**Title:** HighSpeed TCP

**Contacts:** Sally Floyd (floyd@icir.org)

**URLs / RFCs / Papers:**

- "HighSpeed TCP for Large Congestion Windows," Sally Floyd, Internet draft draft-floyd-tcp-highspeed-02.txt, Work in progress, February 2003.
- http://www.icir.org/floyd/hstcp.html

**Principle / Description of Operation**

It aims at improving the loss recovery time of standard TCP by changing standard TCP's AIMD algorithm. This modified algorithm would only take effect with higher congestion windows. i.e,
If congestion window <= threshold, use Standard AIMD algorithm
Else use HighSpeed AIMD algorithm

| Standard TCP | HighSpeed TCP |
|---|---|
| The standard AIMD algorithm is as follows:<br>*on the receipt of an acknowledgement,*<br>w = w + 1/w ; w -> congestion window<br>*and in response to a congestion event*,<br>w = 0.5 * w | The modified HighSpeed AIMD algorithm is as follows:<br>*on the receipt of an acknowledgement,*<br>w = w + a(w)/w; higher 'w' gives higher a(w)<br>*and in response to a congestion event*,<br>w = (1-b(w))*w; higher 'w' gives lower b(w) |
| The increase and decrease parameters of the AIMD algorithm are fixed at 1 and 0.5 | The increase and decrease parameters vary based on the current value of the congestion window |
| With a 1500-byte packets and a 100 ms RTT, achieving a steady state throughput of 10 Gbps would require a packet drop rate at most once every 1 2/3 hours | For the same packet size and RTT, a steady state throughput of 10 Gbps can be achieved with a packet drop rate at most once every 12 seconds |

**Supported operation mode:**
disk-to-disk (i.e. file transfer protocol, not general transport),
memory to memory (general transport)

**Authentication**: No

**Implementations / API**:

- HighSpeed TCP implementation for Linux 2.4.19 and initial experimental results from Argonne National Lab
- Tom Dunigan has added HighSpeed TCP to the Linux 2.4.16 Web100 kernel.
- Experiments of TCP Stack measurements, from SLAC comparing HighSpeed TCP, FAST TCP, Scalable TCP, and stock TCP. Experiments include TCP Stacks comparison with a single stream, comparisons with multiple streams, and more.
- HighSpeed TCP implementation from Gareth Fairey at Manchester University, for Linux 2.4.19, and initial experimental results with Yee-Ting Li (from UCL).

**Congestion Control Algorithms:** HighSpeed TCP retains the slow start phase of the standard TCP's congestion control algorithm and the congestion avoidance phase is modified as explained above.

**Fairness**: The issue of fairness is not explored thoroughly.

**TCP Friendly**: Unfriendliness increases with decreasing packet drop rates

**Predictable Performance Model:** The increase and decrease parameters are based on a modified response function. More explanation on the rationale behind the new response function and how it can achieve high throughput with realistic packet loss rates is available in the IETF draft.

**Results:** Initial experimental results conducted over 100 Mbps links show that HighSpeed TCP performs much better than the standard TCP achieving an improvement of 150%.

**Target Usage Scenario:** Initial experimental shows that it performs much better than standard TCP in a dedicated environment. Deployment of this in the broader internet might affect the standard TCP flows.