International Journal of Innovative Computing, Information and Control
Volume 9, Number 1, January 2013
ICIC International ©2013 ISSN 1349-4198
pp. 377–386

AN OBJECT VERSION TRANSCODING FOR STREAMING MEDIA SERVICES IN WIRELESS NETWORKS

Taeg Won Jeong[1], Jeong Yong Ahn[2] and Chong Deuk Lee[1,*]

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

Received October 2011; revised February 2012

Abstract. *Transcoding has a significant effect on streaming media services in wireless networks. The goal of this paper is to reduce network traffic due to high network bandwidth demand and streaming constraints, and enhance the performance of streaming media services. Therefore, we propose a new OVT (object version transcoding) mechanism based on fuzzy similarity. This work derives a media object relationship to calculate the individual fuzzy similarity of certain versions of a media object. We take into account transcoding constraints such as media object characteristics, cache capacity and the properties of the object versions. Based on the given constraints, the proposed mechanism reduces the startup delay and congestion, and it increases the response rate and cache hit rate. We obtained excellent results in terms of QoS (Quality of service) metrics, average service response ratio, startup latency and cache-hit ratio.*

Keywords: Transcoding, Streaming, Fuzzy similarity, Object version

1. **Introduction.** For achieving streaming media services in wireless networks, streaming service environments such as media servers, cache structures, wireless LANs, and clients should be considered. In contrast to TCP, the main focus is to reduce end-to-end congestions that lead to a higher average delay and jitter. Instead, the protocol provides the means for the timing of media objects, and a sequence number mechanism to detect packet loss. In order to solve this problem, several QoS aware network mechanisms, such as a behavior prediction method based on wireless signals, a pre-caching strategy based on location, and a temp-storing strategy based on capability, have been proposed to help wireless clients rapidly restore the topology and data forwarding [1-3]. However, unlike traditional streaming services, streaming service in wireless networks cannot provide service guarantees for inter-domain moving mobile devices. In particular, streaming media requires more real-time service guarantees compared with that required by file downloading or file sharing. When a streaming media service is implemented in a wireless network environment, it affects the quality of streaming due to the problems of traffic, limited resources, and bandwidth constraints. Transcoding is a means to overcome these constraints.

Transcoding is the procedure used to change a coded signal into a signal of another code. In the streaming media service environment, transcoding is a technique that transforms compressed moving pictures into another specification of moving pictures or adjusts the bit rate, resolution, and error resilience [4,6]. The typical transcoding mechanism performs caching in addition to web proxy functions. The conventional web proxy, however, has the problem of deteriorating QoS in the process of streaming media objects due to delay, congestion, and interference [6-9]. To overcome this limitation, Cardelli et al. [10]

proposed a coverage-based algorithm and a demand-based algorithm. The coverage-based algorithm caches the original version of objects, while the demand-based algorithm caches the transcoded version. These algorithms, however, have congestion and delay problems, since they do not consider the size of media objects, cache capacity of the proxy, similarity of media objects, segment version or sizes of media objects, bit rate, or frame rate [11,12]. Hefeeda et al. [5] proposed a media streaming mechanism that leverages underlying peer-to-peer streaming support. This mechanism relies on the underlying topology of sender candidates and network connection qualities to infer goodness of the peers to choose the best media streaming. However, the above mechanism is not specifically considered for transcoding for media objects and none of above mechanisms considers relationship about which media objects are frequently related and referenced.

This paper presents a new object version transcoding method based on fuzzy similarity to solve the problems of existing methods and enhance the quality of wireless streaming services. The proposed method applies fuzzy similarity for partitioned versions of streaming media objects. The fuzzy similarity specifies the relationships between the partitioned versions of media objects.

Thus, the proposed method aims to provide a ceaseless streaming media service by minimizing the QoS problem in wireless networks, and to reduce network traffic due to the high network bandwidth demand and streaming constraints. The rest of this paper is organized as follows. In Section 2, we talk about some related works. The object version transcoding models are introduced in Section 3. In Section 4, we present some simulation results. Finally, the conclusions are drawn in Section 5.

2. **Related Works.** Typical media transcoding mechanisms neither consider streaming by partitioning segment versions of media objects nor perform services by integrating various versions of media objects [2,4,6]. Chang et al. [4] proposed a transcoding mechanism using ORG (Object Relation Graph). This mechanism is efficient for segmenting web objects. For media objects, however, this mechanism has restrictions due to the large size. Also, this mechanism has the problem of difficult proxy adaptation for clients with different environments. Miao and Ortega [13] proposed a selective caching mechanism to improve playback quality. This is an intermediate frame selection mechanism that caches frames according to the transmission rate. This mechanism, however, has the problem that it can not cache frames directly after the initial segment.

Cardelli et al. [10] proposed a hierarchical caching structure for the media transcoding. This mechanism distributes transcoding objects on hierarchical paths. This mechanism is efficient in reducing the proxy load, while it has the problem of high transcoding cost due to network latency.

Kao et al. [15] proposed a profit-based function to calculate general object gain by using a weighted transcoding graph. This mechanism performs transcoding by using a weighting factor in streaming, and performs cache replacement based on the gain function. This mechanism, however, requires information about which objects are frequently used.

Shen et al. [14] proposed three caching strategies for TEC (Transcoding-Enabled Caching). This mechanism assumes that there are variants of the same video on the network. The first two algorithms of this mechanism select one of the video objects and caches. This mechanism handles with different bit rates according to user requests. The third algorithm caches multi-versions of the same video objects to reduce the processing load for transcoding.

These algorithms, however, have ambiguity in transcoding relations. Thus, in the point of streaming media services, we find that the existing mechanisms are not well

TABLE 1. The comparison of existing mechanisms

Approach used	Drawbacks
A selective caching mechanism [13]	Does not cache frames directly after the initial segment.
A hierarchical caching mechanism [10]	The problem of high transcoding cost due to network latency.
ORG (Object Relation Graph) [4]	The problem of proxy adaptation for clients with different environments.
A weighted transcoding graph [15]	Does not support frequency metrics of objects.
TEC (Transcoding-Enabled caching) [14]	Incorrectness in transcoding relations.
A segment-based buffer management mechanism [3]	Should know in advance the segment information of media objects.

appropriated for supporting the service performance according to the transcoding. Table 1 shows the comparison of the existing mechanisms.

3. The Proposed Object Version Transcoding Model.

3.1. Fuzzy similarity. Fuzzy similarity can be used to analyze the relationships between object versions. For the relationships between object versions based on fuzzy similarity, we present the Fuzzy Similarity Graph (FSG), which is used to specify transcoding in more detail. As shown in Figure 1, FSG G_i is a directed graph with membership function μ [16] with a value in the range $[0, 1]$.

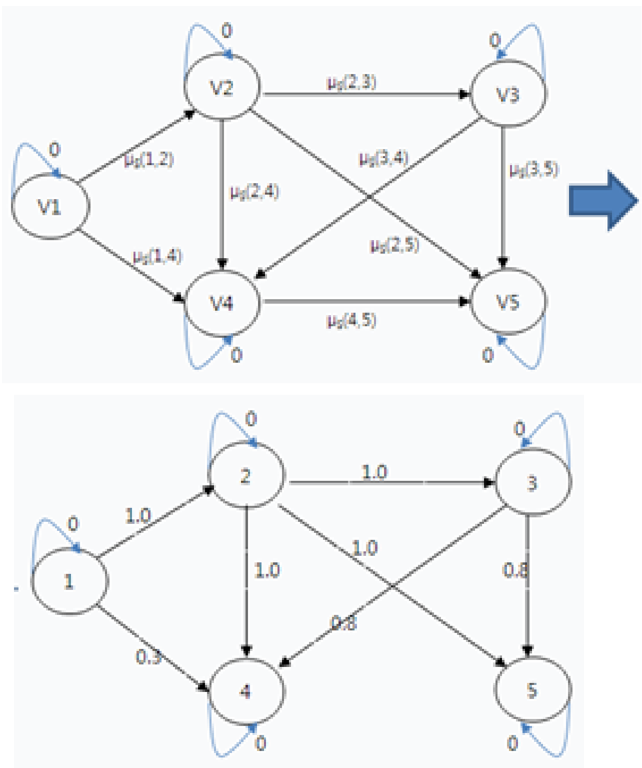
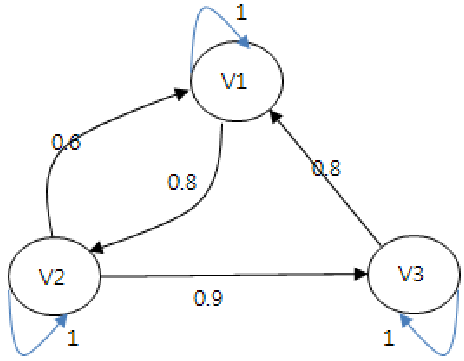


FIGURE 1. Fuzzy similarity graph with respect to membership function μ

In the graph, G_i denotes the fuzzy similarity between the versions of object segments i. For each vertex $v \in V[G_i]$, v is the version of transcodable object i. If there exists a directed edge $(u, v) \in E[G_i]$, then version u for the object segment i is transcoded into version v. The cost for the transcoding of version u to version v, denoted by $\mu_R : u \times v \to [0, 1]$, is the fuzzy similarity from version u to version v.

In FSG, the fuzzy similarities for edges from u to l and from v to l are denoted by $\mu_R(u, l) = 0.8$, $\mu_R(l, v) = 0.4$, respectively. Thus, the fuzzy similarity via the vertex l is $\mu_R(u, l) \wedge \mu_R(l, v) = 0.8 \wedge 0.4 = 0.4$. The fuzzy similarity from u to v is the maximum value of similarities for direct or indirect paths from u to v. For example, if media block objects o_1 and o_2 have partitioned object versions $o_1 = o_2 = \{v_1, v_2, v_3\}$ and a fuzzy relationship is given by a fuzzy matrix M_R, then the FSG for M_R may be represented by Figure 2.



$$M_R = \begin{pmatrix} & V_1 & V_2 & V_3 \\ V_1 & 1 & 0.8 & 0 \\ V_2 & 0.6 & 1 & 0.9 \\ V_3 & 0.8 & 0 & 1 \end{pmatrix}$$

FIGURE 2. Fuzzy relationship for fuzzy matrix M_R

In Figure 2, $\mu_R(u, v)$ denotes the edge connecting two vertices, (u, v). The fuzzy similarity and α-level [16] are applied in FSG. As an example, if the fuzzy matrix for the partitioned object segment version $P = \{v_1, v_2, v_3\}$ is given by

$$M_R = \begin{bmatrix} 1 & 0.9 & 0.6 \\ 0 & 0.6 & 0 \\ 0.9 & 1 & 0.4 \end{bmatrix},$$

Then the application of α-level to M_R and FSG is depicted by Figure 3.

As shown in Figure 3, the depicted fuzzy similarities are $v_1 \to v_2$, $v_1 \to v_3$, $v_3 \to v_1$, $v_3 \to v_2$ for $M_{R,0.6}$, and $v_1 \to v_2$, $v_3 \to v_1$, $v_3 \to v_2$ for $M_{R,0.9}$. Thus, the reference priorities for the object versions are determined by the fuzzy similarity.

3.2. **Version transcoding.** Object version transcoding is used to optimize the streaming media service by transcoding media objects that exceed the capacity of the proxy cache. In general, the size of media block objects is larger than that of the proxy cache. Thus, transcoding is performed to minimize the cache delay and response delay due to the large media block size. Since the object versions are not given priority, the streaming request for these objects causes a longer delay. To solve this problem, this paper performs transcoding of object versions. In this paper, the fuzzy similarity relation between object versions is defined as follows.

Definition 3.1. *A fuzzy similarity relation on X is the fuzzy relation satisfying the following properties.*

1) Reflexive Relation: $\mu_R(x, x) = 1, \forall_x \in X$

2) Symmetric Relation: $\mu_R(x, y) = \mu_R(y, x)$

3) Transitive Relation: $\mu_R(x, z) = \bigcup_y \{\mu_R(x, y) \cap \mu_R(y, z)\}$

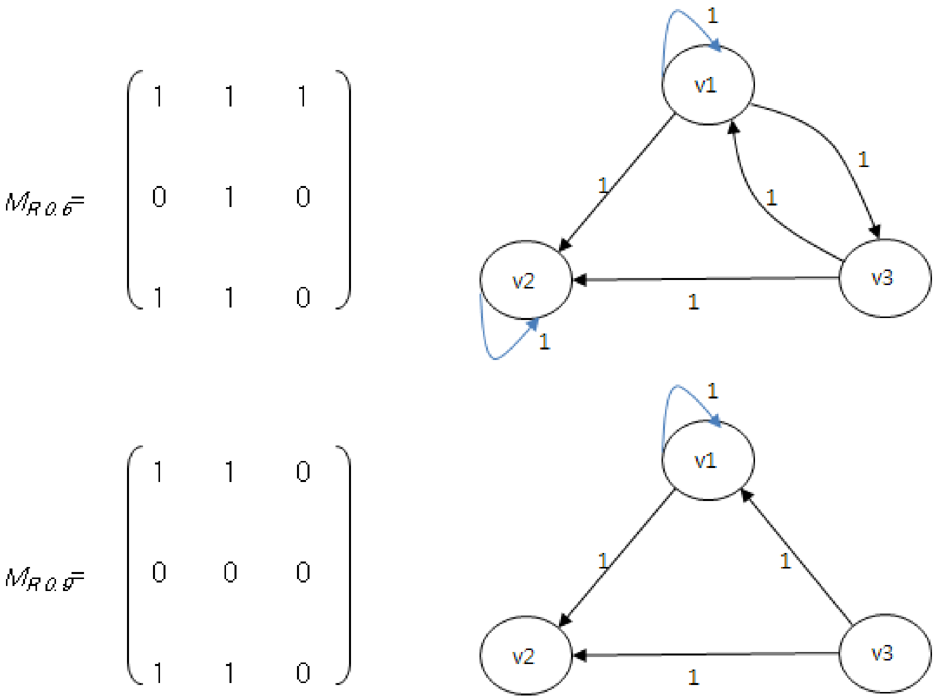


FIGURE 3. Relationship for M_R and FSG

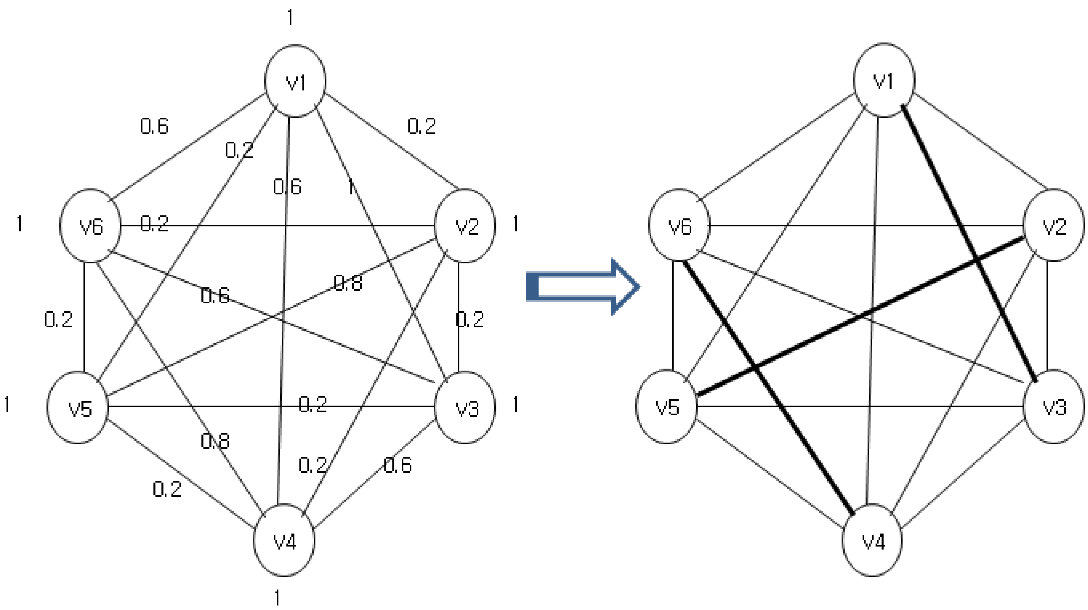


FIGURE 4. Fuzzy relationship

$A_\alpha = \{x \,|\mu_A(x) \geq \alpha\}$ is applied to derive the equivalence relation of R_α for fuzzy set A. Hence, a smaller α value means a lower similarity, and a larger α value means a higher similarity. Media objects are arranged according to the similarity, and transcoding priorities are determined.

As an example, consider the fuzzy relationship shown in Figure 4.

Figure 4 shows that the similarity relation for $\alpha \geq 0.2$ is $\{v_1, v_2, v_3, v_4, v_5, v_6\}$, which means a very low similarity transcoding relation. If α is larger than or equal to 0.8, then the similarity relation is $\{v_1, v_3\}$, $\{v_2, v_5\}$, and $\{v_4, v_6\}$. A low similarity relation indicates large media object sizes or the presence of ambiguity among media objects. Thus, the transcoding becomes clearer according to the increase of α-level or the similarity from $\alpha \geq 0.6$ to $\alpha \geq 0.8$ and $\alpha \geq 1$.

3.3. **Streaming control.** This section solves streaming problem by presenting a mapping method. Streaming control is for an efficient media service that is restricted by mapping. Thus, in this paper, we restricted the streaming via disjunction mapping ($\cup$), conjunction mapping ($\cap$), and filtering mapping. For restrictions C and $c \in C$, the mapping for streaming control is classified as the follows.

3.3.1. *Disjunction mapping.* The disjunction ($\cup$) mapping policy replaces objects with M-in-Max relations considering the cache capacity and the size of object versions. $M^{C,C}_{disjunction}$ and $M^{s}_{disjunction}$ denote the replacement mapping that considers the cache capacity and the size of object versions, respectively. Thus, disjunction ($\cup$) mapping is defined as the following.

Definition 3.2. $M^{(C,C)\cup s}_{disjunction} : \forall c,\ c' \in C$, *$c$ is $c \cup c'$. C is cache capacity, and s is the size of object versions.*

3.3.2. *Conjunction mapping.* Conjunction ($\cap$) mapping, which considers the capacity of the cache, the size of the object versions, and the Fuzzy Similarity Relationship (FSR), is the policy used to replace objects with Min-Max relationships. $M^{C,C}_{disjunction}$, $M^{s}_{disjunction}$, and $M^{FSR}_{conjunction}$ represent conjunction mappings considering the capacity of the cache, the size of the object versions, and FSR, respectively. Thus, conjunction mapping is defined as follows.

Definition 3.3. $M^{(C,C)\cap s \cap FSR}_{conjunction} : c \cap c' \in C,\ \forall c, c' \in C$ *where* $\forall c, c' \in C$, *Ø is null.*

3.3.3. *Filtering mapping.* Filtering mapping is the policy used to delete objects that do not satisfy disjunction mapping ($\cup$) or conjunction mapping ($\cap$). Filtering mapping, which is performed by disjunction mapping and conjunction mapping, is denoted by $M^{FSR}_{filtering}$ for $f_i \in F$ and defined as follows.

Definition 3.4. $M^{FSR}_{filtering} = \{M^{FSR}_{filtering}(O)\,|Min\{\alpha \leq \mu(B_i)$ *and* $\mu(S_j) \geq \alpha\}$ *or* $M^{FSR}_{filtering} = \{M^{FSR}_{filtering}(O)\,|FSR < 0.6\}$.

As an example, if $M^{0.1\cup 0.4}_{filtering}(o)$ is performed for each object cache block CB_1, CB_2, and CB_3 in Table 2, then according to the above definitions such objects as $CB_1 = \{o_1, o_3\}$, $CB_2 = \{o_2, o_4\}$, and $CB_3 = \{o_3, o_4\}$ are filtered out.

TABLE 2. Object cache block structure

CB \ o	o_{11}	o_{12}	o_{13}	o_{14}
CB_1	0.3	0.7	0.2	
CB_2	0.6	0.1	0.8	0.4
CB_3	0.5	0.9	0.3	0.2

4. **Simulation Results.** In the simulation, the total number of media objects is set to N, and 3 VoD sources are used as standard video sequences. The VoD sources have media object frames of 1,350, 1,450, and 1,500, respectively. To simplify the simulation, the media object version is assumed to have a Pareto distribution and is limited to a size of 5Mbytes. The other parameters are as follows: the maximum bit rate is 1.5Mbps, the

packet size is 512kb, the link bandwidth is 10/100Mbps, and the average link bandwidth is 1.2Mbps. The simulation continued for 560s with $\mu \geq 0.7$, $0 < \alpha < 1$, and the time stamp of stream t_s in $[1, 20\text{s}]$. The mobile client is assumed to be connected to the IP backbone network using a wireless IP. We evaluated the performance of the proposed scheme by using the simulation parameters shown in Table 3.

Table 3. Simulation parameters

Parameters	Value
Total number of media objects	4,300
Number of Partitioned versions	5
Media object block size	25Mbytes
Object version size	5Mbytes
Total simulation time	560s
Request time interval	2s
Time stamp	$[1, 20\text{s}]$
Maximum cache full	90%
Fuzzy similarity	$0 \leq \mu < 1$
Link bandwidth	10/100Mbps
Average link bandwidth	1.2Mbps
Maximum bit rate	1.5Mbps

We evaluated the performance by changing the fuzzy similarity, the size of the media objects, and the cache size. The major metrics used in the evaluation are the average startup latency, loss by average response ratio, and average cache hit ratio. The proposed method is compared with the other existing methods of the coverage-based method [10], demand-based method [10] and weight-based method [15].

In the simulation, we analyzed the performance of the average startup latency, average loss by response ratio and average cache hit ratio with increasing media object size when the fuzzy similarity is 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, and 0.9, respectively.

In the first simulation, we analyzed the performance of the average startup latency and the average loss by response ratio with varying fuzzy similarity.

In the second simulation, we analyzed the performance of the average cache hit ratio.

Figures 5-7 are simulation results obtained by increasing the media object block size from 5Mbytes to 25Mbytes.

As shown in Figure 5, the proposed method showed improvement in average performance compared with coverage-based method, demand-based method, and weight-based method.

Figure 6 is the simulation result obtained by varying the fuzzy similarity. As shown in Figure 5, the proposed method showed improvement in average performance. The reason is because the proposed scheme applies the fuzzy similarity. On the contrary to this, they degrade performance compared with the proposed scheme because other schemes do not apply the fuzzy similarity.

Figure 7 is the average cache hit ratio obtained by varying the media object block size from 5Mbytes to 25Mbytes. We simulated the performance with respect to the increase of fuzzy similarity, and obtained the best results when the fuzzy similarity was 0.8 and 0.9.

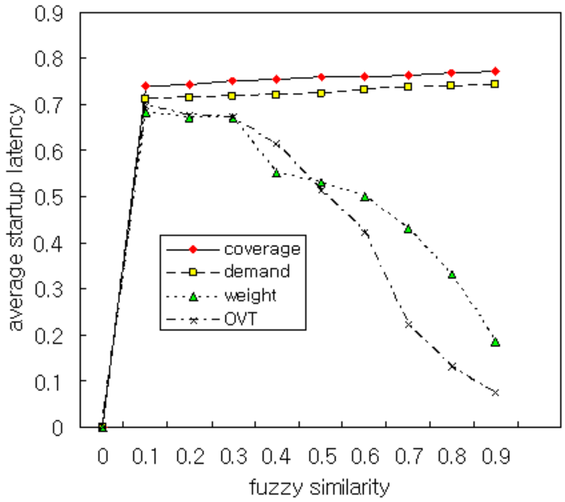


FIGURE 5. Average startup latency according to the fuzzy similarity

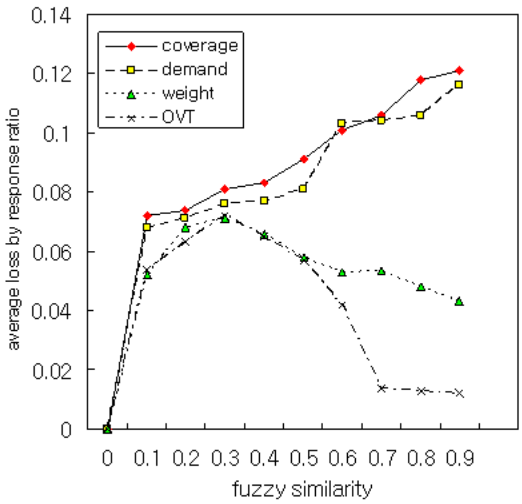


FIGURE 6. Average response loss ratio according to the fuzzy similarity

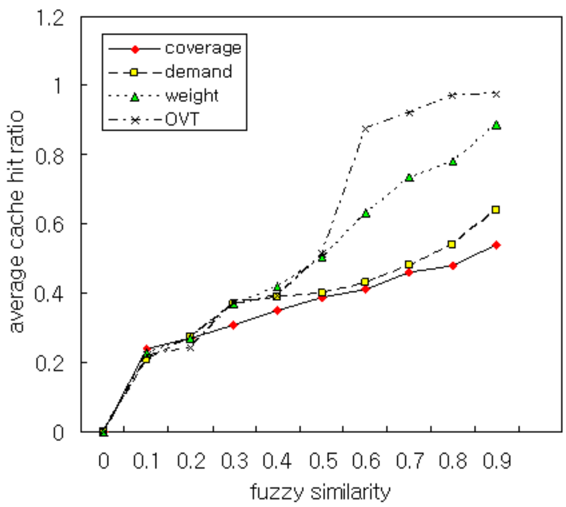


FIGURE 7. Average cache hit ratio according to the fuzzy similarity

The proposed method has better results than the weight-based method, which achieves excellent performance. This shows that the proposed method is not influenced by wireless network constraints, media object characteristics, cache capacity or overhead constraints. Hence, the proposed method can perform efficient transcoding, and streaming media services can maintain a stable state.

5. **Conclusions.** Transcoding in wireless networks is an important means to enhance QoS of streaming media services. In this paper, we proposed a new object version transcoding method based on fuzzy similarity for streaming QoS. We constructed a fuzzy similarity graph for the transcoding specification, which was used to determine the transcoding relationship for object segment versions of media objects. The proposed method determines transcoding for streaming media services of media object versions. We utilized the α-level to eliminate the ambiguity of transcoding and to improve the performance of streaming media services. We analyzed the performance of the proposed method to confirm the simulation results. The simulation was performed by changing the media object size and fuzzy similarity in order to evaluate the performance. According to the simulation results, the proposed method showed improvement in average performance compared with the coverage-based method, demand-based method, and weight-based method.

Acknowledgment. This paper was supported by research funds from Chonbuk National University in 2010.

REFERENCES

[1] K. T. Yang, W. K. Lai and C. S. Shieh, A cross-layer approach to packet size adaptation for improved utilization in mobile wireless networks, *International Journal of Innovative Computing, Information and Control*, vol.5, no.11(B), pp.4335-4346, 2009.

[2] C. D. Lee, T. W. Jeong and J. Y. Ahn, TRM-based multimedia streaming optimization scheme in wireless networks, *International Journal of Innovative Computing, Information and Control*, vol.7, no.4, pp.1651-1660, 2011.

[3] M. Qin and R. Zimmermann, An adaptive strategy for mobile ad hoc media streaming, *IEEE Trans. on Multimedia*, vol.12, vol.4, pp.317-329, 2010.

[4] C. Chang and M. S. Chen, On exploring aggregate effect for efficient cache replacement in transcoding proxies, *IEEE Trans. on Parallel Distributed System*, vol.14, no.6, pp.611-624, 2003.

[5] M. Hefeeda, A. Habib, B. Botev, D. Xu and B. Bhargava, Promise: Peer-to-peer media streaming using collectcast, *ACM International Conference on Multimedia*, Berkeley, California, USA, 2003.

[6] C. D. Lee and T. W. Jeong, Fuzzy filtering based segment grouping for user-centered multimedia streaming service in P2P-distribution mobile networks, *Journal of Internet Technology*, vol.11, no.5, pp.651-658, 2010.

[7] M. Prangl, T. Szkaliczki and H. Hellwagner, A framework for utility-based multimedia adaptation, *IEEE Trans. on Circuits and Systems for Video Technology*, vol.17, no.6, pp.719-728, 2007.

[8] W. Jiang, H. Li, H. Jin and X. Liao, Service guarantee for inter-domain movement in wireless peer-to-peer live streaming system, *The 10th International Symposium on Pervasive Systems, Algorithms, and Networks*, pp.533-538, 2009.

[9] M. Qin and R. Zimmermann, An adaptive strategy for mobile ad hoc media streaming, *IEEE Trans. on Multimedia*, vol.12, no.4, pp.317-329, 2010.

[10] V. Cardelli, P. S. Yu and Y. W. Huang, Collaborative proxy system for distributed web content transcoding, *Proc. of ACM Int'l Conf. Information and Knowledge Management*, pp.520-527, 2000.

[11] N. Dimokas, D. Katsaros and Y. Manolopoulos, Cache consistency in wireless multimedia sensor networks, *Ad Hoc Networks*, vol.8, pp.214-240, 2010.

[12] I. Akyildiz, T. Melodia and K. R. Chowdhury, A survey of wireless multimedia sensor networks, *Computer Networks*, vol.51, pp.921-960, 2007.

[13] Z. Miao and A. Ortega, Proxy caching for efficient video services over the internet, *Packet Video Workshop*, pp.1-21, 1999.

[14] B. Shen, S. J. Lee and S. Basu, Caching strategies in transcoding-enabled proxy systems for streaming media distribution networks, *IEEE Trans. on Multimedia*, vol.6, no.2, pp.375-386, 2004.
[15] C. F. Kao and C. N. Lee, Aggregate profit-based caching replacement algorithms for streaming media transcoding proxy systems, *IEEE Trans. on Multimedia*, vol.9, no.2, pp.221-230, 2007.
[16] K. T. Atanassov, Intuitionistic fuzzy sets, *Fuzzy Sets and Systems*, vol.20, pp.87-96, 1986.