ICIC Express Letters ICIC International ©2018 ISSN 1881-803X
Volume **12**, Number **3**, March **2018** pp. **199–204**

PROXY JITTER REDUCTION STRATEGY FOR WIRELESS COVERAGE NETWORKING SERVICES

Chongdeuk Lee and Taegwon Jeong

Department of Electronic Engineering
Chonbuk National University
567 Baekje-daero, Deokjin-gu, Jeonju-si, Jeollabuk-do 54896, Korea
{ cdlee1008; ttwjeong }@jbnu.ac.kr

Received September 2017; accepted November 2017

Abstract. *The proxy in wireless coverage networks has a problem for jitter caused by large-scale media traffic. A proxy jitter increases the startup latency and stream congestion, and also it degrades the system throughput and cache hit ratio. This paper proposes a new proxy jitter reduction scheduling algorithm to transmit the fast stream packets and to support the high quality for large-scale media traffic in the wireless coverage network. The proposed algorithm analyzes and manages stream packets under SMART (stream monitoring and response timely) scheme. A SMART has advantages that improve the byte hit ratio and the occupied caching assign latency, which provides insights for reliable proxy caching and QoS (quality of service) assuring. Simulation results show efficiency of our proposed algorithm.*
Keywords: Proxy jitter, Stream congestion, Wireless coverage network, SMART

1. **Introduction.** Recently, proxy caching in the wireless coverage network concentrates on increasing the caching performance, enabling stream packet for large-sized media traffic and reducing delay caused by caching congestions. Generally, when communication between a server and clients is performed, proxy caching suffers from various performance degradation. One of them is proxy jitter. Proxy jitter has many failure modes, and each failure degrades the caching performance and stream packet throughput. With the increasing media traffic such as video and audio streaming applications in the wireless network, this type of failures increases compared to traditional proxy caching mechanism for non-continuous media such as textual and image traffics [1,2]. The proliferation of media traffic makes the efficient proxy caching difficult and it provides the reasons for degradation of throughput. In general, different types of media traffics have different characteristics [1]. Media traffic is sensitive to jitter and it generates the retransmission and delay. Consequently, jitter caused by large-sized media traffic may have performance degradation under the proxy caching. Furthermore, large-sized media traffic to be cached may not be equivalently important and some traffic may have less importance compared to others. Thus, proxy jitter varies greatly according to the size of media traffic, capacity, and network conditions, and it has a heavy effect on cache performance and delay.

To solve the problems caused by large-scale media traffic, many proxy caching algorithms that cache partial segments of media traffic instead of entire media traffic have been proposed [3,4]. Among many strategies, algorithms that are popular are known as SPC (segment-based proxy caching) algorithm [1], traffic-camera assisted routing algorithm [5], and LACH (load aware channel hopping) algorithm [6]. The SPC algorithm focuses on reducing the client perceived the delayed startup ratio by always giving a high priority to caching the beginning segments of media traffic, and it also aims at reducing network traffic and server workload by improving proxy caching efficiency. The traffic-camera assisted routing algorithm uses the buffering selection and intelligent relay techniques to mitigate media traffic, and this algorithm minimizes the startup delay by delivering media stream packets to the mobile target [6], and the LACH algorithm enables nodes to dynamically adjust their schedules based on their traffic loads [7]. However, they do not automatically provide continuous stream packet delivery to the client since only partial stream packets are cached in the proxy. It is important for the proxy to fetch and relay the uncached stream packets in time whenever necessary. During this process, the proxy meets tremendous traffic volumes [7].

To reduce proxy jitter in the wireless coverage network, we propose a new proxy jitter reduction scheduling scheme based on SMART. The proposed scheme makes it cache uncached stream packets by the real time monitoring and response in the proxy. Thus, for the best interest of clients, the proposed scheme aims to minimize proxy jitter, and it aims at improving the reliable caching ability, maintaining the low packet loss, and minimizing the proxy management overhead.

The rest of this paper is organized as follows. In Section 2, we describe a SMART-based proxy jitter reduction scheduling procedure, and Section 3 describes the simulation results. Finally, in Section 4 we describe the conclusions.

2. **SMART-Based Proxy Jitter Reduction Scheduling.** The wireless coverage network is sensitive to network conditions and resource constraints. Especially, the cache resources in the proxy have a limited cache capacity and it can provide the source of proxy jitter due to the stream explosion when stream packets in clients request the proxy caching resource at the same time. To reduce proxy jitter, we describe a SMART-based proxy jitter reduction scheduling procedure. SMART-based proxy jitter reduction scheduling procedure in this section has jitter reduction monitoring strategy, proxy jitter avoidance strategy, and reliability strategy.

2.1. **Jitter reduction monitoring strategy.** SMART can reduce proxy jitter by monitoring the occupied buffer capacity traffic congestion in real time before media stream packets are accessed. As shown in Figure 1, when media stream packets msp_{11}, msp_{21}, and msp_{31} in mobile clients are forwarded to proxy at the same time, the proxy is suffered from traffic jitter due to the large-scale media stream packets.

In the case of large-scale media stream packets such as video, the buffer capacity and occupation in the proxy that each media stream packet currently forward should be considered. If not, media stream packets will meet the miss ration between the occupied buffer capacity and size, and it will eventually fail to satisfy the client's request. One to solve this problem is to re-allocate media stream packets by searching the relation between the miss-ratio monitoring and the proxy resource requirement for each type of

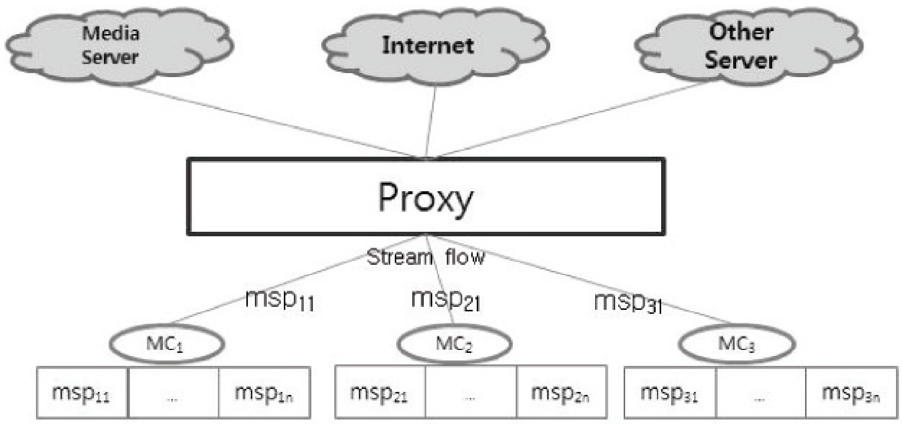

FIGURE 1. Proxy jitter monitoring

media stream packet. Let S_i represent the size that the media stream packet i currently occupies, P_i represent the priority of the media stream packet i, and TJ_i represent the traffic jitter for the i-th media stream packet. Then, the re-allocation RA_i for the i-th media stream packet is defined as Equation (1).

$$RA_i = \frac{TJ_i}{S_i} \times P_i \tag{1}$$

By considering parameters such as the occupied capacity, importance, and size, the media stream packet is re-allocated and sorted as $RA'_1, RA'_2, \ldots, RA'_m$. We apply the jitter detection coefficient (JDC) μ to detecting the jitter for MSP (media stream packet) in the proxy. If $|RA'_m - RA'_1| < \mu$, then all MSPs cannot suffer from the jitter due to the above parameters. This can reduce the possibility of jitter. However, if $|RA'_m - RA'_1| > \mu$, MSPs should be re-monitored and re-allocated from RA'_m to RA'_1. Therefore, jitter reduction monitoring considered by the size of MSP is defined as Equation (2).

$$C_{error}(t) = \sum_{i=k-L+1}^{t} e_n^2(i) > \mu \tag{2}$$

where L is the media traffic length for upstream MSPs.

When caching error due to buffer overload in the proxy occurs, the detection for caching error at time t is defined as Equation (3) and Equation (4).

$$e_n(t) = x(t) - \varepsilon(t) \tag{3}$$

$$C_{error}(t) = \sum_{i=1}^{t} e_n^2(i) \tag{4}$$

When the forwarding traffic exceeds the capacity of buffer cache or when the unfair caching process proceeds, then a caching jitter can occur.

2.2. **Jitter avoidance strategy.** If an MSP has not been served for a given period time, it will be provided a jitter source due to the delay. Jitter causes a packet loss, and it results in degrading the throughput and reliability. MSP_{delay} can have different characteristics among different types of MSP. To avoid the jitter due to MSP_{delay}, we consider the capacity of proxy cache. Therefore, jitter avoidance strategy has the following steps.

Step 1) Generate capacity requests for MSPs, which have not been served within MSP_{delay}.

Step 2) Generate requests for those in the miss queues with the average satisfaction for MSPs, which are used while caching MSP in the time interval t.

Here capacity requests for MSPs follow the priority queue scheduling. It is served first higher priority MSP and lower priority MSP is served afterwards.

Step 3) Accept the caching requests, if there is still some capacity left. It has the following sub-steps:

- Calculate whether there is caching request delay, CR_{delay}, to avoid the jitter for MSPs in the proxy as Equation (5).

$$CR_{delay} = \sum_{i=0}^{n} (|CR(MSP_{tot\text{-}delay} - MSP_{cur\text{-}delay})|) \times \mu \tag{5}$$

  where n is the number of caching requests for MSP that should be served at any time t. $MSP_{tot\text{-}delay}$ is the total delay for MSPs, and $MSP_{cur\text{-}delay}$ is the current delay for an MSP.
- Calculate whether there is sufficient caching capacity, CC_{left}, to accept the MSP in the proxy.

If the proxy violates the above steps, it causes the jitter. We enhance the proxy performance by applying the jitter avoidance strategy.

2.3. **Caching reliability strategy.** If we have adequately the above conditions, it will provide higher reliability for proxy performance. Reliability is a strategy that the proxy ensures throughput, jitter reduction, and caching optimization for a given period of time. The proxy generally has to support the caching reliability. If the forwarding MSPs toward proxy do not have traffic failure, the proxy will be ensured the caching reliability. Otherwise, it will be met the traffic congestion. To see the caching reliability, we discuss the request configuration for the sequence-connected MSPs and parallel-connected MSPs.

If the units in the sequence-connected MSPs do not interact, then the traffic failures are independent, and the caching reliability for the sequence-connected MSPs $CR_{sc}(t)$ denotes the product of the reliabilities of the individual constituent units. The caching reliability function with the sequence-connected MSPs is defined as Equation (6).

$$CR_{sc}(t) = \prod_{i=1}^{k} R_i(t) = \prod_{i=1}^{k} [1 - F_i(t)] \tag{6}$$

where $R_i(t)$ and $F_i(t)$ are the reliability and traffic failure distribution functions, respectively, of the i-th MSP unit. In the parallel-connected MSPs, a number of similar individual MSPs are connected in parallel. Figure 2 shows a parallel connection with k units.

FIGURE 2. Parallel connection with k units

To see the caching reliability for parallel-connected MSPs, we suppose the following two assumptions. (1) All MSPs are active and share the proxy resources, and (2) all units are statistically independent. For no identical units, traffic failure distribution function $TFD_{pc}(t)$ at any time t is defined as Equation (7).

$$TFD_{pc}(t) = \prod_{i=1}^{k} TF_i(t) \tag{7}$$

where $TF_i(t) = 1 - R_i(t)$ denotes the traffic failure distribution of the i-th unit. Therefore, since $CR_{pc}(t) + TFD_{pc}(t) = 1$, the caching reliability function for parallel connection is defined as Equation (8).

$$CR_{pc}(t) = 1 - \prod_{i=1}^{k} [1 - R_i(t)] \tag{8}$$

Generally, MSPs have many traffic failure modes due to the constrained proxy resource. Each traffic failure degrades the proxy caching performance. An excellent and complete jitter control scheduling algorithm for ensuring caching reliability of wireless coverage networking services can be accomplished by analyzing strategies described in the above conditions.

3. **Simulation Results.** In this paper, we simulated the simulation results using the MatLab to show the performance of the proposed scheme. For simulation, we used parameters such as network size, number of nodes, cache reliability, jitter coefficient, number of packets, transmission area, and simulation time. To see the efficiency of the proposed scheme, we compared the performance of the proposed scheme with that of the LRU scheme and the SPC scheme. We considered the following simulation scenarios to see the performance of the proposed scheme.

- scenario 1: We simulate the cache throughput by dividing the jitter coefficients from 0.1 to 0.9.
- scenario 2: We simulate the cache delay rate by dividing the number of stream packets from 10000 to 60000.

Figure 3 shows the simulation results of scenario 1. As shown in the simulation results, when the jitter coefficients are 0.7 or more, it showed that the proposed scheme is superior to other schemes. This is because the proposed scheme corresponds to large-scale traffic adaptively by monitoring proxy jitter in real time.

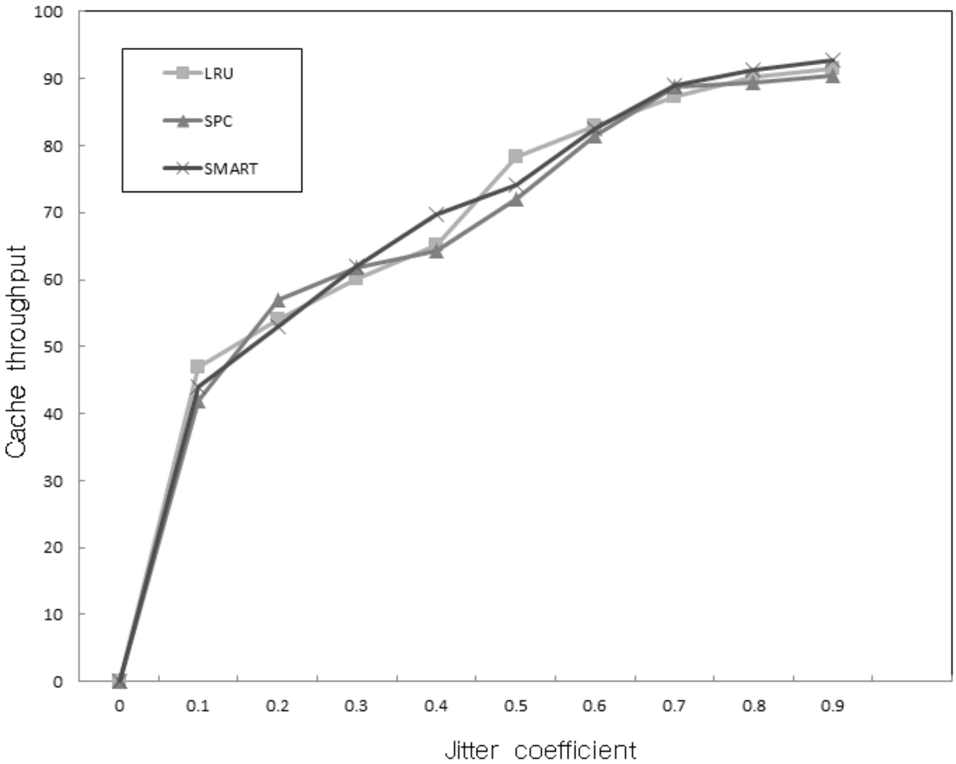

Figure 3. Cache throughput

Figure 4 shows the simulation results of scenario 2. In Figure 4, the cache delay was superior to the other schemes when the number of stream packets was increased. It can be seen that the proposed scheme is less affected by proxy jitter due to real-time monitoring even when the number of stream packets increases.

Therefore, when the jitter control coefficient is large, we can see that the cache throughput is more efficient and also the cache delay rate is controlled stably. As a result, it has the advantage of keeping proxy caching stable.

4. **Conclusions and Future Works.** The proxy jitter in wireless coverage networks degrades cache hit ratio, system throughput, and reliability due to the massive traffic. In this paper, we proposed a new proxy jitter reduction scheduling algorithm to transmit the fast stream packets and to support the high quality for the massive media traffic in

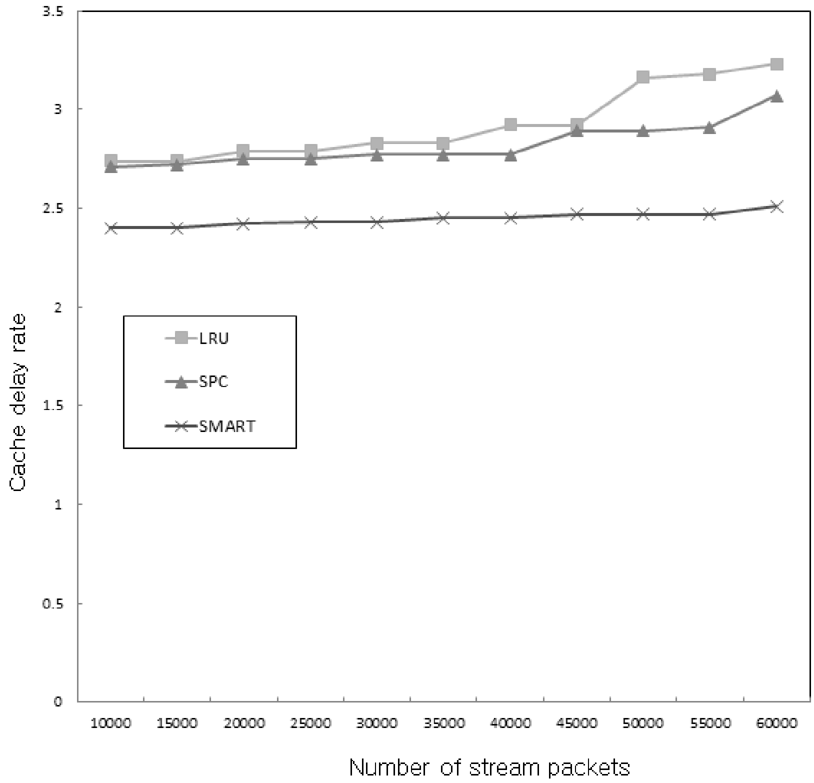

FIGURE 4. Cache delay rate

the wireless coverage network. The proposed algorithm efficiently controls the jitter by detecting proxy jitter sources and caching errors according to SMART procedure. We monitored stream packets according to the caching reliability strategy and found that the SMART procedure effectively detects jitter. Simulation results showed that the larger the jitter coefficient is, the more efficient the detection of proxy jitter and traffic error is. As the future work, we will study research to minimize proxy jitter by monitoring huge volumes of traffic in real time, and to apply the proposed algorithm in real application domain.

Acknowledgment. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2017R1A2B4006687).

REFERENCES

[1] C. Wang, B. Li, K. Sohraby, M. Daneshmand and Y. Hu, Upstream congestion control in wireless sensor networks through cross-layer optimization, *IEEE Journal on Selected Areas in Communications*, vol.25, no.4, pp.786-795, 2007.
[2] C. D. Lee and G. K. Lee, An alpha cut filter-based proxy caching control for wireless coverage streaming services, *Wireless Personal Communications*, vol.86, no.1, pp.35-55, 2016.
[3] C. D. Lee, T. W. Jeong, K. S. Han 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.
[4] 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 and Control*, vol.9, no.1, pp.377-386, 2013.
[5] H. Hu, H. Ma, L. Liu and P. Yuan, A traffic-camera assisted cache-and-relay routing for live stream delivery in vehicular ad hoc networks, *Wireless Networks*, vol.23, no.7, pp.2051-2067, 2017.
[6] C. M. Chao, H. C. Tsai and C. Y. Huang, Load-aware hopping protocol design for mobile ad hoc networks, *Wireless Networks*, vol.23, no.1, pp.89-101, 2017.
[7] C. Rotsos, D. King and A. Farshad, Network service orchestration standardization: A technology survey, *Computers Standards & Interfaces*, vol.54, no.4, pp.203-215, 2017.