Hands-on Tutorial on Science DMZ
Session 1: iPerf3, TCP Buffer Size, Packet Loss

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

Jason Zurawski
EPOC and ESnet

2022 NSF Campus Cyberinfrastructure PI Workshop

Minneapolis, Minnesota
September 19th, 2022

Hands-on Tutorials on Science DMZ

- Webpage with PowerPoint presentations:

  http://ce.sc.edu/cyberinfra/workshop_2022_cc_pi.html

- Session 1 (1:00-1:50pm): to access labs for Session 1 (TCP, buffers,…), register here:

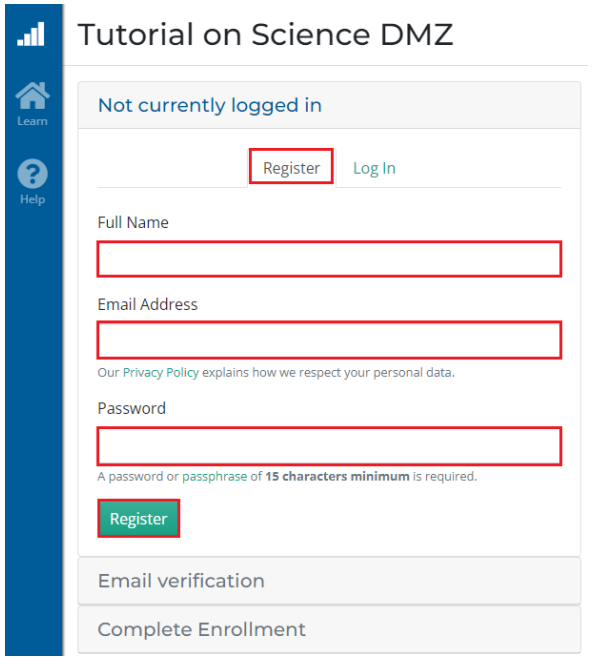
  https://portal.netdevgroup.com/learn/ca3pgf/enroll/

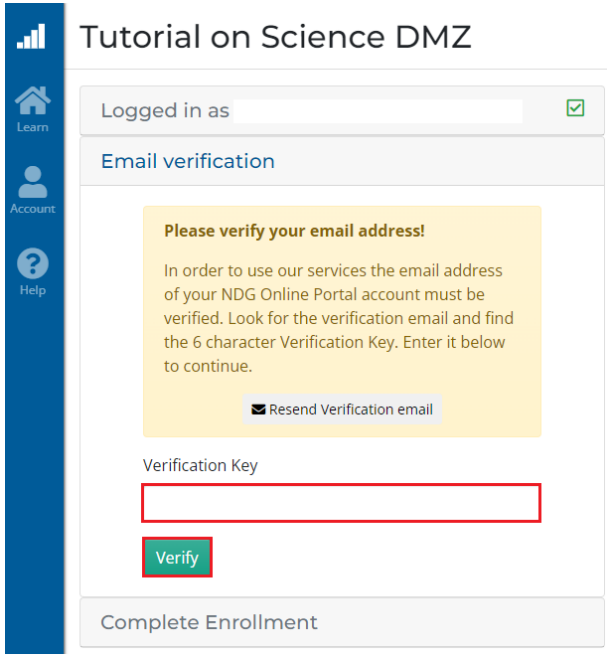
- Session 2 (2:15-3:05): to access labs for Session 2 (perfSONAR), register here:

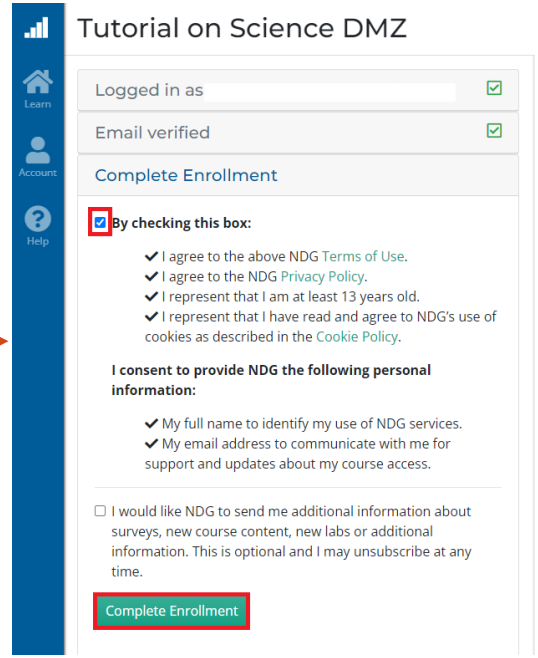
  https://portal.netdevgroup.com/learn/j39z9e/enroll/

Registering to the Netlab Portal

1. Click on the enrollment link: https://portal.netdevgroup.com/learn/ca3pgf/enroll/
2. Register and check your email for the verification key
3. Finalize the registration by claiming your free access

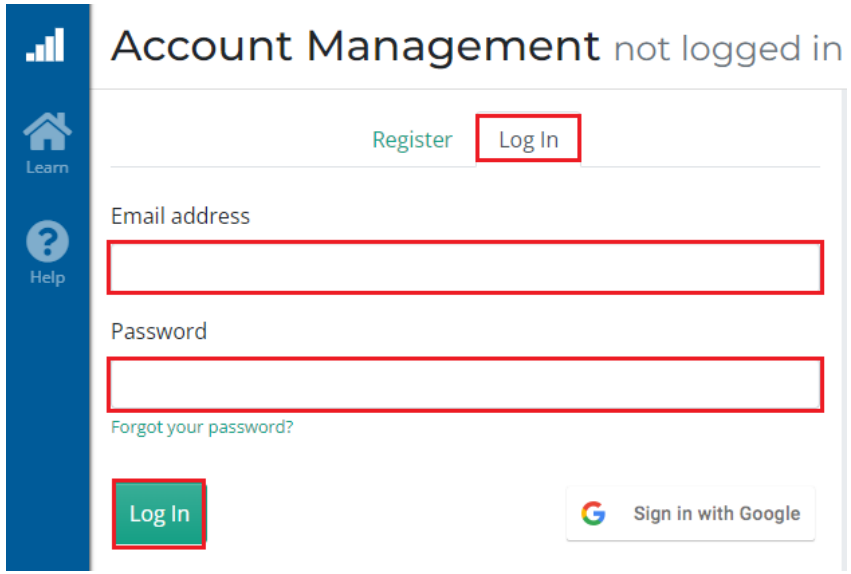


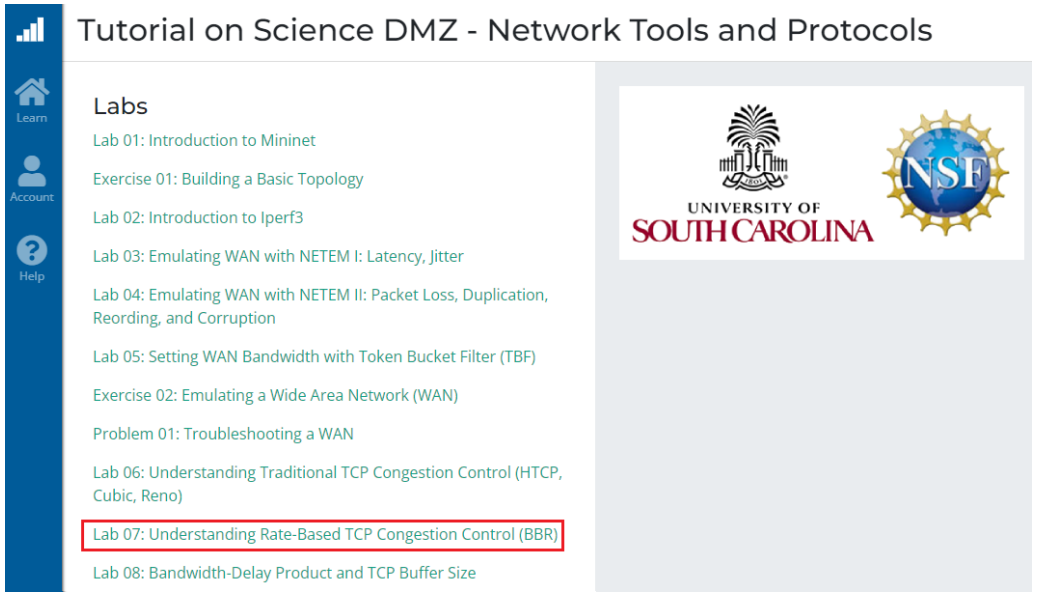




Accessing the Virtual Labs

1. If already registered, login to the portal: https://portal.netdevgroup.com/account/login
2. Click on the course "*Tutorial on Science DMZ- Network Tools and Protocols*"
3. Select the lab you want to run (e.g., Lab 7)





NTP Lab Series

- Lab experiments

Lab 1: Introduction to Mininet
Lab 2: Introduction to iPerf
Lab 3: WANs with latency, Jitter
Lab 4: WANs with Packet Loss, Duplication, Corruption
Lab 5: Setting WAN Bandwidth with Token Bucket Filter (TBF)
Lab 6: Traditional TCP Congestion Control (HTCP, Cubic, Reno)
Lab 7: Rate-based TCP Congestion Control (BBR)
Lab 8: Bandwidth-delay Product and TCP Buffer Size
Lab 9: Enhancing TCP Throughput with Parallel Streams
Lab 10: Measuring TCP Fairness
Lab 11: Router's Buffer Size
Lab 12: TCP Rate Control with Pacing
Lab 13: Impact of Maximum Segment Size on Throughput
Lab 14: Router's Bufferbloat
Lab 15: Hardware Offloading on TCP Performance
Lab 16: Random Early Detection
Lab 17: Stochastic Fair Queueing
Lab 18: Controlled Delay (CoDel) Active Queue Management
Lab 19: Proportional Integral Controller-Enhanced (PIE)
Lab 20: Classifying TCP traffic using Hierarchical Token Bucket (HTB)

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

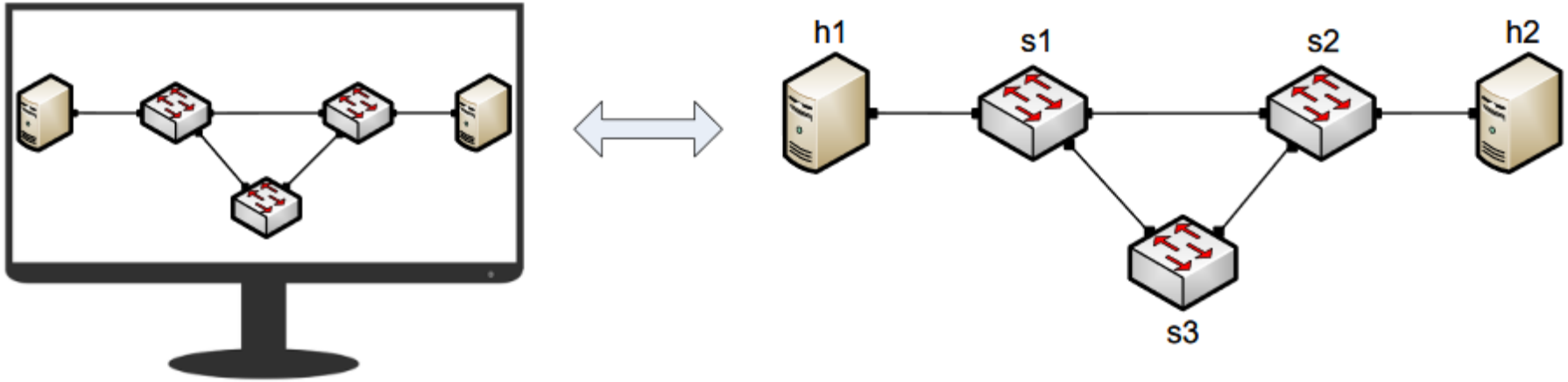
- Background information of the topic being covered (e.g., fundamentals of perfSONAR)
- Section 1 is optional (i.e., the reader can skip this section and move to lab directions)

Section 2… n

- Step-by-step directions

Mininet

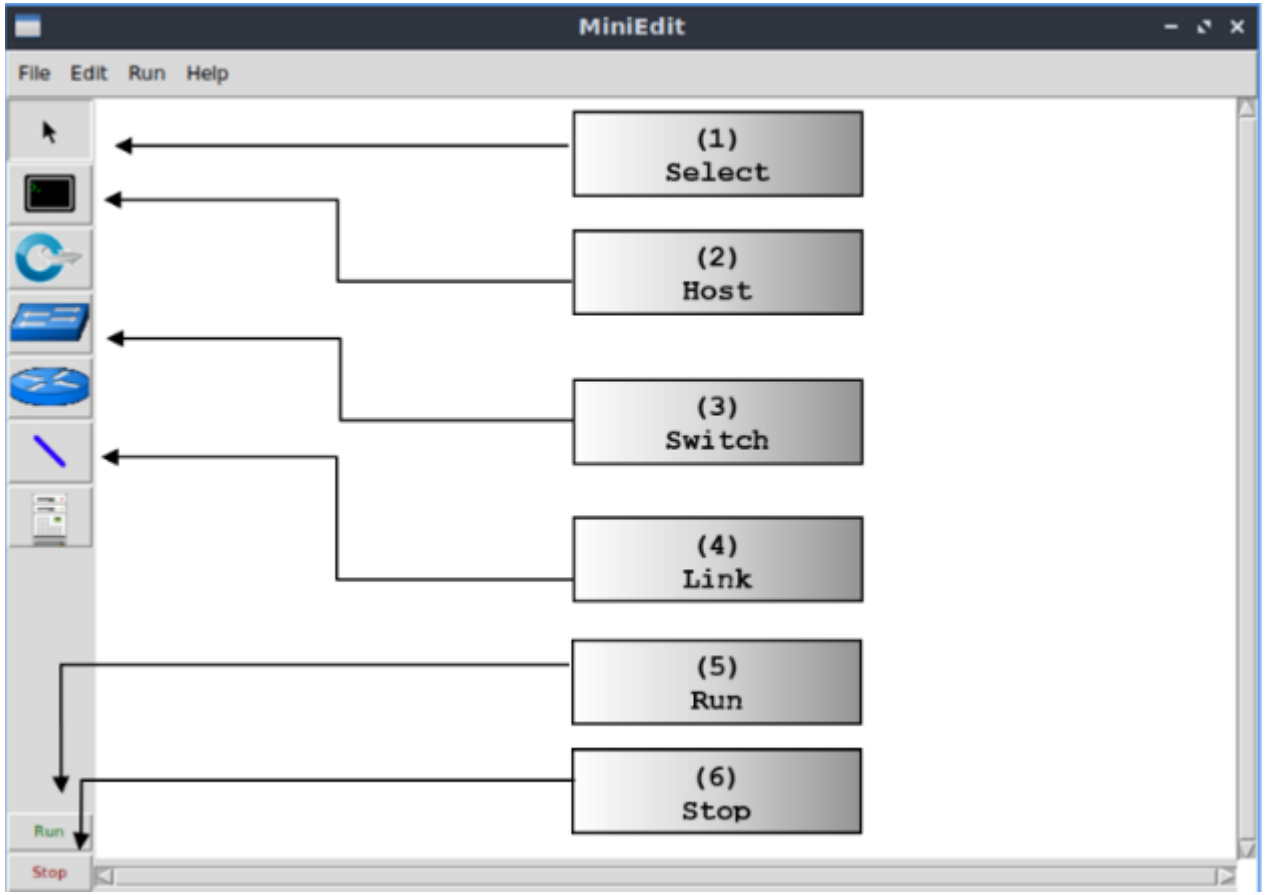
- Mininet provides network *emulation* opposed to simulation, allowing all network software at any layer to be simply run as is
- Mininet's logical nodes can be connected into networks
- Nodes are sometimes called containers, or more accurately, *network namespaces*
- Containers consume sufficiently few resources that networks of over a thousand nodes have been created, running on a single laptop



Mininet Emulated Network **Hardware Network**

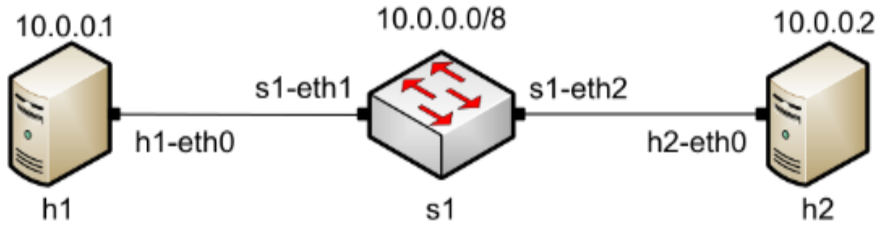
MiniEdit

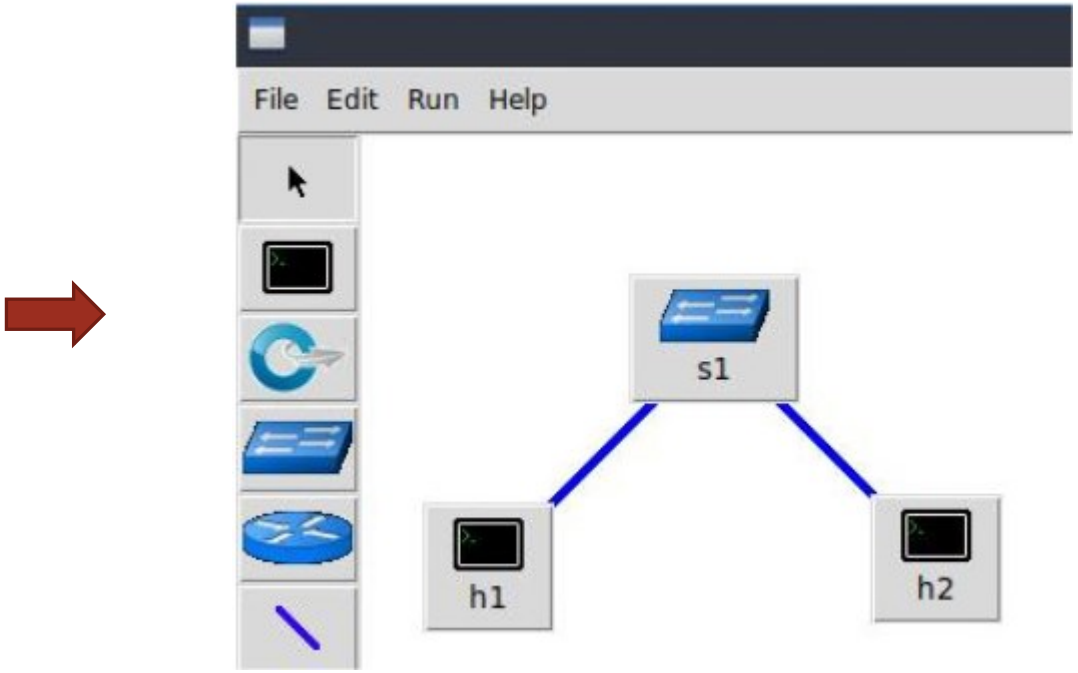
- MiniEdit is a simple GUI network editor for Mininet



MiniEdit

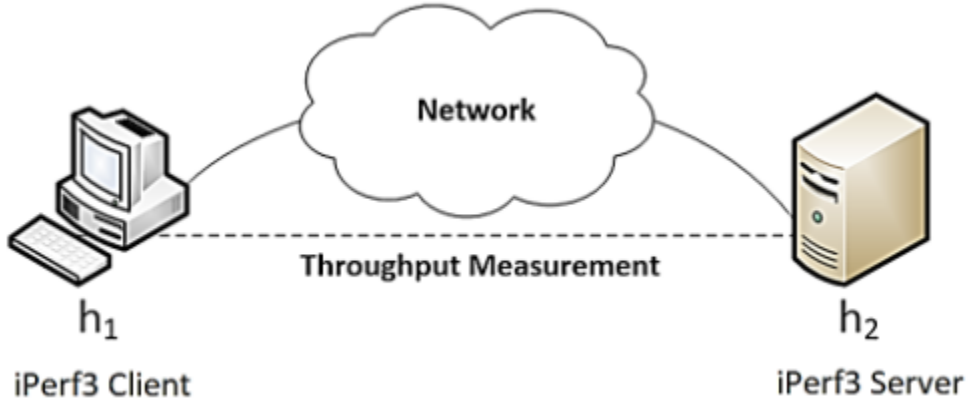
- To build Mininet's minimal topology, two hosts and one switch must be deployed





iPerf3

- iPerf3 is a real-time network throughput measurement tool
- It is an open source, cross-platform client-server application that can be used to measure the throughput between the two end devices
- Measuring throughput is particularly useful when experiencing network bandwidth issues such as delay, packet loss, etc.



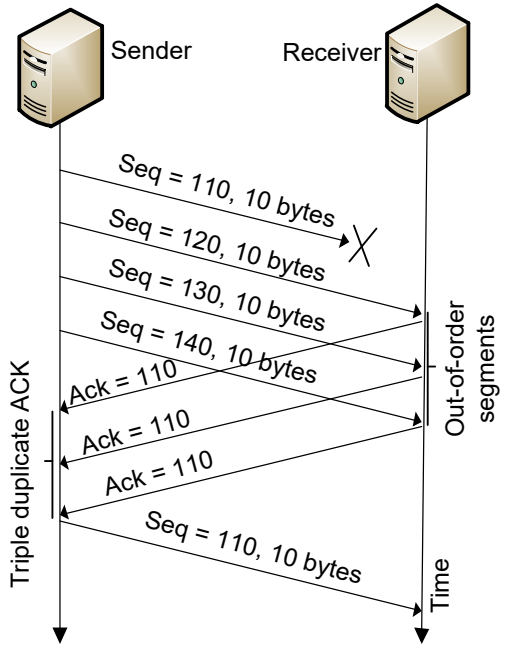
iPerf3

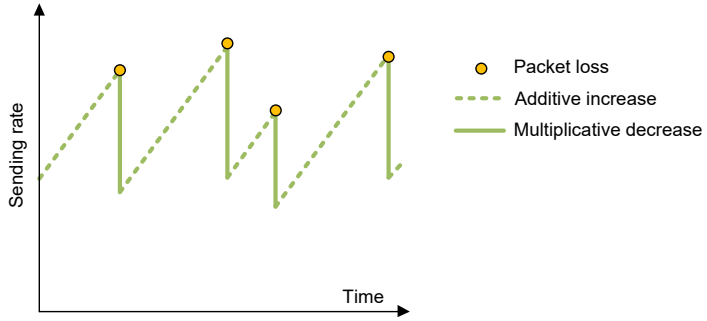
- iPerf3 can operate on TCP, UDP, and SCTP, unidirectional or bidirectional way
- In iPerf3, the user can set *client* and *server* configurations via options and parameters
- iPerf3 outputs a timestamped report of the amount of data transferred and the throughput measured

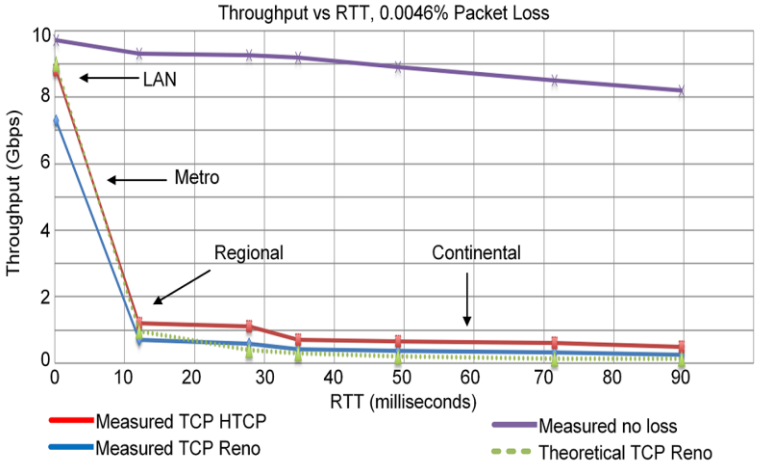
```
Connecting to host 10.0.0.2, port 5201
[ 13] local 10.0.0.1 port 59414 connected to 10.0.0.2 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[ 13]   0.00-1.00   sec  5.18 GBytes  44.5 Gbits/sec    0    843 KBytes
[ 13]   1.00-2.00   sec  5.21 GBytes  44.7 Gbits/sec    0   1.11 MBytes
[ 13]   2.00-3.00   sec  5.20 GBytes  44.7 Gbits/sec    0   1.18 MBytes
[ 13]   3.00-4.00   sec  5.21 GBytes  44.7 Gbits/sec    0   1.24 MBytes
[ 13]   4.00-5.00   sec  5.19 GBytes  44.6 Gbits/sec    0   1.24 MBytes
[ 13]   5.00-6.00   sec  5.22 GBytes  44.8 Gbits/sec    0   1.30 MBytes
[ 13]   6.00-7.00   sec  5.24 GBytes  45.0 Gbits/sec    0   1.44 MBytes
[ 13]   7.00-8.00   sec  5.22 GBytes  44.9 Gbits/sec    0   1.44 MBytes
[ 13]   8.00-9.00   sec  5.21 GBytes  44.8 Gbits/sec    0   1.45 MBytes
[ 13]   9.00-10.00  sec  5.22 GBytes  44.8 Gbits/sec    0   1.52 MBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[ 13]   0.00-10.00  sec  52.1 GBytes  44.8 Gbits/sec    0             sender
[ 13]   0.00-10.04  sec  52.1 GBytes  44.6 Gbits/sec                  receiver

iperf Done.
root@admin-pc:~#
```

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



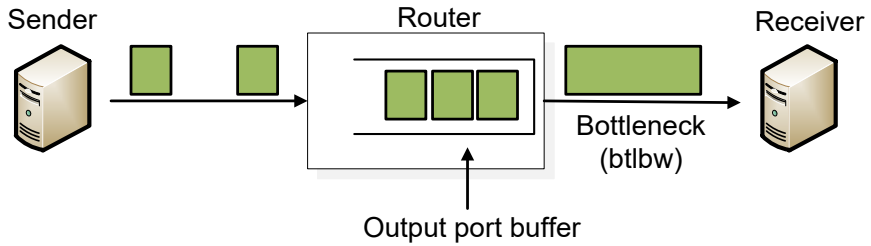


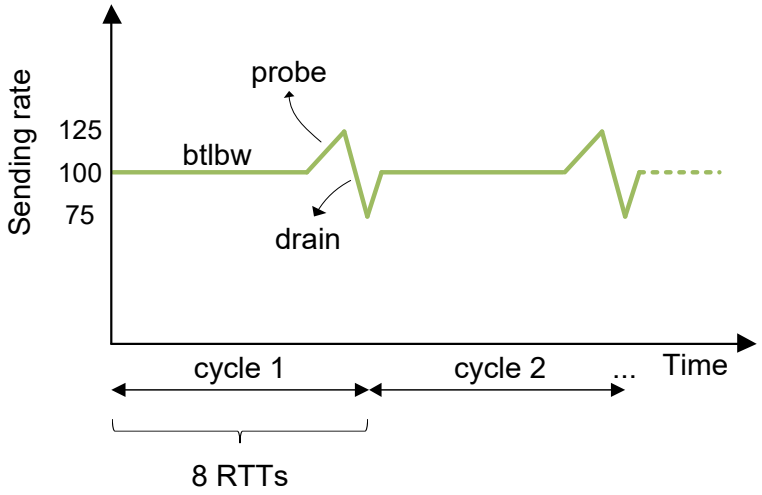


1. V. Jacobson, M. Karels, Congestion avoidance and control, ACM SIGCOMM Computer Communication Review 18 (4) (1988).

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 the use packet loss as a signal of congestion
- At any time, a TCP connection has one slowest link bottleneck bandwidth (btlbw)





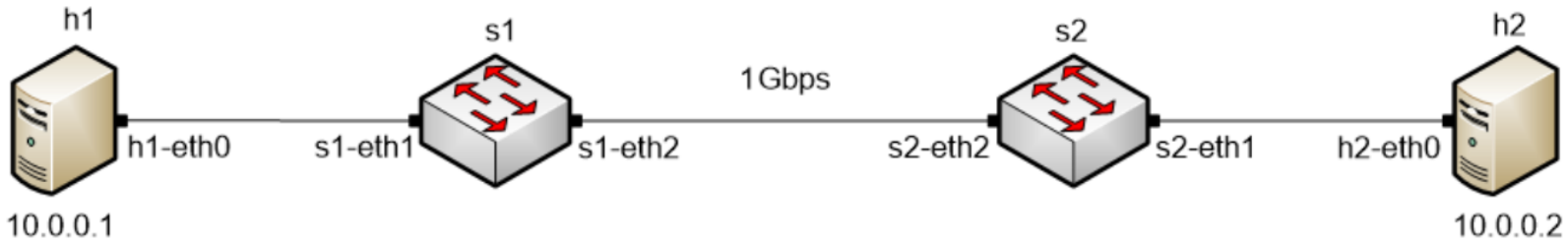
1. N. Cardwell et al. "BBR v2, A Model-based Congestion Control." IETF 104, March 2019.

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**



Lab 7: Understanding Rate-based TCP Congestion Control (BBR)

Lab Goal and Topology

- Deploy emulated WANs in Mininet
- Modify the TCP congestion control algorithm in Linux using sysctl tool
- Compare the performance of TCP Reno and TCP BBR in high-throughput high-latency networks
  - Without 30ms propagation delay
  - With 30ms propagation delay
- Demonstrating the impact of packet loss on the throughput of TCP
- Lab topology: