QoS-Adaptive Proxy Caching for Multimedia Streaming Over the Internet

Fang Yu, Qian Zhang, *Member, IEEE*, Wenwu Zhu, *Senior Member, IEEE*, and Ya-Qin Zhang, *Fellow, IEEE*

***Abstract*—This paper proposes a quality-of-service (QoS)-adaptive proxy-caching scheme for multimedia streaming over the Internet. Considering the heterogeneous network conditions and media characteristics, we present an end-to-end caching architecture for multimedia streaming. First, a media-characteristic-weighted replacement policy is proposed to improve the cache hit ratio of mixed media including continuous and noncontinuous media. Secondly, a network-condition- and media-quality-adaptive resource-management mechanism is introduced to dynamically re-allocate cache resource for different types of media according to their request patterns. Thirdly, a pre-fetching scheme is described based on the estimated network bandwidth, and a miss strategy to decide what to request from the server in case of cache miss based on real-time network conditions is presented. Lastly, request and send-back scheduling algorithms, integrating with unequal loss protection (ULP), are proposed to dynamically allocate network resource among different types of media. Simulation results demonstrate effectiveness of our proposed schemes.**

***Index Terms*—Bandwidth estimation, multimedia streaming, prefetching, proxy server, QoS-adaptation, rate control, replacement policy, resource allocation, web caching.**

I. Introduction

WITH the popularity of the World Wide Web (WWW), web proxy caching has become a useful approach because it can alleviate network congestion and reduce latency through distributing network load [1], [2]. Traditional proxy servers were designed to serve web requests for noncontinuous media, such as textual and image objects [3], [4]. With the increasing advent of video and audio streaming applications, continuous-media caching has been studied in [5]–[7]. Most recently, there is interest in caching both continuous and noncontinuous media [8]. However, efficiently caching both continuous and noncontinuous media faces many challenges. First, different types of media have different characteristics. Specifically, real-time media such as video or audio is delay sensitive but capable of tolerating moderate packet-loss events, while nonreal-time media such as Web data is less delay sensitive but requires reliable transmission. Consequently, different types of media may have different quality impairments under the same network condition. Furthermore, within each type of media, objects to be cached may not be equivalently important and some objects may depend on others. Secondly, the current Internet only provides best-effort service and does not provide quality of service (QoS) guarantee. Network conditions such as bandwidth, packet-loss ratio, delay, and delay jitter vary from time to time. Thus, real-time characteristic of streaming media requires new functionalities such as scheduling and resource management to be incorporated into the proxy caching. Thirdly, the bandwidth of access networks vary greatly, the client's access model, the access pattern, and the network bandwidth for a given link vary greatly from time to time. Therefore, the heterogeneities in clients and networks need to be considered when designing the proxy.

Replacement policy is one of the key components in the proxy design. The existing caching replacement policies for web data can be roughly categorized as recency-based and frequency-based. Recency-based algorithms, e.g., least recently used (LRU) [9], exploit the locality of reference inherently in the programs. Frequency-based algorithms, e.g., least frequently used (LFU) [10], are suited for skewed access patterns in which a large fraction of the accesses go to a disproportionately small set of hot objects. To balance frequency- and recency-based algorithms, several improved algorithms named LRU-k and LRFU are proposed in [11] and [12].

For most of the data accessed on the web today, which contain text and static images, the above algorithms seem adequate. However, as streaming of continuous media data becomes popular, different media characteristics and access patterns need to be considered. To the best of our knowledge, fewer works to date have clearly addressed how to efficiently cache mixed media, especially for multimedia streaming applications. In [8], a cache replacement algorithm for mixed media according to media's bandwidth and space requirement is proposed. However, the bandwidth resource considered therein is the fixed bandwidth of the disk, rather than the varying network bandwidth from client to proxy and from proxy to server.

Prefetching between proxy and client has been proven to be very effective if the proxy can accurately predict the users' access patterns [13], [14]. Since continuous media has a strong tendency to be accessed sequentially, Sen *et al.* proposed a proxy prefix caching scheme for multimedia streams [5]. The prefix of a multimedia stream is stored in the proxy in advance. Upon receiving a request for the stream, the proxy immediately initiates transmission to the client while simultaneously requesting the remaining portion from the server. In this way, the latency between server and proxy can be hidden. However, several issues, such as how much should be prefetched under different network conditions and how to schedule the

Manuscript received February 22, 2001; revised March 21, 2002. This paper was recommended by Associate Editor H. Gharavi.

F. Yu was with Microsoft Research Asia, Beijing, China, and Fudan University, Shanghai, China. She is now with the Department of Computer Science, University of California, Berkeley, CA 94720 USA.

Q. Zhang, W. Zhu, and Y.-Q. Zhang are with Microsoft Research Asia, Haidian District, Beijing 100080, China (e-mail: wwzhu@microsft.com).

Digital Object Identifier 10.1109/TCSVT.2003.809829

pre-fetch and miss requests, are not addressed. Rejaie *et al.* proposed a quality-adaptive proxy caching mechanism for multimedia streaming [15], which took network bandwidth into consideration. This work was performed specifically for layered video, rather than mixed media with continuous and noncontinuous ones.

To address the above issues, this paper first proposes a media-characteristic-weighted replacement algorithm that takes the network bandwidth into account and adapts to the media characteristics as well as users' request patterns. Then, a network-condition- and media-quality-adaptive resource-management mechanism is presented to dynamically allocate cache resource among different types of media according to the client's request-model and network conditions. Moreover, we propose an efficient prefetching scheme and a QoS-adaptive miss strategy by taking available network bandwidth and users' access patterns into account. Note that, here, QoS-adaptive miss strategy denotes media-characteristic and network-condition based policy handling cache miss. In addition, weighted request-scheduling and send-back scheduling schemes are introduced, which utilize the network resource efficiently according to media characteristics.

The rest of this paper is organized as follows. In Section II, we present an end-to-end architecture for QoS-adaptive caching of mixed media streaming over the Internet. In Section III, a replacement policy adapting to the characteristics of different types of media and a network-adaptive cache-resource-management scheme are proposed. Section IV introduces network-bandwidth-adaptive scheduling algorithm with network status estimation, and network resource allocation for requesting miss objects and prefetching from proxy to server. In addition, network resource allocation for client send-back is also discussed. Section V gives the simulation results. Finally, conclusions are made in Section VI.

II. An End-to-End Architecture for QoS-Adaptive Mixed Media Caching

Proxies are emerging as an important way to reduce user-perceived latency and network resource requirements in the Internet. A proxy server in general is located near edge router or gateway in the Internet. It is connected to clients via a local access network while connected to servers via a wide area network. The local access network could be LAN, xDSL access network, cable access network, or wireless access network. Generally, the network status between client-proxy and proxy-server are quite different. Fig. 1 shows the end-to-end architecture of our proposed caching scheme for multimedia streaming. Multimedia here refers to mixed media, such as audio, video, images, WWW data, etc. To adapt caching to the varying network conditions of client-proxy and proxy-server, two bandwidth monitor modules are explicitly introduced in this architecture.

In this architecture, the proxy server works as follows. When a proxy cache receives a request for a particular media object, it checks whether the corresponding media object is cached locally or not. If the requested object is in the cache, i.e., in the case of a cache hit, the object is streamed from the proxy over the local access network to the client without interacting with the destination server. However, if it is not available in the cache, i.e., in the case of a cache miss, the proxy forward the request to the appropriate original server if necessary. Upon receiving such a request, the original server streams the requested media object to the proxy server. The proxy relays the requested stream to the requesting client and meanwhile caches the media object in its local storage. If the local storage at the proxy server is full, the proxy decides which media object needs to be removed from the cache to make room for the new object. If the replacement algorithm fails to free up enough space for the new object, the object is streamed from the original server directly to the client.

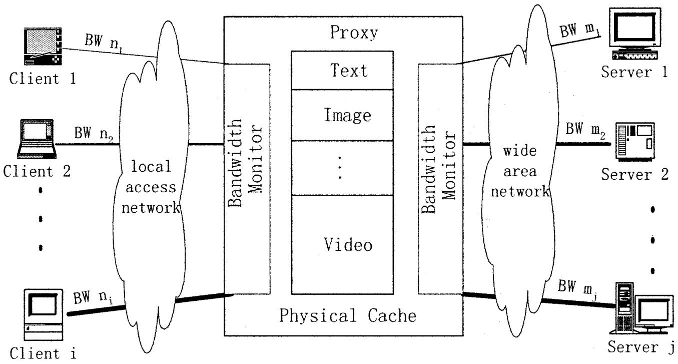

Fig. 1. End-to-end architecture of our caching scheme for mixed media streaming.

There are two important issues that have not been considered in the above caching procedure. One is the characteristics and request patterns of different types of media, and the other is the varying network conditions of client-proxy and proxy-server. Taking these two issues into account, we proposed a QoS-adaptive caching scheme for mixed media. Fig. 2 illustrates the detailed block diagram of this scheme.

Taking the different characteristics of continuous media and noncontinuous media into consideration, different page management and replacement policies are adopted for different types of media. The *resource management* module periodically re-allocates the current resource according to the media characteristics, the access frequencies of different types of media, and the varying network conditions of client-proxy and proxy-server.

Considering the request patterns of continuous media, *pre-fetching* policy is employed for the hit request. Future portion of the continuous media is requested from proxy to the server when the proxy streams the requested media objects to the client. User-perceived latency can be reduced and overall quality of the requesting media can be improved by pre-fetching.

As mentioned above, two *network bandwidth-monitoring* modules are used to dynamically estimate the available bandwidth between client and proxy as well as between proxy and server. Note that in general there exists a bandwidth mismatch between client-proxy and proxy-server, thus, *request scheduler* and *send-back scheduler* modules are introduced. The *request scheduler* allocates the bandwidth between proxy and server for miss and prefetch requests. The *send-back scheduler* allocates the bandwidth between client and proxy for different types of media.

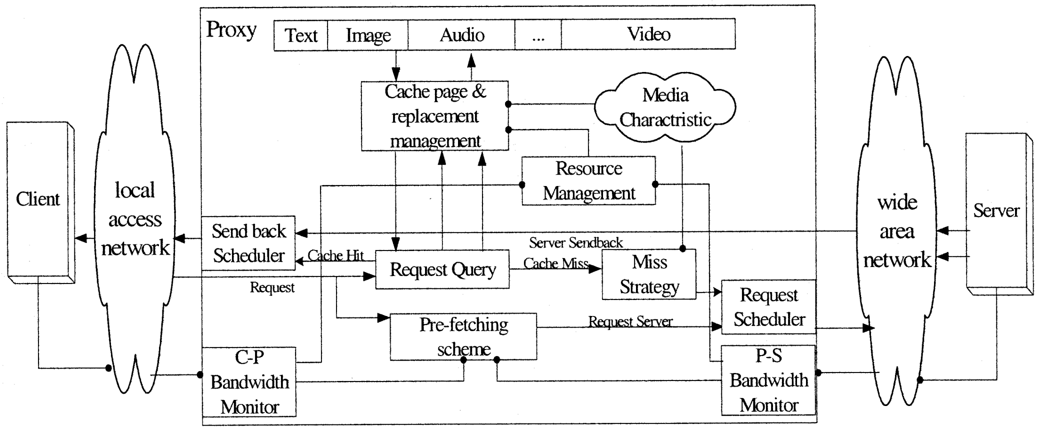

Fig. 2. Block diagram of our proposed QoS-adaptive caching scheme for mixed media.

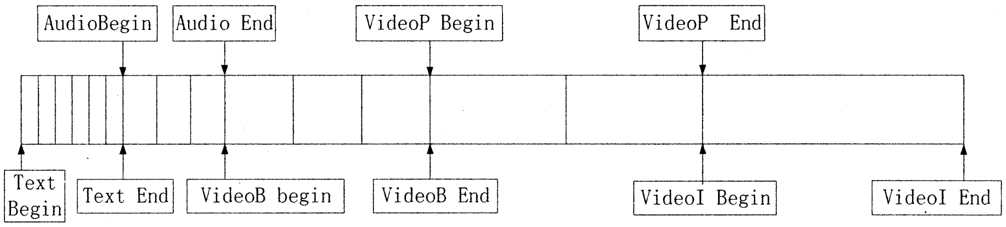

Fig. 3. Multiple-level page size management scheme.

In summary, the cache and network resources are allocated based on the media characteristics and network conditions in our proposed QoS-adaptive caching scheme to achieve good quality for mixed media. The overall scheme consists of: 1) client-proxy and proxy-server bandwidth monitors; 2) a media-characteristic-weighted replacement policy; 3) a network-condition- and media-quality-adaptive cache resource-management mechanism; 4) a QoS-adaptive miss strategy and pre-fetching algorithm; and 5) client send-back scheduler and server request scheduler.

III. Cache Resource Management

In this section, we address several cache resource-management issues such as page size management, cache replacement policy, and cache resource re-allocation by considering the media characteristics and network conditions.

A. Multiple-Level Page-Size Management Scheme

It is known that different types of media have different sizes. For example, text objects may have the smallest size (0.208 kbytes, on average, for 1998 World Cup traces obtained from http://ita.ee.lbl.gov/html/contrib/WorldCup.html), image object may have medium size (1.68 kbytes, on average), audio object may have medium high size (about 4.6 kbytes, on average), and video object may have the largest size (1.407 Mbytes, on average). For video objects, each frame (e.g., I, B, and P) also have different sizes. In the case of multilayered scalable video, the base layer has the smallest size, and the sizes of the enhancement layers are increased gradually. As we know, the traditional proxy server supports fixed page size. It, however, may not be suited for mixed media caching. If the page size is large, it is not good for caching text object since the resource is wasted. On the other hand, if the page size is small, video object may have a large index. Considering the variable bit rate (VBR) nature of the continuous media, one may naturally consider assigning unequal page sizes to each different frame/layer. However, it may be complex and difficult in terms of the cache management.

Considering both the complexity and efficiency of page management, we propose a multiple-level page-size management scheme, which adopts different page sizes for different types of media. To be specific, for the same type of media, the page size is fixed. For instance, we may have the following page size relation: $S_1 < S_2 < S_3 < S_4 < S_5 < S_6$, where S_1 stands for text, S_2 for image, S_3 for audio, S_4 for video B frame, S_5 for video P frame, and S_6 for video I frame. In the cache, we store the objects of the same level together, as shown in Fig. 3. Our proposed scheme works as follows. We first divide the physical cache into several parts. Each part stores a specified type of media and the page size in this part is equal. However, different parts have different page sizes. Specifically, the two adjacent parts have different page sizes, and the latter one is multiple of the former one. Here, we design the page size for each type of media, on one hand, based on the statistics result obtain from the typical traces, and on the other, in order for the efficient resource re-allocation management, which will descried in detail in Section III-C. The space each type of media occupies is variable and adaptively changes according to the hit/miss ratio. After the specific period of time, we enlarge the space of the media with high hit-ratio and decrease the one with high miss-ratio. The details will be discussed later.

B. Media-Characteristic-Weighted Cache Replacement Policy

In general, a proxy server has a fixed amount of storage. When the new request comes in while storage is full, the proxy must evict one or more media objects based on a certain caching replacement policy. The goal of the replacement policy is to make the best use of available resources, including disk/memory space as well as network bandwidth. To achieve this goal, the cache replacement policy should be able to accurately predict future popularity of objects and determine how to use its limited space in the most advantageous way.

Media characteristics could significantly affect the performance of the caching replacement algorithm. First, different types of media have different request patterns. For continuous media, it has the tendency to be sequentially accessed; meanwhile, there exist certain reference relations among media objects (e.g., the P and B frames in a video sequence highly depends on the I frame). However, for noncontinuous media objects, they are usually randomly and independently accessed. Secondly, different types of media have different quality impacts, e.g., audio objects may have a higher quality impact than that of video objects. Thirdly, different types of media have different sizes. Taking network bandwidth, storage space, and media characteristic into consideration, we propose a media-characteristic-weighted replacement policy (MCW-n) for mixed-media caching.

In our scheme, each object in the cache has a weight as a measure for replacement. When a new object comes in, if there is no free space in the cache, the proxy flushes out the object with the lowest weight. As discussed above, the value of the caching gain not only depends on the estimation of the time-to-reaccess (TTR) or the access probability of the object, but also depends on the importance of the object. Therefore, the weight function is defined as follows:

$$W = \text{Priority} \times (\beta \times \text{Tendency} + (1-\beta) \times \text{Frequency}) \quad (1)$$

where Priority represents the importance of an object. The proxy can assign different priorities to different types of objects. The priority assignment may depend on different applications. Tendency indicates the impact of the current request on the following requests according to the continual characteristic of the media. Frequency represents the popularity of the object to be accessed. β is the control parameter balancing the impact between Tendency and Frequency, which can be selected based on the network condition. More details will be given in the simulation stage in Section V-A. In summary, by using this weight function, several characteristics for mixed media access, such as the traditional object's access probability Frequency, the sequential tendency or dependency of continuous media Tendency, and the importance of the object priority have been considered.

Tendency shows the probability of the following requests' hitness. For text and images, we may use the data mining rule to calculate probability of the following requests' hitness. As for the continuous media such as audio and video, Tendency essentially represents continuity of the continuous media in the time scale. Because the continuous media is usually highly correlated, one can infer future events from the past ones. Upon receiving a request of the video or audio object in the proxy, the weights of objects within the pre-determined window are enhanced according to the weight calculation function. There are several approaches for calculating the Tendency of a certain media. One way is to use a mathematical distribution such as Gaussian distribution to represent Tendency.

We can use the method proposed in [15] to calculate Frequency, which is described as follows:

$$\text{Frequency} = 1/\mathrm{MTTR} \quad (2)$$

where mean time-to-re-access (MTTR) is measured as the weighted sum of the inter-arrival times between the previous accesses. Denote the access times of the last n accesses as $t_n, t_{n-1}, \ldots, t_0$, where t_i is the time of the last ith access. Let the weighting function $w(i)$ satisfy

$$w(i) = \alpha \times w(i-1), \quad \alpha \leq 1 \quad \text{and} \quad w(0) = 1-\alpha. \quad (3)$$

Then, we have

$$\mathrm{MTTR} = \sum_{i \geq 0} (t_i - t_{i-1}) \times w(i). \quad (4)$$

Thus, for a given time t_0, $\mathrm{MTTR}(t_0) = (1-\alpha)(t_0 - t_1) + \alpha \times \mathrm{MTTR}(t_1)$. Note that the averaging factor α can be tuned to a bias for or against recency.

In summary, our proposed caching replacement policy (MCW-n) has the following characteristics.

- It can store the media objects hierarchically based on their priorities. For example, the enhancement layer alone is not useful if the base layer or the lower layers are missed. Our algorithms ensure that we drive out the higher layers (less important layers) first through setting their priorities low.
- Since tendency, frequency and priority are used to determine the weight of each object, we can increase the hit ratio for continuous media and noncontinuous media simultaneously.
- It supports VCR functions of video. Specifically, the I frame has higher priority and thus has higher weight. When a VCR function (e.g., forward) is triggered, even if the media of that part is not fully in the cache, we can still send back some related I frame. In the case of scalable video, the lower layer has higher priority, and thus it has a lower probability of being driven out.

C. Network-Condition- and Media-Quality-Adaptive Resource Re-Allocation Scheme

As discussed above, we adopted the multiple-level page size management approach for mixed media caching. Based on media characteristics, different types of media need to occupy different cache resource. That is, larger resource is usually allocated to continuous media, such as video and audio, than noncontinuous media such as text and image. Note that in our scheme, cache is not divided based on a fixed proportion. In addition, with the varying network conditions and media request patterns in mind, we periodically re-allocate the cache resource to dynamically match the present condition.

To efficiently re-allocate the cache resource, several issues are addressed in our scheme. First, we take the priorities of different media into account. For example, text's priority is usually very

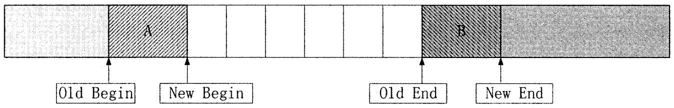

Fig. 4. Caching resource re-allocation.

high. As a result, text will be replaced lastly. In the case of scalable media such as video, the priority of the base layer should be set higher than that of enhancement layers so that relatively more objects in the base layer can be kept in the cache [24]. Secondly, the size that each media currently occupies is considered. Thirdly, the miss ratio, which shows how often the cache cannot meet client requests, is considered.

Ideally, the optimal solution can be achieved by establishing the relation between the miss-ratio gain and the cache resource requirement for each type of media. For the sake of simplicity, we present here a simple criterion for resource re-allocation. Let S_i represent the size that the media i currently occupies, P_i denote the priority of the media i, and MR_i represent the miss ratio for the ith media. Then, the re-allocation demand D_i for the ith media can be calculated as

$$D_i = P_i \times \mathrm{MR}_i / S_i. \tag{5}$$

The re-allocation demand for each type of media D_i is calculated and sorted as $D'_1 < D'_2 \cdots < D'_m$. The threshold value δ is introduced to determine whether it is necessary to re-allocate cache resource among different types of media. If $D'_m - D'_1 < \delta$, then all the media are in a similar condition and no cache resource should be moved. This can reduce the possibility of thrashing.

Even if the pages that are required to move are not adjacent, the proxy can move page pointer, as illustrated in Fig. 4, using the scheme described as follows.

If $D'_m - D'_1 \geq k \times \delta$ for $k > 1$, Then, re-allocate some resource from media m to media 1. More specifically

$$\left| \frac{\sqrt{\mathrm{PageSize}_m \times \mathrm{PageSize}_1}}{\mathrm{Max}_{\mathrm{pagesize}}} \times k \right| \times \frac{\mathrm{Max}_{\mathrm{pagesize}}}{\mathrm{PageSize}_m} \tag{6}$$

pages are moved from media m to media 1.

Repeat the above procedure until $D'_{m-i} - D'_{i+1} \leq \delta$.

Since the page size of media m may differ greatly from that of media 1, $\sqrt{\mathrm{PageSize}_m \times \mathrm{PageSize}_1}$ is calculated as the average page size from media m to media 1. In addition, we should move k times the predefined moving page number if the difference of $N'_m - N'_1$ is k times δ. Furthermore, we want to re-allocate the cache resource as a multiple of the largest page size in the cache $\mathrm{Max}_{\mathrm{pagesize}}$, so that it is easy to re-allocate the cache resource and no fragment in the cache occurs. Note that because the two media have different page sizes and the current one is a multiple of the previous one, page movement can be achieved by just moving the pointer of the previous one's *end* and that of the current one's *begin*.

IV. Network Bandwidth-Adaptive Scheduling

The traditional caching scheme assumes that system is limited by disk performance. However, considering the multimedia streaming application, most retrieving objects need to be delivered from the remote servers. Thus, the network resource becomes a bottleneck. Several network-related issues, including dynamical bandwidth allocation and weighted scheduling, are addressed in our caching scheme.

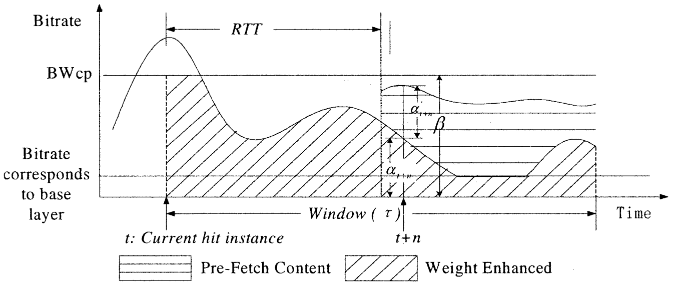

Fig. 5. Pre-fetching for scalable continuous media.

As mentioned previously, there are two network-monitoring modules in our architecture. To cope with packet dropping and bandwidth fluctuation in the current Internet, we use our proposed multimedia streaming TCP-friendly protocol (MSTFP) [16] for available bandwidth estimation.

A. Network Resource Allocation for Miss and Pre-Fetch Requests

If a request is missed in the cache, the corresponding data should be requested from the server if necessary. On the other hand, if a request for continuous media is hit in the cache, certain following segments of data may need to be pre-fetched from the server. Consequently, the traffic between the proxy and server consists of data for miss and pre-fetch requests. Efficiently allocating the network resource among different types of requests can improve the performance.

1) Miss Strategy: If a request is missed in the cache, we use the miss strategy to decide whether to request it from the server or not. In our miss strategy, different policies for different types of media are assigned according to the media characteristics. Specifically, considering real-time media is delay sensitive, round-trip time (RTT) is used to decide whether the request for real-time media can be obtained within the delay bound. If not, this request is simply rejected. Otherwise, it is added to an appropriate queue. We maintain separate queue for each media based on the delay requirements. Therefore, we provide differentiated services for different media according to the media characteristics.

2) Pre-Fetch Strategy: There is a strong tendency for continual requests for continuous media. So when a video or audio object is requested, we enhance the weight of following objects in the range of a sliding window, which starts from the hitting point to a pre-set value τ. We may pre-fetch the requested object that is not in the cache.

Considering the latency from server to proxy, the objects that immediately after the current hit will not be pre-fetched. Instead, we start pre-fetching from the RTT (from proxy to cache) distance away of the current hit place (see Fig. 5). As shown in Fig. 5, while pre-fetching scalable media object in the window, objects in the cache have bandwidth α_{t+n} and the current es-

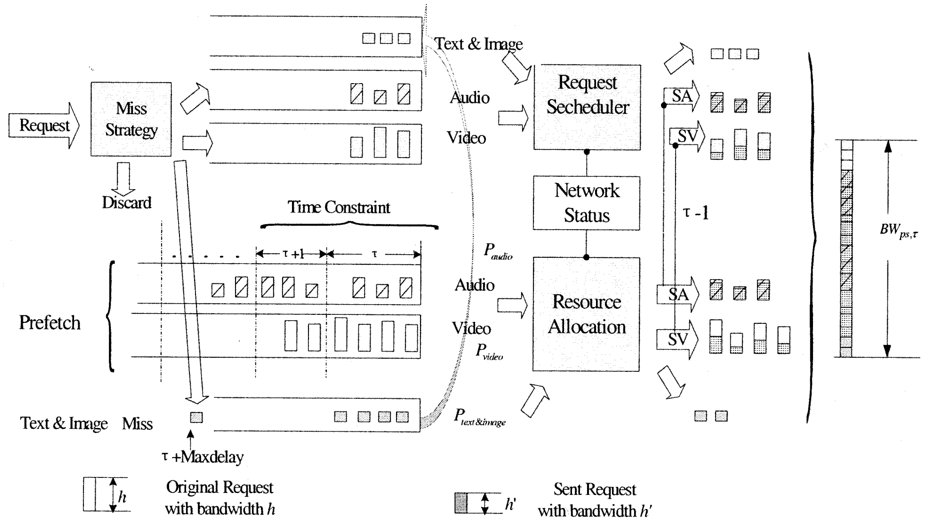

Fig. 6. Scheduling scheme for pre-fetch and miss requests.

timated bandwidth from proxy to server is β. The actual prefetching bandwidth α'_{t+n} is determined according to the current estimated bandwidth and other client requests.

We use SV_τ to represent the satisfaction ratio of client request for video during time period τ. It is represented as $SV_\tau = \alpha'_{i+n}/\beta$. There are two basic assumptions. First, each client should be given approximately the same satisfaction ratio. Secondly, the satisfaction ratio of each client should not vary greatly from time to time. The satisfaction ratio of audio SA_τ is defined in a similar way.

3) Unequal Loss Protection for Multiple Media: The bandwidth of the current Internet is generally time-varying, resulting in bursty pattern of packet-loss occurrence. Therefore, error-control techniques such as forward error correction (FEC) and automatic repeat request (ARQ) are necessary to ensure high quality video transmission. Because of the strict delay constraint for real-time video transmission, it is often considered more beneficial to apply FEC than to use ARQ.

Different types of media, such as text, image, audio, and video, have different tolerances to packet loss in media streaming. For example, it is known that the aural sense is more sensitive to disturbances than the visual one. Therefore, it is appropriate to give audio stronger protection than video. In other words, we use error-control code with various strengths to protect the source data unequally. That is, important data are protected by stronger codes while less important data are protected by weaker codes [17].

Reed–Solomon (RS) codes [18] are used for error control in our scheme. RS codes are a well-known class of block codes with good error-correction properties, especially for channel burst errors. An RS code is represented as $RS(n, k)$, where k is the length of source symbols and $n-k$ is the length of protection symbols. It is known that an RS code can guarantee to correct up to $t = (n-k)$ errors with the knowledge of error position. Suppose the desired packet-loss ratio ($l_{\rm des}$) for audio is 10^{-4}, and the current network packet-loss ratio ($l_{\rm cur}$) is 10^{-2}, then, protection needs to be added for audio to achieve the desired packet-loss ratio requirement. The probability of correcting t error packets in n packets, $P(n, t)$, is calculated as

$$P(n, t) = \binom{t}{n} \times l_{\rm cur}^{t} \times (1 - l_{\rm cur})^{n-t}. \tag{7}$$

To achieve the desired packet-loss ratio, a suitable t is needed to satisfy the following constraint:

$$\sum_{t=0}^{n-k} P(n, t) \geq (1 - l_{\rm des})^{n}. \tag{8}$$

Then the required resource for the given media source $R_{\rm source}$ is calculated as

$$R_{\rm required} = n/(n-t) \times R_{\rm source} \tag{9}$$

which can be further represented as

$$R_{\rm required} = \Re(l_{\rm cur}, l_{\rm des}) \times R_{\rm source}. \tag{10}$$

4) Delay-Constrained Weighted Scheduling for Pre-Fetch and Miss Request: Having the above strategies, it is essential to schedule different types of requests based on the network condition and media characteristic. To date, there are some works [19]–[21] on scheduling of the queue. In this paper, we propose a delay-constrained weighted scheduling algorithm to allocate network resources among different types of requests.

As shown in Fig. 6, if a cache-miss occurs, the miss policy would decide whether to request it from the server or not. There are several queues in the scheduling scheme, each with different weight. The larger weight a queue has, the higher probability it will be served. Then the necessary requests are classified and put into the appropriate queues. To be specific, noncontinuous media request is sealed with the time-stamp and put into the least weighted queue. If the request has not been served within the next $\rm Max_{delay}$ time, it will be moved to the highest weighted queue. The $\rm Max_{delay}$ can be different among different types of noncontinuous media. For the sake of simplicity, the same $\rm Max_{delay}$ is adopted in the following section.

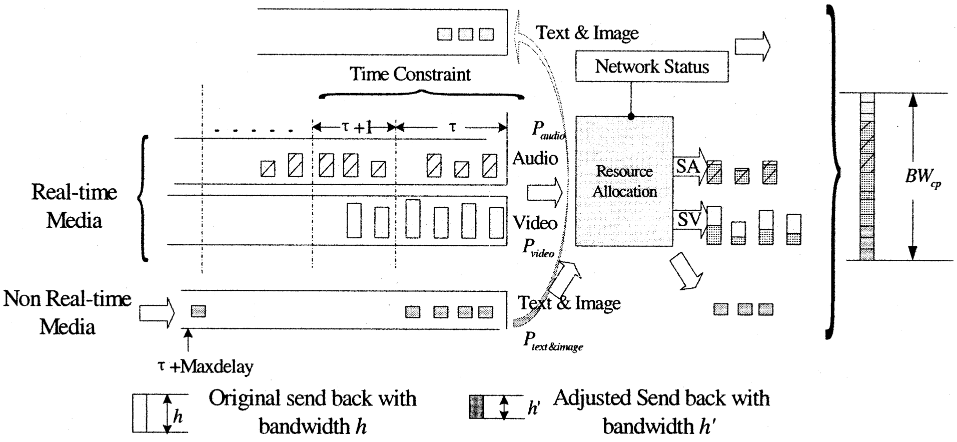

Fig. 7. Scheduling scheme for mixed media sending back.

The following algorithm is used to schedule the requests so as to efficiently utilize the available bandwidth while avoiding network congestion as much as possible. It is composed of the following steps.

Step 1) Generate requests for text and image, which have not been served within $\mathrm{Max}_{\mathrm{delay}}$.

Step 2) Generate requests for those in the miss queues with the average satisfaction ratio of $SA_{\tau-1}$ and $SV_{\tau-1}$, which are used while pre-fetching audio and video in the time interval $\tau - 1$. Note that higher priority media (e.g., audio) is served first and lower priority media (e.g., video) is served afterwards.

Step 3) If there is still some bandwidth $\mathrm{BW}_{\mathrm{left}}$ left, generate the pre-fetching requests. It includes the following steps:

- Calculate the bandwidth needed for each type of media's requests, $\mathrm{RBW}_{\mathrm{audio}}$, $\mathrm{RBW}_{\mathrm{video}}$, and $\mathrm{RBW}_{\mathrm{text\&image}}$, as follows:

$$\mathrm{RBW}_{\mathrm{audio}} = \sum_{i=0}^{n} \Re(l_{\mathrm{cur},ps}, l_{\mathrm{des,\,audio}}) \times \mathrm{RBW}_{i,\,\mathrm{audio}} \quad (11)$$

where n is the number of pre-fetching requests for audio that should be served during this time period, $\mathrm{RBW}_{i,\,\mathrm{audio}}$ is the bandwidth needed for the ith audio request, $l_{\mathrm{cur},ps}$ is the current estimated packet-loss ratio between proxy and server, and $l_{\mathrm{des,\,audio}}$ is the desired packet-loss ratio for audio. Similarly

$$\mathrm{RBW}_{\mathrm{video}} = \sum_{i=0}^{m} \Re(l_{\mathrm{cur},ps}, l_{\mathrm{des,\,video}}) \times \mathrm{RBW}_{i,\,\mathrm{video}} \quad (12)$$

where m is the number of pre-fetching requests for video that should be served during this time period, $\mathrm{RBW}_{i,\,\mathrm{video}}$ is the bandwidth needed for the ith video request, and $l_{\mathrm{des,\,video}}$ is the desired packet-loss ratio for video. For noncontinuous media, we have

$$\mathrm{RBW}_{\mathrm{text\&image}} = k \times \mathrm{RRBW}_{\mathrm{text\&image}} \quad (13)$$

where k is the number of requests for text and image. Note that reliable transmission is required for noncontinuous media, so a predefined reserved bandwidth $\mathrm{RRBW}_{\mathrm{text\&image}}$ is used for text and image in our scheme.

- Allocate the remaining bandwidth $\mathrm{BW}_{\mathrm{left}}$ among each media according to its priority. Given each media priority, P_{audio}, P_{video}, or $P_{\mathrm{text\&image}}$, we have

$$\mathrm{BW}_{\mathrm{audio}} = \frac{\mathrm{RBW}_{\mathrm{audio}} \times P_{\mathrm{audio}}}{\sum\limits_{i \in \mathrm{allmedia}} (\mathrm{RBW}_i \times P_i)} \times \mathrm{BW}_{\mathrm{left}} \quad (14)$$

$$\mathrm{BW}_{\mathrm{video}} = \frac{\mathrm{RBW}_{\mathrm{video}} \times P_{\mathrm{video}}}{\sum\limits_{i \in \mathrm{allmedia}} (\mathrm{RBW}_i \times P_i)} \times \mathrm{BW}_{\mathrm{left}} \quad (15)$$

$$\mathrm{BW}_{\mathrm{text\&image}} = \frac{\mathrm{RBW}_{\mathrm{text\&image}} \times P_{\mathrm{text\&image}}}{\sum\limits_{i \in \mathrm{allmedia}} (\mathrm{RBW}_i \times P_i)} \times \mathrm{BW}_{\mathrm{left}}. \quad (16)$$

- Calculate SA_τ and ${}_SV_\tau$ for audio and video, which are, respectively, given by

$$SA_\tau = \frac{\mathrm{BW}_{\mathrm{audio}} + \sum\limits_{i=0}^{n} \alpha_{i,\mathrm{audio}}}{\sum\limits_{i=0}^{n} \beta_{i,\mathrm{audio}}} \quad (17)$$

$$SV_\tau = \frac{\mathrm{BW}_{\mathrm{video}} + \sum\limits_{i=0}^{m} \alpha_{i,\mathrm{video}}}{\sum\limits_{i=0}^{m} \beta_{i,\mathrm{video}}}. \quad (18)$$

- Calculate the pre-fetching bandwidth for each request, i.e.,

$$\alpha'_{i,\,\mathrm{audio}} = \beta_{i,\,\mathrm{audio}} \times SA_\tau - \alpha_{i,\,\mathrm{audio}} \quad (19)$$

$$\alpha'_{i,\,\mathrm{video}} = \beta_{i,\,\mathrm{video}} \times SV_\tau - \alpha_{i,\,\mathrm{video}}. \quad (20)$$

- Generate requests for the first $\mathrm{BW}_{\mathrm{text\&image}}/\mathrm{RRBW}_{\mathrm{text\&image}}$ requests for text and image. If there are some requests for text or image that have not served but are going to reach the $\mathrm{Max}_{\mathrm{delay}}$, move them to the highest priority queue.

Notice that in our scheduling scheme, the two satisfaction ratios SA and SV can be adapted to the varying network condition.

B. Network Resource Allocation for Client Send-Back

Considering the bandwidth difference between client-proxy and proxy-server as well as different media's quality impact, it is important to adopt a scheduling scheme to allocate available resource among different types of media so as to achieve the optimal overall quality. Fig. 7 illustrates a weighted scheduling scheme to control data send-back from proxy to client.

As shown in Fig. 7, during the time interval τ, the data send-back to the client is determined by the estimated bandwidth from client to proxy, $\mathrm{BW}_{\mathrm{cp},\,\tau}$. Since the client may have multiple requests at the same time, we need to schedule what to be sent back so as to maximally use the bandwidth resource while avoiding network congestion based on $\mathrm{BW}_{\mathrm{cp},\,\tau}$.

Similar to the resource allocation for pre-fetch and miss requests, we first allocate enough resources to nonreal-time media, which has been delayed by $\mathrm{Max}_{\mathrm{delay}}$. Then we allocate the rest of network bandwidth among continuous media and noncontinuous media according to their priorities. Our algorithm consists of the following steps.

Step 1) Allocate network bandwidth for text and image, which have not been served within $\mathrm{Max}_{\mathrm{delay}}$.

Step 2) Allocate remaining network resource, $\mathrm{BW}_{\mathrm{left}}$, among different types of media as follows:

- Calculate the bandwidth needed for each type of media, $\mathrm{RBW}_{\mathrm{audio}}$, $\mathrm{RBW}_{\mathrm{video}}$, and $\mathrm{RBW}_{\mathrm{text\&image}}$. That is

$$\mathrm{RBW}_{\mathrm{audio}} = \sum_{i=0}^{n} \Re(l_{\mathrm{cur,\,cp}},\, l_{\mathrm{des,\,audio}}) \times \mathrm{RBW}_{i,\,\mathrm{audio}} \tag{21}$$

  where n is the number of requests for audio that should be served during this time period, $l_{\mathrm{cur,\,cp}}$ is the current estimated packet-loss ratio between client and proxy, and $\mathrm{RBW}_{i,\,\mathrm{audio}}$ and $l_{\mathrm{des,\,audio}}$ are defined as in (11). Similarly

$$\mathrm{RBW}_{\mathrm{video}} = \sum_{i=0}^{m} \Re(l_{\mathrm{cur,\,cp}},\, l_{\mathrm{des,\,video}}) \times \mathrm{RBW}_{i,\,\mathrm{video}} \tag{22}$$

  where m is the number of requests for video that should be served during the time period, and $\mathrm{RBW}_{i,\,\mathrm{video}}$ and $l_{\mathrm{des,\,video}}$ are defined as in (12). For text and image, we have

$$\mathrm{RBW}_{\mathrm{text\&image}} = k \times \mathrm{RRBW}_{\mathrm{text\&image}} \tag{23}$$

  where k is the number of requests for text and image.

  Allocate the remaining bandwidth $\mathrm{BW}_{\mathrm{left}}$ among each media according to its priority. Given each media priority, P_{audio}, P_{video}, or $P_{\mathrm{text\&image}}$, we calculate $\mathrm{BW}_{\mathrm{audio}}$, $\mathrm{BW}_{\mathrm{video}}$, and $\mathrm{BW}_{\mathrm{text\&image}}$ as in (14)–(16), respectively.
- Calculate SA_τ and SV_τ for audio and video, respectively, i.e.,

$$SA_\tau = \frac{\mathrm{BW}_{\mathrm{audio}}}{\mathrm{RBW}_{\mathrm{audio}}} \tag{24}$$

$$SV_\tau = \frac{\mathrm{BW}_{\mathrm{video}}}{\mathrm{RBW}_{\mathrm{video}}}. \tag{25}$$

- Send back audio object with bandwidth $\mathrm{BW}_{\mathrm{audio}}$ and video object with $\mathrm{BW}_{\mathrm{video}}$.
- Send back the first $\mathrm{BW}_{\mathrm{text\&image}}/\mathrm{RRBW}_{\mathrm{text\&image}}$ requests for text and image. If there are some requests for text or image that have not been served by $\mathrm{Max}_{\mathrm{delay}}$, they will be moved to the highest priority queue.

TABLE I
BREAKDOWN OF DOCUMENT TYPES AND SIZES FOR ALL DATA SETS

Web Server	HTML	Images	Sound	Video	Dynamic	Formatted	Other
Waterloo	38.7	50.1	0.01	0.006	0.3	3.7	7.18
Calgary	47.1	50.3	0.1	0.3	0.04	1.0	1.16
Saskatchewan	55.6	36.5	0.1	0.004	6.7	0.02	1.08
NASA	30.7	63.5	0.2	1.0	2.6	0.01	1.99
ClarkNet	19.9	78.0	0.2	0.007	1.2	0.01	0.68
NCSA	51.1	48.1	0.2	0.1	0.01	0.006	0.48

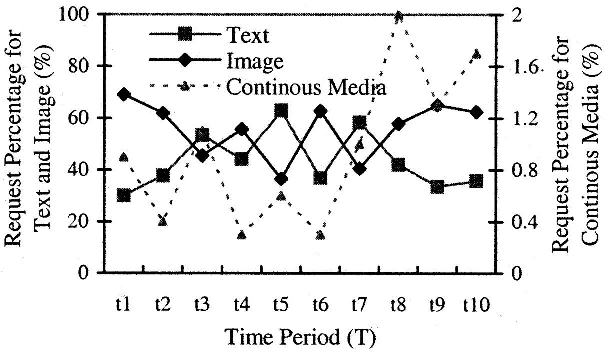

Fig. 8. Variation of requests from different types of media.

V. SIMULATION RESULTS

The simulations in this section are to demonstrate effectiveness of our proposed QoS-adaptive proxy-caching scheme for mixed media.

A. Simulation Setup

1) Client Request Model: In our experiment, client requests follow Zipf distribution [22] with ON/OFF behavior [23]. As shown in [23], we use the following Pareto probability distribution function to model ON/OFF behavior:

$$\Gamma_w(t_w) = 1 - e^{-(t_w/a)^b} \tag{26}$$

where $a = 0.328$ and $b = 1.47$.

Table I depicts the distribution patterns of client requests among different types of media shown in [23]. We use the distribution of those data to generate requests in our simulation.

2) Media Distribution: We analyze the behavior of cache resource management. The generated requests under different distributions for different types of media are shown in Fig. 8. Note that since the number of requests for continuous media is rather small compared to that of other media, we use an enlarged scale on the right side for continuous media. In fact, even a very small variation in continuous-media requests will affect the caching performance greatly because a huge amount of data needs to be transferred for one request.

We use a layered scalable codec, PFGS [24], as video object. PFGS source coder encodes input video into two layers: one is the base layer (BL) that carries the most important information, and the other is the enhancement layer (EL) that carries less important information. Different priorities are assigned for BL and EL.

3) Network Condition: A two-state Markov model is used to simulate the network condition (see Fig. 9) [25]. Loss process denotes the series that results from measuring the packet status continuously, which can be modeled as a discrete-time Markov

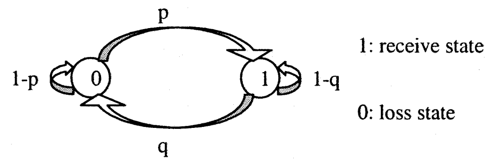

Fig. 9. Two-state Markov model.

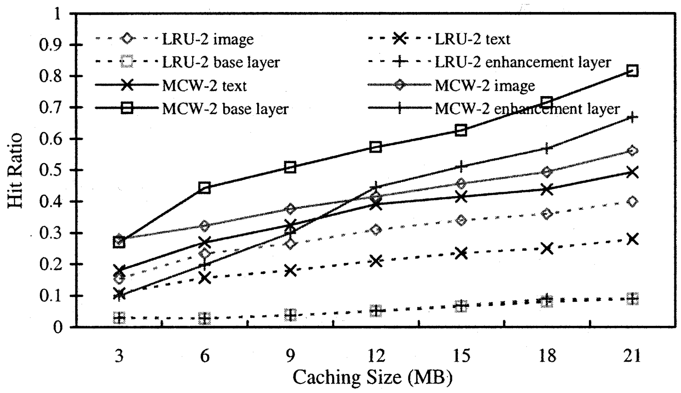

Fig. 10. Comparison of MCW-2 and LRU-2 for all types of media.

chain with two states. Note that here, “1” denotes correctly receives a packet and “0” denotes a packet loss. The current state x_i of the stochastic process depends only on the previous value x_{i-1}.

To demonstrate the effectiveness of our caching scheme, we conducted simulations under two different kinds of network conditions. In the high-bandwidth case, the bit rate varies from 920 to 1080 kbits/s. In the low-bandwidth case, the bit rate varies from 100 to 160 kbits/s. We choose these two cases to simulate typical Ethernet and DSL access. We had also tested in the medium-bandwidth case, which has a bit rate which varies from 360 to 480 kbits/s; similar results were achieved.

4) Video Quality Measurement: We use peak signal-to-noise ratio (PSNR) as a metric to measure objective quality of video. For an 8-bit image with intensity values between 0 and 255, the PSNR is defined as $\mathrm{PSNR} = 20\log_{10}(255/\mathrm{RMSE})$, where RMSE stands for root-mean-squared error. Given an original $N \times M$ image f and the compressed or degraded image f', the RMSE can be calculated as

$$\mathrm{RMSE} = \sqrt{\frac{1}{N \times M} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} [f(x, y) - f'(x, y)]^2}. \quad (27)$$

B. Simulation Results

1) Performance of Replacement Policy: This simulation is to demonstrate effectiveness of our media-characteristic-weighted replacement policy, MCW-2. The MCW-2 algorithm is compared with the recency-based replacement policy using LRU-2. The total caching size varies from 3 to 21 Mbytes).

Fig. 10 shows comparison results of the hit ratio for mixed media using MCW-2 and LRU-2. It can be seen that MCW-2 outperforms LRU-2 for all types of media. Having considered the tendency and priority of continuous media in MCW-2, the hit ratio of continuous media, especially for the base layer of the media, is significantly higher than the one obtained by LRU-2.

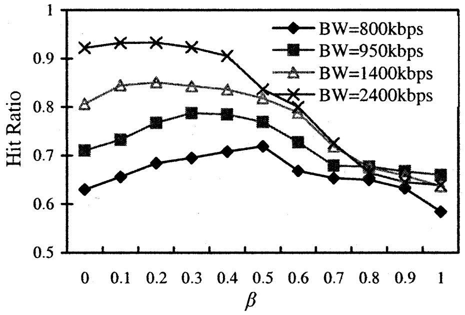

Fig. 11. Performances of parameter β in the weight function under different conditions.

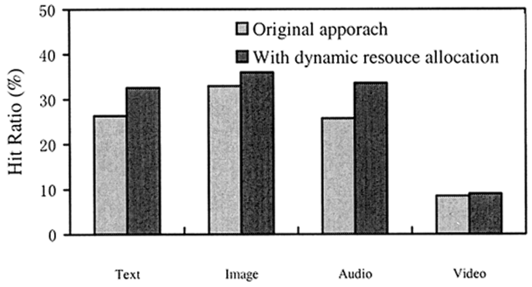

Fig. 12. Performance of cache resource re-allocation scheme.

Next, we analyze the cache performance under different β values in the weighting function. If parameter β is 0, we use the tendency as the only criterion for weight. If the parameter β is set to 1, the frequency is used as the only criterion for weight. Fig. 11 shows the performance of hit ratio in the weight function under different situations. It can be seen from Fig. 11 that combining the frequency and tendency together can outperform either one alone. More specifically, if the bandwidth from the proxy to server is low, the highest hit ratio occurs when β is relatively large. This is because under such a low-bandwidth condition, the probability of successful pre-fetching is relatively low; increasing the Tendency impact on weight calculation would prevent the successive content to be evicted out of cache. Therefore, choosing a larger β would yield a better hit ratio, and vice-versa in the high-bandwidth case. That is to say, different situations (e.g., bandwidth) will yield different optimal β values. So, in (1), the β value is not fixed and left the system (proxy) administrator to set it up based on the real network configuration.

2) Performance of Dynamic Cache Resource Re-Allocation: This simulation is to show the performance of our dynamic cache resource re-allocation scheme. Note that in this experiment, to reduce the influence of pre-fetch and to show the effectiveness of dynamic resource re-allocation alone, the hit ratio of continues media we studied here is the one without using pre-fetching. In this simulation, the total number of requests been send out is 100 000. We assign different importance levels for different types of media. More specifically, the priorities of text, image, audio, and video are set as 30, 50, 40, and 60, respectively.

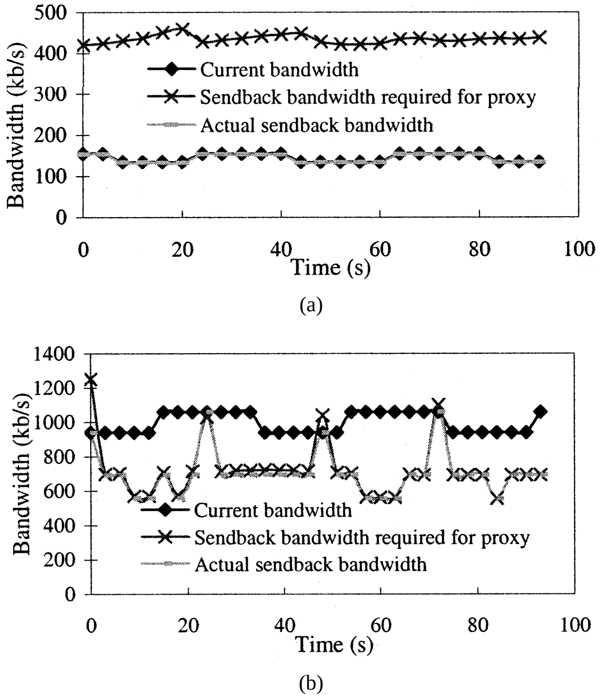

Fig. 13. Send-back scheduling for clients. (a) Low-bandwidth case. (b) High-bandwidth case.

TABLE II
PSNR Comparison Results for Client Scheduling Scheme

Scheme	Low bandwidth	High bandwidth
With client scheduler	19.97	35.55
Without scheduler	14.15	30.15

Fig. 12 shows the hit-ratio comparisons with and without cache resource re-allocation scheme. As shown in Fig. 12, with the dynamic resource re-allocation, better performance can be achieved. We obtain higher hit ratios for those four types of media since we take the request patterns into account. Note that the hit-ratio's increment of important media is higher than those of unimportant ones. In real applications, these priorities can be set by users or the cache manager.

3) Performance of Network Resource Management:

a) Performance of Client Send-Back Scheduling: To demonstrate effectiveness of client send-back scheduler, simulations are applied to two kinds of clients. The first type of clients has low bandwidth with an average value of 144 kbits/s. The second one has high bandwidth with an average value of 1 Mbits/s. We assume infinite bandwidth on the link from proxy to server in this simulation so as to study the client send-back scheduling only. The finite-bandwidth case would be analysis in simulation D.

Fig. 13 shows the performance of client send-back scheduling. It can be seen from Fig. 13 that the proposed send-back scheduling works well in both cases. It can adapt to the current network condition and can efficiently utilize the bandwidth resource.

Table II shows the comparison results of average PSNR for the test sequence *Foreman* with and without send-back scheduling. It can be seen that better video quality can be achieved with the client send-back scheduling scheme under different bandwidth.

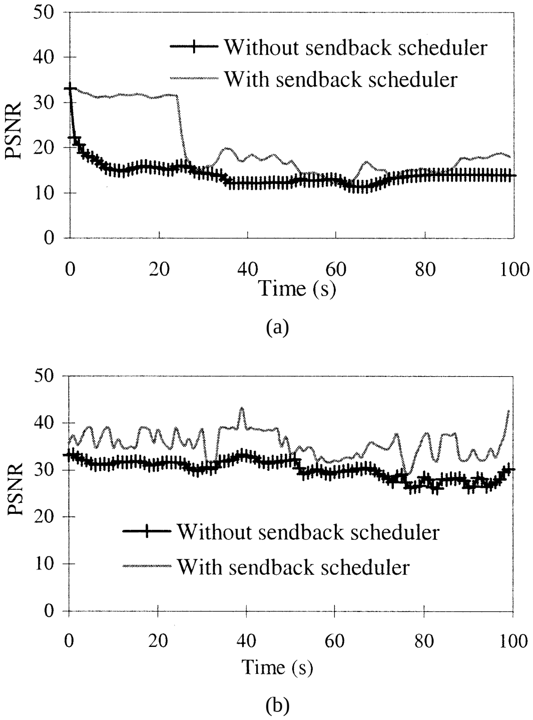

Fig. 14. PSNR of the PFGS video send-back by proxy. (a) Low-bandwidth case. (b) High-bandwidth case.

Fig. 14 illustrates the PSNR comparison results of the video send-back scheduling using the two algorithms. It can be seen that the video send-back by proxy with scheduler usually has higher PSNR than that without a scheduler. Notice that in the low-bandwidth case, there is a sharp drop in the PSNR at frames 25 and 26 in our scheme. This is because there is a packet loss in the base layer. In the scheme without a scheduler, packet losses occur more frequently and many packet losses occur in the base layer, since no scheduling or error protection are used.

Fig. 15 shows reconstructed frames of sequence *Foreman* with and without send-back scheduler under different network conditions. From Figs. 14 and 15 and Table II, it can be seen that the proxy with our proposed client send-back scheduling obtains better results than the one without scheduling scheme in packet-loss networks both subjectively and objectively.

b) Performance of Server Request Scheduling: This simulation is to demonstrate that our proposed server-request scheduling scheme can adapt to the available network bandwidth. The performance of server-request scheduling scheme for miss and pre-fetch requests is illustrated in Fig. 16. It can be seen that the request-scheduling scheme can generate request adapting to the available network bandwidth.

c) Performance of Integrating Two Scheduling Schemes: The request and send-back scheduling schemes are combined in the following simulation. The bandwidth from proxy to server was set to an average of 2 Mbits/s.

Fig. 17 shows the PSNR comparison results in four cases (with or without server-request and send-back scheduler). It can be seen that better video quality can be achieved using combination of request and send-back schedulers. Notice that in

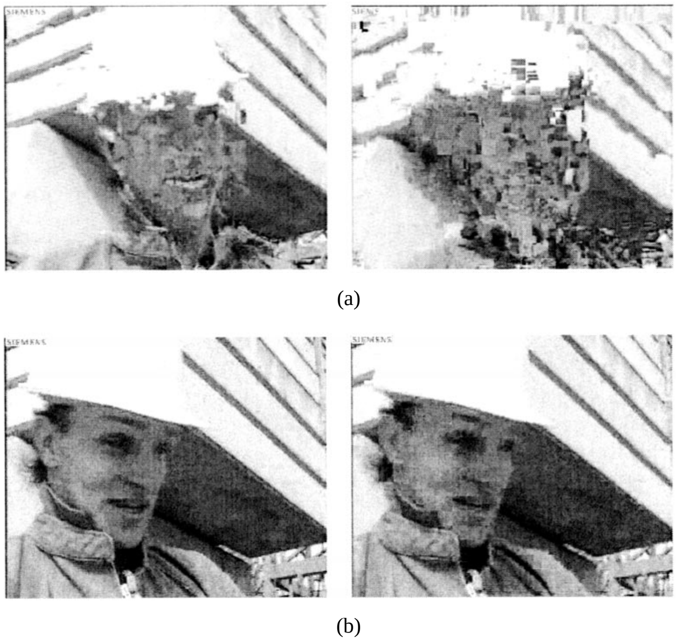

Fig. 15. Subjective quality comparisons of the send-back scheme for *Foreman*. (a) Low-bandwidth case. (b) High-bandwidth case. Left: with send-back scheduler. Right: without send-back scheduler.

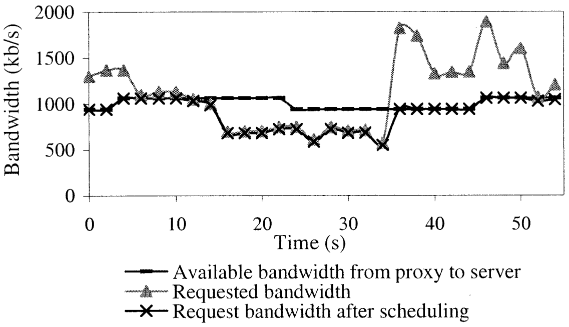

Fig. 16. Server request scheduling scheme.

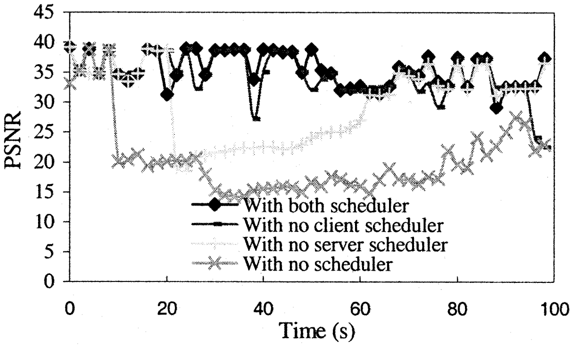

Fig. 17. PSNR comparison results of request and send-back scheduling schemes.

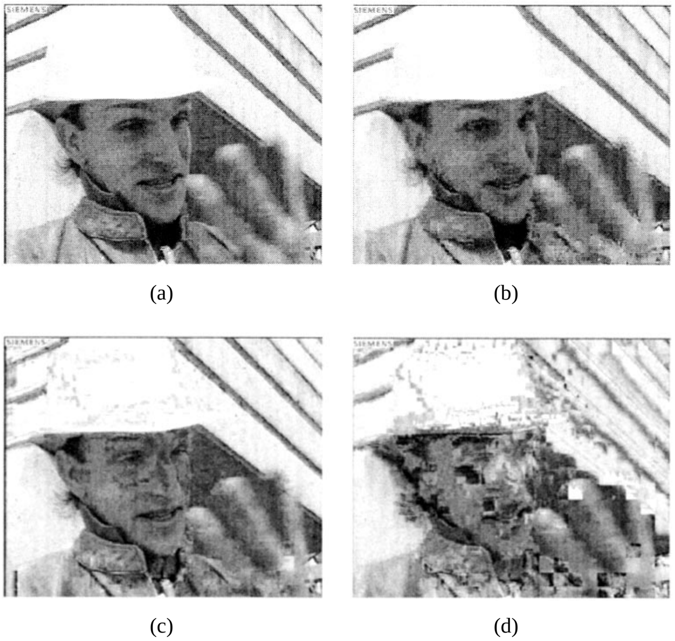

Fig. 18. Subjective quality comparisons of the two scheduling schemes for *Foreman*. (a) With send-back and server-request scheduling scheme. (b) With send-back and without server-request scheduling scheme. (c) With server-request and without send-back scheduling scheme. (d) Without send-back and server-request scheduling scheme.

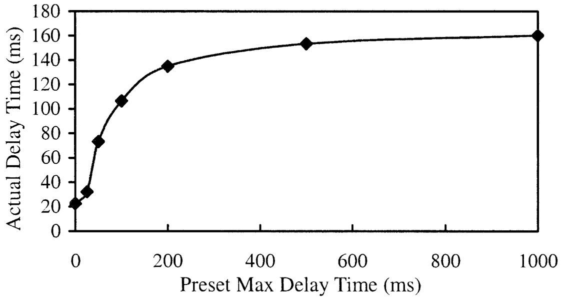

Fig. 19. Nonreal-time media delay analysis.

Fig. 17, there is sharp drop in the scheme without sever-request scheduler. That is caused by base-layer packet loss. In the high-bandwidth case, most of the cached objects can be sent back, therefore, the curve without client send-back is very close to that with both schedulers. The results of client with low bandwidth have not been shown in this paper. This is because in that case, the result of send-back without scheduler is too poor to be decoded at all.

Fig. 18 shows reconstructed frames of sequence *Foreman* with and without client /server scheduler in the high-bandwidth case.

From Figs. 17 and 18, it can be seen that our proposed approach obtains better results than the other schemes under packet-loss networks, both subjectively and objectively.

4) Analysis of Nonreal-Time Media Scheduling Delay: This simulation is performed to analyze the scheduling delay for nonreal-time media. As shown in Fig. 19, if we set the delay bound to 0 ms, there will still be a 22-ms delay because of the system overhead. If we set it larger, the actual delay will increase rapidly at the beginning, but will slow down soon. This is because if we set the delay bound high enough, most of the requests will have high possibility to be well served before this delay bound. This implies that the actual delay will not be very long, even if we set a large delay bound. Notice that in all the previous simulations, we set our max delay bound to 200 ms and the average actual delay is 114 ms.

In summary, our simulation results presented in this section conclusively demonstrate that: 1) our media-characteristic-weighted caching replacement algorithm is very effective to cache both continuous media and noncontinuous media; 2) our cache resource re-allocation approach can adapt to the current request patterns and achieve good results; 3) client send-back scheduling can adaptively allocate resource among different types of media under varying network conditions, ensure real-time delivery of continuous media, and provide protection for base layer; 4) server-request scheduling can schedule miss and pre-fetch requests based on the estimated bandwidth from proxy to server; and 5) the scheduling latency of nonreal-time media generated by our scheme is moderate.

VI. Conclusions

This paper addresses how to cache mixed media in a proxy server for multimedia streaming over the Internet. The main contributions of this paper are summarized as follows. First, it presents the client-proxy and proxy-server bandwidth monitors that can dynamically estimate the available network bandwidth. Then, a replacement policy considering the characteristics of different types of media and different request patterns are introduced, which can improve the hit ratio for multiple types of media. A cache resource re-allocation scheme is presented to improve the cache utilization by adapting to network conditions and media characteristics. Moreover, a QoS-adaptive miss strategy and pre-fetching algorithm that fully consider the continuous- and noncontinuous-media characteristics are also described. Lastly, a weighted request scheduling scheme that efficiently allocates the network resource between proxy and server among different type of requests together with a send-back scheduling scheme that efficiently utilizes the network resource between client and proxy based on media characteristics are proposed and analyzed.

Simulation results show that (from Fig. 10) our proposed replacement algorithm can achieve high hit ratio for each media and also the overall quality. Simulations using mixed media with multiple priorities and different request patterns demonstrated that our proposed caching scheme adapts fairly well to network bandwidth variations and achieves good quality under different network conditions (as in Figs. 13–18).

Acknowledgment

The authors would like to thank Prof. A. Zhou from Fudan University for the support of F. Yu on this work. The authors also respectively thank Prof. Y. H. Hu and B. Schwarz from University of Wisconsin at Madison, and Prof. L. Gao from the University of Massachusetts for discussions and critiques. They thank Prof. S.-H. G. Chan from Hong Kong University of Science and Technology for reading this manuscript. Dr. S. Li and Dr. F. Wu from Microsoft Research China for providing PFGS codec for the simulations are acknowledged.

References

[1] C. Aggarwal, J. L. Wolf, and P. S. Yu, "Caching on the world wide web," *IEEE Trans. Knowledge Data Eng.*, vol. 11, pp. 94–107, Jan./Feb. 1999.

[2] G. Barish and K. Obraczka, "World Wide Web caching: Trends and techniques," *IEEE Commun. Mag., Internet Technol. Series*, pp. 178–184, May 2000.

[3] A. Chankhunthod, P. Danzig, C. Neerdaels, M. Schwartz, and K. Worrell, "A hierarchical internet object cache," in *Proc. USENIX Tech. Conf.*, Jan. 1996, pp. 153–163.

[4] A. Ortega, F. Carignano, S. Ayer, and M. Vetterli, "Soft caching: Web cache management techniques for images," *IEEE Multimedia Signal Processing*, pp. 475–480, June 1997.

[5] S. Sen, J. Rexford, and D. Towsely, "Proxy prefix caching for multimedia streams," in *IEEE INFOCOM*, Mar. 1999, pp. 1310–1319.

[6] A. San and D. Sitaram, "Buffer management policy for an on-demand video server," in *IBM Research Report RC 19347*. NY: T.J. Watson Research Center, Jan. 1993.

[7] E. Bommaiah, K. Guo, M. Hofmann, and S. Paul, "Design and implementation of a caching system for streaming media over the Internet," in *IEEE Real-Time Technology and Applications Symp. (RTAS'2000)*, May 2000, pp. 111–121.

[8] R. Tewari, H. M. Vin, A. Dan, and D. Sitaram, "Resource-based caching for web servers," in *Proc. SPIE/ACM Conf. Multimedia Computing and Networking (MMCN)*, Oct. 1998, pp. 191–204.

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

[10] H. Chou and D. DeWitt, "An evaluation of buffer management strategies for relational database systems," in *Proc. 11th VLDB Conf.*, Aug. 1985, pp. 127–141.

[11] E. J. O'Neil, P. E. O'Neil, and G. Weikum, "The LRU-k page replacement algorithm for database disk buffering," in *Proc. Int. Conf. Management of Data*, May 1993, pp. 297–306.

[12] T. P. Kelly, Y. M. Chan, S. Jamin, and J. K. MacKie-Mason, "Biased replacement policies for Web caches: Differential Quality-of-Service and aggregate user value," in *4th Int. Web Caching Workshop*, San Diego, CA, Mar. 1999.

[13] E. Cohen and H. Kaplan, "Prefetching the means for document transfer: A new approach for reducing Web latency," in *Proc. IEEE INFOCOM'2000*, Mar. 2000, pp. 854–863.

[14] L. Fan, Q. Jacobson, P. Cao, and W. Lin, "Web prefetching between low-bandwidth clients and proxies: Potential and performance," in *Proc. SIGMETRICS '99*, June 1999, pp. 178–187.

[15] R. Rejaie, M. Handley, H. Yu, and D. Estrin, "Proxy caching mechanism for playback streams in the internet," in *Proc. 4th Int. Web Caching Workshop*, San Diego, CA, Mar. 1999.

[16] Q. Zhang, W. Zhu, and Y.-Q. Zhang, "Resource Allocation for multimedia streaming over the Internet," *IEEE Trans. Multimedia*, vol. 3, pp. 339–355, Sept. 2001.

[17] W. Zhu, Q. Zhang, and Y.-Q. Zhang, "Network-adaptive rate control with unequal loss protection for scalable video over Internet," in *Proc. ISCAS 2001*, vol. 5, May 2001, pp. 109–112.

[18] R. E. Blahut, *Digital Transmission of Information Reading*. Reading, MA: Addison-Wesley, 1990.

[19] J. R. Moorman and J. W. Lockwood, "Multiclass priority fair queuing for hybrid wired/wireless Quality of Service support," in *IEEE Int. Workshop of Mobile Multimedia Communications (MOMUC)* , Aug. 1999, pp. 43–50.

[20] T.-G. Kwon, S.-H. Lee, and J.-K. Rho, "Scheduling algorithm for real-time burst traffic using dynamic weighted round robin," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 6, June 1998, pp. 506–509.

[21] M. Katevenis, E. Markatos, and I. Mavroidis. Weighted round-robin scheduler using per-class urgency counters' (ICS-FOURTH). [Online]. Available: http://archvlsi.ics.forth.gr/muqpro/classSch.html.

[22] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. IEEE INFOCOM*, Mar. 1999, pp. 126–134.

[23] M. F. Arlitt and C. L. Williamson, "Web server workload characterization: The search for invariants," in *ACM/SIGMETRICS*, May 1996, pp. 126–137.

[24] 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, Dec. 1999.

[25] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and modeling of the temporal dependence in packet loss," in *Proc. IEEE INFOCOM*, Mar. 1999, pp. 345–352.

Fang Yu received the B.S. degree in computer science from Fudan University, Shanghai, China, and the M.S. degree in computer science from the University of California at Berkeley, where she is currently working toward the Ph.D. degree in the Electrical Engineering and Computer Science Department.

Her research interests include streaming applications, network protocols, and optical networks.

Qian Zhang (M'00) received the B.S., M.S., and Ph.D. degrees from Wuhan University, Wuhan, China, in 1994, 1996, and 1999, respectively, all in computer science.

She joined Microsoft Research Asia, Beijing, China, in July 1999, as an Associate Researcher in the Internet Media Group and is now a Researcher of the Wireless and Networking Group. She has published over 40 refereed papers in international leading journals and key conferences in the areas of wireless/Internet multimedia networking, wireless communications and networking, and overlay networking. She has been granted more than a dozen pending patents. Her current research interest includes multimedia delivery over wireless, Internet, next-generation wireless networks, P2P network/ad hoc network. Currently, she is participating in TCP/IP header compression in ROHC WG in IETF. She is the principal contributor of the IETF ROHC-TCP WG draft.

Wenwu Zhu (S'92–M'97–SM'01) received the B.E. and M.E. degrees from the National University of Science and Technology, Hefei, China, in 1985 and 1988, respectively, the M.S. degree from Illinois Institute of Technology, Chicago, and the Ph.D. degree from Polytechnic University, Brooklyn, NY, in 1993 and 1996, respectively, all in electrical engineering.

From August 1988 to December 1990, he was with the Graduate School, University of Science and Technology of China (USTC), and the Chinese Academy of Sciences (the Institute of Electronics), Beijing, China. He joined Microsoft Research Asia, Beijing, China, in 1999 as a Researcher in the Internet Media Group. He is currently a Research Manager of the Wireless and Networking Group. Previously, he was with Bell Labs, Lucent Technologies, Whippany, NJ, Holmdel, NJ, and Murray Hill, NJ, as a Member of Technical Staff during 1996–1999. While with Bell Labs, he performed research and development in the areas of Internet video, video conferencing, and video streaming over IP networks. He has published over 100 refereed papers in international leading journals and key conferences in the areas of wireless/Internet video delivery, wireless/Internet multimedia communications and networking, and has contributed to the IETF ROHC WG draft on robust TCP/IP header compression over wireless links. He is the inventor of more than a dozen pending patents. His current research interest is in the area of wireless/Internet multimedia delivery and multimedia networking.

Dr. Zhu has served as Guest Editor for Special Issues on Streaming Video and Wireless Video in IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (T-CSVT). He also serves a Guest Editor for the Special Issue on Advanced Mobility Management and QoS Protocols for Wireless Internet in the IEEE JOURNAL OF SELECTED AREAS IN COMMUNICATION. He received the Best Paper Award in IEEE T-CSVT in 2001. He is a member of Eta Kappa Nu, the Visual Signal Processing and Communication Technical Committee, and the Multimedia System and Application Technical Committee of the IEEE Circuits and Systems Society. He is also a member of the Multimedia Communication Technical Committee of the IEEE Communications Society.

Ya-Qin Zhang (S'87–M'90–SM'93–F'98) received the B.S. and M.S. degrees in electrical engineering from the University of Science and Technology of China (USTC), Hefei, Anhui, China, in 1983 and 1985, and the Ph.D. degree in electrical engineering from George Washington University, Washington, DC, in 1989.

He is currently the Managing Director of Microsoft Research Asia, Beijing, China, in 1999. Previously, he was the Director of the Multimedia Technology Laboratory, Sarnoff Corporation, Princeton, NJ (formerly David Sarnoff Research Center and RCA Laboratories). Prior to that, he was with GTE Laboratories Inc., Waltham, MA, from 1989 to 1994. He has been engaged in research and commercialization of MPEG2/DTV, MPEG4/VLBR, and multimedia information technologies. He has authored and co-authored over 200 refereed papers in leading international conferences and journals, and has been granted over 40 U.S. patents in digital video, Internet, multimedia, wireless, and satellite communications. Many of the technologies he and his team developed have become the basis for start-up ventures, commercial products, and international standards. He serves on the Board of Directors of five high-tech IT companies and has been a key contributor to the ISO/MPEG and ITU standardization efforts in digital video and multimedia.

Dr. Zhang served as the Editor-In-Chief for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY from July 1997 to July 1999. He was the Chairman of the Visual Signal Processing and Communications Technical Committee of the IEEE Circuits and Systems (CAS) Society. He serves on the editorial boards of seven other professional journals and over a dozen conference committees. He has received numerous awards, including several industry technical achievement awards and IEEE awards, such as the CAS Jubilee Golden Medal. He was named "Research Engineer of the Year" in 1998 by the Central Jersey Engineering Council for his "leadership and invention in communications technology, which has enabled dramatic advances in digital video compression and manipulation for broadcast and interactive television and networking applications." He recently received The Outstanding Young Electrical Engineer of 1998 award.