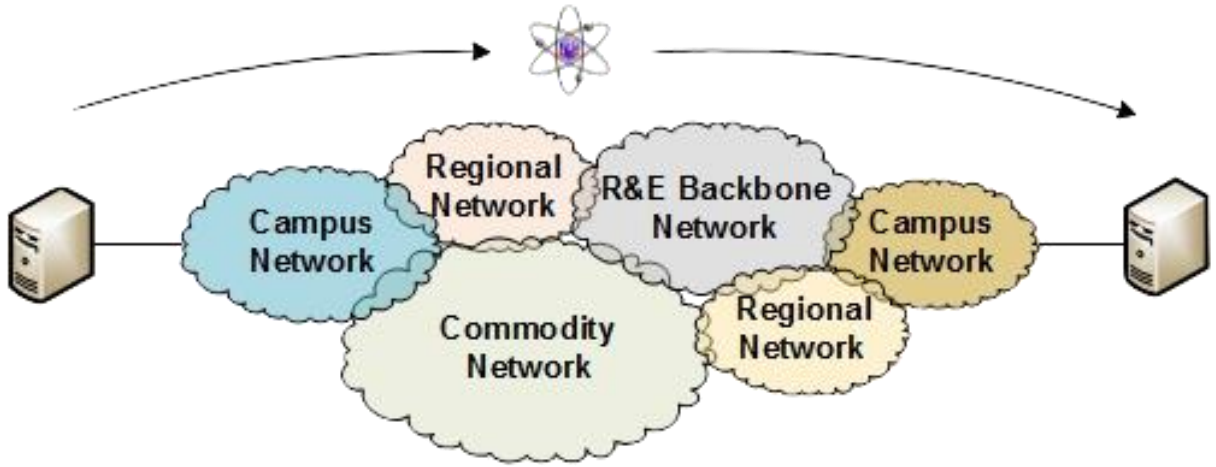
University of South Carolina (USC)

Thursday Sep. 12, 2024
Colorado State University

Motivation

- The global Research & Education network ecosystem is comprised of hundreds of international, national, regional, and local-scale resources
- Each of them is owned and operated independently
- This complex, heterogeneous set of networks must operate seamlessly from “end to end” to support science and research collaborations
- Typically, this type of collaboration is distributed globally

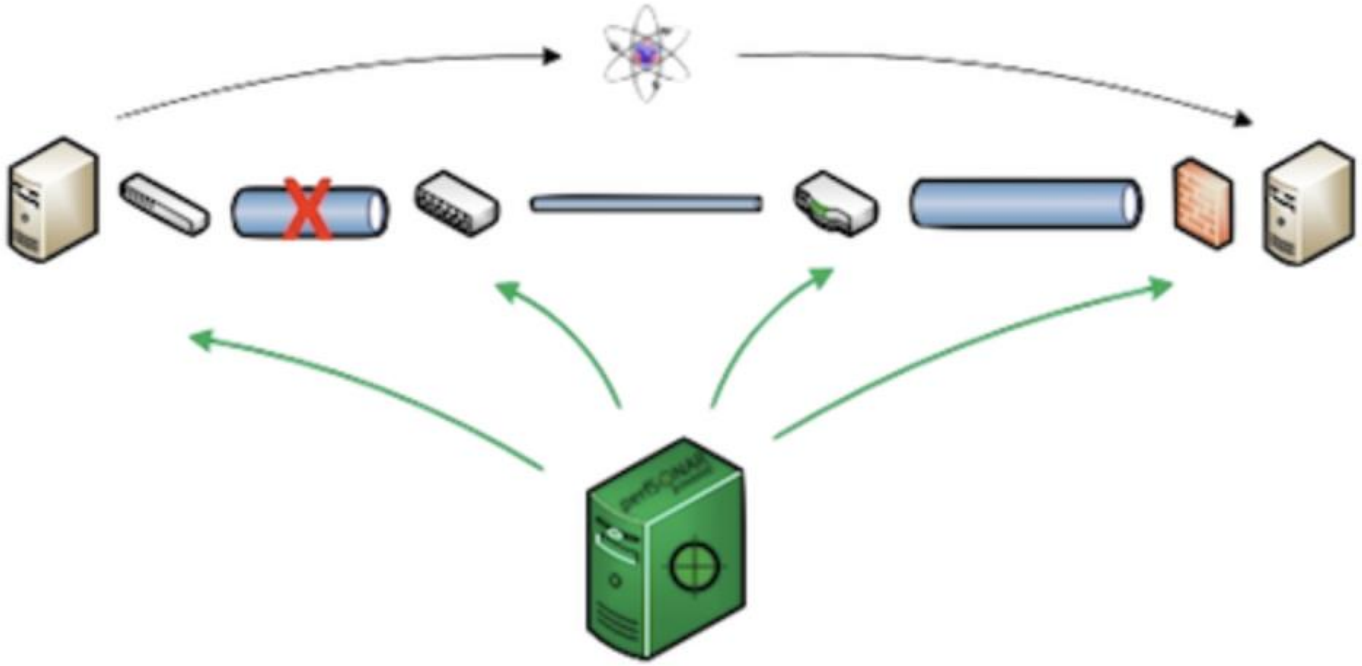




Source: ESNet, “*Monitoring end-to-end systems*”, [Online]. Available: https://tinyurl.com/4tppsccv

Motivation

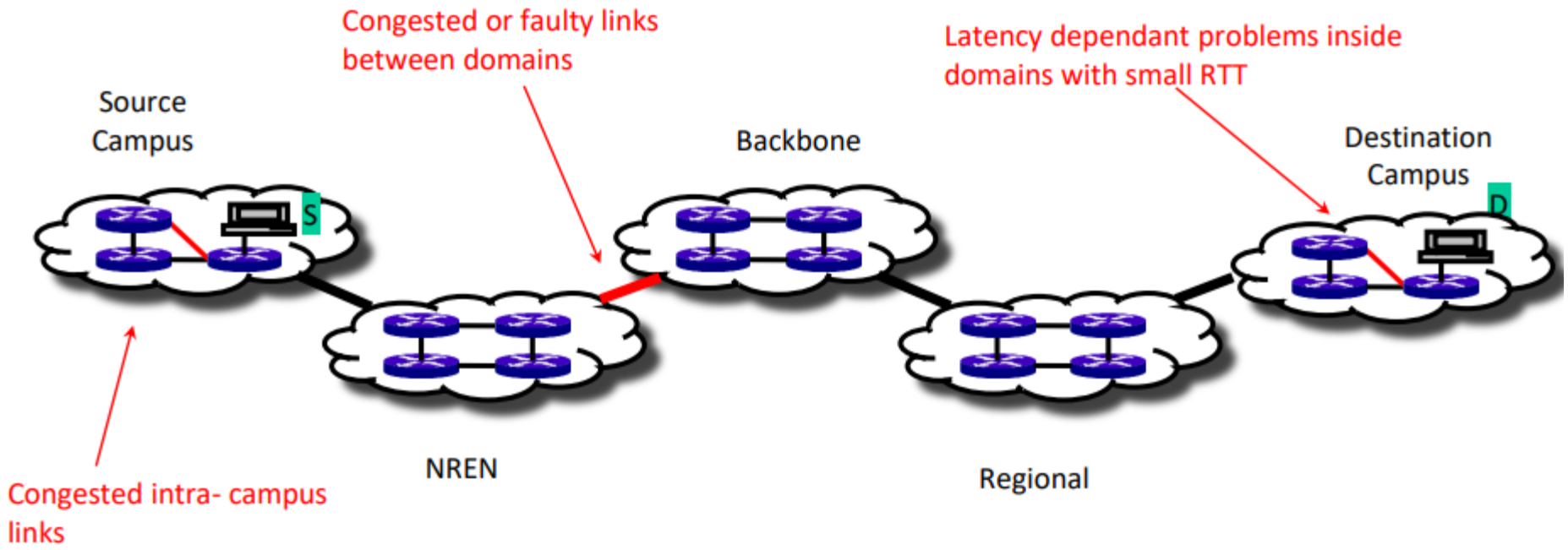
- Organizations must understand the behavior of their network by monitoring the performance metrics to ensure that the underlying system is functional



Source: ESNet, "*Monitoring end-to-end systems*", [Online]. Available: https://tinyurl.com/4tppsccv

Motivation

- Network issues can have different sources and locations
- Performing local testing will not find the cause of these problems



Source: ESNet, "*Monitoring end-to-end systems*", [Online]. Available: https://tinyurl.com/4tppsccv

Soft Network Failures

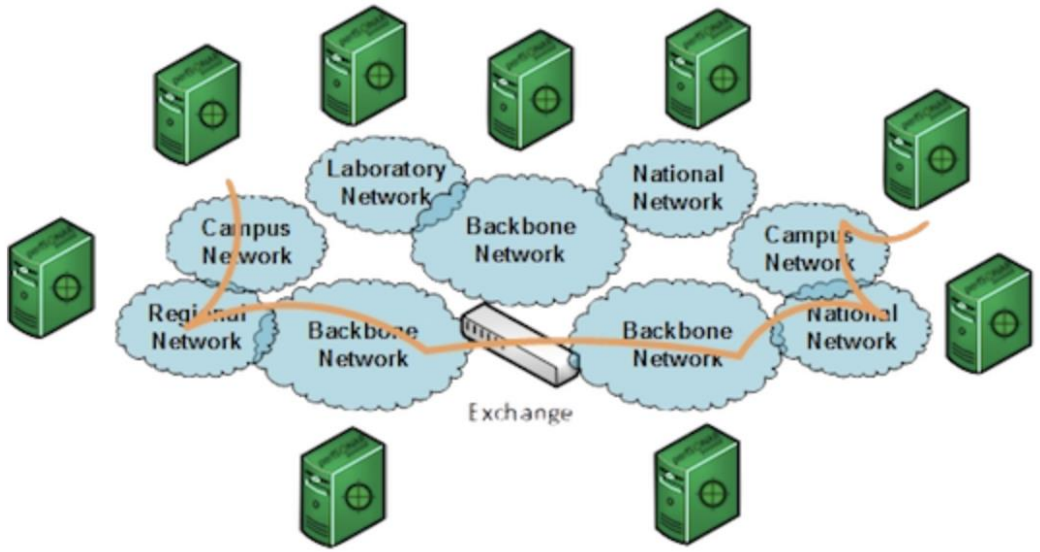
- Soft failures affect basic connectivity functions (e.g., long delays, packet losses)
- Some soft failures may only affect high bandwidth long RTT flows
- TCP was intentionally designed to hide transmission errors from the user
- Soft failures are difficult to detect and fix
- They can be hidden for years and cause resource underutilization

Hard Network Failures

- On the other hand, hard failures are easier to detect and fix
- These types of failures are easy to understand
  - Fiber cut
  - Power failure takes down routers
  - Hardware malfunction
- Classic monitoring systems are good at alerting hard failures
- For example, the network operator visualizes an alert in the system's dashboard

perfSONAR

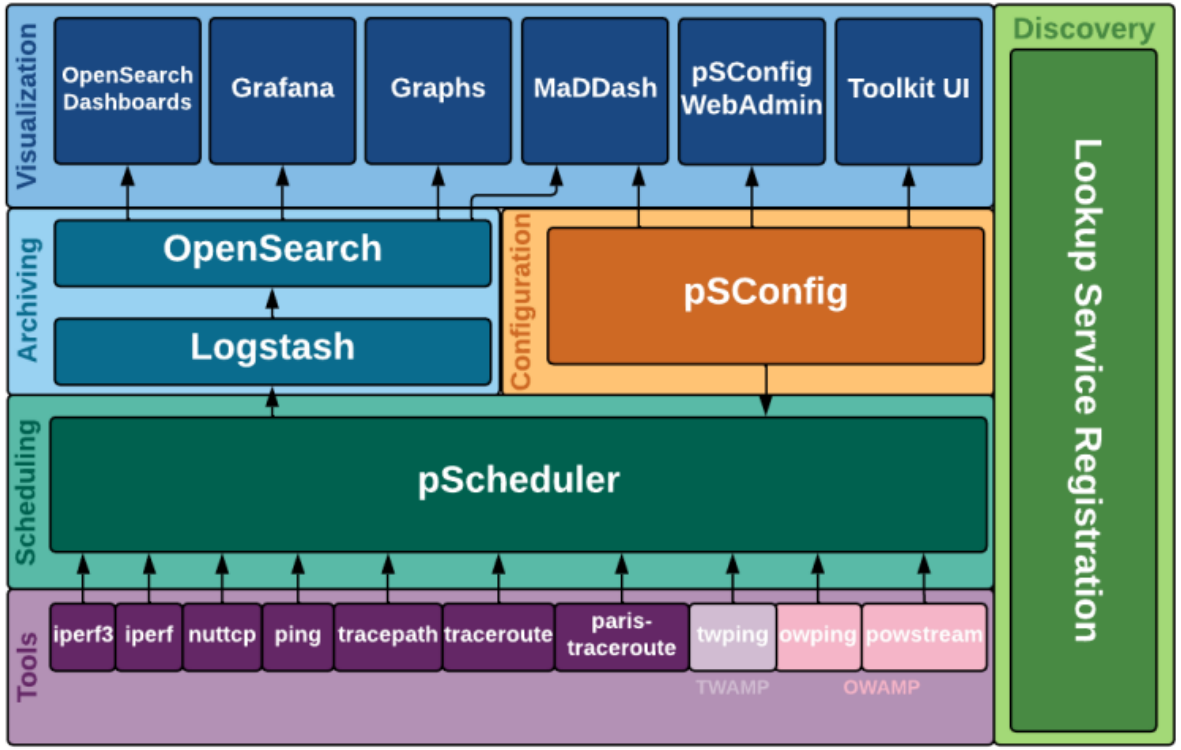
- perfSONAR is a network measurement tool designed to provide federated coverage of paths and help to achieve end-to-end usage expectations
- The tool facilitates diagnosing, visualizing, and troubleshooting network performance issues
- perfSONAR can collect metrics such as throughput, latency, and packet losses



Source: ESNet, "*Monitoring end-to-end systems*", [Online]. Available: https://tinyurl.com/4tppsccv

perfSONAR

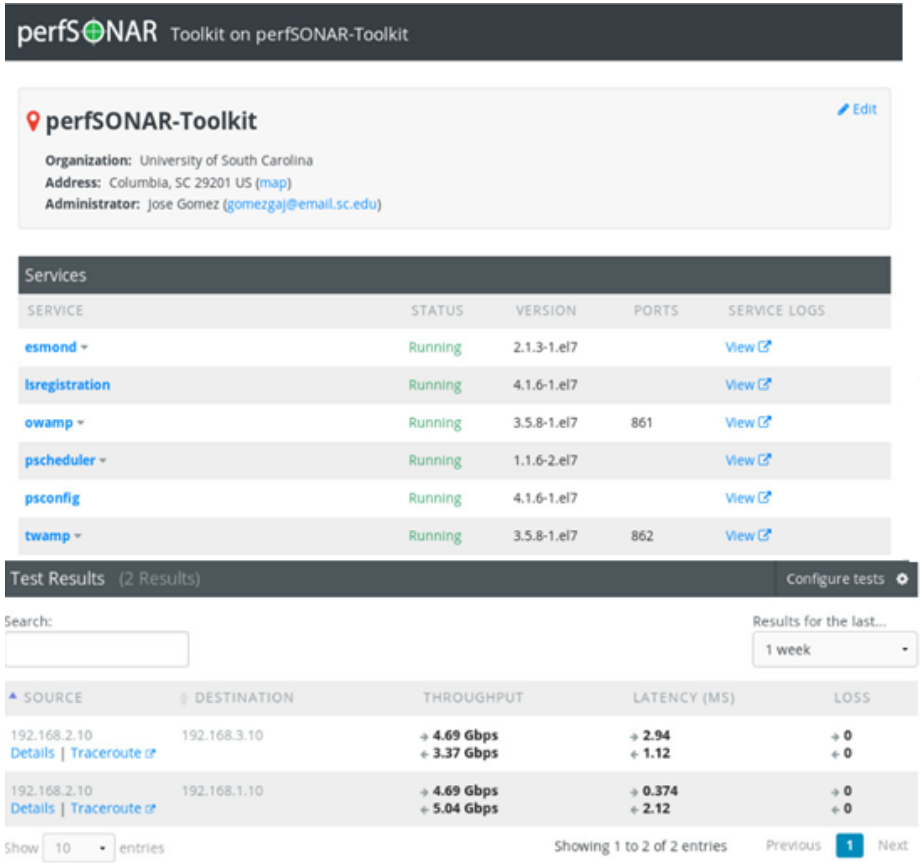
- perfSONAR provides a set of resources to orchestrate regular tests using open-source tools such as ping, traceroute, iperf3, and others



perfSONAR layers

perfSONAR

- perfSONAR allows scheduling measurements, storage of data in uniform formats, and methods to retrieve data and generate visualizations



perfSONAR Toolkit GUI

perfSONAR test results

perfSONAR 5 Lab Library

- Lab experiments
  - Lab 1: Introduction to Mininet
  - Lab 2: Setting Administrative Information via perfSONAR Toolkit GUI
  - Lab 3: Scheduling Regular Tests Using perfSONAR GUI
  - Lab 4: Configuring Regular Tests Using pScheduler CLI Part I (throughput, latency, and traceroute)
  - Lab 5: Configuring Regular Tests Using pScheduler CLI Part II (repeat, store, monitor, and cancel)
  - Lab 6: Defining Regular Tests with a pSConfig Template
  - Lab 7: Configuring pScheduler Limits
  - Lab 8: Visualizing pScheduler Measurements using Grafana
  - Lab 9: Observing the Impact of TCP Window Scaling and Small TCP Buffer Sizes
  - Lab 10: Investigating the Effects of MTU Mismatch
  - Lab 11: Running Regular pScheduler Tests over IPv6 Networks

Running Regular pScheduler Tests over IPv6 Networks

Lab activities are described in Lab 11, Introduction to perfSONAR 5 lab series

Organization of the labs

Each lab starts with a section *Overview*

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

Section 1

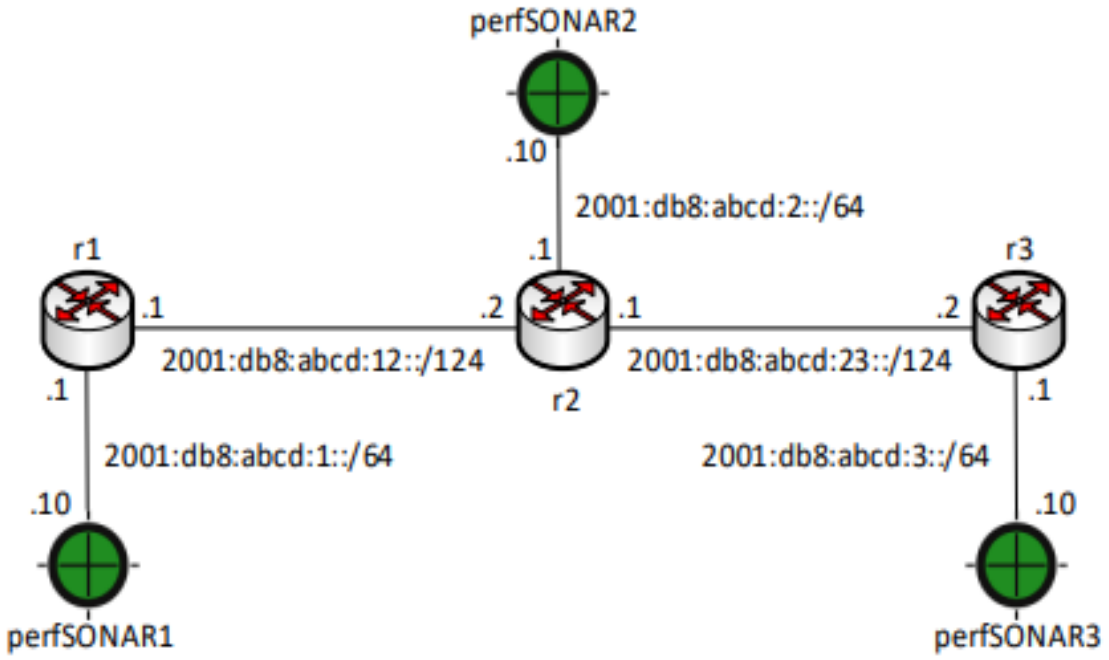
- Background information of the topic being covered (e.g., fundamentals of perfSONAR)
- Section 1 is optional (i.e., the reader can skip this section and move to lab directions)

Section 2… n

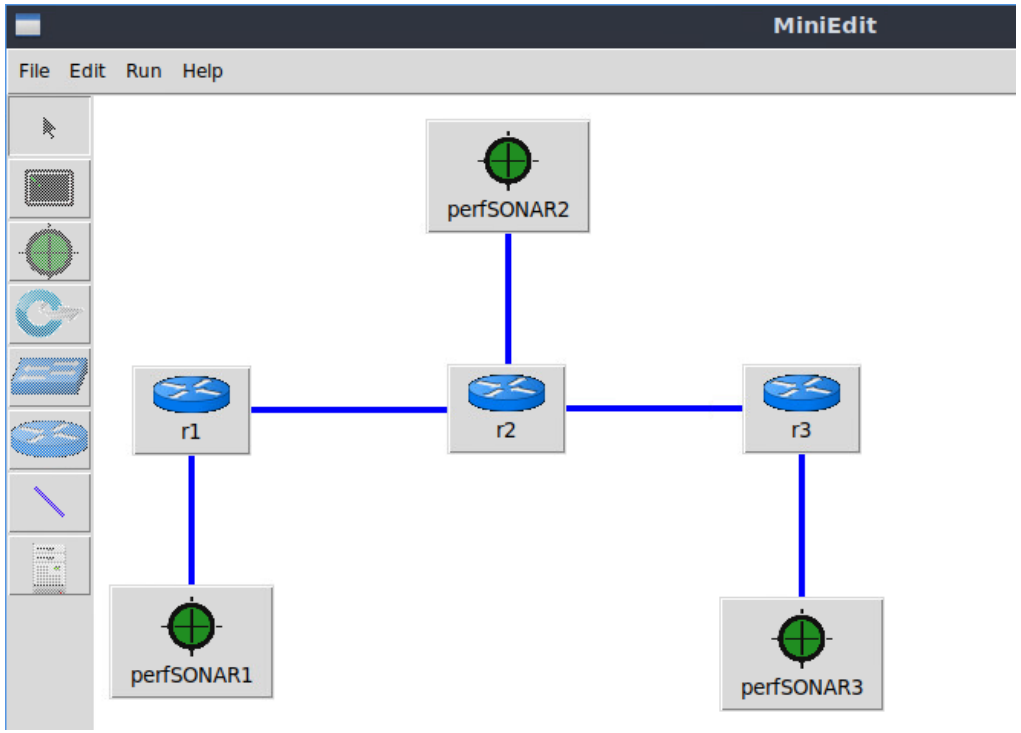
- Step-by-step directions

Lab Description

- This lab shows how to use pScheduler to run latency, throughput, and trace tests over IPv6 networks
- The lab shows how to use default and specific perfSONAR tools
- The user will interact with a perfSONAR node via the CLI



Lab scenario

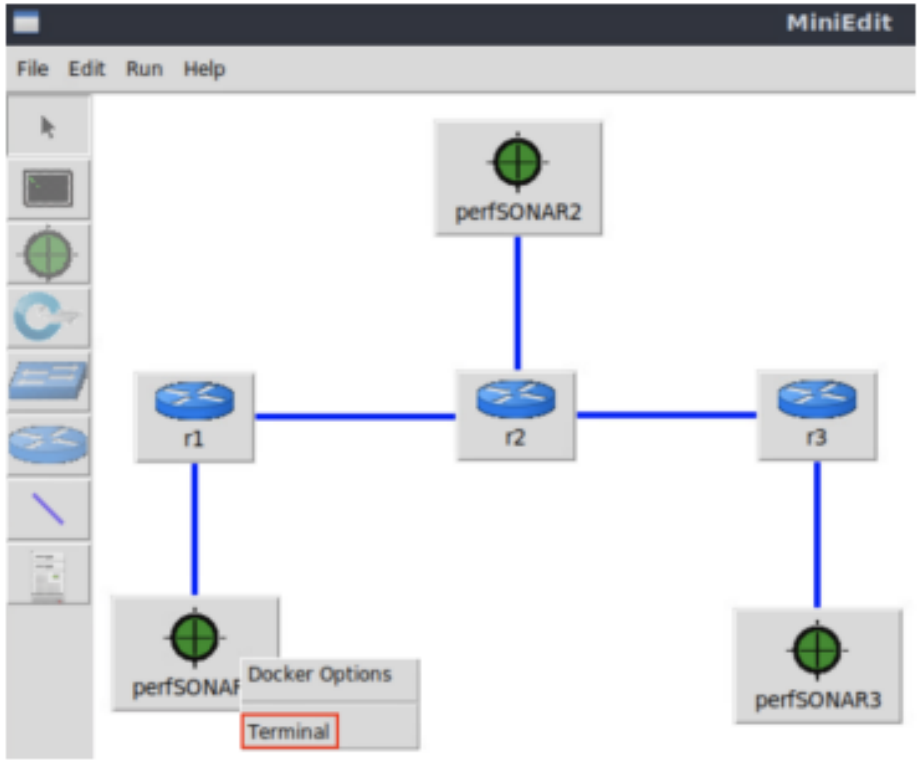


Mininet topology

Lab Description

- The user can coordinate regular test between local and remote nodes for troubleshooting purposes

```
pscheduler task throughput --source 2001:db8:abcd:1::10 --dest
2001:db8:abcd:2::10
```



Accessing perfSONAR's CLI

```
@perfSONAR1:/
[root@perfSONAR1 /]# pscheduler task throughput --source 2001:db8:abcd:1::10 --dest 2001:db8:abcd:2::10
Submitting task...
Task URL:
https://[2001:db8:abcd:1::10]/pscheduler/tasks/2d4a90df-126d-4bc4-9758-2d30bf611117
Running with tool 'iperf3'
Fetching first run...

Next scheduled run:
https://[2001:db8:abcd:1::10]/pscheduler/tasks/2d4a90df-126d-4bc4-9758-2d30bf611117/runs/ea62498e-d031-4244-8558
Starts 2024-01-24T20:32:21+00:00 (~5 seconds)
Ends   2024-01-24T20:32:40+00:00 (~18 seconds)
Waiting for result...

* Stream ID 5
Interval       Throughput     Retransmits    Current Window
0.0 - 1.0      959.20 Mbps    180            3.66 MBytes
1.0 - 2.0      943.89 Mbps    0              3.93 MBytes
2.0 - 3.0      943.62 Mbps    0              4.16 MBytes
3.0 - 4.0      943.82 Mbps    0              4.35 MBytes
4.0 - 5.0      943.72 Mbps    45             3.19 MBytes
5.0 - 6.0      943.72 Mbps    0              3.34 MBytes
6.0 - 7.0      943.71 Mbps    0              3.46 MBytes
7.0 - 8.0      943.72 Mbps    0              3.56 MBytes
8.0 - 9.0      943.72 Mbps    0              3.63 MBytes
9.0 - 10.0     943.71 Mbps    0              3.68 MBytes

Summary
Interval       Throughput     Retransmits    Receiver Throughput
0.0 - 10.0     945.28 Mbps    225            928.75 Mbps

No further runs scheduled.
[root@perfSONAR1 /]#
```

Running a regular test with pScheduler over IPv6

P4 + perfSONAR

- USC has deployed a monitoring system that consists of perfSONAR and a P4 programmable switch or programmable data plane (PDP)

1. P4 stands for Programming Protocol-independent Packet Processors

P4 + perfSONAR

- USC has deployed a monitoring system that consists of perfSONAR and a P4[1] programmable switch or programmable data plane (PDP)
- A P4 PDP is a domain-specific processor for networking
- It enables the programmer to
  - define and parse new protocols
  - measure events with high precision
  - run custom applications at line rate

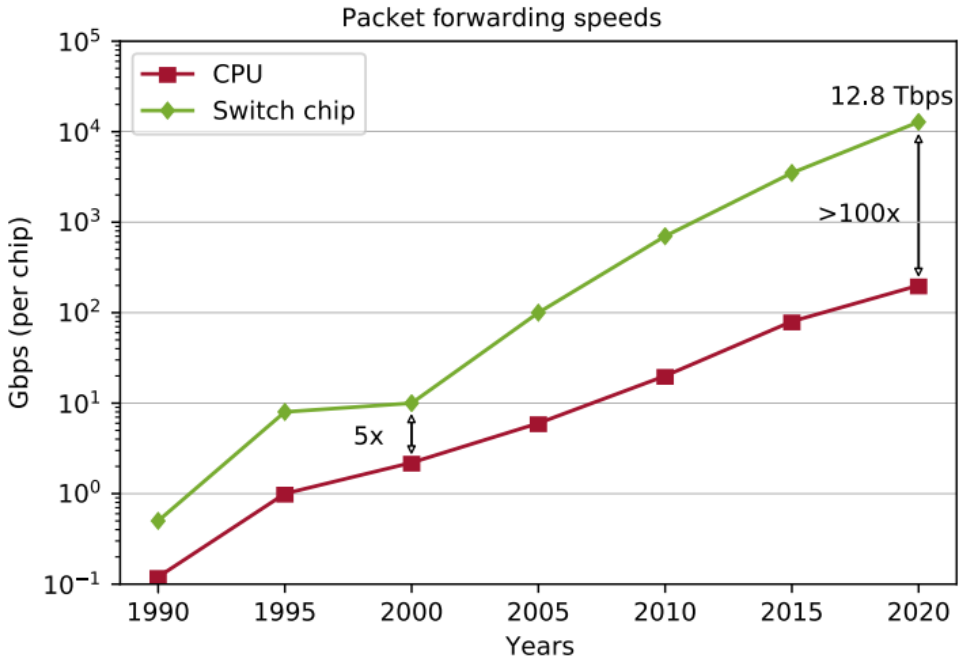
```
136  /*************************************************************************
137  ********************** P A R S E R  ***********************************
138  *************************************************************************/
139
140      state parse_ethernet {
141          packet.extract(hdr.ethernet);
142          transition select(hdr.ethernet.etherType) {
143              TYPE_IPV4: parse_ipv4;
144              default: accept;
145          }
146      }
147
148      state parse_ipv4 {
149          packet.extract(hdr.ipv4);
150          verify(hdr.ipv4.ihl >= 5, error.IPHeaderTooShort);
151          transition select(hdr.ipv4.ihl) {
152              5             : accept;
153              default       : parse_ipv4_option;
154          }
155      }
```

P4 code

1. P4 stands for Programming Protocol-independent Packet Processors

P4 + perfSONAR

- USC has deployed a monitoring system that consists of perfSONAR and a P4 programmable switch or programmable data plane (PDP)
- A P4 PDP is a domain-specific processor for networking
- It enables the programmer to
  - define and parse new protocols
  - measure events with high precision
  - run custom applications at line rate

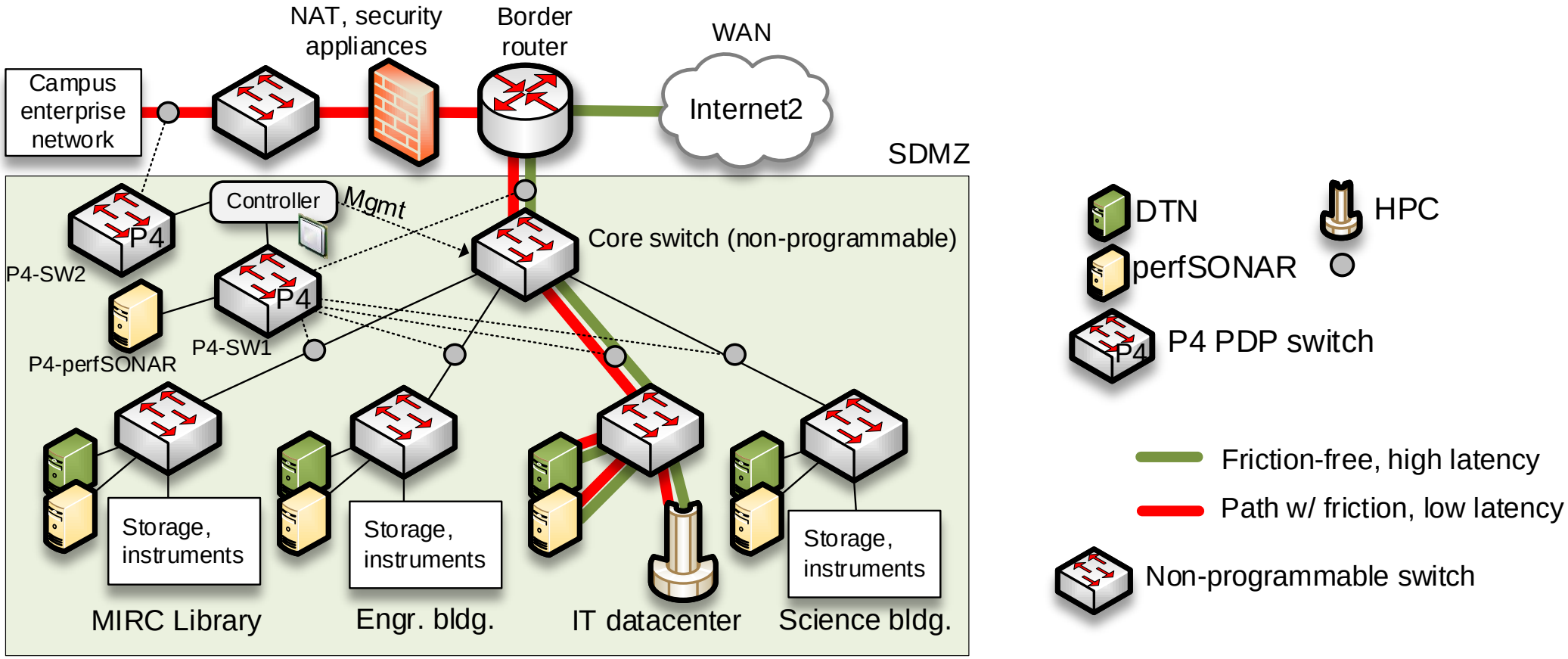


Evolution of the packet forwarding speeds[1]

1. Reproduced from N. McKeown. Creating an End-to-End Programming Model for Packet Forwarding. Available: https://www.youtube.com/watch?v=fiBuao6YZl0&t=634s

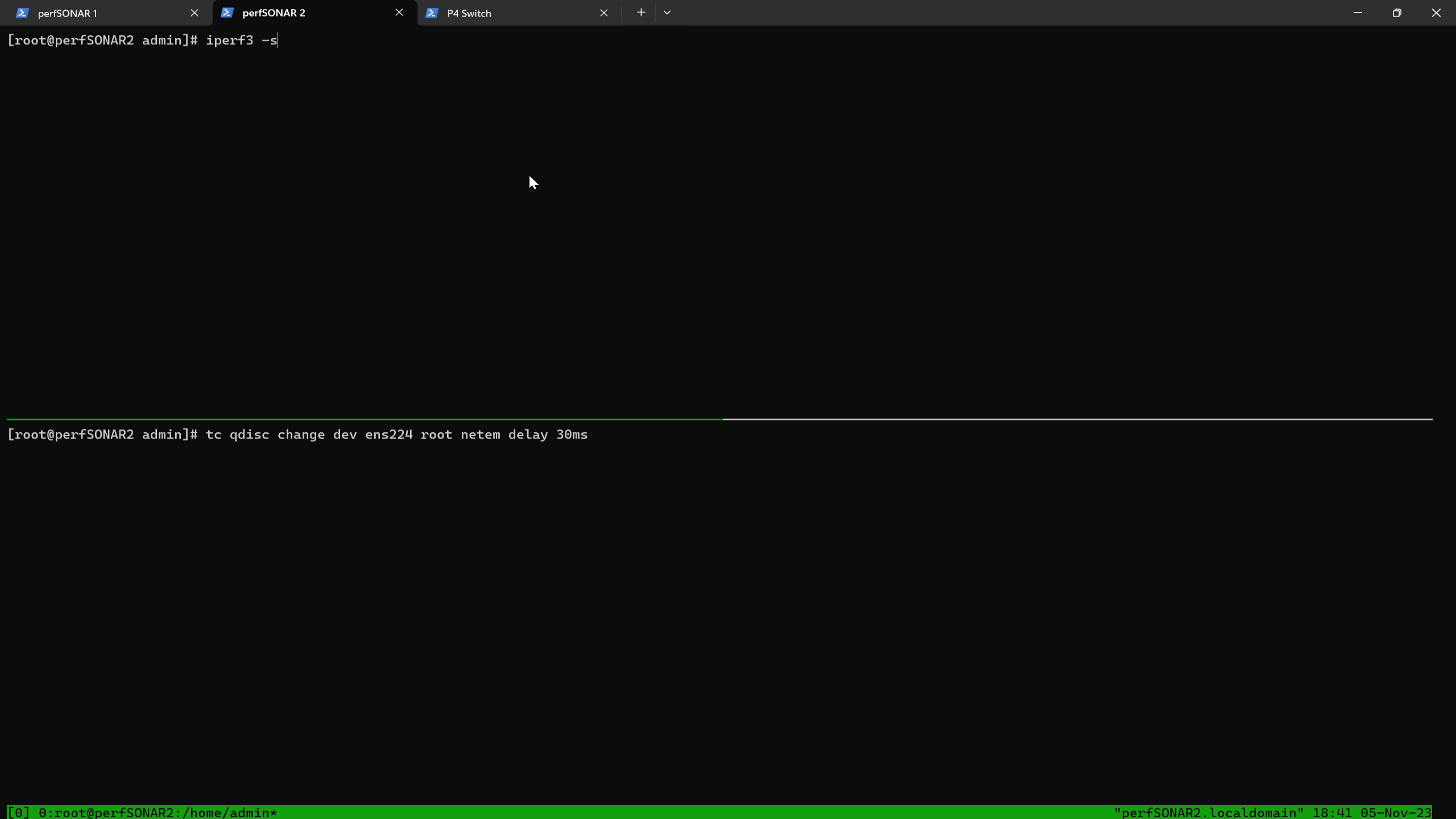
P4 + perfSONAR

- The project is supported by NSF CC*, area 3: “Network Integration and Applied Innovation”



Demo 1
Throughput Measurements

Demo 2
Throughput and RTT Measurements


perfSONAR 1
perfSONAR 2
P4 Switch
[root@perfSONAR2 admin]# iperf3 -s
[root@perfSONAR2 admin]# tc qdisc change dev ens224 root netem delay 30ms
[0] 0:root@perfSONAR2:/home/admin*
"perfSONAR2.localdomain" 18:41 05-Nov-23

Labs on perfSONAR and P4

- The project is supported by NSF CC*, area 3: “Network Integration and Applied Innovation”

Lab 1	Introduction to Mininet
Lab 2	P4 Program Building Blocks
Lab 3	Measuring Flow's Throughput
Lab 4	Monitoring the RTT of TCP Flows using P4
Lab 5	Configuring Regular Tests Using pSchedular CLI
Lab 6	Connecting perfSONAR to Grafana Dashboard
Lab 7	Retrieving Per-flow Statistics from the Data Plane
Lab 8	Collecting P4 Measurements using perfSONAR's Archiver