Science DMZs and Networking for All

Jorge Crichigno, Elie Kfoury, Jose Gomez
Cyberinfrastructure Lab
College of Engineering and Computing, University of South Carolina
http://ce.sc.edu/cyberinfra

Nikunja Swain
South Carolina State University

MS-CC All Hands Meeting
July 27, 2023 – Online

Tutorial on Science DMZs and Networking for All

- We are organizing a tutorial on Science DMZs and other network-related topics
- The tutorial will be co-located to Internet2 Technology Exchange Conference, September 18-22, 2023

  https://internet2.edu/2023-internet2-technology-exchange/

Tutorial on Science DMZs and Networking for All

Goals

- Understand the network elements required for high-performance data transfers
- Describe the key elements of a Science DMZ
- Measure the performance of different TCP congestion control algorithms on high-throughput high-latency networks
- Describe the operation of perfSONAR and use perfSONAR GUI to configure tests
- Use pScheduler's command-line interface (CLI) to schedule tests
- Visualize end-to-end performance using Grafana
- Understand FABRIC

Tutorial on Science DMZs and Networking for All

Intended Audience

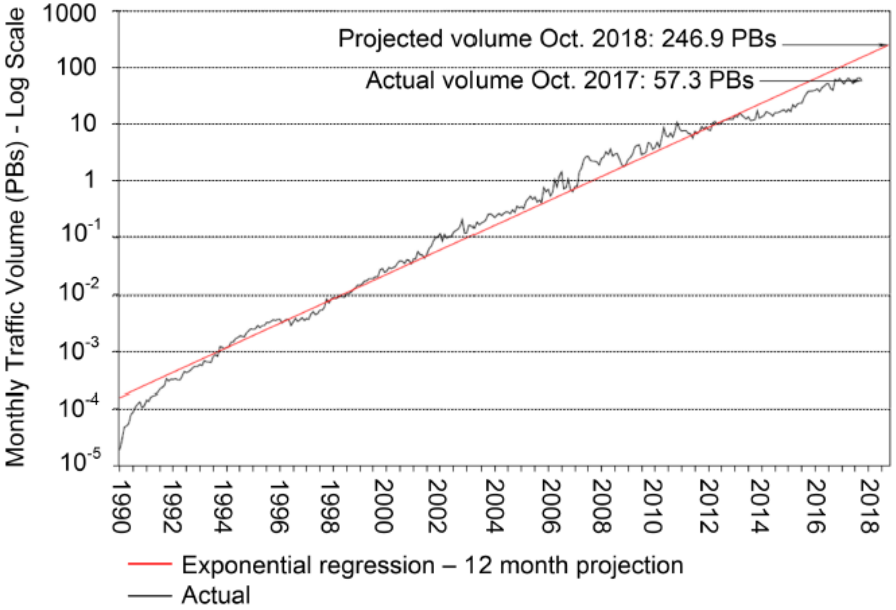
- IT professionals
- CI Engineers
- High-Performance computing specialists
- Research systems administrators
- Security professionals
- IT educators

Pre-requisites

- Basic knowledge of computer networks

Motivation for a High-Speed Science Architecture

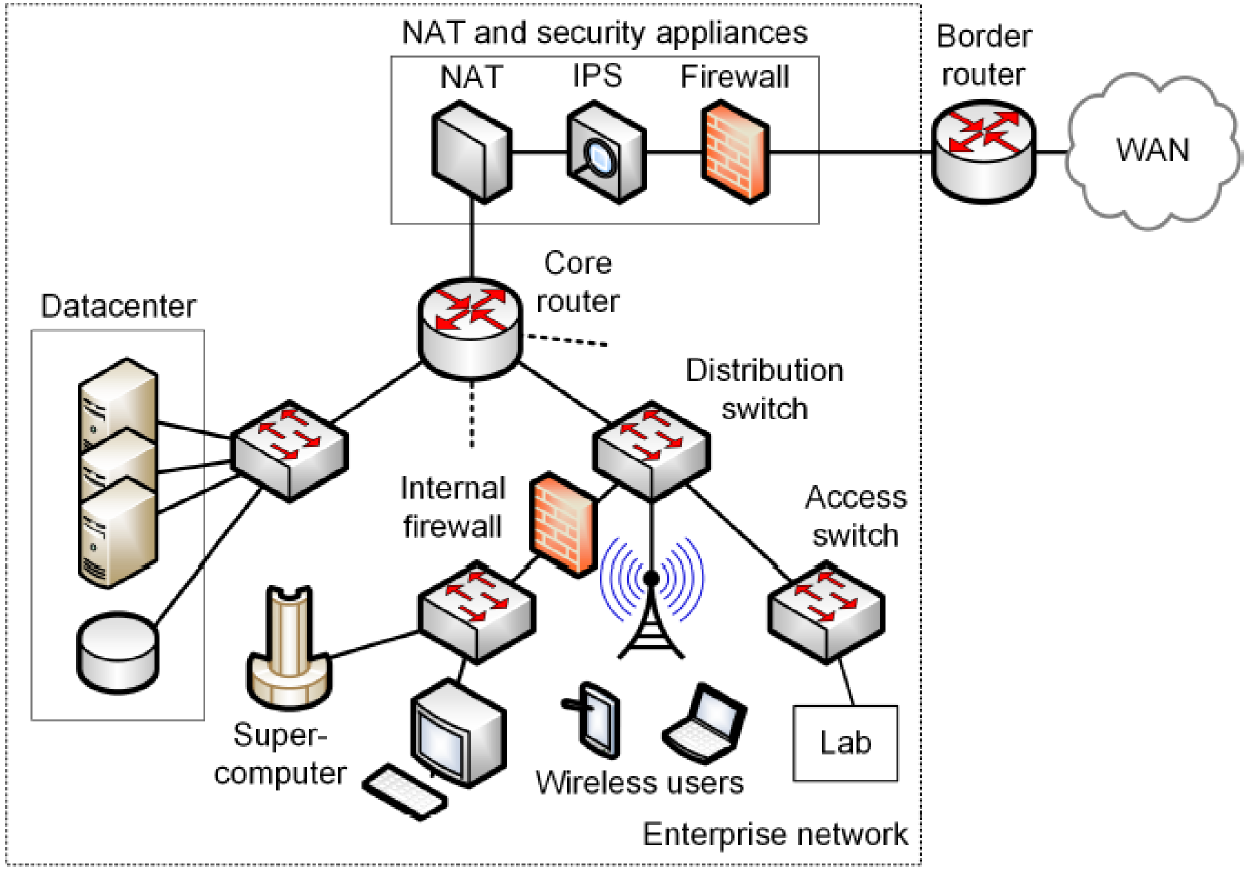
- Science and engineering applications are generating data at an unprecedented rate
- Instruments produce hundreds of terabytes in short time periods (“big science data”)
- Data must be typically transferred across high-bandwidth high-latency Wide Area Networks (WANs)



The Energy Science Network (ESnet) is the backbone connecting U.S. national laboratories and research centers

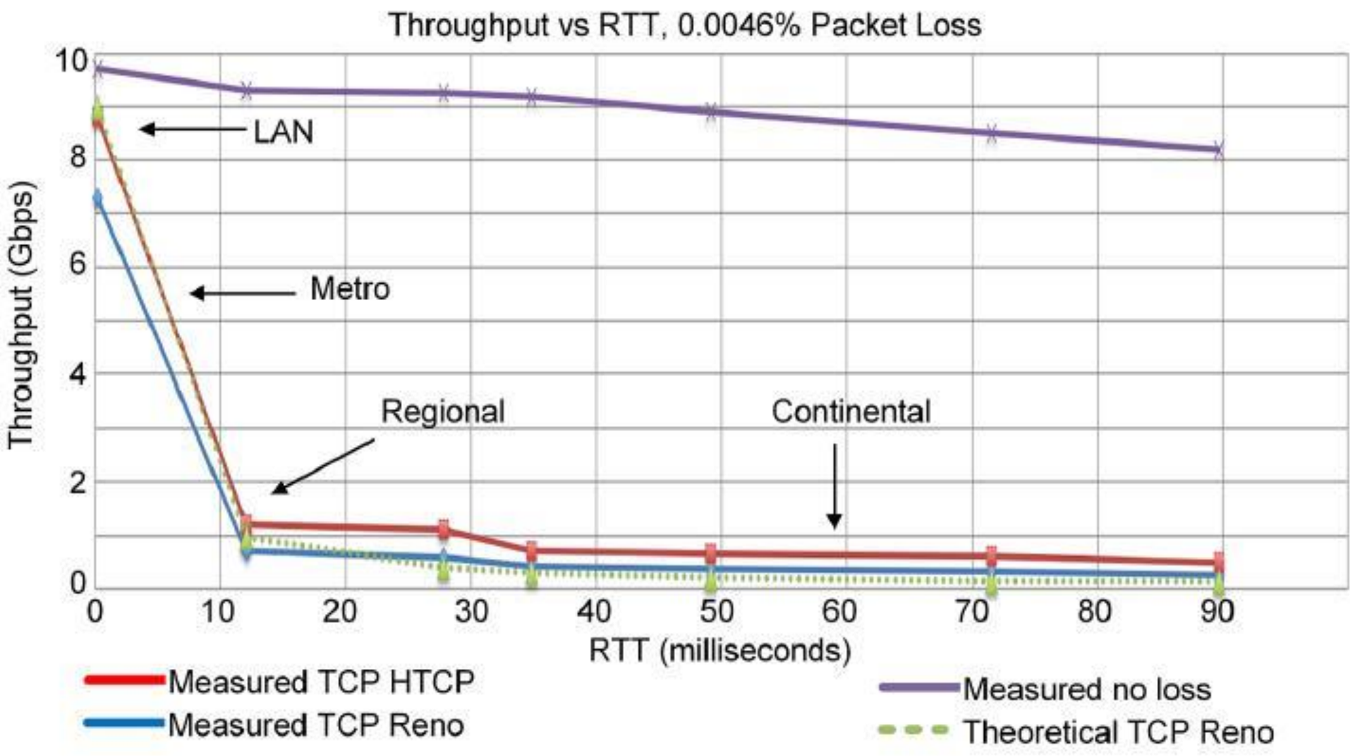
Enterprise Network Limitations

- Security appliances (IPS, firewalls, etc.) are CPU-intensive
- Inability of small-buffer routers/switches to absorb traffic bursts
- End devices incapable of sending/receiving data at high rates
- Lack of data transfer applications to exploit available bandwidth
- Many of the issues above relate to TCP



Enterprise Network Limitations

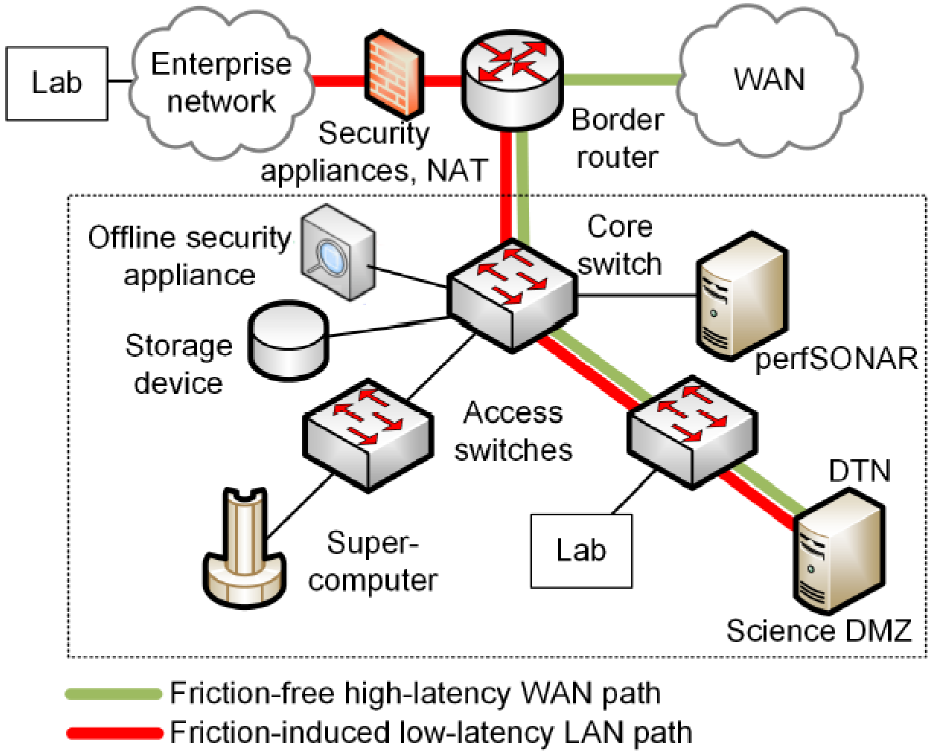
- Effect of packet loss and latency on TCP throughput
  - ➢ Data transfer between two devices
  - ➢ 10 Gbps network
  - ➢ Throughput as a function of the distance (milliseconds) between the two devices
  - ➢ Performance without (purple curve) and with packet loss (1/22,000 packet loss rate)



E. Dart, L. Rotman, B. Tierney, M. Hester, J. Zurawski, “The science dmz: a network design pattern for data-intensive science,” *International Conference on High Performance Computing, Networking, Storage and Analysis*, Nov. 2013.

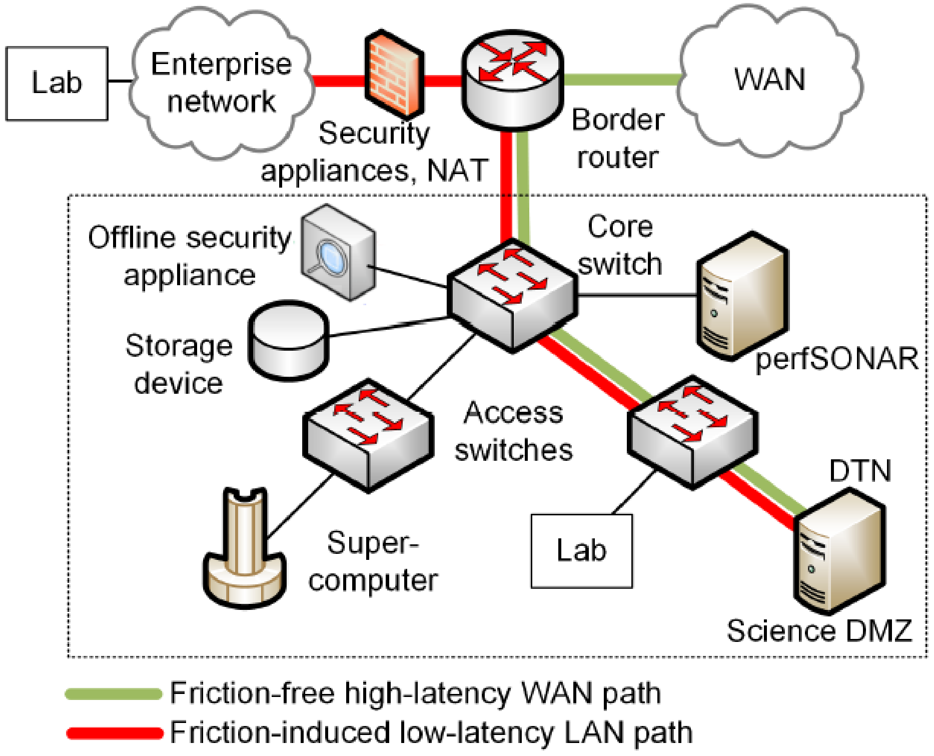
Science DMZ

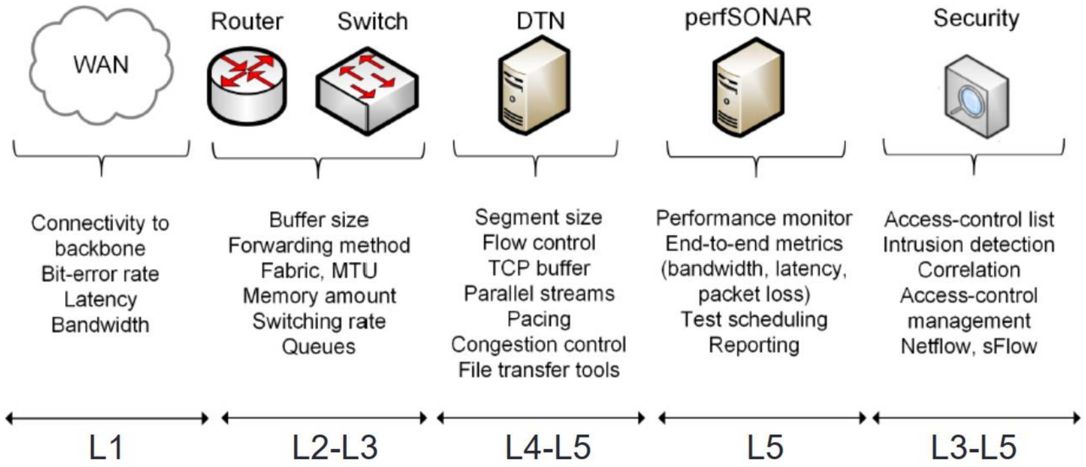
- The Science DMZ is a network designed for big science data
- Main elements
  - High throughput, friction free WAN paths
  - Data Transfer Nodes (DTNs)
  - End-to-end monitoring = perfSONAR
  - Security tailored for high speeds



Science DMZ

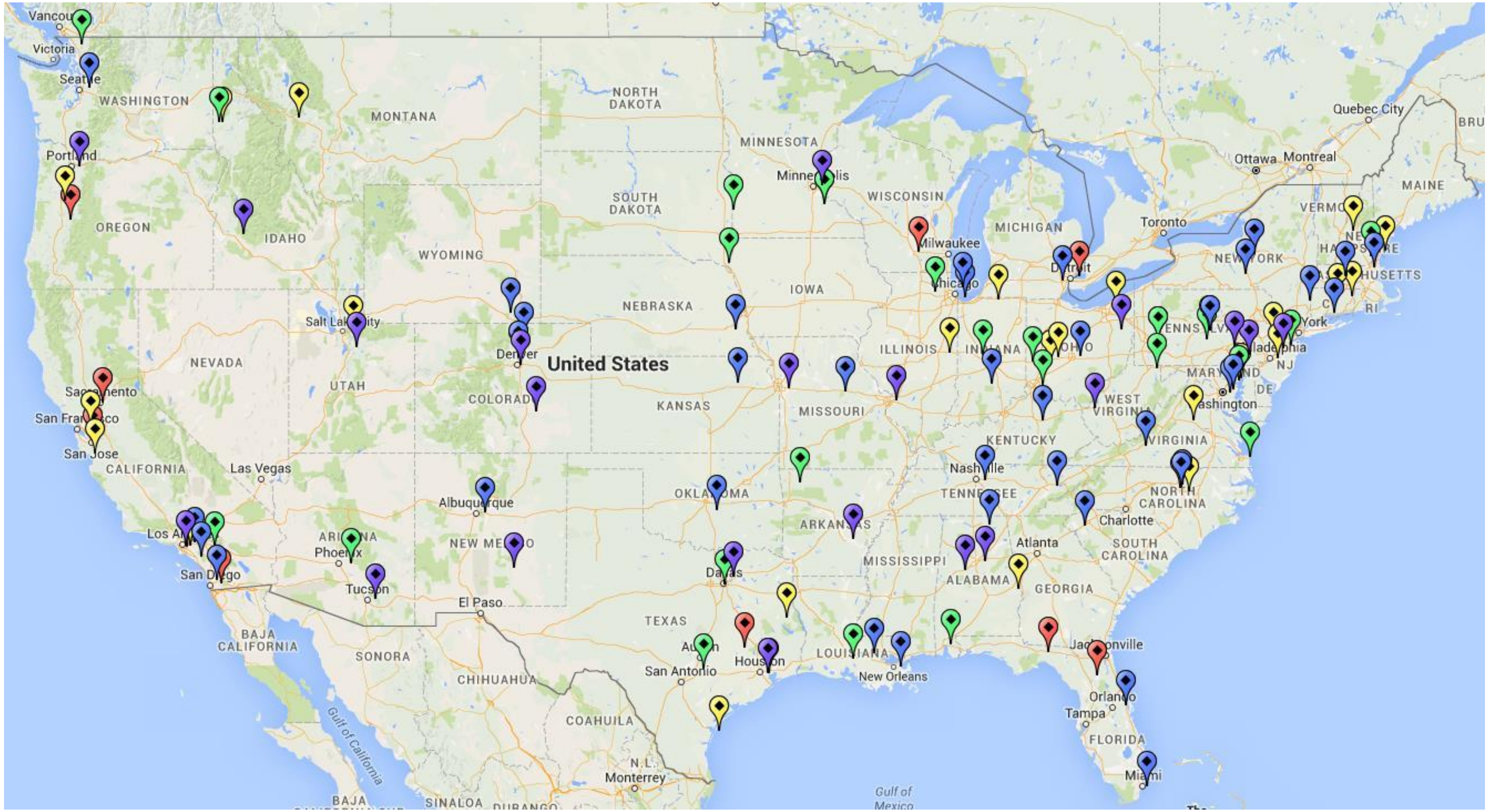
- The Science DMZ is a network designed for big science data
- Main elements
  - High throughput, friction free WAN paths
  - Data Transfer Nodes (DTNs)
  - End-to-end monitoring = perfSONAR
  - Security tailored for high speeds





Science DMZ

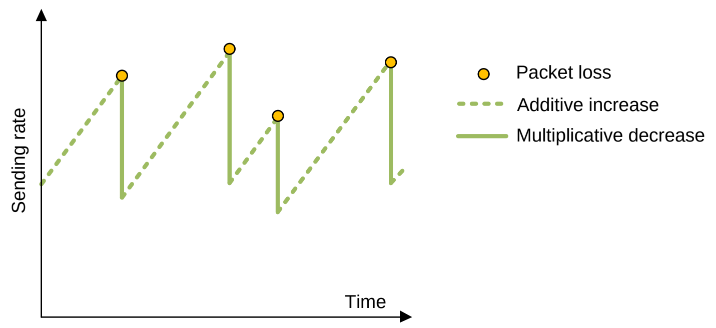
- Science DMZ deployments, U.S.



TCP Congestion Control, Parallel Streams, Maximum Segment Size (MSS), TCP buffers

TCP Traditional Congestion Control

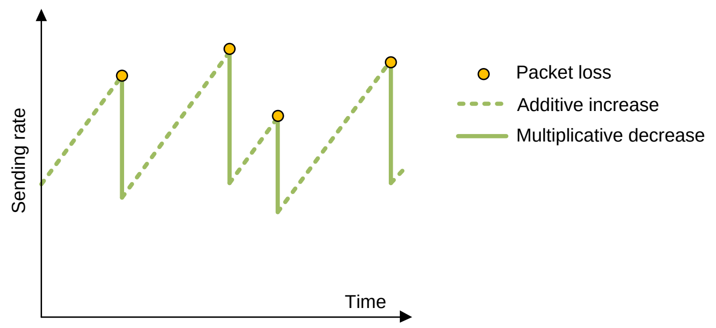
- The principles of window-based CC were described in the 1980s[1]
- Traditional CC algorithms follow the additive-increase multiplicative-decrease (AIMD) form of congestion control

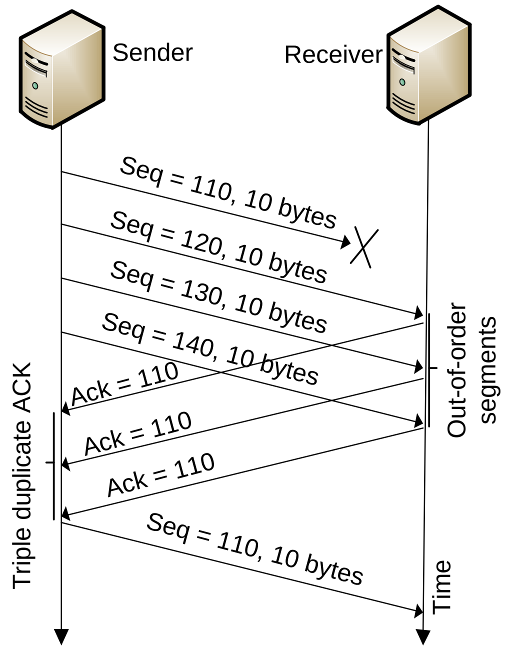


1. V. Jacobson, M. Karels, Congestion avoidance and control, ACM SIGCOMM Computer Communication Review 18 (4) (1988).

TCP Traditional Congestion Control

- The principles of window-based CC were described in the 1980s[1]
- Traditional CC algorithms follow the additive-increase multiplicative-decrease (AIMD) form of congestion control

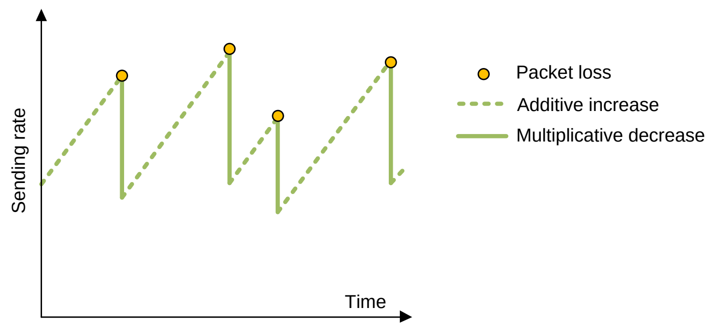


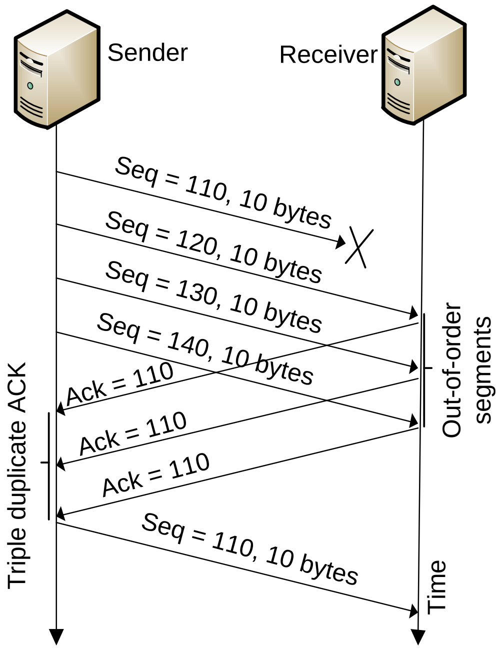


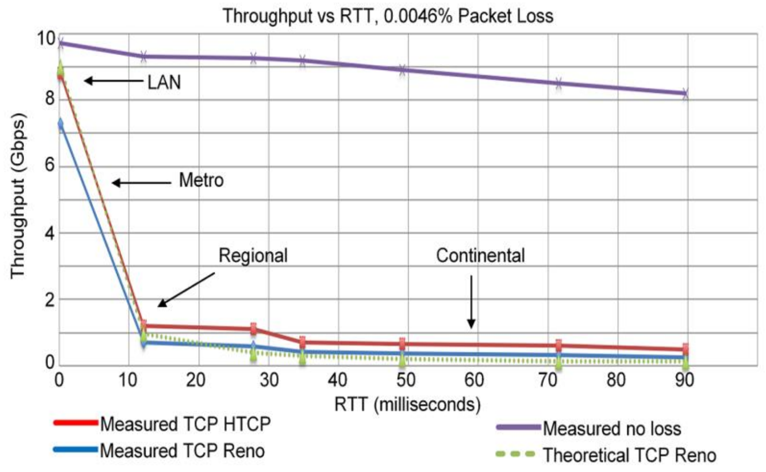
1. V. Jacobson, M. Karels, Congestion avoidance and control, ACM SIGCOMM Computer Communication Review 18 (4) (1988).

TCP Traditional Congestion Control

- The principles of window-based CC were described in the 1980s[1]
- Traditional CC algorithms follow the additive-increase multiplicative-decrease (AIMD) form of congestion control



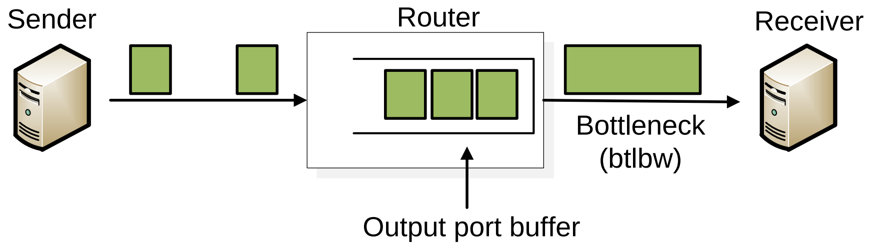




1. V. Jacobson, M. Karels, Congestion avoidance and control, ACM SIGCOMM Computer Communication Review 18 (4) (1988).

BBR: Model-based CC

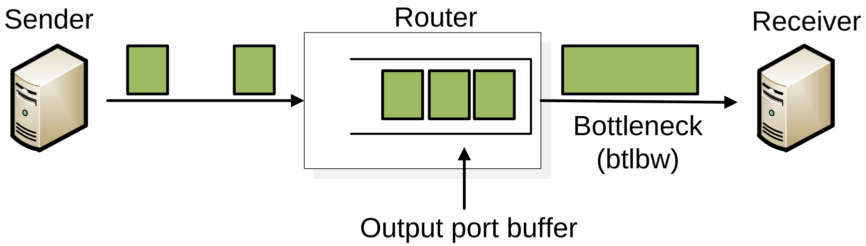
- TCP Bottleneck Bandwidth and RTT (BBR) is a rate-based congestion-control algorithm[1]
- BBR represented a disruption to the traditional CC algorithms:
  - is not governed by AIMD control law
  - does not use packet loss as a signal of congestion
- At any time, a TCP connection has one slowest link bottleneck bandwidth (btlbw)

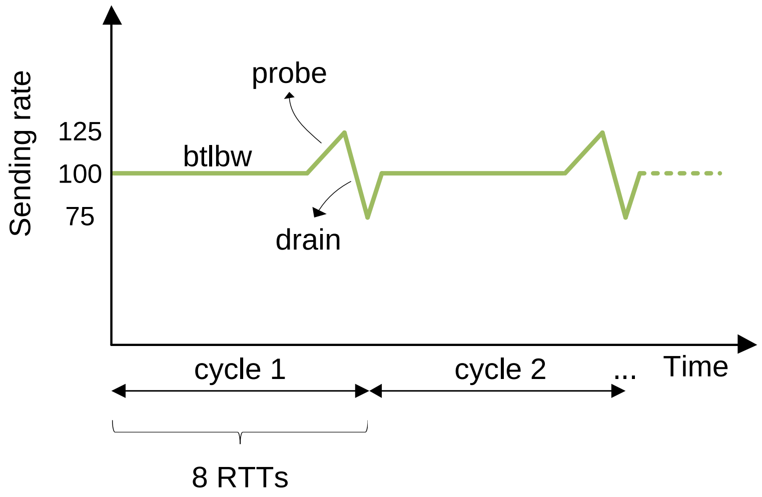


1. N. Cardwell et al. "BBR v2, A Model-based Congestion Control." IETF 104, March 2019.

BBR: Model-based CC

- TCP Bottleneck Bandwidth and RTT (BBR) is a rate-based congestion-control algorithm[1]
- BBR represented a disruption to the traditional CC algorithms:
  - is not governed by AIMD control law
  - does not use packet loss as a signal of congestion
- At any time, a TCP connection has one slowest link bottleneck bandwidth (btlbw)

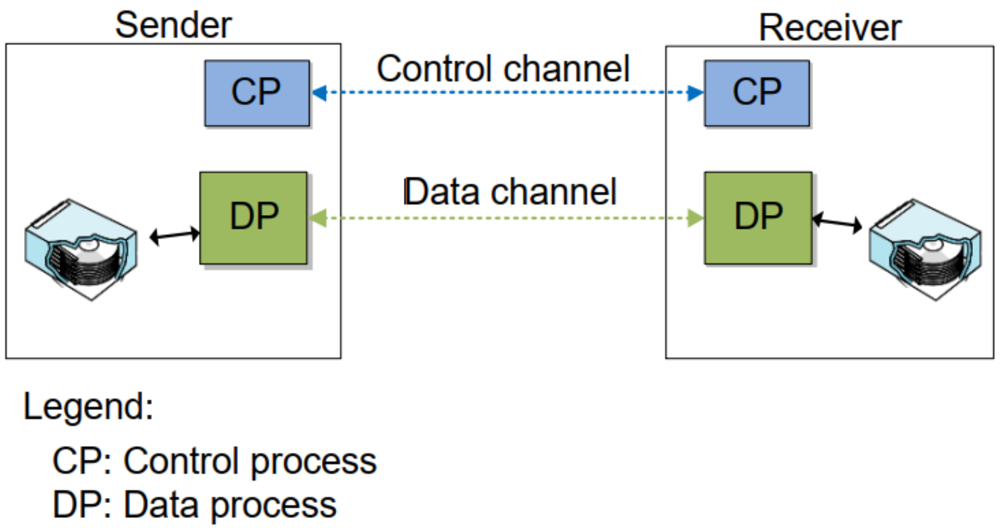




1. N. Cardwell et al. "BBR v2, A Model-based Congestion Control." IETF 104, March 2019.

Parallel Streams

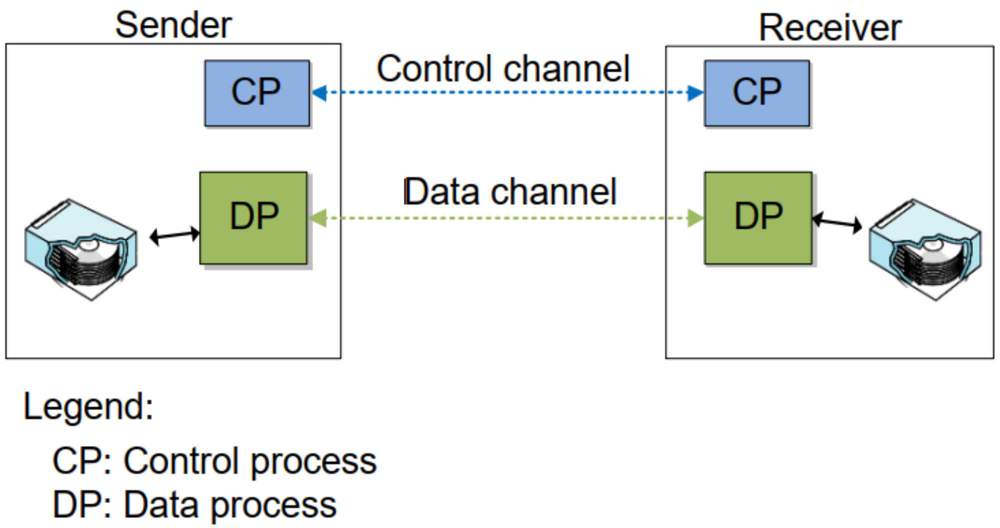
- Conventional file transfer protocols use a control channel and a (single) data channel (FTP model)



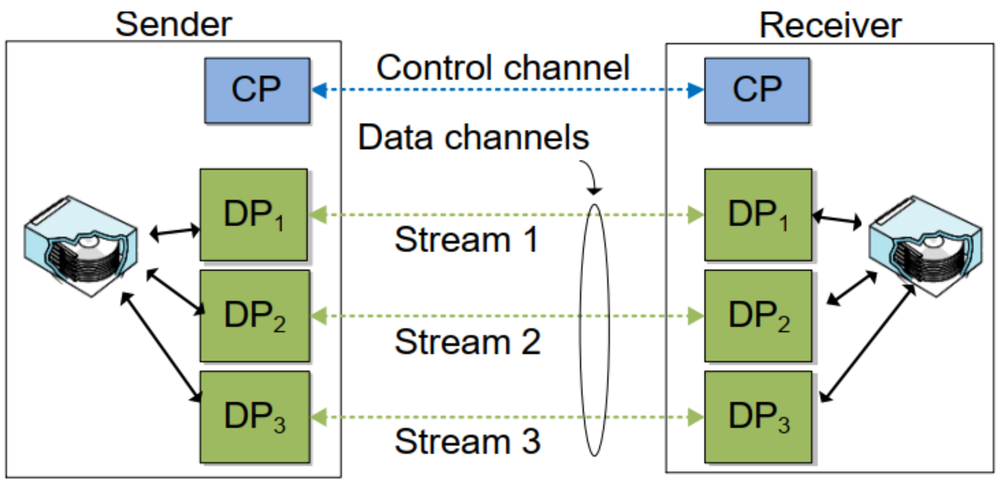
FTP model

Parallel Streams

- Conventional file transfer protocols use a control channel and a (single) data channel (FTP model)
- gridFTP is an extension of the FTP protocol
- A feature of gridFTP is the use of parallel streams



FTP model



gridFTP model

Advantages of Parallel Streams

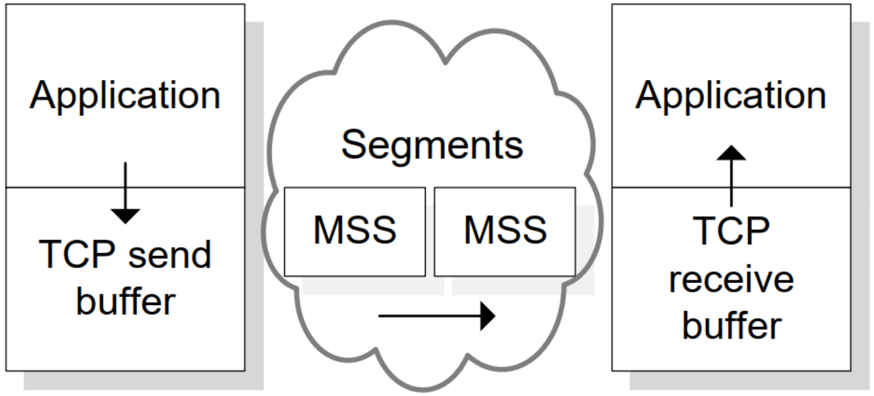
- Combat random packet loss not due congestion
  - Parallel streams increase the recovery speed after the multiplicative decrease

Advantages of Parallel Streams

- Combat random packet loss not due congestion
  - Parallel streams increase the recovery speed after the multiplicative decrease
- Mitigate TCP round-trip time (RTT) bias
  - A low-RTT flow gets a higher share of the bandwidth than that of a high-RTT flow
  - Increase bandwidth allocated to big science flows

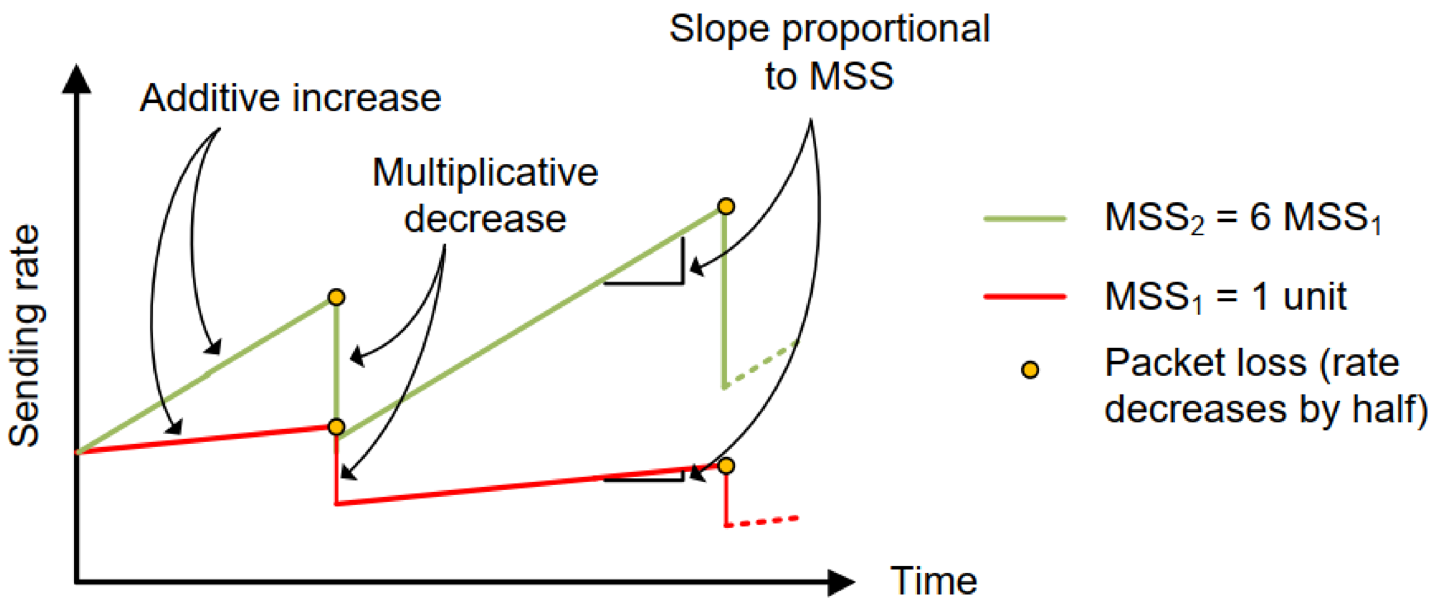
Maximum Segment Size (MSS)

- TCP receives data from application layer and places it in send buffer
- Data is typically broken into MSS units
- A typical MSS is 1,500 bytes, but it can be as large as 9,000 bytes



Advantages of Large MSS

- Less overhead
- The recovery after a packet loss is proportional to the MSS
  - ➢ During the additive increase phase, TCP increases the congestion window by approximately one MSS every RTT
  - ➢ By using a 9,000-byte MSS instead of a 1,500-byte MSS, the throughput increases six times faster

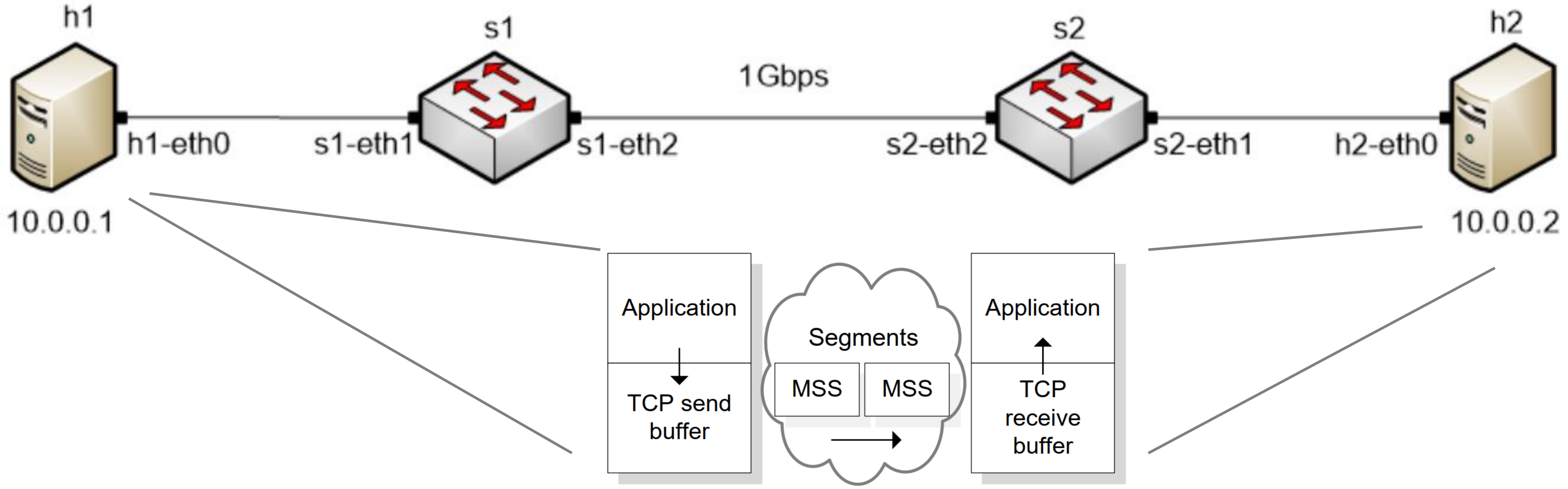


TCP Buffer Size

- In many WANs, the round-trip time (RTT) is dominated by the propagation delay
- To keep the sender busy while ACKs are received, the TCP buffer must be:

Traditional congestion controls: **TCP buffer size ≥ 2BDP**

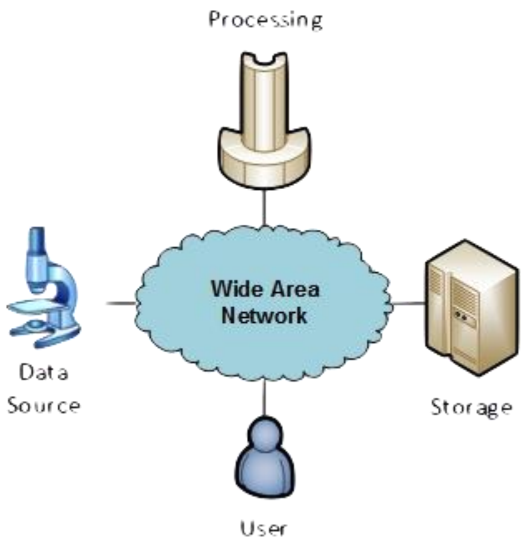
BBRv1 and BBRv2: **TCP buffer size must be considerable larger than 2BDP**

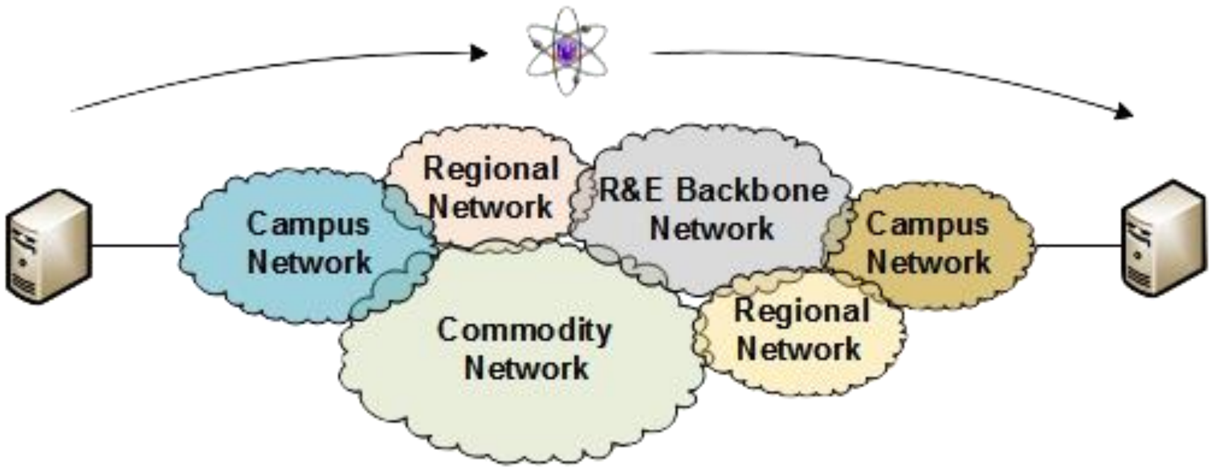


perfSONAR

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

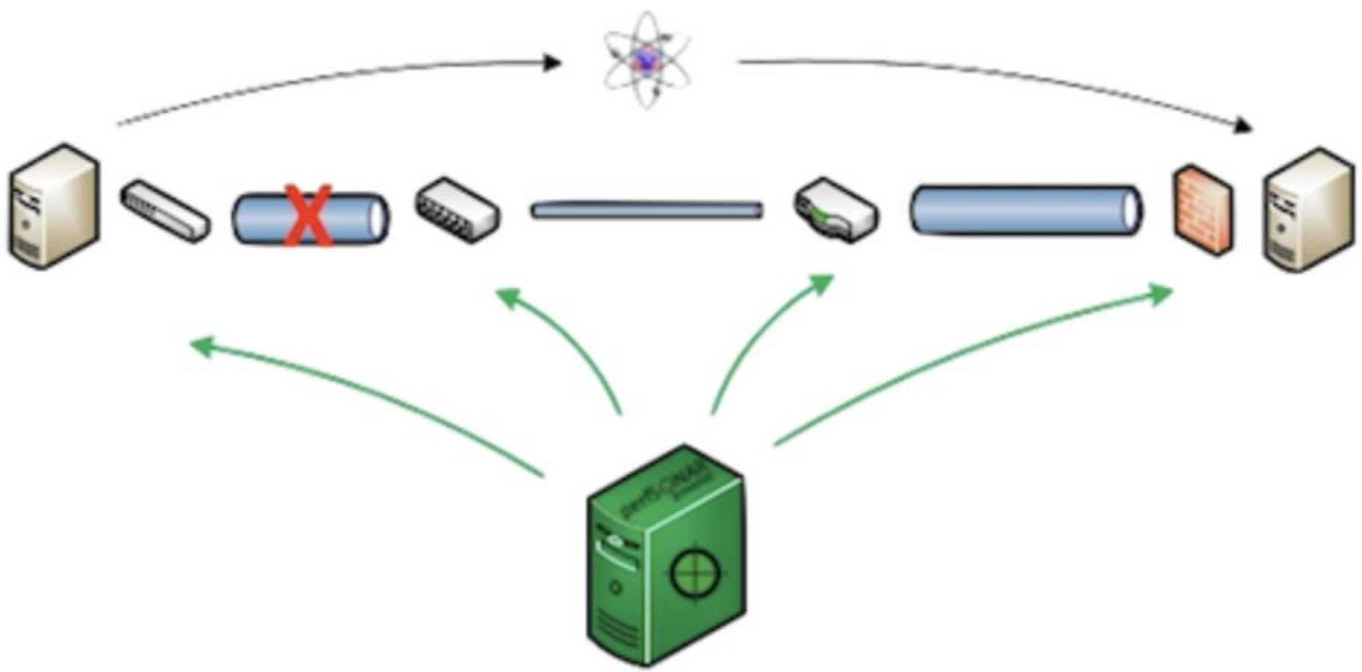




Source: ESNet, "*Monitoring end-to-end systems*", [Online]. Available: https://tinyurl.com/4tppsccv

Motivation

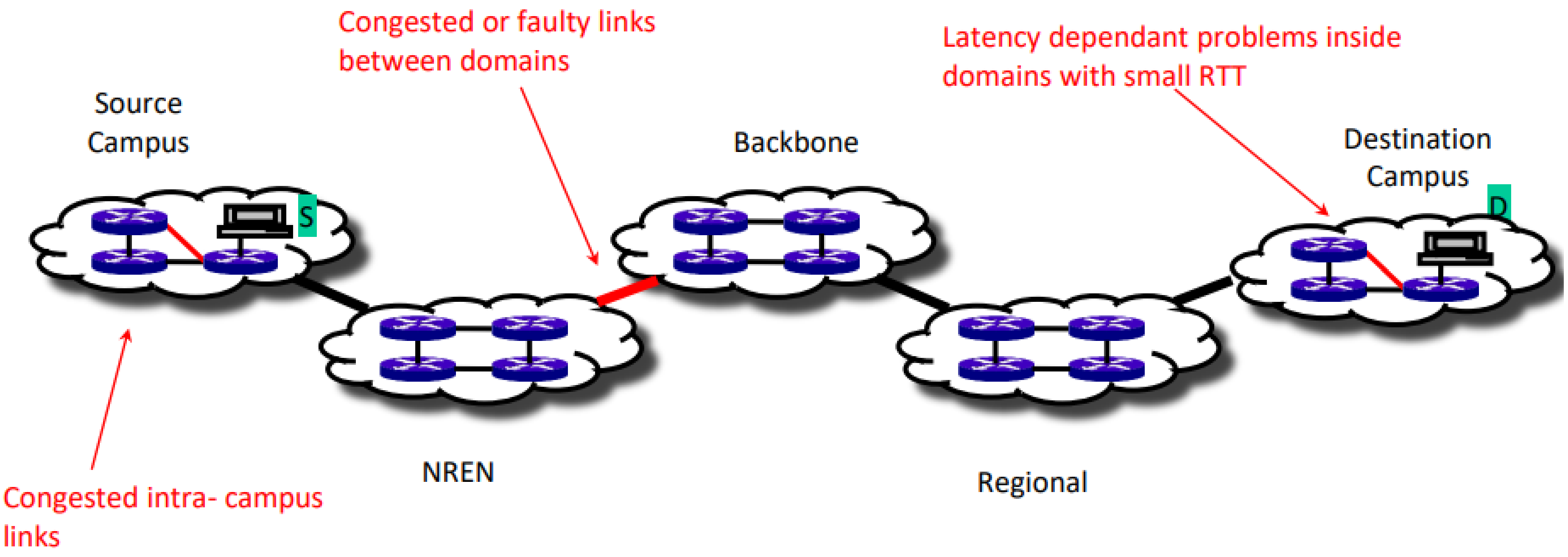
- Organizations must understand the behavior of their network by monitoring the performance metrics to ensure that the underlying system is functional



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

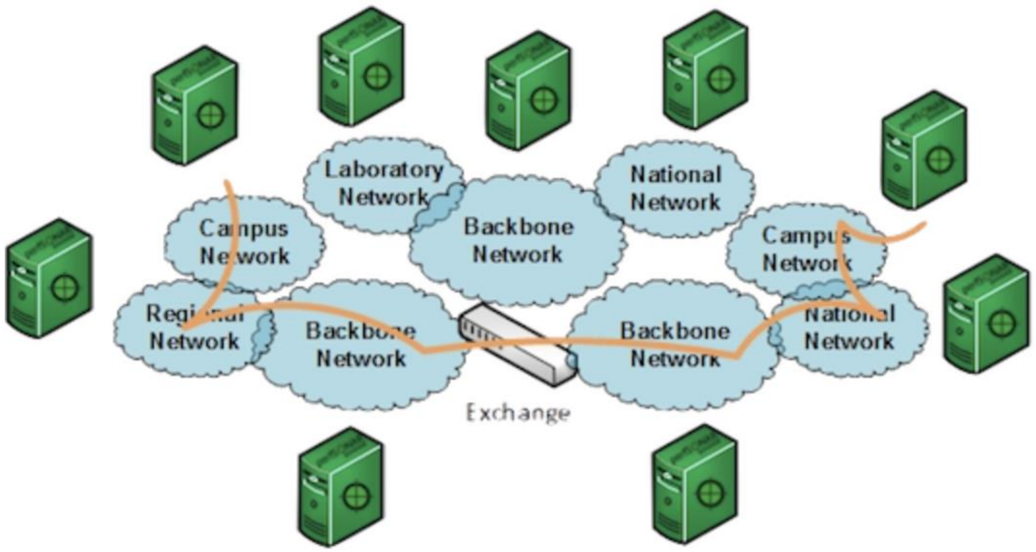
- Soft failures affect basic connectivity functions (e.g., long delays, packet losses)
- Some soft failures may only affect high bandwidth long RTT flows
- TCP was intentionally designed to hide transmission errors from the user
- Soft failures are difficult to detect and fix
- They can be hidden for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

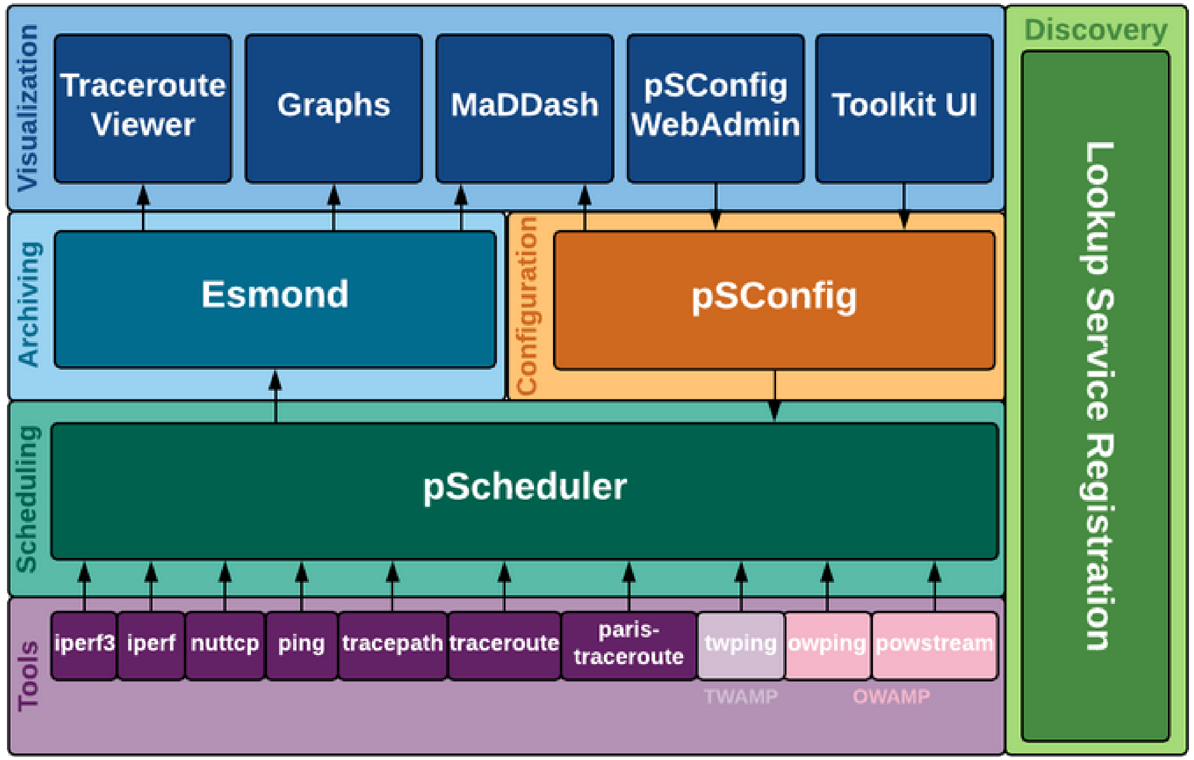
- perfSONAR is a network measurement tool designed to provide federated coverage of paths and help to achieve end-to-end usage expectations
- The tool facilitates diagnosing, visualizing, and troubleshooting network performance issues
- perfSONAR can collect metrics such as throughput, latency, and packet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



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

perfSONAR

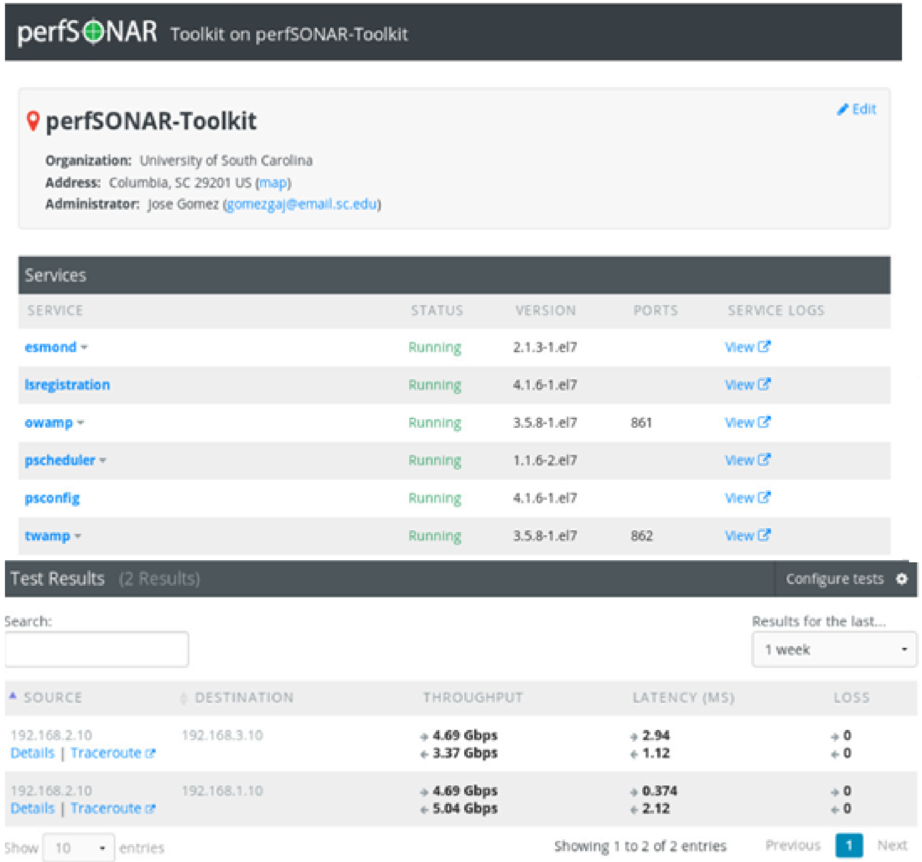
- perfSONAR provides a set of resources to orchestrate regular tests using open-source tools such as ping, traceroute, iperf3, and others



perfSONAR layers

perfSONAR

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

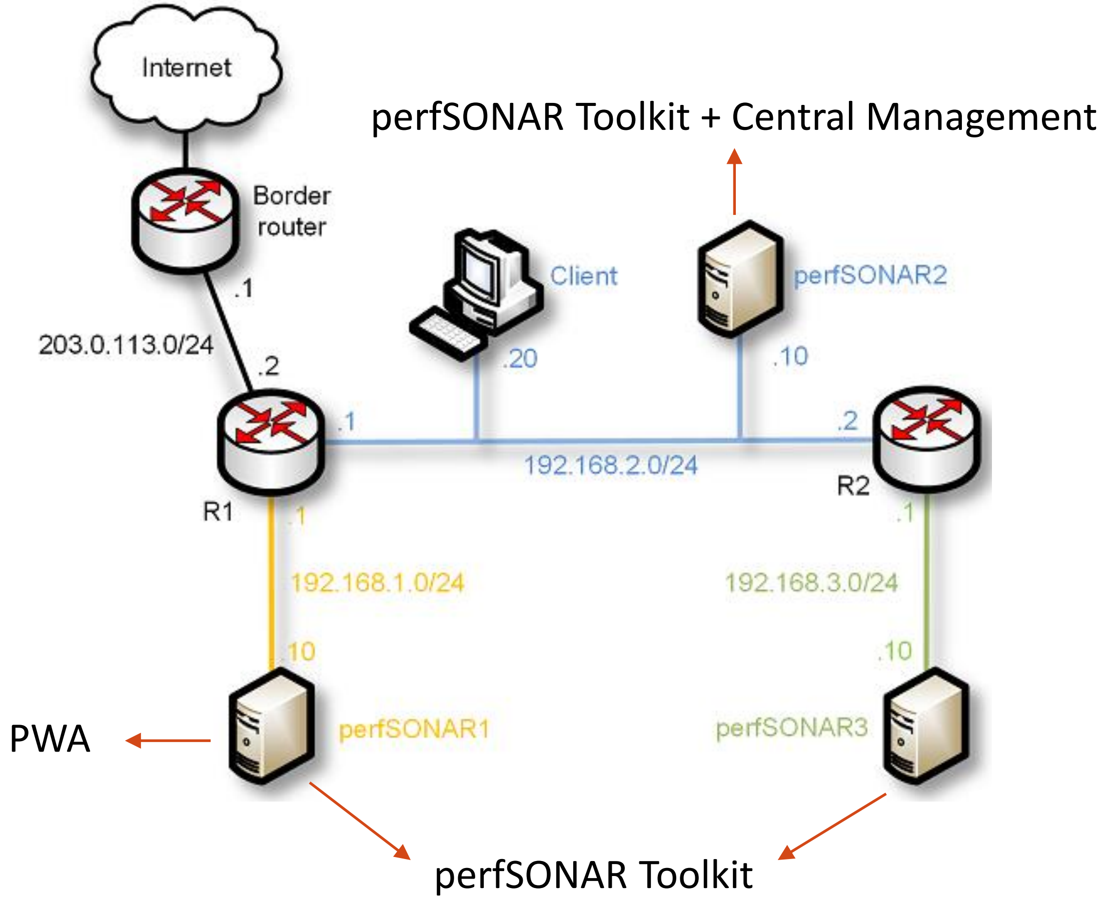


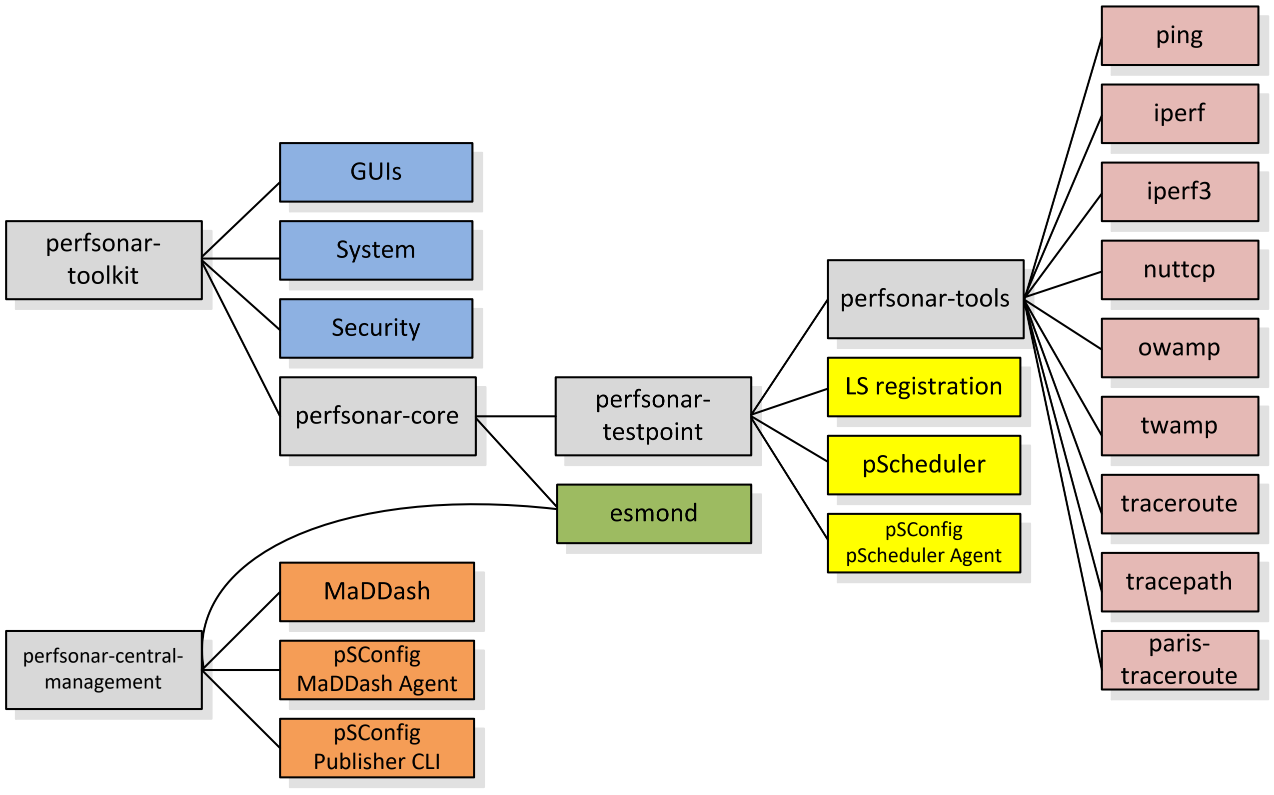
perfSONAR Toolkit GUI

perfSONAR test results

Getting Started with perfSONAR

- The CI-Lab at the University of South Carolina (USC) developed a set of hands-on labs that navigate through the components of perfSONAR





perfSONAR installation options

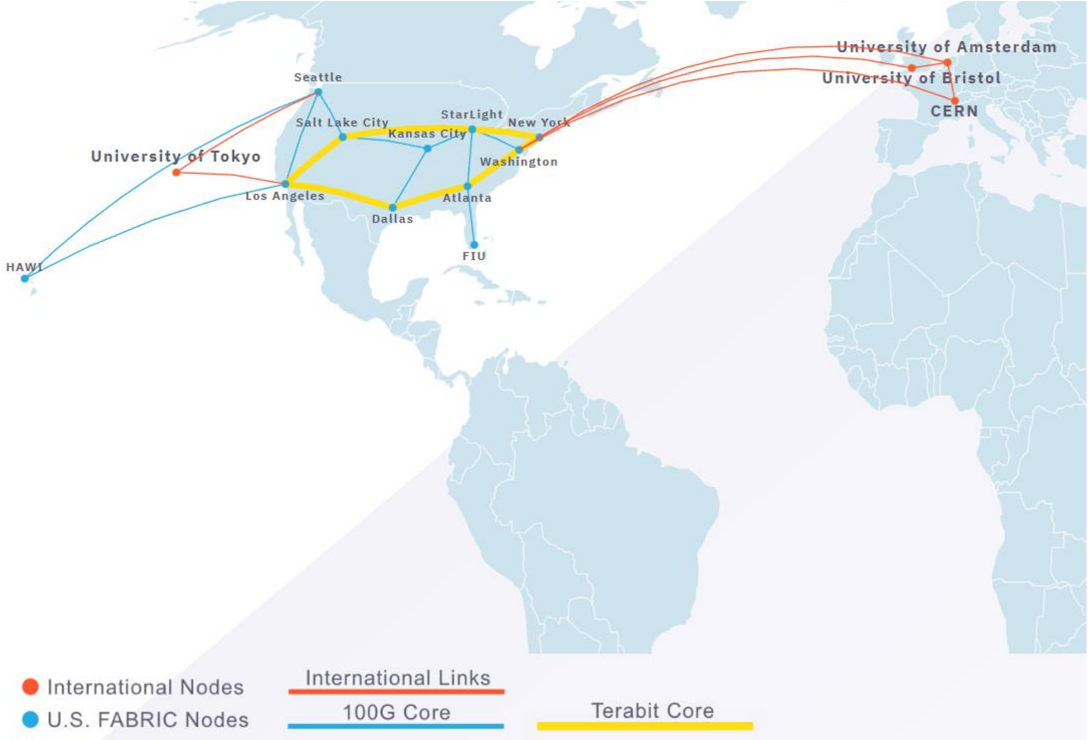
Useful Resources

- perfSONAR official website
  - URL: https://www.perfsonar.net/gtk_whatis.html
- perfSONAR documentation
  - URL: https://docs.perfsonar.net/
- ESNet website
  - URL: https://www.es.net/network-r-and-d/perfsonar/
- The CI-Lab website
  - URL: http://ce.sc.edu/cyberinfra/cybertraining.html

FABRIC

FABRIC Testbed

- FABRIC is an NSF-funded international infrastructure for at-scale experimentation and research
- Areas include networking, cyber, distributed computing, storage, 5G, ML, etc.
- Equipment is located at commercial collocation spaces, U.S. national labs, and campuses – 29 FABRIC sites

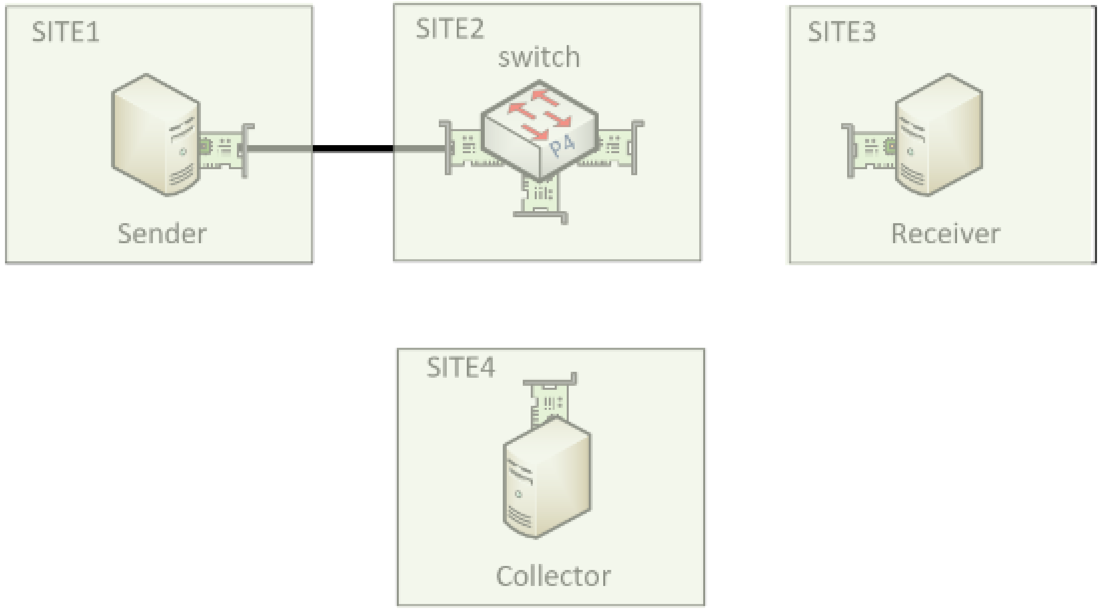


Cybertraining on FABRIC

- FABRIC is a real network with physical propagation delays and high-speed links
- With its integrated JupyterHub, it can be ideal for cybertraining:
  - ➢ P4 programmable switches/NICs
  - ➢ High-speed networks (SDMZ)
  - ➢ PerfSONAR
  - ➢ Measurement and telemetry
  - ➢ Cybersecurity (Zeek, Suricata, etc.)
  - ➢ Etc.

Step 3.7: Connecting site1 and site2

Create a site-to-site network between site1 and site2 connecting Sender and the P4 switch




```
net1 = slice.add_l2network(name='net1', interfaces=[sender_iface, switch_iface1])
```

Cybertraining on FABRIC

- FABRIC is a real network with physical propagation delays and high-speed links
- With its integrated JupyterHub, it can be ideal for cybertraining:
  - P4 programmable switches/NICs
  - High-speed networks (SDMZ)
  - PerfSONAR
  - Measurement and telemetry
  - Cybersecurity (Zeek, Suricata, etc.)
  - Etc.

Step 8.4: Starting iPerf3 on server2

```
[107]: server2.execute_thread('iperf3 -s')

[107]: <Future at 0x7fee04ab7b50 state=running>
```

Step 8.5: Starting iPerf3 client on server1

```
[114]: server1.execute('iperf3 -c 192.168.2.10 -P 2')

Connecting to host 192.168.2.10, port 5201
[  5] local 192.168.1.10 port 57904 connected to 192.168.2.10 port 5201
[  7] local 192.168.1.10 port 57908 connected to 192.168.2.10 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  64.5 MBytes   541 Mbits/sec  427   1.25 MBytes
[  7]   0.00-1.00   sec  72.4 MBytes   607 Mbits/sec  1050   1.47 MBytes
[SUM]   0.00-1.00   sec   137 MBytes  1.15 Gbits/sec  1477
- - - - - - - - - - - - - - - - - - - - - - - - -
[  5]   1.00-2.00   sec  60.0 MBytes   503 Mbits/sec   31    952 KBytes
[  7]   1.00-2.00   sec  70.0 MBytes   587 Mbits/sec   47   1.10 MBytes
[SUM]   1.00-2.00   sec   130 MBytes  1.09 Gbits/sec   78
- - - - - - - - - - - - - - - - - - - - - - - - -
[  5]   2.00-3.00   sec  56.2 MBytes   472 Mbits/sec    0   1024 KBytes
[  7]   2.00-3.00   sec  66.2 MBytes   556 Mbits/sec    0   1.18 MBytes
[SUM]   2.00-3.00   sec   122 MBytes  1.03 Gbits/sec    0
- - - - - - - - - - - - - - - - - - - - - - - - -
```

Organization of the labs

Each lab starts with a section *Overview*

- Objectives
- Lab topology
- Roadmap: organization of the lab

Part 1

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

Part 2… n

- Step-by-step directions

Labs on P4 Programmable Data Planes over FABRIC

- The following labs have been developed:
  - Lab 1 – Preparing the Environment
  - Lab 2 – P4 Program Building Blocks
  - Lab 3 – Parser Implementation
  - Lab 4 – Introduction to Match-action Tables
  - Lab 5 – Populating Match-action Tables from the Control Plane
  - Lab 6 – Checksum Calculation and Packet Deparsing
  - Lab 7 – Fine-grained Queue Measurement

Upcoming Lab Libraries over FABRIC

- Advanced P4 Programmable Data Planes: Applications, Stateful Elements, and Custom Packet Processing
- Writing Cybersecurity Applications on P4 Programmable Data Planes
- PerfSONAR 5 (perfSONAR's components, Measurements with Grafana Dashboard)
- High Speed Networks (TCP Congestion Control, Buffer Size, BDP, TCP Fairness, etc.)
- Software-defined Networking and Open vSwitch (OVS)
- Introduction to SmartNICs