Web object transfer latency over SCTP in the initial slow-start phase

Yong-Jin Lee[1a)] **and M. Atiquzzaman**[2]
[1] *Department of Technology Education, Korea National University of Education*
7 Darak-ri, Chungwon-kun, Chungbuk 363–791, Korea
[2] *School of Computer Science, University of Oklahoma*
Norman, OK 73019, USA
a) *lyj@knue.ac.kr*

Abstract: Most current web application use small objects to increase the data transfer speed between server and client. This leads to completion of the object transfer while the TCP is still in the initial slow-start phase, which starts with an initial window size. SCTP, a new transport layer protocol, uses congestion control mechanism that is similar to that of TCP but complements TCP's deficiencies in web applications. This paper presents an analytical model of object transfer latency for HTTP over SCTP as a function of the initial window size during the initial slow-start phase. Validation of our model using experimental testbed shows that our model results are within 4% of experimental results.

Keywords: SCTP, web application, object transfer latency
Classification: Science and engineering for electronics

References

[1] S. Fu, M. Atiquzzaman, L. Ma, and Y. Lee, "Signaling cost and performance of SIGMA," *Wireless Communication and Mobile Computing*, vol. 5, no. 5, pp. 825–845, 2005.
[2] R. Stewart and Q. Xie, et al., "Stream control transmission protocol," *IETF RFC 2960*, 2000.
[3] D. Emma, S. Loverto, A. Pescape, and G. Ventre, "Measuring SCTP throughput and jitter over heterogeneous networks," *Proceedings the 20th International Conference on Advanced Networking and Applications (AINA-06)*, pp. 395–399, 2006.
[4] P. Natarajan, J. Iyengar, D. Amer, and R. Stewart, "SCTP: An innovative transport layer protocol for the web," *Proceedings 15th International world wide web conference*, pp. 615–624, 2006.
[5] F. Siddiqui and S. Zeadally, "SCTP multihoming support for handoffs across heterogeneous networks," *Proceedings the 4th annual communication networks and services research conference*, pp. 243–250, 2006.
[6] Y. Lee and M. Atiquzzaman, "HTTP transfer latency over SCTP and TCP in slow-start phase," *IEEE International conference on communications (ICC-2007)*, June 24–28, Glasgow, Scotland, 2007.

DOI: 10.1587/elex.6.218
Received December 03, 2008
Accepted February 04, 2009
Published March 10, 2009

1 Introduction

Stream Control Transfer Protocol (SCTP) [1, 2] has been standardized by the Internet Engineering Task Force, and absorbs many of the strengths of TCP, such as window-based congestion control, error detection, and retransmission. Moreover, SCTP incorporated several new features that are not available in TCP. The new features include robustness to DOS attacks, multi-streaming to alleviate head-of-line blocking, and multi-homing for multiple paths between two endpoints, utilizing multiple IP addresses for each point. Like TCP, SCTP congestion control consists of an initial slow-start phase during which the sender probes the network for available amount of bandwidth. Since most HTTP transfers are short and involve only a small amount of data, they are therefore completed within the initial slow-start period, which starts with an initial window size. Consequently, increasing the initial window size is expected to reduce the object transfer latency of HTTP over SCTP. We are interested in quantitatively studying the change in object transfer latency as a function of the change in the initial window size. Previous studies [3, 4, 5] measured the latency in real testbed and simulation. The paper [6] is unique in presenting an analytical model to measure latency in TCP and SCTP for congestion window (cwnd) of one and two, respectively. However, the model is not generalized; it only deals with restricted values of cwnds. Furthermore, it does not present the model validation through real experiments.

In this paper, we derive a generalized analytical model of HTTP transfer latency for any initial congestion window size and present the comparison between analytical results and experimental results. This model can be used to study the performance for SCTP during the initial slow-start phase and performance comparison between TCP and SCTP for web object transfers.

2 Transfer latency modeling in the initial slow-start phase

A timing diagram for object transfer using HTTP over SCTP is illustrated in Figure 1 for initial window size (x) of two MTUs. Therefore, the window size after one and two Round Trip Time (RTT) is four and eight MTUs, respectively. In Figure 1, L_{mtu} and μ represent maximum transfer unit for SCTP (bits) and link transmission rate from the server to the client (bps), respectively. We define T_r as RTT between client and server (sec).

SCTP uses a four-way handshake, where a cookie mechanism is used to stop SYN attacks during association establishment. A web client sends an INIT chunk to the web server. The web server returns an INIT-ACK to the web client. INIT-ACK contains a cookie composed of information that only the web server can verify. On receipt of the INIT-ACK, the web client replies with a COOKIE-ECHO chunk that echoes the cookie and contains the GET request for objects. On receiving the COOKIE-ECHO, the web server checks the cookie's validity, and the sends HTTP replies (object). We define L_{obj} as the size of the object (bits) to be transferred. Then, L_{obj}/L_{mtu} is the number of segments in the object; in Figure 1, $L_{obj}/L_{mtu} = 14$.

DOI: 10.1587/elex.6.218
Received December 03, 2008
Accepted February 04, 2009
Published March 10, 2009

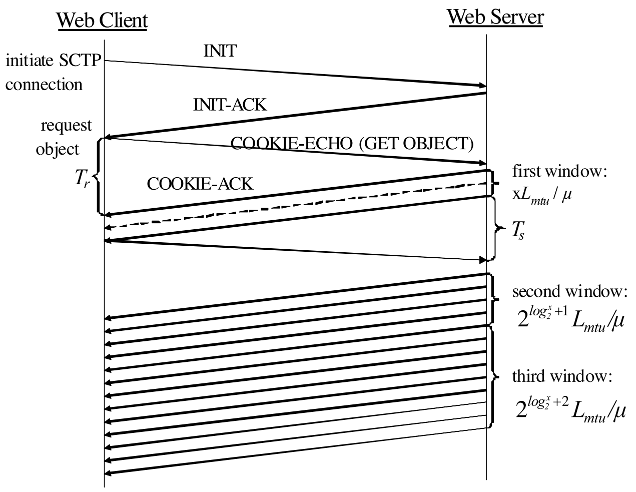

Fig. 1. Timing diagram of HTTP over SCTP during slow-start

We consider the number of segments that are in a window. The first, second and third windows contain two, four, and eight segments, respectively. More generally, the k^{th} window contains 2^k segments. We need α windows in order to completely send the object; in Figure 1, $\alpha = 3$. Generally, α can be expressed in terms of L_{obj}/L_{mtu} as follows:

$$\begin{aligned}\alpha &= \min\left\{k : 2^{x-1} + 2^x + \cdots + 2^{y+k-1} \geq \frac{L_{obj}}{L_{mtu}}\right\} \\ &= \left\lceil \log_2^{\left(\frac{L_{obj}}{L_{mtu}}+x\right)} \right\rceil - \log_2^x \\ &\quad \textit{where } y = \log_2^x\end{aligned} \tag{1}$$

The server may stall after transmitting a window of data and waiting for an acknowledgement. For example, in Figure 1, the server stalls after transmitting the first window. We now consider the stall time after transmitting the k^{th} window. The time from the start of transmission of the k^{th} window until the reception of acknowledgement of the first segment is $2^{\log_2^{x+k-1}} \times L_{mtu}/\mu + T_r$.

The transmission time of the k^{th} window is $(L_{mtu}/\mu)2^k$. Server stall time (T_s) is defined as the difference between these two quantities. In Figure 1 $(x = 2)$, the stall time for the first window $(k = 1)$ is T_r; the second window $(k = 2)$, $T_r - 3L_{mtu}/\mu$; the third window $(k = 3)$, $T_r - 7L_{mtu}/\mu$. The generalization leads to

$$\begin{aligned}T_s &= \frac{L_{mtu}}{\mu} + T_r - 2^{y+k-1}\left(\frac{L_{mtu}}{\mu}\right) \\ &\quad \textit{where } y = \log_2^x\end{aligned} \tag{2}$$

The server may stall after the transmission of each of the first $\alpha - 1$ windows. The object transfer latency for HTTP over SCTP is composed of setting up the SCTP connection, requesting the object, transmission of the object, and the sum of all the stalled times. Thus,

DOI: 10.1587/elex.6.218
Received December 03, 2008
Accepted February 04, 2009
Published March 10, 2009

$$LT = 2T_r + \frac{L_{obj}}{\mu} + \sum_{k=1}^{\alpha-1}\left[\frac{L_{mtu}}{\mu} + T_r - 2^{y+k-1}\left(\frac{L_{mtu}}{\mu}\right)\right] \qquad (3)$$
$$\textit{where } y = \log_2^x$$

To obtain a more general expression for the latency, we introduce the number of server stalls when the object contains an infinite number of segments (β). We obtain Eqn. (4) by using a derivation similar to that for α. The actual number of times (γ) that the server stalls is $\gamma = \min\{\alpha - 1, \beta\}$.

$$\begin{aligned}\beta &= \max\left\{k : \frac{L_{mtu}}{\mu} + T_r - \frac{L_{mtu}}{\mu}2^{y+k-1} \geq 0\right\}\\ &= \left\lfloor \log_2^{\left(1+\frac{\mu T_r}{L_{mtu}}\right)} \right\rfloor + 1 - y \qquad (4)\\ &\textit{where } y = \log_2^x\end{aligned}$$

Combining Eqns. (3) and (4), we obtain Eqn. (5) for the object transfer latency of HTTP over SCTP.

$$\begin{aligned}LT_{sctp} &= 2T_r + \frac{L_{obj}}{\mu} + \sum_{k=1}^{\gamma}\left[\frac{L_{mtu}}{\mu} + T_r - 2^{y+k-1}\left(\frac{L_{mtu}}{\mu}\right)\right]\\ &= 2T_r + \frac{L_{obj}}{\mu} + \gamma\left(T_r + \frac{L_{mtu}}{\mu}\right) - 2^y(2^\gamma - 1)\frac{L_{mtu}}{\mu} \qquad (5)\\ &\textit{where } y = \log_2^x\\ &\gamma = \min\left\{\left\lfloor \log_2^{\left(1+\frac{\mu T_r}{L_{mtu}}\right)} \right\rfloor + 1 - \log_2^x, \left\lceil \log_2^{\left(\frac{L_{obj}}{L_{mtu}}+x\right)} \right\rceil - \log_2^x - 1\right\}\end{aligned}$$

Even though theoretical initial congestion window of TCP is equal to that of SCTP, TCP on real operating system sends less data than the initial congestion window. This means that TCP needs more RTT than SCTP to transfer the same amount of data. Since two additional RTTs for TCP on real operating system are required, the transfer latency for HTTP over TCP is larger than HTTP over SCTP by 2 × RTT. Thus,

$$\begin{aligned}LT_{tcp} &= 4T_r + \frac{L_{obj}}{\mu} + \gamma\left(T_r + \frac{L_{mtu}}{\mu}\right) - 2^y(2^\gamma - 1)\frac{L_{mtu}}{\mu}\\ &\textit{where } y = \log_2^x \qquad (6)\\ &\gamma = \min\left\{\left\lfloor \log_2^{\left(1+\frac{\mu T_r}{L_{mtu}}\right)} \right\rfloor + 1 - \log_2^x, \left\lceil \log_2^{\left(\frac{L_{obj}}{L_{mtu}}+x\right)} \right\rceil - \log_2^x - 1\right\}\end{aligned}$$

3 Performance evaluation

In order to validate our model, we simulated the web server and client by transferring objects between two machines using SCTP and TCP sockets, respectively. Table I gives details of our experimental setup.

Table II shows mean transfer latencies obtained from our model and experiments when $L_{obj} = 50\,\text{KB}$, $L_{mtu} = 1500\,\text{B}$, $T_r = 0.1$ second and number of objects = 5. NIST emulator installed on the router was used to simulate

DOI: 10.1587/elex.6.218
Received December 03, 2008
Accepted February 04, 2009
Published March 10, 2009

Table I. Experimental setup

	client	*server*	*router*
Hardware	Pentium IV 2.4 GHz RAM 1.28 GB HDD 40 GB	Pentium P3 500 MHz RAM 128 MB HDD 528 MB	Pentium III 500 MHz RAM 1.28 GB HDD 40 GB
Operating System	Linux UBUNTU (Kernel 2.6.15)	Linux UBUNTU (Kernel 2.6.15)	Free BSD 6.1
IP Address	192.168.0.10	192.168.1.10	192.168.1.1 192.168.0.1
SCTP module	lksctp-2.6.15-1.0.5	lksctp-2.6.15-1.0.5	

Table II. Mean transfer latency for model and experiments

Bandwidth	Exp(SCTP)	Model(SCTP)	Exp(TCP)	Model(TCP)
28 Kbps	14.12	14.68	14.71	15.08
56 Kbps	7.36	7.54	7.70	7.94
100 Kbps	4.39	4.40	4.70	4.80
1 Mbps	1.14	1.10	1.49	1.50
10 Mbps	1.02	0.84	1.42	1.24

various bandwidths between the server and client. The client program for HTTP over SCTP sent five objects on multiple streams using round robin; on the other hand, the client program for HTTP over TCP sends five objects on a single stream using round robin. Mean transfer time was obtained from 10-run experiments with data captured with ethereal protocol analyzer. Exp(TCP) and Exp(SCTP) represent transfer latencies for HTTP over TCP and HTTP over SCTP in the experiment setup, respectively. Model(SCTP) and Model(TCP) represent mean transfer latencies from Eqns. (5) and (6), respectively. In Table II, the mean difference between the model and experimental results is 4%. This small difference in the results between model and experiment were due to the inaccuracies of the NIST emulator in simulating bandwidth. The performance evaluation, therefore, demonstrates the accuracy of our model.

4 Conclusions

We investigated the mean object transfer latency for HTTP over SCTP in the initial slow-start phase, which had not been considered in the literature. Most previous studies compared the performance between TCP and SCTP in web environment using simulation or experiments. On the contrary, we derived analytical models to represent the mean object transfer latency with varying initial window. We validated our model using experimental testbed. Comparison of results from our model and experiment shows that our model results are within 4% of experimental results.

DOI: 10.1587/elex.6.218
Received December 03, 2008
Accepted February 04, 2009
Published March 10, 2009