International Journal of Innovative Computing, Information and Control
Volume **10**, Number **4**, August **2014**
ICIC International ©2014 ISSN 1349-4198
pp. 1339–1349

PROXY-DRIVEN STREAMING SERVICE FOR WIRELESS MULTIMEDIA ZONE SERVICES

Chongdeuk Lee
Division of Electronic Engineering
Chonbuk National University
Backje-daero, Deokjin-gu, Jeonju-si, Jeollabuk-do 561-756, Korea
cdlee1008@jbnu.ac.kr

Received August 2013; revised December 2013

Abstract. *Proxy-driven streaming service in wireless multimedia service zones (WMS Z) is challenging due to rate distortion and high channel error rate. To facilitate proxy-driven streaming media services in wireless multimedia service zones, streaming operations such as distortion detection, distortion adjustment, and caching management should be considered. A new proxy-driven streaming service (PDSS) is proposed to support high streaming quality in IEEE 802.11-based wireless multimedia service zones. The proposed mechanism performs disjunction mapping-based management operation, conjunction mapping-based management operation, and filer mapping-based management operation to derive the optimized proxy caching for the individual versions of a media object. The goal of this study is to reduce the distortion due to limited bandwidth, high bit rate error, and proxy constraints, and to enhance the throughput of streaming media service. Simulation results show that the proposed method has better performance than the existing server-only streaming method, resource-based caching method, and frequency-based caching method.*
Keywords: Proxy-driven streaming service, Rate distortion, WMSZ, Proxy caching

1. **Introduction.** In wireless multimedia service zones (WMSZ), streaming media service is performed by processing various mobile client information requests. Generally, in wireless multimedia service zones (WMSZ), mobile clients access the wireless network through base stations, which are connected by access routers to perform WLANs. For the streaming media service in this environment, managing substantial large-sized media objects and forecasting rate distortion are very difficult [1-3].

Caching and managing important media objects at locations close to mobile clients are promising strategies for improving the quality of streaming media services. In particular, an attractive means to reduce client-perceived access latencies as well as server/network loads is to cache frequently used media objects at proxies close to clients. However, wireless multimedia networks have a number of unique challenges compared with TCP-based streaming services [2,4]. Wireless multimedia zone service quality can fluctuate highly due to higher channel bit error, rate distortion, interference, flow inconsistency, and proxy congestion [5-7]. To solve such issues, Krishnamachari et al. [8] proposed an adaptive cross-layer protection strategy to enhance the quality of scalable media transmission. Majumdar et al. [9] proposed a hybrid FEC/ARQ method to increase the robustness of media streaming in IEEE 802.11 wireless LAN (WLAN). Wang et al. [10] proposed a media proxy mechanism in the base station to reduce ARQ delay. While these methods enhance the performance of media streaming over wireless networks, none of them have considered the optimized caching mechanism to solve rate distortion caused by delay and congestion.

The proxy-driven streaming service (PDSS) is done to differentiate streaming media services while considering rate distortion and channel bandwidth, since the transport protocol needs to decrease the sending rate only when there is a rate inconsistency in the network.

Proxy-driven streaming mechanism supports communication services such as coverage service, zone service, and communication services within hundreds of meters in the real wireless communication environment. As shown by applications to various services, proxy-driven streaming mechanism supports media streaming services with other users within the service area efficiently allowing free migration within a communication area.

Therefore, a proxy-driven streaming service mechanism is proposed that can be used to provide high-quality media streaming service over wireless multimedia service zones. Instead of using an established mechanism, disjunction mapping-based caching operation, conjunction mapping-based caching operation, and filer mapping-based caching operation are applied to optimize the streaming performance from the buffer cache. The proposed mechanism selectively controls media units based on their relative popularity so that the distortion can be reduced under certain cache constraints. Thus, the proposed mechanism has shorter distortion and better performance than other mechanisms.

This paper is organized as follows. Related works are introduced in Section 2. Section 3 introduces the proxy-driven service mechanism to improve the quality of streaming media services. The simulation results are introduced in Section 4, and concluding remarks are introduced in Section 5.

2. **Related Work.** In wireless multimedia service zones, a proxy-driven media service mechanism provides an optimum streaming service according to the distortion control conditions. In the past, several proxy caching mechanisms have been studied to improve the quality of media streaming [3,11].

Tewari et al. [12] proposed a new disk-based caching policy called resource-based caching (RBC) mechanism, which considers bandwidth as well as storage capacity constraints, and caches a mixture of intervals and full files that have the greatest caching gain. Almeida et al. [13] proposed a revised RBC mechanism to allow sharing an allocated bandwidth of cached files. In the revised RBC mechanism, whenever a new full file is added to the cache, its bandwidth allocation information is added to a bandwidth table. When a request for a cached file arrives, a new stream from the bandwidth table is assigned to that request. Rejaie et al. [14] proposed a frequency-based caching (FBC) mechanism to simply cache the files or partial files that are estimated to have the highest access frequency at the current time. This method is efficient for segmenting web objects. For streaming media objects, however, this method has problems due to constraints such as higher channel bit error, rate distortion, and caching congestion.

Kalman et al. [15,16] proposed a source rate control method in the receiver to reduce receiver terminal delay by adjusting the streaming rate. Zhu et al. [17] proposed a source rate control method in the transmitter. This method controls the source rate in the transmitter by analyzing the channel conditions and QoS requirements of application domains.

Recently, various media proxy caching mechanisms have been proposed to improve the performance of streaming [18,19]. Chen et al. [20] proposed a segment-based proxy caching mechanism for streaming large media objects. Blocks of a media stream received by a media server are divided into segments of variable size. The segment-based proxy caching mechanism is very effective when the media size is limited, when the bandwidth is constant, and when streaming requests are restricted. However, this method has drawbacks that it needs to know in advance the size of media objects, the channel bandwidth, and the sequence of media object streaming.

Lee et al. [2] proposed a TRM-based multimedia streaming optimization method to improve the streaming QoS by considering minimum bandwidth, maximum bandwidth, and transmission rate in multimedia application domains. Yan et al. [21] proposed a media-friendly congestion control mechanism to improve the streaming QoS using a utility-based model. This method used two time slots: one for video quality and the other for TCP-friendliness. All of these mechanisms, however, suffer packet loss in the case of rate distortion and suffer degradation of the overall system performance in a wireless multimedia channel.

Thus, in the point of streaming media services, the existing mechanisms are not appropriate for supporting the proxy-driven streaming service. Table 1 shows the comparison of the existing mechanisms.

Table 1. The comparison of existing mechanisms

Approach used	Advantages	Disadvantages
Resource-based caching (RBC) [12]	The greatest caching gain.	Insufficient bandwidth allocation information.
Frequency-based caching (FBC) [14]	A highest access frequency.	Higher channel bit error and rate distortion.
Source rate control [17]	Assuring of a higher QoS.	Analysis of channel conditions in the transmitter.
Segment-based proxy caching [20]	Effective when the media size is limited.	To know in advance the streaming constraints
Transmission Rate Monitoring [2]	Assuring the quality of streaming service.	Should be known the status of resources directly.

3. **Proxy-Driven Streaming Service Mechanism.** In this section, a proxy-driven mechanism for streaming media service in a wireless multimedia service zone is described.

The proposed proxy-driven streaming service (PDSS) mechanism can be applied in various wireless communication service environments such as proxy caching service session, wireless coverage networking session, wireless multimedia service session, and scale free networking session.

Thus, the proxy-driven streaming service (PDSS) mechanism in this paper alleviates the degradation of streaming performance due to limited channel bandwidth in wireless communication environments based on 802.11.

The system structure and proxy-driven mechanism are described in more detail.

3.1. **System structure.** The objective of the proxy-driven streaming service mechanism is to optimize requesting transmission for streaming units from different service requesters, and to relay them to the client to minimize the perceived quality degradation.

This section introduces a proxy-driven streaming service structure, PDSS, to control distortion due to the inconsistencies in streaming rates of the proxy caching unit (PCU) and the mobile client (MC). Figure 1 shows the proposed PDSS structure, which consists of a streaming media server, a proxy caching unit, a mobile client, and a set of peers (mobile client m, $m \in \{1, \ldots, M(t)\}$). $M(t)$ is the number of peers in WMSZ at any time.

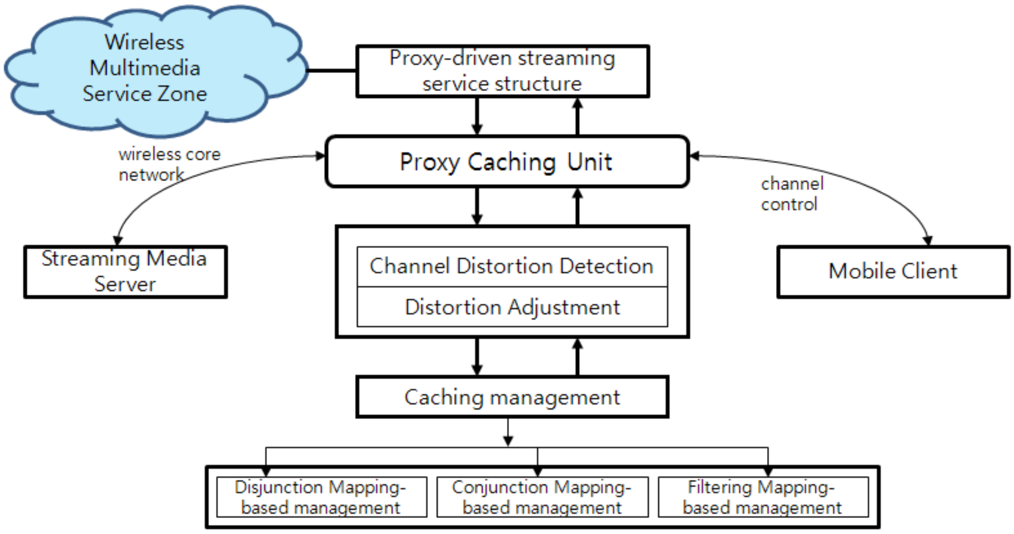


FIGURE 1. Proxy-driven streaming service structure

Media objects (mo_i) related to proxy caching in the PDSS have size s_i, decoding time T_i, and distortion rate D_i. The size s_i is the size of mo_i or streaming service in the proxy caching unit. The decoding time T_i is the time at which mo_i arrives at the client. The distortion rate D_i is the overhead rate when mo_i does not arrive at the proxy caching unit on time. The channel distortion control between client node m and PCU n is considered to optimize the caching operation.

3.2. **Proxy caching unit.** The proxy caching unit (PCU) is used to optimize the streaming media service while interacting with a streaming media server and mobile client. The proposed PCU includes channel distortion detection, distortion adjustment, and caching management operations.

3.2.1. *Channel distortion detection.* In Figure 1, consider the channel distortion between client node m and PCU n, CD_{mn}, $m, n \in \{0, \ldots, M(t)\}$ and $m \neq n$. Moreover, a forward path and backward path are considered to detect the channel distortion. As in Figure 1, each channel includes one forward path and a backward path with random delay distortion. The forward path is defined as the path from client node m to PCU n, and the backward path is the reverse. It is supposed that there is delay distortion when a client node m does not successfully arrive at PCU n. Therefore, for CD_{mn}, the distortion loss $L_{mn}^{FP(Forward\ Path)}$ can be derived due to delay distortion and the channel distortion $CD_{mm}(x) = e^{-\delta_{mn}^{FP}(x)}$ $(x \geq 0)$ between client node m and PCU n, where δ_{mn}^{FP} is a delay constant. If PCU does not meet the distortion loss and channel distortion between the forward path and backward path, the streaming packet sent by client node m will be correctly received by PCU at time T. Otherwise, PCU will encounter distortion loss and channel distortion. Therefore, the channel distortion caused by the forward path can be derived as the following equation:

$$CD_{mn}^{Forward\ Path}(T - \overline{T}) = (1 - L_{mn}^{Forward\ Path}) \int_0^T CD_{mn}(x)dx \tag{1}$$

where $\overline{T}$ is the duration from client node m to PCU n.

Similarly, the channel distortion from the backward path can be used to derive the distortion loss $L_{mn}^{BP(Backward\ Path)}$ and the channel distortion

$$CD_{mn}^{Backward\ Path}(x) = \delta_{mn}^{Backward\ Path} e^{-\delta_{mn}(x)}, \quad x \geq 0.$$

Therefore, the channel distortion caused by the backward path can be derived as the following equation:

$$CD_{mn}^{Backward\ Path}(\overline{T}-T) = (1-L_{mn}^{Forward\ Path}) (1-L_{mn}^{Backward\ Path}) \int_0^T CD_{mn}^{Backward\ Path}(x)dx \tag{2}$$

3.2.2. *Distortion adjustment.* If the transmission rates of the buffer cache and the requested media objects from clients are balanced, then the streaming media service is optimized. However, if the balance is violated, there will be distortion. This section describes the distortion adjustment based on the burst rate control operation and buffer cache control operation.

a) Distortion adjustment by burst rate: Generally, the size of the media object is dynamic. However, all fixed-size media objects and variable-size media objects are considered. Fixed-size media objects and variable-size media objects suffer from the same burst distortion in the proxy caching. To adjust the distortion with burst, The encoding rate and channel bandwidth between the client node and PCU are considered. Hence, the distortion adjustment for a requested media objects from clients, $D_{adjustment}$, for mo_i in the buffer cache can be defined as the following equation:

$$D_{adjustment} = mo_i \times T_{synchro} - \frac{E_r}{BW} \times s_i \tag{3}$$

where $T_{synchro}$ is the time synchronization for mo_i, s_i is the size of the i^{th} media object, and E_r is the average encoding rate for $SL(x)$.

The buffer cache size for mo_i should be considered after the determination of $D_{adjustment}$. The buffer cache size $BUR_{cache\text{-}size}$ for mo_i is defined as the following:

$$BUR_{cache\text{-}size} = T_{synchro} \times s_i \times \frac{E_r - BW}{E_r} \tag{4}$$

If $BUR_{cache\text{-}size} = 0$, then buffer caching is stopped and distortion occurs. Therefore, distortion adjustment operation is processed only for $BUR_{cache\text{-}size} \neq 0$.

b) Distortion adjustment by encoding rate: The rate of a requested media object and the caching rate can differ. Therefore, the following operations for the distortion adjustment from the encoding rate are considered.

1) $BC > S(x)$ and $BUR_{encoding} > 0$

This is the case of satisfying the encoding rate in the buffer cache (BC). It is necessary to check the cache capacity, since the buffer cache control depends on the media object size. Also, the encoding rate at the buffer cache provides better responsiveness at the client. Hence, distortion adjustment by the encoding rate in the buffer cache, $BUF_{encoding}$, is defined as follows:

$$BUF_{encoding} = \left\{ (T-\overline{T}) \sum_{i=1} \{S(x_i) - S(x_i - 1)\} \right\} \times \lambda \tag{5}$$

where λ $(0 < \lambda < 1)$ is the encoding rate, and $S(x)$ is a media object cached in the buffer cache. If $BUF_{encoding} = 0$, then the distortion adjustment is stopped and the process is started from the beginning.

2) $BC < S(x)$ and $BUR_{encoding} < 0$

This is the case of unsatisfying encoding rate in the buffer cache (BC). In the case of unsatisfying encoding rate, the relative popularity and cache capacity for media objects of the buffer cache should be checked from the beginning. The distortion adjustment by the encoding rate depends on the encoding rate λ. Therefore, larger λ means better distortion adjustment, and higher caching quality is continuously sustained.

3.3. **Caching management.** This section considers the disjunction mapping operation ($\cup$), conjunction mapping operation ($\cap$), and filtering mapping operation for efficient caching management. For restrictions C and $c \in C$, the mapping operations is defined as follows.

3.3.1. *Disjunction mapping-based caching management.* The disjunction ($\cup$) mapping operation controls objects with a min-max relation considering the cache capacity and the size of media objects. $M^{C,C}_{disjunction}$ and $M^{s}_{disjunction}$ denote the control mapping that considers the cache capacity and the size of media objects, respectively. Disjunction ($\cup$) mapping is defined as follows:

Definition 3.1. $M^{(C,C)\cup s}_{disjunction}$ *is* $\forall c,\ c' \in C$, c *is* $c \cup c'$. C *is the cache capacity, and* s *is the size of a media object.*

3.3.2. *Conjunction mapping-based caching management.* The conjunction ($\cap$) mapping operation considers the capacity of the cache, the size of media objects, and popularity, and it controls the objects with a min-max relationship. $M^{C,C}_{conjunction}$, $M^{s}_{conjunction}$ and $M^{popularity}_{conjunction}$ represent conjunction mappings considering the capacity of the cache, size of media objects, and popularity, respectively. Thus, the conjunction mapping is defined as follows:

Definition 3.2. $M^{(C,C)\cap s\cap popularity}_{conjunction}$ *is* $c \cap c' \in C$, $\forall c,\ c' \in C$, *where* $\forall c,\ c' \in C$, $\emptyset$ *is null.*

3.3.3. *Filtering mapping-based caching management.* Filtering mapping operation is done to control objects that do not satisfy disjunction mapping ($\cup$) or conjunction mapping ($\cap$). Filtering mapping operation is performed by disjunction mapping and conjunction mapping, and is denoted by $M^{popularity}_{filtering}$ for $f_i \in F$ and defined as follows:

Definition 3.3. $M^{popularity}_{filtering} = \{M^{popularity}_{filtering}(O)\,|Min\{(mo_i) \leq \mu \text{ and } \lambda(s_j) < \delta_{mn}\}$
μ $(0 \leq \mu < 1)$ *is a relative popularity, and* δ_{mn} *is a delay constant.*

Consider k media objects in the local buffer cache (BC) and size s_j for a media object o_i. In the local BC, n media objects are assumed to satisfy popularity. Figure 2 shows the local BC structure for media objects to which popularity is applied in the local BC. The caching management table for this is shown in Tables 2-4.

To efficiently manage new incoming media objects in the local BC, they are stored in the temporary bucket of the BC structure in Figure 2. Each media object is managed according to the popularity: filtering, relevant, and very relevant for the popularities of [0, 0.5], [0.5, 0.7] and [0.7, 1], respectively. As shown in Table 4, the new incoming object, I/0.72, satisfies the condition of B_2/0.7 in the block BC_2. Thus, media objects stored in the temporary bucket are managed in the local BC to perform streaming media services.

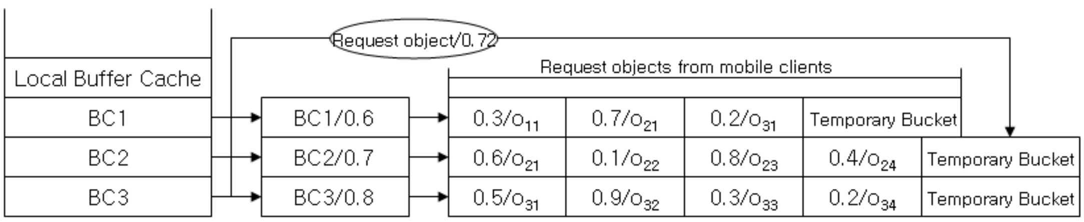


FIGURE 2. Local buffer cache structure

Table 2. Caching management table

BC \ O	o_{11}	o_{12}	o_{13}	o_{14}
BC_1	0.3	0.7	0.2	
BC_2	0.6	0.1	0.8	0.4
BC_3	0.5	0.9	0.3	0.2

Table 3. Mapping operation for $M^{0.1 \cup 0.4}_{filtering}(o)$

BC \ O	o_{21}	o_{22}	o_{23}	o_{24}
BC_1	**0.3**		**0.2**	
BC_2		**0.1**		**0.4**
BC_3			**0.3**	**0.2**

Table 4. Mapping operation for $M^{0.7 \cup 0.9}_{filtering}(o)$

BC \ O	o_{31}	o_{32}	o_{33}	o_{34}	
BC_1		**0.7**			
BC_2			**0.8**		**0.72**
BC_3		**0.9**			

4. **Simulation Results.** In the simulation, the total number of media objects is set to N, and 3 VoD sources are used as streaming media objects. A group of media objects for 3 VoD sources have 1,350, 1,450 and 1,500 media object frames respectively. To simplify the simulation, the media object is assumed to have a Pareto distribution and is limited to 5 Mbytes in size.

The other parameters are as follows: the maximum bit rate is 1.5 Mbps; the packet size is 512 kbps; the link bandwidth is 10/100 Mbps; the average link bandwidth is 1.2 Mbps. The simulation continues for 560 s with $0 < \lambda < 1$, and the time stamp of $(T - \overline{T})$ and $(\overline{T} - T)$ is set to 0.2 s. The mobile client is assumed to be connected to the IP backbone network using a wireless IP. The performance of the proposed method was evaluated using the simulation parameters shown in Table 5.

Table 6 summarizes the simulation result for proxy-driven streaming service (PDSS) and non-PDSS. Proxy-driven streaming service (PDSS) has excellent performance indices in average distortion rate and throughput.

As shown in Table 6, the average distortion rate and throughput of the proposed proxy-driven streaming service (PDSS) show that it is better than non-PDSS.

Simulations were done while changing the encoding rate, the cache capacity, and the burst length. The major metrics used in the simulation are the average distortion rate and the average throughput. The proposed method is compared and analyzed with the other schemes: server-only streaming method [3], resource-based caching method [13], and frequency-based caching method [14].

In the simulation, the performance of the average distortion rate was analyzed with the encoding rate, cache capacity, and burst length.

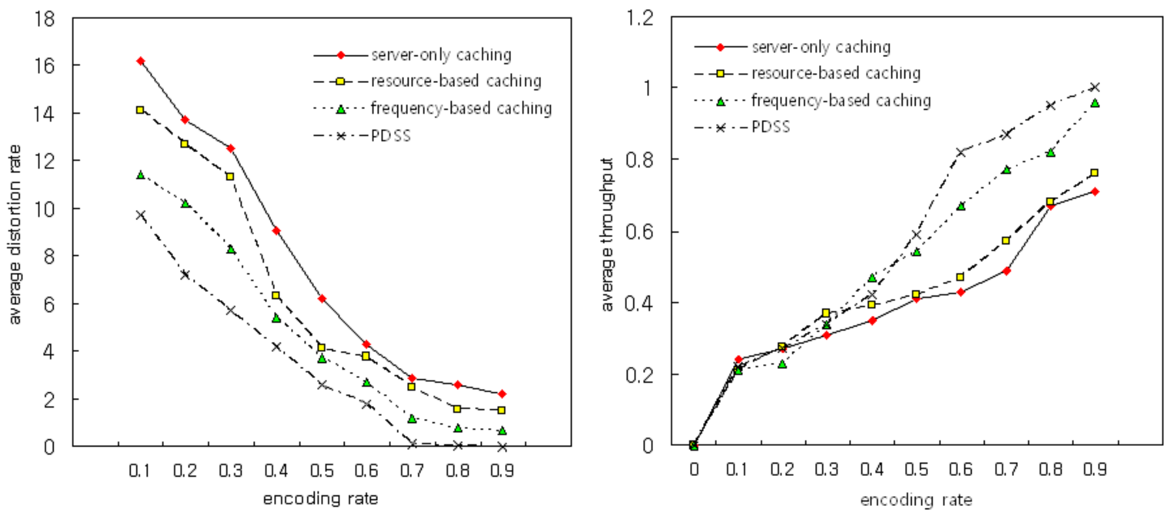
Figure 3(a) and Figure 3(b) show the performance of the average distortion rate and the average throughput, respectively, for different encoding rates ranging from 0.1 to 0.9.

Table 5. Simulation parameters

Parameters	value
Total number of media objects	4,300
Group size	2.5 Gbytes
Media object size	10 Mbytes
Total simulation time	560 s
Request time interval	2 s
Time stamp	0.2 s
Maximum cache capacity	100%
Relative popularity	0≤ μ <1
Link bandwidth	10/100 Mbps
Maximum bit rate	1.5 Mbps
Encoding rate	0< λ <1

Table 6. Performance comparison between PDSS and non-PDSS

Performance	PDSS	Non-PDSS
Average distortion rate	0.98	0.54
Throughput	0.98	0.71



(a) Average distortion rate (b) Average throughput

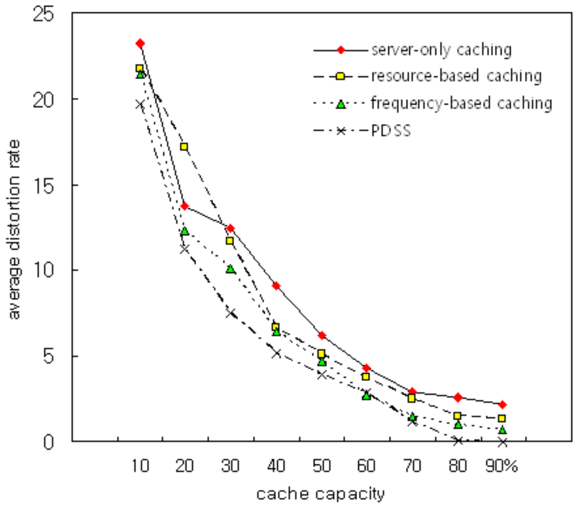
Figure 3. Average distortion rate and average throughput with encoding rate

As shown in Figure 3(a) and Figure 3(b), when the encoding rate is higher, the proposed method achieves better performance than other methods.

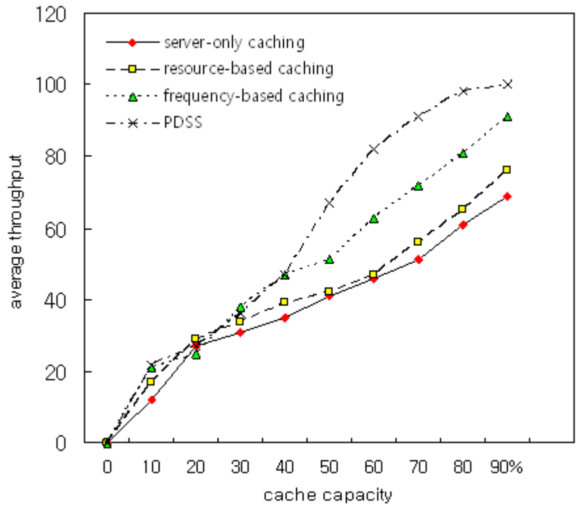
Figure 4(a) and Figure 4(b) show the performance of the average distortion rate and the average throughput, respectively, for cache capacities ranging from 10 to 90%.

As shown in Figure 4(a) and Figure 4(b), when the cache capacity is larger, the proposed method achieves better performance than other methods. Figure 5(a) and Figure 5(b) show the performance of the average distortion (dB) and the average throughput, respectively, for burst lengths ranging from 1 to 9.

As shown in Figure 5(a) and Figure 5(b), when the burst length is larger, the proposed method correctively controls the average distortion (dB) and average throughput.

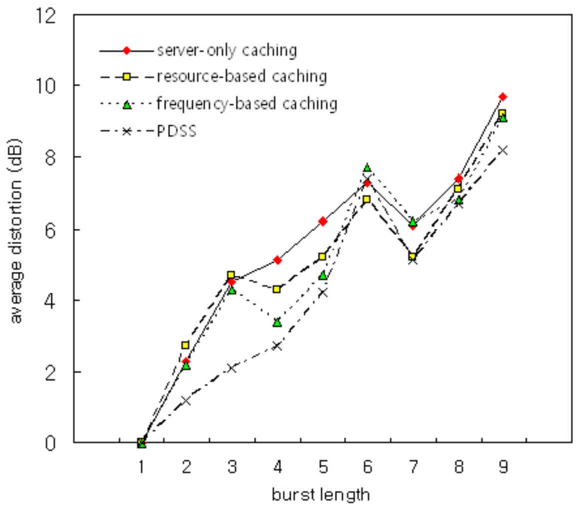


(a) Average distortion rate

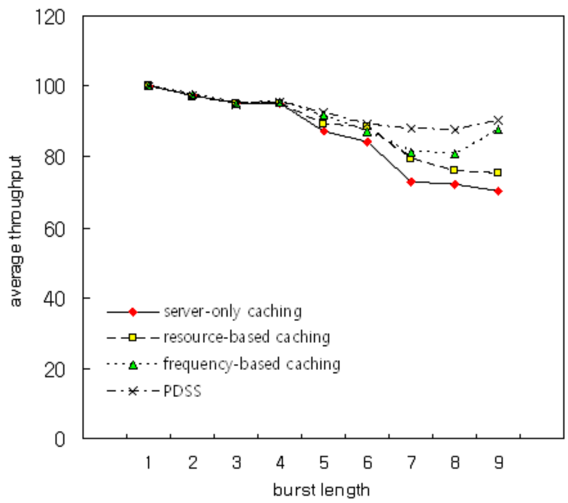


(b) Average throughput

FIGURE 4. Average distortion rate and average throughput with cache capacity



(a) Average distortion



(b) Average throughput

FIGURE 5. Average distortion and average throughput with burst length

This result is because the proposed method is applied to the distortion detection, distortion adjustment, and caching management for media objects. The simulation results showed that the proposed method has better performance compared with the server-only caching method, resource-based caching method, and frequency-based caching method. The proposed method is not influenced by the channel distortion, encoding rate, cache capacity, or burst length. Hence, the proposed method efficiently controls the distortion, and streaming media services are maintained in a stable state.

5. **Conclusions.** A new proxy-driven streaming service (PDSS) mechanism has been proposed to support high-quality media streaming service in wireless multimedia service zones. In PDSS, a proxy caching unit (PCU) processes the streaming media service while interacting with a streaming media server and mobile client. PDSS assures optimized proxy caching for the individual versions of a media object from the disjunction mapping-based management operation, conjunction mapping-based management operation, and

filer mapping-based management operation. Relative popularity was applied to efficiently control and manage the distortion under certain proxy cache constraints. As a result, the proposed mechanism maintained stable streaming service.

The simulation results showed that the proposed method has better performance than the server-only caching method, resource-based caching method, and frequency-based caching.

Acknowledgment. This research was supported by Basic Science Research Program through the National Research Foundation of Korea, (NRF) funded by the Ministry of Education (No. 2013R1A1A4A01005033).

REFERENCES

[1] C. D. Lee, T. W. Jeong, K. H. Choi and J. Y. Ahn, A proxy caching scheme based on partition and mapping of media block segments, *International Journal of Innovative Computing, Information Control*, vol.8, no.2, pp.1493-1500, 2012.
[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 Control*, vol.7, no.4, pp.1651-1660, 2011.
[3] D. Li, C. N. Nee, G. Cheung and S. J. Ben Yoo, MUVIS: Multi-source video streaming service over WLANs, *Journal of Communications and Networks*, vol.7, no.2, pp.144-156, 2005.
[4] C. D. Lee, T. W. Jeong and S. Lian, Tournament-based congestion control protocol for multimedia streaming in ubiquitous sensor networks, *International Journal of Communication Systems*, vol.24, no.6, pp.1246-1260, 2011.
[5] C. D. Lee and T. W. Jeong, FRCA: A fuzzy relevance-based cluster head selection algorithm for wireless mobile ad-hoc sensor networks, *Sensors*, vol.11, no.3, pp.5383-5401, 2011.
[6] C. D. Lee, Streaming media service based on fuzzy similarity in wireless mobile networks, *The Journal of Supercomputing*, vol.65, no.1, pp.86-105, 2013.
[7] T. W. Jeong, J. Y. Ahn and C. D. Lee, An object version transcoding for streaming media services in wireless networks, *International Journal of Innovative Computing, Information Control*, vol.9, no.1, pp.377-386, 2013.
[8] S. Krishnamachari, M. Van Der Schaar, S. Choi and X. Xu, Video streaming over wireless LANs: A cross-layer approach, *Proc. of Packet Video Workshop*, Nantes, France, pp.1-9, 2003.
[9] A. Majumdar, D. G. Sachs, I. V. Kozintsev, K. Ramchandran and M. M. Yeung, Multicast and unicast real-time video streaming over wireless LANs, *IEEE Transactions on Circuits and Systems for Video Technology*, vol.12, no.2, pp.524-534, 2002.
[10] T. Wang, H. Fang and L. Chen, Low-delay and error-robust wireless video transmission for video communications, *IEEE Transactions on Circuits and Systems for Video Technology*, vol.12, no.12, pp.265-268, 2002.
[11] 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.8, pp.651-658, 2010.
[12] R. Tewari, H. Vin, A. Dan and D. Sitaram, Caching in bandwidth and space constrained hierarchical hyper-servers, *Tech. Rep. CS-TR-96-30*, Department of Computer Sciences, University of Texas Austin, 1997.
[13] J. M. Almeida, D. L. Eager and M. K. Vernon, A hybrid caching strategy for streaming media files, *Proc. of Multimedia Computing and Networking*, San Jose, CA, USA, pp.1-13, 2001.
[14] R. Rejaie, H. Yu, M. Handley and D. Estrin, Multimedia proxy caching mechanism for quality adaptive streaming applications in the internet, *IEEE INFOCOM*, Tel Aviv, Israel, pp.1-11, 2000.
[15] M. Kalman, E. Steinbach and B. Girod, Adaptive media playout for low-delay video streaming over error-prone channels, *IEEE Transactions on Circuits and Systems for Video Technology*, vol.14, no.6, pp.841-851, 2004.
[16] B. Xie and W. Xeng, Rate-distortion optimized dynamic bit-stream switching for scalable video streaming, *IEEE International Conference on Multimedia and Expo*, pp.1327-1330, 2004.
[17] P. Zhu, W. Zeng and C. Li, Joint design of source rate control and QoS-aware congestion control for video streaming over the Internet, *IEEE Transactions on Multimedia*, vol.9, no.2, pp.366-376, 2007.

[18] S. Ghandeharizadeh and S. Shayandeh, A comparison of block-based clip-based cooperative caching techniques for streaming media in wireless home networks, *Proc. of the International Conference on Wireless Algorithms, Systems, and Applications*, Boston, MA, USA, pp.1-10, 2009.

[19] J. Liu and J. Xu, A survey streaming media caching, *Web Information Systems Engineering and Internet Technologies*, vol.2, pp.197-214, 2005.

[20] S. Chen, B. Shen, S. Wee and X. Zhang, Segment-based streaming media proxy: Modeling and optimization, *IEEE Transactions on Multimedia*, vol.8, no.2, pp.243-256, 2006.

[21] J. Yan, K. Katrinis, M. May and B. Platter, Media-and TCP-friendly congestion control for scalable video streams, *IEEE Transactions on Multimedia*, vol.8, no.2, pp.196-206, 2006.