Title: Scalable TCP

Contacts: Tom Kelly (ctk21@cam.ac.uk) Cambridge University, UK

URLs / RFCs / Papers:

- Tom Kelly, "Scalable TCP: Improving Performance in HighSpeed Wide Area Networks," First International Workshop on Protocols for Fast Long-Distance Networks, Geneva, February 2003
- http://www-lce.eng.cam.ac.uk/~ctk21/scalable/

Principle / Description of Operation

The main goal of this work is to improve the loss recovery time of the standard TCP. The idea is built on the idea of HighSpeed TCP.

Packet loss recovery times for a traditional TCP connection (as well as HighSpeed TCP connection) are proportional to the connection's window size and RTT whereas a Scalable TCP connection's packet loss recovery times are proportional to connection's RTT only.

Slow start phase of the original TCP algorithm is unmodified. The congestion avoidance phase is modified as follows:

For each acknowledgement received in a round trip time,

Traditional TCP	*Scalable TCP*
cwnd = cwnd + 1/cwnd	cwnd = cwnd + 0.01

and on the first detection of congestion in a given round trip time

Traditional TCP	*Scalable TCP*
cwnd = cwnd – 0.5 * cwnd	cwnd = cwnd – 0.125 * cwnd

Like HighSpeed TCP this has a threshold window size and the modified algorithm is used only when the size of the congestion window is above the threshold window size. Though values of 0.01 and 0.125 are suggested for the increase and decrease parameters, they (as well as threshold window size) can be configured using the proc file system (by a superuser). The default threshold window size is 16 segments.

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

Authentication: No

Implementations / API:
An implementation for linux kernel 2.4.19 is available at
http://www-lce.eng.cam.ac.uk/~ctk21/scalable/

Congestion Control Algorithms: This work focuses on the congestion control algorithms and proposes a modification for the congestion control algorithm used in the standard TCP.

Fairness: No

TCP Friendly: Claims it does not affect the other standard TCP flows and shows some experimental results involving web traffic to substantiate the claim. More exploration is required before any conclusion can be arrived at this.

Predictable Performance Model: The values of 'a' and 'b' are selected by considering the convergence speed and instantaneous rate variation. The goal was to have faster convergence and smaller instantaneous rate variation.

Results:
The experiments were conducted using a testbed consisting of 12 high performance PCs (6 in Chicago and 6 in CERN, Geneva). The clusters are connected through 2 cisco routers with a 2.4 Gbps link. The PCs are connected to each router through gigabit ethernet ports. The roundtrip time of the connection is 120 ms.
A modified kernel with device transmit queue and receive queue increased to 2000 and 3000 respectively is called gigabit kernel. A significant throughput improvement of 60% to 180% was observed with gigabit kernel compared to standard kernel and a further improvement of 34% to 175% was observed with scalable TCP compared to gigabit kernel Using 16 scalable tcp flows 81% of the maximal performance possible was achieved

Target Usage Scenario:
This is intended to improve the performance of bulk data transport of large data sets with negligible impact on the network traffic. Detailed analysis of the impact on web traffic is yet to be done.