HighSpeed TCP for High Bandwidth-Delay Product Networks

Raj Kettimuthu

Introduction

- Bulk data transfer has become one of the key requirements in many Grid applications
- GridFTP has been widely deployed for high-speed data transport services
- These services normally require reliable data transfer resulting in TCP as the preferred common base protocol
- Unfortunately TCP performs sub optimally in achieving maximum throughput on the currently available “long fat networks” over the Internet
- This work involves
  - Appropriate instrumentation and study of standard Linux TCP stack, incorporating the recently proposed modifications for high-speed transport

Congestion Control in TCP

- TCP uses two algorithm for congestion control: slow start and congestion avoidance
- The maximum data that can be in flight is min(congestion window, advertised window)
  - Advertised window: flow control imposed by receiver
  - Congestion window: flow control imposed by sender
- Slow start: Congestion window is initialized to 1 segment and each time an ACK is received, the congestion window is increased by one segment

Congestion Control in TCP

- Congestion avoidance: increase in congestion window should be at most one segment each round-trip time (regardless of the number of ACKs are received in that RTT)
- Slow start threshold is used to switch between slow start and congestion avoidance. Exit slow start and enter congestion avoidance when the congestion window goes above slow start threshold
- Fast retransmit and recovery are proposed to improve the performance of TCP by retransmitting without waiting for the retransmit timer to expire

Limited Slow Start

- The Problem:
  - The current slow-start procedure effectively doubles the congestion window in the absence of delayed acknowledgments.
  - For TCP connections that are able to use congestion windows of thousands of segments, such an increase can easily result in thousands of packets being dropped in one round-trip time.
  - This is often counterproductive for the TCP flow itself and is also hard on the rest of the traffic sharing the congested link.

Limited Slow Start

- The Solution – Limited Slow-Start:
  - Limits the number of segments by which the congestion window is increased during slow-start, in order to improve performance for TCP connections with large congestion windows
  - Introduces another threshold called “limited slow start threshold”
    - Enter limited slow-start when the congestion window goes above this threshold
    - During limited slow-start, the congestion window is increased by at most half of the maximum segment size for each arriving acknowledgment
    - Exit limited slow-start and enter congestion avoidance when the congestion window goes above the “slow-start threshold”

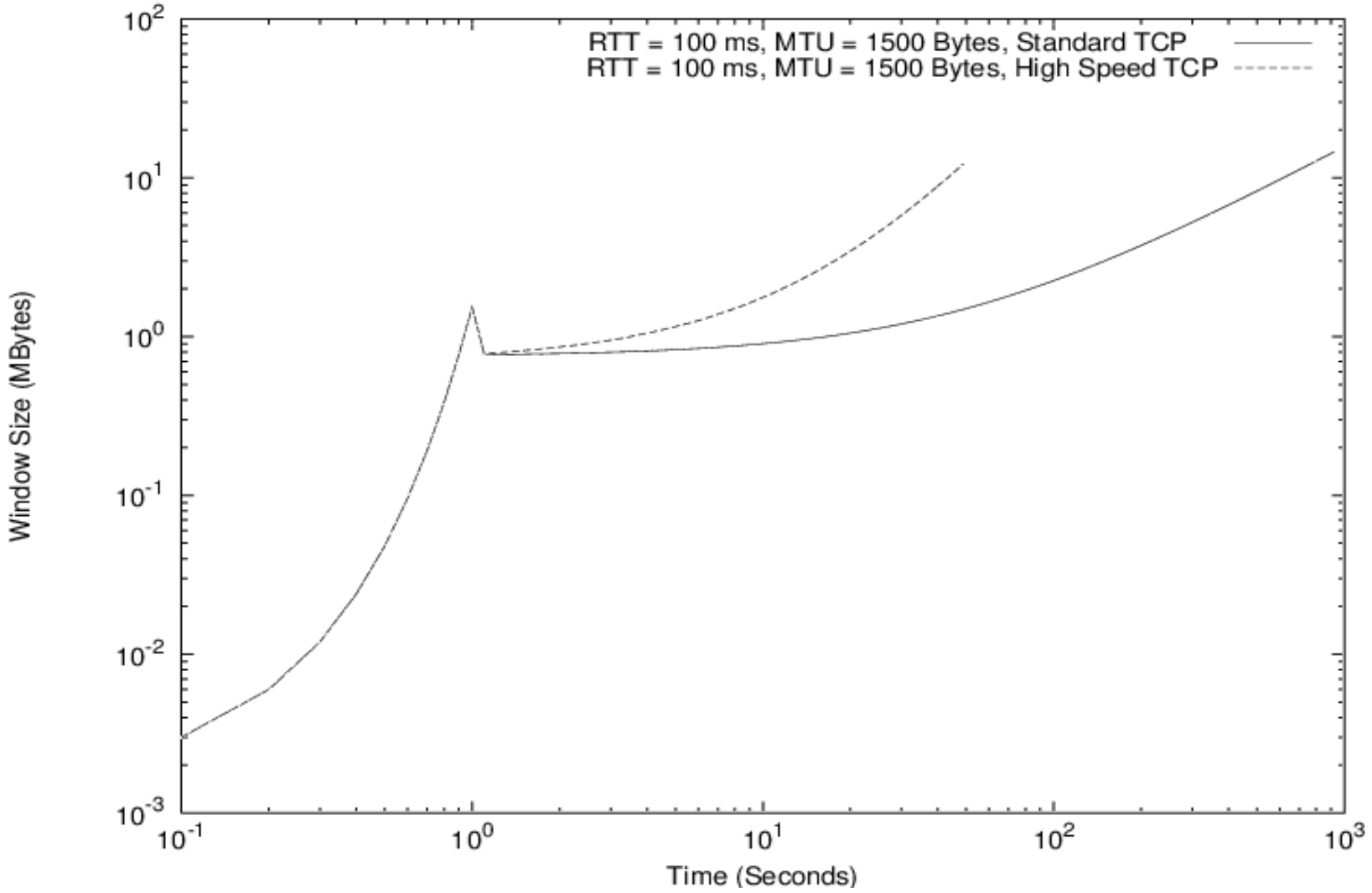
High Speed TCP

- Current standard TCP places a serious constraint on the congestion windows that can be achieved by TCP in realistic environments
- High-speed TCP is a modification to TCP's current congestion control mechanism for high-delay, bandwidth networks
- It introduces a threshold value. If the congestion window is less than the threshold, it uses the normal AIMD algorithm where the additive value is 1 and the decrease factor is 0.5
- If the congestion window is greater than the threshold, it uses High Speed response function to calculate alternate values for AIMD

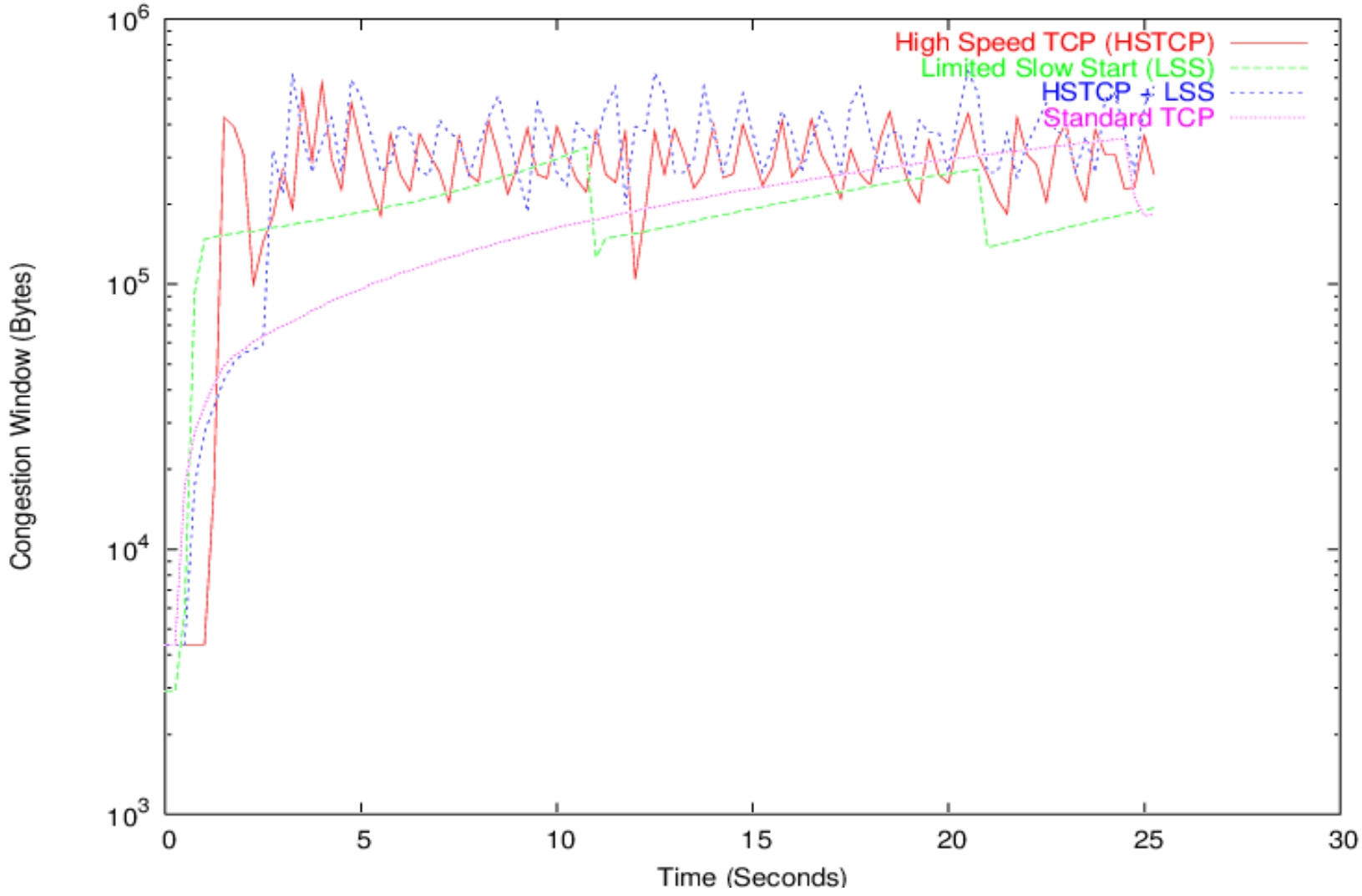
High Speed TCP

- Benefits:
  - Achieves high per connection throughput without requiring unrealistically low packet loss rates
  - Reaches high throughput without long delays when recovering from multiple retransmit timeouts
- The proposed change to the AIMD algorithm may impose a certain degree of unfairness as it does not reduce its transfer rate as much as standard TCP

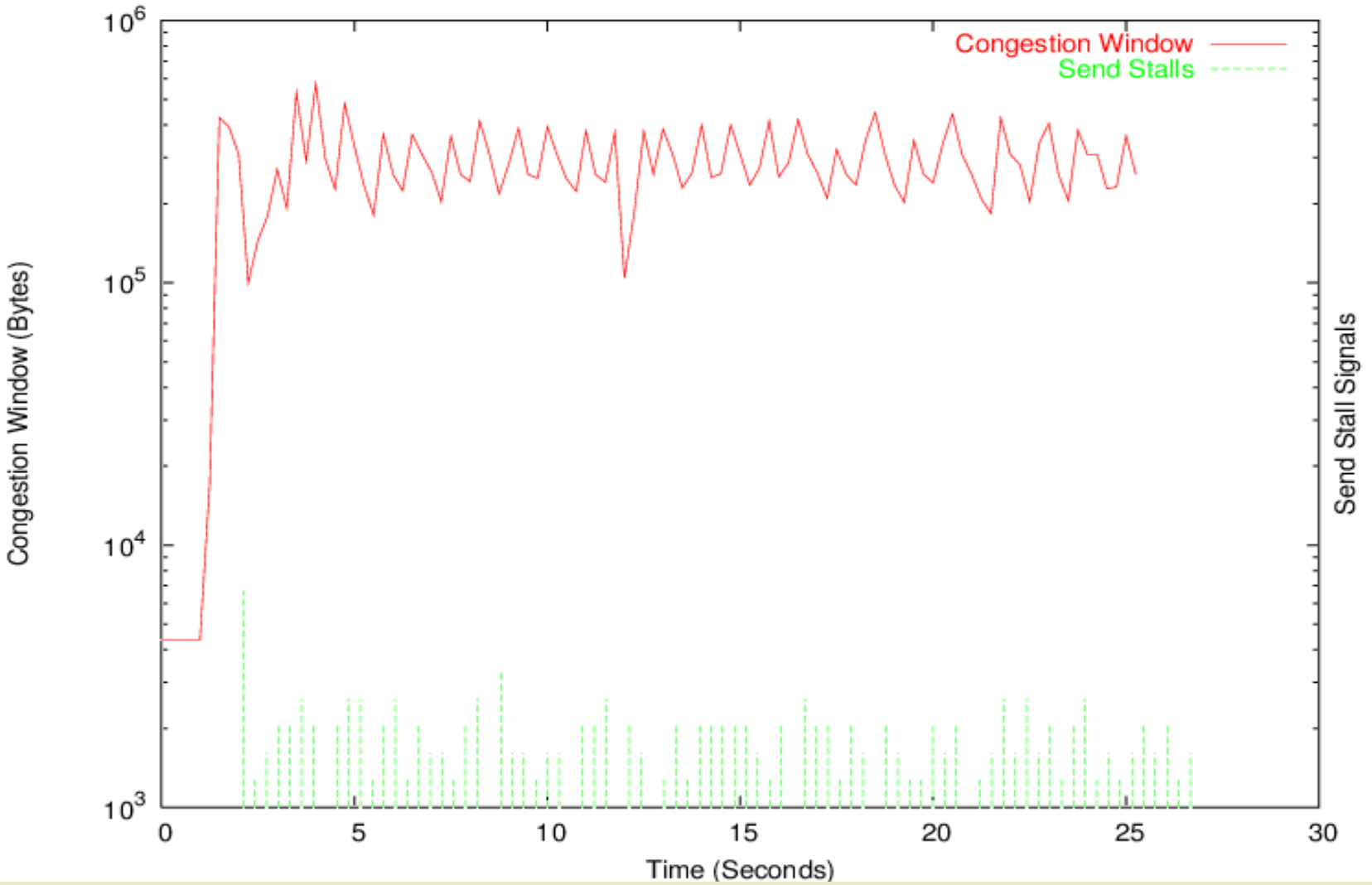
Idealized send window with single congestion event at 1 second



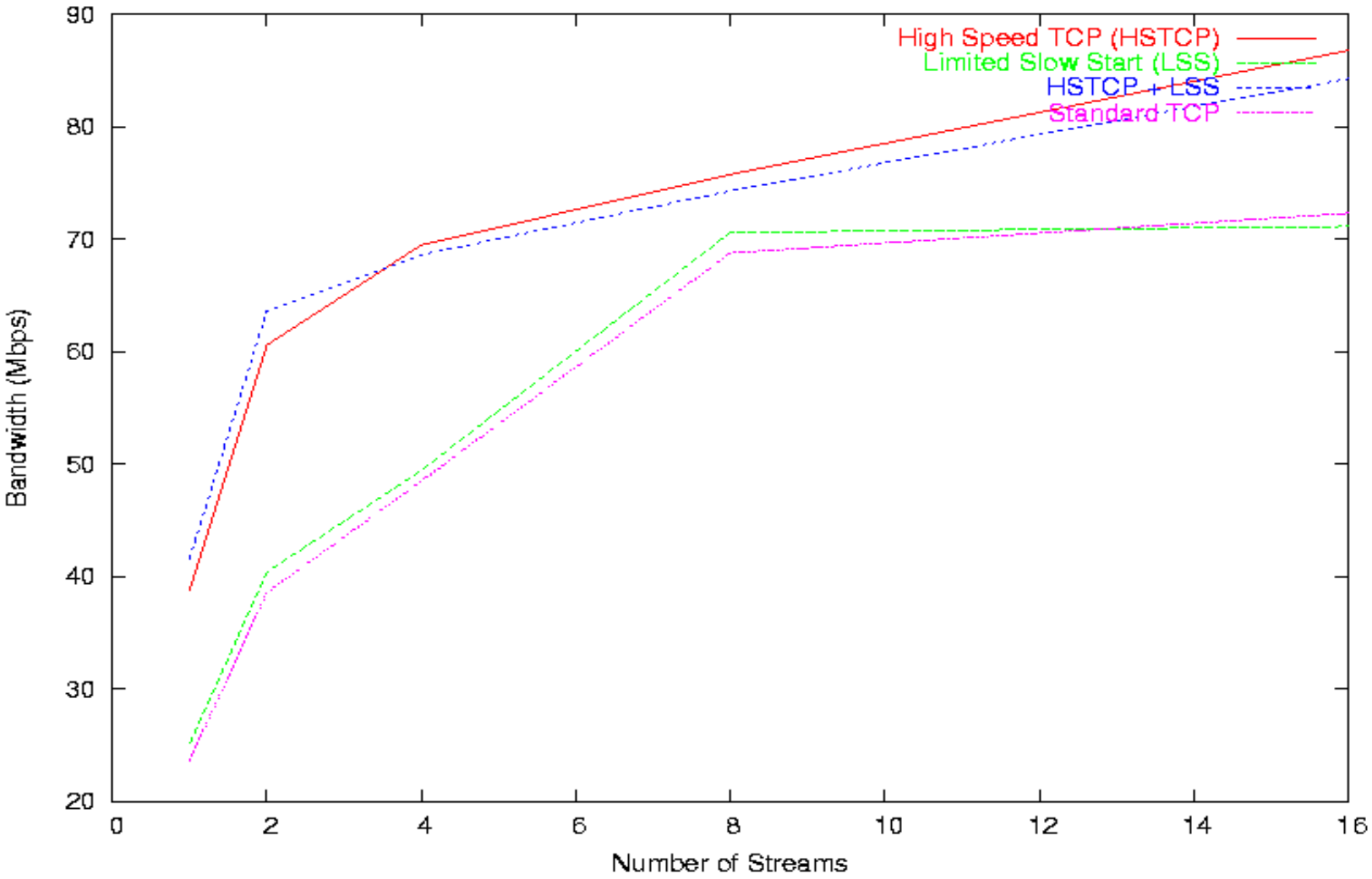
Performance of High-Speed TCP and Limited Slow-Start



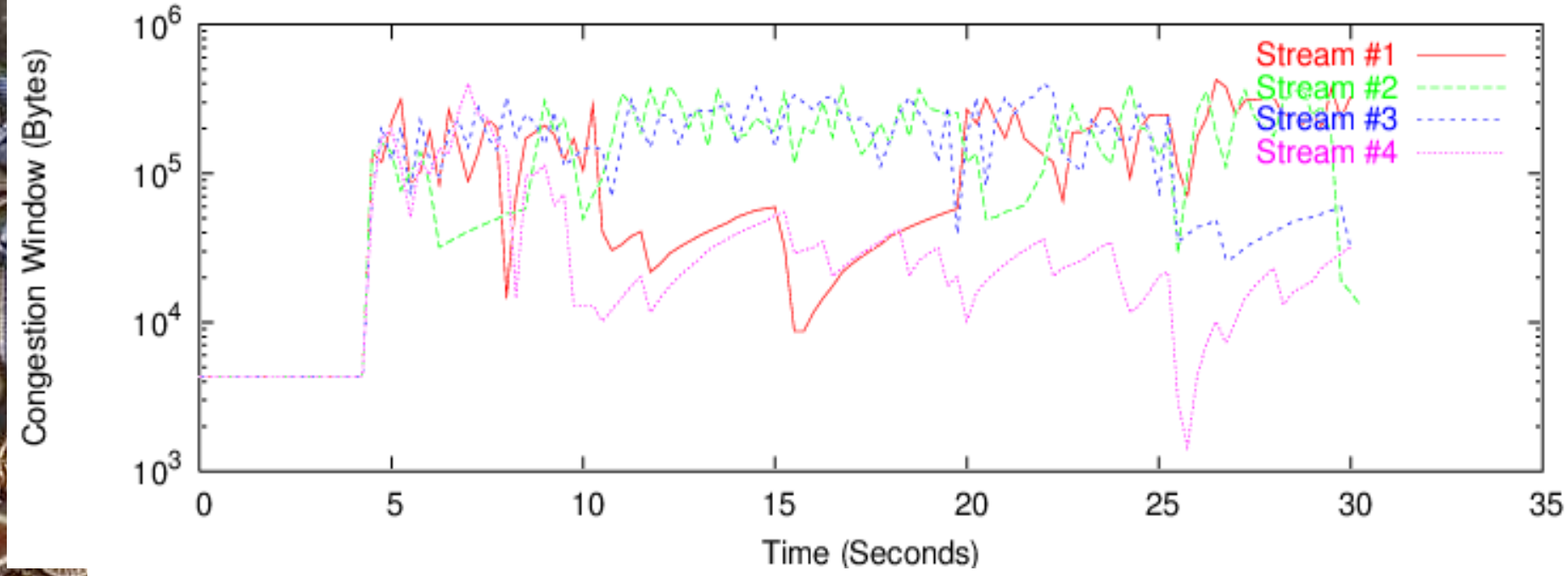
Send stalls

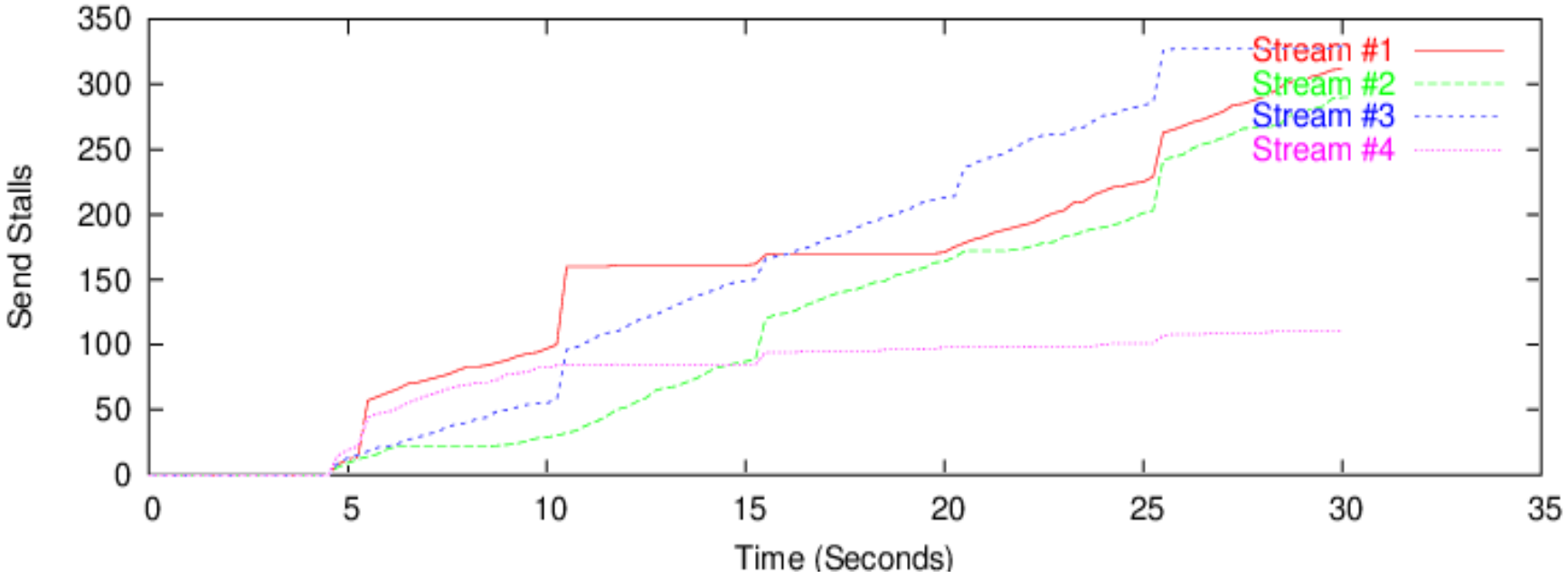

Congestion Window
Send Stalls
Congestion Window (Bytes)
Send Stall Signals
Time (Seconds)
10^6
10^5
10^4
10^3
0
5
10
15
20
25
30

Multiple streams



High-Speed TCP: Interaction of multiple streams





Web100

- Provides improved TCP instrumentation, helps understanding the underlying operation of TCP within a host
- Includes tools for measuring performance and network diagnosis to get a dynamic view of the behavior of the TCP sessions
- Helped identify the cause of extreme round-trip time variance in a recent bulk data transfer experiment
- Helped identify the various possible reasons for drop in the congestion window
- Does not provide information about the process id for a TCP stream;

Future work

- NETBLT (NETwork BLock Transfer):
  - Transfer the data in a series of large data aggregates called “buffers”. The sending NETBLT must inform the receiving NETBLT of the transfer size during connection setup
- RUDP (Reliable UDP)
  - Layered on UDP/IP protocols and provides reliable in-order delivery. EACK is used to specify the out-of-order segments received and unlike TCP the receiver RUDP receiver cannot discard the out-of-order segments
- RBUDP (Reliable Blast UDP)
  - UDP augmented with aggregated acknowledgements to provide reliable bulk data transmission. Acknowledgements are delivered at the end of the transmission phase using a bit vector
- Tsunami
  - A hybrid TCP/UDP based file transfer protocol. It uses UDP for payload and TCP for signaling including request for retransmission