Proxy Cache Admission and Replacement Policies for Layered Video Streaming*

Chung-Ming Huang, Chia-Ching Yang and Chi-Chan Lee
Laboratory of Multimedia Mobile Networking
Department of Computer Science and Information Engineering
National Cheng Kung University, Tainan, Taiwan, R.O.C.
Correspondence: huangcm@locust.csie.ncku.edu.tw

***Abstract*—In this paper, ”ubiquitous streaming” is defined as a scenario that streaming client is allowed to change the video streaming session (i) from A network to B network, e.g., from ADSL to 3.5G or from 3.5G to WLAN, and even (ii) from A device to B device, e.g., from a desktop to a smartphone, on the run time, i.e., the streaming session keeps continuity when the attached network and the device of streaming client are changed. In the 3-tier Server-Proxy-Client network architecture, the proxy can provide cached and relayed capabilities to reduce the workload of server and distribute streaming traffic. In order to satisfy with the characteristics of heterogeneous network and diversity of devices, the proxy should provide different qualities of media streaming using scalable video coding. To achieve adaptive network and device for ubiquitious streaming, proxy cache admission and replacement policies are designed with advances of scalable video coding for the 3-tier Server-Proxy-Client network architecture in this paper. The proposed layered cache admission and replacement policies can manage the proxy cache in consideration of the quality-of-service and the video popularity. In our experiments, we exhibit merits and performance results of the proposed cache admission and replacement mechanism.**

***Index Terms*—Proxy, scalable video coding, cache admission and replacement policies.**

I. Introduction

In recent years, video streaming is an attractive application service and rising rapidly over the current Internet, e.g., movie-on-demand. The traditional video streaming utilizes a 2-tier Server-Client network architecture to develop corresponding application services. But the 2-tier Server-Client system architecture is not suitable and unreliable to deal with multiple requests from a great number of clients. Thus, the 3-tier network architecture, Server-Proxy-Client, was proposed to reduce the workload of server and distribute network traffic. Each client is able to select a nearby proxy to decrease the transmission time. In addition to video streaming services, the technique of proxy was also widely used in the web application [1][2]. The major difference between the web application and video streaming service is with respect to proxy cache management, i.e., admission and replacement. Generally speaking, the video streaming service needs more storage space occupied in the proxy. Related cache admission and replacement policy become important issues for the video streaming service over the 3-tier network architecture.

Up to now, there are a variety of methods accessing to the Internet, like ADSL, WiFi, 3G, 3.5G, WiMAX, etc. Each kind of the connection possesses different capabilities to deliver video data. On the other aspect, various devices adopted by client, e.g., desktop, notebook, TV, PDA, smartphone, are emerging to fit the requirements of diverse situations. In this paper, we define ”ubiquitous streaming” as a scenario that streaming client is allowed to change the streaming session (i) from A network to B network, e.g., from ADSL to 3.5G or from 3.5G to WLAN, and even (ii) from A device to B device, e.g., from a desktop to a smartphone, on the run time, i.e., the streaming session keeps continuity when the attached network and the device are changed. The heterogeneous network and device become a challenge to provide ubiquitous streaming. In order to resolve challenges of ubiquitous streaming, scalable video coding is an appropriate solution for the characteristics of heterogeneous network and device. However, the traditional proxy cache management is not suitable for streaming layered video data. Thus, how to manage the proxy cache efficiently and execute the replacement policy properly for layered video data in ubiquitous streaming are our major concerns in this paper.

In this paper, we investigate layered proxy cache admission and replacement policies for ubiquitous streaming using the 3-tier Server-Proxy-Client network architecture. The proposed policies can manage the cache space in the proxy. There are two important issues to be addressed in this paper.

(1) Cache allocation issue: By the scalable video coding [4][5], each video clip can be compressed into one base-layer bitstream (BL) and several enhancement-layer (ELs). More ELs can improve visual quality, including larger spatial-resolution, smoother motion and better SNR quality [3][6]. Since BL and ELs possess different QoS functions, the traditional cache allocation scheme should be improved for dealing with layered video data.

(2) Cache admission and replacement issue: Since the capacity of the cache space in the proxy is limited, another basic and important issues are to decide whether a segment should be cached or not and to design replacement policies to replace previous cached data with new ones.

The rest of this paper is organized as follows. Section II overviews related works. Section III describes the proposed 3-tier ubiquitous video streaming system architecture. Section IV presents the proposed cache admission and replacement

*This research is supported by the National Science Council of the Republic of China, Taiwan under the contract number NSC 96-2219-E-006-008, National Cheng Kung University Yen Tjing-Ling Industrial Technology Research & Development Center and China Motor Corporation, and Intel Microelectronics Asia Ltd., Taiwan Branch.

policy for layered videos in the proxy. Section V exhibits experiment results. Finally we summarize and conclude this paper in Section VI.

II. Related Works

In this paper, we propose an adaptive cache admission and replacement policies for ubiquitous streaming using layered video in the 3-tier Server-Proxy-Client network architecture. Some researches related with these topics are investigated as follows.

When the user moves to a new network environment and changes to a newly attached proxy, which is called proxy handoff [1], the new proxy needs to provide video data for users as soon as possible to make video playback more smooth and reduce the user's waiting time. Thus, how to select the layered video source from the original proxy or the server (called buffer forwarding) is the significant issue. In this paper, we adopt our colleague's previous work to address related challenges of proxy handoff and buffer forwarding (see details in [7]). In [7], an Application-layer Proxy Handoff (APH) was proposed to deal with the multimedia session switching from one proxy to another. APH employs application-layer anycast to select one of the candidate proxies as the next proxy based on (1) the network condition between the mobile client and each candidate proxy and (2) the load balance among candidate proxies. The vital issues of switching to a new proxy that the proposed APH generalizes are (1) how to select the next proxy and (2) how to recover the session on the next proxy. Unlike the past works, the authors distill the switching process to be independent of L2 and/or L3 handoff and solves both issues in APH.

Besides, some researches studied proxy handoff for web applications over the mobile network environment [9][1]. In [9], Kim et al. proposed a handoff message protocol to enable handoff among the distributed proxy server system. When a MN (mobile node) moves to another proxy service area, handoff needs to be executed. The processing of proxy handoff is divided into three steps, which are (1) the proxy handoff initiation, (2) distilled data synchronization between two proxies, and (3) the handoff processing between the proxy and the web server. MN issues a handoff request message to the current proxy server, and then the proxy server sets up the file list which are not yet received by the MN. After that, the current proxy server may request the past proxy to transfer them instead of sending HTTP requests to the origin server. Finally, the original server is requested to transmit those files which have not been received by the past proxy.

In order to manage the connection between the server and the proxy, some researches investigated the problem of efficiently streaming video assets to end clients over a distributed infrastructure [10][11], including original servers and proxy caches. Chitra et al. introduced the server scheduling algorithms (Batching/Patching/Batch-Patching) and the proxy caching algorithms (Full/Partial/No caching with or without caching patch bytes), and analyzed the minimum backbone bandwidth consumption [8]. Their proposed Batch-Patching scheme manages the request between the proxy and the server. If the proxy possesses prefix-cached data at the end of the patching interval, the proxy will deliver prefix-cache data to the client, and suffix data requested from the server are forwarded to the client at the end of the batching interval. On the contrary, if the proxy contains no prefix cache data, the proxy will forward the request to the server and deliver video data at the end of the batching interval. The advantage of this scheme can effectively reduce the connection bandwidth needed between server and proxy. But the system performance depends on the batching interval time selection.

[1] In a mobile environment, the user may change his location to anywhere arbitrarily and hope to take the streaming service continuously during his whole journey, e.g., taking train from Paris to Berlin. Thus, in addition to Layer 2 base station handoff and Layer 3 IP handoff, the proxy also needs to be changed to have better performance

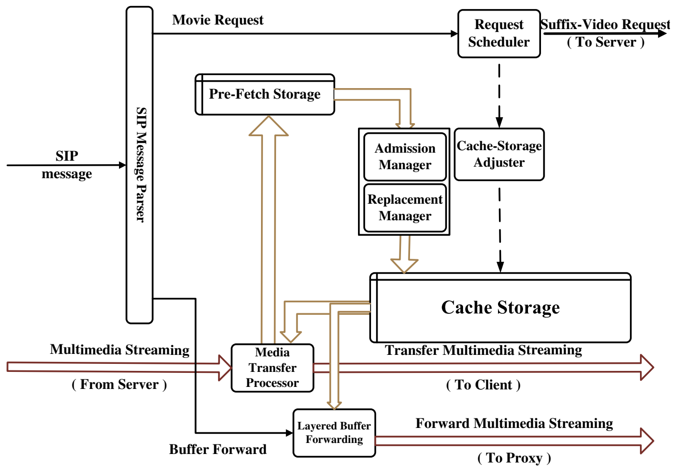

Fig. 1. Messages and data flows that are in proxy components.

III. System Architecture

Main components of the proposed 3-tier network architecture for ubiquitous streaming using layered video include proxy, proxy agent and client. Among which, the major functionality of the proxy is responsible for the layer-based cache management, and this management consists of the cache admission and replacement policies. Besides, the proxy agent takes the critical position in the proposed system. It is a bridge between the proxy and clients to offer proxy system information and cached data to clients. The client can refer these information from the proxy agent to select an appropriate proxy to request and stream preferred video data. Fig. 1 and Fig. 2 show messages and data flows of the main system components that are described as follows.

A. *Messages and data flows of proxy components*

In our proposed system architecture, the SIP protocol is adopted (i) between proxy and proxy agent, (ii) between proxy agent and clients, and (iii) between clients and proxy. For incoming messages, the proxy parsers and recognizes SIP messages. If the proxy's received request is a movie request, the proxy will check whether the segments of this requested movie are in the cache storage or not. If requested segments

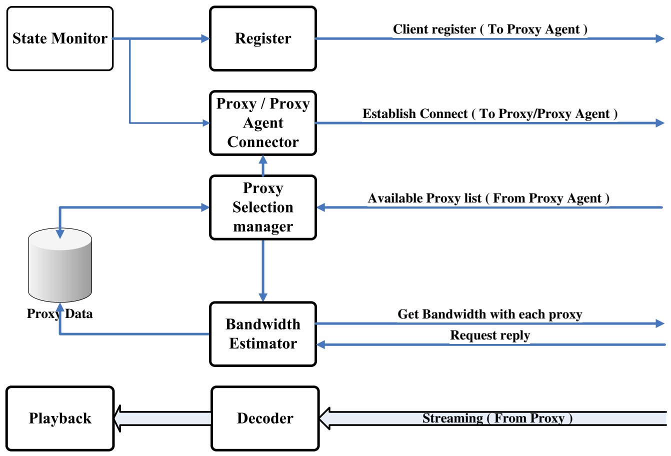

Fig. 2. Messages and data flows that are in client components

are in the cache storage, then it will transfer those segments to the client. If there are no requested segments in the cache storage, then the proxy will send a request to the server to get the corresponding multimedia data. If the proxy's received request is for buffer forwarding, then the proxy transfers the corresponding segments to the new proxy.

For incoming video data, when the proxy receives video data from the original server or other proxies, it will execute layer-based cache management to handle those video data. Among which, the proposed cache admission is to determine whether those received video data should be cached or not. When those received video data are determined to be cached and the cache storage capacity is not enough, then the cache replacement will be performed to release more cache capacity.

B. Messages and data flows of proxy agent components

The proxy agent is proposed to provide the service for the client's registration management and server information management. Besides, the proxy agent can provide the list of available proxy candidates. For incoming messages, if the received request is 'SIP REGISTER/UPDATE', then recording related information in the proxy agent database. If the received request is 'Proxy List Request', then fetching the available proxy list from the proxy agent database. The available proxy list is extracted from the proxy agent database by the client's request.

C. Messages and data flows of client components

The client will register to the proxy agent and get the available proxy list before establishing connections and getting video data. The proxy selection algorithm will be executed to determine a suitable proxy from the received proxy list. Then the client sends a request to the proxy, and get video data to decode and playback them. If the client changes the attached network, then the 'State Monitor' will record playback information and it will register/connect to the new network environment.

IV. The Proposed Cache Admission and Replacement Policies for Layered Streaming Videos

With the advances of scalable video coding, video resolution and quality can be scaled to users' distinct quality requests. However, the traditional proxy design is not suitable for managing layered video data. Thus we propose a corresponding proxy cache allocation, cache admission and replacement policies and suffix-video request scheduling for ubiquitous streaming using layered videos.

A. Dynamic proxy cache allocation

Considering heterogeneous networks and devices, layered video coding is adopted for the adaptive video streaming. Since the traditional proxy cache mechanism is not suitable for cache layered video data, a dynamic proxy cache allocation is proposed in this Section to store these layered video data based on the users' requesting behaviors. Essentially, users may request different qualities movies, and all of them also don't seldom request the same movie at the same time.

Generally speaking, the larger base-layer storage capacity can provide more distinct videos for users' requests, and the larger enhancement-layer storage capacity can provide more enhancement layers of each video to improve the video quality. In addition, the user's requesting behavior is another effective factor considered for the dynamic division of the proxy's storage. In many situations, the users' requesting behaviors are diverse with each other. According to the user's requesting behavior, the proposed dynamic cache division rule can flexibly adjust proxy cache storage for distinct layered data.

In the proposed dynamic proxy cache allocation, the base-layer space occupies N% of total capacity and the enhance-layer space occupies (100-N)% of total capacity. Thus, the determination of N that trades off the BL and ELs capacity is the first major concern in our work. By default, the initial value of N is 50. According to the users' requested types of video layers and the users' requested number of on-demand videos, parameter N will be changed continuously. The users' requested types of video layers may be a base-layer request REQ_{BL} or 'base and enhancement-layer' request REQ_{BL+EL}. When the frequency of REQ_{BL} is larger than the frequency of REQ_{BL+EL} , it means that the proxy needs more capacity for storing base layer data to satisfy heterogeneous requests. For example, when more clients request distinct videos, more base-layer capacity should be assigned to store more different videos. On the contrary, if the frequency of REQ_{BL+EL} is more than REQ_{BL}, it indicates that there are more clients requesting high quality video data. Thus, the proxy needs more capacity to store enhancement-layer data. The corresponding calculations for N are as follows:

$$R = REQ_{BL} + REQ_{BL+EL} \tag{1}$$

$$V = V_{BL} + V_{BL+EL} \tag{2}$$

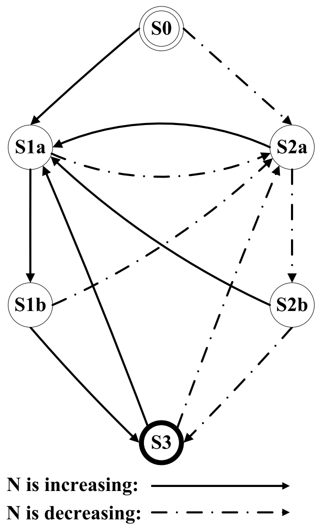

Fig. 3. The cache storage judgment to avoid the ping-pong effect.

$$N = 0.5 * [(REQ_{BL}/R) + (V_{BL}/V)], \tag{3}$$

where the number of videos of all requests is denoted as V_{BL} and the number of videos of enhancement layer requests is V_{BL+EL}.

Referring to Fig. 3, in order to avoid 'Ping-Pong Effect' in the dynamic division of the cache storage, we utilize the state diagram to determine whether the value of N needs to be changed or not. The initial state is S0, and state S1a and state S1b mean that the N's value is increasing once and twice respectively; state S2a and S2b mean that the N's value is decreasing once and twice respectively. When N decreases three times continuously or N increases three times continuously, the state diagram will be proceeded to the final state, on which N can be adjusted to change the division of the proxy's cache storage.

B. The cache admission and replacement policies

Under the condition of limited proxy cache capacity, the cache admission and replacement policies are to decide whether a segment should be cached or not based on the requested frequency of this segment. How to efficiently manage the storage space is the major concern.

The proposed algorithms are described in Algorithms 1 and 2. There are three kinds of combination for the incoming segments: base layer only, base layer and enhancement layer, and enhancement layer only. In the beginning of Algorithm 1, each incoming video stream is analyzed to identify which kind of segments it belongs to. Generally speaking, clients' requests only have two options: base layer only, base and enhancement layer. But in some situations, for the same video, the first client may request the base-layer video, and then the follow-up clients may request enhancement-layer data. Nevertheless, the proxy may cache only base-layer video first, and then send enhancement-layer requests to the server when other clients need more higher quality of this video. In our proposed algorithm, three conditions are all considered: base layer only, base layer and enhancement layer, and enhancement layer only. Each incoming video stream will be put into the pre-fetch cache storage at first, and the segment will be delivered to the client directly or to be cached in the proxy after the decision of the cache admission. The cache admission algorithm will verify incoming segments from the server: if the segment belongs to a single layer,e.g. a base layer or an enhancement layer, it will be compared with the request frequency of this segment using the threshold of the request frequency, and decide whether it should be cached or not. This threshold is defined as the least request frequency. When the storage space of the proxy is not enough after calculating the residual cached space, the proxy will make the decision of executing replacement policies.

Algorithm 1 The proposed CACHE ADMISSION algorithm

```
PROCEDURE CACHE_ADMISSION()
STORED ← FALSE
if (the incoming data belongs to a single layer) then
  if ( the segment belongs to base layer) then
    if (request number > threshold) then
      SEGMENT_STORED(Base)
    end if
  else
    SEGMENT_STORED(Enhancement)
  end if
else
  if (the request number of base layer segments > threshold)
  then
    while (STORED ≠ TRUE) do
      if (base layer segment size < free space of the base layer
      cache storage) then
        if (the enhancement layer segment size < free space
        of the enhancement layer cache storage) then
          add those segments into the cache storage
          STORED ← TRUE
        else
          CacheReplacement( Enhancement )
        end if
      else
        if (enhancement layer segment size < free space of
        the enhancement layer cache storage) then
          CacheReplacement( Base )
        else
          CacheReplacement( Enhancement )
        end if
      end if
    end while
  end if
end if
```

When the cache storage capacity is not enough, it needs to find the victim of out-of-date video segments based on the segment access rate and video access rate. In our proposed dynamic proxy cache allocation, the cache storage space is divided into two partitions, which are the base layer partition and the enhancement layer partition. The selected victim of segment will belong to the same layer partition as the next incoming segment that is decided to cached. If the next incoming segment decided to be cached belongs to base layer, the victim is chosen from the base layer partition;

otherwise, the victim is chosen from the enhancement layer partition. The replacement policies is depicted in Algorithm 3. The proposed algorithm is based on the frequency-based replacement mechanism. However, the chosen segment for replacement may be one of the popular movie segments. For this reason, we use the score function to calculate the priority, for which the segment with the least value of the priority will be the victim to replace. The request frequency of segment is denoted as f_s, and the time interval of the cached segment, which is from the time instant of caching into the proxy to the time instant of removing from the proxy, is denoted as T_s. The access rate of segment is

$$Rs = fs/Ts \tag{4}$$

The request frequency of video is considered as the mean of request frequency before each segment:

$$Rv(i) = (\sum_{i=1}^{i-1} Rs(i))/(i-1), i \neq 1 \tag{5}$$

The replacement priority is

$$P(i) = W * Rs(i) + (1 - W) * Rv(i) \tag{6}$$

According to the calculated replacement priority of each cached segment, the proxy can determine which cached segments will be replaced with the newly coming segments.

Algorithm 2 The SEGMENT STORED DECISION algorithm

```
PROCEDURE SEGMENT_STORED(LAYER_FLAG)
STORED ← FALSE
while (STORED ≠ TRUE) do
    if (segment size < free space of the base layer cache storage)
    then
        add this segment into the cache storage
        STORED ← TRUE
    else
        CacheReplacement( LAYER_FLAG )
    end if
end while
```

Algorithm 3 The CACHE-REPLACEMENT algorithm

```
REMOVE ← FALSE
id ← 0
if (LAYERFLAG is Base) then
    SegmentRemovePriority[id] ← CalculatePriority()
    while (REMOVE = FLASE) do
        for all (SegmentRemovePriority[id]) do
            if (LOCKED(id) = FALSE) then
                Remove this base layered segment
                if (Enhancement segment of Segment[id] exist) then
                    Remove this enhancement layered segment
                end if
                REMOVE = TRUE
            end if
        end for
    end while
else
    SegmentRemovePriority[id] ← CalculatePriority()
    while (REMOVE = FLASE) do
        for all (SegmentRemovePriority[id]) do
            if (Enhancement segment of Segment[id] exist) then
                if (LOCKED(id) = FALSE) then
                    Remove those segments
                    REMOVE = TRUE
                end if
            end if
        end for
    end while
end if
```

V. Experiments

In this Section, we show the merit of the proposed proxy cache allocation in consideration of the distinct user's behavior of accessing on-demand videos. As mentioned in Section IV-(A), if more distinct videos that users' request are accessed, more BL(Base Layer) capacity should be allocated; else if more high-quality videos are requested by users, more EL(Enhancement Layer) capacity should be allocated. The trade off allocation ratio N is defined in Eq.(3). In our analysis, it is assumed that the total buffer capacity is 100 GB. BL segment of each video is 10 MB, and EL segment of each segment is 100 MB. The hit ratio H_R is the evaluation index and means a probability that user searches his preferred video or needed ELs successfully.

Table I shows the relationship of distinct allocation algorithms w.r.t. hit ratio. We consider the following scenarios of user preferences: (1) more on-demand video that users request (2) high quality video that user request. In the first scenario, our proposed dynamic proxy cache allocation allocates much BL capacity, e.g. 80 percentage , and thus the hit ratio (0.8) is larger than that of fixed proxy cache allocation (0.5). In the second scenario, our proposed dynamic proxy cache allocation allocated much EL capacity, e.g. 70 percentage , and thus the hit ratio (0.7) is larger than that of fixed proxy cache allocation (0.5). It is noting that one important factor that may affect hit ratio is the total number of videos (T). With the larger of T, the hit ratio of the fixed and proposed algorithms both decrease and the minor difference may be obtained.

In addition to the aforementioned theorical analysis, the results of real implementation are also shown in Fig. 4 and Fig. 5. In Fig. 4, it is assumed that the ratio of a user requests for BL and (BL+ELs) is 1:3, that is, more high-quality videos are requested, and thus more EL capacity should be allocated. The result show that the EL hit ratio of the proposed dynamic proxy cache allocation is higher than that of the fixed proxy storage cache allocation. Besides, in Fig. 5, the more distinct videos are requested, and thus more EL capacity should be allocated. The result shows that when the number of videos increases, the BL hit ratio of the fixed and proposed algorithms both decrease, but the BL hit ratio of the proposed method is still better than that of fixed allocation algorithm.

VI. Conclusion

In this paper, we proposed the layered cache admission and replacement policies to manage the proxy cache for ubiquitous streaming. With the advance of scalable video coding, three major contributions are proposed dynamic proxy cache allocation, cache management. In consideration of the quality-of-service and the video popularity, the proposed proxy cache

TABLE I
THE HIT RATIO COMPARISONS FOR FIXED AND PROPOSED PROXY CACHE ALLOCATION.

	Fixed proxy cache allocation		**Proposed proxy cache allocation**					
	N = 0.5		**N = 0.3**		**N = 0.5**		**N = 0.8**	
	BL	EL	BL	EL	BL	EL	BL	EL
Allocated capacity	500	500	300	700	500	500	800	200
#videos in cache	50	50	30	70	50	50	80	20
Hit Ratio T=100	0.5	0.5	0.3	0.7	0.5	0.5	0.8	0.2
Hit Ratio T=200	0.4	0.4	0.15	0.35	0.4	0.4	0.4	0.1
Hit Ratio T=400	0.125	0.125	0.075	0.175	0.125	0.125	0.2	0.05

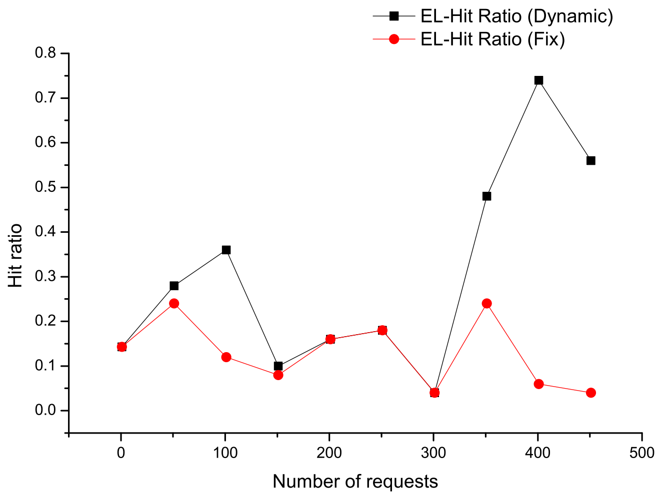

Fig. 4. The Hit ratio comparsion for fix N and dynamic N

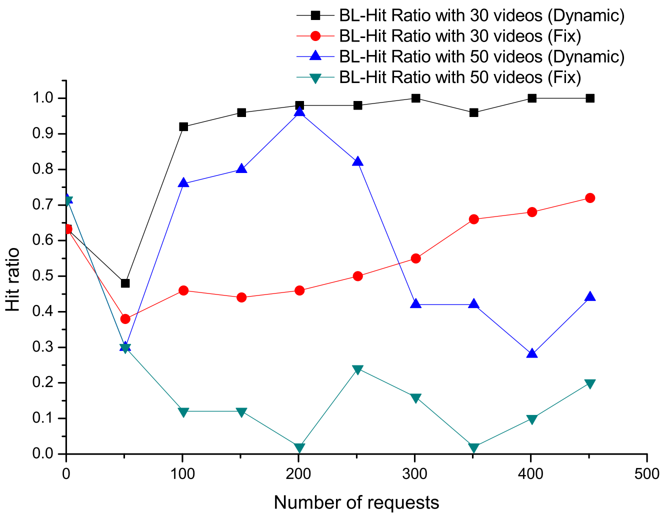

Fig. 5. The BL Hit ratio comparsion for different number of videos

allocation is able to allocate proper BL and EL capacities, and the proposed cache management is able to decide whether the segment should be cached or not. In our analysis, the proposed layered cache admission and replacement policies are able to increase the hit ratio effectively, so that the user can find out his preferred videos or needed video ELs in the proxy cache easily. Thus, corresponding streaming techniques are feasible solutions for delivering and managing layered video data in the 3-tier Server-Proxy-Client network environment.

REFERENCES

[1] G. Wenzheng and H. Abdelsalam, "Extended internet caching protocol: A foundation for building ubiquitous web caching", *Proceedings of the 2003 ACM Symposium on Applied Computing*, pp. 901-907, 2003.
[2] J. Aguilar, E. L. Leiss, "A coherence-replacement protocol for web proxy cache systems", *International Journal of Computers and Applications*, VOL. 28, NO. 1, pp.12-18, January 2006.
[3] M. Schaar, H. Radha, "A hybrid temporal-SNR fine-granular scalability for Internet video", *IEEE Transactions on Circuits and Systems for Video Technology*, VOL. 11, NO. 3, pp. 318-331, August 2003.
[4] T. Wiegand, J. Gary, G.J. Sullivan, et al., "Overview of the H.264/AVC video coding standard", *IEEE Transactions on Circuits and Systems for Video Technology*, VOL. 13, NO. 7, pp. 560-576, July 2003.
[5] F. Wu, S. Li, Y. Zhang, "A framework for efficient progressive fine granularity scalablevideo coding", *IEEE Transactions on Circuits and Systems for Video Technology*, VOL. 11, pp. 332-344, Mar 2001.
[6] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard", *IEEE Transactions on Circuits and Systems for Video Technology*, VOL. 11, pp. 301-317, Mar 2001.
[7] C.M. Huang and C.H. Lee, "Layer 7 Multimedia Proxy Handoff Using Anycast/Multicast in Mobile Networks", *IEEE Transactions on Mobile Computing*, VOL. 6, pp. 411-422, April 2007.
[8] C. Venkatramani, O. Verscheure, P. Frossard, K. Lee ,"Optimal Proxy Management for Multimedia Streaming in Content Distribution Networks", *Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video*, pp.147-154, 2002.
[9] K. Kim, H. Lee, and K. Chung,"A Distributed Proxy Server System for Wireless Mobile Web Service", *Proceedings of the 15th IEEE International conference on Information Networking*, pp. 749-754, February 2001.
[10] L Guo, S Chen, X Zhang, "Design and Evaluation of a Scalable and Reliable P2P Assisted Proxy for On-Demand Streaming Media Delivery", *IEEE Transactions on Knowledge and Data Engineering*, VOL. 18, NO. 5, pp.669-682, May 2006.
[11] S.H. Chang, R.I Chang, J.M. Ho, Y.J. Oyang, "A Priority Selected Cache Algorithm for Video Relay in Streaming Applications", *IEEE Transactions on Broadcasting*, VOL. 53, NO. 1, pp.79-91, March 2007.