Mean Waiting and Turnaround Time for Multiple Web Transaction

Yong-Jin Lee
Dept. of Computer Science
Woosong University
Taejon, 300-718, Korea
yjlee@woosong.ac.kr

M. Atiquzzaman
School of Computer Science
University of Oklahoma
Norman, OK 73019, USA
atiq@ou.edu

Keywords: mean waiting time, mean around time, web transaction, HTTP, TCP, T-TCP

Abstract

Hyper Text Transfer Protocol (HTTP) is a transfer protocol used by the World Wide Web distributed hypermedia system to retrieve objects. HTTP is a connection-oriented protocol which uses TCP (Transmission Control Protocol) as the transport layer protocol. HTTP does not interact very well with TCP. In this paper, we study the factors that affect the performance of HTTP over TCP and the transaction times of TCP and T-TCP.

We introduce two new performance criteria, mean waiting and turnaround times, as measures of Quality of Service (QoS) of multiple end users accessing web servers and develop the analytical models providing the theoretical upper bound for such QoS. Results from our proposed model match well with experimental data. The model can be applied to dimensioning of network link bandwidths and distribution of web servers to satisfy a given QoS to end users as measured by the mean waiting and turnaround times.

1. INTRODUCTION

The World Wide Web uses the Hyper Text Transfer Protocol (HTTP) at the application layer to transfer objects between computers in the Internet. HTTP uses on the Transmission Control Protocol (TCP) [2] as the transport layer protocol. HTTP over TCP suffers from several problems such as the connection setup, slow-start, Time_Wait delay and small MSS (Maximum Segment Size) [3,4,5]. A number of solutions to the above problems arising due to the use of TCP as the underlying transport protocol for HTTP have been reported in the literature. For example, T-TCP [6,7] reduces the Time_Wait delay from 240 seconds to 12 seconds and also decreases the transaction time of TCP (which is at least 2 × RTT + file transmission time) to RTT + file transmission time by eliminating the three-way connection setup delay. In the case of persistent HTTP [8] (including HTTP/1.1 [9], HTTP-NG [10], and S-HTTP [5]), transactions following the first one do not require new connections to be setup, and therefore, avoids slow-start on each new transaction.

There have been a number of studies [11-14, 15, 16] on the performance problems of HTTP. They compute the transaction time, taking into account the network bandwidth, window size and slow-start. However, the previous studies are limited to *single user* cases, where the *transaction time* is a good measure of the end to end delay experienced by a user. The *motivation* 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 transaction time may not be a good measure of end to end user delay. The *focus* of this paper is to determine the user level performance of HTTP in the case of multiple users. The results reported in this paper 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 QoS perceived by end users. The *objective* of this study is to find the theoretical upper bound of the actual waiting and turnaround times of users in a real environment. We achieve our objectives by developing an analytical model for such QoS and validating the model against real world experiments on a test bed. In *contrast to previous work*, which only considered the transaction time, without taking into account the load on the server or multiple users, the results of this paper will allow us to compute the end-to-end QoS experienced by a user in the case of multiple users. *The authors are not aware of any previous paper which have used the mean waiting and turnaround times as QoS of end users*.

The *contributions* of this paper are as follows: (i) Developing a mathematical framework to compute the mean waiting and turnaround time of an end user when multiple users simultaneous access the server, (ii) Dimensioning of network link bandwidth and distribution of web servers to satisfy end user QoS

The rest of the paper is organized as follows. In section 2, we analyze the transaction times for TCP and T-TCP respectively and derive analytical expressions for the mean waiting and turnaround times. Section 3 discusses and compares results from real world experiments, followed by concluding remarks in section 4.

2. MODELING AND ANALYSIS

In this section, we discuss the mean waiting and turnaround times that are the most important QoS factors perceived by end users. Mean waiting and turnaround time are defined as the average time when users need for the start and the finish of web service respectively when multiple users access the web server simultaneously. Touch et. als [11] measured the inefficiency of a HTTP protocol when inter-working on TCP. They analyzed the interaction time of HTTP and computed the connection time per transaction, and an upper limit on the additional slow start time and compared it with the most suitable transmission time without slow start. Also, they assumed that the processing capability of a host is only affected by the bandwidth of a network; they ignored the processing time of server and disk input/output bottlenecks. Moreover, they don't consider the mean waiting and turnaround times that are important criteria for end to end QoS defined in this paper. If the mean waiting and turnaround times desired by users are given, we can find the minimum bandwidth or server distribution strategy to satisfy end-users QoS. We define the following notations that will be used in our model in section 2 and 3.

2.1 Terminology and Notations

R: RTT represents the round trip time between the client and the server.

bw: bandwidth represents the transmission speed of link (bps: bit per second)

MSS: maximum segment size (packet size)

K: the number of packets in the file which is given by $K = \left\lceil \frac{filesize}{MSS} \right\rceil$

L: the number of packets in one window, and is given by $L = \frac{bw \times R}{MSS}$

M: maximum useful window size (lower bound). It is given by the smaller of the number of packets filling the pipe during an RTT at the maximum available window size and the number of packets in the file. It, therefore, represents the maximum number of packets that can be transmitted at any time, and is given by $min\ (L, K)$.

S: number of round trips during TCP's slow-start (upper bound). TCP's initial sending window starts at one MSS. However, in most BSD implementations, it is increased to two on receipt of ACK for the SYN packet. The window for data transport, therefore, starts at one and increases by power of two for each round trip time. It is therefore given by $S = \lfloor \log_2 \lceil M/2 \rceil \rfloor$.

The slow start time is given by $R \times S$. From Figure 1, the actual file transmission time is the sum of the minimum file transmission time (T_f) and the additional delay ($R \times S$) due to slow start. Note that although $R \times S$ has to be included as part of the time for TCP Data (reply), it is depicted in the upper part of the figure in order to avoid confusion with T_{min} to be discussed later.

W: wasted time. It is the sum of one round trip time for connection setup and at most one roundtrip time for slow-start. It is given by W = slow start time + connection setup time = $R \times S + R$.

T_f: minimum file transmission time without considering slow-start. It is obtained by dividing the file size by the bandwidth, i.e. $T_f = filesize/bw$. However, because TCP uses slow-start in a real environment, this mechanism will limit the initial window to only one packet until the first ACK is received, and subsequently increase the window gradually by power of 2. Thus, even if the network has enough bandwidth, the client cannot use the full bandwidth until the window grows large enough, i.e. the actual file transmission time will be longer due to the influence of slow-start.

T_{min}: minimum transaction time. It is the sum of RTT for HTTP request and the minimum file transmission time without slow start time. It is given by $T_{min} = R + T_f$.

T: transaction time for TCP. It is given by $T = T_{min} + W = 2R + T_f + R \times S$.

$T_{T\text{-}TCP}$: transaction time for T-TCP. It is obtained by excluding the connection setup time (R) in T and is expressed as $T_{T\text{-}TCP} = R + T_f + R \times S$.

t_w: total waiting time

t_T: total turnaround time

$\overline{t_w}$: mean waiting time

$\overline{t_T}$: mean turnaround time

The following additional notations are used when all packet transaction times are same.

m: total number of users

n: the number of packets requested per user

t: transaction time per packet (T/n or $T_{T\text{-}TCP}/n$)

The following additional notations are used when packet transaction times are different.

P: The number of different types of links depending on their bandwidths, such as T1, E1, etc.

m_i: the number of users using bandwidth type i(=1,2…,P)

m: total number of users as given by $\sum_{i=1}^{P} m_i$

t_i: transaction time per packet using bandwidth type i

 ISBN: 1-56555-284-9

2.2 HTTP over TCP and T-TCP

HTTP uses one TCP connection for a transaction and does not need any control channel in the server or the client. The timeline for file transmission in HTTP is shown in Figure 1.

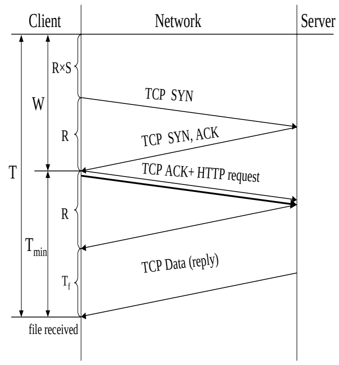

Figure 1. Timeline for file transmission using HTTP over TCP

We have already discussed several problems incurred as a result of the use of TCP by HTTP. As shown in Figure 1, TCP must set up a connection and execute the congestion control mechanisms such as slow-start before sending the HTTP request.

In Figure 1, the most desirable transmission (optimal transmission) will not include the connection setup process and congestion management mechanism. That is, it is to send HTTP request and to receive a file with the maximum transmission speed corresponding to the bandwidth (*bw*).

Figure 2 shows the timeline for file transmission using HTTP over T-TCP. In Figure 2, the reason that the position of $R \times S$ is appeared in the first part of the figure is same as the reason in Figure 1.

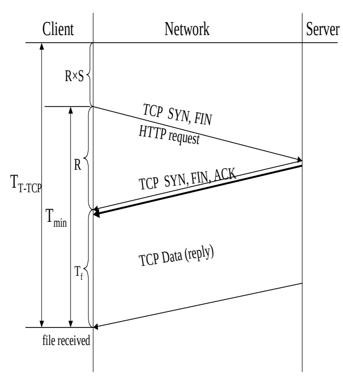

Figure 2. Timeline for file transmission using HTTP over T-TCP

We make the following assumptions for developing the model for the mean waiting and turnaround time:

(1) Web server processes packets in a round robin fashion.
(2) All users request the same file, i.e. the number of packets per user is same
(3) Server processing time is ignored. For example, CPU and disk I/O performance of the server are not considered.

2.3 Analysis of mean waiting and turnaround times

In this section, we develop analytical models for the mean waiting and turnaround times for two cases depending on whether the packet transactions times are same or not.

The transaction time (T) found earlier is total time for a user to connect to a web server and download a file. Since we assume per packet round robin for web service and the download consists of several packets, packet transaction time, $t = T/n$ (assuming same time for all transactions). Also, n is given by (*file size*) / *MSS*. Now, if there are multiple clients requesting the same file from web server, each user's expected transaction time (T) will be the same. However, the actual finishing time of client will be affected by the number of users who are accessing the server simultaneously. When the packet transaction times (t) of users are same, the mean waiting and turnaround times are given by Eqs. (2) and (4) respectively. On the other hand, when the transaction times per packet are different, the mean waiting and turnaround times are given by Eqs. (5) and (6) respectively.

2.3.1. Case 1: Same packet transaction times

When the web servers are connected to the external users through only one link, the total waiting time, the mean waiting time, total turnaround time and mean turnaround time are given by the following equations:

$$t_W = \sum_{i=1}^{m}(m-i)t + m(n-1)(m-1)t \quad (1)$$

$$\overline{t_W} = \frac{\sum_{i=1}^{m}(m-i)t + m(n-1)(m-1)t}{m} \quad (2)$$

$$t_T = mt \times [m(n-1) + \frac{m+1}{2}] \quad (3)$$

$$\overline{t_T} = [m(n-1) + \frac{m+1}{2}]\,t = [\frac{2mn-m+1}{2}]\,t \quad (4)$$

2.3.2 Case 2: Different packet transaction times

When the web servers are connected to the external users through many links of different bandwidths, the mean waiting and turnaround time are given by Eqs. (5) and (6)

 ISBN: 1-56555-284-9

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) × t_i + (total packet transaction time excluding i^{th} group)]. The waiting time of i^{th} user is the sum of transaction 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 t_0 are zeros in the equation.

$$\overline{t_W} = \frac{(n-1)\sum_{i=1}^{P} m_i[(m_i-1)t_i + \sum_{i=1, j\neq i}^{P} m_j t_j]] + \sum_{i=1}^{P}[\sum_{j=1}^{i} m_i(m_{j-1}t_{j-1}) + \sum_{j=1}^{m_i}(j-1)t_i]}{m} \quad (5)$$

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 transaction time (t_i)]. The turnaround time of any user in the second interval is the sum of transaction times of other users prior to him and his own packet transaction time. Thus, by generalizing and adding this all, we obtain the following equation. Both m_0 and t_0 are zeros in the equation.

$$\overline{t_T} = \frac{m(n-1)\sum_{i=1}^{P} m_i t_i + \sum_{i=1}^{P}[\sum_{j=1}^{i} m_i(m_{j-1} \cdot t_{j-1}) + \sum_{j=1}^{m_i} j t_i]}{m} \quad (6)$$

2.3.3 Case 3: Distributed servers

If the web servers are distributed over the individual lines, we first find the mean waiting and turnaround times by applying Eqs. (2) and (4) to each line individually. Since the number of different bandwidth types is assumed to be P, we add up individual mean waiting and turnaround times and divide by P. Hence, two mean values are same as Eq. (7) and Eq. (8) respectively.

$$\overline{t_w} = \sum_{k=1}^{P} \frac{\overline{t_w^k}}{P} = \sum_{k=1}^{P} [\frac{\sum_{i=1}^{m_k}(m_k - i)t_k + m_k(n_k - 1)(m_k - 1)t_k}{m_k}] / P \quad (7)$$

$$\overline{t_T} = \sum_{k=1}^{P} \frac{\overline{t_T^k}}{P} = \sum_{k=1}^{P} \frac{[m_k(n_k - 1) + \frac{(m_k + 1)}{2}]t_k}{P} \quad (8)$$

In the equation, $\overline{t_w^k}$, $\overline{t_T^k}$, m_k, n_k, and t_k represent the mean waiting time, the mean turnaround time, the number of users, the number of packets, and the packet transaction time of the k^{th} line respectively.

3. PERFORMANCE EVALUATION

In our experiment, we connected a HTTP server (Linux 6.1 OS / Apache 1.3.x Web Server) and clients (SUN ultra-5 WS / Solaris 2.6 OS) directly through 10 Mbps Ethernet. We used files sizes of 3000, 6000 and 12000 bytes. *MSS* was 1460 bytes and *R* (*RTT*) was measured as 0.5 ms. We composed the socket interface using UNIX C language in the client, and varied the number of users (1, 5, and 10) with the same number of threads as the number of users. Table 1 shows results obtained by applying terminology and notations in section 2.1 to the experimental environment of which bandwidth = 10 Mbps, *RTT* = 0.5 ms and *MSS* = 1460 bytes. The reason that T_{min} and $T_{T\text{-}TCP}$ are the same in the table is because *S* was calculated as zero in the experimental setup. If *S* is not zero, $T_{T\text{-}TCP}$ becomes larger than T_{min} by $R \times S$ as shown in section 2.2.

file size (byte)	*K* (pkts)	*L* (pkts)	T_f (ms)	*M* (pkts)	*S* (pkts)	*W* (ms)	T_{min} (ms)	*T* (ms)	$T_{T\text{-}TCP}$ (ms)
3000	2.05	0.43	2.46	0.43	0	0.5	2.96	3.46	2.96
6000	4.11	0.43	4.92	0.43	0	0.5	5.42	5.92	5.42
12000	8.22	0.43	9.83	0.43	0	0.5	10.33	10.83	10.33
20000	13.70	0.43	16.83	0.43	0	0.5	16.88	17.38	16.88
50000	34.25	0.43	40.96	0.43	0	0.5	41.46	41.96	41.46

Table 1. Results from the analytical model in the experimental environment

Table 2 and 3 show the comparison of mean waiting and turnaround times of the proposed model and experiment for same and different packet transaction time respectively. *T* and $T_{T\text{-}TCP}$ of the computation column in Table 2 have been reproduced from Table 1; the corresponding values in the experiment column are the average of results obtained by executing the above C program ten times for the same size file and the number of users. In Table 2, the number of packets (*n*) represents the ceiling of value (*K*) that divided file size into MSS. Each packet transaction time (*t*) is a value to divide total transaction time (*T*) into number of packets (*n*). We obtained $\overline{t_w}$ and $\overline{t_T}$ by applying *n*, *m*, *t* to Eq. (2) and Eq. (4) of section 2.3.1. For the different packet transaction times, we divide the users into groups of five-user (m_1) and ten-user (m_2) for each file size. In the computation column of Table 3, *T* (t_i) and $T_{T\text{-}TCP}$ (t_i) are reproduced from Table 1, and those in the experiment column are the minimum among values obtained by executing the C program ten times. $\overline{t_w}$ and $\overline{t_T}$ were found by using Eqs. (5) and (6) in section 2.3.2 respectively.

The reason that the computation values are less than the experimental ones is because the server processing time in the C program is included in the latter. We can infer that the

 ISBN: 1-56555-284-9

difference between the computational and experimental values represents the processing time.

Table 2. Comparison of results between model and experiment for same packet transaction times (All times are in milliseconds)

file size (bytes)	*n*	*m*	computation [Eq. (2)~ Eq. (4)]					
			TCP			T-TCP		
			$T(t)$	$\overline{t_W}$	$\overline{t_T}$	$T_{T\text{-}TCP}(t)$	$\overline{t_W}$	$\overline{t_T}$
3000	3	1	3.46 (1.15)			2.96 (0.99)		
		5		12	15		10	13
		10		26	29		22	25
6000	5	1	5.92 (1.18)			5.42 (1.08)		
		5		21	27		19	25
		10		48	54		44	49
12000	9	1	10.83 (1.20)			10.33 (1.15)		
		5		41	52		39	49
		10		92	103		88	98
file size (byte)	*n*	*m*	experiment					
			TCP			T-TCP		
			$T(t)$	$\overline{t_W}$	$\overline{t_T}$	$T_{T\text{-}TCP}(t)$	$\overline{t_W}$	$\overline{t_T}$
3000	3	1	14.30 (4.77)			14.25 (4.75)		
		5		48	62		47	61
		10		107	122		107	121
6000	5	1	18.75 (3.75)			18.25 (3.65)		
		5		68	86		66	84
		10		152	171		148	166
12000	9	1	35.63 (3.96)			35.13 (3.90)		
		5		135	170		133	168
		10		303	339		298	333

Table 3. Comparison of results between model and experiment for different packet transaction times (All times are in milliseconds)

file size (byte)	*n*	*user/ group* (m_i)	computation [Eq. (5)~Eq. (6)]					
			TCP			T-TCP		
			$T(t_i)$	$\overline{t_W}$	$\overline{t_T}$	$T_{T\text{-}TCP}(t_i)$	$\overline{t_W}$	$\overline{t_T}$
3000	3	5	3.46 (1.15)	36	38	2.96 (0.99)	31	33
		10						
6000	5	5	5.92 (1.18)	70	75	5.42 (1.08)	64	69
		10						
12000	9	5	10.83 (1.20)	138	148	10.33 (1.15)	133	142
		10						
file size (byte)	*n*	*user/ group* (m_i)	experiment					
			TCP			T-TCP		
			$T(t_i)$	$\overline{t_W}$	$\overline{t_T}$	$T_{T\text{-}TCP}(t_i)$	$\overline{t_W}$	$\overline{t_T}$
3000	3	5	18.0(6.0)	157	168	17.5(5.8)	152	163
		10	13.0(4.3)			12.5(4.2)		
6000	5	5	22.0(4.4)	224	240	21.5(4.3)	218	234
		10	17.0(3.4)			16.5(3.3)		
12000	9	5	38.0(4.2)	471	505	37.5(4.2)	464	498
		10	36.0(4.0)			35.5(3.9)		

Table 4 shows the transaction time for several bandwidths. We assumed bandwidths as T1 (1.544 Mbps), E1 (2.048 Mbps), T3 (44.736 Mbps), and ATM OC-3 (155 Mbps) and used 512 bytes as the *MSS* of TCP in WAN. *RTT* is the measured value in the dedicated line to outer WAN and it was 0.1 ms. As for the file size, 6 KB were used and these values mean lengths of files stored in web server. Of course, we may compute while changing the above parameters. *K*, *L*, T_f, *M*, *S*, *W*, T_{min}, *T*, and $T_{T\text{-}TCP}$ computed on each bandwidth are same as Table 4.

Table 4. Transaction time for several bandwidths

bandwidth (Mbps)	*file size* (byte)	*K* (pkts)	*L* (pkts)	T_f (sec)	*M* (pkts)	*S* (pkts)	*W* (sec)	T_{min} (sec)	*T* (sec)	$T_{T\text{-}TCP}$ (sec)
T1 (1.544)	6000	11.72	37.70	0.03	11.72	2	0.3	0.13	0.43	0.33
	1 M	2048	37.70	5.56	37.70	4	0.5	5.66	6.16	6.06
E1 (2.048)	6000	11.72	50.00	0.02	11.72	2	0.3	0.12	0.42	0.32
	1 M	2048	50.00	4.19	50.00	4	0.5	4.29	4.79	4.69
T3 (44.736)	6000	11.72	1092	0.00	11.72	2	0.3	0.10	0.40	0.30
	1 M	2048	1092	0.19	1092	9	1.0	0.29	1.29	1.19
ATM-OC3 (155)	6000	6000	3784	0.00	3784	2	0.3	0.10	0.40	0.30
	12000	12000	3784	0.00	3784	3	0.4	0.10	0.50	0.40

To see the effect of server distribution, we considered six groups while changing the number of users from 10 to 100. Also, we assumed two T1's, T1 and E1, two E1's, T1 and T3, E1 and T3, and mixed T1, E1, T3 for the combination of different lines. The number of users (m_i) for each group were assumed as half of total number of users when the number of lines is two and a ratio of 3:3:4 of those when the number of lines is three. Time divided total transaction time (*T*) into the number of packets (*n*) in Table 4 was used as the packet transaction time (t_i) of each group. Table 5 represents the result when the file size is 6,000 bytes and *MSS* is 512 bytes.

Table 5. Mean waiting and turnaround time with the mixed bandwidth environment

Group	*bandwidth*	m_i	*a*	*a'*	*b*	*b'*
each number of lines is one	T1	50	20.29	20.72	20.29	20.72
	E1	50	19.72	20.14	19.72	20.14
	T3	50	18.60	18.99	18.60	18.99
1	T1 * 2	25/line	9.94	10.37	20.29 (10.15)	20.72 (10.36)
2	T1	25	9.94	10.37	19.60 (9.80)	20.45 (10.23)
	E1	25	9.66	10.08		
3	E1 * 2	25/line	9.66	10.08	19.72 (9.66)	20.14 (10.08)
4	T1	25	9.94	10.37	19.05 (9.53)	19.87 (9.94)
	T3	25	9.11	9.50		
5	E1	25	9.66	10.08	18.77 (9.39)	19.58 (9.79)
	T3	25	9.11	9.50		
6	T1	15	5.76	6.19	18.61 (6.20)	19.85 (6.62)
	E1	15	5.64	6.05		
	T3	20	7.21	7.61		

Now, we define savings rate in case of server distribution as the Eq. (9).

$$Savings\ \ Rate = \frac{a(a') - b(b')}{a(a')} \times 100\ \ (\%) \qquad (9)$$

 ISBN: 1-56555-284-9

In Equation, *a*, *a'*, *b*, and *b'* represent the mean waiting time without server distribution, the mean turnaround time without server distribution, the mean waiting time with server distribution, and the mean turnaround time with server distribution respectively. *a*, *a'* are obtained by applying Eq. (2) and Eq. (4) respectively. b, b' are obtained by applying Eq. (7) and Eq. (8) in section 2.3.3. By applying the above-mentioned procedure to 1 MB and 6 KB files, we found *a*, *a'*, *b*, and *b'* respectively. Then, by Eq. (9), we obtained Figure 3 showing the savings rate (%). It is known that in Figure 3, the savings rate of the mean waiting time is superior to that of the mean turnaround time in both 6 KB and 1 MB files and becomes up to 50 ~51 percent generally.

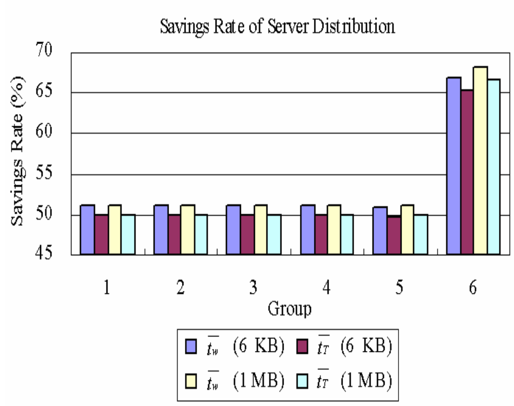

Figure 3. Effect of server distribution

4. CONCLUSIONS

The explosive growth in demand for web service in recent years has increased the demand for the Internet. However, the real access speed experienced by end users is dropping as compared to the early Internet. Although HTTP, a representative protocol of web, was designed regardless of the underlying network structure, TCP/IP is the predominant underlying protocol. Therefore, HTTP has to interact with the TCP layer. Since the two protocols don't inter-work optimally, a number of problems such as several hundreds of ms of delay in connection setup, strong influence of slow-start due to the small size of web files, and constant value of the Time_Wait status in connection release reduce the end to end performance of users.

In this study, we first examined performance problems of HTTP over TCP, and then computed the transaction time on each bandwidth while considering window size and slow start. We derived analytical models to compute the *mean waiting* and *turnaround time* for simultaneous access of the web server by several users. This study *focused* on the performance degradation of the TCP transaction time as a function of the increasing number of users. The *objective* of study is to provide the theoretical upper limit about the real environment when we assume the mean waiting and turnaround time as QoS of end-user. Our analytical model has been validated through experimental results. We found that the mean waiting time can be reduced by 50% by properly distributing the load on the web servers. The results reported in this paper can be used to find alternatives about how to distribute amount of bandwidths in order to satisfy the mean waiting and turnaround times experienced by end-users. On the other hand, we confirmed that the waiting and turnaround times of T-TCP are better than that of TCP, and it becomes more efficient as the file size or the bandwidth becomes smaller. Future extension of this work includes performance analysis of other improved TCP protocols and environments with stochastic transaction times.

References

[1] Berners-Lee, J., Fielding, R. and Frystyk, H. 1995. *Hypertext Transfer Protocol (HTTP/*1.0), RFC-1945.

[2] Postel, J. 1981. *Transmission Control Protocol*. RFC-793 / STD-007.

[3] Stevens, R. 1991. *TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP, and the UNIX Domain Protocols*, Addison-Wesley.

[4] Lee, Y. 1997. *The study on the agent for directory in e-commerce.* Korea Electronics Telecommunication Research Institute: 56-108.

[5] Rescorla. E. and Schiffman. A. 1996. “The secure Hypertext Transfer Protocol”

[6] Braden. R. 1992. *Extending TCP for Transactions-Concepts*, RFC-1379.

[7] Braden. R. 1994. *T/TCP - TCP Extensions for Transactions: Functional Specification*. RFC-1644.

[8] Mogul. J. 1995. “The Case for Persistent-Connection HTTP.” In *Proceedings of the ACM Sigcomm '95*: 299-313.

[9] Fielding. R. Frystyk. H. and Lee. T. 1996. *Hypertext Transport Protocol (HTTP/1.1)*, RFC-2616.

[10] Simon. E. 1995. *Next Generation Hypertext Transport Protocol*, Internet Draft.

[11] Heidemann. J. Obraczka J. and Touch. J. 1997. “Modeling the Performance of HTTP over Several Transport Protocols,” *IEEE/ACM Transactions on Networking*, 5, no. 5: 616-630.

[12] Heidemann. J. 1997. “Performance Interactions between P-HTTP and TCP Implementations,” *ACM Computer Communications Review*: 65-73.

[13] Jacobson. V. 1988. “Congestion Avoidance and Control,” In *Proceedings Of the ACM Sigcomm '88*.

[14] Jacobson. V. and Karels. M. 1998. “Congestion Avoidance and Control,” *ACM CCR*, 18, no.4: 314-329.

[15] Mogul, J. and Deering, S. 1990. *Path MTU Discovery*, RFC-1191, DECWRL, Stanford University.

[16] Padmanabhan, V. and Mogul, J. 1994. “Improving HTTP Latency,” In *Proceedings of the Second International WWW Conference*.

 ISBN: 1-56555-284-9