TCP IN LARGE DATA TRANSFERS

Jorge Crichigno
Department of Integrated Information Technology
University of South Carolina
jcrichigno@cec.sc.edu

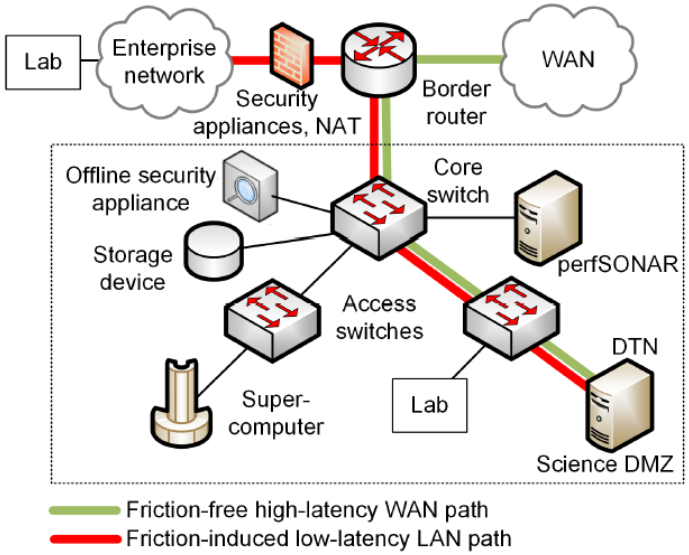
AZ / NM Tribal Consortium / Arizona Community College, Technical Workshop
University of Arizona
January 28, 2020

Agenda

- Science DMZs
- TCP considerations
  - Traditional congestion control algorithms
  - Rate-based congestion control algorithms
  - Routers' buffer size
  - Bandwidth allocation fairness
  - Buffer size management / active queue management (AQM)
- Experimental evaluations
- Resources available for deploying, configuring, and troubleshooting TCP, perfSONAR, Zeek / Bro

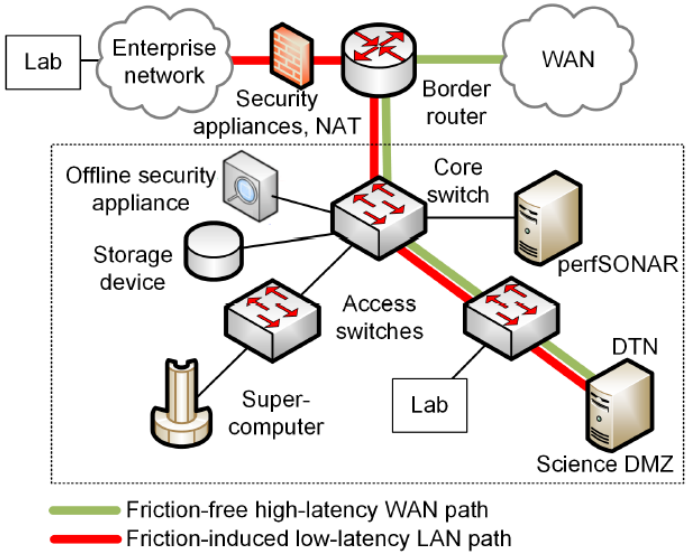
Science DMZ

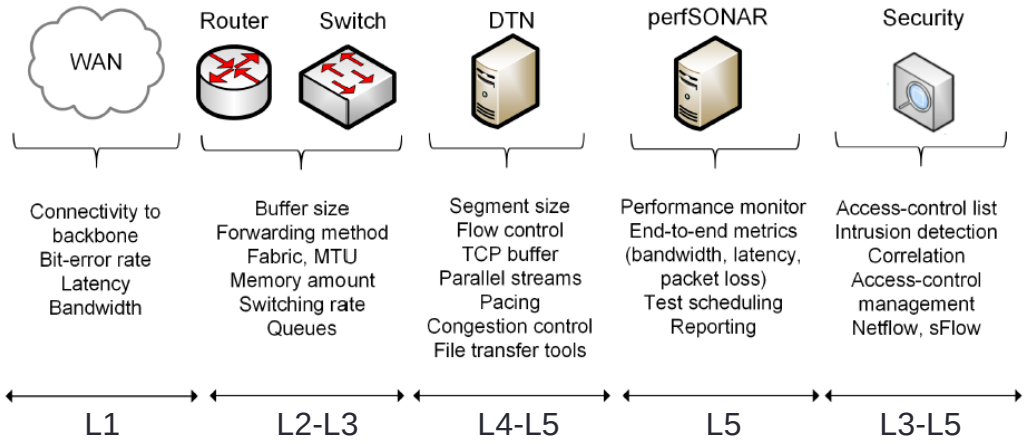
- The Science DMZ is a network designed for big science data
- Main elements
  - High throughput, friction free WAN paths
  - Data Transfer Nodes (DTNs)
  - End-to-end monitoring = perfSONAR
  - Security tailored for high speeds



Science DMZ

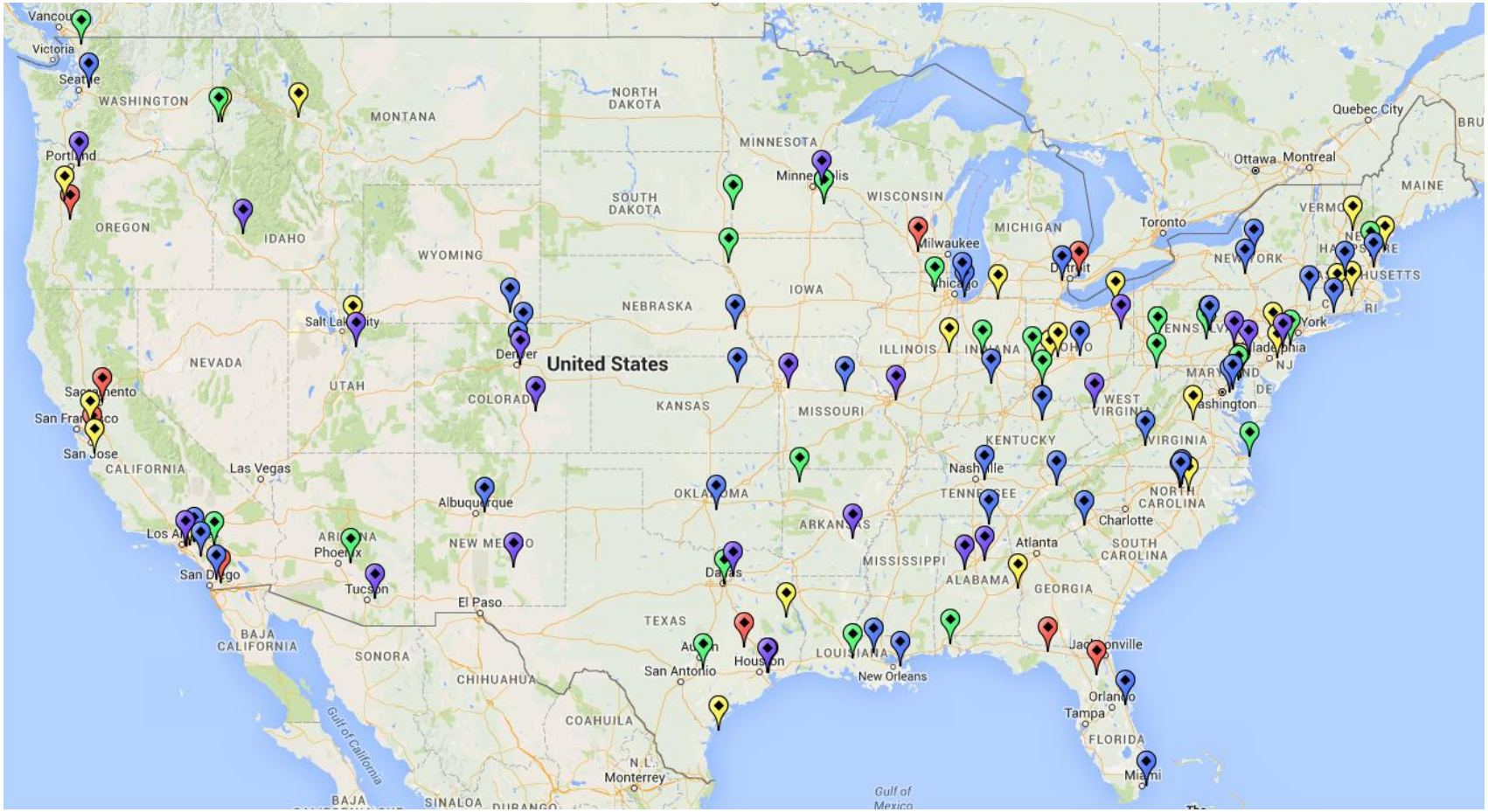
- The Science DMZ is a network designed for big science data
- Main elements
  - High throughput, friction free WAN paths
  - Data Transfer Nodes (DTNs)
  - End-to-end monitoring = perfSONAR
  - Security tailored for high speeds





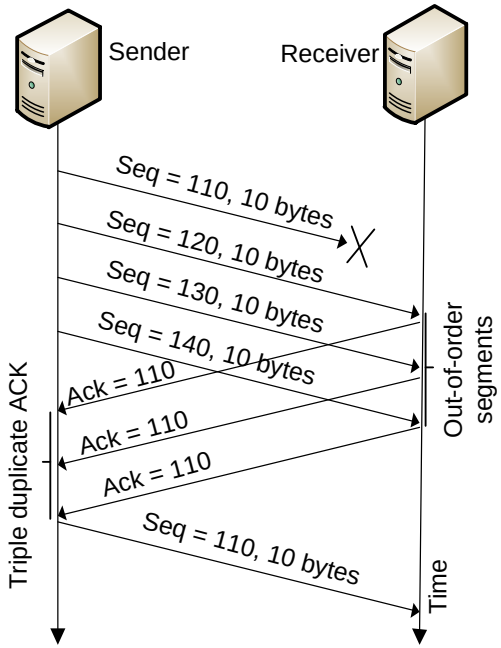
Science DMZ

- Science DMZ deployments, U.S.



TCP Traditional Congestion Control (CC)

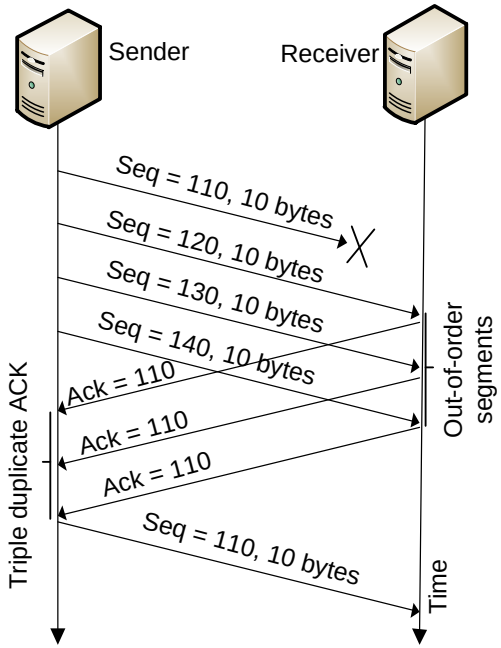
- The principles of window-based CC were described in the 1980s[1]
- Traditional CC algorithms follow the additive-increase multiplicative-decrease (AIMD) form of congestion control

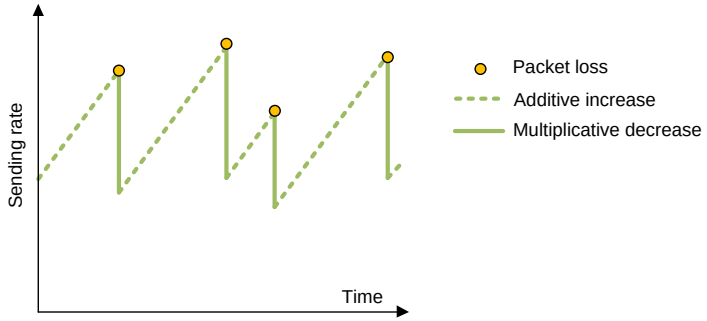


1. V. Jacobson, M. Karels, Congestion avoidance and control, ACM SIGCOMM Computer Communication Review 18 (4) (1988).

TCP Traditional Congestion Control (CC)

- 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

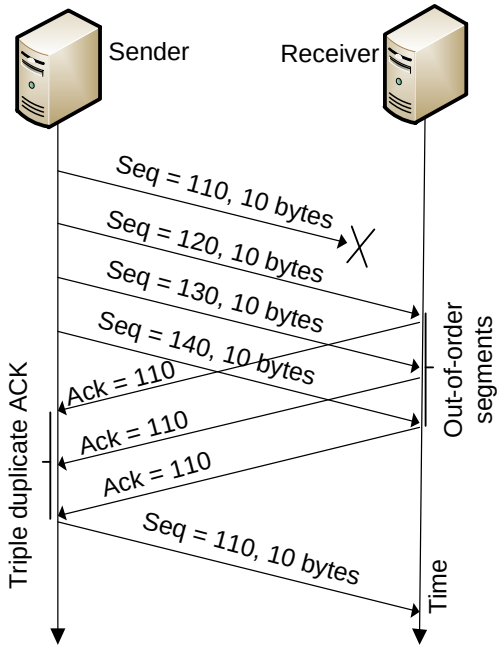


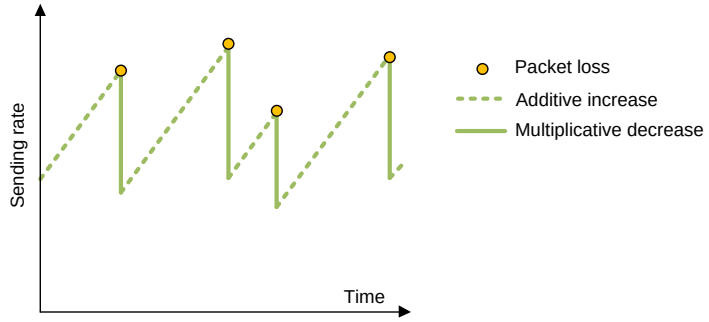


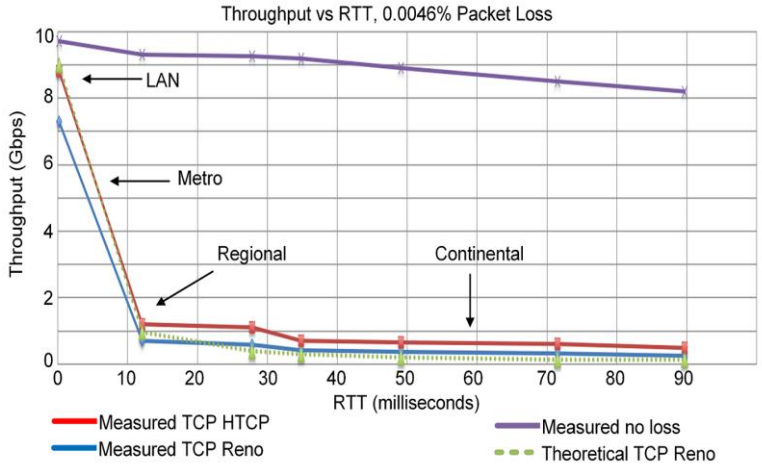
1. V. Jacobson, M. Karels, Congestion avoidance and control, ACM SIGCOMM Computer Communication Review 18 (4) (1988).

TCP Traditional Congestion Control (CC)

- 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).

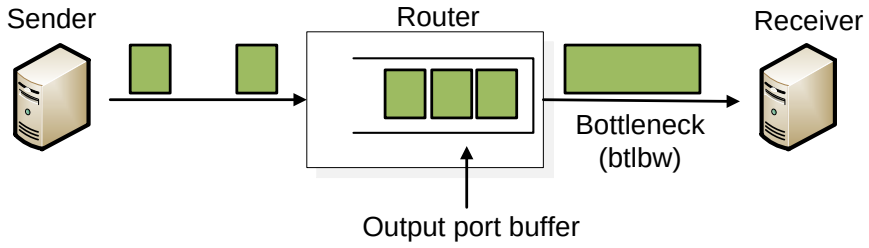
BBR: Rate-based CC

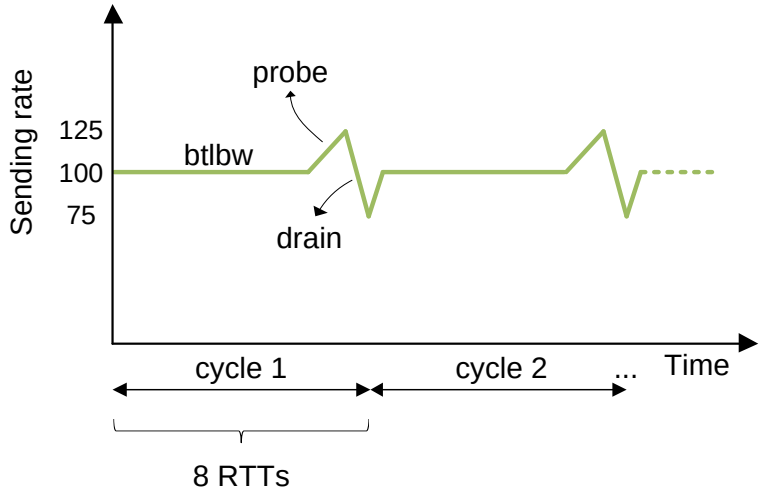
- TCP Bottleneck Bandwidth and RTT (BBR) is a rate-based congestion-control algorithm
- BBR represented a disruption to the traditional CC algorithms
- BBR
  - is not governed by AIMD control law
  - does not the use packet loss as a signal of congestion
- At any time, a TCP connection has one slowest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

1. N. Cardwell, Y. Cheng, C. Gunn, S. Yeganeh, V. Jacobson, “BBR: congestion-based congestion control,” *Communications of the ACM*, vol 60, no. 2, pp. 58-66, Feb. 2017.
2. https://www.thequilt.net/wp-content/uploads/BBR-TCP-Opportunities.pdf

BBR: Rate-based CC

- BBR tries to find btlbw and set the sending rate to that value
  - The sending rate is independent of current packet losses; no AIMD rule

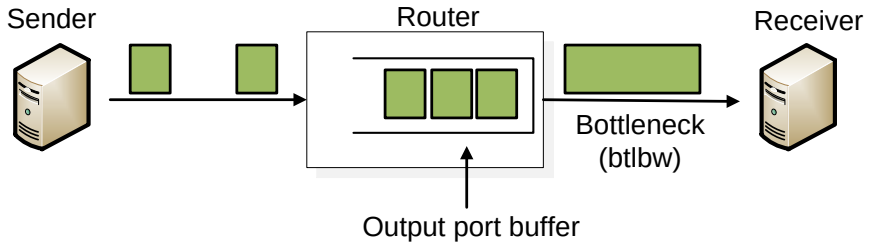


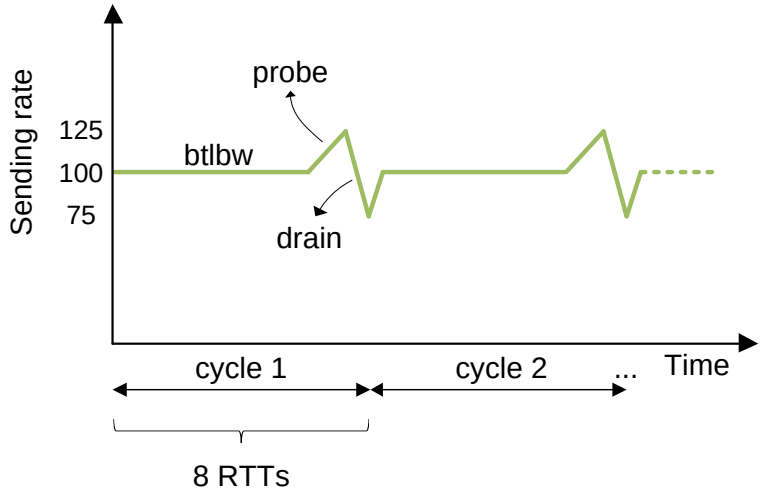


1. N. Cardwell, Y. Cheng, C. Gunn, S. Yeganeh, V. Jacobson, “BBR: congestion-based congestion control,” *Communications of the ACM*, vol 60, no. 2, pp. 58-66, Feb. 2017.
2. https://www.thequilt.net/wp-content/uploads/BBR-TCP-Opportunities.pdf

BBR: Rate-based CC

- BBR tries to find btlbw and set the sending rate to that value
  - The sending rate is independent of current packet losses; no AIMD rule





- BBRv2 has been released for testing:
  - BBR v2: A Model-based Congestion Control: IETF 105 Update - ICCRG (Jul 2019)

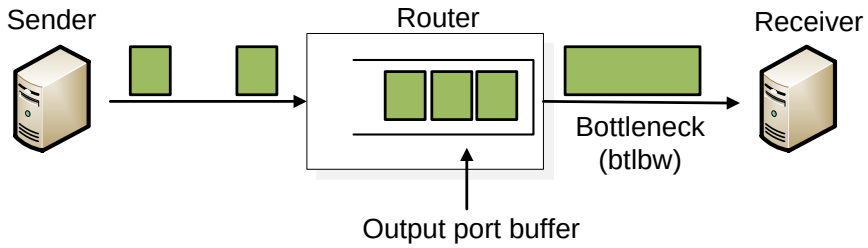
1. N. Cardwell, Y. Cheng, C. Gunn, S. Yeganeh, V. Jacobson, “BBR: congestion-based congestion control,” *Communications of the ACM*, vol 60, no. 2, pp. 58-66, Feb. 2017.
2. https://www.thequilt.net/wp-content/uploads/BBR-TCP-Opportunities.pdf

Open Research Question

What congestion control algorithm is the best for Science DMZ environments?

Buffer Size

- The router's buffer plays an important role in absorbing traffic fluctuations; it avoids losses by momentarily buffering packets as transitory bursts dissipate



Buffer Size

- The rule of thumb has been that the amount of buffering (in bits) in a router's port should equal the RTT (in seconds) multiplied by the capacity C (in bits per seconds) of the port[1]:

$$\text{Router's buffer size} = C \cdot RTT$$

1. C. Villamizar, C. Song, "High performance TCP in ansnet," ACM Computer Communications Review, vol. 24, no. 5, pp. 45-60, Oct. 1994.

Buffer Size

- The rule of thumb has been that the amount of buffering (in bits) in a router's port should equal the RTT (in seconds) multiplied by the capacity C (in bits per seconds) of the port[1]:

$$\text{Router's buffer size} = C \cdot RTT$$

- ESnet: "…you need 50ms of line-rate output queue buffer, so for a 10G switch, there should be around 60MB of buffer…"

1. C. Villamizar, C. Song, "High performance TCP in ansnet," ACM Computer Communications Review, vol. 24, no. 5, pp. 45-60, Oct. 1994.

Buffer Size

- The rule of thumb has been that the amount of buffering (in bits) in a router's port should equal the RTT (in seconds) multiplied by the capacity C (in bits per seconds) of the port[1]:

$$\text{Router's buffer size} = C \cdot RTT$$

- ESnet: "…you need 50ms of line-rate output queue buffer, so for a 10G switch, there should be around 60MB of buffer…"
- When there is a large number of TCP flows passing through a link, say N, the amount of buffering can be reduced to[2]:

$$\text{Router's buffer size} = C \cdot RTT/\sqrt{N}$$

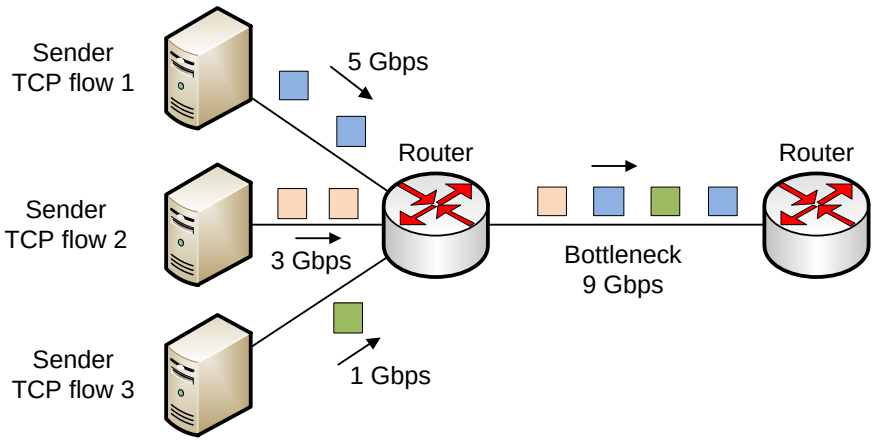
1. C. Villamizar, C. Song, "High performance TCP in ansnet," ACM Computer Communications Review, vol. 24, no. 5, pp. 45-60, Oct. 1994.
2. G. Appenzeller, I. Keslassy, N. McKeown, "Sizing router buffers," in Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications, pp. 281-292, Oct. 2004.

Open Research Question

How large should buffers be?

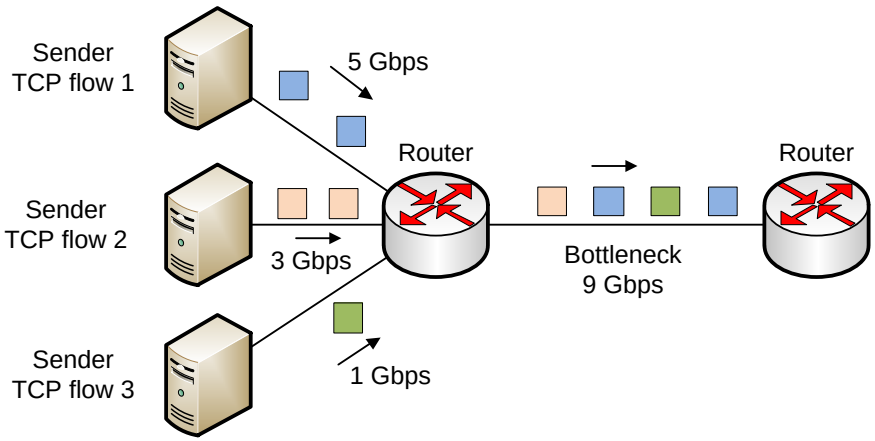
Fairness

- Networks do not use bandwidth reservation mechanism for TCP flows
- Routers simply forward packets based on destination IP address
- The TCP congestion control algorithm ‘allocates’ bandwidth



Fairness

- Fairness is typically measured using the fairness index[1]
- A totally fair system has an index of 1
- A totally unfair system has an index of 0
- The fairness index for the example below is 0.77



1. R. Jain, D. Chiu, W. Hawe, “A quantitative measure of fairness and discrimination for resource allocation in shared computer systems,” DEC Research Report TR-301, Sep. 1984.

Open Research Question

How fair are the congestion control algorithms?

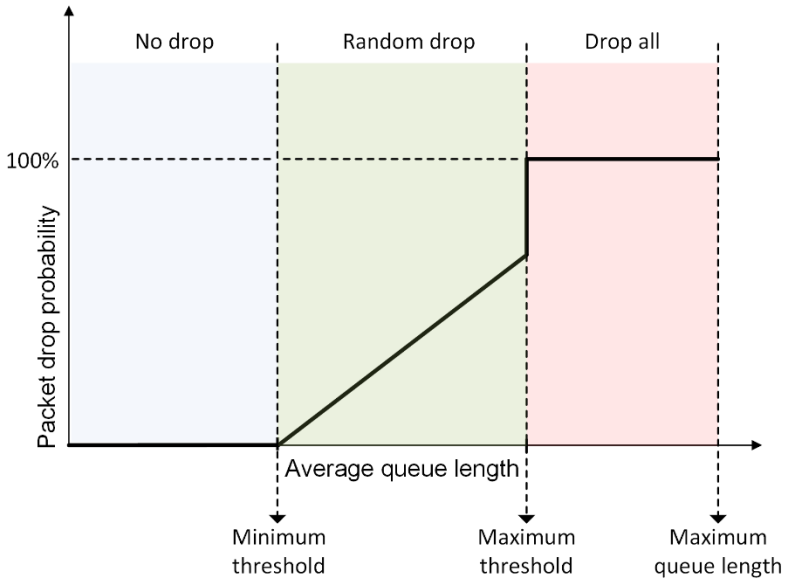
Cubic, Reno, BBRv1, BBRv2…

Active Queue Management (AQM)

- AQM encompasses a set of algorithms to reduce network congestion
- AQM algorithms try to prevent buffers from remaining full
- If the buffer is full, a packet must be dropped
  - Easiest policy is Tail Drop: newly arriving packets are dropped until the queue has enough room to accept incoming traffic

Active Queue Management (AQM)

- AQM encompasses a set of algorithms to reduce network congestion
- AQM algorithms try to prevent buffers from remaining full
- If the buffer is full, a packet must be dropped
  - Easiest policy is Tail Drop: newly arriving packets are dropped until the queue has enough room to accept incoming traffic
  - Random Early Detection: when the queue size is between min. and max. thresholds, drop with certain probability



Active Queue Management (AQM)

- AQM encompasses a set of algorithms to reduce network congestion
- AQM algorithms try to prevent buffers from remaining full
- If the buffer is full, a packet must be dropped
  - Easiest policy is Tail Drop: newly arriving packets are dropped until the queue has enough room to accept incoming traffic
- Other modern policies (some implemented in routers) are
  - Flow Queue Controlled Delay (FQ-CoDEL)[1]
  - Common Applications Kept Enhanced (CAKE)[2]

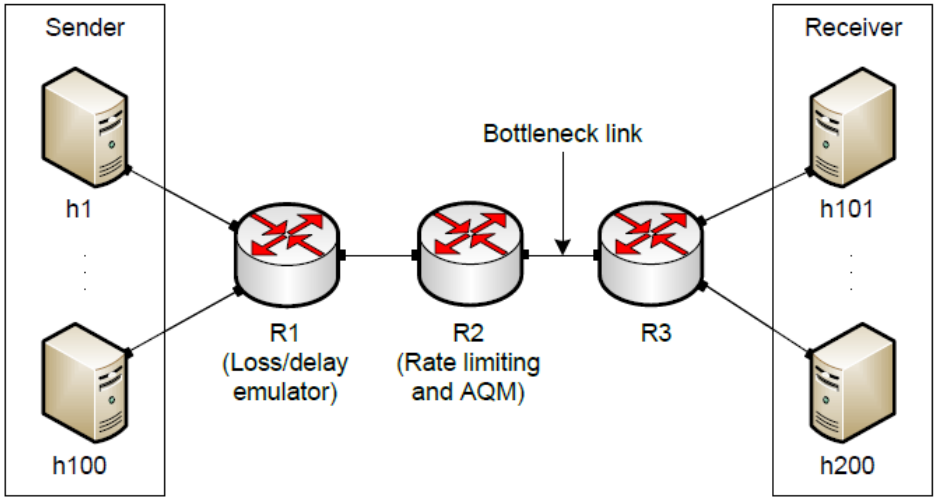
1. T. Hoeiland-Joergensen, P. McKenney, D. Taht, J. Gettys, E. Dumazet, The flow queue CoDel packet scheduler and active queue management algorithm, RFC 8290, 2018.
2. T. Høiland-Jørgensen, D. T¨aht, J. Morton, Piece of CAKE: a comprehensive queue management solution for home gateways, IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), IEEE, 2018, pp. 37–42.

Open Research Question

What is the best AQM policy? How do AQM policies interact with different congestion control algorithms?

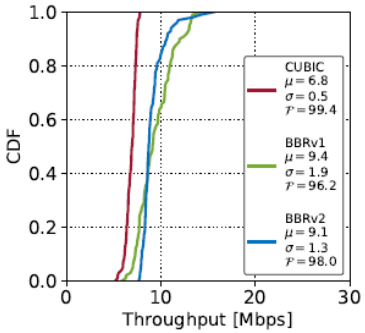
Experiment Results

- Up to 100 simultaneous flows
- Tail drop AQM policy by default
- Mininet network, Linux protocol stack
- Average results are reported

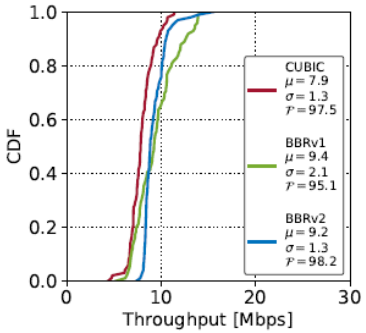


Buffer Size – CC Algorithm

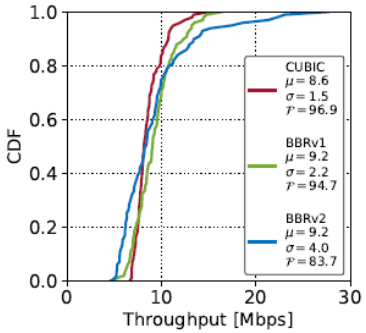
- 100 simultaneous flows, 30ms propagation delay
- Bottleneck is 1 Gbps (ideal allocation is 10 Mbps per flow)
- Cumulative distribution function, mean, standard deviation, and fairness



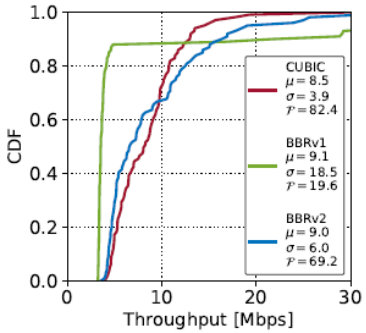
(a) Buffer size: 0.01BDP.



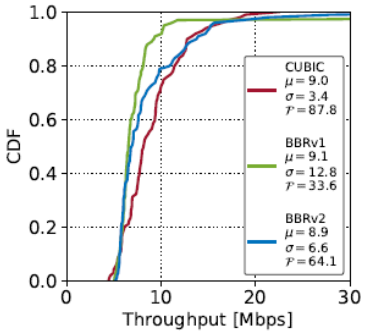
(b) Buffer size: 0.1BDP.



(c) Buffer size: 1BDP.

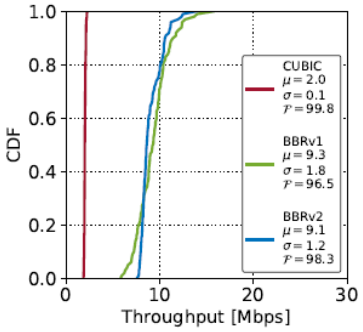


(d) Buffer size: 10BDP.

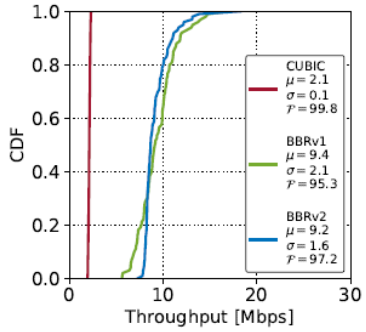


(e) Buffer size: 100BDP.

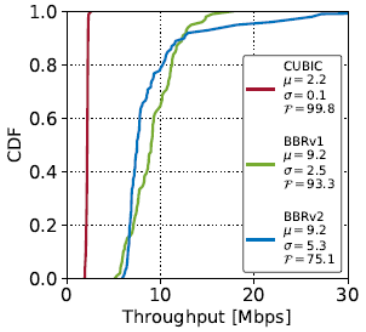
No emulated packet losses



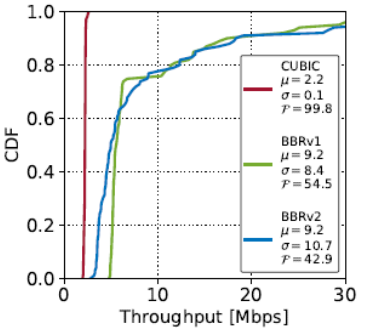
(a) Buffer size: 0.01BDP.



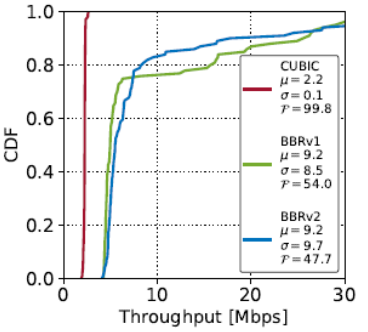
(b) Buffer size: 0.1BDP.



(c) Buffer size: 1BDP.



(d) Buffer size: 10BDP.

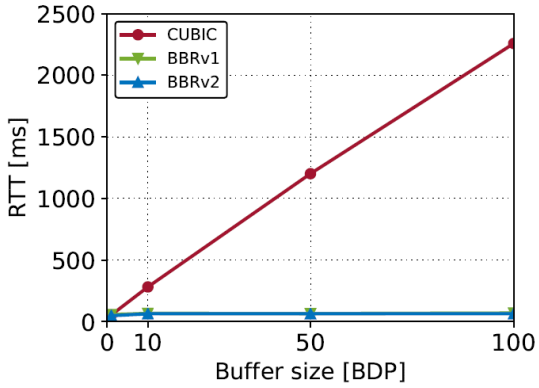


(e) Buffer size: 100BDP.

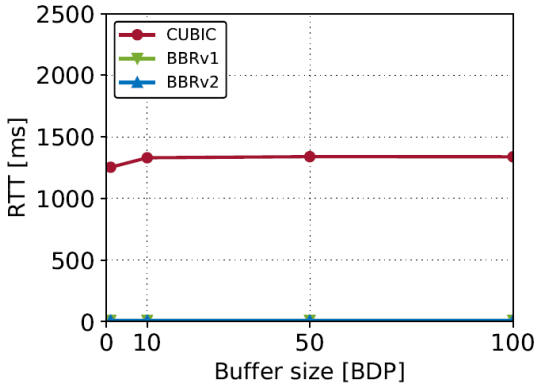
1% packet loss rate

Buffer Size – Queueing Delay

- 30ms propagation delay
- Bottleneck is 50 Mbps
- Round-trip time (RTT) ≈ propagation delay + queueing delay
- Buffer size of 1BDP, 10BDP, 50BDP and 100BDP



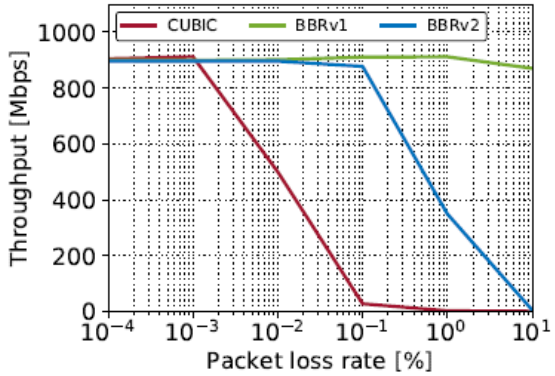
(a) Round-trip time, 2 flows.



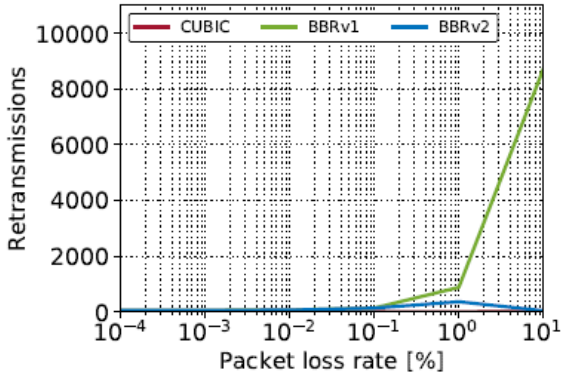
(b) Round-trip time, 100 flows.

Throughput and Retransmissions

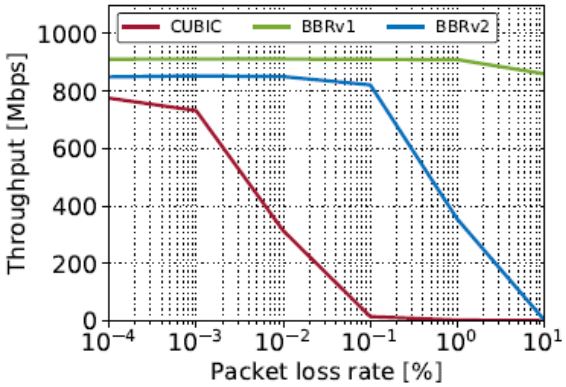
- 30ms propagation delay, 1 flow, 1 Gbps bottleneck
- Throughput and retransmissions, with variable buffer size and packet loss rate



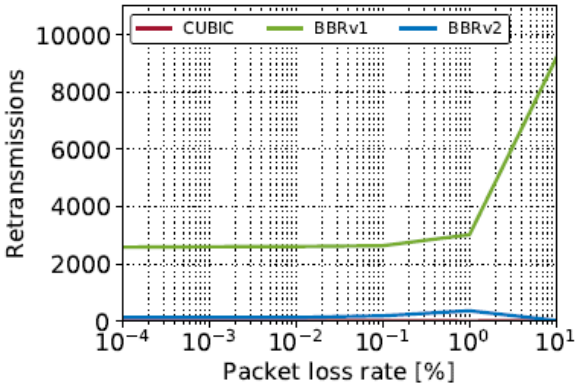
(a) Buffer size: 0.1BDP.



(b) Buffer size: 0.1BDP.



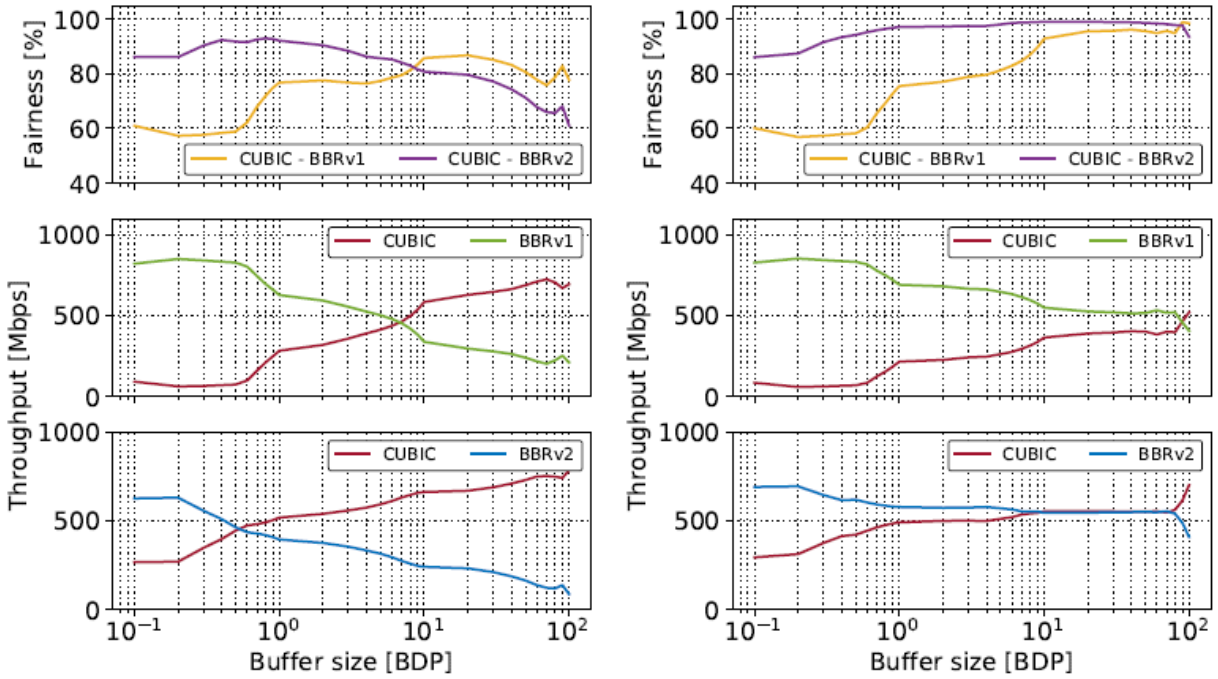
(c) Buffer size: 1BDP.



(d) Buffer size: 1BDP.

Fairness – Congestion Control

- 30ms propagation delay, 1 Gbps bottleneck
- 2 flows using different CC algorithms

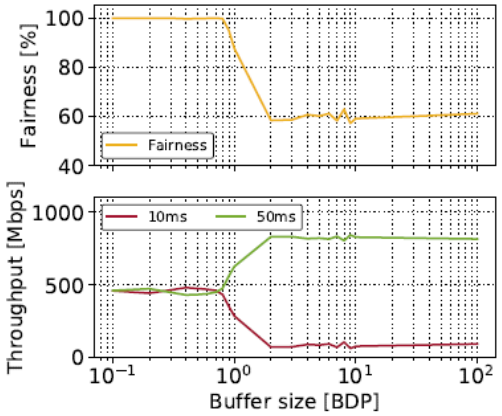


(a) No packet loss, 2 flows.

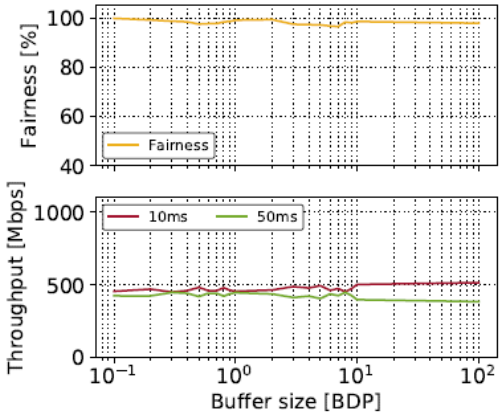
(b) 0.01% packet loss rate, 2 flows.

Fairness - RTT

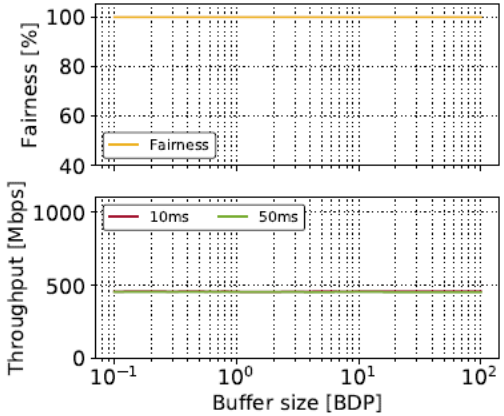
- Traditional CC favors flows with small RTTs
- BBRv1 favors flows with large RTTs; BBRv2?
- One flow with 10ms RTT, competing with another with 50ms RTT



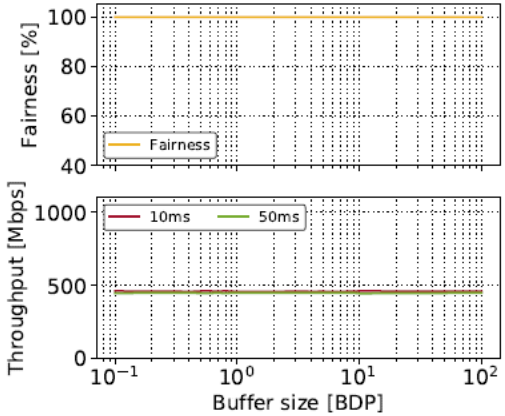
(a) BBRv1 with Tail Drop policy.



(b) BBRv2 with Tail Drop policy.



(c) BBRv1 with FQ-CoDel policy.



(d) BBRv2 with FQ-CoDel policy.

TCP/IP Troubleshooting and Configuration

- Topics covered in this lecture are described with details at:

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

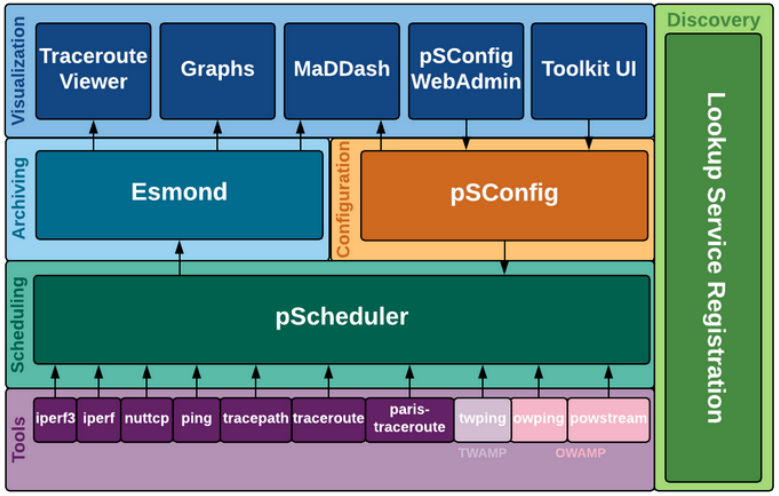
- Virtual training platform includes production devices with step-by-step directions on how to configure and troubleshoot them

Lab Series: Networks Tools and Protocols

- Lab 1: Introduction to Mininet
- Lab 2: Introduction to iPerf
- Lab 3: Emulating WAN with NETEM I Latency, Jitter
- Lab 4: Emulating WAN with NETEM II Packet Loss, Duplication, Reordering, and Corruption
- Lab 5: Setting WAN Bandwidth with Token Bucket Filter (TBF)
- Lab 6: Understanding Traditional TCP Congestion Control (HTCP, Cubic, Reno)
- Lab 7: Understanding 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 Series: perfSONAR

- Lab 1: Configuring Admin. Information Using perfSONAR Toolkit GUI
- Lab 2: PerfSONAR Metrics and Tools
- Lab 3: Configuring Regular Tests Using perfSONAR GUI
- Lab 4: Configuring Regular Tests Using pScheduler CLI Part I
- Lab 5: Configuring Regular Tests Using pScheduler CLI Part II
- Lab 6: Bandwidth-delay Product and TCP Buffer Size
- Lab 7: Configuring Regular Tests Using a pSConfig Template
- Lab 8: perfSONAR Monitoring and Debugging Dashboard
- Lab 9: pSConfig Web Administrator
- Lab 10: Configuring pScheduler Limits



perfSONAR layers

Lab Series: Zeek / Bro

- Lab 1: Introduction to the Capabilities of Zeek
- Lab 2: An Overview of Zeek Logs
- Lab 3: Parsing, Reading and Organizing Zeek Files
- Lab 4: Generating, Capturing and Analyzing Network Scanner Traffic
- Lab 5: Generation, Capturing and Analyzing DoS and DDoS-centric Network Traffic
- Lab 6: Introduction to Zeek Scripting
- Lab 7: Advanced Zeek Scripting for Anomaly and Malicious Event Detection
- Lab 8: Preprocessing of Zeek Output Logs for Machine Learning
- Lab 9: Developing Machine Learning Classifiers for Anomaly Inference and Classification
- Lab 10: Profiling and Performance Metrics of Zeek

Summary

- There are many aspects of TCP / transport protocol that are essential to consider for high-performance networks
  - Parallel streams
  - MSS
  - TCP buffers
  - Router's buffers, and others
- Still there is a need for applied research; e.g.,
  - Performance studies of new congestion control algorithms
  - TCP pacing
  - Application of programmable switches