January 1, 2004

Cost-based cache replacement and server selection for multimedia proxy across wireless Internet

Q Zhang
Z Xiang
WW Zhu
LX Gao

Cost-Based Cache Replacement and Server Selection for Multimedia Proxy across Wireless Internet*

Qian Zhang† *Zhe Xiang*† *Wenwu Zhu*† *Lixin Gao*‡

† Microsoft Research Asia, No. 49 Zhichun Road, Haidian District, Beijing, China, 100080

‡Electrical and Computer Engineering, University of Massachusetts, Amherst, MA 01002

ABSTRACT

Multimedia proxy plays an important role in multimedia streaming over wireless Internet. Since wireless network exhibits different characteristics from the Internet, multimedia proxy caching over wireless Internet faces additional challenges. In this paper, we present a study of cache replacement for single server and server selection for multiple servers across wireless Internet. By considering multiple objectives of multimedia proxy, we design a unified cost metric to measure proxy performance in wireless Internet. Based on the defined unified cost metric, we propose a novel replacement algorithm for single server and a new server selection policy for multiple servers to improve the end-to-end performance such as throughput, media quality, and start-up latency. To effectively handle errors occurred on wireless link, channel-adaptive unequal error protection (UEP) is deployed according to distinct QoS (quality of services) requirements of layered or scalable media. Simulation results demonstrate that our approaches achieve significantly better performance than the known cache replacement algorithms and sever selection schemes, respectively.

Index terms- caching, multimedia proxy, replacement policy, streaming media, server selection, wireless Internet.

EDICS: 5-BEEP, 6-WORK

* Prof. Lixin Gao performed part of this work when she was a visiting researcher at Microsoft Research Asia.

I. INTRODUCTION

The advent of explosive growth in the Internet and dramatic increase in wireless access has accelerated the development of multimedia applications over wireless Internet. The key challenges in deploying multimedia application over wireless Internet are Quality of Service (QoS) requirements of multimedia applications, the demand on network resources, and the lack of reliability of wireless channel. To alleviate network congestion, reduce latency and workload on multimedia servers, multimedia proxy has been proposed to cache popular contents in proxy located close to client side and server replication has been used to enhance the streaming service performance. The effectiveness of the proxy server architecture depends largely on cache replacement policy and the server selection algorithm.

Most of the wireless multimedia proxies proposed in the literature are designed as a rate control module for relaying streaming [1] or a transcoder to perform format transferring [2][3]. Considering the inherent unreliability of the wireless channel that results from fading or shadowing effects, it is essential to adopt efficient error control scheme to cope with errors in wireless channel. This unique requirement affects the proxy replacement policy. Media replication is a widely used technique in the Internet for improving streaming service performance and reducing network load. A good server selection scheme can significantly improve the client's performance. Traditional server selection algorithms focus on decreasing latency for non-continuous contents such as text and image [4-7]. These techniques are not suitable for continuous media streaming applications.

In this paper, a multimedia proxy, which resides in the base station or the gateway, is proposed for multimedia streaming over wireless Internet. Considering the multiple objectives of multimedia proxy, we design a unified cost metric, which takes the network, latency, and media distortion into account, on measuring the performance of the proxy. Based on the cost metric, a novel replacement algorithm for single server and a new server selection policy for multiple severs are proposed to improve the proxy performance by saving network bandwidth, reducing latency, and improving media quality. To handle errors in wireless channel, computation cache concept is introduced for reducing the computation complexity using channel adaptive error protection [10]. Moreover, a collaborative parameter collecting method and a probability based probing scheme are designed in our proposed server selection policy to measure network and server load proactively. In addition, scalable video is used to demonstrate the effectiveness of our proposed scheme.

The rest of this paper is organized as follows. In Section II, we present an architecture for multimedia proxy across wireless Internet. A computation caching strategy is proposed to provide channel protection for unreliable wireless channel while saving computation resource of the proxy. In Section III, a cost-based cache replacement policy, which is capable of achieving multiple proxy

objectives and adapting network bandwidth variation, is proposed for the single server environment. Section IV presents an extended cache replacement policy and a cost-based server selection algorithm for the environment in which content are replicated in several video servers. In Section V, simulation results are given to demonstrate the performance of our cache replacement policy and server selection scheme under varying network conditions. Finally, Section VI concludes the paper.

A. Related work

To date, most of work on multimedia proxy has been focused on the Internet. Issues addressed are cache management [8-9], prefetching or prefix caching [10-11], and cache replacement [8-9] [12]. It is known that the key effectiveness of proxy cache is determined by the performance of its replacement policy. LRU (Least Recently Used) [13], LRU-Threshold [14], and LFU (Least Frequently Used) [15] are widely used for web data caching. For continuous media, Rejaie et al. introduced a replacement policy for layered-media [8]. Moreover, a GreedyDual-Size replacement scheme is proposed in [16], which considered file size, latency, and network cost. For both continuous and non-continuous media, Tewari et al. proposed a resource based caching (RBC) policy to balance the usage of cache space and disk I/O [12]. All of those mentioned replacement algorithms use hit rate or byte hit rate as the performance metric without addressing network cost or latency, which are important for multimedia proxy caching. Consequently, Yu et al. presented a priority-based replacement policy for both continuous and non-continuous content [9]. Note that the above schemes employed a potential cost function derived from different factors. However, none of them had a comprehensive analysis for the factors that essentially affect the performance of multimedia proxy cache in wireless Internet.

Most studies on multimedia server selection to date focus on the load balancing issue of distributed media servers in the Internet. In [17-18], server selection is performed to balance load among video servers and minimize the time spent for service request. However, there still exist several problems not addressed. For example, the network characteristics, such as best-effort nature of Internet and error-prone nature of wireless, affect QoS of streaming media. The effect of Internet transmission on server selection hasn't been studied extensively. To the best of our knowledge, there is no reported work on server selection in the content of wireless Internet and there is only one work studying the impact of network condition on server selection in multimedia environment [17], where an algorithm combining path selection with server selection was proposed. However, detailed multimedia characteristics were not considered and path selection is not applicable for the wireless Internet.

II. PROBLEM FORMULATION

Figure 1 illustrates a scenario of a multimedia proxy in wireless Internet. All content for scaleable video are stored at the video servers across Internet to support streaming service for end clients. When the client requests the video streaming, the traffic between client and remote video server is always routed through the multimedia proxy. Thus, the proxy is able to intercept each streaming and cache it in its storage. As shown in Figure 1, the multimedia proxy is located at the edge of Internet connecting both remote servers and end clients. On the proxy-server side, the backbone network between proxy and server is a best-effort network, i.e., the network conditions such as bandwidth, packet loss ratio, delay and jitter vary from time to time. On the proxy-client side, two types of clients access proxy via different network. Internet clients access proxy via LAN, x-DSL or the like. Since multimedia proxy is very near to the end clients, the network status is rather stable for Internet clients. In contrast, wireless clients connect proxy via wireless channel, such as W-LAN (wireless local area network), wireless wide area network (W-WAN) like GPRS and 3G, etc. In general, the wireless channel exhibits time-varying characteristics resulting in unreliability and varying bit error rate.

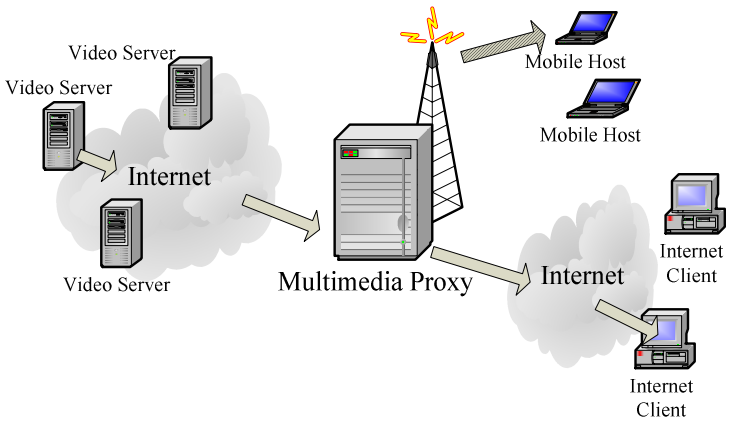

Figure 1. Multimedia Proxy across wireless Internet.

The clients always send their requests for a particular video to the multimedia proxy. In order to provide efficiently streaming service for both Internet and wireless clients, the following questions should be addressed in the proxy:

1) How to provide high quality video streaming service for both Internet clients and wireless clients?
2) How to manage limited cache resource and computation resource in multimedia proxy to achieve high performance?

3) How to evaluate and select video replicas in the Internet to relay streaming for end clients?

To address the above questions, we depict the diagram for our proposed multimedia proxy across wireless Internet in Figure 2. The key components of this diagram are the *computation cache*, *caching management* and *server selection* modules. We describe those modules in detail as follows.

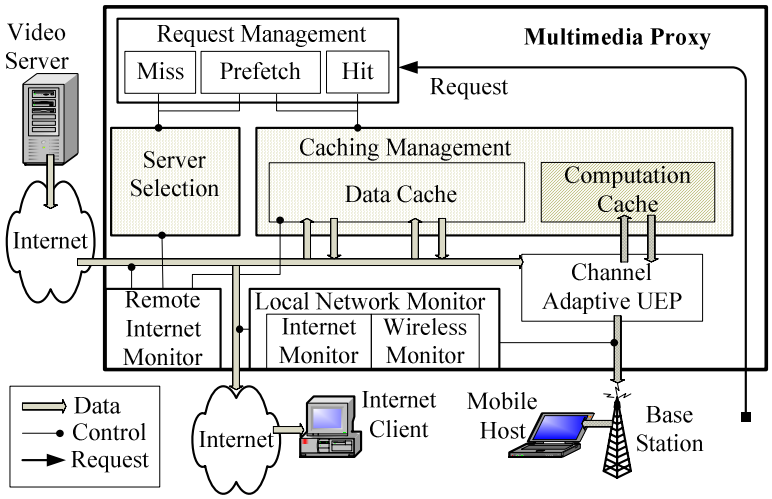

Figure 2. Diagram of multimedia proxy in wireless Internet.

To handle errors occurred in the wireless channel, *channel adaptive UEP* scheme is used considering the different QoS requirements of different layers of scalable video [19]. In [19], unequal error protection is applied to different layers of video according to the distortion and dependency among each layer. Rate-distortion (R-D) based resource allocation is performed by allocating available resources between source and channel protection to achieve best video quality. However, the complexity of generating R-D based UEP channel coding is high. As a result, the computation burden is in-tolerable for serving a large amount of wireless clients simultaneously in multimedia proxy. To alleviate this problem, in this work, we cache the popular redundant data in the proxy so as to mitigate the computation overhead for the proxy. Since this portion of cache is used to reduce the computation overhead, we call it *computation cache*. Besides the *computation cache*, the rest part of cache is served for caching the popular multimedia objects, which is named as *data cache*. It is obvious that the computation cache can improve the media quality for wireless clients; meanwhile, it competes with *data cache* for the limited cache resource. Thus, it is a challenging task to determine the distribution of *data cache* and *computation cache* to achieve high performance for the overall system.

In *caching management* module, a cost-based replacement algorithm is proposed, which associates a cost value called cache gain with each video object. Cache gain indicates the object's contribution of proxy to achieve high performance. The video object with higher cache gain has higher probability to be cached in the proxy. When the local storage at the proxy is full, the proxy removes the video object with the lowest cache gain from the cache to make room for the coming objects. To provide high QoS for client thus to improve the revenue for proxy, the replacement policy should balance the three aspects: network cost for fetching this object, startup latency cost for getting this object, and the media distortion cost for caching this object. The distortion cost can be further divided into source-distortion cost and channel-distortion cost for wireless clients. Note that how much protection data should be cached in proxy is also determined by cost-based replacement policy according to the cost of caching these redundant data object. Furthermore, considering the varying network conditions, it is important that replacement policy should be able to adapt to the time-varying network resources.

If the requested video is replicated in several video servers, it is essential for proxy to select an appropriate server for miss or prefetch requests. The *server selection* module in this architecture is adopted to perform this type of selection. Since the status of the network connections between proxy and different servers vary dramatically, server selection algorithm has a significant impact on the performance of on-going request. First, the perceived video quality at client side is determined by the network resource between server and proxy. Second, the distance between proxy and server also affects the start-up latency and network transmission cost. Third, due to the shared nature of network resource, the algorithm also has underlying impact on future requests from other clients when allocating shared resource among different requests.

Take the varying network condition between proxy and server(s) as well as the varying network condition between proxy and client(s) into consideration, another two important modules, *remote Internet monitor* and *local network monitor*, are introduced. *Remote Internet monitor* is used to monitor and estimate the Internet condition between proxy to remote video server, such as latency, available bandwidth, etc. The estimated information is used by *caching management* module and *server selection* module to perform cache replacement and select remote video server. The *local network monitor* is composed of *Internet monitor* and *wireless monitor*, which monitor the local Internet and wireless channel condition, respectively. Specifically, *wireless monitor* is also responsible to measure and predict the BER (bit error rate) of wireless channel which is crucial for *channel adaptive UEP* scheme. The detail of how those network monitors work can be found in [19][20].

III. COST-BASED CACHE REPLACEMENT IN SINGLE SERVER ENVIRONMENT

In this section, a revenue model is first introduced to analyze the performance of proxy caching, in which three performance metrics, video quality, network cost saving, and start-up delay, are discussed. Then, a cost-based replacement scheme is proposed to achieve multiple objectives requirements for multimedia proxy.

A. Multiple Objectives for Multimedia proxy

Replacement algorithm for proxy caching plays an important role in cache management. Traditional replacement algorithm aims at improving the hit ratio (HR) or byte hit ratio (BHR) of caching, thus using those metrics to measure the effectiveness of caching replacement policy. However, these metrics are not suitable for evaluating multimedia proxy caching, because multimedia proxy caching exhibits some characteristics and objectives, which are explained as follows.

First, different portion of continuous media, such as video and audio, have different characteristics, thereby resulting in different quality impacts. For example, the higher layers of scalable video cannot be decoded if their lower layers haven't been successfully received. Thus, the base layer is more important than the higher layers objects and provides higher contribution to video quality. In multimedia caching, proxies should preemptively cache objects that provide higher contribution for overall media quality. In this work, the lower layers have higher probability to be cached in the proxy.

Second, multimedia proxy caching can greatly reduce the consumption of network resource by aiming at reducing the throughput of Internet transmission between proxy and remote video server. Under the time-varying network conditions, requests may be blocked in some hot links due to the competitive access among different clients. Therefore, we need proxy to efficiently tradeoff both proxy caching resource and network resource to achieve high performance. Note that for scalable video discussed in this work, if the base layer of video cannot be transmitted to the client, we call that request is blocked.

Third, it is useful to cache video prefix in order to reduce startup latency for streaming service. In general the end-to-end latency between remote video server and clients are considerable large. To handle possible congestion problem in the Internet, buffer is usually used to cache a reasonable amount of streaming data on the client side. This further increases startup latency for client. On the contrary, multimedia proxy is deployed near to the client so that the delay between proxy and client can be fairly small. By caching video prefix in proxy, the request

of client can be responded promptly, and proxy can also prefetch the rest part of streaming data from remote server simultaneously so as to reduce the perceived startup latency for client.

In conclusion, the multimedia proxy has multiple performance objectives. However, these objectives compete with each other for cache usage. To date most of criteria for multimedia caching use separate metric alone. In this paper we unify multiple objectives and present a revenue model to measure the performance of proxy. The *revenue* of each request for proxy is determined by three factors: video quality perceived by end client, startup latency endured by end client, and network resource consumed by serving the request. The goal of cache management is to maximize revenue in multimedia proxy.

B. Video Quality Revenue Rate

In order to provide high quality video for end client, it is essential to cache video object with high quality in proxy. In this subsection, we define a metric to measure the quality of each video object to improve the perceived video quality for end clients.

For a client, packet loss in Internet or bit error in wireless channel results in the loss of video objects during transmission, therefore remarkable video distortion is induced at client end. In this work, we use *RMSE* [23] to measure how much distortion between original video and distorted video is. For scalable video, video object in different layers have different video quality impact. More specifically, the loss of lower layer video objects results in higher video quality distortion, thus yielding higher *RMSE*.

We address the quality metric from the system point of view. Usually, different video objects have different request frequencies. Video objects with higher request frequencies provide more quality revenue for end client than those with lower request frequencies. Thus, we introduce video quality revenue rate Rev_Q of a video object v as follow

$$Rev_Q(v) = RMSE(v) \times f(v), \quad (1)$$

where $RMSE(v)$ represents the quality of video object and $f(v)$ is the request frequency of the video object.

To support wireless clients, computation caching is designed in this work as mentioned above. In general, caching redundant video object can recover errors occurred in the wireless channel so as to improve video quality for wireless clients. Therefore, the redundant video object in the computation cache also has video quality revenue. Given a redundant video object v_R, we define the video quality revenue rate

$$Rev_Q(v_R) = RMSE(v_R) \times f(v_R). \quad (2)$$

Given the population of wireless client among all clients, $f(v_R)$ can be

$$f(v_R) = f(v) \times \lambda_{wireless}, \quad (3)$$

where $\lambda_{wireless}$ indicates the proportion of the wireless clients to all clients.

Note that video quality revenue rate represents how much contribution each video object provides for end clients, either for cached objects or non-cached objects. In order to improve the perceived video quality for end clients, caching video object with high video quality revenue rate is an important goal for multimedia proxy. Motivated by this fact, we propose a cache management scheme taking the video quality revenue rate of each video object into account. Considering that reducing network resource consumption and decreasing startup latency are another two important objectives for multimedia proxy, we will study the network saving revenue rate and startup latency revenue rate in the follow two subsections.

C. Network Saving Revenue Rate

In order to reduce the consumption of network resource, it is useful to cache the video object with high network saving revenue. We divided the network saving revenue into two sub-revenues: throughput revenue and network utilization revenue.

C.1 Throughput revenue rate

Considering the huge data size of video streaming, it is essential for proxy to cache video objects to reduce throughput consumption of network.

Given a video object v, we define the throughput revenue for this object as the cost of fetching it from server to proxy, which is a function of the size of the object and the distance (or number of hops) between the corresponding server and proxy. Based on the user request pattern, the throughput revenue rate for the video object v can be presented as

$$Rev_T(v) = Size(v) \times Dist(RTT(v)) \times f(v), \quad (4)$$

where $Size(v)$ is the size of the video object, and $Dist(RTT(v))$ is the distance between remote server and proxy, which is a function with respect to round trip time (RTT).

Note that for the redundant video object cached in proxy, it does not have throughput revenue because the redundant object is generated in proxy and does not consume the network resource between proxy and remote server.

C.2 Network utilization revenue rate

Internet network resource is a competitive resource and the utilizations of different links also vary greatly. For example, the video servers containing more popular video will attract more clients, thus the utilizations of those hot links are usually very high. In those hot links, due to the resource constrains, some requests may be blocked if there is not enough resource to meet the basic demand of base layer transferring. On the other hand, cold links with few requests often have sufficient network resource.

Proxy caching is helpful for balancing the utilization of difference network links. Caching video object from hot links not only lowers the link utilization but also decreases the requests blocking rate. On the contrary, the requests for video from cold link are served directly from remote server thus improving the utilization of cold link without consuming expensive caching resource. Given a video object v, network utilization revenue rate $Rev_U(v)$ for video object v is defined as

$$Rev_U(v) = Size(v) \times \mu(v,t) \times f(v), \tag{5}$$

where the network utilization $\mu(v,t)$ for video v at time t and it is defined below.

Suppose video object v is originally stored at server S_j, the network utilization $\mu(v,t)$ is determined by the average required bandwidth and average available bandwidth of the link between proxy and server as follows.

$$\mu(v,t) = \exp[(-1) \times (\frac{\overline{BW_A(j,t)}}{BW_R(j)})^{\alpha}], \tag{6}$$

where $\overline{BW_A(j,t)}$ is average available bandwidth at time *t*, $BW_R(j)$ is the average required bandwidth for this link, and α is the control parameter that is set to 3.0. We use exponential function in (6) to limit the value of utilization in the range of [0, 1]. As the link available bandwidth $\overline{BW_A(j,t)}$ varies with time greatly, we measure the value of link available bandwidth periodically and use timely value for accurate estimation.

The average available bandwidth is estimated by Internet network monitor. In order to avoid network fluctuation, we use weighted network measurement to calculate the average available bandwidth mathematically, we have

$$\overline{BW_A(j,t)} = \overline{BW_A(j,t-T_k)} \times \beta + BW_A(j,t) \times (1-\beta), \tag{7}$$

where T_k is bandwidth measurement cycle time, $\overline{BW_A(j,t-T_k)}$ is average available bandwidth calculated at the last measure cycle, $BW_A(j,t)$ is available bandwidth measured at

time *t*, and β is a weighting parameter. Since the proxy caching aims at long time performance, we set bandwidth measurement cycle to 10 minutes, and the weight parameter β is set to 0.7.

The average required bandwidth $BW_R(j)$ is determined by user access pattern. Given a user access pattern, the average required bandwidth is composed of consumed bandwidth of all video objects originally stored in the server. For example, the required bandwidth of server S_j is composed of required bandwidth of all media contents in server S_j, i.e.,

$$BW_R(j) = \sum_{if\ v in S_j} (f(v) \times Size(v)). \quad (8)$$

Similar to throughput revenue rate, the redundant video objects have no network utilization revenue rate because no network resource is consumed between proxy and server.

Given definition of throughput revenue rate and network utilization revenue rate, we can further define the totally network saving revenue rate $Rev_N(v)$ for the video object v as

$$Rev_N(v) = Rev_T(v) + Rev_U(v). \quad (9)$$

Note that in our work, the multimedia proxy calculates the network saving revenue rate for both cached and non-cached video objects.

D. Startup latency revenue rate

As mentioned earlier, prefix caching is an effective way to reduce startup latency for end clients. In order to measure the performance of prefix caching and decide which prefix should be cached in proxy, we study the startup latency revenue rate in this section.

In general, latency between remote video server and proxy dominates end-to-end latency because multimedia proxy is deployed at the edge of network that is near to clients. Here, if we cache the prefix of video content with sufficient length, e.g., L_{td}, the streaming service can be started directly from nearby proxy no matter how much latency is between remote server and proxy. The prefix cache can significantly reduce the perceived startup latency for end users. Without loss of generality, we assume that the distance between the client to the proxy is much smaller than the one between the proxy and the server. That is to say, to make the analysis easier, in our work we can ignore the distance between the proxy to the client.

Given a video object v, the startup latency revenue rate is determined by whether the object belongs to prefix or not, which can be defined as

$$Rev_L(v) = \begin{cases} Delay(RTT(v)) \times f(v) & if\ v \in Prefix \\ 0 & else \end{cases}, \quad (10)$$

where $Delay(RTT(v))$ is the delay for delivering prefix from server to proxy, which is a function of RTT between server and proxy. We assign the same value of startup latency revenue rate for prefix object in any video layer. But we do not treat those objects equally in our cache replacement policy since there are dependencies among different layers of objects, e.g., the object in the higher layer can not be decoded unless the ones in the lower layers are properly received. The object dependency is considered in replacement policy in the next subsection.

Since the redundant video objects cached in proxy do not consume the network resource between proxy and server, they do not have startup latency revenue rate.

E. Cost-based replacement policy for multimedia proxy

Given the definition of revenue rate for video quality, network saving, and startup latency, we now define the total revenue rate, $\mathrm{Re}v(v)$, for video object v, as

$$Rev(v) = p_Q \times \frac{Rev_Q(v)}{Rev_Q^M(v)} + p_N \times \frac{Rev_N(v)}{Rev_N^M(v)} + p_L \times \frac{Rev_L(v)}{Rev_L^M(v)}, \tag{11}$$

where $Rev_Q^M(v)$, $Rev_N^M(v)$ and $Rev_L^M(v)$ are constants representing the maximized value of video quality revenue rate, network cost saving revenue rate, and startup latency revenue rate, respectively. We use those constants to normalize each revenue rate. Then we can choose the value for p_Q, p_N and p_L, which are the weighting parameters that stand for the unit price of video quality revenue, network saving revenue, and startup latency revenue, respectively. Unfortunately, it is usually hard to find a theoretical method to choose the value of these weighting parameters because there is no explicit correlation with those three revenue rates. However, we can choose the value of each unit price practically depending on application requirements. For example, the streaming service provider can choose the value of weighting parameters proportional to the unit price for network consumption he will pay and unit prices of video quality as well as startup latency the clients will pay. If the clients will pay more for video quality, the service provider can increase the weight parameter of video quality revenue rate accordingly, vice versa.

For redundant video object v_R, we define its total revenue rate $Rev(v_R)$ as

$$Rev(v_R) = p_Q \times \frac{Rev_Q(v_R)}{Rev_Q^M(v_R)}, \tag{12}$$

where p_Q is the weighting parameter that stand for the unit price of video quality revenue rate and $Rev_Q^M(v_R)$ is a constant that represents the maximal value of video quality revenue rate. Since the redundant video objects generated in proxy do not consume the network resource between proxy and server, the redundant video object has no revenue rate for networking saving and startup latency.

To achieve the maximal revenue rate for multimedia proxy, in this work we propose the cost-based replacement policy. Considering the limited caching size, we associate each object v_o, ($v_o \in \{v, v_R\}$), with a cache gain, $Gain(v_o)$, as follow.

$$Gain(v_o) = \frac{Rev(v_o)}{Size(v_o)}. \tag{13}$$

Our cost-based replacement algorithm is to cache video object with the highest cache gain. As illustrated in Figure 2, for a new requested object, if the proxy is not full, the object is simply cached in the proxy. On the other hand, if the proxy is full, the object with the lowest cost will be removed. This operation stops only when the cost of the requested object is lower than those of all the objects in proxy.

Note that, considering the inter-media relation, there may be dependency among different requested objects. This dependency of objects is taken into account in our replacement policy. For example, the video objects in higher layers depend on the corresponding lower layer objects, and redundant video objects depend on original video objects. If the dependent video objects are not cached in proxy, new video object can not be replaced into proxy.

IV. COST-BASED CACHE REPLACEMENT AND SERVER SELECTION IN MULTIPLE SERVER ENVIRONMENT

In this section, we will discuss the scenario that the requested video is replicated across several video servers. Note that the replication environment has notable impacts on multimedia proxy across wireless Internet. First, when proxy can not provide streaming directly with cached video object, it should select a remote server to relay streaming for client. Because video replicas exhibit different QoS characteristics, an efficient server selection algorithm is needed for multimedia proxy to achieve high performance. Second, due to distributed replicas of video content, the cost and revenue achieved from the same video may vary with the location of replica. To achieve the maximal revenue rate for proxy, it is essential for replacement policy to take replication information into account. Consequently, it introduces a challenging task for

multimedia proxy to do server selection. In this section, a cost-based server selection algorithm is proposed to efficiently perform server selection for client across wireless Internet in which an extended cost-based replacement policy for replication environment is introduced.

A. Cache replacement policy

As discussed in section III, our cost-based replacement policy caches the video object with highest cache gain to achieve the highest revenue rate for proxy. However, due to the replication of video, the revenues defined in section III may not suitable for replication environment.

Recall the network utilization revenue rate defined in Equation (5), it is proportional to the network utilization of the link between proxy and original server. However, if the video is replicated across several video servers, a cached video object has relations with more than one Internet link. To cope with the replication environment, we extend the network utilization revenue rate for the video object v as

$$Rev_U(v) = BW(v) \times \phi(v,t), \tag{14}$$

where $\phi(v,t)$ is the integration network utilization for video v at time t. The integration network utilization $\phi(v,t)$ can be further calculated as

$$\phi(v,t) = \frac{\sum_{if\ v in S_j} (\overline{BW_A(j,t)} \times \mu(j,t))}{\sum_{if\ v in S_j} \overline{BW_A(j,t)}}, \tag{15}$$

where $\overline{BW_A(j,t)}$ is the average available bandwidth at time *t*, and $\mu(j,t)$ is network utilization of the link between proxy and server S_j.

Based on Equation (14) and (15), the video with fewer replicas gets a higher network utilization revenue rate. Therefore, the video with fewer replicas has more chance to be cached in proxy. Moreover, for a given video, if most links have higher network utilization, the corresponding network utilization revenue rate is higher. Consequently, the video from hot link has high probability to be cached in proxy.

B. Server selection algorithm

To achieve maximal revenue rate for multimedia proxy under replication environment, we propose a cost-based server selection algorithm in this section.

Upon receiving a request from client, the proxy first determines whether the requested video is fully cached in proxy or not. If there is fully copy of video in cache, the streaming is pumped directly from cache. Otherwise, the proxy will forward the request to an appropriated original

server to fetch or pre-fetch video needed for the client. In general, the goal of our proposed server selection system is to improve the performance of requested client. As discussed above, the client performance is measured as the revenue of proxy, which is composed of three aspects: perceived video quality at client side, startup latency of client, and networking cost saving. The multimedia proxy measures the potential revenue of each candidate video server for the current request.

For a given request, suppose the requested video is v, replicas of v are stored in video server $\{S_j\}$, where $j=(1,2,...,m)$. Then, we define the potential revenue for any server S_j as:

$$Revenue_S(S_j,v)=\sum_{all\ v\in Cache}(p_Q\times\frac{Revenue_Q(v)}{Revenue_Q^M(v)}-p_L\times\frac{Revenue_L(v)}{Revenue_L^M(v)} - p_N\times\frac{Revenue_T(v)}{Revenue_T^M(v)}), \quad (16)$$

where $Revenue_Q(v)$, $Revenue_L(v)$, and $Revenue_T(v)$ are the revenues for video quality, startup latency and network throughput saving, respectively, and $Revenue_Q^M(v)$, $Revenue_L^M(v)$ and $Revenue_T^M(v)$ are constants representing the maximized value of video quality revenue, network cost saving revenue, and startup latency revenue, respectively. These constants are used to normalize different revenues. Here, p_Q, p_N, and p_L are the weighting parameters that stand for the unit price of video quality revenue, network saving revenue, and startup latency revenue, respectively.

Similar to the definition of corresponding revenue rate in Section III, the definitions of these sub-revenues are as given as

$$Revenue_Q(v)=RMSE(v),$$

$$Revenue_T(v)=Size(v)\times Dist(RTT(v))$$

$$Revenue_L(v)=\begin{cases}Delay(RTT(v)) & if\ v\in Prefix\\ 0 & else\end{cases}. \quad (17)$$

It is known that network resource such as bandwidth is competitive resource among different requests, and server selection consumes network resource from server to proxy. Thus, it is necessary for server selection to perform load balance for consumed network resource. To avoid the overload for hot link while improving utilization of cold link, we define network utility revenue similar to (5) to revise the potential revenue:

$$Revenue_U(S_j,v)=\frac{Rev_U(v)}{Rev_U^M(v)}\times p_N, \quad (18)$$

where $Rev_U(v)$ is defined in Equation (14).

Then we assign each candidate server with a selection gain as follows.

$$SelectionGain(S_j, v) = \gamma \times Revenue_S(S_j, v) + (1-\gamma) \times Revenue_U(S_j, v). \quad (19)$$

Note that the calculation of selection gain depends on the measurement of network condition, e.g., the available network bandwidth measured by *Internet Monitor* as illustrated in Figure 2. Too frequent network measurements increase workload to proxy; while too sparse measurements decrease the calculation precision. Thus, we need to design an algorithm for efficient network condition measurement.

Generally, it can be observed that for a given period of time, the distribution for some network–related parameters, such as available bandwidth, RTT, changes slowly. In order to save the expense of network measurement, we propose a probability-based selection scheme in our selection algorithm as follows.

Mathematically, given a set of candidate servers, only part of them need to be re-measured when a new request arrival. For a certain server, the re-measurement probability is subject to its historic performance. We sort the candidate servers according to their historic performances in a descent sequence as $\{S_1, S_2, \mathsf{K}, S_n\}$, and then measurement probability for server S_j can be calculated as

$$\Pr(S_j) = \min\left\{ 1, \lambda \times \frac{LastSelectionGain(S_j)}{\max_{i \le n}\{LastSelectionGain(S_i)\}} \right\}. \quad (20)$$

Here λ is a weighting parameter that determines measurement scope, and $LastSelectionGain(S_j)$ represents historic performance.

We also define an expiration period T_e for each server S_j. If the historic parameters of one server hadn't been calculated in the past period T_e, they are considered as stale and should be re-measured in the current competition as follows.

$$\Pr(S_j) = 1 \qquad (if\ \ Duration(S_j)) \ge T_e). \quad (21)$$

By assigning each candidate server with a measurement probability, we re-measure network and renew selection gain for the candidate server.

In this work, we use our proposed TCP-friendly protocol, MSTFP, for available network estimation [20]. Therein, we send probing packets regularly, the MSTFP estimates the available network bandwidth based on the measured RTT and estimated packet loss ratio.

After the network measurement, the server selection gain is calculated. At last, the server with maximum selection gain is then chosen for end client.

V. SIMULATION RESULTS

The simulation is to demonstrate that (1) our cost-based replacement scheme outperforms any other replacement schemes in comparison with perceived video quality, startup delay, and network cost saving; (2) our cost-based server selection policy achieves highest performance among known algorithms; (3) our proposed proxy caching system can adapt to the environment variation, such as networking bandwidth, wireless Internet client population, and BER of wireless channel, fairly well.

A. Simulation Setup

A.1 Media Distribution

In our simulations, we suppose that there are totally 2000 video clips in our system. Those video are encoded using MPEG-4 multi-layer scalable PFGS (Progressive Fine Granularity Scalable) [21] format. PFGS source coder encodes those 1,700Kbps video into two layers: one is the base layer (BL) that carries the most important information; the other is the enhancement layer (EL) that carries less important information. The enhancement layer can be further cut arbitrarily at any point adapts to the network bandwidth. For the sake of simplification, we divide enhancement layer into 3 sub-layers. Together with the base layer, we use 4-layers-video in our proxy system. We suppose the average video length is 1,800 seconds. As mentioned in Section III, we divide each video clip into video objects. In this simulation, we set the length of each video object as 10 seconds, each video object only belongs to one certain video layer. Thus, a 1,800 seconds video clip is composed of $4 \times 1800/10=720$ video objects.

We run our simulation in two cases, one is single-server case and the other is multiple-server case. In the multiple-server case, the number of replicas of each video clip is ranged from 1 to 4, and those replicas are randomly distributed to 20 video servers.

A.2 Network Conditions

Multimedia proxy is allocated at the edge of Internet. Considering the heterogeneity of network, we assume the bandwidth from each video server to proxy ranges from 1.6Mbps to 64Mbps, and the round trip time (RTT) of each link ranges from 5 ms to 10 ms. We also fluctuate each link's bandwidth from 50% to 150% of its normal value in period of 24 hours to simulate the day-night bandwidth variation.

Two types of clients, wireless clients and Internet clients, request video from proxy. Generally, we set the proportion of wireless clients be 50% of total clients. Due to the bandwidth

limitation of wireless channel, we assume the average wireless channel bandwidth is 800Kbps, and the mean BER of wireless channel is 0.7%. We also vary these parameters in our simulation to test the adaptation of proxy system. On the contrary, the Internet clients are near to the proxy located at the edge of Internet. We assume there is sufficient bandwidth to support media streaming from proxy to those Internet clients.

A.3 User Access Pattern

Our proxy simulation system is request-driven systems. Assuming the mean request frequency of our system is 0.4 requests per second, we use Poisson distribution to calculate the inter-arrival time of requests for generating requests. In order to simulate the behaviors of clients choosing video, we use Zipf distribution [22] to formulate the video popularity. Suppose the video clips are sorted in descending order of their popularity as $c_1, c_2 ... c_k$, the popularity of content c_i is r_i where the values of popularity are normalized, that is $\sum r_i = 1$, thus we have $r_i = i^{-(1-\alpha)} / C$, where $C = \sum i^{-(1-\alpha)}$ and α is a control parameter called skew factor. The commonly used skew factor is 0.271 for multimedia on-demand service [22].

A.3 Other parameters

Other important parameters used in our simulation are the weighting parameter p_Q, p_N, and p_L which stand for the unit price of video quality revenue, network saving revenue, and startup latency revenue rate, respectively. As mentioned in Section III, those parameters are chosen practically depending on application requirement. In our simulation, we set all these parameters to 1 because we assume these three objectives are all equal important and have the same priority in our proxy.

We summarize the parameters used in our simulation in Table I. In each of our simulation, we run the simulation for about 144 hours. The first 72 hours is to "heat" the system to a normal status, and the second 72 hours is used to study the system performance.

TABLE I. PRAMETERS USED FOR SIMULATION

Parameters	**Default Value**	**Range**
Number of media	2000	N/A
Media Length (sec)	1800	N/A
Media Rate (Kbps)	1734	N/A
Server-Proxy Bandwidth (Gbps)	N/A	1.5-64
Server-Proxy RTT (ms)	N/A	5-10
Request Frequency (1/sec)	0.4	N/A
Skew Factor	0.271	N/A
Wireless Channel Bandwidth (Kbps)	800	N/A

Internet Client Bandwidth (Kbps)	1750	N/A
Wireless BER (%)	0.7	N/A
Wireless Client Proportion (%)	50	30-80
Proxy Cache Size (Gb)	400	150-750
Prefix Length (sec)	30	N/A

B. Performance of cost-based replacement algorithm for single server

To evaluate the performance of our cost-based replacement algorithm, we compare it with (1) LRU (Least Recently Used) algorithm [13], (2) LRU-2 algorithm [14], and (3) LFU (Least Frequently Used) algorithm [15]. When a new object is need to cache into proxy, LRU algorithm replaces the least recently used object while LFU algorithm replaces the object with lowest request frequency. LRU-2 algorithm is a tradeoff of LRU and LFU algorithms. It keeps track of last two access time for each object and uses a compromise value as the replacement criteria. In this section, we compare performance of perceived video quality, startup latency, and network cost saving of each policy.

Figure 3 illustrates perceived video quality of Internet clients using different testing algorithms. With increasing of caching size, the video quality is improved accordingly in each policy. It can be seen that our cost-based replacement algorithm outperforms all other algorithms. The high performance that cost-based algorithm achieves is mainly due to the fact that video characteristics such as dependency and video distortion are taken into account by the proxy cache management. Furthermore, by adopting network utilization revenue rate into cache gain, the limited network resource is more efficiently used in cost-based algorithm than in others.

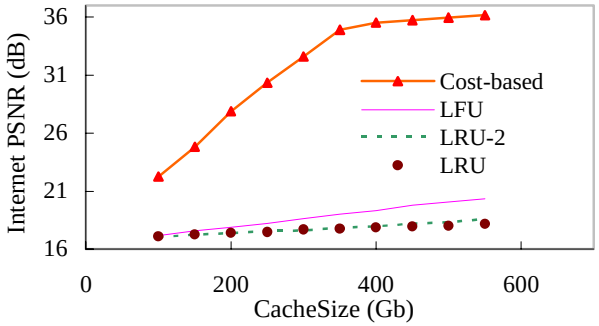

Figure 3. Comparison of replacement algorithms on video quality for Internet clients.

Figure 4 shows the perceived video quality of wireless clients with different algorithms. It can be seen that the video quality of wireless clients is lower than that of Internet clients. This is because that wireless channel has a smaller (<800Kbps) bandwidth and a higher BER (>0.7%) than Internet link. However, as demonstrated in Figure 4, the video quality for wireless client in

our cost-based policy also approaches 30 dB while cache size is 500Gb. On the other hand, the corresponding video quality in other policies is not more than 10dB. This shows that by caching redundant video object into proxy, the video quality is well protected across the unreliable wireless link.

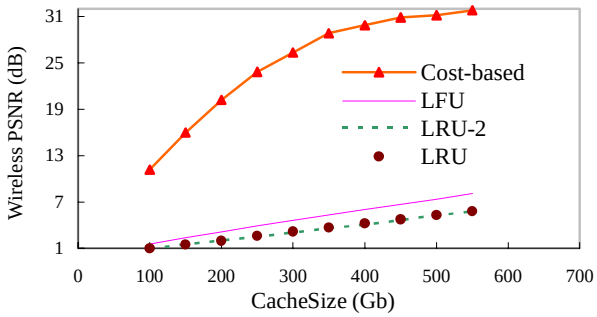

Figure 4. Comparison of replacement algorithms on video quality for wireless clients.

Figure 5 shows the startup latency using the above replacement algorithms. By introducing startup latency revenue rate into cache gain, more video prefixes are cached into proxy. So the startup latency in our policy is greatly reduced comparing with that in other policies. It can be seen from Figure 5 that startup latency with our policy is less than 50% of one using other policies.

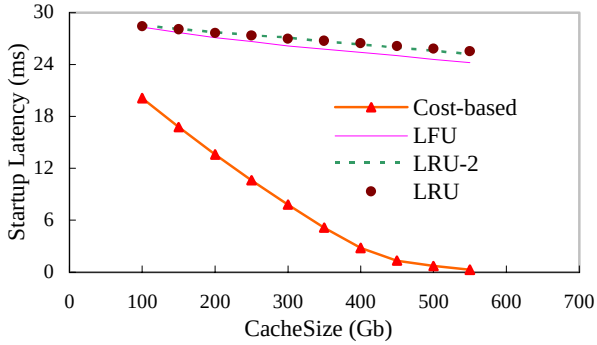

Figure 5. Comparison of cache replacement algorithms on startup latency.

In Figure 6, we study the network cost saving using the above replacement algorithms. It can be seen that with the increasing of cache size, the network cost saving increases proportionally under each policy. Note that our algorithm achieves the best cost saving performance than all the other algorithms by explicitly defining network cost as an optimization objective.

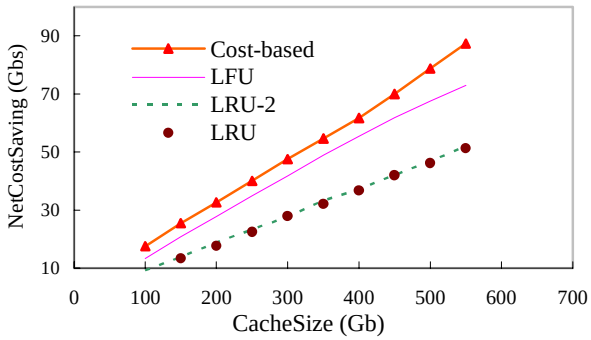

Figure 6. Comparison of cache replacement algorithms on network cost saving.

As mentioned above, Hit Ratio and Byte Hit Ratio are not suitable for exactly measuring the performance of multimedia proxy caching. However, we still compare different policies by using BHR as metric. As shown in Figure 7, with the increasing of cache size, BHR increases accordingly for all policies. This is because larger cache can store more objects which is helpful to improve BHR of proxy. Figure 7 also shows that our proposed algorithm yields the best performance among all the algorithms under all conditions. This demonstrates that our multimedia proxy aims to improve the adaptability of cached objects, so is the BHR of proxy.

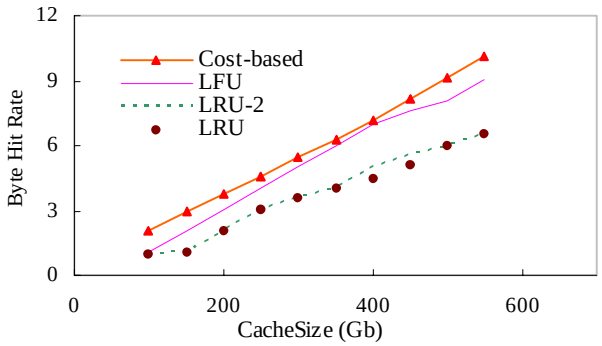

Figure 7. Comparison of replacement algorithms on Byte Hit Ratio.

B.1 Performance of replacement policy for single server in varying network environments

In this sub-section, we vary the number of wireless clients with various BERs in wireless channel to study the service stability of our proxy replacement policy.

B.1.1 Effect of percentage of wireless clients

In this simulation, the proportion of wireless clients to total clients is varied to study the effectiveness of client distribution on system performance. The simulation results are plotted in Figure 8. As demonstrated in this figure, the video quality for wireless client improves as wireless

client proportion increases. When we increase the percentage of wireless clients, the redundant video objects get higher gain to be cached in proxy thus higher video quality is achieved for wireless clients.

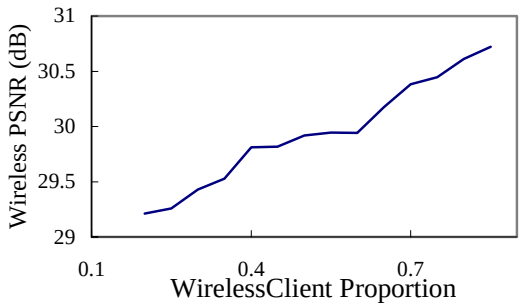

Figure 8. Effect of wireless client proportion on video quality for wireless clients.

B.1.2 Effect of BER in wireless channel

In this simulation, we vary the bit error rate (BER) of wireless channel to evaluate the performance variation of multimedia proxy. The simulation results are shown in the Figure 9. It can be seen that when BER increases, the wireless channel becomes more unreliable, therefore channel adaptive UEP module in our proxy increases the protection level for source code accordingly. However, due to the limitation of cache size, the protection can not catch up with the distortion of video quality. As a result, the wireless video quality decreases as BER increases.

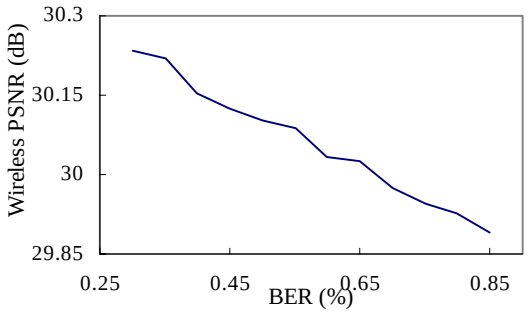

Figure 9. Effect of BER on video quality for wireless clients.

C. Performance of cost-based server selection for multi-servers

In this section, we study the performance of cost-based server selection algorithm. We compare our algorithm with (1) Fixed server selection; (2) Random server selection; (3) Greedy Server Selection. Upon receiving a request from a client, fixed selection scheme chooses a fixed

video server to provide streaming service while random selection scheme randomly chooses a video server. Greedy selection policy always chooses the server that provides the best performance for the current request. Notice that our cost-based not only considers the performance for current request, but also balances the network load to achieve totally high performance for the system rather than Greedy algorithm.

To validate the performance of our algorithm, we set the cache size to 400Gb and compare the four algorithms. The comparison results are listed in Table 2. From Table 2, it can be seen that cost-based server selection algorithm outperforms the other three algorithms. The video quality obtained by the Internet client in cost-based algorithm is about 0.9dB higher than that in Greedy algorithm, it is approximate 1.8dB higher than that in random and fixed algorithms. Note that, as shown in Table 2, the video quality of wireless client in each algorithm is similar to each other. This is because video objects in lower layer have more chance to be cached in proxy so that wireless clients, which has low access bandwidth, is always served from proxy directly. Consequently, the remote server selection has slight impact on it. The startup latency and network cost saving are similar in the case of the wireless video quality.

TABLE 2. COMPARISON RESULTS OF DIFFERENT SERVER SELECTIONALGORITHMS

	Internet PSNR	Wireless PSNR	Delay	Network Saving
Cost-Based	37.4788	29.6763	5.8868	56553527
Greedy	36.5783	29.6744	6.1126	53785689
Random	35.6427	29.6738	6.1968	55343785
Fixed	35.6647	29.6738	6.1898	55092158

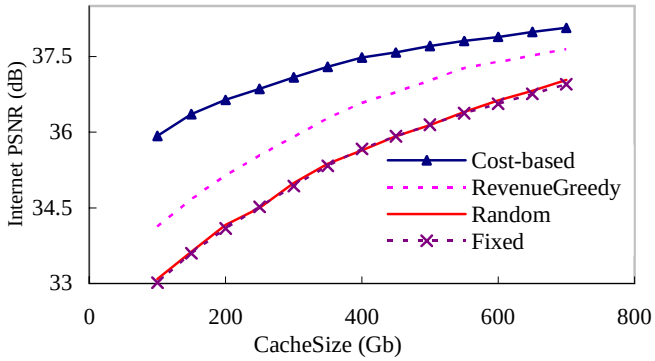

Figure 10. Comparison results of different server selection algorithms.

Because the server selection algorithm has significant impact on Internet video quality, we further run simulation to study the performance of Internet video quality using those server

selection algorithms. The simulation results are illustrated in Figure 10. From Figure 10, we can see that the video quality of Internet clients improves while increasing cache size of proxy. Note that our cost-based server selection outperforms the other algorithms. Specifically, the video quality in our cost-based policy is at least 1.5dB higher than those in any other algorithm when there is a small (<300Gb) cache size. This shows that, by considering both performance of a request and network load balancing, the cost-based server selection algorithm achieves better performance than any other algorithms.

C.1 Performance of server selection policy for multi-server in varying network environments

In this sub-section, we vary different simulation parameters to study the stability and adaptability of our server selection policy. Internet bandwidth between proxy and server are first varied to validate the network adaptation of proxy. Then, we vary the number of video with various zipf distributions to study the service stability of proxy for different type of video clips.

C.1.1 Effect of Internet bandwidth variation

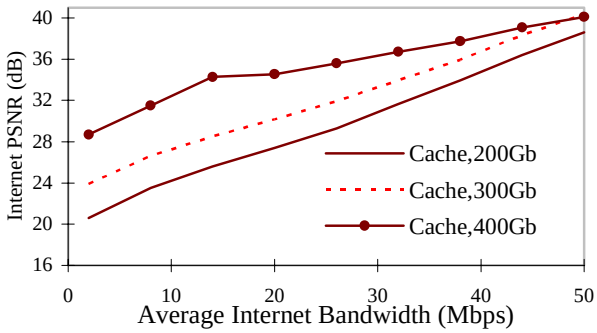

Figure 11. The impact of varying Internet bandwidth on video quality for Internet clients.

In this simulation, we vary the Internet bandwidth between proxy and server to study perceived video quality at client side. The simulation results of Internet clients and wireless clients are plotted in Figure 11 and Figure 12, respectively. As shown in Figure 11 and Figure 12, the perceived video quality, no matter of Internet clients or wireless clients, improves as average Internet bandwidth increases. Because the limited Internet bandwidth between proxy and server restricts the QoS of streaming for end client, some requests can't be well served when there is not enough available Internet bandwidth. Moreover, as shown in Figure 11 and Figure 12, adding cache size has similar effect as adding Internet bandwidth. This shows that, by introducing proxy caching, the proxy can overcome the limitation of network resource with locally storage resource.

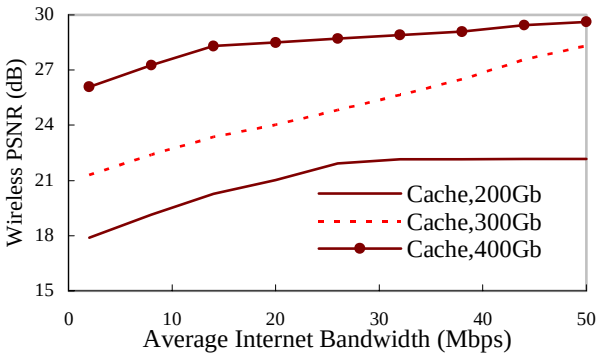

Figure 12. The impact of varying Internet bandwidth on video quality for wireless clients.

C.1.2 Effect of the number of video clips

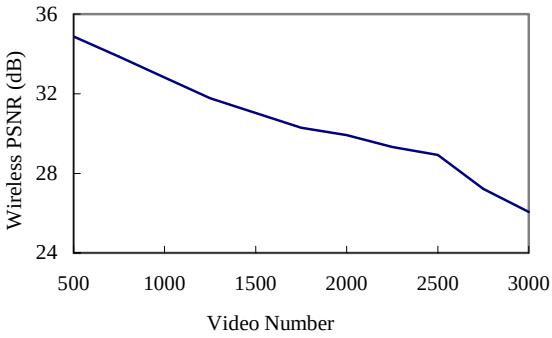

Figure 13. Effect of video number on wireless video quality.

In this simulation, we vary the total video number from 500 to 3000 to study the perceived video quality on client side. The simulation results of Internet client and wireless client are plotted in Figure 13 and Figure 14, respectively. As shown in Figure 13 and Figure 14, the perceived video quality, no matter of Internet client or wireless client, decreases with increasing video number. This reveals that when number of video is small, high percentage of video are cached in proxy which is helpful for system to achieve high performance. When the video number reaches a high rank, only a small amount of videos are cached, thus the performance is decreased, accordingly. From the Figure 13 and Figure 14, we also observe that video quality for wireless clients is more sensitive than that of Internet clients. This is because that the perceived video quality of wireless clients highly relies on the cached videos. When the number of video increasing, the video quality for wireless clients decreases greatly.

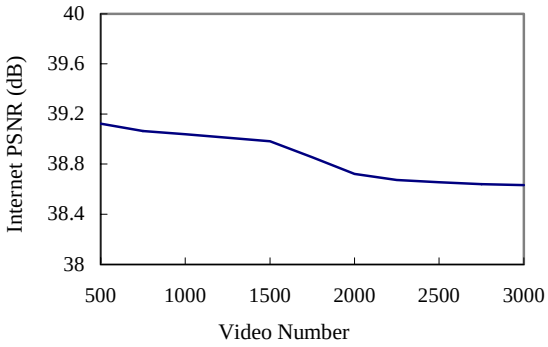

Figure 14. Effect of video number on Internet video quality.

C.1.3 Effect of the skew factor of Zipf distribution

In this simulation, the effect of skew factor of Zipf distribution on system performance is studied. The skew factor reflects the characteristic of popularity for the videos. A higher skew factor indicates that more requests focus on the fewer hot videos; a lower skew factor indicates the popularity of each video tends to similar. The perceived video quality for wireless clients is plotted in Figure 15. As shown in Figure 15, when increasing skew factor, the hotter videos get higher popularity. So, the redundant video objects for those hotter video achieve higher gain to be cached in proxy. As a result, the perceived video quality for wireless client increases accordingly.

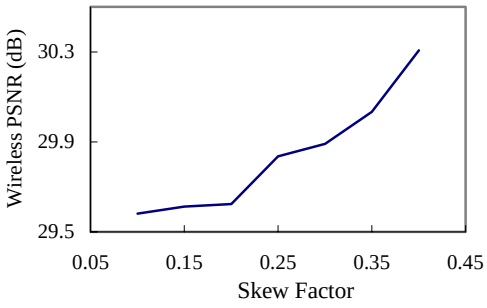

Figure 15. Effect of skew factor of Zipf on wireless video quality.

VI. CONCLUDING REMARKS

In this paper, we presented an architecture for multimedia proxy over wireless Internet where *data cache* and *computational cache* were introduced considering the different characteristics of wireless and Internet links. A unified cost metric based on multiple caching objectives is proposed in this work to evaluate the performance of multimedia proxy. Based on our defined cost metric, for single-server case, we proposed a novel cost-base replacement algorithm so as to

improve all three aspects of performances, i.e., throughput, video quality, and start-up latency, for multimedia proxy over wireless Internet. For multi-servers case, we designed a new cost-based server selection policy for multimedia proxy to improve the video quality when multiple replicated video reside on different servers. Simulation results show that significantly better performance was achieved using our proposed schemes compared to the existing approaches.

ACKNOWLEDGEMENT

Authors would like to thank Dr. Shipeng Li and Dr. Feng Wu from Microsoft Research Asia for providing MPEG-4 PFGS video codec for simulations.

REFERENCES

[1] D.-.H. Nam and S. Park, Adaptive multimedia stream presentation in mobile computing environment, *IEEE TENCON 1999*, pp. 966-969.

[2] J. Vass, S. Zhuang, J. Yao, and X. Zhuang, Mobile video communications in wireless environments, *IEEE 3rd Workshop on Multimedia Signal Processing*, 1999, pp. 45 -50.

[3] M. Margaritidis and G.C. Polyzos, On the application of continuous media filters over wireless networks, *IEEE ICME'00*, vol. 3, 2000, pp. 1241 -1244.

[4] M.E. Crovella and R.L. Carter, Dynamic Server Selection in the Internet, *Third IEEE Workshop* on *Architecture and Implementation of High Performance Communication Subsystems*, 1995. (HPCS '95), 1995, pp. 158 -162

[5] A. Sayal, P. Scheuermann, and P. Vingralek, Selection algorithms for replicated web servers, *Proceedings of the Internet Server Performance Workshop (in conjunction with SIGMETRICS'98)*, 1998.

[6] R.L. Carter and M.E. Crovella, Server selection using dynamic path characterization in wide-area networks, *IEEE INFOCOM '97*, vol. 3, 1997, pp. 1014 -1021.

[7] S.G. Dykes, K.A. Robbins, and C.L. Jeffery, An empirical evaluation of client-side server selection algorithms, *IEEE INFOCOM'00*, vol. 3, 2000, pp. 1361 -1370.

[8] R. Rejaie, M. Handley, H. Yu, and D. Estrin, Proxy Caching Mechanism for Multimedia Playback Streams in the Internet, *Proceedings of the 4th International Web Caching Workshop*, CA., March 1999.

[9] F. Yu, Q. Zhang, W. Zhu, and Y.-Q. Zhang, QoS-adaptive Proxy Caching for Multimedia Streaming over the Internet, in Proc. *First IEEE Pacic-Rim Conference on Multimedia, Australia*, Dec. 13-15, 2000.

[10] L. Fan, Q. Jacobson, P. Cao, and W. Lin, Web Prefetching Between Low-Bandwidth Clients and Proxies: Potential and Performance, *Proceedings of the international conference on Measurement and modeling of computer systems*, 1999, pp. 178 – 187.

[11] Z.-L. Zhang, Y. Wang, D. H.C. Du, and D. Su, Video Staging: A Proxy-Server-Based Approach to End-to-End Video Delivery over Wide-Area Networks, *IEEE/ACM Trans. Networking*, vol. 8, no. 4, 2000, pp. 429 – 442.

[12] R. Tewari, H.M. Vin, A.Dan, and D.Sitaram, Resource-based Caching for Web Servers, in *Proc. SPIE/ACM Conference on Multimedia Computing and Networking*, January 1998.

[13] A. Dan, and D. Towsley, "An approximate analysis of the LRU and FIFO buffer replacement schemes," in *ACM SIGMETRICS*, pp. 143-152, May, 1990.

[14] H. Chou and D. DeWitt, "An evaluation of buffer management strategies for relational database systems," *Proceedings of the 11th VLDB Conference*, 1985.

[15] E.J.O'Neil, P.E.O'Neil, and G. Weikum, "The LRU-k page replacement algorithm for database disk buffering," in *Proceedings of International Conference on Management of Data*, May, 1993.

[16] P. Cao and S. Irani, GreedyDual-Size: A Cost-Aware WWW Proxy Caching Algorithm, *2nd Web Caching Workshop*, Boulder, Colorado, June 1997.

[17] Z. Fu and N. Venkatasubramanian, Combined path and server selection in dynamic multimedia environments, *Proceedings of the 7th ACM international conference on Multimedia*, 1999, pp. 469 – 472.

[18] C. P. Low, H. Yu, J M. Ng, Q. Lin, and Y. Atif, An efficient algorithm for the video server selection problem, Global Telecommunications Conference, *IEEE GLOBECOM'00*, vol. 3, 2000, pp. 1329 -1333.

[19] Q. Zhang, W. Zhu, G. Wang, and Y.-Q. Zhang, Resource Allocation with Adaptive QoS for Multimedia Transmission over W-CDMA Channels, *IEEE WCNC'2000*, 2000.

[20] Q. Zhang, W. Zhu, and Y.-Q. Zhang, "Resource Allocation for Video Streaming over the Internet", special issue on Multimedia over IP in *IEEE Trans. on Multimedia*, Sept. 2001.

[21] S. P. Li, F. Wu, and Y.-Q. Zhang, "Study of a new approach to improve FGS video coding efficiency," *ISO/IEC JTC1/SC29/WG11, MPEG99/m5583*, December 1999, Maui.

[22] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: evidence and implications", *IEEE INFOCOM*, 1999.

[23] http://www.xycoon.com/lsmodelperformance.htm, definition of *RMSE (*Root Mean Squared Error).