Performance Modeling and Evaluation of Web Systems with Proxy Caching

Yasuyuki FUJITA, Masayuki MURATA and Hideo MIYAHARA[a]

[a] Department of Infomatics and Mathematical Science
Graduate School of Engineering Science, Osaka University
Toyonaka, Osaka 560–8531, Japan

As the number of WWW (World Wide Web) is widely used on the Internet, it becomes impormtant how *high–quality service* is provided to WWW users. For this, it is necessary to properly estimate a required amount of network resources which includes the Web server machine and network capacities. For modeling and performance evaluation of the Web systems, we conducted the benchmark tests and in this paper, we report the result for the Proxy server. So, we first summarize the results of benchmark tests, and give some implications obtained through the benchmark tests for performance modeling of Web systems. Then we propose and evaluate a performance model of the entire Web systems.

1. INTRODUCTION

The Internet is growing rapidly. Accordingly, studies on traffic characteristics of the Internet become important to deeply understand the traffic behaivior of the Internet and to build the efficient Internet. The characteristics of WWW traffic have already been investigated by several researchers; see, e.g., [1]. However, their studies do not consider the performance of the Web server. If the Web server has many requests, it is likely that the Web server easily becomes a bottleneck. That is, the quality of service offered to users is affected not only by the network bandwidth but also by the Web server processing power.

On the other hand, in [2], the characteristics of the Web server performance is investigated by collecting the fundamental results through the benchmark tests. Then, the performance model of the Web server is proposed based on the experimental data. By the benchmark tests, we confirmed that the Web server exhibits high performance by preparing the helper process for the *httpd* daemon on the Web server. In this case, the Web server can be modeled by the combination of FIFO queueing discipline at the dispatcher and PS scheduling discipline once the request is assigned to one of the helper processes. Accordingly, in [3], the Web server is modeled as an M/G/1/PS queue with a limited number of jobs allowed in the server. Using our approximate analytical method, the way to improve the Web server performance is discussed. However, the above studies focus on the characteristics of the Web server performance, and do not consider the modeling of the entire Web system. Recent Web system often uses the Proxy caching. Therefore we need to take account of the effect of decreasing the network traffic load by the document caching and the overhead by caching on the Proxy server. Concerning the Proxy server, we can find a lot of studies on the caching algorithm and the distributed Web server system based on the caching technologies [4, 5].

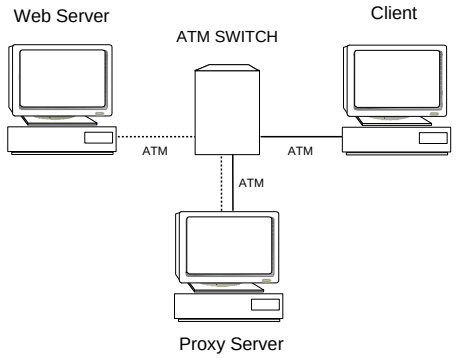

Figure 1: Experimental system configuration.

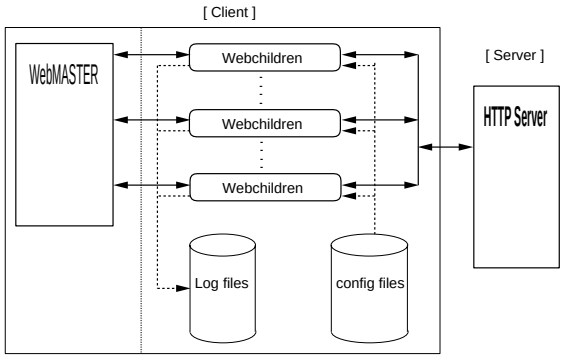

Figure 2: WebStone structure

We first investigate quantitative performance characteristics of the Proxy server by the benchmark test. Recently some benchmark tools are available. Those benchmark tools run independently of the server platform or server software running on it, and clients prepared by the tools generate requests to the server to examine the server's behavior and performance (e.g., the response time, server throughput and so on). In this paper, we show the benchmark results of the Proxy server by using WebStone [6]. We will explain the outline about WebStone in netx section.

The remainder of this paper is organized as follows. We introduce our experimental environment in Section 2. We then show the experimental results for modeling the Proxy server performance in Section 3. In Section 4, a modelling approach of the Web server and the Proxy server models. We also show the examples of performance evaluation of the Web system in order to demonstate the applicability of our approach. Futhermore, we investigate the applicability of the analytical method to slove the above network models. Finally, we conclude our paper in Section 5.

2. EXPERIMENTAL CONFIGURATION

First, our experimental system is illustrated in Figure 1. We used the ATM switch to interconnect the client, the Web server and the Proxy server to avoid the network being bottleneck. The document request from the client is sent to the Proxy server first. If the document does not exist in the Proxy server, the Proxy server forwards the request to the Web server. As the Proxy server retrieves the document, the Proxy server transfers its document to the client.

In our experiments, we control the document hit ratio explicitly to investigate the performance capability of the Proxy server. When we set the hit ratio to be 100%, the client always request the document located in the Proxy server. On the other hand, when the hit ratio is set to 0%, the client does not request the document on the Proxy server, but on the Web server. We set the Proxy server option that the Proxy server does not cache the document in the latter case, by which the document requested from the client is always forwarded to the Web server. Then, the performance of the Proxy server can be found for given hit ratio. While the hit ratio must depend on the the caching algorithm implemented within the Proxy server, the differences are not large as shown in [1].

Table 1: Experimental system

	Web Server	Proxy Server	Client
Machine	Indigo2(SGI)	Indigo2(SGI)	Origin200(SGI)
CPU	IP26 75 MHZ	IP22 250 MHZ	IP27 180MHz *2
Main Memory	320 Mbytes	192 Mbytes	256 Mbytes
Web Server	Apache 1.2.5	—	—
Proxy Server	—	Squid 1.1.20	—
Benchmark	—	—	WebStone 2.0.1

The hardware and software configurations in our experiments are summarized in Table 1. We used Apache (version 1.2.5) [7] and Squid (version 1.1.20) [8] for the HTTP server and for the Proxy server, respectively. Both of two servers are configured as follows; the logging option is set as default. Document files are stored in the local disks of the Web server and the Proxy server. On all of the experimental machines, we did not modify the operating system for tuning TCP/IP stack. That is, we set up the experimental system similar to the commonly used Web servers and Proxy servers.

We used WebStone (version 2.0.1) [6] by which we specify the distribution of requested document sizes, the frequency of access to the Proxy server, and the number of clients generating loads for the Proxy server. Figure 2 illustrates the WebStone structure. The "Webchilderen" are controlled by the "WebMASTER" which remotely spawns the multiple Webclidrens on one or more client machines. Each Webchildren issues the request one after another to the Web server. More precisely, after the Webchildren establishes the HTTP connection with the Web server, it sends one request at a time. After it receives the document, it immediately closes the connection. Then, the new connection is established to obtain the next document. Namely, it employs HTTP version 1.0 [9]. Thus, our environment does not reflect a recent improved version of HTTP 1.1, and thus our results shown in this paper may offer the lower limits if we need to take account the case where the document requests continuously come from the same user. Noting that we used the ATM switch, we found that the overhead due to the network delay is negligible when compared with the time elapsed at the servers. Thus, our experimental results can be used to discuss the perofrmance of the servers.

While WebStone specifies the run rule for conducting the benchmark, we did not follow it in our experiments since we aim at collecting the quantitative data for Proxy server modeling. In such cases, we will explicitly present the configuration. In each simulation setting, we tested five experiments, each of which runs ten minutes, to obtain the reliable results. Then, the average of those five experiments are presented in the below. Last, we note that in our experiments, we only considered the document transfers.

3. EXPERIMENTAL RESULTS

In this section, we present the results of our experiments. First, we show the effect of the number of clients to access the Proxy sever at the same time, and next is the effect of the hit ratio of the cache.

In our experiments, we used the way that the helper processes are prepared on the Web server for *httpd* daemons. The number of helper processes is fixed 16. Discussions on the way to pre-

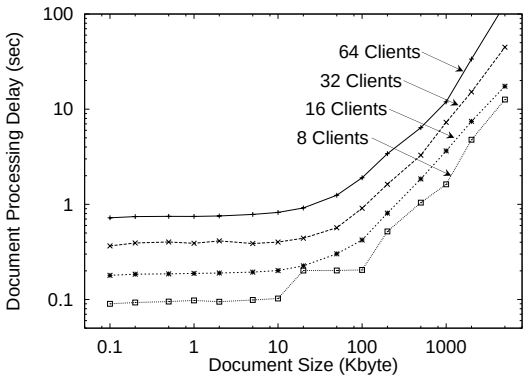

Figure 3: Effects of the number of clients [100% hit ratio] (log scale).

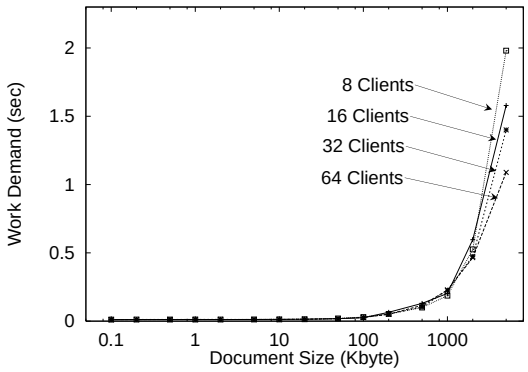

Figure 4: Work demand in the case of 100% cache hit.

pare helper process and its effectiveness can be found in [2]. In our approach, the size of all requested documents is fixed in each experiment.

3.1. Experiment 1: The effect of the number of clients to access the Proxy server

First, we investigate the effect of the number of clients to access the Proxy server at the same time. We change the number of clients to be 8, 16, 32 and 64. The hit ratio of all documents are set to be 100%, by which we can identify the basic performance of the Proxy server because it can exclude the processing time at the Web server. We show the result in Figure 3 where the vertical axis is illustrated in the log-scale. From the figure, we can observe that in the case of the small sized documents, the response time is not in proportion to the document size but is almost fixed. On the other hand, as the document size becomes over about 100Kbyte, the response times increase almost linearly with the document size.

We next show Figure 4 where the response times are divided by the number of the clients. Namely, Figure 4 represents the "work demand" for each request on the Proxy server against the document size. From the figure, we can confirm that the work demand cannot be modelled directly from the the document size. However, the effect of the number of clients is negligible except very heavy traffic load conditions where the context switch for the processes at the server becomes overhead.

3.2. Experiment 2: The effect of the cache hit ratio

In this subsection, we want to investigate the response time characteristics of cached and no–cached documents for given cache hit ratio. In this experiment, the hit ratio is controlled as follows. The client requests two kinds of documents; the cached document and no–cached documents on the Proxy server. The requested document cached on the Proxy server is directly returned from it. On the other hand, the document not on the Proxy server is transferred from the Web server, but the document is not cached on the Proxy server by setting the document name in the configuration file `squid.conf` on the Proxy server. By changing the the access rate of the cached and no–cached documents in `filelist.standard` of WebStone, we can vary the hit ratio on the Proxy server.

Figure 5 presents the fact that as the hit ratio becomes largeer, the response time becomes smaller as expected. And then, we can observe the same tendency as in the previous case (100%

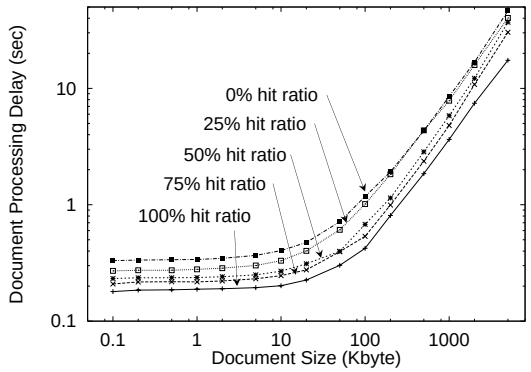

Figure 5: Effects of the cache hit ratio on the average response times (log scale).

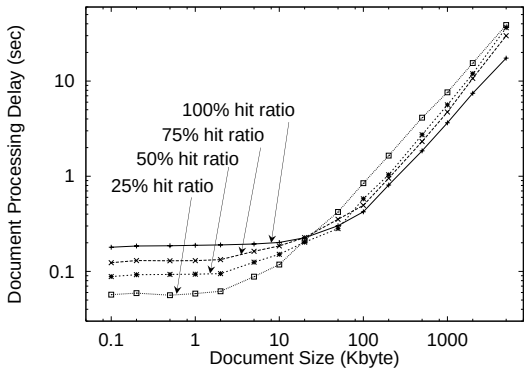

Figure 6: Response times for only cached documents.

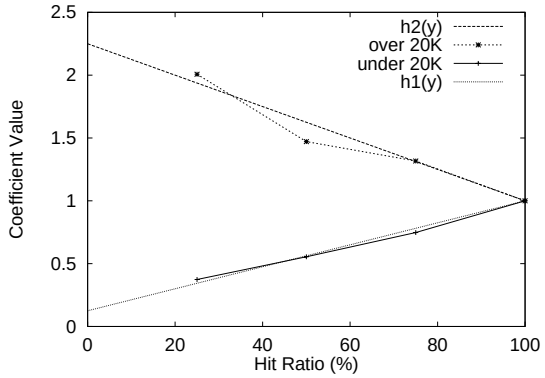

Figure 7: Coefficients of work demands against response times with 100% hit ratio.

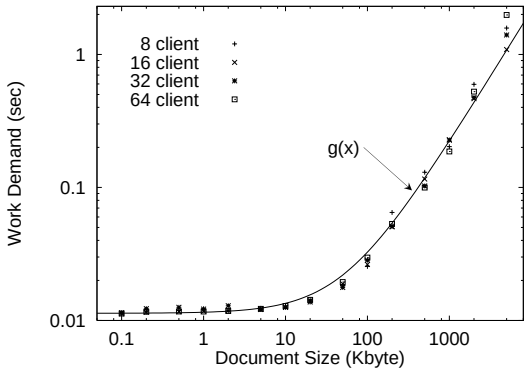

Figure 8: Approximated function for work demand.

hit ratio case, which is also shown in the current figure).

Of course, repsonse times of only cached–documents exhibit different appearance as shown in Figure 6. When the document size is small, the response time becomes large as the hit ratio gets high. It is due to high perocessing load on the Proxy server. However, when the document size becomes large, the lower hit ratio leads to the larger response times. In this case, the processing overhead is balanced between the Proxy and Web servers. Thus, we need to consider the effect of the document size as well as the hit ratio to determine the work demand on the Proxy server. For the above purpose, we take a following simple approach. As shown in the previous figure, the tendencies of the response times are changed when the document size is 20Kbyte. Thus, we consider two regions according to the document size. When the document size is smaller than 20Kbyte, the response times are fictitiously decreased as the hit ratioes becomes small. On the other hand, when the document size is over 20Kbyte, the response times are enlarged according to the hit ratio. Fortunately, the relation between the response times and hit ratios is simple as shown in Figure7; i.e., a linear relation can be found as shown in the figure. Thus, for given document size, we can use linear functions, $h1(y)$ or $h2(y)$, for given hit ratio y, to determine the

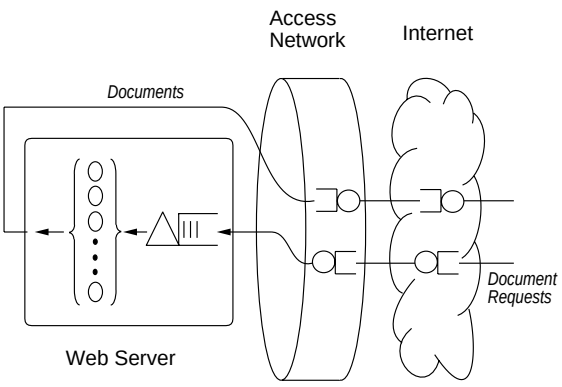

Figure 9: Web server system model.

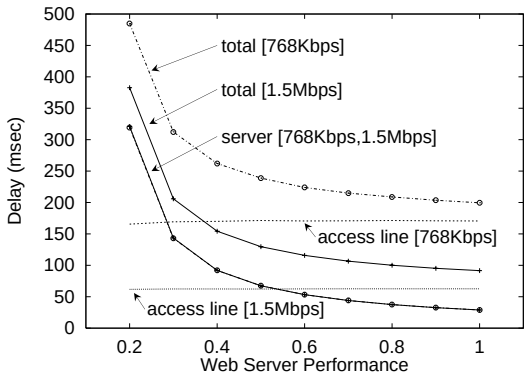

Figure 10: Delays dependent on the web server performance.

work demand once the work demand for the case of 100% hit ratio is found. The work demand for 100% ratio case has already been shown in Figure 4. By utilizing the curve fitting approach, we have the following relation.

$$g(x) = ax + b \tag{1}$$

where $a = 0.01121$, $b = 0.0002329$. See Figure 8, where we plot $g(x)$ for given document size x on Figure 4.

4. WEB SYSTEM MODELING AND EXAMPLES

In this section, we demonstrate applicabilities of our Web system modeling approach to evaluate performance of the Web system.

4.1. Evaluation of the Web server system

We consider the case where the Web site is publicly open to the Internet as shown in Figure 9. The Web server is modeled as an M/G/1/PS with a limited number of jobs [2]. The Web server is accessed via the access line from the Internet users. The access line is modeled as a M/G/1/PS queue [1]. The performance results derived from our model are mainly affected by the Web server performance and the bandwidth of the access line connected to the Internet. Of course, the Internet backbone is also an important factor affecting the performance, but it is beyond our scope how the Internet backbone should be improved. In the current paper, the Internet backbone is expressed by an IS (Infinite Server) queue and the delays follow the Erlangian distribution [10]. Its mean is set to be 195 msec according to [10]. It was obtained by `ping` command between the sites located in the East and the West of United States. As a result, we have an open queueing network model. However, we conducted simulation experiments, since the scheduling discipline taken at the Web server does not satisfy the product–form network. Note that in the current subsection, we do not have the Proxy server in the model. The effect of the Proxy caching will be investigated in the next subsection.

We see the effect of the Web server performance. Figure 10 shows the delays experienced at the Web server and the access line separately. The total delays are also shown in the figure.

In the figure, two cases of the access line capacity are shown; 768 Kbps and 1.5 Mbps. The document request rate is fixed at 5 request/sec. The horizontal axis shows the fictitious relative server performance by setting our Web server to be 1. From the figure, we can observe that the improvement of the Web server performance is important to improve the total delays in the current parameters setting. However, dramatic improvements cannot be observed as the Web server processing power becomes large since the delay within the Internet becomes dominant in that region. Then, one must wait the advancement of the Internet backbone for further performance improvement after the access line and Web server are adequately prepared.

4.2. Evaluation of the model including the Proxy server system

In this subsection, we show the method of modeling the Web system to evaluate the quality of service given to users within a certain network (e.g., the network of Internet service provider). It is illustrated in Figure 11. Recently, most of ISP (Internet Service Provider) provides the Proxy server to improve the document response times. The model is intended to show the applicability of our approach to investigate the effect of the Proxy caching.

See Figure 11 where the hit or miss of the document on the Proxy server is decided independently with given a hit ratio. The validation of this "independent assumption" is given in [1] where the authors show that an correlation effect of the caching algorithm is negligible. A rational behind this result is that if the caching table is large enough, the document misses are likely to happen only due to the wide spread of the WWW document popularity. Queueing models for the Web server, the access line and the Internet backbone are just same as in the previous subsection. Here, we consider the mixed queueing network model where open and closed chains exist in the model. The users within some ISP are assumed to be fixed, and they request the document repeatedly after they get the response. We first show the delays experienced on the

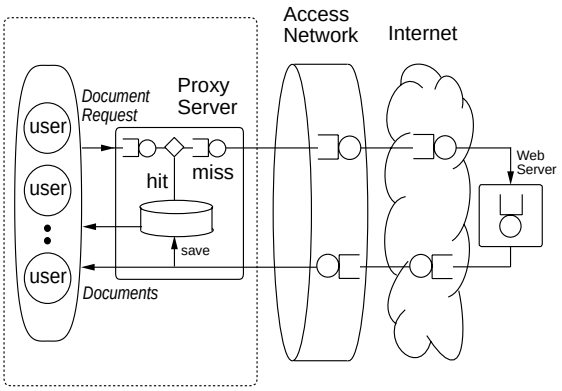

Figure 11: ISP model including the Proxy server.

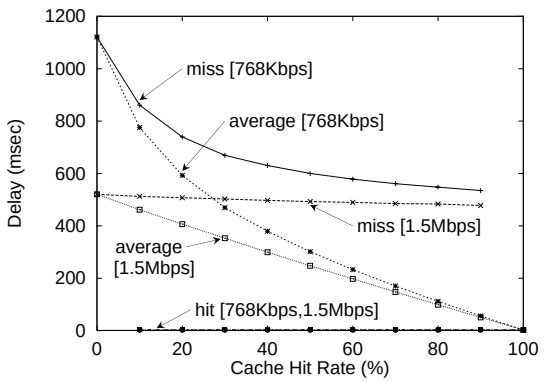

Figure 12: Mean delays on the access line dependent on the cache hit ratio.

access line dependent on the cache hit ratio of the Proxy server in Figure 12. The delays averaged over cache–hit and cache–miss documents are also shown in the figure. The 768 Kbps and 1.5 Mbps access lines are considered. In obtaining the figure, the delay within the Internet backbone is assumed to follow the Eralngian distribution with mean 195msec. In case where

the access line is 1.5 Mbps, the mean delay of the documents is not changed independently of the cache hit ratio. This is because the access line is not the bottleneck in this case, i.e., there is enough bandwidth to transfer the requested document even if the miss ratio is 100%. On the other hand, when the access line bandwidth is decreased to 768 Kbps, the improved cache hit ratio can lead to the smaller document transfer delays to some extent. However, when the cache hit ratio exceeds about 50%, the delays are not very much improved due to the fact that other resources become bottleneck. Since the current caching algorithms offer 50% or 60% hit ratios, 1.5Mbps of the access line bandwidth is a critical value, but more bandwidth is not necessary for ISP to save the cost. Furthermore, the result implies that a more complicated and slightly improved caching algorithm does not help improving the delay within the access network.

Finally, we show the location of the performance bottleneck in our closed queueing network model. Figure 13 illustrates the delays at the access line, the Web server, the Proxy server and the Internet backbone. The horizontal axis shows the delay experienced within the Internet backbone. Other parameters are not changed and the access line capacity is set to be 1.5Mbps. The hit ratio at the Proxy server is 51%. The delay at the Web server is only for document with cache miss. Figure 14 represents the ratio of each delay to the total delay. Noting that we have used 195 msec mean delay for the Internet backbone in the previous examples, the figures imply the effect of higher backbone networks. Of course, the faster Internet backbone improves the total response time as shown Figure13. However, it does not always lead to the dramatic improvement. As shown in Figure 14, by the faster Internet backbone, the performance bottleneck moves to other location; it is the access line in the case of the figure. Our modeling method can identify it, which is one of main purposes of this paper.

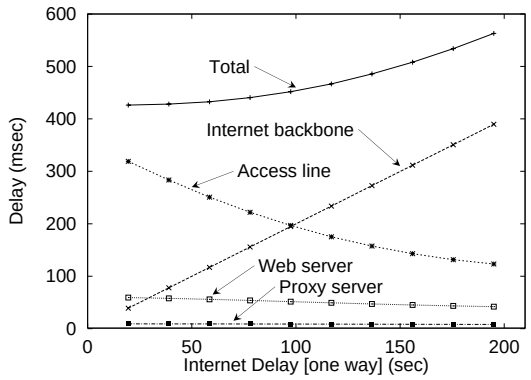

Figure 13: Speed–up effect of the Internet backbone.

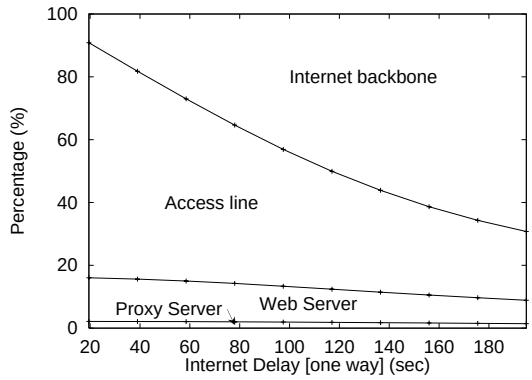

Figure 14: Ratios of document processing and transfer delays.

4.3. Accuracies of an approximate analytical method

So far, we have used the simulation technique to evaluate the queueing network model. It is because the scheduling discipline in the Web server do not satisfy the condition of a product–form solution (see, e.g., [11]). In this subsection, we investigate the applicability of the analytical method to solve the queueing network model. For this purpose, we model the Web server as the Infinite Server (IS) queue where the work demand at the IS queue is obtained from a separate

analysis of the Web server. The response time at the Web server can be obtained by using the method presented in [2] where the arrivals of document requests follow the Poisson distribution. Then, the queueing network models shown in this paper become product–form networks, and we can utilize the convolution algorithm and/or the MVA method for effective numerical computation [11]. However, the arrival rate at the Web server is not known a priori when it is applied to the closed queueing network model.

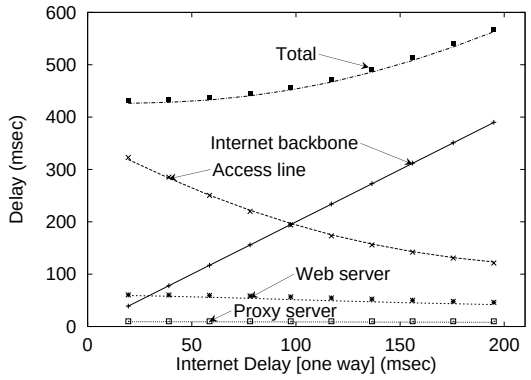

Figure 15: Comparisons of analytical and simulation results.

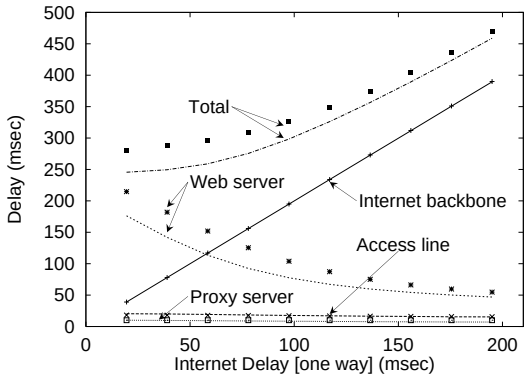

Figure 16: The case of 6Mbps access line.

To assess the accuracy of the above approximate analysis, we compare those with simulation experiments. The access line is set to be 1.5 Mbps. The lines show the analytical results while the symbols are for the simulation results. In obtaining the figure, the delay of the Internet backbone is changed; i.e., the figure corresponds to Figure 13. In the figure, good accuracies of the analytic method can be observed. In this case, however, either the Internet backbone and the access line is the bottleneck. Since we introduced the approximation in the Web server queue, we need to investigate the situation that the Web server also becomes the bottleneck. For this purpose, we

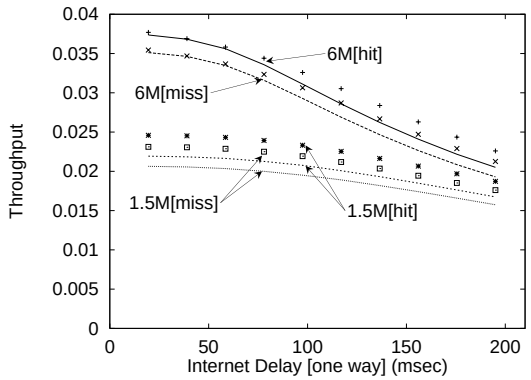

Figure 17: Comparisons of throughput.

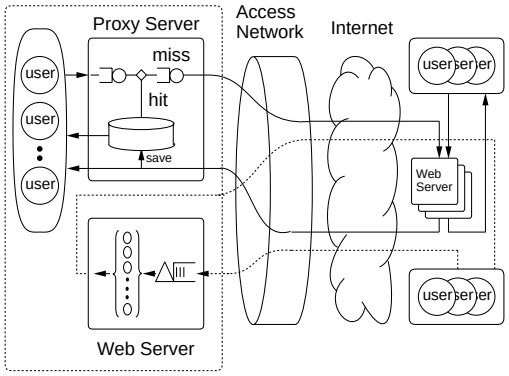

Figure 18: The entire Web system model

change the access line from 1.5 Mbps to 6 Mbps. The comparative results for delays are shown in the Figure 16. In this case, the accuracies are lost, but are sill kept in the reasonable level. The corresponding results for the throughput in the above two cases are summarized in Figure 17.

5. CONCLUSION

In this paper, we have first presented experimental results to investigate the effect of the Proxy caching for the Web system. Based on the experimental results, we have built the model of the Proxy server. Our modeling approach is then demonstrated by using two models; (1) the model in which the Web site is publicly open to the Internet users, and (2) the Proxy server is provided for ISP users within the access network. Our approach can identify the performance bottleneck of the Web system and can be used for its performance planning. In this paper, we have presented the evaluation result of the Web server system and the Proxy server system separately. The entire model would become the one illustrated in Figure 18, and it can be easily evaluated.

In this paper, we have only considered the Web document transfer. It is true that Web traffic dominates the Internet in recent days, but the Web system allows various applications. It is especially important to take account of `cgi` and similar tools since it affects the Web server performance. Further investigations are necessary regarding this problem.

REFERENCES

1. M.Nabe, M. Murata and H. Miyahara, "Analysis and Modeling of WWW Traffic with Cache for Designing Internet Access Networks," *Proceedings of SPIE Conference on Performance & Control of Network Systems*, Nov. 1997.
2. Y. Fujita, M. Murata and H. Miyahara, "Analysis of Web Server Performance towards Modeling and Performance Evaluation of Web Systems," *Proceedings of 1998 IEEE SICON*, pp.221–224, 1998.
3. Y. Fujita, M. Murata and H. Miyahara, "Performance Modeling and Evaluation of Web systems," *Proceedings of 1998 IEEE Communication Quality and Reliability Workshop*, May 1998.
4. Roland P. Wooster, Marc Abrams, "Proxy Caching that Estimates Page Load Delays," 1997.
5. Carlos Maltzahn, Kathy J. Richardson, "Performance Issues of Enterprise Level Web Proxies," *Proceesings of SIGMETRICS'97*, 1997.
6. Silicon Graphics,Inc., "WebStone: World Wide Web Server Benchmarking," available at `http://mail.mindcraft.com/webstone`.
7. Apache HTTP SERVER PROJECT, available at `http://www.apache.org/`.
8. Squid Internet Object Cache, availabe at `http://squid.nlanr.net/Squid/`.
9. T. Berners-Lee, R. Fielding, H. Frystyk, "Hypertext Transfer Protocol – HTTP/1.0," RFC1945, available at `http://ds.internic.net/rfc/rfc1945.txt`, 1996.
10. P. Manzoni and D. Ghosal, "Impact of Mobility on TCPIP: An Integrated Performance Study." IEEE Journal on Selected Areas in Communications, vol.13, no.5, pp.858–867, June 1995.
11. S. S. Lavenberg, *Computer Performance Modeling Handbook*, Academic Press (1983).