Modeling and Performance Evaluation of Mean Waiting Time for Multiple Web Access in the Narrowband Network

Y. –J. Lee

Department of Technology Education, Korea National University of Education
Cheongju, 363-791, South Korea
lyj@knue.ac.kr

Abstract

This paper presents analytical models to find the mean waiting time for multiple access web service by using HTTP over SCTP. Mean waiting time is important measures of Quality of Service (QoS) in simultaneous users accessing a web server. The proposed mean waiting time model assumes the multiple packet losses and a narrowband network, which does not allow fast retransmission because of the small size window. Our practical experiments show that the differences between the results from the model and those from the experiments are very small below about 4% on average. We also find that the mean waiting time for HTTP over SCTP is less than that for HTTP over TCP. The model can be used for planning and dimensioning of the network bandwidth to satisfy the QoS constraint of end-users.

Keywords: *Mean Waiting Time, Multiple Web Access, HTTP over SCTP.*

1. Introduction

SCTP(stream control transmission protocol) [1,2] was proposed as a transport layer protocol which has multi-streaming capability to transmit several independent streams of chunks (or messages) in parallel. When a packet loss occurs in a stream, it affects the relevant stream only. TCP [3], on the other hand, uses a single stream preserving byte order in the stream by assigning a sequence number to each packet. However, there is no known work on waiting time of HTTP over SCTP using an analytical model in the multiple users' environment.

Response time for single user is affected by data size and transmission time according to transmission rate of link as well as by congestion control mechanism. The congestion control mechanism of SCTP is similar with window-based one of TCP. Their common functions are slow-start, congestion avoidance, timeout, and fast retransmission.

Padhye [4] considered large amount of data transmission on steady state over TCP. Most of TCP connections for HTTP data transmission, however, are short for small amount of data instead of large one in current internet environment. Connection setup or slow-start time dominates the performance of web in this environment. Cardwell [5] extended the above steady state model but he did not consider delay of TCP after time-out. Jiong [6] enhanced the Cardwell's model by considering slow-start time after timeout of retransmission. However, since the above models assumed wideband network, they cannot be applied to the narrowband network environment, which this paper considers. That is because the narrowband network environment does not allow fast retransmission of data due to the very small size of window [7]. Furthermore, the previous studies are limited to single user cases, where the response time is a good measure of the end-to-end delay experienced by a user.

Chang et al. [8] studied the performance of File Transfer Protocol (FTP) over SCTP, and Lu [9] analyzed the performance of Session Initiated Protocol (SIP) over SCTP. Fei Ge [10] presents a simple closed-form formula to estimate the HTTP latency over FAST TCP, taking into account the network parameters such as packet size, link capacity, and propagation delay. Eklund et al. [11] developed a model that predicts the transfer times of SCTP messages during slow start. However, mean waiting time model for HTTP over SCTP in multiple users' environment has not yet been presented.

The focus of this paper is to study the case of multiple users accessing a server, where the waiting and turnaround times depend on the server load. In such a case, the response time may not be a good measure of end-to-end delay.

Our model can be used by network engineers to dimension a network in terms of bandwidth requirement and to develop scheme distributing the load among a number of web servers, in order to improve the waiting delay perceived by end users. We aim to find the theoretical upper bound of the actual waiting and turnaround times of users in a real environment when they download web objects using HTTP over SCTP in the narrowband network, which does not allow fast retransmission.

By developing an analytical model to compute the mean waiting and turnaround time of an end user when multiple users simultaneously access the web server, we achieved our objectives. Previous works [12,13,14] only considered the response time of an object for single user, however, we first consider the response time for single user and then find waiting delay for multiple users. Therefore, we can compute more realistic end-to-end delay experienced by a user in the real environment.

The estimated mean waiting time in this paper can be used as a benchmark to pre-estimate waiting time by considering size of objects, bandwidth, and round trip time. In order to validate the proposed mean waiting time model, we experimented in a simple test-bed and compared the results with estimated value. Additionally, we compared the values with the mean waiting time of HTTP over TCP. Earlier version of this paper was presented in [15].

We describe the estimation model and algorithm of mean response and waiting time for HTTP over SCTP, respectively in Sections 2 and 3. We discuss performance evaluation and analysis in Section 4. We conclude this paper in section 5.

2. Mean response time for single user

We first describe the mean response time model, when single user retrieves a web object in the narrowband network [14].

The congestion control mechanism of SCTP in the narrowband network in Fig. 1. In Fig.1, *th*(1), *th*(2), and *th*(3) are the slow start thresholds and initially *th*(1)=∞. y coordinate is the congestion window(*cwnd*) and its initial value is 2×*mtu*. Here, *mtu* represents the maximum transfer unit of the link. Thus SCTP executes the slow-start period by increasing *cwnd* exponentially such as 2, 4, 8, ... and detects the packet loss when timeout occurs at ①. SCTP responds to this as following.

$$th(2) = \max(\frac{cwnd}{2},\ 2 \times mtu)$$
$$cwnd = 1 \times mtu \qquad (1)$$

The threshold of next stage is reduced to half size of the window in which packet loss occurred and slow-start period is repeated with congestion windows exponentially increased from 1 to 2, 4, 8, etc. When the congestion window exceeds threshold *th*(2), congestion avoidance period is started. Since this period needs an acknowledgement every packet, it is called linearly increasing period. If a packet loss occurs as Fig. 1, ② in this period, there are two choices according to timeout. First of all, using (1) new threshold (*th*(3)) is obtained. If three duplicate acknowledgements are obtained before timeout, then fast retransmission (Fig. 1, ③) is started. Otherwise slow-start (Fig. 1, ④) is executed. This paper assumes the narrowband network which is not able to receive three duplicate acknowledgements during timeout. Thus the slow-start is executed.

To simplify the model, it is assumed that sizes of web objects are identical and received packets are transmitted in an upper layer in terms of window unit. We let the size of an object to transfer be θ bits and maximum transfer unit *mtu* bits, then the number of packets to transfer for an object is $n = \lceil \theta/mtu \rceil$.

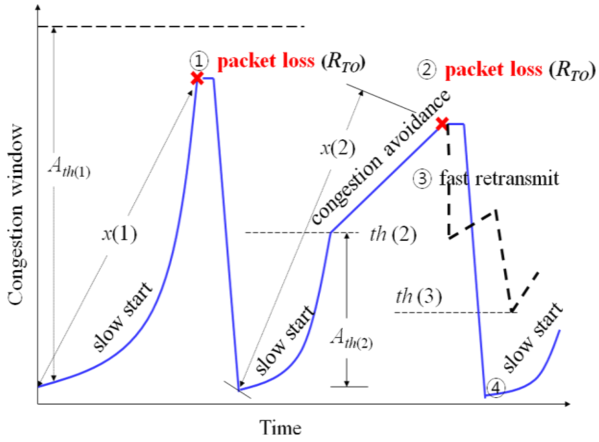

Fig. 1 Congestion control of SCTP in the narrowband network

When the probability of a packet loss is *p*, the expected number of total packet loss is $\alpha = \lceil np \rceil$ in terms of binomial distribution. Any packet loss occurs during either slow-start phase or congestion avoidance phase.

We can identify the packet loss phase by comparing, for k^{th} packet loss, the possible number of packets ($A_{th(k)}$) to transmit until the threshold (*th(k)*, *k*=1,2,..,*a*) at which congestion avoidance starts, with the expected number of packets (*x(k)*: *k*=1,2,..,*a*) transmitted before the packet loss. At this time, *x*(*k*) is calculated as a function of remained packets *N*(*k*) and packet loss rate *p*.

We can determine that an arbitrary k^{th} packet loss occurs either during slow-start phase or congestion avoidance phase, when either $x(k)<A_{th(k)}$ or $x(k)\geq A_{th(k)}$, respectively. In Fig. 1, the total number of packets transmitted is $x(1)$ until the first loss ① and the possible number of packets to transmit is $A_{th(1)}$ until $th(1)$. And since $x(1)<A_{th(1)}$, it is considered that the packet loss occurs during slow-start phase. Similarly, since the number of packets sent before the loss ② is $x(2)>A_{th(2)}$, it is determined that the packet loss occurs during congestion avoidance. Mean response time for HTTP over SCTP is given by Eq. (2).

$$E(T_{sctp}) = \sum_{k=1}^{\alpha}\left[\beta E(T^{k}_{slow}) + (1-\beta)E(T^{k}_{cong})\right] + R \qquad (2)$$

The first packet loss (k=1) of SCTP in (2) occurs always during slow-start phase as shown in Fig. 1, so, $E(T^{1}_{slow})$ needs to be added. Packet losses after second one occur during either slow-start phase or congestion avoidance phase. $E(T^{k}_{slow})$ and $E(T^{k}_{cong})$ represent mean response time, when the k^{th} packet loss (k=2,3,...a) occurs during slow-start phase and congestion avoidance phase, respectively. Because an arbitrary packet loss cannot occurs simultaneously during slow-start phase and congestion avoidance phase, ß is either 0 or 1 for the given k^{th} packet loss. That is, if k^{th} packet loss occurs during slow-start phase and ß=1, then $E(T_{sctp})$ is accumulated by adding $E(T^{k}_{slow})$. Similarly, if k^{th} packet loss occurs during congestion avoidance phase and ß=0, then $E(T_{sctp})$ is accumulated by adding $E(T^{k}_{cong})$. Therefore the total mean response time of an object needs to add either $E(T^{k}_{slow})$ or $E(T^{k}_{cong})$ (k=1,2,...,a) as the expected value of lost packet number (a).

We can compute R, which is the time to transfer the remained data, $N(a+1)$ after the last packet loss occurred, without considering additional packet losses since the expected value of packet losses is already equal to a. That is, if $N(a+1)$ is less than the possible amount of data to transfer until the last threshold $th(a+1)$, the transmission is completed during slow-start phase. Therefore R is sum of slow-start time ($ST(N(a+1))$) and transmission time ($N(a+1)\times mtu/\mu$) until then. μ represents the bandwidth of the link. Otherwise the transmission is completed during congestion avoidance phase. Thus R is sum of slow-start time ($ST(A_{th(a+1)})$) and transmission time ($N(a+1)\times mtu/\mu$) until the threshold adding the extra time ($(N(a+1)–A_{th}(a+1))\times rtt$) in congestion avoidance phase.

3. Mean waiting time for multiple users

In the previous section, we found the mean response time of HTTP over SCTP (E(T_{sctp})), which is total time for a user to connect to a web server and download an object. We can define the mean waiting time as the performance measure when multiple users access the web server simultaneously.

We assume the asynchronous TDM (time division multiplexing) based on packet for web service. A web object consists of n packets, thus, packet response time (τ) is equal to E(T_{sctp})/n when every τ is the same. Also, n is given by $\lceil \theta/mtu \rceil$. Now, if we assume that four clients (a, b, c, d; m=4)request the same file, each user's expected response time (E(T_{sctp})) will be the same. For example, we consider the case where n=3 with the asynchronous TDM. When a client requests an object from the server, three packets are included in the object. E(T_{sctp}) means total response time that each client expects. Fig. 2 depicts this situation.

The transmission sequence at the server is a_1 (the first packet of a), b_1 (the first packet of b) and so on. We are interested in the mean waiting time of end-users. The waiting times for user a are (b_1+c_1+d_1) + (b_2+c_2+d_2) and the finish time of a_3 respectively. Thus, the mean waiting time is obtained by dividing the total waiting time by the number of users. Although each client expects his finish time as the theoretical response time (E(T_{sctp})), the actual finishing time of client will be affected by the number of users who are accessing the server simultaneously. Generally, the packet response times (t) of users are different. Thus, we develop analytical models for the mean waiting time when the packet responses times are different.

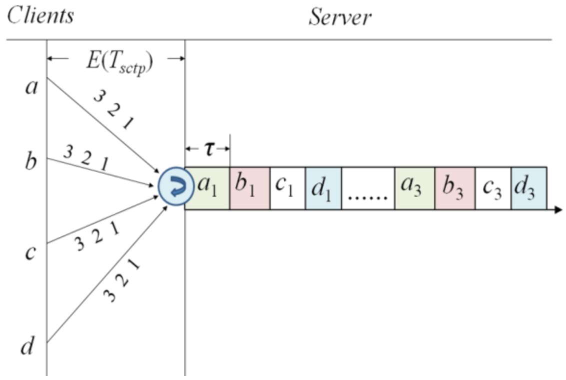

Fig. 2 Four clients and three packets per object

Now, we develop analytical models for the mean waiting and turnaround times for two cases depending on whether the packet response times are same or not.

When the web servers are connected to the external users through only one link, the total waiting time, the mean waiting time(W_{sctp}^{same}), total turnaround time, and mean turnaround time(T_{sctp}^{same}) are given by the following equations:

$$total\ waiting\ time = \sum_{i=1}^{m}(m-i)\tau + m(n-1)(m-1)\tau \qquad (3)$$

$$W_{sctp}^{same} = \frac{\sum_{i=1}^{m}(m-i)\tau + m(n-1)(m-1)\tau}{m} \qquad (4)$$

$$total\ turnaround\ time = m\tau[m(n-1) + \frac{m+1}{2}] \qquad (5)$$

$$T_{sctp}^{same} = \frac{m}{m}[m(n\text{-}1) + \frac{m+1}{2}]\tau = [\frac{2mn\text{-}m+1}{2}]\tau \qquad (6)$$

When the web servers are connected to the external users through several links of different bandwidths, the mean waiting and turnaround time are given by (7) and (8) respectively. First, we consider the mean waiting time. To find the waiting time of i^{th} user, we divide the total time into two intervals: the first interval represents the time when all the packets except the last packet of each user has been received; the second interval represents the time when the last packet of each user has been received. Total waiting time of i^{th} user until the first interval is (the number of packets–1)×[(the number of users for group including i^{th} user–1)×τ_i+(total packet response time excluding i^{th} group)]. The waiting time of i^{th} user is the sum of response times of other users prior to him. By generalizing and adding this all, we obtain the following equation for the mean waiting time. Both m_0 and τ_0 are zeros in the equation.

$$W_{sctp}^{diff} = \frac{(n-1)\sum_{i=1}^{P} m_i[(m_i - 1)\tau_i + \sum_{i=1, j\neq i}^{P} m_j\tau_j]] + \sum_{i=1}^{P}[\sum_{j=1}^{i} m_i(m_{j-1}\tau_{j-1}) + \sum_{j=1}^{m_i}(j-1)\tau_i]}{m} \qquad (7)$$

Now, we consider the mean turnaround time. If we use the same procedure as the waiting time, total turnaround time of i^{th} user until the second interval is (the number of packets–1)×[the number of users (m_i)×the sum of packet response time (τ_i)]. The turnaround time of any user in the second interval is the sum of response times of other users prior to him and his own packet response time. Thus, by generalizing and adding this all, we obtain the following equation. Both m_0 and τ_0 are zeros in the equation.

$$T_{sctp}^{diff} = \frac{m(n-1)\sum_{i=1}^{P} m_i\tau_i + \sum_{i=1}^{P}[\sum_{j=1}^{i} m_i(m_{j-1}\tau_{j-1}) + \sum_{j=1}^{m_i} j\tau_i]}{m} \qquad (8)$$

4. Performance evaluation

We can construct an algorithm for the whole procedure as in Algorithm 1 (Fig. 3) by using the model developed in section 2 and 3. Given that the number of packets for an object is *n*, the complexity of the algorithm is *O*(*n*).

We consider simulating web server for TCP and SCTP, and an environment to emulate HTTP. That is, in order to fairly compare TCP and SCTP, we do not use HTTP based on TCP. Because the web server based on SCTP is incomplete now, and even though it is implemented its performance is not tuned comparing with TCP. Since the basic objective function of the model proposed in this paper is mean waiting time, it is assumed in the simulation environment that web objects are simply requested and transmitted. It, however, has no problem to validate the analytical model.

Algorithm 1. mean waiting and turnaround time for multiple users

```
Begin
Compute the total number of packets in object
  ($n = \lceil \theta/mtu \rceil$)
Compute the expected number of packet loss
  ($\alpha = \lceil np \rceil$)
Set $N(1) = n$ and $th(1) = \infty$
Set $E(T_{sctp}) = 0$
for all $k$ such that $k$=1,2…, α do
    Find $E(T_{slow}^{k})$ and $E(T_{cong}^{k})$
end for
Find the mean response time, $E(T_{sctp}) = E(T_{sctp}) + R$
Find the packet response time, $\tau = E(T_{sctp}) / n$
If $\tau$ is same for all bandwidth type $i$,
    Find mean waiting and turnaround time using (4) and (6) respectively.
else
    Find mean waiting and turnaround time using (7) and (8), respectively.
endif
End
```

Fig. 3 Mean waiting and turnaround time for multiple access users

Desktop computers with Redhat Linux 9 kernel 2.6.6 are used as client-server to send data. In order to simulate real network, we use a laptop computer with NIST emulator

[16] between a client and a server, and adjust various network conditions such as packet loss ratio, bandwidth, and RTT. Two Linux C server programs are written to imitate HTTP over SCTP and HTTP over TCP for the experiment. And two client programs are written to simulate pipelining (TCP/SCTP) and multi-streaming (SCTP).

Table 1~3 show the experimental results and mean waiting times of the model proposed in this paper. W_{sctp} and W_{tcp} represent mean waiting times, for HTTP over SCTP of proposed model and HTTP over TCP, respectively. Except the number of initial windows, HTTP over TCP model is basically same as HTTP over SCTP. That is, except that mean response time for the case of first packet loss occurred in slow-start phase is computed differently, the procedures are almost same. T_{sctp} and T_{tcp} represent experimental values, for HTTP over SCTP and HTTP over TCP, respectively. Mean object size (θ) is 13.5 KB and maximum transmission unit (*mtu*) is 536 B. A HTML file contains five web objects.

First, we fixed *rtt* and link transmission rate (μ), as 256 ms and as 40 Kbps, respectively. And then, we changed packet loss ratio (p) as shown in Table 1. When looking at the values, according as p decreases, the number of retransmission is close to 0. And slow-start time and retransmission time are close to 0 too. The reason is that slow-start time to retransmit the lost packet is needed only for the case of packet loss.

Table 1: Mean waiting time comparison for varying packet loss ratio

packet loss ratio (p)	***m***	W_{sctp}	T_{sctp}	W_{tcp}	T_{tcp}
0.4 %	5	3.18	2.84	3.26	2.85
	10	3.58	3.20	3.27	3.21
	20	3.77	3.37	3.88	3.39
	30	3.84	3.43	3.94	3.45
1 %	5	3.19	2.85	3.18	2.90
	10	3.59	3.20	3.58	3.26
	20	3.78	3.38	3.78	3.45
	30	3.85	3.44	3.85	3.51
2 %	5	3.26	2.88	3.34	2.90
	10	3.66	3.24	3.75	3.26
	20	3.87	3.42	3.96	3.45
	30	3.93	3.48	4.03	3.51

Second, we fixed p = 1 % and *rtt* = 0.256 seconds, and when increasing link transmission rate (μ), mean waiting times of HTTP over TCP and HTTP over SCTP became almost same in Table 2. The reason is that, when μ grows, mtu/μ reduces retransmission time remarkably.

Table 2: Mean waiting time comparison for varying link rate

link rate (μ)	***m***	W_{sctp}	T_{sctp}	W_{tcp}	T_{tcp}
4 Kbps	5	2.84	3.27	2.88	3.35
	10	3.20	3.68	3.24	3.76
	20	3.38	3.88	3.42	3.97
	30	3.44	3.95	3.48	4.04
400 Kbps	5	0.69	0.72	0.77	0.80
	10	0.78	0.81	0.87	0.90
	20	0.82	0.85	0.92	0.95
	30	0.84	0.87	0.93	0.97
3000 Kbps	5	0.51	0.55	0.58	0.63
	10	0.58	0.62	0.65	0.71
	20	0.61	0.65	0.69	0.74
	30	0.62	0.66	0.70	0.76

Third, after we fixed p = 1 % and μ = 40 Kbps, we changed *rtt*. Table 3 shows that mean waiting time grows rapidly as *rtt* increases. It shows that mean waiting time of HTTP over SCTP is most sensitive to *rtt*.

Table 3: Mean waiting time comparison for varying RTT

round trip time (rtt)	***m***	W_{sctp}	T_{sctp}	W_{tcp}	T_{tcp}
55 ms	5	2.31	2.13	2.32	2.15
	10	2.60	2.40	2.61	2.42
	20	2.74	2.53	2.75	2.56
	30	2.79	2.58	2.80	2.60
80 ms	5	2.32	2.27	2.33	2.28
	10	2.61	2.55	2.62	2.57
	20	2.75	2.70	2.77	2.71
	30	2.80	2.74	2.82	2.76
256 ms	5	2.84	3.27	2.86	3.35
	10	3.20	3.68	3.22	3.77
	20	3.38	3.88	3.40	3.98
	30	3.44	3.95	3.46	4.05

Fig. 4 depicts mean waiting times for each p, μ, *rtt* from Table 1 ~ Table 3. In the figure, MODEL_SCTP and EXPE_SCTP represent W_{sctp} and T_{sctp}, respectively. MODEL_TCP and EXPE_TCP also represent W_{tcp} and T_{tcp}, respectively. Fig. 4 shows that both model for HTTP over SCTP and HTTP over TCP overestimates mean waiting times for p and μ, respectively, but, model underestimates them for *rtt*.

Now, we define the mean difference ratio between models and experiments by Eq. (9).

$$DIFF_{mean} = \sum_{i=1}^{n} \left[\frac{W_{sctp} - T_{sctp}}{W_{sctp}} + \frac{W_{tcp} - T_{tcp}}{W_{tcp}} \right] / n \times 100 \qquad (9)$$

The computed $DIFF_{mean}$ is 4.17 % from Table 1 ~ Table 3, so our model is well fitted to the real environment. This small error is due to the inaccuracy of the NIST emulator. Additionally, in Table 1 ~ Table 3, we find that the mean waiting time of HTTP over SCTP is less than HTTP over TCP on both the model and experiment.

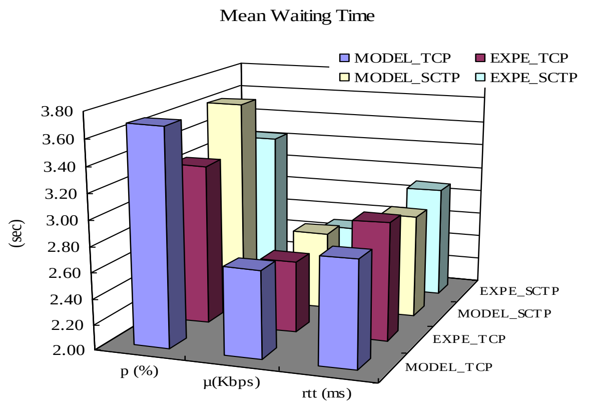

Fig. 4 Mean waiting times for *p*, *μ*, *rtt*

5. Conclusions

Mean waiting time for multiple users is one of essential parameters to evaluate web performance. In this paper, we present an analytical model to estimate mean waiting time of web service using HTTP over SCTP in the narrowband network when multiple users access web server. We first describe the mean response time model for single user, which is one of QoS offered to web users. We then extend the mean response time model to the mean waiting and turnaround time models for multiple users. Simple test-bed simulation results show that the mean difference ratio, between the analytical model and experiment, is very small. Future works include more sophisticated models which can be applied to both wired and wireless environment.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012R1A1A4A01003651)

References

[1] R. Stewart, "Stream control transmission protocol (SCTP), RFC 4960, http://www.ietf.org/rfc/rfc4960.txt (2007).

[2] L. Budzisz, J. Garcia, A. Brunstrom, and R. Ferrus, "A Taxonomy and Survey of SCTP research", ACM Computing Surveys, vol. 44, No. 4, 2012, pp. 1-36.

[3] V. Paxon, M. Allman, and W. Stevens, "TCP's congestion control", RFC 2581, 1999.

[4] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Reno performance: A simple model and its empirical validation", ACM Transactions on Networking, Vol. 8, No. 2, 2000, pp. 133-145.

[5] N. Cardwell, S. Savage, and Y. Anderson, "Modeling TCP latency", Proceeding of the 2000 IEEE Infocom Conference, 2000, pp. 1742-1751.

[6] Z. Jiong, Z. Shu-Jing, and Qi-Gang, "An adapted full model for TCP latency", Proceedings of. the 2002 IEEE TENCON Conference, 2002, pp. 801-804.

[7] D. Oliveria and R. Braun, "A dynamic adaptive acknowledgement strategy for TCP over multihop wireless networks", Proceedings of the IEEE INFOCOM Conference, 2005, pp. 1863-1874.

[8] Lin-Huang Chang, Ming-Yi Liao and De-Yu Wang, "Analysis of FTP over SCTP in Congested Network", 2007 International Conference on Advanced Information Technologies (AIT), 2007, pp. 82-89.

[9] Chia-Wen Lu and Quincy Wur, "Performance study on SNMP and SIP over SCTP in wireless sensor networks", 14th International conference on advanced communication technology (ICACT), 2012, pp. 844-847.

[10] Fei Ge, Liansheng Tan, Jinsheng Sun, and Moshe Zukerman, "Latency of fast TCP for HTTP transactions", IEEE Communications Letters, Vol. 15, No. 11, 2011, pp. 1259-1261.

[11] J. Eklund, K. Grinnemo, A. Brunstorm, G. Cheimnidis, and Y. Ismailov, "Impact of Slow Start on SCTP Handover Performance", Proceedings of the 20th international conference on computer communications and networks, 2011, pp. 1-7.

[12] Y. Lee, M. Atiquzzaman, and S. Sivagurunathan, "Mean response time estimation for HTTP over SCTP in wireless environment", Proceedings of the 2006 IEEE International Conference on Communications, 2006, pp. 164-169.

[13] Y. Lee and and M. Atiquzzaman, "Mean waiting delay for web object transfer in wireless environment", Proceedings of the 2009 IEEE International Conference on Communications, 2009, pp. 1-5.

[14] Y. Lee, "Mean response delay estimation for HTTP over SCTP in wireless Internet", Journal of the Korea Contents Association, Vol. 8, No. 6, 2008, pp. 43-53.

[15] Y. Lee, "Mean waiting time of an end-user in the multiple web access environment", The Sixth International Conference on Communication Theory, Reliability, and Quality of Service (CTRQ-2013), 2013, pp.1-4.

[16] M. Carson and D. Santay, "NIST Net – A Linux-based Network Emulation Tool", ACM SIGCOMM Computer Communication Review, Vol. 33, No. 3, 2003, pp. 111-126.