Degraded Quality Service Policy with Bitrate based Segmentation in a Transcoding Proxy

Jung-hwa Lee, Yoo-hyun Park, *Member, KIMICS*

***Abstract*—To support various bandwidth requirements for many kinds of devices such as PC, notebook, PDA, cellular phone, a transcoding proxy is usually necessary to provide not only adapting multimedia streams to the client by transcoding, but also caching them for later use. Due to huge size of streaming media, we proposed the 3 kinds of segmentation – PT-2, uniform, bitrate-based segmentation. And to reduce the CPU cost of transcoding video, we proposed the DQS service policy.**

In this paper, we simulate the combined our previous two researches that are bitrate-based segmentation and DQS(Degraded Quality Service) policy. Experimental results show that the combined policy outperforms companion schemes in terms of the byte-hit ratios and delay saving ratios

***Index Terms*—adaptive streaming service, transcoding proxy, degraded quality service, video segmentation, streaming**

I. INTRODUCTION

The explosive increase in commercial usage of the Internet has resulted in a rapid growth in demand for audio and video streaming. This trend is expected to continue and a larger portion of Internet traffic will consist of multimedia streams in the future[1]. To reduce client-perceived access latencies as well as network traffic, an effective solution is to cache frequently used media at proxies close to clients. Streaming media could gain significant performance improvement from proxy caching, but traditional proxy systems are generally optimized for delivering conventional Web object, which may not meet the requirements of streaming applications [2].

And to support various bandwidth requirements for the heterogeneous ubiquitous devices, a transcoding proxy is usually necessary to provide not only adapting multimedia streams to the client by transcoding, but also caching them for later use. Due to transcoding proxy is inherited from the characteristics of multimedia proxy, it is useful to adopts the partial caching scheme to solve the problems caused by large size media objects. The researches about the segment-based transcoding proxy are only [3][4] and [5] that is our previous work. The result of [5] is outperformed than other strategies, but it uses the uniform segment, so there are lots of replacements. So, in our previous paper[6], we proposed the bitrate-based segmentation to reduce the number of replacements.

In general, video transcoding is a computation-intensive task. So, we proposed the DQS(degraded quality service) policy to reduce the CPU cost of transcoding, in our second previous paper[7].

In this paper, we combine the previous two ideas and simulate the combined idea. Our combined idea has good performance such as, DSR(delay saving ratio), BHR(byte hit ratio).

The remainder of this paper is organized as follows. Firstly, in section 2, we describe related works on the transcoding proxy. In section 3, we present our system architecture, describe the bitrate-based segmentation strategy and the DQS policy. In section 4, we present experiment results of the competing cache algorithm, and finally in section 5, we summarize our work and conclude the paper.

II. RELATED WORKS

Several caching strategies for streaming media have been proposed in recent years by caching a portion of a video file at the proxy such as sliding-interval caching[8], rate-split caching[9], prefix caching, and segment caching. Specially, prefix caching[10] algorithm caches the initial portion of a media object, called the prefix, at a proxy. Upon receiving a client request, the proxy immediately delivers the prefix to the client, meanwhile fetching the remaining portion, the suffix, from the origin server and relaying to the client. As the proxy is generally closer to the clients than the origin server, the startup delay for a playback can be remarkably reduced.

Segment caching generalizes the prefix caching paradigm by partitioning a media object into a series of segment, differentiating their respective utilities, and accordingly making caching decision. Various segment caching algorithms have been proposed in the literature by employing different segmentations and utility calculations. Segment-based caching strategies cache media objects in segments instead of in full. Recently, two types of segmentation strategies had been developed according to how the object is divided. One uses uniform

Manuscript received April 28, 2010; revised May 18, 2010; accepted June 4, 2010.

Jung-hwa Lee is with the Department of Computer Software Engineering, Dongeui University, Busan, 614-714, Korea (Email: junghwa@deu.ac.kr)

Yoo-hyun Park is with the Department of Computer Software Engineering, Dongeui University, Busan, 614-714, Korea (Email: yhpark@deu.ac.kr)

segmentation[1][5] and the other uses exponential segmentation[11].

Outstanding transcoding-enabled algorithm for adaptive content delivery are the full version only(FVO) and transcoded version only (TVO) schemes, and TeC. FVO[12] tries to reduce the network demand by caching full object version, while TVO[12] reduces the computation demand by caching the transcoded version.

Under the TeC scheme, Transcoding-enabled Caching(TeC)[13] is defined by TeC11, TeC12 and TeC2. TeC11 and TeC-12 cache at most one version of a video object at the proxy at any time. On the other hand, TeC2 may cache multiple versions of the same video and hence reduces the processing load on the transcoder. Tests of their performance using a synthesized and enterprized trace-driven simulation showed that TeC11 and TeC12 perform better than TeC2 under similar network capacity conditions, and that the performance of TeC2 is superior with heterogeneous network connectivity. However, all TeC algorithms evict one complete version even if the newly transcoded version is sufficiently smaller than the size of the victim.

And we revised PT2(Partial Transcoding)[14] from TeC system. In PT2, a video object is divided into two equal size. If proxy has no more room for caching new data, proxy finds a victim version, first, and then checks where the victim version is full-size or partial size. If it is full-size version, proxy evicts the suffix of the victim version. If it is partial-size(half-size of full-size), proxy evicts the version completely. Until the proxy has enough room to cache the new data, above procedures are repeated.

Chang et al. [15] formulated a generalized Profit Function(PF) to evaluate the profit obtained by caching each version of an object. However, their approach considers only the aggregate caching efficiency from caching multiple versions of small web objects, rather than video objects which need much more space for caching. Additionally, it employs only the delay saving ratio as a metric, whereas in fact the byte-hit ratio must also be considered, since the byte-hit ratio is important for streaming content caching. Kao et al.[16] formulated a generalized Video Profit Function(VPF) to estimate the benefit of caching partial or whole clips of various versions of video objects. VPF is revised function from PF.

Chang and Chen[3][4] proposed the ORG(Object Relation Graph) to manage the static relationships between video versions and CORT(Cached Object Relation Tree) to manage video objects cached in the proxy dynamically. This study use exponential segmentation to quickly discard a big chunk of video objects. But they didn't compare the uniform segmentation.

In this paper, we use the VPF as the replacement function and *bitrate-based segmentation* for reducing the replacement count. And also, we use the *DQS policy* to reduce the transcoding overheads.

III. BITRATE-BASED SEGMENTATION FOR TRANSCODING PROXY

The transcoding proxy must have features of streaming server as well as caching and transcoding system. From the aspect of the streaming server, it is useful to use special H/W like NS card[17]. NS card is devised to rectify some limitations of existing file I/O for networked multimedia, such as unnecessary data copying, lack of guaranteed storage access and overhead of network protocol processing. So, if the transcoding server use NS card, the system can get the efficient storage retrieval and transmission of large amount of video data. Figure 1 depicts the components of the generic transcoding proxy[5].

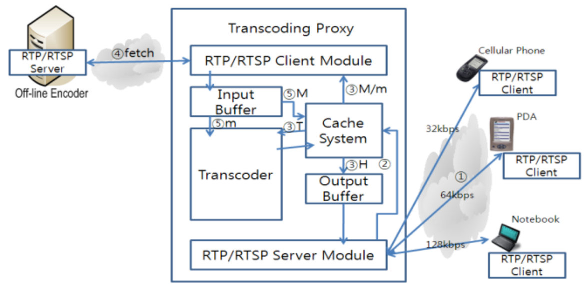

Figure 1: System architecture of the transcoding proxy

Recently, proxy caching of streaming media has been explored in a number of settings. Researchers have observed that most clients tends to watch the initial partitions of media objects, and make fewer accesses to later portions. Segment-based proxy caching strategies have been developed based on this observation. Segment-based caching strategies cache media objects in segments instead of in full. Among these schemes, prefix caching[10] was proposed earlier to segment the media object as a prefix segment and a suffix segment. More recently, two types of segmentation strategies had been developed according to how the object is divided. One uses uniform segmentation and the other uses exponential segmentation. From the perspective of optimization goals, these strategies emphasize either reducing the startup latency perceived by clients, or improving caching efficiency.

A. Bitrate-based Segmentation for transcoding proxy

This strategy was proposed in [6]. Let us assume that object O has *n*-versions at bitrates $b_0, b_1, \ldots, b_{n-1}$ (i.e., $b_0>b_1>\ldots>b_{n-1}$) and $b_i = b_{i+1} * 2$. Bitrate-based segmentation is a uniform segmentation in the same version, but it is exponential segmentation among versions. For example, in [Figure 4], if the bitrate of the highest version(version 0) is 512kbps, version 1 is 256kbps, version 2 is 128kbps, version 3 is 64kbps, and version 4 is 32kbps, the basic segment size is decided by 32kbps * *m*. The basic segment size is the minimum segment size in our system. If *m* is 1, the segment size of version 4 is 32kbps, and the size of segment according to its bitrate. So, the version 4 is divided by 4,096 byte(32 * 1024 / 8), and the version 0 is divided by 65,536 byte (512 * 1024 / 8). Consequently, the total number of segments in each version is same. So, if all version of same content are played at the same time, the consuming number of segments at given time is same.

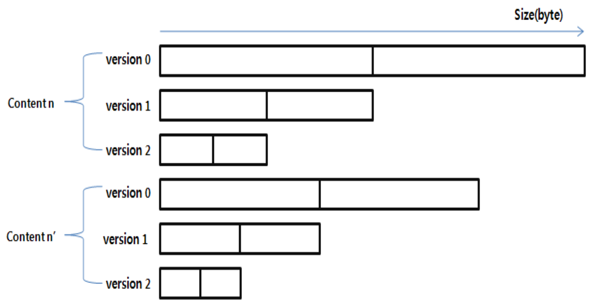

Figure 2: PT-2 Segmentation[14]

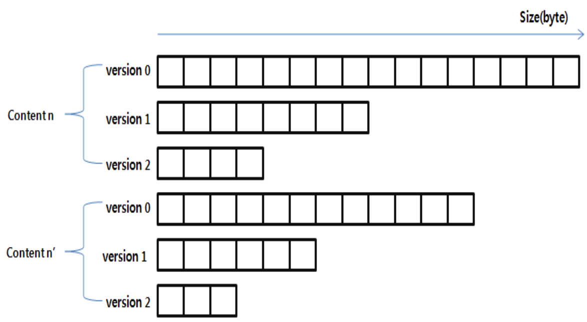

Figure 3: Uniform Segmentation[5]

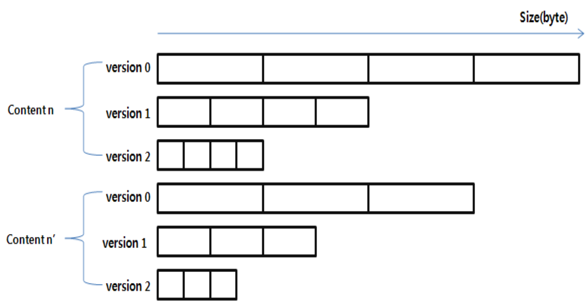

Figure 4: Bitrate-based Segmentation[6]

B. DQS(Degraded Quality Serivice) Policy

This policy was proposed in [21], and we simulated it in [7]. Transcoding is often computing intensive, especially for multimedia content. Research on developing efficient real-time transcoding algorithms has received much attention[20][21]. In the previous research[21], the authors use the term of "secondary hit" to reduce the CPU load in transcoding. Secondary hit is said to occur when the cache at the proxy has a lower fidelity version than requested. In that paper, they just proposed the "secondary hit". So, we adopted that idea and simulated it[7].

Let us assume that the proxy cached the *n*-th version of object $O(O_n)$. If it requested the *k*-th version of object $O(O_k,\ k < n,\ Quality(O_k) > Quality(O_n))$, normal transcoding proxy must make the object O_k to adapt the streaming service by transcoding. But, in DQS policy, the proxy directly send the lower quality version O_n for CPU resource saving.

IV. EXPERIMENTAL RESULTS

A. Simulation Model

We implemented a simulator for performance analysis. In the client model, client devices can be partitioned into 5 classes (15%, 20%, 30%, 20%, 15%). That is, each data item can be transcoded to 5 different versions by the transcoding proxy to satisfy the users' requirements. The bitrate of the 5 versions of each media object are assumed to be 512, 256, 128, 64 and 32 kbps. The popularity of the video object follow a Zipf distribution with a skew factor α of 0.47. And we use 500 CBR video clips, whose lenghs are uniformly distributed (30 sec – 12 min). The simulation lasts 400 simulation hours with 100,000 accesses that follow a Poisson distribution. The delays for fetching the first several segments of video objects from the original server are exponentially distributed (μ = 1.5) and the ratio of the access duration to the total duration of a video sequence in a partial viewing environment is randomly distributed. The cache capacity is assumed to be (0.5 ~ 0.8) * (Σ 128 kbps bitrate object size). And the default segment size is 64 kbyte. The transcoding delay for the first sufficient segments of version from *i* to *j* is determined by real measured values.

B. Experimental Results

The main performance metrics of the proxy system are the byte-hit ratio and delay saving ratio. The byte-hit ratio(BHR) is defined as the number of bytes that are delivered to the client directly from the proxy divided by the total number of bytes requested by the client. It is the major metric for evaluating the reduction of the network traffic to the server. The delay saving raito(DSR) is defined as the fraction of communication and server delays which is saved by satisfying the references from the cache instead of the server.

We compare the performance of the transcoding proxy with TeC2, VPF, BRB and proposing BRB-DQS. BRB is BitRate Based segmentation and BRB-DQS is the DQS policy with bitrate-based segmentation.

Figure 5 and Figure 6 plot the results of the evaluations of DSR and BHR under values of DQS start ratio, respectively. When the proxy always uses DQS policy, the system performance is better, but the service quality is lower. So, system administer select the suitable start point of DQS policy. As shown in Figure 5 and Figure 6, DQS with bitrate based segmentation has good performance than others.

Figure 7 and Figure 8 plot the results of the evaluations of DSR and BHR under values of Zipf parameter, respectively. The Zipf parameter determines the access pattern of the system. A larger value indicated that the accesses are biased to a certain set of popular objects. If the system has enough capacity to maintain these popular objects in the proxy, it can achieve a high hit ratio. Consequently the delay saving ratio and the byte hit ratio increase with the Zipf parameter.

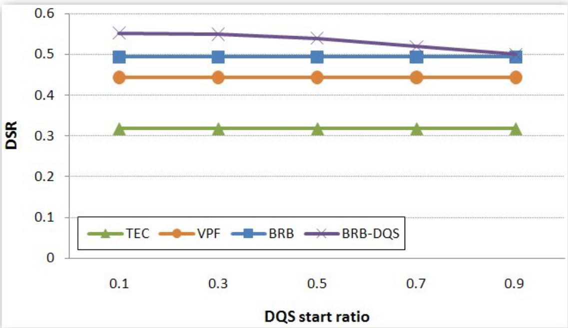

Figure 5: Delay Saving Ratio under various values of DQS start ratio

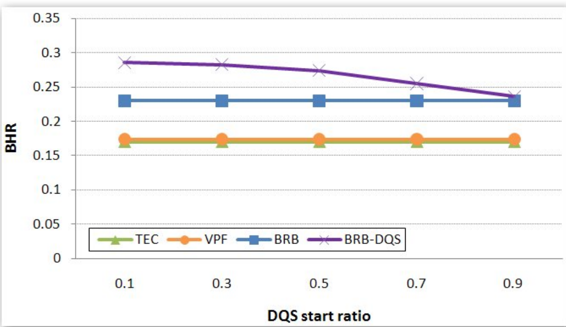

Figure 6: Byte Hit Ratio under various values of DQS start ratio

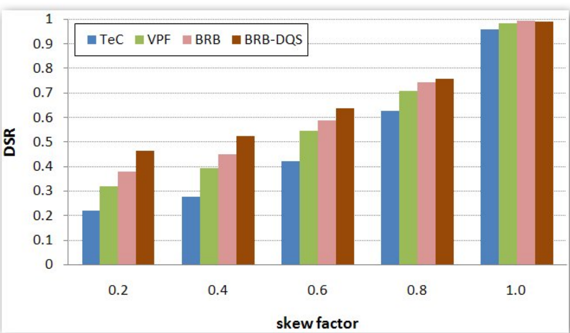

Figure 7: Delay Saving Ratio under various values of Zipf parameter

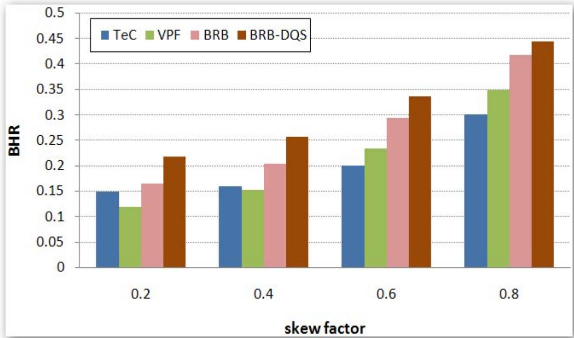

Figure 8: Byte Hit Ratio under various values of Zipf parameter

Figure 9 and Figure 10 plot the results of the evaluations of DSR and BHR under values of σ(access pattern), respectively. As elsewhere [13], accesses to different versions of the objects were assumed to follow a normal distribution with mean m, which stands for the dominant version. The access probability of a version x is

$$p(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(x-m)^2/2\sigma^2}$$

where m and σ^2 represent the mean and the variance, respectively, of the accesses of the versions. When σ is small, one version(m) tends to be preferentially accessed. As σ is large, the accesses are evenly distributed among the different versions[16].

Figure 9 and Figure 10 show that BRB and BRB-DQS have higher delay saving and byte-hit ratio, regardless of popularity. As σ increases, accesses are more heterogeneous; and each version is more evenly accessed. Consequently, the performances of all methods drop. BRB-DQS and BRB have almost same performance or BRB-DQS has better performance than BRB.

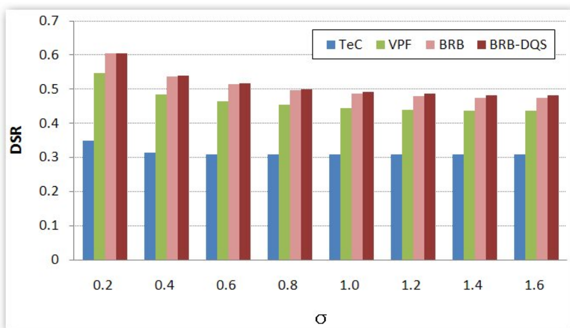

Figure 9: Delay Saving Ratio under various values of σ

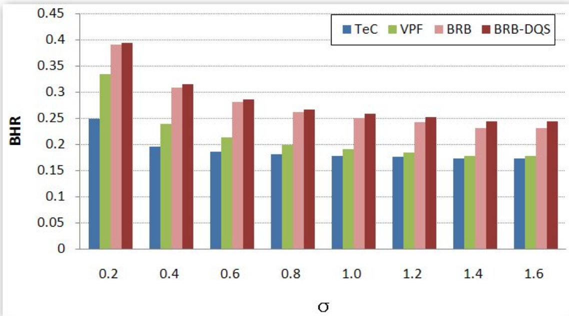

Figure 10: Byte Hit Ratio under various values of σ

IV. CONCLUSIONS

Using proxy to support media delivery is cost-effective, but challenging due to the nature of large media size and the low-latency and continuous streaming demand. And for the effective content adaptation, the transcoding proxy is attracting. The transcoding proxy is also the multimedia proxy, if the proxy system use the segmentation strategies, the performance could be better. Among the various segmentations, the bitrate-based segmentation has many benefits. And in general, video transcoding is a computation-intensive task. So, we proposed the DQS(degraded quality service) policy to reduce the CPU cost of transcoding. In this paper, we combine the previous two ideas(bitrate-based segmentation and DQS policy) and simulate the combined idea. Our combined idea has good performance such as, DSR(delay saving ratio), BHR(byte hit ratio).

ACKNOWLEDGMENT

This work was supported by Dong-eui University Research Grant(2009AA181).

REFERENCES

[1] Reza Rejaie, Haobo Yu, Mark Handley, Deborah Estrin, “Multimedia Proxy Caching Mechanism for Quality Adaptive Streaming Applications in the Internet”, Proceeding of IEEE INFOCOM, Tel-Avis, Israel, Mar. 2000
[2] Jiangchuan Liu, Jianliang Xu, “Proxy Caching for Media Streaming Over the Internet”, IEEE Communications Magazine, Aug. 2004
[3] Kuei-Chung Chang, Tien-Fu Chen, “Efficient segment-based video transcoding proxy for mobile multimedia services”, Journal of Systems Architecture, ELSEVIER, Nov. 2007
[4] Kuei-Chung Chang, Ren-Yo Wu, Tien-Fu Chen, “Efficient segment-based video transcoding proxy for mobile multimedia services”, Proceedings of International Conference of Multimedia and Expo(ICME 2005), Jul. 2005
[5] Yoohyun Park, Yongju Lee, Hagyoung Kim, Kyongsok Kim, “Efficient Segment based Streaming Media Transcoding Proxy for Various Types of Mobile Devices”, Pacific-Rim Conference on Multimedia, Hong Kong, Dec. 2007
[6] Yoohyun Park, Hagyoung Kim, Kyongsok Kime, “Segement-based Cache Replacement Policy in Transcoding Proxy”, The KIPS Transactions : Part A, Vol. 15-A, No.1, Feb. 2008
[7] Yoohyun Park, “A Degraded Quality Service Policy for reducing the transcoding loads in a Transcoding Proxy”, The KIPS Transactions : Part A, Vol. 16-A, No.3, June 2009
[8] Asit Dan, Dinkar Sitaram, “A Generalized Interval Caching Policy for Mixed Interactive and Long Video Workloads”, Proceedings of SPIE/ACM Conference on Multimedia Computing and Networking(MMCN’96), San Jose, CA, Jan. 1996
[9] Z.L.Zhang, Y.Wang, D.Du, D.su, “Video staging: A proxy-server based approach to end-to-end video delivery over wide-area networks”, IEEE/ACM Trans. Networking, 8(4), 2000
[10] S.Sen, J.Rexford, D.Towsley, “Proxy prefix caching for multimedia streams”, in Proc. IEEE INFOCOM’99, New York, NY, Mar. 1999
[11] Kun-Lung Wu, Philip S. Yu, Joel L. Wolf, “Segment-Based Proxy Caching of Multimedia Streams”, Proceedings of the 10th International Conference World Wide Web, 2001, pp. 36-44
[12] Xueyan Tang, Fan Zhang, Samuel T. Chanson, “Streaming Media Caching Algorithms for Transcoding Proxies”, Proceedings of the International Conference on Parallel Processing(ICPP02), 2002
[13] Bo Shen, Sung-Ju Lee, Sujoy Basu, “Caching Strategies in Transcoding-Enabled Proxy Systems for Streaming Media Distribution Networks”, IEEE Transactions on Multimedia, Vol.6, No.2, Apr. 2004
[14] YongJu Lee, YuHyeon Bak, OkGee Min, HagYoung Kim, CheolHoon Lee, "The PT-2 Caching Algorithm in the Transcoding Proxy Cluster to Facilitate Adaptive Content Delivery", International Workshop on Multimedia Content Analysis and Mining 2007(MCAM01), WeiHai, China, Jun. 2007
[15] Cheng-Yue Chang, Ming-Syan Chen, “On Exploring Aggregate Effect for Efficient Cache Replacement in Transcoding Proxies”, IEEE Transactions on Parallel and Distributed Systems, Vol.14, No.6, Jun. 2003
[16] Chi-Feng Kao, Chung-Nam Lee, “Aggregrate Profit-Based Caching Replacement Algorithms for Streaming Media Transcoding Proxy Systems”, IEEE Transactions on Multimedia, Vol.9, No.2, Feb. 2007
[17] Yuhyeon Bak, Kapdong Kim, Youngju Lee, Songwoo Sok, Changsoo Kim, Hagyoung Kim, Myung-Joon Kim, Kyongsok Kim, "SMART : High Performance Internet Server for High Definition Streaming Service", International Journal of Smart Home(IJSH), Vol.1, No.1, 2007
[18] Songqing Chen, Bo Shen, Susie Wee, Xiaodong Ahang, “Designs of High Quality Streaming Proxy Systems”, Proceedings of IEEE INFOCOM (INFOCOM'2004), Hong Kong, China, March 7-11, 2004
[19] Songqing Chen, Bo Shen, Susie Wee, Xiaodong Zhang, “Segment-Based Streaming Media Proxy : Modeling and Optimization”, IEEE Transactions on Multimedia, Vol.8, No.2, Apr. 2006
[20] Dongyu Liu, Songqing Chen, Bo Shen, “AMTrac:Adaptive Meta-caching for Transcoding”, Proceedings of ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 2006), Newport, Rhode Island, May 22-23, 2006
[21] Aameek Singh, Abrishek Trivedi, Krithi Ramamritham, Prashant Shenoy, “PTC:Proxies that Transcode and Cache in Heterogeneous Web Client Environments”, World Wide Web:Internat and Web Information Systems, Vol.7, No.1, Mar. 2004

Jung-hwa Lee received his B.S. and the Ph.D. degrees at the Department of Computer Science from Pusan National University, Korea, in 1995 and 2001, respectively. He is a professor at the Department of Computer Software engineering, Dongeui University in Korea. His research interests include database, Hangeul information processing, and semantic web, etc.

Yoohyun Park received the B.S. and M.S. degrees in computer science from Pusan National University, Korea, 1996 and 1998 and Ph.D. degree in computer science from Pusan National University, Korea, in 2008, respectively. During 2000, he worked at KIDA (Korea Institute for Defense Analyses) in Seoul, Korea as a researcher and from 2001 to 2009, he worked at ETRI (Electronics and Telecommunications Research Institute), Daejeon, Korea as a researcher. Now, he has joined Dong-eui University, Busan, Korea since 2009 as f faculty. His research interests include transcoding proxy, multimedia content delivery, network-storage system, database and so on.