This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.

HTTP Transfer Latency over SCTP and TCP in Slow Start Phase

Yong-Jin Lee, *Member*; *IEEE* and M. Atiquzzaman, *Senior Member, IEEE*

***Abstract*— Current web applications use HTTP (Hyper Text Transfer Protocol) over TCP (Transmission Control Protocol) to retrieve objects over the Internet. SCTP (Stream Control Transmission Protocol) is a recently proposed transport protocol with very similar congestion control mechanisms as TCP, except the initial congestion window during the slow start phase. In this paper, we present an analytical model of object transfer latency during the slow start phase for HTTP over SCTP and compare with the latency of HTTP over TCP. Validation of the model using experimental results, show that the average object transfer latency for HTTP over SCTP during the slow start phase is less than that for HTTP over TCP by 11%.**

Index Terms—SCTP, transfer latency, slow start, congestion control.

I. Introduction

WORLD WIDE WEB distributed hypermedia system uses HTTP (hyper text transfer protocol) as a transfer protocol to retrieve object in the Internet. Since HTTP is a connection-oriented protocol, both web server and clients use TCP (transmission control protocol) as the transport layer protocol. Web server (TCP sender) regulates its sending rate as a function of perceived congestion by using TCP's congestion control algorithm consisting of slow-start and congestion avoidance.

A new TCP connection in a web server starts in a *slow start phase* with a congestion window of one segment. On receipt of the first acknowledgement after one round trip time (RTT), the web server increases its congestion window to two segments. The process continues, with the congestion window doubling every RTT until the congestion window reaches a predefined threshold, after which TCP enters the congestion avoidance phase. During the congestion avoidance phase, the web server linearly increments the congestion window every RTT. Most HTTP transfers are short and involve only a small amount of data; most transfers are therefore completed within the slow start period. Consequently, this paper *focuses* on the slow start period of TCP and SCTP.

SCTP (Stream Control Transfer Protocol) [1]-[2] has been standardized by IETF, 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. SCTP's slow start, however, differs from that of TCP in the initial congestion window size.

The *objective* of this paper is to determine the effect of the differences in the slow start mechanisms of TCP and SCTP on the performance of object transfers over the Web using HTTP. We carry out the study by developing analytical models for object transfer latency of HTTP over TCP and SCTP. The *contribution* of this paper is analytical models for comparing object transfer latencies of HTTP over TCP and SCTP.

This work was supported in part by the Korea Research Foundation Grant funded by the MOEHRD (KRF-2006-521-D00399).

Yong-Jin Lee is with the Department of Technology Education, Korea National University of Education, 363-791, Korea (e-mail: lyj@ knue.ac.kr).

M. Atiquzzaman is with the School of Computer Science, University of Oklahoma, Norman, OK 73019, USA. (e-mail: atiq@ou.edu).

II. Object Transfer Latency for HTTP over SCTP and TCP

A. Congestion Control of SCTP and TCP

SCTP's congestion control is based on and very similar to the well proven rate-adaptive window-based congestion control of TCP. The common features include the adoption of slow start, congestion avoidance, timeout and fast retransmit algorithms. However, there are several major differences between the congestion control mechanisms of TCP and SCTP. We list below the differences between the congestion control of TCP and SCTP [3].

- SCTP doesn't have an explicit fast-recovery phase. SCTP achieves fast recovery implicitly through the use of SACK [1].
- SCTP begins slow start algorithm from cwnd = 2 instead of one in TCP.
- Mandatory use of SACK in SCTP allows more robust reaction in the case of multiple losses from a single window of data. This avoids a time-consuming slow start stage after multiple segment losses, thus saving bandwidth and increasing throughput.
- TCP begins fast retransmit after the receipt of three Duplicate Acknowledgements (DupACKs); SCTP begins after four DupACKs. However, SCTP is able to clock out new data

on receipt of the first three DupACKs, and can also retransmit a lost segment by ignoring whether the flight size is less than cwnd.

B. Notations

The following notations are used in our analytical models for HTTP over SCTP and TCP.

L_{obj}	size of the object to be transferred (bits)
L_{mtu}	maximum transfer unit for SCTP (bits)
L_{mss}	maximum segment size for TCP (bits)
μ	link transmission rate from the server to the client (bps)
T_r	round trip time between client and server (sec)
ST	stall time during which server cannot transfer (sec)
α	number of windows necessary to transfer the object
β	number of server stalls when the object contains an infinite number of segments
γ	actual number of server stalls
LT_{TCP}	object transfer latency for HTTP over TCP
LT_{SCTP}	object transfer latency for HTTP over SCTP

C. Object Transfer Latency for HTTP over SCTP

In SCTP, the initial congestion window before data transmission or after a sufficiently long idle period must be equal to two MTUs (maximum transfer unit), whereas TCP starts with an initial congestion window of one MSS (maximum segment size). Of course, according to the current TCP congestion control specification, TCP implementations can use an initial congestion window up to four segments [4]. For example, NetBSD v1.3 uses an initial congestion window with the value four. In this paper, however, we compare the standards tracks RFC-2001 [5] for TCP and RFC-2960 [2] for SCTP.

A timing diagram for object transfer using HTTP over SCTP is illustrated in Figure 1. SCTP uses a four-way handshake, where a cookie mechanism establishes an association to stop blind SYN attacks. A Web client sends an INIT chunk to web server. The web server returns an INIT-ACK to the web client. This INIT-ACK, in addition to GET request for objects, 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 which echoes the cookie that the web server sent. On receiving the COOKIE-ECHO, the web server checks the cookie's validity, and the sends HTTP replies (object).

L_{obj}/L_{mtu} is the number of segments in the object; in Figure 1, $L_{obj}/L_{mtu}=14$. We consider the number of segments that are in each of the windows. The first window contains two segments, the second window contains four segments, and the third window contains eight segments. More generally, the ρth window contains 2^ρ 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:

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

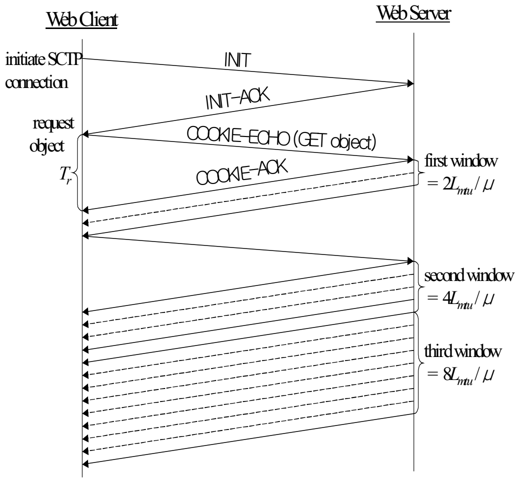

Fig. 1 Timing diagram of HTTP over SCTP during slow-start

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 ρth window. The time from the start of transmission of the ρth window until the reception of acknowledgement of the first segment in the windows is $2L_{mtu}/\mu+T_r$. The transmission time of the ρth window is $(L_{mtu}/\mu)2^\rho$. Stall time (ST), defined as the difference between these two quantities, is given by

$$ST = \left[2\frac{L_{mtu}}{\mu} + T_r - 2^\rho \left(\frac{L_{mtu}}{\mu} \right) \right] \tag{2}$$

The server may stall after the transmission of each of the first α-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,

$$LT_{SCTP} = 2T_r + \frac{L_{obj}}{\mu} + \sum_{\rho=1}^{\alpha} \left[2\frac{L_{mtu}}{\mu} + T_r - 2^\rho \left(\frac{L_{mtu}}{\mu} \right) \right] \tag{3}$$

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

$$\begin{aligned}\beta &= \max\{\rho : 2\frac{L_{mtu}}{\mu} + T_r - \frac{L_{mtu}}{\mu}2^{\rho} \geq 0\} \\ &= \max\{\rho : 2^{\rho} \leq (2 + \frac{T_r}{L_{mtu}/\mu})\} \\ &= \max\{\rho : \rho \leq \log_2(2 + \frac{T_r}{L_{mtu}/\mu})\} = \left\lfloor \log_2(2 + \frac{T_r}{L_{mtu}/\mu}) \right\rfloor\end{aligned} \qquad (4)$$

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

$$\begin{aligned}LT_{SCTP} &= 2T_r + \frac{L_{obj}}{\mu} + \sum_{\rho=1}^{\alpha}\left[2\frac{L_{mtu}}{\mu} + T_r - 2^{\rho}\left(\frac{L_{mtu}}{\mu}\right)\right] \\ &= 2T_r + \frac{L_{obj}}{\mu} + \gamma\left[T_r + 2\frac{L_{mtu}}{\mu}\right] - (2^{\gamma+1} - 2)\frac{L_{mtu}}{\mu}\end{aligned} \qquad (5)$$

where

$$\gamma = \min\left\{\left\lfloor \log_2(2 + \frac{T_r}{L_{mtu}/\mu})\right\rfloor,\ \left\lceil \log_2(\frac{L_{obj}}{L_{mtu}})\right\rceil - 2\right\}$$

D. Object Transfer Latency for HTTP over TCP

A timing diagram to transfer an object using HTTP over TCP is illustrated in Figure 2. TCP uses a three-way handshake. The Web client sends a SYN packet to the web server. The web server returns a SYN-ACK to the web client. On reception of SYN-ACK, the web client replies with an ACK packet which contains the HTTP request (GET object). On receiving the ACK packet, the web server sends the requested object.

Following an approach similar to that used for SCTP, we can obtain the following object transfer latency for HTTP over TCP [6]-[7].

$$LT_{TCP} = 2T_r + \frac{L_{obj}}{\mu} + \gamma\left[T_r + \frac{L_{mss}}{\mu}\right] - (2^{\gamma} - 1)\frac{L_{mss}}{\mu} \qquad (6)$$

where

$$\gamma = \min\left\{\left\lfloor \log_2(1 + \frac{T_r}{L_{mss}/\mu}) + 1\right\rfloor,\ \left\lceil \log_2(\frac{L_{obj}}{L_{mss}}) + 1\right\rceil - 1\right\}$$

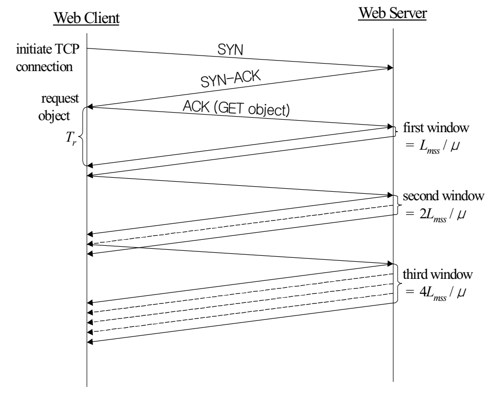

Fig. 2 Timing diagram of HTTP over TCP during slow-start

III. Numerical Results

Table 1 shows the object transfer latencies of TCP (LT_{TCP}) and SCTP (LT_{SCTP}) for varying link transmission rate (μ) and RTT (T_r) during the slow-start phase using Eqns. (5) and (6). For a fair comparison, we set the same value for MTU (L_{mtu}) and MSS (L_{mss}) as 536 B in Table 1. The object size (L_{obj}) is 10 KB.

We can see that savings of SCTP over TCP are increased as μ and T_r are increased. This implies that SCTP can reduce the transfer time in the environment with the long round trip time or high-speed link more than TCP. From the numerical experiments, the transfer latency over SCTP is about 11% less than over TCP on the average.

TABLE I: Comparison of transfer latency (sec) between TCP and SCTP during slow start

		RTT (T_r) = 0.1 (sec)		RTT (T_r) = 0.5 (sec)	
PARAMETER \ PROTOCOL		LT_{TCP}	LT_{SCTP}	LT_{TCP}	LT_{SCTP}
μ (bps)	28 K	3.16	3.16	4.74	4.55
	56 K	1.75	1.73	3.62	3.32
	100 K	1.16	1.11	3.33	2.96
	1 M	0.63	0.55	3.03	2.55
	10 M	0.60	0.50	3.00	2.50
		RTT (T_r) = 1.0 (sec)		RTT (T_r) = 2.0 (sec)	
PARAMETER \ PROTOCOL		LT_{TCP}	LT_{SCTP}	LT_{TCP}	LT_{SCTP}
μ (bps)	28 K	7.24	6.63	13.17	11.63
	56 K	6.59	5.82	12.59	10.82
	100 K	6.33	5.46	12.33	10.46
	1 M	6.03	5.05	12.03	10.05
	10 M	6.00	5.00	12.00	10.00

IV. Conclusions

We derived analytical models to quantify and compare the object transfer latencies using HTTP over SCTP and TCP during the slow-start phase. Our numerical results show that HTTP over SCTP is better than HTTP over TCP by 11% on the average. Future work includes developing an analytical model for the object transfer latency by considering both the slow start and congestion avoidance phases.

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. 7, 2005, pp. 825-845.

[2] R. Stewart, Q. Xie, et al, Stream control transmission protocol, RFC 2960, 2000.

[3] S. Fu and M. Atiquzzaman, "Performance Modeling of SCTP Multihoming", in *Proc.IEEE Globcom*, St. Louis, MO, Nov. 28-Dec. 02, 2005.

[4] M. Allman, S. Floyd, and C. Partridge, Increasing TCP's initial window, RFC 2414, 1998.

[5] W. Stevens, TCP slow start, congestion avoidance, fast retransmit and fast recovery algorithm, RFC 2001, 1997.

[6] J. Heidemann, K. Obraczka and J. Touch, "Modeling the performance of HTTP over several transport protocols", *IEEE/ACM Transactions on Networking*, Vol. 5, no. 5, pp. 616-630, 1997.

[7] N. Cardwell, S. Savage, and T. Anderson, "Modeling TCP latency," in *Proc. IEEE Infocom*, pp. 1742-1751, 2000.