The Effect of Web Caching on Network Planning[*]

Hairong Sun, Xinyu Zang and Kishor S. Trivedi
{hairong, xzang, kst@ee.duke.edu}
Center for Advanced Computing and Communications
Department of Electrical and Computer Engineering
Duke University
Durham, NC 27708

Abstract

In this paper, the effect of Web caching on network planning, in sense of bandwidth computation for the access link interconnecting the ISP's subnet with the Internet, is studied by means of simulations. The latency of a browser retrieving files is studied for given traffic characteristics, number of users, bandwidth of access link and cache hit rate. From our analysis, we find that using a well-designed Web cache with 50 % hit rate is more effective than doubling the bandwidth for an ISP's access link to the Internet, with the respect to decreasing retrieval latency. In other word, we can reduce half of the bandwidth of the access link by using Web caching without compromising retrieval latency. The results are very encouraging and useful to the ISPs, enterprises and universities that are planning to interconnect their Intranets with the Internet.

Index Terms: Web Caching, Network Planning, Pareto Distibution, Weibull Distribution

[*] This research was supported in part by the National Science Foundation under Grant No. EEC9418765, and by North Carolina Advanced Network Program as an enhancement project to the Center for Advanced Computing and Communications in Duke University.

1 Introduction

Due to the proliferation of the World Wide Web (WWW), there has been a large increase in the amount of information delivered over the Internet. As Web popularity grows, the number of browsers accessing popular Web servers grows and so does the network bandwidth required to connect browsers to the servers. However, the growth of Internet capacity is not keeping pace with the demand of browsers, and hence users experience longer delays in retrieving files from remote servers as the WWW grows. Internet is getting more and more overloaded, and without appropriate strategy, the WWW would become a victim of its own success. Several projects aiming at increasing the capacity of the Internet, e.g., Internet 2, vBNS, are in progress. Besides scaling network bandwidth to keep up with client demands, using Web cache to store and prefetch Web pages is an economic and practical alternative to alleviate the congestion on the Internet.

Since the majority of Web document requests are directed to static documents (e.g., home pages, audio and video files), caching at various network points provides a natural way to reduce Web traffic. A common form of Web caching is caching at HTTP (HyperText Transport Protocol) proxies, which are intermediateries between browser processors and Web servers on the Internet. Since documents are stored at the proxy cache, many HTTP requests can be satisfied directly from cache instead of generating traffic to and from Web servers. The proxy caching can reduce network traffic, the load on the busy Web servers, as well as the average latency of fetching Web documents. Some studies even shown that the hit rate for the Web proxy caches can be as high as over 70% for a well designed cache proxy [11]. As proxies have finite storage capacity, it is eventually neccessary to replace less useful documents with a more useful ones. Various Web-specific replacement policies have been proposed and compared in the recent studies, e.g., Least-Recently-Used, Least-Frequently-

Used, Size-Based, Lowest-Latency-First, Pitkow/ Recker, Lowest Relative Value, and Cost Aware [1, 2, 3, 4, 5, 6]. Empirical methods and trace-driven simulations have been widely used to evaluate these policies.

Besides the temporal locality, Web traffic also show the spatial locality, which captures the relationship among the particular pages, i.e., if page i is refered at time t, page $i \pm k$ will be more likely to be refered at time close to t. It is reasonable that the user is more probable to access the files or Web sites having a hyperlink to the page the user is browsing. Rarely does the user type in a URL (Universal Resource Locator) to switch to a new page. The spatial locality can be used to prefetch the files that will very likely be requested in the near future, so that the user's average waiting time can be reduced. Various prefetch schemes have been proposed, e.g., Adaptive Network Prefetch, Top 10 Prefetch and Predictive Prefetch [7, 8, 9].

Because of the spatial and temporal locality existing in the Web traffic, Web caching can be used as a near-term and economic solution to alleviate the congestion on the Internet efficiently. However, the previous studies mainly concentrated on the replacement and prefetching algorithms and their performance evaluations with the performance indices called *hit rate* or *byte hit rate*. Another benefit brought by the Web caching, i.e., reduction of latency of retrieving a Web document, has not been evaluated in-depth. From viewpoint of the network planning, the reduction in network traffic and the average latency of fetching Web documents brought by the Web caching can make a tradeoff to reduce the bandwidth required at the access point to the Internet. However, the effect of Web caching on the network planning has not been addressed, which is very important for the Internet Service Providers (ISP), enterprises and universities that are planning to interconnect their subnet with the Internet. Because bandwidth is more expensive than cache, considering the effect of Web caching while planning the network can cut the budgets effectively. In this paper, we study the latency of fetching Web documents, and quantify the bandwidth saving brought by the Web caching.

The traffic model adopted in this paper is the one presented by S. Deng after analyzing sets of actual traffic data [16]. In his empirical model, the traffic generated by a Web browser is an ON-OFF process. During the OFF period, whose duration is Pareto, no request is generated. During the ON period, whose duration is Weibull, a series of requests are generated. The intervals between the adjacent requests follow another Weibull distribution. The document size distribution was previously found in [13] to be Pareto. In this paper, the latency of a browser retrieving Web files is studied by simulations with given traffic characteristics, number of browsers, bandwidth of access link and cache hit rate. From our analysis, we find that using a well-designed Web cache with 50 % hit rate is more effective than doubling the bandwidth for an ISP's access link to the Internet, with the respect to decreasing retrieval latency. In other words, we can reduce half of the bandwidth of the access link by using Web caching without compromising retrieval latency. The results is very encouraging and useful to the ISP, enterprises and universities that are planning to interconnect their Intranets with the Internet.

The organization of the paper is as follows. In Section 2, the system and traffic model are given. Simulation results obtained by using $CSIM$ are discussed in Section 3. Some further research topics and conclusions are given in Section 4.

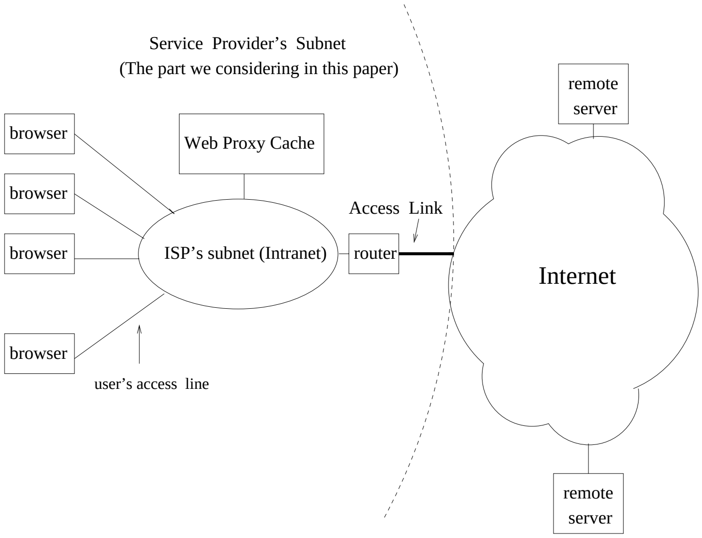


Figure 1: A Typical Network Infrastructure Interconnecting an ISP's subnet (Intranet) with the Internet

2 System Model and Traffic Model

2.1 Web Caching

Fig. 1 illustrates a typical network infrastructure interconnecting a subnet with the Internet. The customers, or browsers, access the subnet via Modem, ISDN BRI (Integrated Service Digital Network- Basic Rate Innterface), Ethernet or Token Ring. When a WWW browser clicks on a hypertext link, several URL requests may be sent from the browser to the Web proxy which may be just one cache or a collection of caches [18, 19, 20, 21]. If the proxy has a copy of the file which the browser is looking for and is consistent with the original copy on the remote server, the proxy sends the copy to the browser. This is called a *hit.* If the proxy does not have a copy of the file that the browser is looking for, the proxy retrieves an original copy from the remote server, sends it to the browser and keeps a copy in the cache. This is called a *miss*. Obviously, although the cache size may increase significantly via collection of caches and some well-designed inter-cache protocols [18, 19, 20, 21], the cache size can not be infinite. When a new file arrives at the cache and the cache is full, some less useful files must be replaced to make room for the new file.

Paging and the replacement policies have been studied extensively by operating system and program behavior researchers in 1970s [22], but Web caches are different from the conventional paging. Files involved in caching and replacement are variable-size and interval between continuous accessses is a complex stochastic process in Web caching while the page replaced is fixed-length and the interval between adjacent accesses is deterministic in the traditional paging case. Finding an optimal replacement algorithm for Web caching is an NP-hard problem. Several Web-specific replacement policies have been proposed in the recent studies, e.g., Least-Recently-Used, Least-Frequently-Used, Size-Based, Lowest-Latency-First, Pitkow/ Recker, Lowest Relative Value, and Cost Aware [1, 2, 3, 4, 5, 6].

In the trace-driven simulations, the structure shown in Fig. 2 is commonly adopted. The traces input to the model are collected in advance, which record the sequence of HTTP requests going through the Web cache proxies and include the entries such as IP addresses for the sources and destinations, time stamps, and file sizes, etc. Some widely used traces are: UC Berkeley Home IP Web Traces [23], Digital's Web Proxy Traces [24], and Virginia Tech's Proxy Traces [25]. Various replacement algorithms and prefetch schemes were compared via trace-driven simulation [1, 2, 3, 4, 5, 6]. Almost all the previous studies concentrated on the evaluation and comparision of *hit rate* or *byte hit rate* which indicate the fraction of documents and bytes being served from the Web cache instead of the remote server. It was found that [10, 11, 17]:

- Different replacement algorithms, prefetch schemes and traces achieve different hit

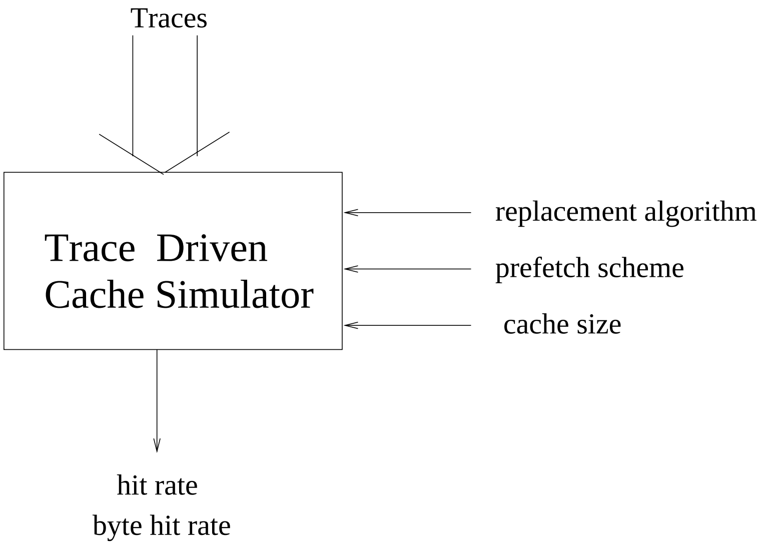


Figure 2: Trace-driven Simulator Structure

rate and byte hit rate.

- The hit rate of a Web cache is proportional to the log of the cache size, and the hit rate can even approach 70 % for some replacement algorithm with 256-Giga-byte cache [11].

Another benefit brought by the Web caching, i.e., reduction of latency of retrieving a document, has not been evaluated in-depth and quantified. From the viewpoint of network planning, the reduction in network traffic and the average latency of fetching Web documents brought by the Web caching can make a tradeoff to reduce the bandwidth required at the access point to the Internet. However, the effect of Web caching on the network planning has not been addressed, which is very important for the ISPs, enterprises and universities that are planning to interconnect their subnets with the Internet.

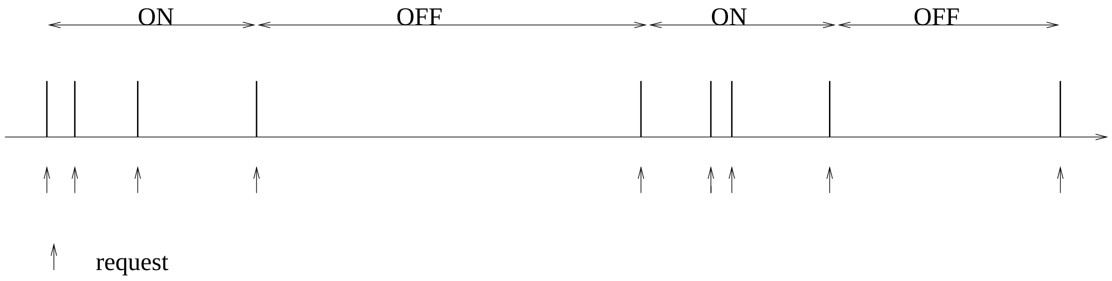


Figure 3: ON-OFF Traffic Model

2.2 Traffic Model

The traffic characteristics for WWW traffic is very complicated and depends on the behavior of a WWW browser, WWW file size and even the design style of the Web pages. A tractable empirical traffic model was constructed to capture the behavior of the WWW browsers by S. Deng from the traces collected at GTE Laboratories Inc. in 1996 [16].

In his model, the traffic generated by a WWW browser is an ON-OFF process (see Fig. 3). The ON periods are initiated by user's clicks on the hypertext links. One click may generate several URL requests among which the first one is created by the browser's click and the followings may be automatically generated by the client program or by the broswer's other clicks. The ON period is found to follow a Weibull distribution whose probability density function (pdf) is given by

$$p_{on}(x) = \frac{k}{\theta}(\frac{x}{\theta})^{k-1}exp(-(x/\theta)^k) \tag{1}$$

with $k = 0.77$ to 0.91 and $\theta = e^{4.4}$ to $e^{4.6}$. The intervals between adjacent requests follow another Weibull distribution with k = 0.5 and $\theta = 1.5$.

During the OFF period, no request is generated. The duration of OFF period follows a Pareto distribution whose probability density function is given by

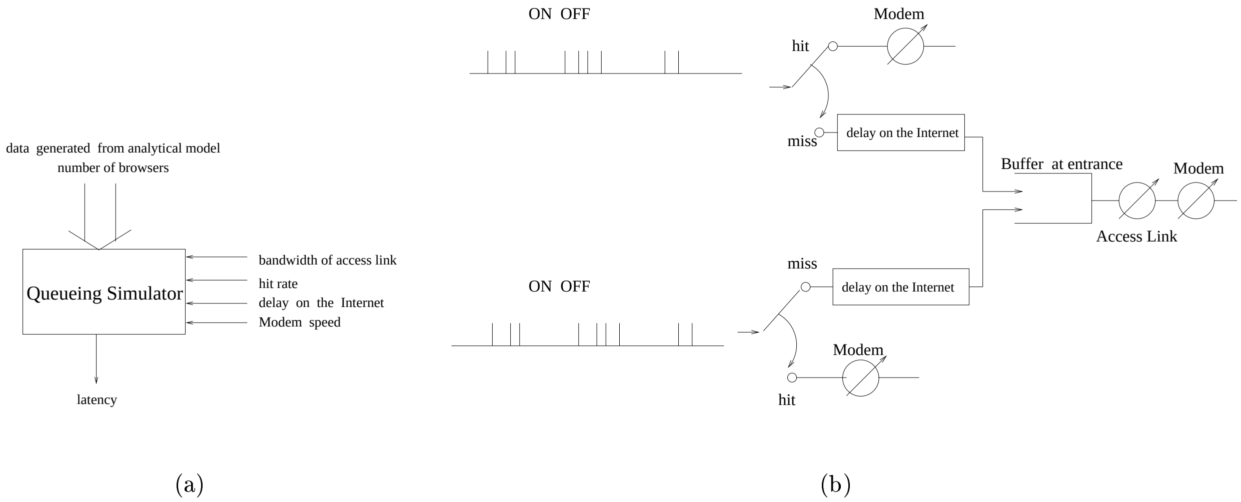


Figure 4: Our Simulation Model

$$p_{off}(x) = \alpha k^{\alpha} / x^{\alpha+1} \tag{2}$$

with $\alpha = 0.58$ to 0.9, $k = 60$.

The document size distribution was previously found in [13] to follow another Pareto distribution with $\alpha = 1.1$ to 1.3, and k is determined by $E[file]$, the mean length of files

$$k = (\alpha - 1)E[file]/\alpha \tag{3}$$

In S. Deng's model, a sequence of document requests with inter-arrival times less than 60 seconds are considered to form an ON period, and the occurrence of a request more than 60 minutes after the previous request indicates an OFF period.

This model is consistent with the other researchers' results [12, 13, 15].

2.3 Simulaton Model

What we are interested in is the file delivering latency, or response time, in Fig. 1, which is defined as the time interval from the browser clicking an object to the requested object being displayed on the monitor. Excluding some trival items, e.g., the delay of monitor displaying and cache retrieving, and the HTTP interaction within the subnet, the response time consists of the following items:

- t_1, the delay related with HTTP interaction in the Internet, which consists of a request sent from the proxy to the server, followed by a response sent back from the server to the proxy. This item depends heavily on the version of HTTP. In HTTP 1.0, the transfer of each HTML (HyperText Marking Language) or image file involves setting up and tearing down a new TCP connection, while in HTTP 1.1, a persistent connection mechanism is defined. It also depends on the propogation delay which depends on the distance between the proxy and the requested server.

- t_2, the delay on the remote server's subnet which depends on the network speed and the load of the remote server.

- t_3, transmission delay and queueing delay on the Internet which depends on the speed of routers the TCP connection traversing and the traffic load along the route. If Web caching is widely adopted, t_4 will decrease significantly. If only a few ISPs adopt Web caching, the reduction of network load and t_4 is not significant because the Internet is an ocean of information where the drainage of a river will not reduce the water level significantly.

- t_4, the time of the files sojourning on the subnet and browser's access line, which consists of the queueing delay and transmission delay. Suppose the subnet is a high speed network, e.g., constructed by Fast Ethernet or LAN switch, or ATM switch, the queueing delay in the subnet is negligible compared with the delay on the Internet. We assume the system in Fig. 1 is an ISP, and the browsers use Modems to access the ISP's subnet. Then

$$t_4 = 8 \;\; f/speed \tag{4}$$

where f is the file length in bytes and *speed* is the speed of Modem in terms of bits per second.

- t_5, the time of the files sojourning at the entrance of the subnet which depends on the speed of the access link and the arrival process of the Web files. We assume that the buffer size at the entrance is infinite. Then

  $$t_5 = 8 \;\; qlength/bandwidth \tag{5}$$

  where *qlength* is the queue length of the entrance buffer in terms of bytes and *bandwidth* is the bandwidth of the access link in terms of bits per second.

In this paper, we study the impact of Web caching on the bandwidth saving for service providers. We simplify the first three items above by reducing the summation of t_1, t_2 and t_3 to be a constant t_{net}, e.g., 4 seconds. This assumption is very coarse and inaccurate, however, because summation of t_1, t_2 and t_3 is on the Internet side and beyond the ISP's network planning, it can make us pay our attention on the delay in the ISP's subnet and at the entrance of the ISP's subnet. Therefore, the total latency or response time t_{latecy} is

$$t_{latency} = \begin{cases} t_{net} + t_4 + t_5, \text{if it is a miss} \\ t_4, \text{if it is a hit} \end{cases} \tag{6}$$

Table 1: Parameters for the ON-OFF Traffic Model

ON duration	Weibull (k=0.9, θ =$e^{4.4}$)
OFF duration	Pareto (k=60, α =0.5)
Interarrival distribution during ON period	Weibull (k=0.5, θ = 1.5)
File size	Pareto (k=300, α =1.3)

Our simulation model is illustrated in Fig. 4, where inputs include the data generated by the empirical traffic model, hit rate, Modem speed and t_{net}. The response time t_{latecy} is the performance index we are interested in. Because the hit rate only represents the ratio of the number of requests served by the Web cache to the total requests, and does not give any information on how the hits or misses happen: all the requests in the same ON period are hit or missed together, or independently. We call the former as bursty and the latter as random. Intuitively, the bursty hit is closer to the real system. All the hits are assumeed to be bursty in this paper unless we indicate otherwise.

3 Simulation Results and Discussion

The simulation program is developed using the package *CSIM* [26]. The simulation time is 500 ON-OFF periods for each browser. Considering the duration of one ON-OFF period is longer than 1 minute, the duration of 500 ON-OFF periods is longer than 8 hours. The Modem speed is 34kb/s. Table 1 gives the parameters of our traffic model.

First we assume that t_{net} is 4 seconds. Fig. 5 compares the Cumulative Distribution Function (CDF) of the response time $t_{latency}$ with 500 browsers for three cases:

- *bandwidth* =256 kb/s, with no Web cache.

- *bandwidth* =512 kb/s, with no Web cache.

- *bandwidth* =256 kb/s, a Web cache with hit rate 0.5.

Clearly, the last two cases have nearly the same CDF, and Web cache with 50 % hit rate is more effective than doubling the bandwidth for an ISP's access link to the Internet, with the respect to decreasing retrieval latency.. Note that 500 here is the number of browsers logging on the Internet *simultaneously*, which approaches the real situation.

Fig. 6 compares the CDF of the response time $t_{latency}$ with 800 browsers and $t_{net} = 4$ seconds for three cases:

- *bandwidth* =384 kb/s, with no Web cache.

- *bandwidth* =768 kb/s, with no Web cache.

- *bandwidth* =384 kb/s, a Web cache with hit rate 0.5.

Once again, the same behavior as that in Fig. 5 is observed.

Fig. 7 compares the CDF of the response time $t_{latency}$ with 800 browsers and $t_{net} = 4$ seconds for three cases:

- *bandwidth* =768 kb/s, with no Web cache.

- *bandwidth* = 1.5 Mb/s, with no Web cache.

- *bandwidth* =768 kb/s, a Web cache with hit rate 0.5.

The same behavior as that in Fig. 5 is observed as well.

In order to study the effect of t_{net}, Fig. 8 compares the CDF of the response time $t_{latency}$ with 800 browsers and $t_{net} = 8$ seconds for two cases:

- *bandwidth* =768 kb/s, with no Web cache.

- *bandwidth* =384 kb/s, a Web cache with hit rate 0.5.

Fig. 9 compares the CDF of the response time $t_{latency}$ with 800 browsers and $t_{net} = 2$ seconds for two cases:

- *bandwidth* =768 kb/s, with no Web cache.

- *bandwidth* =384 kb/s, a Web cache with hit rate 0.5.

The same behavior as that in Fig. 5 is observed in Fig. 8 and Fig. 9.

Fig. 10 compares the CDF with 500 browsers and $t_{net} = 4$ seconds for two cases: the hit is bursty or random. It is seen that the difference is significant. Therefore the hit pattern is an important index for network planning. The random hit assumption underestimates the bandwidth required for network planning.

Fig. 11 compares the CDF of the response time $t_{latency}$ with 500 browsers, 512kb/s access link and $t_{net} = 4$ seconds for hit rates equal to 0.3, 0.42 and 0.54 respectively. Fig. 12 compares the corresponding probability density function. It can be seen that increasing the hit rate changes the pattern of CDF significantly.

Fig. 13 presents the mean value of the response time versus hit rate with 500 browsers, 512kb/s access link and $t_{net} = 4$ seconds.

The model in this paper assume implicitly that there is no congestion in the Internet. If there is congestion in the Internet, obviously, the bottleneck will be in the Internet in stead of ISP's exit/entrance. In this case, increasing the bandwidth of the access link can not alleviate the congestion effectively, and caching is more effective in sense of decreasing the response time.

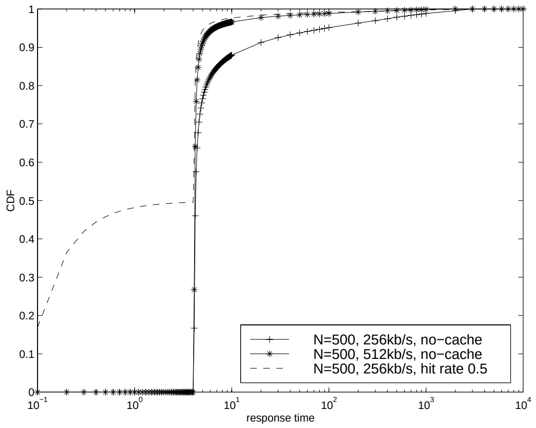


Figure 5: CDF of Response Time $t_{latency}$ (seconds)

4 Conclusions and Future Work

The simulation results presented in this paper are very encouraging and useful to the ISPs, enterprises and universities that are planning to interconnect their subnets with the Internet. However, the simulation results depend on the accuracy of traffic model presented in [16]. A more accurate model to study the effect of Web caching on the network planning is the combination of trace-driven simulation in Fig. 2 and the distribution-driven simulation of Fig. 4, and illustrated in Fig. 14, where the real traces are used instead of the empirical traffic model. This model will be studied in the future.

References

[1] M. A.brams, C. R. Standridge, G. Abdulla, S. Williams, and E. A. Fox, "Caching Proxies: Limitations and Potentials", http:// ei.cs.vt.edu/ succeed/WWW4/ WWW4.html.

[2] M. A.brams, C. R. Standridge, G. Abdulla, S. Williams, and E. A. Fox, "Removal

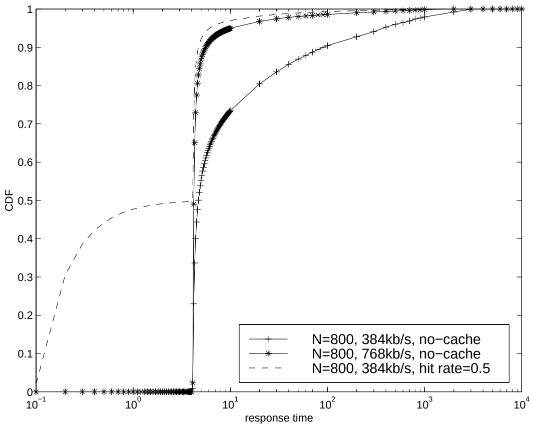


Figure 6: CDF of Response Time $t_{latency}$ (seconds)

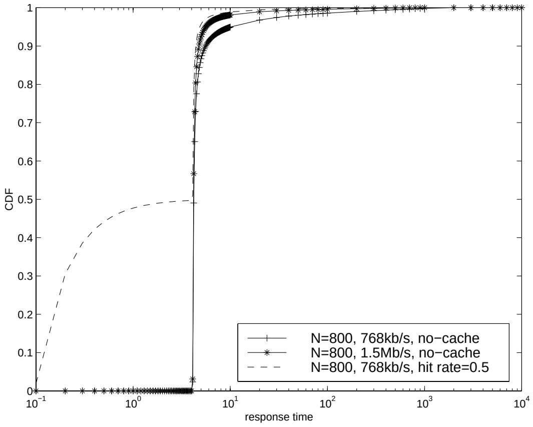


Figure 7: CDF of Response Time $t_{latency}$ (seconds)

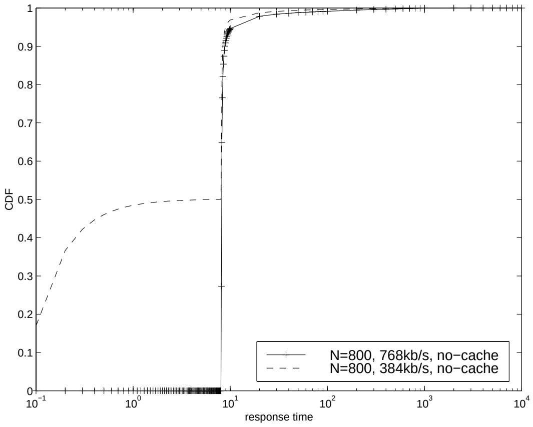


Figure 8: CDF of Response Time $t_{latency}$ (seconds), $t_{net} = 8$ seconds

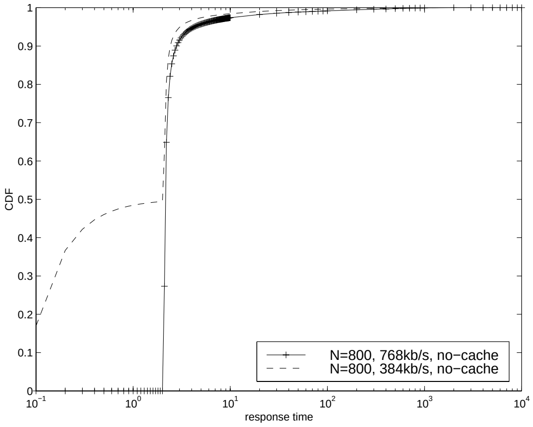
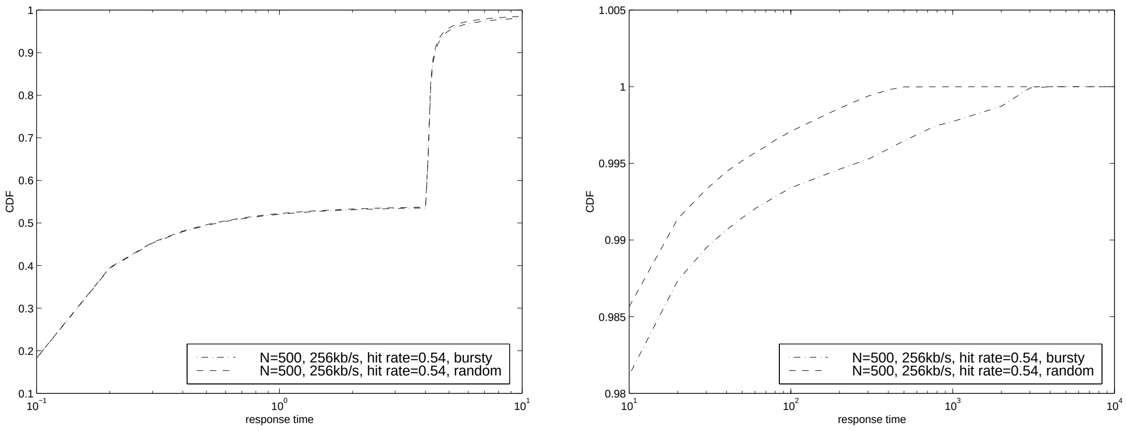


Figure 9: CDF of Response Time $t_{latency}$ (seconds),$t_{net} = 2$ seconds



(a) 0-10 seconds

(b) 10- 10000 seconds

Figure 10: CDF of Response Time $t_{latency}$ (seconds), Comparision of bursty hit and random hit

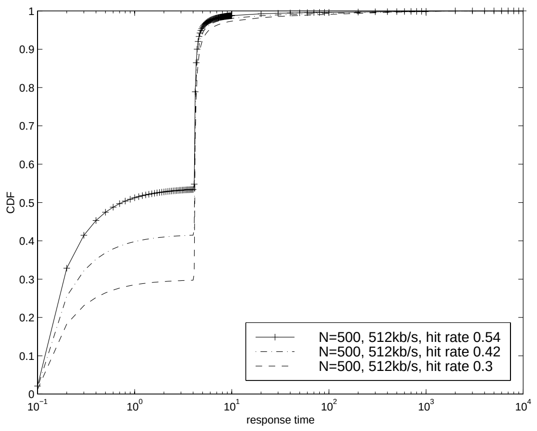


Figure 11: CDF of Response Time $t_{latency}$ (seconds)

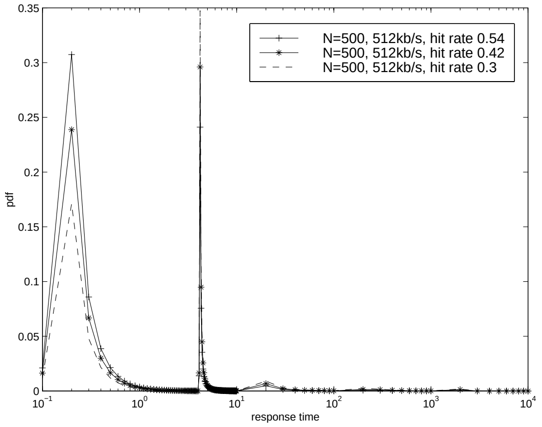


Figure 12: pdf of Response Time $t_{latency}$ (seconds)

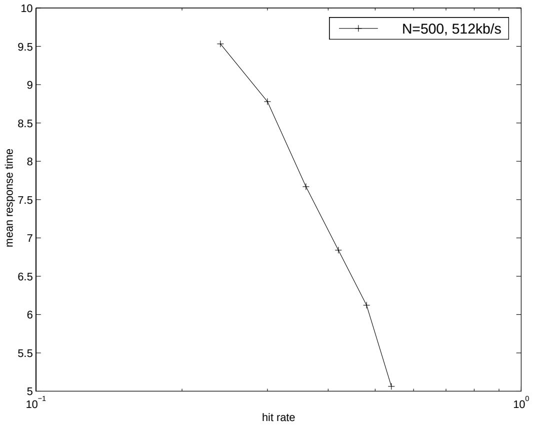


Figure 13: The Mean Response Time (seconds) vs. Hit Rate

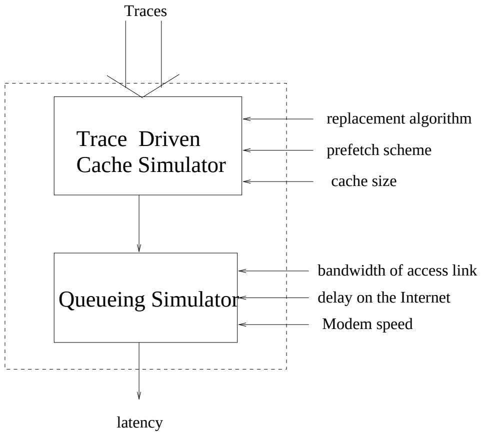


Figure 14: The More Accurate Model for Further Study

Policies in Network Caches for World-Wide Web Documents. SIGCOMM'96, pp293-309.

[3] P. Lorenzetti, L. Rizzo, L.Vicisano, "Replacement Policies for a Proxy Cache", http://www.iet.unipi.it/ luigi/caching.ps

[4] I. Tatarinov, "Cache Policies for Web Servers", http:// ncstrl.cs.cornell.edu:80 /Dienst/ UI/1.0/Display/ncstrl.ndsu_cs/NDSU-CSOR-TR-97-05

[5] I. Tatarinov, "Performance Analysis of Cache Policies for Web Servers", http:// ncstrl.cs.cornell.edu:80/Dienst/UI/1.0/ Display/ncstrl.ndsu_cs/NDSU-CSOR-TR-97-06

[6] P. Cao and S. Irani, " Cost-Aware WWW Proxy Caching Algorithms", http:// ncstrl.cs.cornell.edu:80/Dienst/UI/1.0/Display /ncstrl.uwmadison/CS-TR-97-1343

[7] Z. Jiang and L.Kleinrock, "An Adaptive Network Prefetch Scheme", *IEEE J-SAC*, April, pp358-368, 1998.

[8] V. N. Padmanabhan and J. C. Mogul, “Using Predictive Prefetching to Improve World Wide Web Latency”, ACM, Computer Communication Review, pp22-36.

[9] E. P. Markatos and C. E. Chronaki, “ A Top-10 Approach to Prefetching on the Web”, http://www.ics.firth.gr/proj/arch-vlsi/www.html

[10] L. Breslau, P. Cao, L. Fan, G. Phillips and S. Shenker, “Web Caching and Zipf-like Distributions: Evidence and Implications”, http://www.cs.wisc.edu/ cao/papers/zipf-implications.html

[11] Martin Arlitt, Rich Friedrich, Tai Jin, “Performance Evaluation of Web Proxy Cache Replacement Policies”, Computer Performance Evaluation: Modelling Techniques and Tools, *LNCS*, 1469. p193-206, 1998.

[12] V. Almeida, A. Bestavros, M. Crovella and A. Oliveira, “Characterizing Reference Locality in the WWW”, http://ncstrl.cs.cornell.edu:80/Dienst/UI /1.0 /Display /ncstrl.bu_cs/96-011

[13] M. Crovella, A. Bestavros, “ Explaining World Wide Web Traffic Self-Similarity”, http://ncstrl.cs.cornell.edu: 80/Dienst/UI/ 1.0/Display/ ncstrl.bu_cs/95-015

[14] G. Abdulla, E. A. Fox, M. Abrams and S. Williams, “WWW Proxy Traffic Characterization with Application to Caching”, http://ncstrl.cs.cornell.edu:80 /Dienst/UI /1.0/Display/ncstrl.vatech_cs/TR-97-03

[15] V. Almeida and A. Oliveira, “On the Fractal Nature of WWW and Its Application to Cache Modeling”, http://ncstrl.cs.cornell.edu:80/Dienst/UI /1.0/Display/ncstrl.bu_cs/96-004

[16] S. Deng, "Empirical Model of WWW Document Arrivals at Access Link", *ICC'96*, pp1797-1802.

[17] B. M. Duska, D. Marwood, M. J.Freeley, "The Measured Access Characteristics of World-Wide-Web Client Proxy Caches", University of British Columbia, Technical Report.

[18] K. W. Ross, "Hash Routing for Collections of Shared Web Caches", *IEEE Network*, pp37-44, 1997

[19] D. Wessels, K. Claffy, "Internet Cache Protocol (ICP), version 2", RFC2186

[20] D. Wessels, K. Claffy, "Application of Internet Cache Protocol (ICP), version 2", RFC2187

[21] V.Valloppillil, "Cache Array Routing Protocol v1.0", INTERNET-DRAFT, ¡draft-vinod-carp-v1-03.txt¿

[22] J. R. Spirn, "Program Behavior: Models and Measurements", *Elsevier, NewYork*, 1977.

[23] http://www.cs.berkeley.edu/ gribble/traces/index.html, UC-Berkeley's trace.

[24] ftp://ftp.digital.com/pub/DEC/traces/proxy/webtraces.html, DEC's trace.

[25] ftp://ei.cs.vt.edu/pub/succeed/Sigcomm96/, Virginia Tech's Traces

[26] Herb Schwetman, "CSIM Users' Guide", Microeletronics and Computer Technology Corporation, 1991.