International Journal of Innovative Computing, Information and Control
Volume **8**, Number **2**, February **2012**
ICIC International ©2012 ISSN 1349-4198
pp. 1493–1500

A PROXY CACHING SCHEME BASED ON PARTITION AND MAPPING OF MEDIA BLOCK SEGMENTS

Chong Deuk Lee[1], Taeg Won Jeong[1], Kyung Ho Choi[2]
and Jeong Yong Ahn[3,*]

[1]Division of Electronic Engineering
[3]Department of Statistics (Institute of Applied Statistics)
Chonbuk National University
567 Baekje-daero, deokjin-gu, Jeonju-si, Jeollabuk-do 561-756, Korea
cdlee1008@jbnu.ac.kr; ttwjeong@yahoo.co.kr; *Corresponding author: jyahn@jbnu.ac.kr

[2]Jeonju University
1200 Hyoja-dong, Wansan-gu, Jeonju, Korea
chk414@jj.ac.kr

Received November 2010; revised March 2011

Abstract. *In this article, we propose a new proxy caching scheme with fuzzy filtering based on partition and mapping of media block segments. In the scheme, media block segments are divided into fixed and variable partition reference blocks, and the mapping is performed by semantic relationship. The proposed scheme helps to alleviate the problems of the typical schemes.*

Keywords: Proxy caching, Fuzzy filtering, Block partition, Block mapping, Grouping

1. **Introduction.** Many studies have been performed for streaming of multimedia data in response to the recent increase in requests for wireless services and multimedia services based on distributed networks [7,12,13,15]. The typical schemes for distributed multimedia service are CMD (centralized multimedia distribution), client-server architecture, CDN (content distribution network) and P2P- (peer-to-peer) based scheme [17]. The CMD efficiently manages storage and I/O (input/output) capacity using mirroring in a web-based distributed service environment for the improvement of service functions in the central server. This scheme, however, creates a bottleneck, which greatly influences the performance of distributed service. The client-server architecture is based on batching, fetching and periodic broadcasting. The architecture is difficult to implement and is not appropriate for large-scale streaming service because of the weakness in multicast for IP network. The CDN is based on the large number of CDN servers on the Internet. For efficient service, multimedia contents should be distributed in CDN servers, which can transfer contents to neighbor clients. The scheme requires a lot of investment in the broadband transmission architecture of a network and a lot of resources for multimedia service. P2P-based network service scheme is used in Napster (www.napster.com), Gnutella (www.gnutella.wego.com), FreeNet (freenet.sourceforge.net) and CenterSpan (www. centerspan.com). A peer shares data of neighbor peer group and services data found by the query to neighbor peers or the directory server. This scheme, however, has the problems of service quality, data bandwidth and slow mobility even though it enables the peers to share data of many communities with low cost [1,5,18]. Table 1 lists the features of the typical schemes.

In this paper, we propose a proxy caching scheme with fuzzy filtering based on partition and mapping of media block segments. First, the partition is a procedure to reduce jitter delay and improve service response rates. Unpartitioned large media blocks cause start

Table 1. The features of existing schemes

Schemes	Approach	Drawback
CMD	server-mirroring	network bottleneck
Client-server	batching, fetching, and periodic broadcasting	unavailability of multicast
CDN	a technique based on the number of servers	need much resources
P2P	a technique using data sharing of neighbor peers	low data transfer rate

delay. To solve this problem, we divide media block segments into a fixed partition reference block ($FPRB$, R_fP) and a variable partition reference block ($FPRB$, R_vP). Second, segment mapping is a strategy for speedy cache replacement. In this study, mapping is performed by semantic relationship of the blocks. Finally, the grouping is a procedure to reduce replacement time in the proxy and reply to the request of clients quickly. The features and main contributions of this study are as follows. First, the proposed scheme uses a fuzzy filtering method to improve the quality of streaming service of multimedia data. Second, the scheme does not stream all block data, but only the requested block data. Third, the scheme improves the performance of response time and caching mechanism compared with those of other proxy caching schemes.

In the first part of this article, we describe the streaming procedures of the proposed scheme. In part two, we present the simulation results to compare the performance of the proposed scheme with four other proxy caching schemes.

2. A Scheme for Streaming Service.

A scheme for streaming service should minimize the response time, network bandwidth and the traffic for user-oriented services. The scheme proposed in this study does not stream all block data, but only the requested block data. The streaming is performed by block partition, mapping and merging for frequently referred blocks of data in service layer architecture.

2.1. Block partition.

Block partition is a procedure to divide blocks into fixed blocks and variable blocks. In the procedure, we refer to many factors, such as network bandwidth, traffic and time delay [2,4]. Block partition is determined by the encoding rate and network bandwidth to minimize the proxy jitter for fetched segments. In this study, a partitioned segment of the media blocks, P, is given by Equation (1). The relation for encoding rate and bandwidth is given by Equation (2) to minimize the proxy jitter. The buffer size for the determined total block partition is given by Equation (3).

$$\sum_{i=1}^{n} SL_i - \frac{SL_{n+1} \times (E_r - B_w)}{B_w}, \tag{1}$$

$$\min_{n} \left\{ \sum_{i=1}^{n} SL_i - \frac{SL_{n+1} \times (E_r - B_w)}{B_w} \right\} > 0, \tag{2}$$

$$\frac{SL_{n+1} \times (E_r - B_w)}{E_r}, \tag{3}$$

where SL_i is the length of the i^{th} segment of the object, n is the number of segments saved in the cache, SL_{n+1} is the waiting segment for caching, E_r is the average encoding rate for a certain object segment, and B_w is the average network bandwidth for the proxy-server. The jitter is determined by performing $FPRB$ and $VPRB$ for partitioned blocks.

2.1.1. *Fixed partition reference block.* $FPRB$ is an architecture with the same size of block segments. In $FPRB$, encoding rate and bandwidth should be considered. $FPRB$ R_fP is as follows:

Definition 2.1. *$R_fP = (n+1)SL_i \times T_f - SL_i \times \frac{E_r}{B_w}$, where T_f is the time synchronization for fixed partition multimedia streaming in a distributed environment.*

The buffer size $R_fP(B_{uffer})$ for fixed partition reference block R_fP is:

Definition 2.2. *$R_fP(B_{uffer}) = SL_1 \times T_f \times \frac{E_r - B_w}{E_r}$.*

2.1.2. *Variable partition reference block.* $VPRB$ is the architecture with different sizes of block segments. $VPRB$ R_vP is as follows:

Definition 2.3. *$R_vP = (n+1)SL_i \times T_v - SL_i \times \frac{E_r}{B_w}$, where T_v is the time synchronization for variable partition multimedia streaming in a distributed environment.*

The buffer size $R_vP(B_{uffer})$ for variable partition reference block R_vP is:

Definition 2.4. *$R_vP(B_{uffer}) = SL_1 \times T_v \times \frac{E_r - B_w}{E_r}$.*

If $R_fP = 0$ or $R_vP = 0$, then the segment is not partitioned. In this case, start delay times due to non-partition produces proxy jitter. To avoid proxy jitter and delay time, $[E_r/B_w]^{\text{th}}$ segment block is partitioned. This is a process to search segments with $R_fP \neq 0$ and $R_vP \neq 0$.

2.2. **Mapping.** Block mapping is a procedure to classify block segments as referred segments and unreferred segments [9]. In this study, the mapping is performed by semantic relationship. In other words, if a semantic relationship is satisfied, then we will perform the mapping by R_fP and R_vP.

2.2.1. *Semantic relationship.* Mapping by semantic relationship is a procedure to aggregate block segments of high semantic relationship [10]. This mapping is a triple (R, P, FR), where R is the semantic relationship for partitioned segment, P, and FR is the fuzziness to show the degree of relationship between P. The relation R for mapping is performed by the fixed partition time synchronization (FPTS, T_f) and the variable partition time synchronization (VPTS, T_v) for streaming.

(i) FPTS: T_f

T_f is a R_fP that maintains time synchronization relationship $(R_fP_i, R_vP_i.T_f)$ between P, where P_i is i^{th} partitioned segment. The semantic relation between super-layer and sub-layer is defined as follows:

Definition 2.5. *If $P = T_f(R_fP_1, \ldots, R_fP_n) \Longleftrightarrow$ member-of $\{(R_fP_1, R_fP_1.T_f) \wedge (R_fP_2, R_fP_2.T_f), \ldots, \wedge(R_fP_n, R_fP_n.T_f)\}$, then $P = \{((R_fP_i, R_fP_i.T_f), \cup(R_fP_i, R_fP_i.T_f\}$.*

A media block is composed of a number of contiguous streaming media segments, and each media block is cached independently using a prefix caching allocation, mapping and grouping process.

Figure 1 shows a media object that has five media blocks. The media blocks are partitioned into segments according to T_f and T_v. The partitioned segments preserve semantic relations according to the relevance degree. Figure 2 is an example of the synchronization. If a media block is synchronized with T_f and T_v as Figure 2 and the semantic relations are preserved, the mapping by T_f is represented as the following:

$SynchroVideoFrame_i = T_f(videoFrame_i \cdot \tau,\ vtext_i \cdot \tau, text_i \cdot \tau, scene_i \cdot \tau) \Longleftrightarrow$
member-of $\{(Synchroscene_1 \cdot vtext_1 \cdot T_f, trtext_1 \cdot T_f, scene_1 \cdot T_f)\} \wedge$
member-of $\{(Synchroscene_3 \cdot vtext_3 \cdot T_f, trtext_3 \cdot T_f, scene_3 \cdot T_f)\} = 1.T_f \cup 3.T_f$,
where τ is a temporal synchronization.

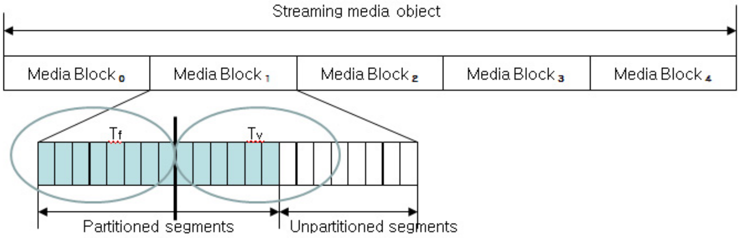

Figure 1. Partition of media blocks

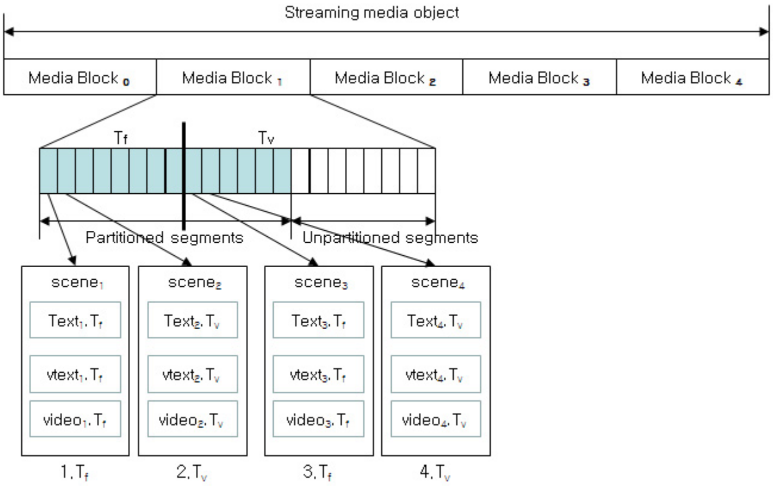

Figure 2. Synchronization by T_f and T_v

(ii) VPTS: T_v

T_v is a R_vP that maintains time synchronization relationship $(R_fP_i, R_vP_i.T_v)$ between P. Likewise to T_f, the semantic relation is defined as follows:

Definition 2.6. *If* $P = T_v(R_vP_1, \ldots, R_vP_n) \Longleftrightarrow$ *member-of* $\{(R_vP_1, R_vP_1.T_vf) \wedge (R_vP_2, R_vP_2.T_v), \ldots, \wedge(R_vP_n, R_vP_n.T_v)\}$, *then* $P = \{((R_vP_i, R_vP_i.T_v), \cup(R_vP_i, R_vP_i.T_v\}$.

As in T_f, the mapping by T_v is represented as the following:
$SynchroVideoFrame_i = T_v(videoFrame_i \cdot \tau,\, vtext_i \cdot \tau, text_i \cdot \tau, scene_i \cdot \tau) \Longleftrightarrow$
member-of $\{(Synchroscene_2 \cdot vtext_2 \cdot T_v, trtext_1 \cdot T_v, scene_2 \cdot T_v)\} \wedge$
member-of $\{(Synchroscene_4 \cdot vtext_4 \cdot T_v, trtext_4 \cdot T_v, scene_4 \cdot T_v)\} = 2.T_v \cup 4.T_v$.

2.2.2. *Mappings by* R_fP *and* R_vP. Mapping by R_fP is a technique to map semantically related R_fP for partitioned P and to filter caching R_fP for streaming. Filtered R_fP reduces time delay and jitter, and efficiently manages bandwidth in streaming. The relation between fuzzy filtering $\mu - cut_{R_fP}$ and the mapping $M_{R_fP}^{filtering}$ by R_fP is as follows:

Definition 2.7. $\mu - cut_{R_fP} = \left\{M_{R_fP}^{filtering}(P) \mid ((R_fP_i, R_fP_i.T_f), \cup R_fP_i) \geq \mu\right\}$, *where* $((R_fP_i,\ R_fP_i.T_f), \cup R_fP_i)$ *is the semantic relation between super-layer and sub-layer,* $(R_fP_i, R_fP_i.T_f)$ *is the time synchronization relationship between* P*, and* μ *is a fuzzy value between* 0 *and* 1 *for fuzzy filtering in the mapping. For the semantic relationship of* $M_{R_fP}^{filtering}$ *the following condition must be satisfied:*

Condition 1: $M_{R_fP}^{filtering}(P, R_fP) \geq \mu - cut$ for ancestors or neighbours($a.o.n.$) of R_fP.

The relation between fuzzy filtering $\mu - cut_{R_vP}$ and the mapping $M_{R_vP}^{filtering}$ by R_vP, and the condition are defined similarly to Definition 2.7 and Condition 1.

Definition 2.8. $\mu - cut_{R_vP} = \left\{ M_{R_vP}^{filtering}(P) \mid ((R_vP_i, R_vP_i.T_v), \cup R_vP_i) \geq \mu \right\}$.

Condition 2: $M_{R_vP}^{filtering}(P, R_vP) \geq \mu - cut$ for *a.o.n.* of R_vP.

Condition 1 and Condition 2 are the computing procedures of fuzzy filtering for the semantic relation in Definition 2.7 and Definition 2.8. In Condition 1, $M_{R_fP}^{filtering}(P, R_fP) \geq \mu - cut$ is used for the selection of media objects satisfying the proxy cache conditions. For example, P is a segment and the fuzzy filtering relation by R_fP and R_vP is as shown in Table 2. Then the mapping by $M_{R_fP}^{filtering}(P) \geq 0.6 - cut$ is $R_fP = \{x_1, x_2, x_4, x_6, x_8\}$, and the mapping by $M_{R_vP}^{filtering}(P) \geq 0.6 - cut$ is $R_vP = \{x_2, x_3, x_4, x_8\}$.

Table 2. Fuzzy filtering relation by R_fP and R_vP

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
R_fP	0.72	0.93	0.48	0.86	0.53	0.78	0.25	0.82
R_vP	0.18	0.70	0.83	0.90	0.37	0.03	0.21	0.65

2.3. **Grouping.** The grouping of related R_fP and R_vP, performed by applying $\mu - cut$ for each of partitioned Ps, reduces the number of caching for streaming. In this study, Ps are grouped according to the results of fuzzy filtering for R_fPs and R_vPs referred by the mapping.

3. Performance Analysis.

3.1. **Simulation model.** We evaluated the performance of the proposed scheme by simulation using the parameters shown in Table 3. Objects between a proxy and the original server are selected at random and simulated for access streaming of object segments. The total number of streaming access is set to $14,000, 0 \leq \lambda \leq 5$ for $ASRR$ (Average Service Response Rate) and $0 \leq \theta \leq 0.35$. $ALTR$ (Average Latency Time Rate) and $ACHR$ (Average Cache Hit Rate) are simulated for $0.5 - cut \leq R_vP \leq 0.9 - cut$ when B_w is $10/100Mbps$ and $0.5 \leq \mu - cut \leq 0.9$ for $\mu - cut$. We evaluated only for R_vP since the performance of R_vP is better than that of R_fP.

3.1.1. *ASRR.* The average service response time is the time between a request and the corresponding response. The block segments partitioned for $ASRR$ have the probabilities $p_i = f_i / \sum f_i$, where $f_i = 1/i^\theta$ $(i = 1, 2, \ldots, n)$, θ is the skew factor $(\theta > 0)$, and n is the number of partitioned segments of the object. $ASRR$ indicates the average service response time rate of the clients is

$$ASRR = \lambda \times \left(\frac{1}{i^\theta} \Big/ \sum \frac{1}{i^\theta} \right).$$

3.1.2. *ALTR.* The average latency time rate is a delay time in streaming service request, which is not pre-fetched timely to the clients. The latency time rate depends on the size of the cache, the encoding rate, E_r, of the segment of the objects. $ALTR$ indicates the delay time, which is not pre-fetched timely to the clients is

$$ALTR = 1 - \frac{\sum_{i=M+1}^{n} \lambda_i \times P \times R_vP \times \alpha \times \beta \times (E_r - B_w)}{\sum_{i=1}^{n} \lambda_i \times P \times B_w}.$$

Table 3. Simulation parameters

Parameters	Meaning
n	The number of partitioned segments of the object.
p_i	The probability for the i^{th} segment of the object.
θ	The skew factor for the partitioned segments.
μ	Fuzzy value of the i^{th} segment of the object.
λ	Access arrival rate for Poission distribution.
T_{cache}	The size of total cache.
$F_{cache-1}$	The size of reserved storage to cache the first segment of the object.
$R_{rest-cache}$	The size of reserved storage to cache the rest segment of the object.
α	The capacity to cache R_vP.
β	The size of reserved storage to cache R_vP.

3.1.3. *ACHR*. Cache hit is the procedure to reduce $ASRR$ by streaming of relevant caching data in grouping architecture when clients request streaming service in proxy caching grouping. The grouping architecture performs streaming by the transcoding of reserved storage to cache the first segment of the object. $ACHR$ indicates the cache hit rate to hit the segment of objects in the grouping architecture is

$$ACHR = 1 - \frac{\sum_{i=M+1}^{n} R_vP \times SL_i \times F_{cache-1} \times \alpha \times \beta \times (E_r - B_w)}{\sum_{i=1}^{n} P \times T_{cache} \times B_w}.$$

3.2. **Simulation results.** In this section, we present a result of the simulation study to compare the performance of the proposed scheme with other schemes, such as the segment-based proxy caching scheme for the streaming of media objects [3,16], the prefix caching scheme for the segment of media objects in the first stage [11,14], the pre-fetching scheme for the minimization of proxy jitter by fetching uncached segments before the access of segments [4,6], and the network overlay scheme for the clustering of application groups of neighbor hosts and peers in network architecture [8].

Objects between a proxy and the server are selected randomly and simulated for access streaming of block segments. For grouping architecture, we divided filtered block segments into four groups. Each group is simulated five times to get the average service response rate, the average latency time rate, and the average cache hit rate. The encoding rate of the cached block segments is set from $128Kbps$ to $256Kbps$, and the network bandwidth of uncached block segments is selected at random as a number between $0.5Mbps$ and $2Mbps$.

Figure 3 shows the proposed scheme provides service more efficiently in respect to the ARSS. The ARSSs of the proposed scheme have about 0.85. It shows the scheme is relatively more efficient than those of the other schemes, while it is similar to the ARSS of the pre-fetching scheme. In addition, the ARSS of the proposed scheme is not affected by the number of blocks, while those of the other schemes are affected. Figure 4 shows the result on the ALTR. The proposed scheme, as shown in Figure 4, has a better ALTR than any other schemes. It decreases as the $\mu - cut$ increases. Figure 5 shows the proposed scheme achieves a relatively high ACHR, which indicates a smaller reduction of network traffic. This is the price to pay for less proxy jitter and the smaller delayed startup time as shown in Figure 3 and Figure 4. As a result, the proposed scheme has better performance by using the priority determined by fuzzy filtering.

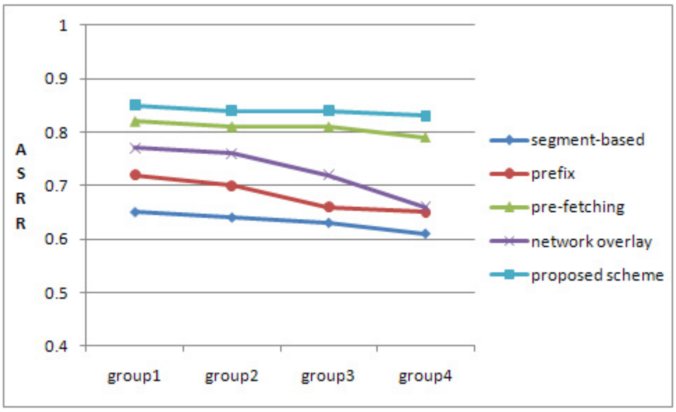

FIGURE 3. Average service response rate

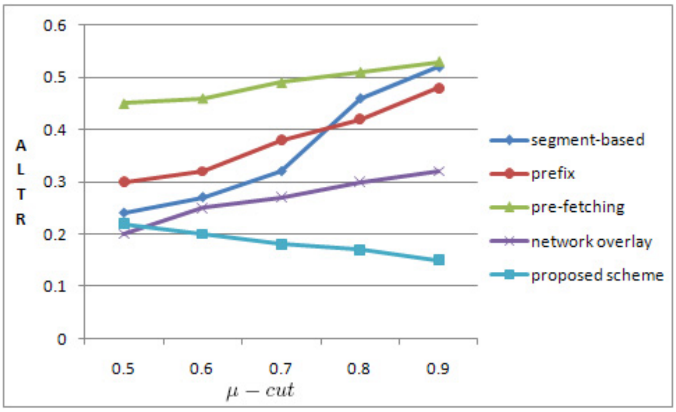

FIGURE 4. Average latency time rate

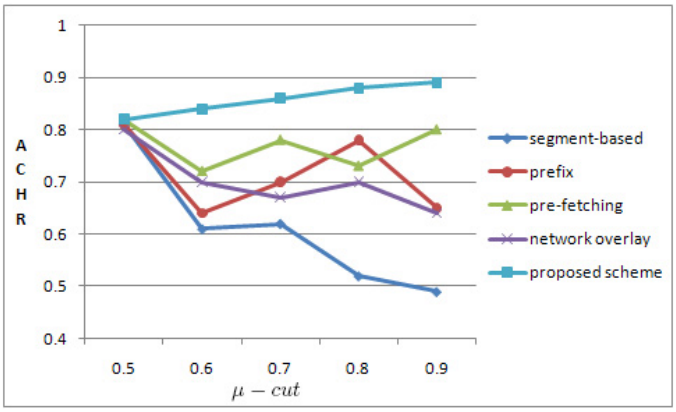

FIGURE 5. Average cache hit rate

4. **Conclusion.** For multimedia service in various distributed environments, many schemes have been proposed. However, most schemes have problems, such as network bandwidth, jitter, low service response rate and timeliness. In this paper, we propose a proxy caching scheme with fuzzy filtering based on partition and mapping of media block segments to improve the problems. The scheme divides media block segments into a fixed partition reference block and a variable partition reference block. The simulation results show the proposed scheme has better performance than the other schemes. Future research is needed to reduce system overhead in various distributed network traffic environments.

REFERENCES

[1] S. Chen, B. Shen, S. Wee and X. Zhang, Adaptive and lazy segmentation based proxy caching for streaming media delivery, *Proc. of the International Workshop on Network and Operating Systems Support for Digital Audio and Video*, pp.429-441, 2003.

[2] S. Chen, B. Shen, S. Wee and X. Zhang, Investigating performance insights of segment-based proxy caching of streaming media strategies, *Proc. of ACM International Conference on Multimedia Computing and Networking*, pp.148-165, 2003.

[3] S. Chen, B. Shen, S. Wee and X. Zhang, Segment-based streaming media proxy: Modeling and optimization, *IEEE Transactions on Multimedia*, vol.8, pp.243-256, 2006.

[4] B. R. Dai, J. W. Huang, M. Y. Yeh and S. Chen, Adaptive clustering for multiple evolving streams, *IEEE Transactions on Knowledge and Data Engineering*, vol.18, pp.1166-1180, 2006.

[5] S. Guha, A. Meyerson, N. Mishra, R. Motwani and L. O'Callaghan, Clustering data sreams: Theory and practice, *IEEE Transactions on Knowledge and Data Engineering*, vol.15, pp.515-519, 2003.

[6] C. M. Huang and T. H. Hsu, A user-aware prefetching mechanism for video streaming, *World Wide Web*, vol.6, pp.353-374, 2003.

[7] D. H. Kim, Packet scheduling algorithm for realistic traffic model of real-time video streaming service in OFDMA system with integrated traffic scenario, *International Journal of Innovative Computing, Information and Control*, vol.6, no.11, pp.4797-4812, 2010.

[8] E. Kusmierek, Y. Dong and H. C. Du, Loopback: Exploiting collaborative caches for large-scale streaming, *IEEE Transactions on Multimedia*, vol.8, pp.233-242, 2006.

[9] C. D. Lee and J. Y. Ahn, A semantic-based post-office box structure for user-centered multimedia services, *Journal of Fuzzy Logic and Intelligent Systems*, vol.16, pp.402-409, 2006.

[10] E. Megalou and T. Hadzilacos, Semantic abstractions in the multimedia domain, *IEEE Transactions on Knowledge and Data Engineering*, vol.15, pp.136-160, 2003.

[11] R. Rejaie, M. H. Yu and D. Estrin, Multimedia proxy caching mechanism for quality adaptive streaming application in the Internet, *Proc. of IEEE Computer and Communications Societies*, vol.2, pp.980-989, 2000.

[12] J. Ren and C. Huo, Load shedding for windowed non-equijoin over sensor data streams, *International Journal of Innovative Computing, Information and Control*, vol.5, no.5, pp.1265-1273, 2009.

[13] M. Rezaei, I. Bouazizi and M. Gabbouj, Fuzzy joint encoding and statistical multiplexing of multiple video sources with independent quality of services for streaming over DVB-H, *International Journal of Innovative Computing, Information and Control*, vol.5, no.7, pp.1837-1850, 2009.

[14] S. Sen, J. Rexford and D. Towsley, Proxy prefix caching for multimedia streams, *Proc. of the IEEE Computer and Communications Societies*, vol.3, pp.1310-1319, 1999.

[15] Z. Su, J. Katto and Y. Yasuda, Robust algorithm to retrieve scalable streaming media over content delivery networks, *International Journal of Innovative Computing, Information and Control*, vol.3, no.6(B), pp.1743-1754, 2007.

[16] K. Wu, P. S. Yu and J. Wolf, Segment-based poxy caching of multimedia streams, *Proc. of the Conference on WWW*, pp.36-44, 2001.

[17] Z. Xiang, Q. Zhang, W. Zhu, Z. Zhang and Y. Zhang, Peer-to-peer based multimedia distribution service, *IEEE Transactions on Multimedia*, vol.6, pp.343-355, 2004.

[18] P. Zhu, W. Zeng and C. Li, Joint design of source control and QoS-aware congestion control for video streaming over the Internet, *IEEE Transactions on Multimedia*, vol.9, pp.366-376, 2007.