Information Quality Based Distribution of Multimedia Streams in Mobile Environments

Arunabha Sen and **Sandeep Gupta**
Department of Computer Science and Engg.
Arizona State University
Tempe, AZ 85287
asen, skgupta@asu.edu

Martin Reisslein
Department of Electrical Engineering
Arizona State University
Tempe, AZ 85287
reisslein@asu.edu

1 Introduction

With the advent of broadband wireless services in the not too distant future, information dissemination through storage and distribution of multimedia streams to wireless mobile devices through the Internet will be a significant component of the Internet traffic. The advantages of bringing *high quality* multimedia streams to a mobile client cannot be over emphasized. This opens up new opportunities in diverse applications such as distance learning, entertainment, information gathering, and surveillance. In distance learning applications, a commuter on a train can utilize the commuting time for virtual attendance in a class lecture offered by a renowned professor at a remote university. In entertainment application, using video–enabled devices such as cell phones and PDAs, parents would be able to entertain their children with video programs during a long car ride.

It may be noted that in all these applications it is *information* that is being delivered through the *audiovisual medium*, i.e., through continuous media objects (multimedia streams). This paper introduces a notion of *quality of information* and uses it for efficient delivery of multimedia streams in a mobile wireless environment. Our notion of quality of information has three components:

- *Quality of content*
- *Quality of audio and visuals delivered to the client*
- *Currency of information*

As the quality of content is decided by the content provider and is beyond our control, we focus our attention on the other two aspects of information quality. Considerable research efforts are under way for the last few years to address the issue of quality of audio and visuals delivered to the client. In the computer network literature this is referred to as the *quality of service* issue. Traditionally, quality of service is evaluated by network parameters such as *delay, jitter, bandwidth, packet drop probability* etc. Our notion of *quality of information* not only considers the quality of video delivered to the client, but also the *currency of information* delivered through the video. If a client is trying to gather information from the web regarding some event, if he/she is provided with the most recent information regarding that event, it may be regarded as information of the *highest* quality. If we consider a scenario with one origin server and two proxy servers, where a file in the origin server is last updated at 11:00 am, the last update at the first proxy is at 10:30 am and the last update at the second proxy is at 10:00 am. If a query is made at 11:15 am, and if the response is provided from the origin server then the *information quality* of the response is *highest*. If it is provided from the first proxy, then the *information quality* is *intermediate* and if it is provided from the second proxy then the information quality is *lowest*.

It might appear that everyone would like to receive the *highest* quality response to their queries. However, we demonstrate in our discussion in later sections that there exists a clear benefit in introducing the notion of *information of varying quality* in the *storage* and *distribution* of multimedia streams, as it offers an opportunity for a trade–off between the *cost of information retrieval* and the *quality of retrieved information*. The benefits become even more apparent in a mobile wireless environment where, even when broadband wireless becomes a reality, bandwidth most likely will continue to be at a premium in comparison with its wired counterpart.

2 Related Work

The notion of information quality based distribution attempts to combine the *cache performance* issues with *cache consistency* issues. Although considerable work has appeared in the literature addressing cache performance issues (e.g., design of efficient replacement algorithms, prefetching of web pages etc.) and cache consistency issues, efforts where these two issues are considered simultaneously have received very little attention [10].

2.1 Cache Performance

The WebExpress project at IBM is the one of the few efforts where cache performance and cache consistency issues have been considered simultaneously. It was noted in [10] that their particular design decision resulted in high cache hit ratios, implying higher performance, while presenting a cache coherence problem. The notion of information quality being introduced here explores this issue in detail to find out the trade–offs between performance and consistency.

The scalability issue in web caching is addressed in [22]. They introduce a notion of *translucent caching* as an alternate to *transparent caching* for the scalability. Prefetching as a performance improvement technique has been reported in [8, 12, 15]. There have been significant amount of work on the design of efficient cache replacement policies for performance improvement. The factors that are taken into account for the design of an efficient cache replacement policy include,

- *Recency of reference*: The recency of reference indicates how recently the web object was requested. The *least recently used (LRU)* replacement algorithm uses the recency of reference as the sole criteria for deciding on the web object (page) to be evicted from the cache.
- *Frequency of reference*: The frequency of reference indicates how frequently the web object was requested. This measure separates the "popular" web objects from the "unpopular" ones. The *least frequently used (LFU)* replacement algorithm uses the frequency of reference as the sole criteria for deciding on the web object (page) to be evicted from the cache.
- *Object size*: Recent studies have shown that the size of a large number of web objects is fairly small. The size of the median object was found to be around 4KB in one of the web proxy workloads studied in [1]. This observation is likely to change drastically with an increased presence of a large number of audio and video objects on the web [25].

The policies where some or all of these three factors have been taken into account include:

- *Least Recently Used* (LRU) replaces the least recently requested web object.
- *Least Recently Used/k* (LRU/k) [23] replaces the web object whose k–th most recent request is least recent.
- *2Q cache management technique* (2Q) [14] low overhead technique that simulates the LRU/k
- *Least Frequently Used* (LFU) replaces the least frequently requested web object.
- *Size* [34] replaces the largest size document in the cache.
- *Lowest Relative Value* (LRV) [30] computes the *utility* value of a web object in cache and then replaces the one with the smallest utility value.
- *Lowest Latency First* [35] is designed to minimize the average latency and as such removes the document with the lowest download latency.
- Greedy Dual (GD) [36] is designed to handle uniform size, variable cost cache objects.
- Greedy Dual Size (GDS) [3] an extension of the GD to handle variable size web objects.

- Greedy Dual Size Frequency (GDSF) [5] an extension of the (GDS) that takes into account the *popularity* of the web objects, measured in terms of frequency of request in recent past.

In [5] the author claims that GDSF improves on the GDS, regarded as the *"current champion"*, of the various cache replacement algorithms. In spite of various claims of performance improvement by these algorithms, arguably the simplest one, LRU, continues to be the most popular replacement algorithm in use. The computational complexity of LRU is $O(1)$. The more sophisticated algorithms, such as LRU/k, GDS and GDSF all maintain a *priority queue* of the objects in the cache to determine the next object to be removed from the cache. Because of the priority queue, the computational complexity of all such algorithms is $O(log\ N)$, where N is the number of web objects in the cache.

2.2 Cache Consistency

Cache consistency issues in the WWW have been considered in [4, 7, 6]. In this proposal we are primarily interested in issues related to caching in a mobile environment. All the works which address caching of location–dependent data in mobile clients use data broadcasting. Based on the data request pattern on the on–demand channel the local server periodically selects and broadcasts some of the most frequently requested data items on the broadcast channel.

There are several works based on the idea of periodically broadcasting invalidation reports by Barbara and Imielinski [2]. Jing et. al.[13] have proposed a scheme to adjust the size of the invalidation report to optimize the use of wireless bandwidth while retaining the effectiveness of cache invalidation. Liu and Maguire [17] have proposed a two–level caching scheme based on mobility agents which take into account the mobility pattern of the user to restrict the broadcasting of the reports in the neighborhood of user's current location. Hu and Lee [11] have proposed broadcasting invalidation report methods which takes into account the update and query rates/patterns and client disconnection time to optimize the uplink query cost. The schemes based on broadcasting invalidation reports have following characteristics:

1. They assume a stateless server and do not address the issue of mobility (except the work by Liu and Maguire [17]).

2. The entire cache is invalid if the client is disconnected for a period larger than the period of the broadcast.

3. Scalability is questionable for large databases.

Coda file system provides support for disconnected operations on shared files in UNIX–like environment. The goal is to support file operations even while the user is disconnected from the network. Balance between speed of validating cache (after a disconnection) and accuracy of invalidations is achieved by maintaining version timestamps on volumes (a subtree in the file system hierarchy). Server replication is used to improve availability of files. This scheme has been specifically designed to support distributed file systems.

2.3 Caching of Video Objects

There are only few studies on distributing video objects through caches, all of which are complementary to the research proposed here. Rejaie *et al.* propose a proxy caching mechanism [27] in conjunction with a congestion control mechanism [26] for the streaming of layered–encoded video to wired clients. A related idea is explored by Wang *et al.* in their study on video staging [33]. With video staging the part of the VBR video stream, that exceeds a certain cut–off rate (i.e., the bursts of a VBR stream) is cached at the proxy while the lower (now smoother) part of the video stream is stored at the origin server. Sen *et al.* [31] propose to cache a prefix (i.e., the initial frames) of video streams at the proxy and to employ work–ahead smoothing while streaming the object from the proxy to the client. Similar ideas are explored by Ma and Du [20] and Rexford *et al.* [28] where the proxy cache is used as staging space that enables the delivery of smoothed video over the local access network (from the proxy to the clients). Rexford

and Towsley [29] extend this idea to smoothing video in a multi–hop delivery scenario; they stage the stream at several intermediate gateways along the origin server to client path. Miao and Ortega [21] propose to cache some video frames depending on the network congestion with the goal to maximize the video quality. In [19] Ma and Du study related ideas, where certain segments (chunks) of the video streams are cached. Tewari *et al.* [32] propose a Resource Based Caching (RBC) scheme for video objects encoded into one CBR layer. They consider caching certain segments (runs) of the video stream and model the cache as a two resource (storage space and bandwidth) constrained (deterministic) knapsack. In a related work, Ma and Du [18] formulate a family of segment caching policies as a (deterministic) knapsack problem and propose heuristics to solve it. In our proposed research we shall take advantage of the valuable insights provided on the caching of video layers (and certain segments or runs of these layers) by these related works.

3 Information Quality Based Caching

The objective of this paper is to develop schemes for delivering high quality multimedia streams to Internet–enabled mobile clients. We focus on providing continuous media service with video that is encoded with an object–based approach, such as MPEG–4. We propose to develop and evaluate strategies for distributing continuous media through a distribution infrastructure to wireless mobile clients. Our research places particular emphasis on developing *integrated* distribution strategies, that is, distribution strategies that encompass resource management, cache management, and consistency management. In particular, we will address the following research questions:

- How to efficiently distribute object–based media, such as MPEG–4 encoded video from servers (including web servers) to mobile end users?
- How to characterize the trade–off between the quality of information delivered (i.e., quality of content, quality of audio and visuals, and currency of content) and the distribution infrastructure and resources?
- How to intelligently trade off the wired infrastructure resources utilized with the quality of the content/video and currency to deliver multimedia streams to mobile clients in a mobility–aware manner?

Towards this goal, this paper introduces the notion of *quality of information* and uses it for the efficient delivery of multimedia streams in a mobile wireless environment. Our notion of quality of information has three components: 1) *Quality of content*, 2) *Quality of audio and visuals delivered to the client*, and 3) *Currency of information*. As the quality of content is decided by the content provider and is beyond our control, we focus our attention on the other two aspects of information quality. Traditionally, quality of service is evaluated by network parameters such as *delay, jitter, bandwidth, packet drop probability* etc. Our notion of *quality of information* not only considers the quality of the video delivered to the client, but also the *currency of information* delivered through the video. We further propose an *integrated* scheme for distributing object–based continuous media in mobile environments which:

- Integrates on–the–fly cache updates into the resource management analysis.
- Integrates consistency management into the resource management analysis.
- Extends resource management analysis to hierarchy of caches consisting of wired proxy cache and cache in wireless mobile client.
- Exploits the knowledge of the mobility pattern of mobile clients and the object–based plus layer–based encoding of the multimedia stream to make the distribution scheme mobility–aware.

We evaluate our multimedia distribution scheme through analysis and simulation. We employ a number of analytical modeling techniques (e.g., stochastic knapsack modeling) and will extend a preliminary cache simulator package, which was developed by Dr. Reisslein's group at GMD FOKUS, Berlin, Germany.

4 Distribution Architecture for Continuous Media Streams in Mobile Environments

Figure 1 illustrates our basic architecture for distributing continuous media to wireless mobile clients.

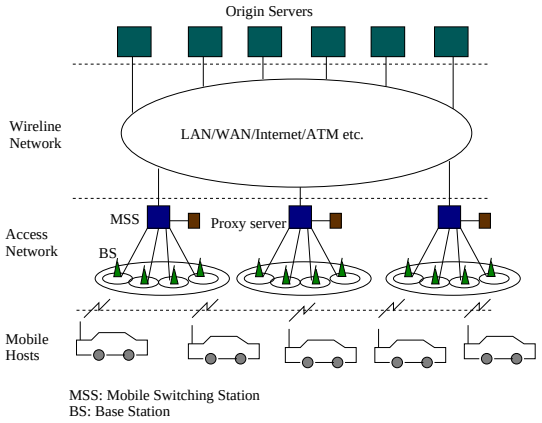

Figure 1: Architecture for distributing continuous media streams to wireless mobile clients.

We consider a network consisting of a wireline backbone and multiple wireless cells (as the last hop). All available continuous media are stored on the origin servers. *The continuous media streams available on the origin servers are prerecorded audio and video streams, such as CD–quality clips, short video clips (e.g., news clips, trailers or music videos) or full length movies or on–line lectures.* Popular content is cached in proxies at the Mobile Switching Stations (MSS). (Proxy caches could also be located at the individual base stations, however, to fix ideas we assume for now that proxies are only located at the mobile switching stations. We shall address the issues and trade–offs of proxy placement in our proposed research.) Each client maintains a local cache of content which is frequently accessed. Before answering any queries from the application the client checks if the requested video is cached at the desired level of quality and freshness. If the cached video does not meet the requested quality and freshness, the client directs its streaming request to the appropriate proxy server. If the requested video (quality and freshness) is not cached in the proxy, it is streamed from the origin server over the wide area network to the proxy. The proxy forwards the stream to the client.

For our analysis we model object–based video with multiple quality layers, i.e., a basic quality layer and several quality enhancement layers. We make the natural assumption that a particular quality enhancement layer can only be decoded if all lower quality layers are available. Therefore a quality enhancement layer is useless for the client if the corresponding lower quality layers are not available. Suppose that there are M videos. As a first approximation we assume that the video quality layers are constant bit rate. Let $r_l(m)$ denote the rate (in bit/sec) of quality layer l, $l = 1, \ldots, L$, of video m, $m = 1, \ldots, M$. We define a j–quality stream as a stream consisting of quality layers $1, 2, \ldots, j$. Let $T(m)$, $m = 1, \ldots, M$, denote the length (in seconds) of video m. We model the bandwidth available for streaming continuous media from the origin servers to each proxy as a bottleneck link of fixed capacity C_p. Each proxy is connected to the base station via a local access network, which we assume provides abundant bandwidth. We model each proxy as having a storage capacity of G_p (bytes).

5 Distribution of Object–based Continuous Media

The focus of the research is on video that has been encoded using object–based encodings techniques, such as MPEG–4 [16, 24, 9]. MPEG–4 is the encoding standard expected to dominate in future wireline and wireless environments as it provides very efficient video coding covering the range from the very low bit rates of wireless communication to bit rates and quality levels beyond high definition television (HDTV). In contrast to the "frame–based" video coding of MPEG–1 and H.263, MPEG–4 is object based. In object–based encoding each scene is composed of Video Objects that are coded individually. These objects may be obtained by applying object segmentation techniques [37] to a given (unencoded) video stream. Alternatively, the concept of video objects may be integrated into the video generation process, that is, the distinct objects of a scene are captured and encoded individually. Consider, for instance, the picture–in–picture format typically used in video taped lectures for distance learning programs, such as National Technological University (NTU) lectures. In NTU videos the slides or the lecturer's writing pad typically occupy a large rectangular area in the center of the video scene, while the head shot of the lecturer is shown in a small rectangular area in the upper right corner. (Should the head shot be unavailable, then this area is filled with the background color.) A simple object–based encoding scheme could encode the lecturer's writing pad and the lecturer's head shot as distinct objects. Each video object is typically encoded into several scalability layers (i.e., base and enhancement layers) which are referred to as video object layers in MPEG–4 terminology. Note that if scene segmentation into multiple objects is not available, then the entire scene is encoded into one video object (similar to MPEG–1 and MPEG–2 encoding). The single video object representing the entire scene may be encoded into several video object layers (similar to MPEG–2's scalability layers).

We investigate integrated strategies for distributing object–based encoded continuous media. Our preliminary work on resource management, cache management, and consistency management forms the basis for the proposed research. We note that there are many distribution solutions possible, where some video objects of a stream are cached in proxies, while the other video objects are streamed. Moreover, some video object layers of a video object may be cached while the other video object layers are streamed upon request. We propose to extend and integrate the resource management, cache management, and consistency management schemes to accommodate distinct video objects of a given video stream. To fully exploit the new degree of freedom that object–based encodings provide, we need to take the objects' "relevance" for the video stream into consideration. Clearly, in the NTU lecture example, the object representing the lecturer's writing pad is more relevant than the lecturer's head shot; the head shot could be omitted (to save networking resources) without dramatically reducing the viewing quality of the lecture video.

5.1 Information Quality based Distribution

In this section we introduce the notion of an *Information Quality* based distribution scheme. In our model the mobile clients are gathering *information* through multimedia streams, i.e., through continuous media objects. In our model the *quality of information* depends on the following three factors.

- *Quality of Information*
  - *Quality of content*
  - *Quality of multimedia streams*
  - *Currency of information*

Among these three factors, the quality of content is decided by the content provider and is beyond our control. Accordingly, we focus our attention on the other two aspects of information quality. More specifically, we address the issues related to *efficient distribution* of *information of varying quality* in a *mobile environment.*

The transmission of multimedia objects must satisfy a set of *quality of service* requirements. These requirements typically consist of lower bounds for bandwidth and upper bounds for delay, jitter, and

packet loss. We would like to extend this notion of quality of service to include the *quality of information* being delivered. In the context of content delivery on the Internet, one important parameter in evaluating the quality of information, is the *recency* of information. If a client is trying gather some information from the web regarding some event, if he/she is provided with the most recent information regarding that event, it may be considered to be information of the *highest* quality. In our model the assumption is that every video clip has a origin server, different version of the clip (of varying information quality), is stored in several different proxies scattered over the network. If we consider a scenario with one origin server and two proxy servers, where a file in the origin server is last updated at time t_1, the last update at the first proxy is at t_2 and the last update at the second proxy is at t_3, where $t_1 > t_2 > t_3$. If a query is made at t_4 ($t_4 > t_1$), and if the response is provided from the main server then the *information quality* of the response is *highest*. If it is provided from the first proxy, then the *information quality* is *intermediate* and if it is provided from the second proxy the information quality is *lowest*.

Obviously, everyone would like to receive the *highest* quality response to their queries. However, there are benefits for the client as well as the network in introducing this notion of hierarchy in the information quality. First we discuss the benefit for the client. In our example scenario with one origin server and two proxy servers, it is possible that the client may get the fastest response if it is delivered from the second proxy, the second fastest response if it is delivered from the first proxy and the slowest response if it is delivered from the origin server. There may exist a clear *trade off issue* between the *quality of information* being provided to the client and the *performance* as observed by the client in terms of *latency of response*. Sometimes the clients may prefer to obtain the *fast* response to the query over the *most accurate response*. If the information retrieval involves a *monetary cost*, it is most conceivable that the most accurate information will also be the most expensive because such information may have to be retrieved from the origin server located far away from the client. Retrieving continuous media objects may involve the *reservation* of network bandwidth. The cost of the bandwidth reservation between Phoenix and Sydney may be considerably higher than that of reserving bandwidth between Phoenix and Los Angeles.

The notion of hierarchy in the information quality has benefit from the network standpoint also. To maintain consistency of information in one of the current schemes (push technology), all the proxy caches are updated every time there is some change in the origin server. However, such a scheme generates considerable amount of network traffic, specially when the number of proxy servers is fairly large. Our view of the hierarchical proxy cache environment is shown in Figure 2. The proxy caches are divided into a set of k tiers, numbered 1 through k, tier 1 being nearest to the origin server and tier k the furthest. In the mobile environment, the k-th tier proxies may be thought of as the base stations for mobile clients. The number of proxies in each tier increases in a geometric series with the number of tiers. In our scheme, proxies in all the tiers are not updated every time there is some change in the origin server. If p and q represent two tiers with $1 \leq p < q \leq k$, the p-tier proxies are updated more frequently than the q-tier proxies. In ours scheme, in the example of Figure 2, tier 1 proxies will be updated every t seconds, tier 2 proxies will be updated every $2t$ seconds, tier 3 proxies will be updated every $3t$ seconds and so on. The value of t may be determined by observing the frequency of change at the origin server. Suppose that there are k tiers of proxies, with tier 1 being the nearest and tier k being the furthest from the origin server. The p-th tier proxies are updated every pt seconds, $1 \leq p \leq k$. As shown in the figure, the number of proxies increases significantly from tier to tier, with fewest proxies near the origin server and largest number of proxies at the *edge* of the network, i.e., near the mobile client. If the number of proxies increase in a geometric progression, as shown in the figure with p proxies in tier 1, p^2 proxies in tier 2, p^k proxies in tier k, the total number of proxies is $(p^{k+1} - p))/(p-1)$. If all the proxies are updated every t seconds, the total number of update messages will be $(p^{k+1} - p)k/(p-1)$, during an interval of time kt. However, following our scheme, if only p-th tier proxies are updated every pt seconds, $1 \leq p \leq k$, then the total number of update messages will be $k \cdot p + (k-1) \cdot p^2 + (k-2) \cdot p^3 + \ldots + 1 \cdot p^k$. Thus, the total savings in update messages may be quite substantial.

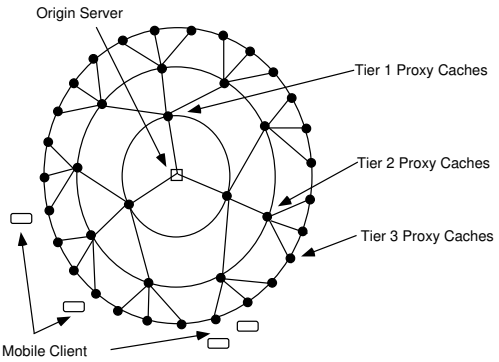

Figure 2: Model of Hierarchical Cache.

5.2 Mobility–Aware Continuous Media Distribution

The distribution of multimedia streams in the mobile environment should be able to adapt to the mobility of the participants in order to provide quality–of–service approaching those provided by wired infrastructure. The challenge here lies in intelligently trading–off wired resources such as bandwidth and storage with quality–of–service components such as hand–off jitter and call termination probability. Exploiting the layered and component based encoding of the multimedia objects in this regards will be the focus of this portion of the research project. In the following we outline some of the strategies we will develop and evaluate to make our multimedia distribution schemes suitable for the mobile environment.

We enhance the mobility–agent based approach developed by Liu and Maguire [17] for mobility–aware dynamic database caching. Similar to [17], we will assign mobility agents (an agent here means a dynamic software entity acting on behalf of some other entity) to the base–stations in a region around the current location (cell) of a mobile client. Each mobility agent for a mobile client is in charge of determining which components of the multimedia stream it should fetch. The purpose of this is to ensure a smooth hand–off of the mobile client to the cell of the mobility agent. Further, it can also select the number of encoding layers for an object within the multimedia stream component it should obtain. These decisions can be based on several factors including the current location of the mobile client, the type of multimedia service desired by the mobile client, the current state of the multimedia stream, the probability that the mobile client will move to the mobile agent's cell in the near future, and the quality of service desired by the mobile client. For example, a mobility agent located in the cell in the predicted path of the mobile client may decide to prefetch all the layers of the component following the current component of the multimedia stream being viewed by the mobile client. Whereas, the mobile agent located in cell just crossed by the mobile client may decide to prefetch only the basic layers of the next component since it is may be unlikely, although not totally improbable, that the mobile client may turn back and revisit the cell it just left. Computation of the probabilities will rely on the known/predicted mobility pattern of the user. Since the system has to support several such users, an important issue in using such an agent based approach is resource management of the resources consumed by mobility–agents. For example, when should a mobility–agent be invoked/terminated at a base station and how many and which mobility agents should be allowed at any given time to operate at a base station ? Our initial plan is to associate a *service–region* with each mobile–client and a *least–recently used* (LRU) counter with a mobility–agent similar to the one used in [17]. The service–region of a mobile client is a region of cells and is based on the current location of the mobile client as well as its velocity. The service region is used to determine which cells in the neighborhood of the current cell of the mobile client should be assigned mobility–agents. The LRU counter is used to determine when to terminate a mobility–agent. Initially, the LRU counter is zero. It is incremented periodically. When there is no room for creating a newly assigned mobility–agent to a base station, one of the existing mobility–agent with greatest LRU counter is evicted. The LRU counter is reset whenever its corresponding mobile client visits the cell of the mobility–agent. The effectiveness

of this simple scheme will be evaluated before we decide to develop more complex schemes.

The above scheme is enhanced to work with the proxy–based distribution discussed in earlier. A mobility–agent will have to decide from which level of the proxy hierarchy it should fetch/prefetch the needed multimedia component/object. This will depend on several factors such as the current state of the network (both wired and wireless), the priority of the component/object to be fetched, and the type of the multimedia application. This implies that when a mobility–agent is created it is made aware of the proxy–hierarchy of the multimedia stream being delivered to its mobile client and it can find out the priority of multimedia objects. Different architectural options is explored to develop a scheme which minimizes the overhead of mobility–agent based scheme while maximizing its flexibility and benefits.

References

[1] M. Arlitt, R. Friedrich, and T. Jin. Performance evaluation of web proxy cache replacement policies. Technical report, HP Labs Technical Report HPL-98-97, May 1998.

[2] D. Barbara and T. Imielinski. Sleepers and Workaholics: Caching Strategies in Mobile Environments. *Very Large Databases Journal*, December 1995.

[3] P. Cao and S. Irani. Cost aware www proxy caching algorithms. In *Proceedings of USENIX Symposium on Internet Technologies and Systems (USITS)*, pages 193–206, 1997.

[4] P. Cao and C. Liu. Maintaing strong cache consistency in the world wide web. *IEEE Transactions on Computers*, 47(4):445–457, April 1998.

[5] L. Cherkasova. Improving www proxies performance with greedy-dual-size-frequency caching policy. Technical report, HP Labs Technical Report HPL-98-69, Nov 1998.

[6] A. Dingle and T. Partl. Web cache coherence. *Computer Networks and ISDN Systems*, 28:907–920, 1996.

[7] C. Dodge, B. Marx, and H. Pfeiffenberger. Web cataloguing through cache expolitation and steps towards consistency maintenance. *Computer Networks and ISDN Systems*, 27:1003–1008, 1995.

[8] L. Fan, Q. Jacobson, P. Cao, and W. Lin. Web prefetching between low–bandwidth clients and proxies: Potential and performance. In *Proceedings of ACM SIGMETRICS '99*, Atlanta, GA, May 1999.

[9] F. Fitzek and M. Reisslein. MPEG–4 and H.263 traces for network performance evaluation. Technical Report TKN–00–06, Technical University Berlin, Dept. of Electrical Eng., Germany, October 2000. Traces available at `http://www-tkn.ee.tu-berlin.de/~fitzek` and `http://www.eas.asu.edu/trace`.

[10] B. C. Housel, G. Samaras, and D. B. Lindquist. WebExpress: A client/intercept based system for optimizing web browsing in a wireless environment. *Mobile Networks and Applications*, 3:419–431, 1998.

[11] Q. Hu and D. K. Lee. Cache algorithms based on adaptive invalidation reports for mobile environments. *Cluster Computing*, 1:39–50, 1998.

[12] Z. Jiang and L. Kleinrock. A general optimal video smoothing algorithm. In *Proceedings of IEEE Infocom '98*, pages 676–684, San Francisco, CA, April 1998.

[13] J. Jing, A. Elmagramid, A. Helal, and R. Alonso. Bit-Sequences: An adaptive cache invalidation method in mobile client/server environments. *Mobile Networks and Applications*, (2):115–127, 1997.

[14] T. Johnson and D. Shasha. 2Q: A low overhead high performance buffer management replacement algorithm. In *Proc. of the 20th VLDB conference*, pages 439–450, 1994.

[15] J. Jung, D. Lee, and K. Chon. Proactive web caching with cumulative prefetching for large multimedia data. *Computer Networks*, 33:645–655, 2000.

[16] R. Koenen. MPEG–4 multimedia for our time. *IEEE Spectrum*, 36(2):26–33, February 1999.

[17] G. Y. Liu and G. Q. Maguire Jr. A Mobility-Aware Dynamic Database Caching Scheme for Wireless Mobile Computing and Communications. *Distributed and Parallel Databases*, (4):271–288, 1996.

[18] W. Ma and D. Du. Design a multiple–level video caching policy for video proxy servers. Technical report, Dept. of Computer Science and Engineering, University of Minnesota, March 1999.

[19] W. Ma and D. Du. Frame selection for dynamic caching adjustment in video proxy servers. Technical report, Dept. of Computer Science and Engineering, University of Minnesota, March 1999.

[20] W. Ma and D. Du. Proxy–assisted video delivery using prefixreissle caching. Technical report, Dept. of Computer Science and Engineering, University of Minnesota, March 1999.

[21] Z. Miao and A. Ortega. Proxy caching for efficient video services over the internet. In *Proceedings of 9th International Packet Video Workshop*, 1999.

[22] K. Obraczka, P. Dantzig, S. Arthachinda, and M. Yousuf. Scalable, highly available web caching. In *Submitted for Review*.

[23] E. J. O'Neil, P. E. O'Neil, and G. Weikum. The lru-k page replacement algorithm for database disk buffering. In *Proceedings of the 1993 ACM Sigmod International Conference on Management of Data*, pages 297–305, 1993.

[24] A. Puri and T. Chen. *Multimedia Systems, Standards, and Networks*. Marcel Dekker, New York, 2000.

[25] M. Reisslein, F. Hartanto, and K. W. Ross. Interactive video streaming with proxy servers. In *Proceedings of First International Workshop on Intelligent Multimedia Computing and Networking (IMMCN)*, pages II–588–591, Atlantic City, NJ, February 2000.

[26] R. Rejaie, M. Handley, and D. Estrin. Quality adaptation for congestion controlled video playback over the internet. In *Proc. of ACM SIGCOMM*, Cambridge, MA, September 1999.

[27] R. Rejaie, H. Yu, M. Handley, and D. Estrin. Multimedia proxy caching mechanism for quality adaptive streaming applications in the internet. In *Proc. of IEEE INFOCOM*, Tel Aviv, Israel, March 2000.

[28] J. Rexford, S. Sen, and A. Basso. A smoothing proxy service for variable–bit–rate streaming video. In *Proceedings of Global Internet Symposium*, December 1999.

[29] J. Rexford and D. Towsley. Smoothing variable–bit–rate video in an internetwork. *IEEE/ACM Transactions on Networking*, 7(2):202–215, April 1999.

[30] L. Rizzo and L. Vicisano. Replacement policies for a proxy cache. *IEEE/ACM Transactions on Networking*, 8(2), Apr 2000.

[31] S. Sen, J. Rexford, and D. Towsley. Proxy prefix caching for multimedia streams. In *Proc. of IEEE INFOCOM*, pages 1310 – 1319, New York, NY, March 1999.

[32] R. Tewari, H.M. Vin, A. Dan, and D. Sitaram. Resource–based caching for web servers. In *Proc. of SPIE/ACM Conf. on Multimedia Computing and Networking*, San Jose, 1998.

[33] Y. Wang, Z. Zhang, D. Du, and D. Su. A network–conscious approach to end–to–end video delivery over wide area networks using proxy servers. In *Proc. of IEEE INFOCOM*, pages 660 – 667, San Francisco, CA, April 1998.

[34] S. Williams, M. Abrams, C. Stanbridge, G. Abdulla, and E. Fox. Removal policies in network caches for world-wide web documents. In *Proceedings of the ACM Sigcomm96*, 1996.

[35] R. Wooster and M. Abrams. Proxy caching the estimated page load delays. In *Proceedings of the 6th International World Wide Web Conference*, 1997.

[36] N. Young. Online caching as the cache size varies. In *2nd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 241–250, 1991.

[37] D. Zhong and S. Chang. An integrated approach for content–based video object segmentation and retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(8):1259–1268, December 1999.