Mean Waiting Delay for Web Object Transfer in Wireless SCTP Environment

Yong-Jin Lee[1] and M.Atiquzzaman[2]

[1]Department of Technology Education, Korea National University of Education
[2]School of Computer Science, University of Oklahoma

***Abstract*-Most current web application use HTTP (Hyper Text Transfer Protocol) and TCP (Transmission Control Protocol) to retrieve objects from the Internet. SCTP (Stream Control Transmission Protocol) is recently proposed transport protocol with congestion control mechanism similar to that of TCP. Waiting delay is an important performance criterion for when transferring web object over the Internet. In this paper, we present an analytical model of mean waiting delay for object transfers over the Internet in a wireless using the SCTP as the transport protocol and compare with the mean waiting in using TCP. Validation of the model using experimental results show that the mean waiting delay for HTTP over SCTP is less than that for HTTP over TCP. This is caused by the small slow-start time of SCTP.**

I. INTRODUCTION

Most web applications use HTTP (hyper text transfer protocol) as the transfer protocol to retrieve objects in the Internet. HTTP is a connection-oriented protocol and uses TCP (transmission control protocol) as the transport layer protocol. TCP provides a single stream of data and strict ordered delivery. Consequently, when a packet is lost in the network, all subsequent packets arriving at the receiver must wait until the lost packet has been retransmitted from the sender and received at the receiver. This phenomenon, called Head of Line (HOL) blocking, results in *waiting delay* [1] that affects the performance of web transfers.

SCTP (Stream Control Transmission Protocol) [2,3,4] has been proposed by IETF as a new transport layer protocol. The design of SCTP absorbed many 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. Two main new features are multi-streaming and multi-homing. SCTP's multi-streaming can alleviate the head-of-line blocking when an HTML page contains multiple objects. However, it cannot avoid the waiting delay between packets when single objects are transferred; the end point must hold the received packet from delivery to the upper layer protocol (HTTP) until they are reordered. This waiting delay is affected by slow-start and retransmission policy.

Several models [5,6,7] that estimate the mean transfer times of TCP application have been presented in the literature; however, they assumed single packet loss. Furthermore, since the waiting delay was not considered in the transfer time, we cannot extract the waiting delay from the previous models. The *objective* of this paper is to develop an analytical model to estimate the waiting delay for multiple packet losses when SCTP is used in a wireless web environment and compare with the mean waiting delay of TCP.

The rest of the paper is organized as follows. Section 2 provides background on the web object transfer in TCP and SCTP environment. Section 3 describes the mean waiting delay time in web object transfer. Section 4 presents mean waiting delay comparison of HTTP over TCP and HTTP over SCTP. Finally, concluding remarks are given in Section 5.

II. WEB OBJECT TRANSFER IN TCP AND SCTP ENVIRONMENT

Consider the simple case of a web browser displaying a web page with embedded objects. Using HTTP/1.1 that supports persistent and pipelined connections, the browser opens a new transport connection to the server, and sends an HTTP GET request with the desired URI (Uniform Resource Identifier). The request may consist of a specific command, a message containing request parameters, information about the client and may contain URI's of embedded objects. The server returns an HTTP response with the page contents. The response includes status information, a success/error code, and a message containing information about the server and information about the response itself. As responses arrive from the server, the browser displays the web page with its embedded objects. In general, objects embedded within a web page are independent of each other. That is, requesting and displaying each object in the page does not depend on the reception of other embedded objects. This leads to HOL blocking problem that causes performance degradation.

To alleviate HOL blocking, web browsers usually open multiple TCP connections to the same web server. Using multiple TCP connections for transferring a single application's data introduces many negative consequences for both the application and the network. It may cause the increased load on web server. Under high loads, some web servers may choose to drop incoming TCP connection requests due to lack of available memory resources. The other drawback is increased connection establishment latency. Each TCP connection goes through a three-way handshake for connection establishment before data transfer is possible. This handshake wastes one round trip for every connection opened to the same web server. Increasing the number of connections increases the chances of losses during connection establishment, thereby increasing the overall average transfer time.

The SCTP distinguishes different streams of messages within one SCTP association. This enables a delivery scheme where only the sequence of messages needs to be maintained per stream (partial in-sequence delivery). The message loss in a particular stream will only hinder delivery of that stream. Therefore, other streams within an association are not affected. By using unordered messages and multi-streaming, SCTP eliminates the head-of-the-line blocking problem. When SCTP is used for web page retrieval; it allows multiple streams in an association and independent streaming delivery of web objects. This avoids the HOL blocking experienced when TCP is deployed for web object retrieval.

In contrast to the three-way handshake that occurs in TCP, SCTP uses a four-way handshake and verification tag to initiate an association. The initialization of an association is completed on both sides after the exchange of four messages (INIT chunk, INIT-ACK chunk, COOK-ECHO chunk and COOKIE-ACK chunk) [6]. The passive side does not allocate resources until the third of these messages has arrived and been validated. When SCTP is deployed for HTTP application this four-way handshake defends against denial-of-service attempts and SYN-type attacks experienced by TCP connection.

The SCTP operates like TCP using window-based flow control with the additional features necessary to transport signaling information. Detection of loss and duplication of data chunks is enabled by numbering all data chunks in the sender with the so-called TSN (Transport Sequence Number). The acknowledgements sent from the receiver to the sender are based on these sequence numbers. Retransmissions are timer-controlled. The timer duration is derived from continuous measurements of the round trip delay. Whenever such a retransmission timer expires, all non-acknowledged data chunks are retransmitted and the timer is started again doubling its initial duration like the TCP. Whenever the sender receives four consecutive SACKS reporting the same data chunk missing, this data chunk is immediately retransmitted (fast retransmit). Data that is discarded, reordered, duplicated or corrupted can be detected and fast re-transmitting and fast recovery for damaged data can be issued. This congestion control mechanism improves the performance of SCTP in wireless environment. The above stated features show that SCTP can be best suited for HTTP application with improved performance.

III. MEAN WAITING DELAY TIME IN WEB OBJECT TRANSFER

Fig. 1 depicts the waiting delay problem when N packets form an object. In case $(x_1+1)^{th}$ packet is lost, packets $1,2,..,x_1$ arrived at client have to wait for the retransmission of packets sent after $(x_1+1)^{th}$ packet in the window to display object.

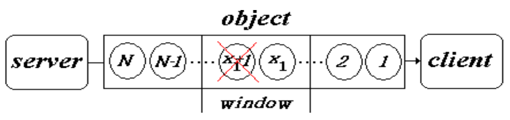

Figure 1. Waiting delay problem in the object transfer

Table 1 represents the notations used in our analytical models for HTTP over SCTP and TCP.

TABLE I. Notations

Symbol	Meaning
θ	size of an object (bits)
λ	maximum transfer unit (bits)
μ	data rate between the server and the client (bps)
N	the total number of packets in an object
p	packet loss probability
k	expected number of packet loss in an object
x_i	the number of packets sent until i^{th} packet loss (i=1,2,..,k)
$\Delta_{TCP}(i)$	the number of packets resent due to i^{th} packet loss (i=1,2,..,k) in TCP
$\Delta_{SCTP}(i)$	the number of packets resent due to i^{th} packet loss (i=1,2,..,k) in SCTP
$R_{TCP}^{\Lambda_{TCP}(i)}$	retransmission time of $\Delta_{TCP}(i)$ packets in TCP
$R_{SCTP}^{\Lambda_{TCP}(i)}$	retransmission time of $\Delta_{SCTP}(i)$ packets in SCTP
$S_{TCP}^{\Lambda_{TCP}(i)}$	slow-start times for transferring $\Delta_{TCP}(i)$ packets in TCP
$S_{SCTP}^{\Lambda_{TCP}(i)}$	slow-start times for transferring $\Delta_{SCTP}(i)$ packets in SCTP
α_{TCP}	the number of times the server would stall in TCP
α_{SCTP}	the number of times the server would stall in SCTP
β_{TCP}	the number of windows that cover the object without packet loss in TCP
β_{SCTP}	the number of windows that cover the object without packet loss in SCTP
γ_{TCP}	window number when x_i packet sent in TCP
γ_{SCTP}	window number when x_i packet sent in SCTP
y_{TCP}	TCP's window size that covers the expected number of packets (x_i) when loss occurs
y_{SCTP}	SCTP's window size that covers the expected number of packets (x_i) when loss occurs
z_{TCP}	the number of packets sent in TCP before the loss in the γ_{TCP} windows
z_{SCTP}	the number of packets sent in SCTP before the loss in the γ_{SCTP} windows
RTT	round trip time
$W_{TCP}(i)$	waiting delay due to i^{th} packet loss in TCP
$W_{SCTP}(i)$	waiting delay due to i^{th} packet loss in SCTP
W_{TCP}^{total}	mean waiting delay for object transfer in TCP
W_{SCTP}^{total}	mean waiting delay for object transfer in SCTP

A. Modeling for waiting delay

The total number of packets (N) in an object is $N = \lceil \theta / \lambda \rceil$ when θ is size of an object (bits) and λ is maximum transfer unit (bits). When p is packet loss probability, by the binomial distribution, the expected number of packet loss is $k = \lceil Np \rceil$. Figure 2 illustrates sending an HTTP request for an object when the first packet is lost in SCTP. x_1 represents the number of packets sent until the first packet loss. That is, $(x_1+1)^{th}$ packet is lost in the γ_{SCTP}^{th} window.

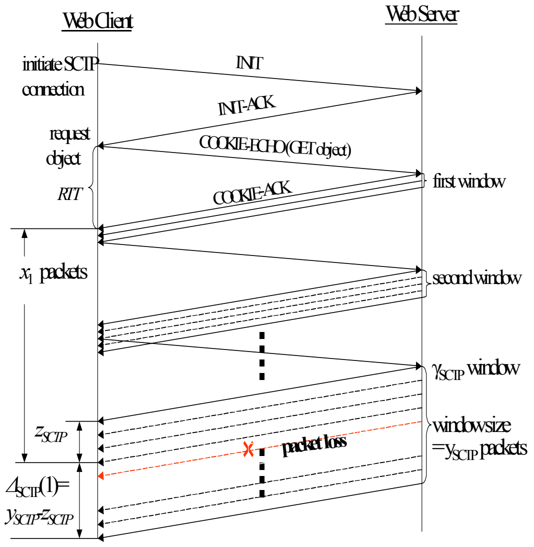

Figure 2. HTTP request for an object in SCTP environment

Let z_{SCTP} be the number of packets sent before the packet loss in the γ_{SCTP}^{th} window, and y_{SCTP} represent the window size of γ_{SCTP}^{th} window. Our model assumes a wireless environment where fast recovery is not possible because of small window. Thus, when the GBN (Go-Back N) is used, the number of resent packets is $y_{SCTP} - z_{SCTP} = \Delta_{SCTP}(1)$.

Congestion mechanism resumes the slow-start phase after the packet loss. Therefore, delay due to the first packet loss includes the retransmission time of $\Delta_{SCTP}(1)$ packets and slow start time incurred by retransmission, giving a waiting delay of $W_{SCTP}(i) = R_{SCTP}^{\Lambda_{TCP}(i)} + S_{SCTP}^{\Lambda_{TCP}(i)}$ $(i=1,2,..,k)$.

B. Slow-start time

To derive waiting delay $W_{TCP}(1)$ for TCP and $W_{SCTP}(1)$ for SCTP corresponding to the first packet loss, we consider the slow-start time for TCP and SCTP, respectively. TCP starts the congestion window from one MSS (Maximum Segment Size) [8]. Of course, according to the current TCP congestion control specification, TCP implementations can use an initial congestion window up to four segments [9]. 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 [8] for TCP and RFC-2960 [7] for SCTP.

We obtain slow-start time (S_{TCP}^{N}) for TCP when N packets for an object are transferred by modifying the equation given in [10]

$$S_{TCP}^{N} = \rho \cdot (RTT + \frac{\lambda}{\mu}) - (2^{\rho} - 1)\frac{\lambda}{\mu} \tag{1}$$

Here, $\rho = \min[\alpha_{TCP}, \beta_{TCP}-1]$. α_{TCP} and β_{TCP} are given by

$$\alpha_{TCP} = \left\lfloor \log_2(1 + \frac{RTT}{\lambda/\mu}) \right\rfloor + 1$$
$$\beta_{TCP} = \left\lceil \log_2(\frac{\theta}{\lambda}) + 1 \right\rceil \tag{2}$$

Meanwhile, SCTP starts the congestion window from two MTUs (Maximum Transfer Unit). Thus, S_{SCTP}^{N}, α_{TCP}, and β_{TCP} are given by

$$S_{SCTP}^{N} = \rho \cdot (RTT + \frac{2\lambda}{\mu}) - (2^{\rho+1} - 2)\frac{\lambda}{\mu} \tag{3}$$

$$\alpha_{SCTP} = \left\lfloor \log_2(1 + \frac{RTT}{\lambda/\mu}) \right\rfloor$$
$$\beta_{SCTP} = \left\lceil \log_2(\frac{\theta}{\lambda}) - 1 \right\rceil \tag{4}$$

The expected number of packets sent before first loss is given by

$$x_1 = \frac{1-(1-p)^N}{p} + (1-p)^N + 1 \tag{5}$$

In case of packet loss, since we send x_1 packets until the loss, current window number in TCP is given by

$$\begin{aligned} \gamma_{TCP} &= \min\{j : 2^0 + 2^1 + \cdots + 2^{j-1} \ge x_1\} \\ &= \min\{j : 2^j - 1 \ge x_1\} \\ &= \min\{j : \log_2(x_1+1)\} = \lceil \log_2(x_1+1) \rceil \end{aligned} \tag{6}$$

Current window number in SCTP is given by

$$\begin{aligned} \gamma_{SCTP} &= \min\{j : 2^1 + \cdots + 2^j \ge x_1 + 1\} \\ &= \min\{j : 2^{j+1} \ge x_1 + 3\} \\ &= \lceil \log_2(x_1+1) \rceil - 1 = \gamma_{tcp} - 1 \end{aligned} \tag{7}$$

Thus, the TCP's window size (y_{TCP}) that covers the expected number of packets (x_1) when loss occurs is given by

$$y_{TCP} = \begin{cases} 2^{\gamma-1} - (2^{\gamma} - N) + 1, & \text{if } N < 2^{\gamma} \\ 2^{\gamma-1}, & \text{otherwise} \end{cases} \tag{8}$$

In eqn. (8), γ is equal to γ_{TCP}. In the same manner, when $\gamma = \gamma_{SCTP}$, SCTP's window size (y_{SCTP}) is given by

$$y_{SCTP} = \begin{cases} 2^{\gamma} - (2^{\gamma} - N) + 1, & \text{if } N < 2^{\gamma} \\ 2^{\gamma}, & \text{otherwise} \end{cases} \tag{9}$$

The number of packets sent in TCP before the loss in the γth windows (when $\gamma = \gamma_{TCP}$) is

$$\begin{aligned} z_{TCP} &= x_1 - (2^0 + 2^1 + \cdots + 2^{\gamma-2}) \\ &= x_1 - 2^{\gamma-1} \end{aligned} \tag{10}$$

z_{SCTP} for the γth windows (when $\gamma = \gamma_{SCTP}$) is

$$\begin{aligned} z_{SCTP} &= x_1 - (2^1 + \cdots + 2^{\gamma-1}) \\ &= x_1 - 2^{\gamma} \end{aligned} \tag{11}$$

Since z_{TCP} packets in TCP must wait in the γ_{TCP} window and $\Delta_{TCP}(1)$ packets are to be retransmitted in the next slow-start time, the number of packets, including the lost packet to be resent, is $\Delta_{TCP}(1) = y_{TCP} - z_{TCP}$. By the same method, $\Delta_{SCTP}(1) = y_{SCTP} - z_{SCTP}$.

Substituting $\Delta_{TCP}(1)$ and $\Delta_{SCTP}(1)$ into S_{TCP}^{N} in eqn. (1) and S_{SCTP}^{N} in eqn. (3) respectively, we can obtain the slow-start time when additional $\Delta_{TCP}(1)$ and $\Delta_{SCTP}(1)$ packets are retransmitted as eqn. (12) and eqn. (13), respectively.

$$S_{TCP}^{\Delta_{TCP}(1)} = \rho \cdot \left[RTT + \frac{\lambda}{\mu} \right] \text{-} (2^{\rho} - 1) \cdot \frac{\lambda}{\mu} \tag{12}$$

where

$$\rho = \min\left\{ \left\lfloor \log_2\left(1 + \frac{RTT}{\lambda/\mu}\right) \right\rfloor + 1, \lceil \log_2(\Delta_{TCP}(1) + 1) \rceil - 1 \right\}$$

$$S_{SCTP}^{\Delta_{SCTP}(1)} = \rho \cdot \left[RTT + \frac{2\lambda}{\mu} \right] \text{-} (2^{\rho+2} - 2) \cdot \frac{\lambda}{\mu} \tag{13}$$

where

$$\rho = \min\left\{ \left\lfloor \log_2\left(2 + \frac{RTT}{\lambda/\mu}\right) \right\rfloor, \lceil \log_2(\Delta_{SCTP}(1) + 1) \rceil - 2 \right\}$$

C. Waiting delay time

Waiting delay time for the first packet loss in TCP is sum of the transfer time and the slow-start time of $\Delta_{TCP}(1)$ packets. The retransmission time ($R_{TCP}^{\Delta_{TCP}(i)}$) is $\Delta_{TCP}(1)\cdot\lambda/\mu$; thus the waiting delay time for TCP is given by

$$W_{TCP}(1) = R_{TCP}^{\Delta_{TCP}(1)} + S_{TCP}^{\Delta_{TCP}(1)} = \Delta_{TCP}(1) \cdot \frac{\lambda}{\mu} + S_{TCP}^{\Delta_{TCP}(1)} \tag{14}$$

The waiting delay time for the first packet loss in SCTP is given by

$$W_{SCTP}(1) = R_{SCTP}^{\Delta_{SCTP}(1)} + S_{SCTP}^{\Delta_{SCTP}(1)} = \Delta_{SCTP}(1) \cdot \frac{\lambda}{\mu} + S_{SCTP}^{\Delta_{SCTP}(1)} \tag{15}$$

Now, we extend our model to *multiple packet losses*. Let x_1, x_2,.., x_k be the number of packets sent until packet loss. When packet loss occurs, we must retransmit the lost packet and the following packets ($\Delta_{TCP}(i)$ for TCP and $\Delta_{SCTP}(i)$ for SCTP: i=1,2..,k) after first, second,…, kth packet loss, respectively. Consequently, additional retransmission times, ($R_{TCP}^{\Delta_{TCP}(i)}$ for TCP and $R_{SCTP}^{\Delta_{TCP}(i)}$ for SCTP: i =1,2,..,k) and slow-start times, ($S_{TCP}^{\Delta_{TCP}(i)}$ for TCP and $S_{SCTP}^{\Delta_{TCP}(i)}$ for SCTP) are required. $W_{TCP}(i)$ and $W_{SCTP}(i)$ represent waiting delay times for the client to wait for the completion of retransmission of $\Delta_{TCP}(i)$ packets for TCP and $\Delta_{SCTP}(i)$ packets for SCTP (i=1,2..,k), respectively. In general, for TCP $W_{TCP}(i) = R_{TCP}^{\Delta_{TCP}(i)} + S_{TCP}^{\Delta_{TCP}(i)}$ and for SCTP, $W_{SCTP}(i) = R_{SCTP}^{\Delta_{TCP}(i)} + S_{SCTP}^{\Delta_{TCP}(i)}$ (i=1,2,..,k).

In order to find $W_{TCP}(2)$ and $W_{SCTP}(2)$, we set $N = N\text{-}x_1+1$ and repeat the above procedure from eqn. (5) to eqn. (15). In the same manner, we can find $W_{TCP}(3)$, $W_{SCTP}(3)$, $W_{TCP}(4)$, $W_{SCTP}(4)$,…, $W_{TCP}(k)$ and $W_{SCTP}(k)$. When total number of packets included in object (N) is sent, the procedure is terminated. Thus, total amount of waiting delay time for an TCP and SCTP object are

$$W_{TCP}^{total} = \sum_{i=1}^{k} W_{TCP}(i) \tag{16}$$

$$W_{SCTP}^{total} = \sum_{i=1}^{k} W_{SCTP}(i) \tag{17}$$

W_{TCP}^{total} and W_{SCTP}^{total} shows the mean waiting delay time for an web object transfer in wireless TCP and SCTP environment, respectively.

IV. PERFORMANCE EVALUATION

In this section, we evaluate the performance of TCP and SCTP in terms of its mean waiting delay. Figure 3 and 4 show the mean waiting delay (sec), as a function of data rate (μ) and packet loss probability (p) for θ=13.5 KB and λ=535 B when RTT = 256 ms and 1 sec, respectively.

As shown in Fig. 3 and 4, mean waiting delay is decreased with an increase of packet loss probability (p). Given the same packet loss probability, mean waiting delay is decreased with an increase of data rate (μ). Especially, mean waiting delay times of SCTP are less than those of TCP in every case. This is caused by the small slow-start time of SCTP. In many cases, slow-start time of SCTP is close to zero. This implies that the start window size policy of SCTP is better than that of TCP in web environment. Our numerical results show that HTTP over SCTP is better than HTTP over TCP by 11 % on the average during the slow start-phase.

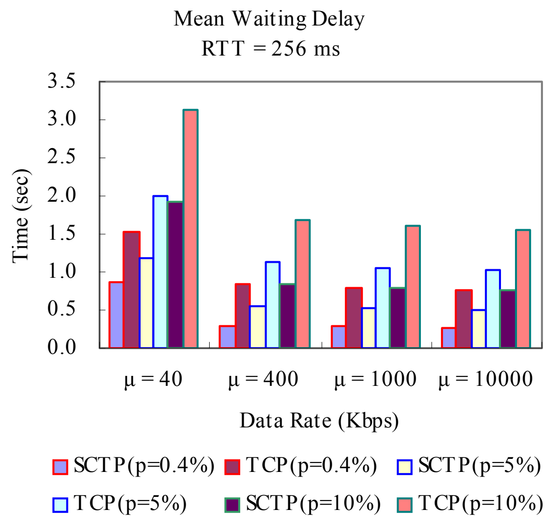

Figure 3. Mean waiting delay for *RTT* = 256 ms

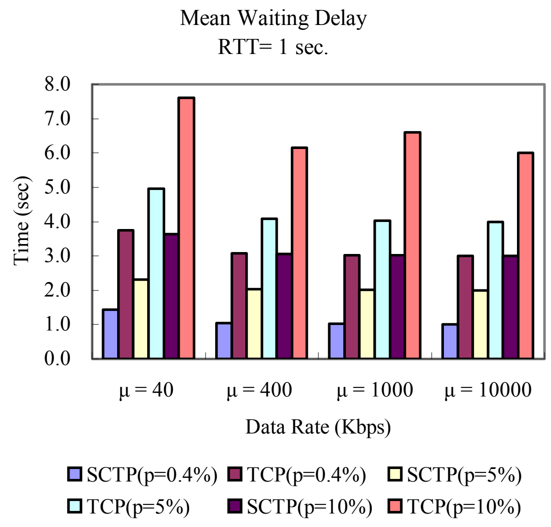

Figure 4. Mean waiting delay for *RTT* = 1 sec.

V. CONCLUSIONS

In this paper, we developed and analytical model to estimate the waiting delay time of HTTP over SCTP and TCP in a wireless environment. This model can be used to quantify the side effect and the avoidance mechanism for object transfer in the wireless web application.

Our numerical results show that the mean waiting delay for HTTP over SCTP is less than that for HTTP over TCP. This is caused by the small slow-start time of SCTP. That is, the slow-start time of HTTP over SCTP is less than that of HTTP over TCP by 11 % on the average. Future work includes the derivation of analytical model for the mean object transfer time in SCTP environment.

REFERENCES

[1] Fu, L. Ma, M. Atiquzzaman and Yong-Jin, Lee, "Signaling Cost and Performance of SIGMA", Wireless Communications and Mobile Computing, vol. 5, pp. 825-845, 2005.
[2] Caro, A., Iyengar, J., Amer, P., Ladha, S., Heinz, G. and Shah, K., "SCTP: A Proposed Standard for Robust Internet Data Transport", IEEE Computer, pp. 20-27, 2003.
[3] R. Stewart, Q. Xie, K. Morneault, and C. Sharp et. al., Stream Control Transmission Protocol, RFC 2960, 2000.
[4] R. Stewart and Q. Xie, Stream Control Transmission Protocol: A Reference Guide, Addison Wesley, 2000.
[5] Cardwell, N., Savage, S. and Anderson, T, "Modeling TCP Latency", IEEE Infocom, pp. 1742-1751, 2000.
[6] Z. Jiong, Z. Shu-Jing, and Qigang, "An Adapted Full Model for TCP Latency", Proceedings of IEEE TENCON, pp. 801-804, 2002.
[7] 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, pp.133-145, 2000.
[8] W. Stevens, TCP Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery Algorithms, RFC 2001, 1997.
[9] M. Allman, S. Floyd, and C. Partridge, Increasing TCP's Initial Window, RFC 2414, 1998.
[10] J. Kurose and K. Ross, Computer Networking, Third Edition, Addison-Wesley, pp. 276-284, 2005.