S-shaped TCP: A new transport protocol for fast long-distance networks

Masayoshi Nabeshima
NTT Cyber Solutions Laboratories, NTT Corporation
Yokosuka-shi Kanagawa 2390847 Japan
nabeshima.masayoshi@lab.ntt.co.jp

Abstract

It is well known that TCP does not fully utilize the available bandwidth in fast long-distance networks. To solve this scalability problem, this paper proposes S-shaped TCP (SS-TCP). The window growth function of SS-TCP is, in the steady state, S-shaped, and it can be divided into three regions. In region I, the window growth rate of SS-TCP is the same as that of standard TCP (STD-TCP). In region II, the window of SS-TCP grows faster than that of STD-TCP. In region III, SS-TCP decreases its window growth rate as the network is becoming congested.

1. S-shaped TCP (SS-TCP)

The window growth function of SS-TCP, in the steady state, forms an S-shaped curve while those of standard TCP (STD-TCP) and HighSpeed TCP (HS-TCP) yield a sawtooth curve. The growth function of SS-TCP can be divided into three regions (Fig. 1).

When the time elapsed since the last congestion event, etime, is less than the predefined time, T, SS-TCP lies in region I. In region I, SS-TCP uses the same congestion control algorithm as STD-TCP. That is, the congestion increase and decrease parameters are 1 and 0.5, respectively. In environments where STD-TCP performs well, the congestion epoch time is likely to be small. Thus, the behavior of SS-TCP in region I achieves good TCP friendliness.

When etime is more than T, SS-TCP enters region II. In region II, SS-TCP increases its window size more aggressively than STD-TCP.

\[
\text{cwnd} \leftarrow \text{cwnd} + a(etime) / \text{cwnd}, a(etime) \geq 1
\]

We used \( a(etime) = 1 + 4 \times (etime - T)^2 \) in our simulations. The value of \( a(etime) \) quadratically increases with etime. This is a good characteristic because longer etime suggests that the current environment is the one where STD-TCP does not perform well. However, too large values are obviously harmful to the network. Thus, the maximum is limited to a constant value, \( a_{\text{max}} \). The behavior of SS-TCP in region II ensures good scalability.

SS-TCP uses the same end-to-end estimate of the available bandwidth as TCP Westwood+. The estimate is obtained by filtering the stream of returning ACK packets. A sample of available bandwidth, \( bw_k = d_k / t_k \), is computed every RTT, where \( d_k \) is the amount of data acknowledged during the last RTT = \( t_k \). The estimated bandwidth, \( BW_E \), is updated using the following time-invariant low-pass filter.

\[
BW_E_k = \alpha \cdot BW_E_{k-1} + (1 - \alpha) \cdot bw_k
\]

where \( \alpha \) is a constant. The default value is 0.9.

SS-TCP computes \( W_{\text{max}} \), which is defined as

\[
W_{\text{max}} = (BW_E \cdot SRTT_{\text{max}}) / \text{segment size}
\]

\( SRTT_{\text{max}} \) is the smoothed maximum measured RTT. The key idea is that SS-TCP anticipates that packet losses will occur when the congestion window is close to \( W_{\text{max}} \). Thus, SS-TCP enters region III when the congestion window is close to \( W_{\text{max}} \). More specifically, SS-TCP calculates the window threshold, \( W_{\text{th}} \),

\[
W_{\text{th}} = W_{\text{min}} + \beta \cdot (W_{\text{max}} - W_{\text{min}})
\]

where

\[
W_{\text{min}} = (BW_E \cdot RTT_{\text{min}}) / \text{segment size}
\]
$RT_{\text{min}}$ is the minimum measured RTT. $\beta$ is a constant. If $cwnd$ is more than $W_{th}$, SS-TCP enters region III. The increase parameter $a$ in region III is determined as follows.

$$a = \frac{(1 - \alpha_{\text{last}})cwnd}{W_{\text{max}} - W_{\text{th}}} + \frac{W_{\text{max}} \cdot \alpha_{\text{last}} - W_{\text{th}}}{W_{\text{max}} - W_{\text{th}}}$$

(5)

where $\alpha_{\text{last}}$ is the last value of the increase parameter used in region II. Fig.2 shows the relationship between $a$ and $cwnd$ in region III. The value of $a$ decreases as $cwnd$ approaches $W_{\text{max}}$. This contrasts strongly with HS-TCP. The value of $a$ in HS-TCP always increases with $cwnd$. The number of dropped packets is likely to be proportional to the value of $a$ just before the loss. Thus, SS-TCP can reduce the number of dropped packets, which leads to high throughputs.

When congestion occurs, SS-TCP sets $cwnd$ to $W_{\text{min}}$. This is the same as TCP Westwood+. That is, SS-TCP follows an adaptive decrease paradigm. In addition, the new $SRTT_{\text{max}}$ is computed as:

$$SRTT_{\text{max}} = \gamma \cdot SRTT_{\text{max}} + (1 - \gamma) \cdot RTT_{\text{max}}$$

(6)

where $RTT_{\text{max}}$ is the maximum measured RTT during the current congestion epoch.

Note that $cwnd$ could be more than $W_{\text{max}}$ in a transient state. SS-TCP enters a probing phase when $cwnd$ grows past $W_{\text{max}}$. In the probing phase, the congestion increase parameter $a$ is determined based on the time elapsed since $cwnd$ exceeded $W_{\text{max}}$.

$$cwnd = cwnd + \frac{a(etime2)}{cwnd}, a(etime2) \geq 1$$

where $etime2$ is the time elapsed since $cwnd$ exceeded $W_{\text{max}}$. We used $a(etime2) = 1 + etime2 \times etime2$ in our simulations. The value of $a(etime2)$ increases quadratically as the increase parameter in region II. However, the increase rate in the probing phase is gentler than that in region II. Since SS-TCP does not anticipate the time when the next packet losses occur, it is desirable to increase the window size more slowly in the probing phase than in region II.

We give the pseudo-code of SS-TCP in Fig.3. A long version of this paper is available from http://www005.upp.so-net.ne.jp/nabesima/

![Figure 2. Increase parameter in region III.](image)

![Figure 3. Pseudo-code of SS-TCP](image)