Cumulative Caching for Reduced User-Perceived Latency for WWW Transfers on Networks with Satellite Links♦

Aniruddha Bhalekar [1] and John Baras [2]

[1] Intelsat Global Service Corporation
3400 International Drive NW, Washington DC 20008, USA
aniruddha.bhalekar@intelsat.com
[2] Center for Satellite and Hybrid Communication Networks, Institute for Systems Research
University of Maryland, College Park 20740, USA
baras@isr.umd.edu

Abstract. The demand for internet access has been characterized by an exponential growth. The introduction of high-speed satellite communications systems providing direct-to-home internet is a response to this increasing demand. However such systems use geo-synchronous satellites and suffer from high latency. Currently, the most popular application layer protocols for the World Wide Web (WWW) are HTTP/1.0 and HTTP/1.1. Since HTTP is a request-response protocol, there are performance issues with using it over high-delay links such as links involving Geo-synchronous Earth Orbit (GEO) satellites. Such usage leads to severely increased user perceived latency which makes "internet browsing" a cumbersome experience. In this paper we investigate this problem and analyze a mechanism to reduce this user-perceived delay.

1 Introduction

In this paper we focus on the cumulative caching scheme which tries to reduce the problem of high user-perceived latency. The scheme relies on caching and does not modify the HTTP protocol in any way. This approach uses the characteristic network topology to its advantage and reduces latency by incorporating minimal changes. The scheme has several advantages which include easy and inexpensive implementation and immediate savings in latency of up to 40%.

This paper is divided into seven sections including this introduction. In the next section, we define the problem we are trying to solve using this scheme by discussing the necessary background.

In the third section, we discuss the motivation for the cumulative cache scheme and then we go on to specify the algorithm it uses. We emphasize that we focus on a

♦ Research supported by NASA under cooperative agreement NCC8235, Hughes Network Systems and the Maryland Industrial Partnerships Program.

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 179-186, 2004.

system that works with a single VSAT terminal supporting multiple users for Internet access i.e. Small Office Home Office (SOHO) setups.

In the fourth section we discuss the topology that is the target for the cumulative cache scheme. We continue in this section by elaborating the algorithm used by the scheme to paint web pages on the user's browsers

In the fifth section we look at the related work that has been done in this area. We first discuss the work done in trying to analyze the nature of web-browsing and show via statistics that our scheme will indeed prove beneficial. In the latter part, we describe Zipf's law, its applications and its impact in this area.

In the sixth section we state our observations and more importantly, we quantify the benefits of this scheme. Also, here, our goal is to touch upon the implementation details of this scheme, where we mention some of the aspects that must be taken into consideration for the commercial deployment of this product.

The last section includes the conclusions and talks about the possible issues with the cumulative cache scheme which could determine the future work in this area.

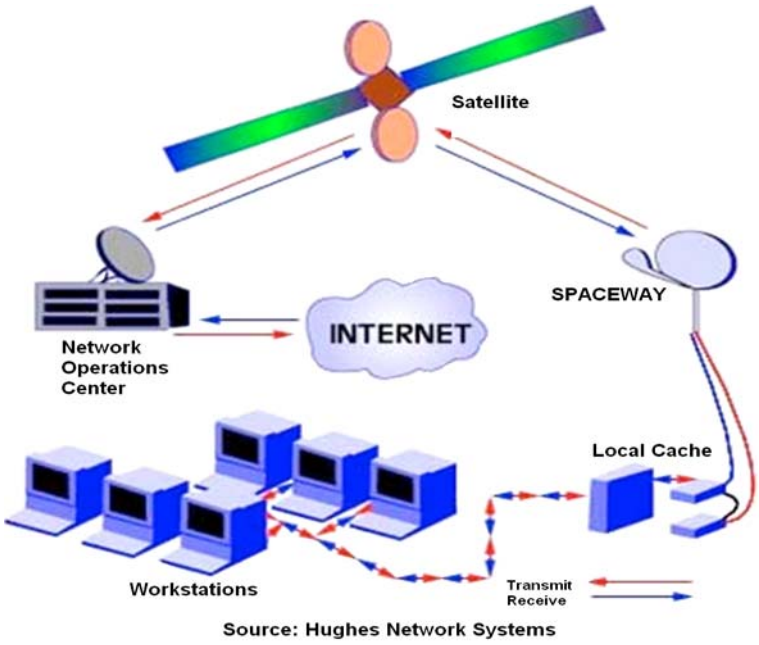

Fig. 1. Network under consideration

2 Background

HTTP is a request-response type application layer protocol [1] on TCP Reno. A HTTP transaction involves setting up a HTTP over TCP connection. This involves the traditional 3-way TCP handshake for connection establishment. The service we

consider is the Hughes Network Systems (HNS) SPACEWAY®. This is a two-way direct to home broadband Internet system. "Two-way" implies that both, the incoming data to the end user's terminal ("user") from the Internet and the outgoing data from the user to the web-server use the satellite segment. The basic network topology is as shown in Figure 1, where the local cache is what we call the cumulative cache. The satellite in consideration is a Geo-synchronous Earth Orbit (GEO) such as, PANAMSAT Galaxy XI. Due to the physical distance between the satellite and earth stations (22,282 miles) the time it takes for radio waves to reach the satellite, once transmitted by the earth station is just more than 1/8th of a second. The transit time on the downlink from the satellite to the hub (NOC) is also just greater than 1/8th of a second. Hence the Round-Trip Time (RTT) i.e. from the remote earth station to the satellite to the hub and from the hub to the satellite and back to the earth station is just larger than half a second.

This large RTT amplifies the latency caused by the TCP triple handshake which is required for a connection to be set up between a client and a server. Also, in TCP typically, the Maximum Segment Size (MSS) is 536 bytes. Client HTTP requests, which are sometimes larger than the TCP segment size, span multiple RTTs in addition to the initial connection setup overhead of HTTP over TCP to get through to the server [2]. TCP's Slow Start algorithm aggravates this delay as the second packet cannot be sent until the first has been acknowledged [3]. The HTTP response may hence require multiple round trips to reach the server. This causes increased user-perceived latency due to HTTP transactions on this segment. This latency becomes unusually apparent for the end-user when the transaction between the client and the server is a request from a computer to a web server and the response information is the base HTML web-page along with several objects embedded in the base page. This makes web-browsing a very cumbersome experience.

3 Motivation and Algorithm

The cumulative cache serves a group of end user machines served by one VSAT terminal for broadband access. This setup is typically a small office or a home office, where different users have common web browsing patterns, such as repeated hits to a web page that has data that is pertinent to the nature of their work. We hence onwards refer to this topology as the SOHO (Small-Office Home-Office) topology and customize our scheme to benefit such users. Note that the cumulative cache is located between the users and the VSAT terminal. The cached content is typically application-layer content. The working of the cumulative cache is as simple as caching all the internet content that passes through it.

This implies that all of the cacheable internet content viewed by all users is cached in this cache. Since the number of users contributing to it could be relatively large, it could be flushed every 24 hours (at lowest usage time of day) and it begins to refill the moment users start browsing next, or after the flushing of the cache finishes. This way, the pages that are painted on the browsers are cumulatively cached and subsequent requests to those pages within the flush period, are cumulative cache hits. When the same client or another client requests the same page, the locally cached version is displayed. At the same time the timestamp of this cached version is sent

over the satellite segment for validation from the cached version at the Network Operations Center (NOC). If the NOC has a cached version of the web resource with a later time-stamp, it sends it over the satellite link to the client, where the client generates the new page and the browser auto-refreshes. Simultaneously, the NOC checks with the web-server and refreshes or upgrades the version it has currently cached. If the web-page cached at the NOC is upgraded, the NOC sends the newer page to the client over the satellite segment resulting in the client receiving the web-page and the browser auto-refreshing. The NOC refreshes/upgrades the page in its cache irrespective of whether the client has the same version as the NOC or otherwise. This ensures the freshness of the cached web resource at the NOC.

Note that this cumulative caching scheme is not the same as pre-fetching. It is a much simpler scheme. It does not use any fetching algorithm. The pages that have not been requested before are fetched from the source. This scheme also does not incorporate any fetching delay in the pages that have not been fetched before, i.e. first-time requests.

4 Related Work

In this section we discuss the related work in this area. We focus on work involving the nature of web browsing and Zipf's law's applications to web browsing.

4.1 Nature of Web Browsing

We now look into why this scheme will actually work and why it is especially beneficial for SOHO user networks. Benefits of caching in this environment are based on the assumption that a large fraction of the HTTP responses have already been received and that these resources may or may not change between accesses. Douglis et al in [4] state that 22% of the web resources referenced in the traces they analyzed were accessed more than once. The first study in a related area, which used "live" users to test and see if the benefits would apply in practice, used two traces from independent sources for a trace-based analysis to quantify the potential benefits from both proxy-based and end-to-end applications [5]. This study claims that users in the same geographic area visit the same websites due to the mirroring of certain web-servers or other reasons.

Also, users with the same nature of work visit the same websites according to this paper. The percentage of traffic, which is repeated by a single user was calculated in terms of "delta-eligible" HTTP responses by this paper. Note that delta-eligible responses are ones, which reply with a different instance of the same resource (HTTP Status code 200). In the traces, 20-30% of the status 200 HTTP responses were delta-eligible i.e. changed slightly from what was cached. Even in the status 200 HTTP responses, 30 % were identical to what was cached. This paper ignored the responses that had the same instance of the resource as the one that was cached (HTTP Status code 304). In spite of trying to filter out these "not-modified-since" responses, that number was 14% of the total number of responses in the trace.

More recent studies show that 15-18% [6], 30% [7] and 37% [8] of HTTP requests responded with Status Code 304, i.e. cached copy is up-to-date. Also, [9] states that it is well known that 20% to 30% of all requests are conditional GET requests with 304 (not modified) replies. This means that for an individual user 15% to 37% of all requests and responses are identical to previous responses. Also, for a single user, up to another 30% are repeated requests, where the response has been a different version of the cached resource. If we were to use the cache as a cumulative cache for a group of users in the same geographical area and with the same nature of work, we achieve at least 40% and possibly up to 100% hits in the cache, per session. The usefulness of the cumulative caching scheme is validated by these statistics.

4.2 Zipf's Law

We also look into some recent work which deals with the application of Zipf's law to the nature of web-browsing. Zipf's law states, "The probability of occurrence of words or other items starts high and tapers off. Thus, a few occur very often while many others occur rarely." Mathematically, this translates to what is popularly known as the 80/20 rule or the Pareto principle (which is a special case Pareto distribution) [10]. This theory when applied to web access has been claimed to be equivalent to the fact that, user visits a certain small percentage of web resources often and visits a large number of other web-pages very infrequently. This means that on an average, 80% of all HTTP requests by a web browser are directed towards only 20% of the online resources it accesses and the remaining 20% of the HTTP requests are for the remaining 80% web resources.

This is very interesting for our scheme, because it means that even if the cumulative cache saves only 20% of all the web resources that pass through it, most users could benefit up to 80% of the time i.e. the perceived network latency will not appear 80% of the time. A study by Pei Cao gives us numbers that validate the fact that the figures related to internet browsing are very close to the 80/20 rule [9]. These include web accesses seen by a proxy. For example, 25% of all documents accounts for 70% of Web accesses in DEC, Pisa and FuNet traces, while it takes 40% of all document to draw 70% of Web accesses in UCB, QuestNet and NLANR. Hence, realistic figures for a Zipf-like distribution for web requests are 70/30.

A study by Breslau et al addresses two similar issues. The first issue is whether Web requests from a fixed user community are distributed according to Zipf's law and the second issue is whether this characteristic is inherent to web accesses or not [11]. On investigating the page request distribution, the paper shows that the answers to the two questions are related. The paper also conforms with [9] in the finding that the page request distribution does not follow Zipf's law precisely, but instead follows a Zipf-like distribution with the exponent varying from trace to trace. They considered a simple model where the Web accesses are independent and the reference probability of the documents follows a Zipf-like distribution and found that the model yields asymptotic behaviors that are consistent with the experimental observations. This suggests that the various observed properties of hit ratios and temporal locality are indeed inherent to Web accesses observed by proxies.

5 Observations and Benefits

The cumulative cache scheme is well supported by the observations made in the papers quoted in the previous section. We now quantify the reduction in latency using this scheme. The reduction in latency by 70% repetition of requests for web resources by an individual user is 40% and not 70%, since 30% requests of theses 70% get a different version of the resource in the response. Out of the remaining 60%, up to 42% could overlap with other users browsing patterns in the SOHO network. We define "cumulative resources" as resources that have been or will be requested by at least one other user in the SOHO network. Hence the probability of requesting a cumulative resource is 0.7 by the 70/30 variant of Zipf's law. "N" users in the SOHO network are equally likely to request a cumulative resource. Hence, the probability of any individual user requesting a cumulative resource is 0.7/N. This implies that the probability of any individual user not requesting a cumulative resource first is 1-0.7/N. Note that not requesting a cumulative resource first, implies that some other user in the SOHO group has requested it earlier. This implies a 100% reduction in the latency for the arrival of this resource at the client. If the number of users in the SOHO network is 10, i.e. N = 10, which is a very realistic figure, the reduction in latency in the remaining 60% is equal to the probability of not requesting a cumulative resource first, which is 93%. This translates into an additional reduction of up to 39% in addition to the 40% reduction in latency due to the self-repetition nature of web requests of the individual user. This amounts to a total reduction in latency of up to 79% using the cumulative cache.

Note that responses from the caches are perceived as instantaneous by the user. If we assume a base page size of 50kB plus 100kB (sum of all embedded object sizes in the web page). Hence the total page size is 150kB. The time to transfer page from the browser cache and from the cumulative cache (Ethernet LAN at a transfer rate of 100 Mbps) is perceived as instantaneous by the end user as compared to the time to transfer the same page over the satellite segment, which is seen to be at least 3½ seconds. This calculation takes into consideration the NOC search time (RAM access), the TCP triple handshake time and the request, response and page transfer times.

These figures show explicitly the benefit of cumulative caching. This implies a 40% through 79% reduction in the user-perceived latency in direct proportion to the hit-ratio of the user to the cumulative cache. Following the discussing on Zipf's law, savings close to this can be achieved even if all of the internet content passing through the cache is not saved and some kind of "smart" caching scheme is employed, which caches only the most requested responses.

6 Implementation

We summarize the implementation details of the cumulative cache in this section. Please note that we do not detail upon what cache replacement algorithms to use and assume a non-realistic but simplistic approach that the benefits we obtain are directly proportional to the cache size.

Currently HNS uses a set-top box which runs at the network layer, as part of the IDU (Indoor Unit) with the DIRECWAY® system at the SOHO VSAT terminal. This "box" needs to be provided with additional memory and enabling its working at the application layer will make it function as the cumulative cache. This might prove to be expensive for the service provider as the equipment cost increases with every kilobyte of storage. Instead of using additional memory provided by the service provider, the end user (SOHO network) could be encouraged to use a part of the existing infrastructure of the network as the cluster cache, to curb additional aggregation to product cost. Due to this, this scheme can be implemented by the service provider, HNS in our case, as optional but recommended. Since SPACEWAY® focuses on the SOHO market, the incoming traffic could be directed through one of the user's computers where a part of the memory could be configured to cache incoming WWW data, using a daemon process.

We may keep this process transparent to the end user or may let the end user know by displaying a "Page is being Verified" sign or equivalent while the cached page is being displayed and confirmation about its freshness has not been received from the source i.e. the web server. The risk involved in displaying outdated web-pages, for a few seconds, is lightened by the fact that most web designers change just some form or appearance of the web-page but not the content of the page in order to give the webpage a fresh look [12]. This method results in instantaneous gratification for the end user. We also note that the probability of an outdated page being displayed to the user is miniscule in the cumulative cache scenario, as the cache may be flushed every 24 hours and it beings to fill at the start of business, everyday.

7 Conclusions and Future Work

By means of our discussion above we observe that the setting up of a cumulative cache for a SOHO-type network achieves very good results in terms of the user's perceived latency for WWW access using a satellite link. Our observations suggest a minimum reduction of 40% and up to 100% (if the user only visits pages that have been visited before) in the user's perceived latency, using this scheme. The application of Zipf's law to this scenario shows that very similar results can be achieved by caching a much smaller number of HTTP responses if a smart caching scheme, which caches only the most requested web-pages is developed.

We are currently analyzing the risk of displaying outdated pages even momentarily to the user and the effect that he/she has knowledge of the same has on his/her view on satisfactory web-browsing. We are also looking into the issue of privacy and security to make sure that no individual user has access to the contents of the cumulative cache as it may contain sensitive information such as personal or financial information of other users. Hence the cumulative cache must be securely protected against unauthorized access and must be used for sharing non sensitive information i.e. the cache should be accessible only by the browser process and this should be transparent to the user.

We also plan to look into the cost/benefit ratio i.e. the size of the cache (cost) to perceived reduction in latency (benefit), to determine the optimal size of the

cumulative cache for a fixed number of users in the SOHO network along with develop an appropriate specific cache replacement algorithm for this application.

References

1. Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T.: Hypertext Transfer Protocol – HTTP/1.1. IETF RFC 2616 (1999)

2. Spero, S.: Analysis of HTTP Performance Problems. Available at `http://metalab.unc.edu/mdma-release/http-prob.html` (1994)

3. Allman, M., Paxson, V., Stevens, W.: TCP Congestion Control. IETF RFC 2581 (1999)

4. Douglis, F., Feldmann, A., Krishnamurthy, B.: Rate of Change and other Metrics: a Live Study of the World Wide Web. Proceedings of USENIX Symposium on Internetworking Technologies and Systems (1997)

5. Mogul, J., Douglis, F., Feldmann, A.: Potential Benefits of Delta Encoding and Data Compression for HTTP. Proceedings of SIGCOMM (1997)

6. Krishnamurthy, B., Willis, C. E.: Piggyback Server Invalidation for Proxy Cache Coherency. 7th International WWW Conference, 185-193, Brisbane, Australia (1998)

7. Nahum, E.: WWW Workload Characterization work at IBM Research. World Wide Web Consortium Web Characterization Workshop, Cambridge, MA (1998)

8. Arlitt, M., Jin, T.: Workload Characterization of the 1998 World Cup Website. Technical Report HPL-1999-35, HP Laboratories, Palo Alto, CA (1999)

9. Cao, P.: Characterization of Web Proxy Traffic and Wisconsin Proxy Benchmark 2.0. Position Paper in World Wide Web Consortium Workshop on Web Characterization, Cambridge, MA (1998)

10. Pitkow, J.E.: Summary of WWW characterizations, Computer Networks and ISDN Systems. vol. 30, no. 1-7, pp. 551-558 (1998)

11. Breslau, L., Cao, P., Fan, L., Phillips, G., Shenker, S.: Web Caching and Zipf-like Distributions: Evidence and Implications. Proceedings of the Conference on Computer Communications, IEEE INFOCOM, New York (1999)

12. Francisco-Revilla, L., Shipman F.M. III, Furuta, R., Karadkar, U., Arora, A.: Perception of Content, Structure and Presentation Changes in Web-based Hypertext. Proceedings of the 12th ACM conference on Hypertext and Hypermedia, Arhus, Denmark (2001)