

# Network Centric Improvements to Resource Caching

R. Zuck, A. Williams, B. Kair, H. Bui, C. Stringfellow, N. Passos

Department of Computer Science  
Midwestern State University  
Wichita Falls, TX 76310, U.S.A.

**Abstract**—Network bandwidth is a precious commodity in the information age. One would be hard-pressed to cite an example of an industry, hobby, or effort that could not benefit, in some way, from a communication network. The ability to minimize the allocation time for communication channels within a network is one of the primary concerns of the network administrator. Network resource caching is one technique that may be used to reduce channel allocation time. Therefore, improvements to the standard cache management algorithms are an important component of achieving this goal. This study demonstrates the potential benefit of the Time Distance Indexing Factor (TDIF) cache replacement algorithm over a traditional technique, Least Recently Used (LRU), in reducing communication channel allocation time. The benefit of this technique translates into an increase in the effective bandwidth of the network.

**Keywords**- *cache replacement; cache management; queue; communication channel; QoS; web page.*

## I. INTRODUCTION

The Internet is considered an immense source of information, and the World Wide Web has been accumulating knowledge for years. The growth of online entertainment and the significant increase of streaming data squeeze network bandwidth to its limits. People using the Internet to shop, listen to music, or watch video, may find these bandwidth limitations frustrating. Websites and media files can take a long time to download. Web caching can play an important role in reducing network bandwidth requirements and transmission delays.

Web caches temporarily store network resources such as HTML pages, media files, and the like for future use. Rather than accessing the original server multiple times, users can retrieve network resources faster through a cache server. Moreover, any user with access to the cache server may benefit from the use of previously cached resources. In general, web browsers have a built in local cache that benefits the local user of that browser. Members of wired and wireless networks may also benefit from the use of a centralized cache from which they have access. Such a cache may contain a more diverse set of resources than that of a local cache.

Most of the web cache replacement policies utilize a user centric design approach, which attempts to minimize a users wait time for resource retrieval. This paper proposes a web cache replacement policy that utilizes a network centric design approach to increase effective network bandwidth.

## II. BACKGROUND

The Internet is composed of many interconnected computer networks. Each network may link tens, hundreds, or even thousands of computers, enabling them to share information with one another and to share computational resources such as powerful supercomputers and databases of information [5]. Since its advent, and after realizing its potential and almost limitless uses, people's demand for resources from the Internet grew enormously. Network bandwidth, defined as the carrying capacity of a circuit, usually measured in bits per second for digital circuits, or hertz for analog circuits [5], is limited. Networks that provide resources, such as web pages, audio and video, and other documents, have to be continuously upgraded or deny service to users. Unfortunately, continuously upgrading networks to handle large data requests is expensive and often impractical, and as a result, network administrators turned to alternative techniques for providing users the data they requested while appeasing the strain on their networks. One of the techniques developed and implemented to reduce network strain was web caching, or internet resource caching, which is a way to store requested Internet resources on a server closer to the requesting site than to the source [5]. Storing resources closer should be understood as making the data available at a location to be accessed faster than making a request to the original server.

### A. Web Caching

Internet resource caching architectures consist of a temporal locality between two networks, where one of them is usually the Internet. Figure 1 depicts one possible relationship of network components. Temporary locality is often achieved by a proxy server, which is used as a buffer or to forward traffic between two networks forming a proxy cache.

Web caching serves as an excellent method of suppressing congestion and freeing bandwidth. As a result, network administrators, concerned with their users' quality of service (QoS), have shown interest in implementing cache servers within their networks. QoS describes how the overall transmission quality, speed, and reliability improve as data transmission, error, and missing data packet rates are measured and then modified to eliminate problems [5].

QoS is important to broadcast media and telecommunication companies, because often the amount of distortion that occurs when data is transmitted to a client determines whether that data is useful or not after transmission.

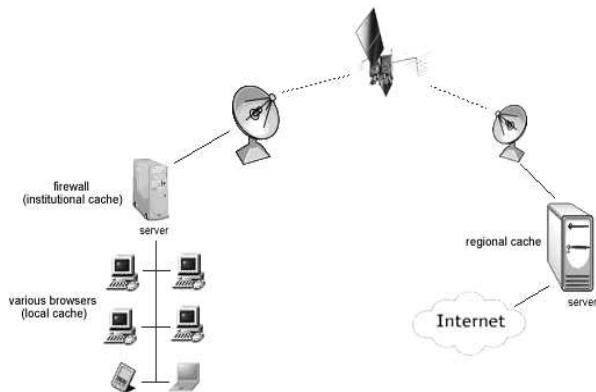


Figure 1. An example of a typical Internet topology that contains several types of caches.

To prevent frequent and unnecessary requests for the same resources, which can be time costly and reduce the quality of service, the proxy server, acting as a cache system, frequently stores resources that pass through it. There are two types of caches: simple and co-operative caches [6]. With simple caches, each time a request is made through the browser, a TCP connection is made to the cache server instead of making a connection to the original server [6]. Unlike simple caching, where one server exists at any particular caching level, co-operative caching architectures consist of multiple cache servers that are able to co-operate using, for example, Internet caching protocol (ICP). ICP is a protocol specially designed to allow collaboration among caches on the same hierarchical level [6].

When a user within a caching architecture makes a request for a resource using a web browser, the request is sent to a proxy server [5]. If the resource does not exist in the cache, then the proxy server makes a request to the server that contains the requested resource. Hyper Text Transport Protocol (HTTP) requests usually contain a URL (Universal Resource Locator). If the resource exists and no error has occurred, the request is sent to the client along with its header information. Header information can contain the TTL (Time to Live) or expiration time information, the rating of the content, and the type of resource it is, which the operating system can use to determine which program is used to open or view it. TTL is the time used to determine whether the resource is considered up-to-date. When the resource expires, it should no longer be kept in the cache, but instead a request to the original server for a new copy of the resource should be made. Whether or not the resource is stored in the cache usually depends on a LRU algorithm mechanism, the TTL, a threshold for the amount of resources that can be cached and other private network caching preferences or other policies. If a new user within the same caching structure requests the same resource, then the proxy server will simply send it to the new user, thus decreasing user latency.

QoS is a concern when serving resources especially to wireless devices such as cellular phones and PDA's. Outdoors antennas, whose signals suffer from attenuation, serve wireless devices. Users of wireless devices are also demanding smaller devices that transfer data at faster rates. As a result of

employing web caching techniques, the QoS of a network can be improved.

### B. Cache Replacement Algorithms

There is a wide range of techniques employed to develop efficient cache replacement strategies. Psounis and Prabhakar propose a random replacement strategy [13], while Segura-Devillechaise, Menaud, Muller, and Lawall propose a technique that attempts to efficiently prefetch resources linked to the user requested resource [14]. Many techniques are employed on individual cache servers, while others are applied to a hierarchy of cache servers.

Among the algorithms employed to individual caches are the traditional Least Recently Used (LRU) and Least Frequently Used (LFU). Each of these approaches is user centric in design [2]. They tend to emphasize the users resource requirements without addressing transmission time or network traffic considerations that are important measures of users QoS. QoS is not only concerned with cache hit miss ratios, but also with reducing server load. LRU and LFU sacrifice network bandwidth to attain an ideal hit miss ratio. Many web cache replacement policies have been studied and evaluated in an attempt to improve network QoS [1, 2, 3, 4, 7]. Williams and Abrams used the size of the web page as a factor for cache replacement [4]. Although they succeeded in improving the byte-hit rate (BHT), there is no guarantee of reducing the server's workload. Megiddo and Modha proposed an adaptive algorithm that utilizes both LRU and LFU techniques of cache replacement in varying proportions to achieve improved hit-miss ratios [11].

Several cache replacement techniques have been proposed that utilize some form of weighting factor in their replacement strategies. Kelly, Chan, Jamin and MacKie-Mason proposed a new algorithm called server-weighted LFU (swLFU), in which they assumed users were charged fees for accessing web resources. The swLFU took the value of servers into consideration [1, 3]. In their proposal, high-valued servers are favored over lower valued servers. As a result, swLFU helps to increase the value hit rate (VHR), however, BHT is decreased significantly, which may lead to an increase in transmission time. Hosseini-Khayat proposes a technique that links the costs of cache misses to the efficiency with which the cache space is utilized [9].

Cache replacement algorithms employed among distributed caches in a network offer unique challenges and benefits. Williamson employs a cache hierarchy structure [8]. Within such a structure, there exist multiple opportunities for resource retrieval at the various levels of the hierarchy. A related technique of employing distributed caches within static or ad hoc networks also appears beneficial [10, 12].

This paper proposes a new web caching replacement algorithm called Time Distance Indexing Factor (TDIF). The goal of the TDIF algorithm is to reduce network loading by favoring the retention of bandwidth costly resources, relative to those previously cached.

### III. ALGORITHM

The improvements to network QoS proposed in this work are achieved by employing a modified LRU cache replacement algorithm. The key component of the TDIF algorithm is a resource classification concept based on the time required for the network server to process a resource request made by the user. The process time is defined as the time required to retrieve a resource not present in the cache at the time of the request. A time threshold is established at the time of instantiating the algorithm that may be dynamically modified to classify cached resources as “Near” or “Far”. These descriptors are used to identify resources that can quickly be retrieved from the server (Near) and those that cannot (Far) relative to the time threshold. The goal of this modification is to make cache replacements that favor the retention of Far resources.

A further modification is required to prevent the replacement of newly requested Near resources while retaining old Far resources. A replacement pool of cached resources is defined as a small subset of all resources contained in the cache. The replacement pool (RP) is composed of only the oldest elements, Near or Far, in the cache. There are three means of cache replacement: 1) If the oldest Near resource is an element of the RP, it is replaced, 2) If there are no Near resources in the RP but the oldest Far resource is in it, it is replaced. 3) If there are no Near or Far resources in the RP, the algorithm degrades to LRU and the oldest resource, Near or Far, is replaced. A more rigorous description of the algorithm is presented below.

#### TDIF REPLACEMENT ALGORITHM

INITIAL CONDITIONS: A STRUCTURE FOR THE ALGORITHM THAT UTILIZES A PAIR OF QUEUES TO CONTAIN RESOURCE RECORDS CLASSIFIED AS NEAR OR FAR.

INPUT: INITIAL TIME THRESHOLD AND THE RP PROPORTION.

FOR EACH VALID RESOURCE REQUEST

IF THE RESOURCE IS A MEMBER OF THE CACHE

RETURN THE RESOURCE TO THE USER.

RE-QUEUE THE RESOURCE IN THE CACHE.

DETERMINE THE MEMBERS OF THE RP.

ELSE

RETRIEVE AND RETURN THE RESOURCE TO THE USER.

IF CACHE IS FULL

IF RP IS NOT EMPTY

IF SET OF NEAR RESOURCES IS NOT EMPTY

REMOVE THE OLDEST NEAR RESOURCE THAT IS A MEMBER OF THE RP.

ELSE

REMOVE THE OLDEST FAR RESOURCE THAT IS A MEMBER OF THE RP.

ELSE

REMOVE A RESOURCE VIA THE LRU ALGORITHM.

INSERT THE RESOURCE INTO THE CACHE.

DETERMINE THE MEMBERS OF THE RP.

### IV. SIMULATION

The simulation of a resource cache implemented with the TDIF algorithm was accomplished by utilizing a sample server log file of 36,712 successful user requests for web resources. There are 2,533 distinct users that requested each of the 6,657 different resources, contained in the log. Each of the resource requests was haphazardly assigned one of three channel capacities for simulation purposes: 1, 2, and 4 Kbps (kilobits per second) respectively. The product of channel capacity and resource size results in the time factor that is compared to the current TDIF time threshold and used to classify each resource within the TDIF cache.

Miss Time (MT), the time required by the cache server to recover a requested resource not present in the cache, is used in this study as a gauge of the effectiveness of a cache replacement algorithm in minimizing communication channel allocation on the network. The Total Miss Time (TMT) is simply the sum of all MT experienced by the cache server for a set of simulated user requests.

Fig. 2 contains the results of the simulation run for both the LRU and TDIF algorithms on caches of various sizes. In all cases, the size of the cache is defined as the number of resource records it contains. The plots of LRU data are provided to aid in the comparison of the two cache replacement techniques. The threshold times depicted in Figs. 2 and 3 represent the time used to classify a resource as Near or Far. Fig. 3 provides the results of the TDIF simulation run for a broader range of cache sizes. Fig. 4 contains the results of the comparison of the LRU and TDIF algorithms with respect to cache size.

### V. OBSERVATIONS

As one can see from Fig. 2, there appears to be a marked reduction in the total time required to recover from a cache miss using the TDIF replacement algorithm when compared to the LRU technique. The difference in TMT between these algorithms may be directly equated to a reduction in communication channel allocation by the network. In addition, there appears to be a clear connection between the choice of

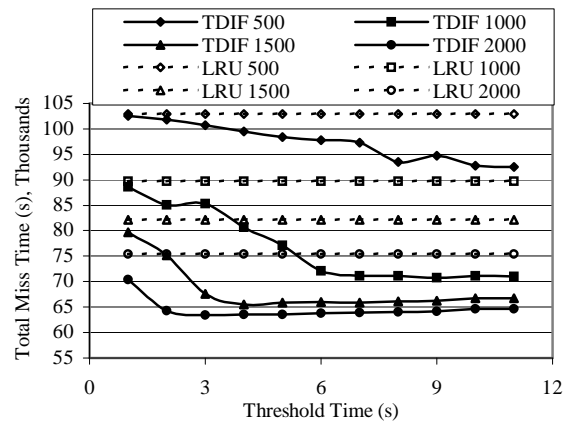


Figure 2. Comparison between TDIF and LRU cache replacement techniques for a range of cache sizes.

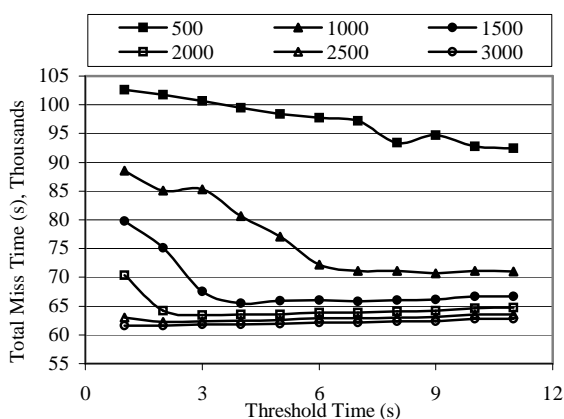


Figure 3. Simulation results of the TDIF algorithm run for various cache sizes.

threshold time and the benefit of the TDIF over the LRU technique.

It is evident from Fig. 3 that the rate of the reduction in the benefit of successive TDIF trials increases with the increase in the proportion of cache size to resource diversity. It is likely that this is attributable to the fact that as cache size increases, the probability of a cache hit increases.

Fig. 4 shows that as the cache size becomes a larger proportion of resource diversity the benefit of the TDIF technique over LRU is reduced for the dataset of this experiment. However, the benefit of the TDIF technique appears to hold for cache sizes that are a modest proportion of resource diversity (less than 50%).

## VI. SUMMARY

The ever-increasing demands on network resources require network administrators to continually search for new methods to reduce the loading of communication channels. Simulations of the TDIF cache replacement algorithm proposed in this paper have demonstrated a significant reduction in total miss time; the sum of the time required to recover from a cache

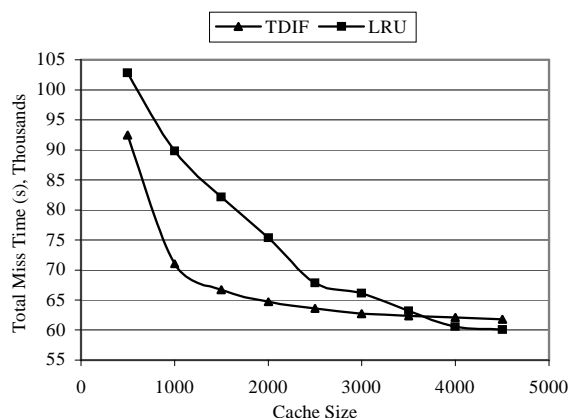


Figure 4. Comparison between TDIF and LRU cache replacement techniques.

miss. Such a reduction may be equated to a reduction in the time a network communication channel must be allocated to the retrieval of a requested resource. The ultimate result is a reduction in overall network loading and an increase in the effective bandwidth of a network employing a TDIF cache.

Future work with the TDIF concept will include the implementation of a technique to dynamically optimize the threshold value. In addition, an analysis of the effects of the implementation of the TDIF algorithm on the average user within the network will be considered.

## ACKNOWLEDGMENT

This work was partially supported by the Texas Advanced Research Program under Grant No. 003656-0108b-2001.

## REFERENCES

- [1] Y. M. Chan, J. P. Womer, S. Jamin, and J. K. MacKie-Mason, "One Size Doesn't Fit All: Improving Network QoS Through Preference-driven Web Caching", *Second Berlin Internet Economics Workshop*, Berlin, Germany, May 28-29, 1999.
- [2] P. Cao and S. Irani, "Cost-Aware WWW Proxy Caching Algorithms", *Proceedings of the 1997 Usenix Symposium on Internet Technologies and Systems*, Monterey, CA, December 1997, pp. 178-185.
- [3] T. P. Kelly, Y. M. Chan, S. Jamin, and J. K. MacKie-Mason, "Biased Replacement Policies for Web Caches: Differential Quality-of-Service and Aggregate User Value", *Fourth International Web Caching Workshop*, San Diego, CA, March 1999.
- [4] S. Williams, M. Abrams, E.A. Fox, G. Abdulla, and C.R. Standridge, "Removal Policies in Network Caches for World-Wide Web Documents", *Conference Proceedings on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Palo Alto, CA, August 1996, pp.293-305.
- [5] D. E. Comer, *Computer Networks and Internets with Internet Applications*, 4<sup>th</sup> ed., Pearson Education, Inc., Upper Saddle River, NJ, 2004.
- [6] M. Afonso, A. Santos, and V. Freitas, "QoS in Web Caching", *3rd International WWW Caching Workshop*, Manchester England, June 1998, pp.2-5.
- [7] S. Podlipning, and L. Boszormenyi, "A Survey of Web Cache Replacement Strategies", *ACM Computing Surveys*, Vol. 35, No. 4, December 2003, pp. 374-398.
- [8] C. Williamson, "On Filter Effects in Web Caching Hierarchies", *ACM Transactions on Internet Technology*, Vol. 2, No. 1, February 2002, pp. 47-77.
- [9] S. Hosseini-Khayat, "Replacement Algorithms for Object Caching", *Symposium on Applied Computing*, Atlanta, GA, 1998, pp. 90-97.
- [10] G. Cao, L. Yin, and C. Das, "Cooperative Cache-Based Data Access in Ad Hoc Networks", *Computer*, Vol. 37, No. 2, February 2004, pp. 32-39.
- [11] N. Megiddo, and D. Modha, "Outperforming LRU with an Adaptive Replacement Cache Algorithm", *Computer*, Vol. 37, No. 4, April 2004, pp. 58-65.
- [12] P. Rodriguez, C. Spanner, and E. Biersack, "Analysis of Web Caching Architectures: Hierarchical and Distributed Caching", *IEEE/ACM Transactions on Networking*, Vol. 9, No. 4, August 2001, pp. 404-418.
- [13] K. Psounis, and B. Prabhakar, "Efficient Randomized Web-Cache Replacement Schemes Using Samples From Past Eviction Times", *IEEE/ACM Transactions on Networking*, Vol. 10, No. 4, August 2002, pp. 441-454.
- [14] M. Sequera-Devilacheise, J. Menaud, G. Muller, and J. Lawall, "Web Cache Prefetching as an Aspect: Towards a Dynamic-Weaving Based Solution", *Proceedings of the 2<sup>nd</sup> International Conference on Aspect-Oriented Software Development*, Boston, MA, March 2003, pp. 110-119.