무선 모바일 스트리밍 미디어 서비스를 위한 객체 세그먼트 그룹화

이종득*

Object Segment Grouping for Wireless Mobile Streaming Media Services

Chongdeuk Lee*

요 약 최근에 무선 모바일 네트워크상에서 모바일 클라이언트 요청증가로 인하여 스트리밍 미디어 객체를 관리하고 서비스하기 위한 새로운 기법이 제안되고 있다. 본 논문에서는 무선 모바일 네트워크상에서 스트리밍 미디어 서비스의 성능을 향상시키기 위한 새로운 객체 세그먼트 그룹화 방법을 제안한다. 제안된 기법은 분할된 객체 세그먼트들에 대해서 유사성 척도를 수행하며, 유사성 척도를 위해 disjunction, conjunction, 그리고 filtering연산을 수행한다. 이들 연산 척도에 따라 분할된 세그먼트들의 그룹화가 결정되며, 스트리밍 미디어 서비스 성능이 결정된다. 시뮬레이션 결과 제안된 기법은 처리율, 평균 시작지연, 그리고 캐시 히트율의 성능이 우수함을 보였다.

주제어 : 모바일 클라이언트, 스트리밍, 객체 세그먼트, 그룹화, 유사성

Abstract Increment of mobile client's information request in wireless mobile networks requires a new method to manage and serve the streaming media object. This paper proposes a new object segment grouping method for enhancing the performance of streaming media services in wireless mobile networks. The proposed method performs the similarity metric for the partitioned object segments, and it process the disjunction, conjunction, and filtering for these metrics. This paper was to decided the partitioned group of object segments for these operation metrics, and it decided the performance of streaming media services. The simulation result showed that the proposed method has better performance in throughput, average startup latency, and cache hit ratio.

Key Words : Mobile Client, Streaming, Object Segment, Grouping, Similarity

1. Introduction

In the wireless mobile network environment, the streaming media service is performed by connecting various mobile devices through the base station. In a wireless mobile network, mobile devices access the wireless network throughput base stations, which are inter-connected by access routers to perform WLANs. For the streaming media service in this environment, managing huge amounts of multimedia data is an important issue [1][2][3]. The reason is because the above environment can utilize efficiently the limited resources and channel bandwidth of wireless networks. Above all, managing and caching popular media object segments at locations close the mobile clients is an effective strategy to improve the quality of streaming media services. Especially, managing streaming media objects at the proxy cache is an attractive approach to reduce network traffic and improve quality of streaming media services, due to the high network bandwidth demand and continuous playback constraints of streaming media applications. To realize this,

*전북대학교 전자공학부 교수

논문접수 : 2012년 4월 25일, 1차 수정을 거쳐, 심사완료 : 2012년 5월 9일

segment-based proxy caching methods have been proposed, and this cache partial segments of media objects instead of entire media objects [5][7][9].

However, typical segment-based proxy mechanism makes streaming media services difficulty due to the large sizes and low-latency and continuous streaming demands of media objects. To solve this problem, segment-based caching methods have been proposed and the existing segment-based methods can be divided into the following two type according to the performance metric. The first type is to reduce startup latency by always giving a higher priority to caching the beginning segments of media objects. The second type is to reduce network traffic and jitter delay due to retransmission workload by improving cache hit ratio. Most proxy caching methods for streaming media services cache only a small portion of the media object in the proxy cache, and let the media server stream the remaining portions to the clients. The startup delay occurs when the request of clients can not be matched in the proxy cache. To solve this problem, the prefix method have been proposed and media objects in this method are divided into several segments, which are segments used by the proxy cache management [3][7]. For larger media objects, however, if segments do not arrange according to size, the delay due to cache replacement will be increased. Thus prefix proxy caching method are not efficient for streaming media services. To address the problems, Wu et al. [7] proposed a segmentation proxy caching method, in which the media object is partitioned into variable-length segments. The sizes of the segments vary based on their distances from the start of the media object. Segment 0 has a size of 1. In this method, cache replacement is based on the caching value a segment, which is a combination of a segment's distance from the start of its object and the reference frequency of the object. However, this method has the drawback of victimizing and caching more than necessary media objects. Therefore, this paper proposes an object segment grouping method for improving the performance of streaming media service in the wireless mobile network. The proposed method decides the grouping of media object segments by using fuzzy similarity, namely, μ-cut. Our method arranges the object segments according to fuzzy value μ, and streams the object segments according to fitness metric. The proposed method not only gives an effective solution to address startup latency problem, but also improves the cache hit ratio and reduces the congestion due to retransmission. The paper is organized as follows. Some related work is introduced in section 2. In section 3, we propose object segment grouping for streaming media services. Some simulation results are given and discussed in section 4. Finally, the conclusions are drawn.

2. Related works

There has been research work on various aspects of multimedia caching and proxy services.

Generally, users in an excellent network service environment may enjoy a high quality transmission service while users with poor a network service environment may not. Thus, most proxy structures perform services with the bit rate unequal to the bandwidth for streaming media clip objects. Typical media proxy mechanisms neither consider streaming by partitioning segment versions of media objects nor perform services by integrating various versions of media objects [11][12]. Chang and Chen [1] proposed a mechanism that partitions different versions of objects during caching. This mechanism is efficient for segmenting web objects. For media objects, however, this mechanism has restrictions due to the large size. 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 cannot cache frames directly after the initial segment. Wu et al. proposed a dynamic control video buffer mechanism and transmission scheduling mechanism for flexible playback. Generally in terms of video referencing, frames are classified as high usage frames and low usage ones. This kind of mechanism appears mainly in Internet videos, such as live news.

Previous segment-based caching strategies cache segments with constant or exponentially increasing lengths, and they typically favor caching the beginning segments of the media objects [6][8][10]. Chen et al. [1] proposed a segment-based buffer management approach for caching large media streams. Blocks of a media stream received by a proxy server are grouped into variable-sized segments. The segment-based caching mechanism is very effective when the cache size is limited, when the media object size is constant, and when streaming requests are restricted. In contrast, this mechanism has the drawbacks in that it needs to know in advance the size of media objects, the caching capacity, and the sequence of media object streaming before the cache is accessed.

3. The proposed object segment grouping mechanism

3.1 Similarity

Similarity analysis is to specify relationship occurring in object segments. This paper reflects the Fuzzy Similarity (FS) of each media object segment to transcode relationship occurring in grouping and to provide differentiated services. Fuzzy similarity is defined by membership function μ,where μ is a number in [0, 1]. Each member in FS has a membership value defined by μ [8]. ϕ(S) is the fuzzy set generated from the existing item set S. Each item of crisp set S has the membership value in [0, 1]. In this section, we describe the fuzzy similarity and fuzzy relation, which is used to derive the fuzzy similarity.

Definition 1 Fuzzy Relation: A fuzzy set A on the domain G×M, where G and M are two crisp sets in a fuzzy relation on G, M. Fuzzy set A on domain U is defined by a membership function μ from U to [0,1], i.e., each item in A has a membership value given by μ. We denote ϕ(S) as a fuzzy set generated from a traditional set of items S. Each item in S has a membership value in [0,1].

Definition 2 Intersection of Fuzzy Sets: The Fuzzy Intersection of A and B is A∩B, and $\mu_{A\cap B}(x)$=Min $\{\mu_A(x),\ \mu_B(x)\}$.

Definition 3 Union of Fuzzy sets: The Fuzzy Union of A and B is A∪B, and

$\mu_{A\cup B}(x)$=Max$\{\mu_A(x),\mu_B(x)\}$.

Definition 4 Max-Min Composition of fuzzy sets: Let P(X, Y) be the fuzzy relation of X and Y, and P(Y, Z) be the fuzzy relation of Y and Z.

Then, the Max-Min combination for P(X, Y) and Q(Y, Z) is $\mu_{p\,\bullet\,q}(x,z)=Max_{y\in Y}Min_{x\in X}\{\mu_p(x,y),\mu_q(y,z)\}$.

where the Max-Min combination represents the relationship between sets X and Z.

After fuzzy similarity analysis is completed, the grouping process is performed to specify relationship in more detail.

3.2 Relationship decision

In the process of segment acquisition of media objects, if there is a relationship satisfying the properties of media objects, the media objects is grouped, if not, the media objects isn't. Generally, the inclusion relation for grouping is determined in the process of the media partition of objects, and then it is important to search most general media object among media object blocks. Here, most general media object determines which media objects is grouped and is operated in a connection with other media objects. Table 1 is the parameters to decide an object relationship.

〈Table 1〉 The parameters for the relationship decision

Parameters	meaning
P(S)	The partitioned media object segments
R(S)	Object segments performed the relationship.
SR(S)	Object segments assigned the similarity
G(S)	Object segments for the grouping decision
F(S)	Object segments satisfied the fuzzy filtering
NF(S)	Object segments don't satisfy the fuzzy filtering

Here, the relationship decision of the partitioned object segments is represented as a SR(S) = {F(S)-NF(N)}.

This paper decides the similarity by using μ-cut, and it plays a role in filtering less referring segments in order as to determine relationship of object segments, the following:

Definition 5 μ-cut={ c | SR(S)≥0.6-cut}

Here, SR(S)≥0.6-cut is to decide the object segments assigned the similarity.

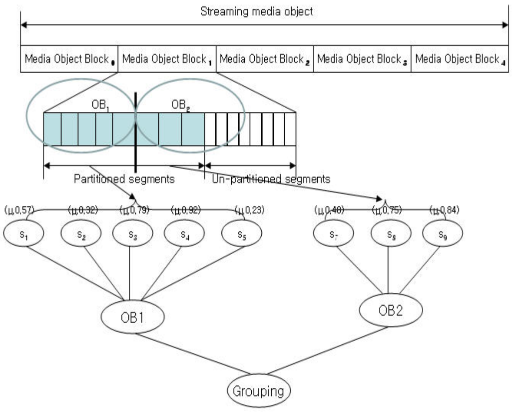
Thus, SR(S)≥0.6-cut must be satisfied a following conditions.

(1) Reflexive relation: $\mu \approx (os_i, os_j)=1$

(2) Symmetric relation: $\mu \approx (os_i, os_j)=(os_j, os_i)$

(3) Transitive relation: $\mu(os_i, os_k) \geq \min\{\mu\text{-cut} \approx (os_i, os_k), \mu \approx (os_j, os_k)\}$

Transitive relation is decided to similarity by μ-cut, and μ is membership function and fuzzy value between 0 and 1. μ is a membership which decide a relationship of SR(S) and Fig.1 depict a hierarchy based on by μ-cut.



[Figure 1] A hierarchy based on by μ-cut

As shown in figure 1, the grouping mechanism performs the similarity analysis to identify segments between partitioned segments and unpartitioned segments in media object blocks.

Here, a segment identification between the partitioned segment and unpartitioned segment in media object blocks is depicted as a SR(S)={F(S)-NF(S)}.

Therefore, when μ-cut is applied in figure 1, segments satisfying the grouping mechanism is selected.

3.3 Grouping decision

Grouping decision process is to drop the objects with low referring similarity in the grouping system. The unbalance between the encoding rate and decoding rate in mobile streaming media services is caused if less relevant objects are referred repeatedly the proxy cache structure. Then, cache underflow occurs if the objects are partitioned into too small segments and cache overflow occurs for too big segments. This problem induces the overhead of startup latency, congestion, and throughput decrease. This paper solves this problem by applying the object segment grouping mechanism. The object segment grouping mechanism is to control the streaming media service that is restricted by similarity, namely, μ-cut. The proposed grouping mechanism are restricted by cache capacity, size of objects, and similarity.

For restrictions C and $c \in C$, the proposed grouping process is carried through grouping by disjunction(∪), grouping by conjunction(∩), and grouping by filtering.

3.3.1 Grouping by disjunction

The Disjunction (∪) mapping groups object segments with Min-Max relation considering the cache capacity and the size of objects.

$G^{C,C}_{disjunction}$ and $G^{s}_{disjunction}$ denote the replacement mapping that considers the cache capacity and the size of objects, respectively. Thus, disjunction (∪) mapping is defined as the following.

Definition 6 $G^{(C,C)\cup s}_{disjunct}$: $\forall\, c, c' \in C$ c is c∪c′.C is cache capacity, and s is the size of object segments.

3.3.2 Grouping by conjunction

The grouping by conjunction(∩) considers the capacity of the cache, the size of objects, and similarity, and it is used to group the object segments with Min-Max relationship.

$G^{C,C}_{conjunction}$, $G^{s}_{conjunction}$, and $G^{sim}_{conjunction}$ represent conjunction grouping considering the capacity of the cache, size of objects, and similarity, respectively. Thus, the grouping by conjunction is defined as follows.

Definition 7 $G^{(C,C)\cap s \cap sim}_{conjunction}$: c∩c′∈C, ∀c,c′∈C where ∀c,c′∈C and ∅ is null.

3.3.3 Grouping by filtering

The grouping by filteringis used to control object segments that do not satisfy similarity.

The grouping by filtering, which is performed by similarity and relationship, is denoted by $G^{sim}_{filtering}$ for $f_i \in F$ and is defined as follows.

Definition 8 $G^{sim}_{filtering}$ ={ $G^{sim}_{filtering}$ (o)|Min{$\alpha \le \mu$ (Bi) and μ(Sj)≥0.6-cut}

As shown in figure 1, if $G^{0.1 \cup 0.6}_{filtering}$ is performed for each media object block OB_1 and OB_2, then such objects as OB_1={s_1,so_4}, OB_2={s_8, s_9} selects as object segments for grouping, and such objects as OB_1={s_1, s_2, s_5} and OB2={s_7} are filtered out. These objects are filtered out since they are segmented into too small sizes or too large sizes.

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. VoD sources have media object frames of 1,500, 1,500, and 1,500 in 3 news video, respectively. To simplify the simulation, the media object version is assumed to have Pareto distribution and is limited to 5Mbyte in size. The other parameters are as follows: maximum bit rate is 1.5Mbps, packet size is 512kb, link bandwidth is 10/100Mbps, and the average link bandwidth is 1.2Mbps. The simulation continued for 560s with $\mu \ge 0.7$ and the time stamp of stream t_s in [1, 20s]. 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 the Table 2.

[Table 2] Simulation parameters

Parameters	value
Total number of media objects	4,500
The partitioned segment number	5
Media object block size	25Mbyte
Segment size	5Mbyte
Total simulation time	560s
Request time interval	2s
Time stamp	[1, 20s]
Maximum cache full	90%
Similarity	$0 < \mu \le 1$
Link bandwidth	10/100Mbps
Average link bandwidth	1.2Mbps
Maximum bit rate	1.5Mbps

We evaluated the performance with changing the similarity, the size of media objects, and the cache full state. Major metrics used in the simulation are throughput, average startup latency, and average cache hit ratio. The proposed method is compared and analyzed with the other existing methods of segment-based method and prefix-based method. Table 3 shows the result of this performance.

[Table 3] The result of performance

performance	segment	prefix	grouping
startup latency	0.41	0.31	0.04
cache hit ratio	0.81	0.86	0.9

In the first simulation, we analyzed the performance of the average startup latency and average cache hit ratio with varying fuzzy similarity. Fig. 2 shows the performance of average startup latency when similarity is 0.2, 0.4, 0.6, and 0.8, respectively.

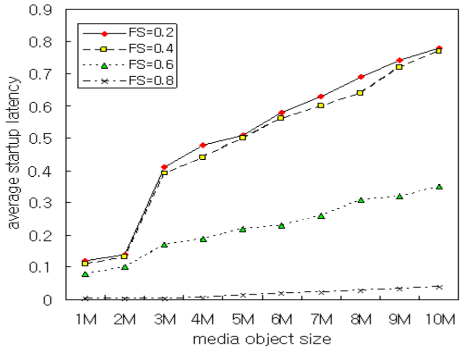


Fig. 2 Average startup latency with fuzzy similarity

As shown in Fig. 2, the average startup latency showed the best performance when similarity is 0.8.

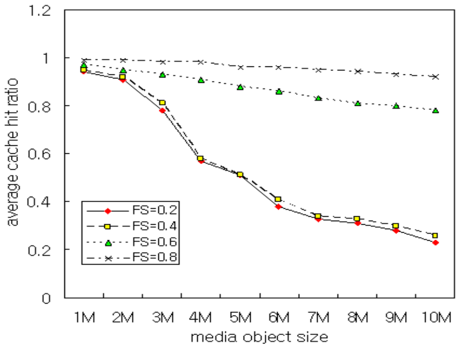


Fig. 3 Average cache hit ratio with fuzzy similarity

Fig. 3 shows the performance of the average cache hit ratio when similarity is 0.2, 0.4, 0.6, and 0.8, respectively. As shown in Fig. 3, the average cache hit ratio showed the best performance when similarity is 0.8. This is because the proposed method performed the grouping based on similarity for media object segments.

In the second simulation, we analyzed the performance of the average startup latency and the average cache hit ratio with increasing similarity.

Fig. 4 is the simulation result performing the media object block size from 10Mbyte to 50Mbyte.

As shown in Fig. 4, the proposed method showed average performance improvement compared with segment-based method and prefix-based method, respectively.

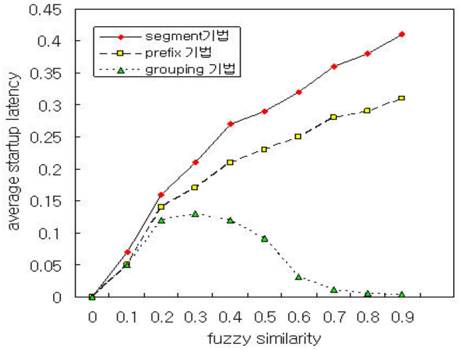


Fig. 4 Average startup latency according to the media object size

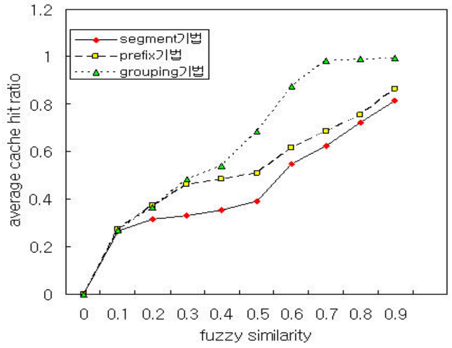


Fig. 5 Average cache hit ratio according to the media object size

Fig. 5 is the average cache hit ratio performing the media object size from 10Mbyte to 50Mbyte. We simulated the performance with the increase of the fuzzy similarity and had better result than others when fuzzy similarity is 0.8 and 0.9.

The proposed method has better result compared with the segment-based method that achieves excellent performance. Therefore, the proposed method is not influenced by wireless network conditions, media object characteristics, cache capacity and overhead constraints. Hence, the proposed method performs an efficient grouping and streaming media services hold a stable state.

5. Conclusions

This paper proposed a grouping method of object segments based on μ-cut, The proposed method constructs groups automatically by applying object partition as the size of the media object becomes larger. The proposed method caches streaming object segments in the proxy system systematically and enables client's request to serve.

The grouping structure keeps an inheritance of the super group and the sub group according to their relationship and establishes the subsumption relation according to the inheritance. The subsumption relation between the concept objects proceed by μ-cut. The proposed mechanism was to be controlled grouping through disjunction, conjunction, and filtering. Therefore, the proposed method not only provides finer partition of media object to allow more effective streaming media service, but also offers a excellent performance. The simulation result showed that the proposed method has better performance in throughput, average startup latency, and cache hit ratio.

References

[1] C. Chang, M. S. Chen. (2003). On Exploring Aggregate Effect for Efficient Cache Replacement in Transcoding Proxies. IEEE Transactions on parallel distributed system, 14(6), 611-624.

[2] C. D. Lee, T. W. Jeong, J. Y. Ahn. (2009). A Streaming Service based on Partition and Mapping of Media Block. ISME2009, 6th International Symposiumon Management Engineering, 5-7 August, Dalian, China, CD, 1-6.

[3] C. D. Lee, T. W. Jeong. (2010). Fuzzy Filtering Based Segment Grouping for User-Centered Multimedia Streaming Service in P2P Distribution Mobile Networks," Journal of Internet Technology, 11(5), 651-658.

[4] C. D. Lee. (2011). Object version Transcoding for streaming media service in Wireless Mobile Networks. KONI, 15(3), 355-363.

[5] C. D. Lee. (2011). Multi-level streaming using Fuzzy Similarity in P2P Distribution Mobile Networks." KONI, 15(3), 364-371.

[6] C. F. Kao, C. N. Lee. (2007). Aggregate Profit-Based Caching Replacement Algorithms for Streaming Media Transcoding Proxy Systems. IEEE Transactions on multimedia, 9(2), 221-230.

[7] K. L. Wu, S. Yu, and J. L. Wolf. (2004). Segmentation of Multimedia Streams for Proxy Caching. IEEE Transactions on Multimedia, 6(5), 770-780.

[8] K. T. Atanassov. (1986). Intuitionistic Fuzzy Sets. Fuzzy Sets and Systems. 20, 87-96.

[9] M. Prangl, T. Szkaliczki, H. Hellwagner. (2007). A Framework for Utility-Based Multimedia Adaptation. IEEE Transactions on Circuits and Systems for Video Technology, 17(6), 719-728.

[10] M. Qin, R. Zimmermann. (2010). An Adaptive Strategy for Mobile Ad Hoc Media Streaming. IEEE Transactions on Multimedia, 12(4), 317-329.

[11] M. Qin, R. Zimmermann. (2007). Supporting Guaranteed Continuous Media Streaming Performance through Adaptive Layer Selection with Scalable Video Coding," in Proc. 15th Annu., ACM

International Conference on Multimedia, 23-29.
[12] W. Jiang, H. Li, H. Jin, X. Liao. (2009). Service Guarantee for Inter-domain Movement in Wireless Peer-to-Peer Live Streaming System. 2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks, 533-538.
[13] Z. Miao, A. Ortega. (1999). Proxy caching for Efficient Video Services over the Internet. In Packet Video Workshop, 1-21.

이 종 득

- 1983년 2월 : 전북대학교 컴퓨터과학과(이학사)
- 1989년 2월 : 전북대학교 컴퓨터과학과(이학석사)
- 1998년 2월 : 전북대학교 컴퓨터과학과(이학박사)
- 1992년 3월~2002년 2월 : 서남대학교 컴퓨터통신학과 교수
- 2002년 2월~2012년 5월 현재 : 전북대학교 전자공학부 교수
- 관심분야 : 무선 모바일 네트워크, 무선센서 네트워크, MIMO, 유비쿼터스 통신, 등