“ROLE OF TCP IN LARGE DATA TRANSFERS”

J. Crichigno
Department of Integrated Information Technology
University of South Carolina

NSF Award 1829698
“CyberTraining CIP: Cyberinfrastructure Expertise on High-throughput Networks for Big Science Data Transfers”
Motivation for a high-speed science architecture
Enterprise network limitations
Science DMZs
TCP considerations
- Congestion control algorithms
- Parallel streams
- Maximum Segment Size (MSS)
- Pacing, fairness, TCP buffers, router’s buffers, … (discussed in labs)
Motivation for a High-Speed Science Architecture

- Science and engineering applications are now generating data at an unprecedented rate
- Instruments produce hundreds of terabytes in short periods of time ("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

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 CC algorithm determines the sending rate.
- Traditional CC algorithms follow an additive-increase multiplicative-decrease (AIMD) form of congestion control.

![Graph showing TCP Traditional Congestion Control](image_url)
TCP Traditional Congestion Control (CC)

- The CC algorithm determines the sending rate
- Traditional CC algorithms follow an additive-increase multiplicative-decrease (AIMD) form of congestion control
TCP Traditional Congestion Control (CC)

- The CC algorithm determines the sending rate
- Traditional CC algorithms follow an additive-increase multiplicative-decrease (AIMD) form of congestion control
BBR: Rate-based CC

- TCP Bottleneck Bandwidth and RTT (BBR) is a rate-based congestion-control algorithm
- At any time, a TCP connection has one slowest link or bottleneck bandwidth (btlbw)

BBR: Rate-based CC

- TCP Bottleneck Bandwidth and RTT (BBR) is a rate-based congestion-control algorithm
- At any time, a TCP connection has one slowest link or bottleneck bandwidth (btlbw)
- 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

Parallel Streams

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

Legend:
CP: Control process
DP: Data process

FTP model

gridFTP model
Advantages of Parallel Streams

- Combat random packet loss not due congestion\(^1\)
  - Parallel streams increase the recovery speed after the multiplicative decrease

Advantages of Parallel Streams

• Combat random packet loss not due congestion\(^1\)
  ➢ Parallel streams increase the recovery speed after the multiplicative decrease

• Mitigate TCP round-trip time (RTT) bias\(^2\)
  ➢ 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

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
- Overcome TCP buffer limitations
  - An application opening K parallel connections creates a virtual large buffer size on the aggregate connection that is K times the buffer size of a single connection

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
Results on a 10 Gbps Network

- 70-second experiments (first 10 seconds not considered)
- Ten experiments conducted and the average throughput is reported
- Impact of MSS and parallel streams on BBR, Reno, HTCP, Cubic

Results on a 10 Gbps Network

Results on a 10 Gbps Network

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