Understanding the Performance of Cooperative Web Caching Systems

Xiaosong Hu
Faculty of Computer Science, Dalhousie University
xiaosong@cs.dal.ca

A. Nur Zincir-Heywood
Faculty of Computer Science, Dalhousie University
zincir@cs.dal.ca

Abstract

Web caching has been recognized as an effective scheme to alleviate the service bottleneck and reduce the network traffic, thereby minimizing the user access latency on the Internet. To maximize the performance of caching, cache cooperation systems such as Hierarchical, Distributed and Hybrid are employed. In this work, we first derive an analytical model to calculate the performance parameters for the aforementioned three caching systems. We then test these systems on a web cache simulator. Results show that the hybrid system is very competitive in terms of the hit ratio and the bandwidth usage compared to the distributed system and hierarchical system, respectively. On the other hand, the hybrid system gives the optimal results in terms of latency and the number of hops.

1. Introduction

With the explosive growth of the World Wide Web which can be considered as a large distributed information system, the web becomes the dominant application on the Internet. Although the internet backbone capacity increases significantly each year, the demand for bandwidth still outstrips the supply as more and more information services are moved onto the Web. Actually, some of today's Internet users are suffering two major problems: congestion and server overloading. One possible way to alleviate these problems is Web Caching.

In fact, web caching is not a new technique since browser caching and proxy caching have been used widely. Researchers work to combine the single proxy cache into a cooperative caching system to improve the Internet performance further. Hierarchical [1], distributed [2, 3] and hybrid [4] systems are some examples of the-state-of-the-art cooperative caching mechanisms.

In hierarchical caching, there are three levels of caches: institutional, regional and national levels [5]. We can treat a regional cache as the parent of some institutional caches and the national cache as the parent of some regional caches. A client can be directly connected to any one of these caches, which then becomes the default cache for the client. When a request is not satisfied by the default cache, it is redirected to the parent cache and the parent cache can in turn forward its unsatisfied requests to its parent cache. If the document is not found at any cache level, the national cache contacts the original server directly. When the document is found, either at a cache or at the original server, it travels down the hierarchy, and each of the intermediate caches along its path makes the decision whether a copy of the document should be cached locally or not, based on the cache content update algorithm used.

In distributed caching, there are no intermediate caches other than the institutional caches, which cooperate to serve each others' misses. In order to decide from which institutional cache to retrieve a miss document, institutional caches need other mechanisms to share the documents they contain. Some of these mechanisms are Inter Cache Protocol ICP [6], Cache Array Routing Protocol [7], Summary Cache [8] and Cache Digest [9].

On the other hand, with hybrid caching, caches may cooperate with other caches at the same level or at a higher level using distributed caching so that the document is fetched from a parent/neighbor cache that has the lowest round trip time (RTT).

In this work, we first develop an analytical model to analyze and compare the performance of the three caching systems. We try to make our assumptions and parameters as reasonable as possible using the current literature. Then, we derive models to calculate the hit ratio, the average number of hops, and the latency experienced by the clients. Furthermore, to confirm the model is reasonable, we carry out several simulations based on the network used by the analytical model.

In the following, section 2 introduces the structure of the analytical model and deduces the formulae to calculate the parameters used to evaluate the performance of the three caching systems. The results of trace-driven simulations are then given in section 3 to analyze the reliability of this analytical model. Finally, conclusions are drawn and future work is discussed in section 4.

2. The analytical model

As Fig-1 shows, we first model the underlying network system topology as a full *O*-ary tree. In this

model, assume O represents the nodal outdegree of the tree, h be the number of hops between two neighbor level networks (assuming the distance of two neighbor routers is one hop), z is the number of links between the national network and the original source server, and l is the level of the tree such that $0 \le l \le 2h+z$, where $l = 0$ is the institutional network and $l = 2h+z$ is the original source. Let D_I, D_R and D_N be the transmission rates of the links at the institutional, regional, and national networks and D be the transmission rate of the links from the national network to the source.

Moreover, it is assumed that the institutional caches are associated with the institutional network, the regional cache with the regional networks and the national cache with the national networks. This in return provides a system to develop the three aforementioned cooperative caching systems.

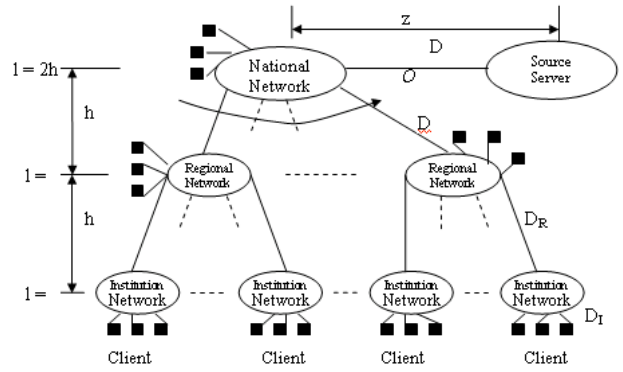

Fig-1: Topology of the network

2.1. Model description

Denote N as the total number of different documents requested and $P_N(i)$ as the conditional probability given the arrival of a request made for document i [10]. Assuming all the documents are ranked in order of their popularity where document i is the i'th most popular document, $P_N(i)$ can be defined as $P_N(i) = \Omega / i^{\alpha}$ where α is a constant that determines how skewed the Zipf distribution is and its range is $0 < \alpha \le 1$. Ω is given by $\Omega = \left[\sum_{i=1}^{N} \frac{1}{i^{\alpha}} \right]^{-1}$. Let S be the average document size. As [10] shows, the document size has no strong correlation with its popularity. With this in mind, we can denote the cache size as the number of documents cached. Thus, let C_q indicate the cache size as the number of documents stored, where $q \in (i, r, n)$ and i, r, n are institutional, regional and national caches, respectively. Since the web page contents are always modified over some time, the cached pages become invalid if the old versions are stored. To this end, Δ represents the average available time for a cached document.

2.2. Hit ratio analysis

To make the hit ratio as high as possible, we need to keep the most popular pages in the cache. This can be fulfilled with the LFU (Least Frequently Used) replacement algorithm. Here LFU is used but it is easy to perform derivations. Let H_q be the hit ratio gained at cache q over all its requests. For a given cache q, the most popular C_q documents are cached. If a request is for a document in C_q and its interval is less than Δ, it is a hit. Therefore,

$$H_q = \sum_{i=1}^{C_q} (P_N(i) \cdot P_q) \qquad (1)$$

where P_q is the probability that there is a request for document i within Δ at cache q. To calculate P_q, let δ as the time into the interval $[0, \Delta]$ at which a request occurs, and $P_{(q|\delta)}$ is the probability that there is a request for document i during the interval $[0, \delta]$ at cache q. Then

$$P_q = \frac{1}{\Delta} \int_0^{\Delta} P_{(q|\delta)} \cdot d\delta \quad \text{where } P_{(q|\delta)} = 1 - e^{-\lambda_{q,i} \cdot \delta}$$

where $\lambda_{q,i}$ is the average request rate for document i at cache q and $\lambda_{q,i} = \lambda_q(\Omega / i^{\alpha})$[11]. Thus,

$$H_q = (C_q / N)^{1-\alpha} - \sum_{i=1}^{C_q} \frac{1}{\lambda \lambda_q} (1 - e^{-\lambda \lambda_q \Omega / i^{\alpha}}) \qquad (2)$$

The next issue is how to solve λ_q. Assuming that the part of arrival rate λ_q from the clients directly connected to it is λ, we then determine that the aggregate request arrival rates at every network level l are λ_l^c, λ_l^d and λ_l^h for the hierarchical, distributed, and hybrid systems, respectively.

For the hierarchical system, the request rate at the levels where caches are located is the sum of λ and the part filtered by the hit ratio at the lower caches. At the other levels, it is just the part filtered by the hit ratio at the lower caches. Thus the aggregate request arrival rate generated by the hierarchical system at a link between the levels l and l+1 is given by

$$\lambda_l^c = \begin{cases} O^l \lambda_i (1 - H_{q=i}) & 0 \le l < h \\ O^{(l-h)} \lambda_r (1 - H_{q=r}) & h \le l < 2h \\ \lambda_n (1 - H_{q=n}) & 2h \le l < 2h + z \end{cases}$$

where λ_r and λ_n are the request rate at the regional and national levels respectively.

As for the distributed system, the aggregate request arrival rate at a link between levels l and l+1 includes two parts. The first part is filtered by the documents already hit in any institutional cache belonging to the subtree rooted at level, H_l. The other part is generated by requests from the institutional caches out of the subtree and satisfied by the institutional caches in the subtree. Therefore, the request rate between levels l and l+1 in distributed system is given by

$$\lambda_l^d = \begin{cases} O^l \lambda_0((1-H_l) + (H_{l=n} - H_l)) & 0 \le l < 2h \\ O^{2h} \lambda_0(1-H_{l=n}) & 2h \le l < 2h+z \end{cases}$$

Finally, for the hybrid system, we configure the sibling caches into groups so that the caches can visit their group members as in the distributed system. Out of the groups, a cache can visit the higher-level cache as the hierarchical system. Therefore, the request rate for hybrid system is very similar to the hierarchical system except the links just above the cache levels. For the links above the cache levels, the request rate is similar to the distributed system. Thus, the request rate between levels l and l+1 in the hybrid system is given by

$$\lambda_l^h = \begin{cases} \lambda_0(1+H_{q=i})(O-2)/O & l = 0 \\ O^l \lambda_0(1-H_{q=i}) & 1 \le l < h \\ \lambda_0(1+H_{q=r})(O-2)/O & l = h \\ O^{l-h} \lambda_0(1-H_{q=r}) & h < l < 2h \\ \lambda_0(1-H_{q=n}) & 2h \le l < 2h+z \end{cases}$$

After deducing the formulae for calculating the hit ratio at a single cache, we now focus on how to calculate the hit ratio over a cooperative caching system. For hierarchical and hybrid system, the requests from different levels are with different hit ratios since some can visit more caches than the others.

Let H_{pq}^C and H_{pq}^H be the average hit ratio achieved at level q for the requests from cache p for the hierarchical and hybrid system respectively: $p,q \in (i, r, n)$ *where i, r and n* is the institutional, regional and national caches in the network system, respectively. H_{pq}^C and H_{pq}^H can be calculated with equation (2). Let H_p^C and H_p^H be the average hit ratio for requests from cache p for the hierarchical and the hybrid system respectively. Thus, H_p^C is the sum of H_{pq}^C where q equals *i, r* and *n* respectively. We can solve H_p^H in the same way. Finally, the average hit ratio for the hierarchical and the hybrid system, H^C and H^H respectively, is the weighted average of hit ratios for requests from institutional, regional and national caches. Thus,

$$H^C = (O^{2h} H_{p=i}^C + O^h H_{p=r}^C + H_{p=n}^C)/(O^{2h}+O^h+1)$$

$$H^H = (O^{2h} H_{p=i}^H + O^h H_{p=r}^H + H_{p=n}^H)/(O^{2h}+O^h+1)$$

On the other hand, the hit ratio over the distributed system can be calculated by modifying equation (2), since caches in the distributed system are only at the institutional level and they can visit all other caches, if the request is missed locally. Thus, all caches become an integrated large cache q and the hit ratio for the distributed system H^D becomes:

$$H^D = (O^{2h} C_i / N)^{1-\alpha} - \sum_{i=1}^{O^{2h} C_i} \frac{(1 - e^{-\alpha O^{2h} \lambda_I / i!})}{\alpha O^{2h} \lambda_I}$$

2.3. Hops

The number of hops is defined as the expected number of links traversed to retrieve a document. For requests from one level, the average number of hops is the sum of the products of hops from a cache to a client and the probability of the request accessing that cache. Let $B_C(l)$ and $B_H(l)$ be the average hops of the hierarchical and the hybrid system for the requests from level l, $l \in (0, h, 2h)$, respectively. The equations for $B_C(l)$ and $B_H(l)$ can be found in [11].

Let B_C and B_H be the weighted average of hops for requests from different levels for the hierarchical and the hybrid systems. B_C and B_H can then be calculated as the following:

$$B_C = (O^{2h} B_C(l=0) + O^h B_C(l=h) + B_C(l=2h))/(O^{2h}+O^h+1)$$

$$B_H = (O^{2h} B_H(l=0) + O^h B_H(l=h) + B_H(l=2h))/(O^{2h}+O^h+1)$$

Since requests on the distributed system can only be generated at the institutional level, average hops, B_D, can therefore be calculated directly as the following:

$$B_D = H^D/O^{2h} + \sum_{l=1}^{2h} \frac{(O-1)H^D}{O^{2h-l+1}}(2l+1) + (1-H^D)(2h+z+1)$$

2.4. Latency analysis

Latency is referred as the time for a request to be satisfied. It depends on: the number of hops, transmission delay and queue delay, where the queue delay depends on the request rate and the processing capacity. For the latter two, the M/M/1 queuing model is used to do the analysis.

The number of hops depends on the number of network links from the client to the cache and the probability that the cache may contain the valid copy of this document. The latter is the hit ratio at that cache. If we use TCP connections and d as the per-hop propagation delay, the connection time of per-hop is 4d due to the three-way handshake. Therefore, the connection time for the three systems is the product of 4d and the average hops for each system.

Let $E[T_T^k(l)]$ be the transmission delay of a link from level l to the level l-1 where $k \in (c, d, h)$ and c, d and h represent the hierarchical, the distributed and the hybrid systems, respectively. We neglect the time of transmitting the request since the size of the request itself is very small. The M/M/1 queuing theory [12] gives $\overline{W} = \frac{1}{\mu C - \lambda}$. Thus, to calculate the transmission delay, $1/\mu$ be the average document size; S, C be the link bandwidth and λ is the packet arrival rate. Then, the transmission delay of a link from level l to the level l-1 becomes:

$$E[T_T^k(l)] = \frac{S}{D_l \mu C_l^k S}$$

For the hierarchical and the hybrid system, to calculate the average transmission delay over the network,

we first need to calculate the transmission delay for the requests from the same level. Let $E[T_T^k](l)$ be the transmission time of the hierarchical ($k=c$), and the hybrid systems ($k=h$) for the requests from level l, $l \in (0, h, 2h)$, respectively. Let L_i be the network level. We define a general formula for calculating the mean transmission delay of the requests from level l as:

$$E[T_T^k](l) = \sum_{l=0,h,2h}^{2h} ((P(L_i >= l)E[T_T^k(l)]) + P(L_i = 2h+z)\cdot E[T_T^k(L_i = 2h+z)]$$

$P(L_i >= l)$ is the probability that a request has to move up higher than or equal to the level l in order to get a copy of the required document. $P(L_i=2h+z)$ is the probability that a request has to move to the source server in order to get a copy of the required document. Since we have deduced the formulae for hit ratio at cache q for requests from level p H_{pq}^C and H_{pq}^H in section 2.2, $P(L_i=2h+z)$ is the miss ratio of the requests from level l and $P(L_i >= l)$ is the miss ratio at level L_i for requests from level l. The over all average transmission delay $E[T_T^k]$, for the hierarchical and the hybrid system is therefore the weighted average of the transmission delay of requests from different levels.

$$E[T_T^k] = \frac{O^{2h}E[T_T^k](l=0)}{O^{2h}+O^h+1} + \frac{O^h E[T_T^k](l=h)}{O^{2h}+O^h+1} + \frac{E[T_T^k](l=2h)}{O^{2h}+O^h+1} \quad \text{where } k\_(c, h)$$

For the distributed system, since all requests are from the institutional level, its transmission delay can be given as the delay from the level l in the hierarchical system with the following equation

$$E[T_T^k] = \sum_{l=0}^{2h} (P(L_i >= l)E[T_T^k(l)] + P(L_i=2h+z)E[T_T^k(Li=2h+z)] + \sum_{l'=0}^{2h-1} ((P(L_i <= l')E[T_T^k(l')]) \quad \text{where } k = d$$

However, compared with the hierarchical and the hybrid systems, the distributed system has one additional part $\sum_{l'=0}^{2h-1} ((P(L_i <= l')E[T_T^D(l')])$, because requests have to travel down to a cooperative institutional cache, and therefore, responses have an additional (moving-up) transmission delay.

2.5. Visualization of the analytical model

To quantitatively compare and contrast the performance of the three caching systems, we use MATLAB to plot out the figures for hit ratio, hops, traffic and latency of the three caching systems based on the aforementioned equations and the parameters in table-1. The results are shown in [11]. We are not intended to show them here because of the space limitation.

3. Simulation results

To analyze the reliability of the analytic model, we develop a web cache system simulator. Because of page limitations, details of the simulator can be found in [11]. The data set is the captured log file from the proxy of Faculty of Computer Science, Dalhousie University. We extract the successful two million "get" requests as records of our data set. For each record, we extract the URL and the document size. The URL includes the path and document name so that the document can be uniquely identified. The data set is used on the topology given in Fig-1 and the performances of the three caching systems are compared according to hit ratios, the number of hops, traffic generated on the networks and the latency.

Table-1: Parameters used for the model and the simulation

Parameter name	value
Nodal outdegree of the tree (O)	3
Hops between neighbor caches (h)	2
Hops from the root to the source (z)	10 [13]
Bandwidth at institution level (D_I)	1 Mb/sec
Bandwidth at regional level (D_R)	5 Mb/sec
Bandwidth at national level (D_N)	10 Mb/sec
Average propagation delay per hop	0.004 sec
Bandwidth to the source (D)	10 Mb/sec
Average propagation delay per hop	0.004 sec
Average document size (S)	10KB[14]
Request rate from the clients (i)	2 req/sec
Total document number	1 million
Document update time (Δ)	12h
Skew factor of Zipf distribution	0.64 [10]
Average propagation delay per hop	0.004 sec

3.1. Hit ratio

As shown in Fig-2, the value of hit ratios at different cache sizes increase logarithmically or as a small power as a function of cache size. With the increase of the cache size from 0.5 to 10 percent, the hit ratio increase from 0.8 to 0.9 for the distributed system, 0.46 to 0.7 for the hierarchical system and 0.57 to 0.8 for the hybrid system. Similar to the analytical model, simulation results show that the distributed system has the highest hit ratio whereas the hierarchical system has the lowest.

3.2. Hops

The average number of hops to retrieve the required documents is shown in Fig-3. Again, the results are compatible with the analytical model. With the increase of cache size from 0.5 to 10 percent, the average number of hops decrease from 9.5 to 9 for the distributed system, 9 to 5.2 for the hierarchical system and 8 to 5.3 for the hybrid system. The major difference between results from the simulation and the analytical model is when the cache size is around 0.5 percent, the average number of hops with the hierarchical system is very close to and may exceed the distributed system. We think this difference is caused by the small cache size.

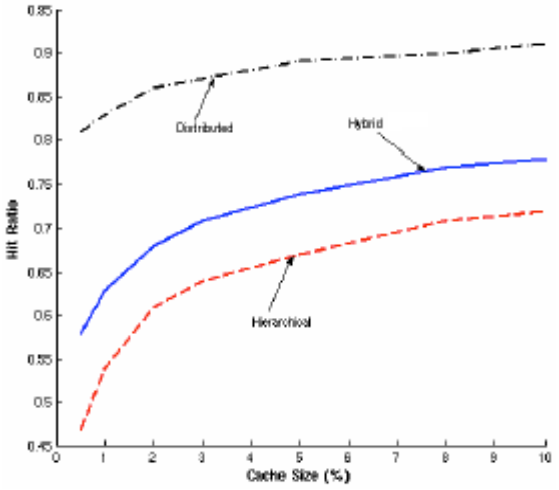

Fig-2: The hit ratio of the three cache systems

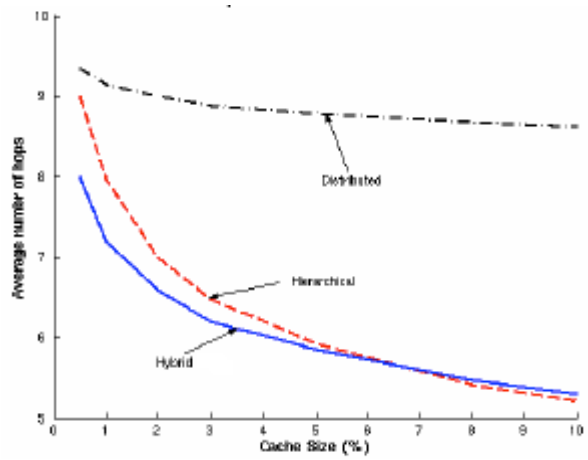

Fig-3: The average number of hops to satisfy requests

3.3. Traffic

As shown in Fig-4, traffic on the link to the source is much higher than the other links for the hierarchical and the hybrid systems. This confirms that this link is the bottleneck. For the distributed system, the traffic on the links at the national level can be higher than the traffic on the link to the source. Again, simulation results are similar to the analytical model.

3.4. Latency

Fig-5 shows the average latency for the three caching systems. Compared with the analytical model, simulation results for the distributed system are a little higher, especially when the data set is small. For the hierarchical and the hybrid system, simulation results are also a little higher when the cache size is small. However, with the increase of the cache size, simulation results are more and more close to the analytical results and when the cache size is 10 percent, results are almost identical.

Among the three systems, simulation results, similar to the analytical results, also show that the distributed system has the highest latency whereas the hybrid system has the lowest.

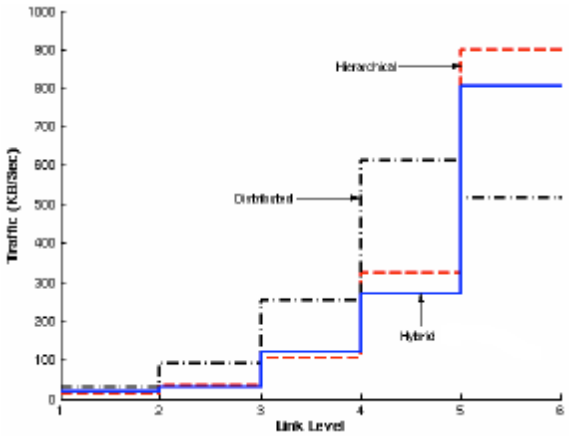

Fig-4: Average traffic on the three cache systems

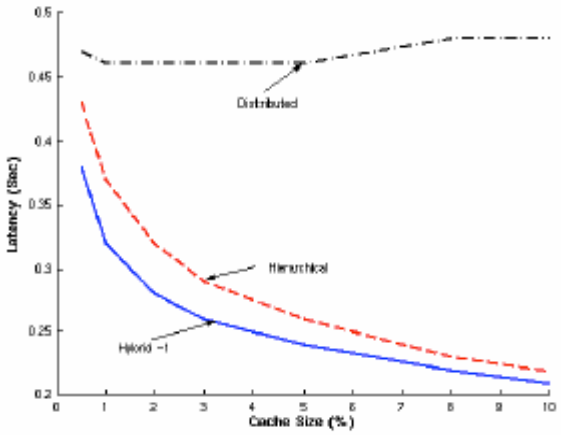

Fig-5: Latency of the three caching systems

4. Conclusion

As it can be seen in section 3, the simulation and analytical model give similar trends in terms of performance factors: hit ratios, number of hops, traffic generated and latency. Based on the results obtained both from the analytical model and simulations, the hit ratio of the distributed system is very high even when the size of a caching sever is very small. This benefit should be attributed to its mechanism, i.e. no redundant copies and all caches work as a whole. The hit ratio of the hierarchical system is low if the size of the cache server is small. As the size of a cache increases, the hit ratio also increases quickly. The hit ratio of the hybrid system is moderate among the three systems. Although the increases in hit ratios of the three systems are different, they all show a logarithmical function or a function with a small power to the cache size.

On the other hand, although the hit ratio with the distributed system is high, hits are uniformly distributed over all the caches on the network. In other words, cached documents are usually far from clients and thus, the average number of hops to satisfy the clients' requests are greater than those of the hierarchical and the hybrid systems except when the cache size is very small. It should also be noted that the hybrid system gives the best possible results among the three systems since it can always keep the average hops at a relatively low level.

As for traffic, the distributed system generates more traffic at the lower network levels. However, as discussed above, because of its high hit ratio, the distributed system could be with much lower traffic to the source server than the hierarchical and the hybrid systems. In contrast, we find the hybrid system is still very competitive.

To most WWW service users, the most important performance indicator is to be able to retrieve the required document as quickly as possible. From this view point, the hybrid system has the best result among the three systems since it can keep the latency always lower than the other two.

References

[1] A. Chankhunthod *et al.*, "A hierarchical internet object cache," in *Proc. 1996 USENIX Technical Conf.,* San Diego, CA, Jan. 1996

[2] D. Povey and J. Harrison, "A distributed Internet cache," in *Proc. 20th Australian Computer Science Conf.*, Sydney, Australia, Feb. 1997.

[3] R. Tewari, M. Dahlin, H. M. Dalin, H. Vin, and J. Kay, "Beyond hierarchies: Design considerations for distributed caching on the Internet", in *Proc. ICDCS'99 Conf.*, Austin, TX, May 1999.

[4] M. Rabinovich, J. Chase, and S. Gadde, "Not all hits are created equal: cooperative proxy caching over a wide-area network", *Computer Networks and ISDN Systems 30, 22-23, pp. 2253-2259*, Nov. 1998.

[5] P. Rodriguez, C. Spanner and W. Biersack, "Analysis of Web Caching Architectures: Hierarchical and Distributed Caching", in *Proc. of the 4th International WWW Caching Workshop*, 1999

[6] D. Wessels and K. Claffy, "ICP and the Squid Web Cache", *IEEE journal on selected areas in communicational, Vol. 16, No. 3*, April 1998

[7] V. Valloppillil and K. W. Ross, "Cache array routing protocol v1.0, Internet Draft <draft-vinod-carp-v1-03.txt>".

[8] L. Fan, P. Cao, J. Almeida, and A. Z. Broder, "Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol", *IEEE/ACM Transactions on Networking, Vol. 8, No. 3*, June 2000.

[9] A. Rousskov and D. Wessels, "Cache Digest", in *Proceedings of 3rd International WWW Conference*, Santa Clara, April 1997.

[10] L. Breslau, P. Cao, L. Fan, G. Phillips and S. Shenker, "Web Cacheing and Zipf-like Distributions: Evidence and implications", in *Proc. of the IEEE Conference on Computer Communications (INFOCOM'99)*, May 1999

[11] X. Hu, "Understanding the Performance of the Cooperative Web Caching Systems with Analysis Models and Simulations", MCS thesis, Faculty of Computer Science, Dalhousie University, Halifax, NS, Canada, http://www.cs.dal.ca/~xiaosong.

[12] OPNET tutorial manual: "M/M/1 Queue Tutorial", http://web.nps.navy.mil/~mceachen/ec3850/computer/comp1tut.pdf, retrieved in June 2003

[13] "Assessing average hop count of a wide area Internet packet", http://www.nlanr.net/NA/Learn/wingspan.html, retrieved in Jun 2003

[14] "Web statistics: size, the average page", http://www.clienthelpdesk.com/statistics_research/web_statistics.html, retrieved May, 2003