Advanced Science and Technology Letters
Vol.65 (Electrical Engineering 2014), pp.81-85
http://dx.doi.org/10.14257/astl.2014.65.20

On Video Content Delivery in Wireless Environments

Po-Jen Chuang and Hang-Li Chen

Department of Electrical Engineering
Tamkang University
Tamsui, New Taipei City, Taiwan 25137, R. O. C.
e-mail: pjchuang@ee.tku.edu.tw

Abstract. To enhance the performance of wireless content delivery networks, this paper presents a new caching strategy based on analytical results of real user video request traces and adaptive caching considerations for videos of varied popularity. It caches videos with high popularity to all helpers first and those with low popularity to the remaining storage later, to lift the request hits. Experimental evaluation shows that the new strategy depicts better request hits than other caching strategies with reasonable complexity.

Keywords: Wireless content delivery networks, radio access networks, caching strategies, video request hits, complexity, experimental evaluation.

1 Introduction

In a content delivery network (CDN) [1], a number of cache servers are set outside the source server which will cache its contents to the cache servers, to help users get the needed data from the nearest cache server when filing requests. Such a practice can increase the delivery speed and reduce the burden on the source server. To pursue better performance for a wireless CDN, we thus need a proper caching strategy to set up desirable cache servers and decide on the cached contents. For instance, the **Greedy** strategy [2, 3] works by creating several cache nodes (helpers) in a single cell, but when the video volume grows far beyond the cache storage of helpers, it can only cache the videos with high popularity. For videos with low popularity (a considerable number), it yields very limited user request hits. This investigation aims to enhance the practice of radio access networks (RANs) by building a new caching strategy over wireless CDN. Based on the analytical results of real user video request traces, our new strategy adopts different approaches to handle videos with different degrees of popularity. It caches videos with high popularity to all helpers and those with low popularity to the remaining storage (if available) to lift the overall request hits. Simulation results show that when the cache storage of helpers goes below 50% of the total videos, **Greedy** attains only few hits for those less popular but numerous videos, **Popular** [3] yields much lower request hits due to caching highly popular videos only, and **Fuzzy Decision** [4] misses even the requests of highly popular videos because it does not consider video popularity at all. Our strategy is shown to yield higher request hits with lower complexity because it performs following the real user request traces.

ISSN: 2287-1233 ASTL

2 The Proposed Strategy

When caching a video, the **Greedy** strategy must ensure that each user can find the video in a nearby helper. But, it is difficult to predict accurately which user will make a request in which position and wrong user location predictions may cause huge impact. If users are evenly distributed, the strategy needs to cache every video to most of the helpers. If the video volume largely exceeds the total helper storage, helpers can cache only limited videos. In such a situation, the strategy which has attempted to handle most user requests by nearby helpers will leave a large amount of requests untended. To solve the problem, we build our new caching strategy following the analytical observations of real user request traces to increase the user request hits. To facilitate our investigation, we check the traces of YouTube requests from the wired campus network at the University of Massachusetts at Amherst, between the second half of 2007 and the first half of 2008 [5]. Table 1 gives one example of these request traces.

Table 1. The trace of a YouTube request.

Timestamp	YouTube server IP	Client IP	Request	Video ID	Content server IP
1189828805.208862	63.22.65.73	140.8.48.66	GETVIDEO	lML9dik8QNw	158.102.125.12

We take the number of requests for each video as its popularity and observe that the average user requests for videos with higher popularity is about 30% of the total requests each day. We also observe that, when removing these 30% *popular* videos, the remaining 70% *non-popular* videos exhibit little popularity differences. Our new caching strategy thus uses different approaches to handle popular and non-popular videos. In the first phase, we cache the popular videos - one by one, in the order of popularity - to all helpers until using up the helper storage or the popular videos. Note that we act differently from the Greedy strategy which starts its operation by caching *each* video to most of the helpers and, as a result, occupies too much helper storage from the beginning. The fact that our strategy begins with caching *popular* videos, instead of *all* videos, can save storage and enable helpers to ensure proper request hits. The design can meanwhile avoid the loophole of the Fuzzy Decision strategy: Helpers may fail to handle considerable requests for popular videos because the strategy will randomly cache videos several times and select the best result without considering the popularity degrees.

After the practice of the first phase, if the helpers still has cache storage, our strategy will start the second phase of operation. As the above trace analysis of the YouTube requests shows little popularity difference for non-popular videos, our strategy evenly distributes these non-popular videos - without considering the popularity - to the helpers in the second phase. Our key consideration in this phase is to cache possibly more non-popular videos to the helpers. By doing so, we can further enhance the request hits, given that most of the video requests are for non-popular videos (e.g., 70% in the above Youtube case). Unlike the Greedy strategy which caches very few non-popular videos or the Popular strategy which caches only popular videos, our strategy can cache significantly more non-popular videos to the helper (so as to increase the request hits), thanks to its second-phase practice. Our strategy is relatively

simple as it deals with the popularity degrees of videos only, involving no predictions of user locations in the Greedy strategy.

3 Experimental Evaluation

Extensive simulation runs are conducted to evaluate the performance of **our new caching strategy** (ours) and other related strategies, including **Greedy**, **Popular** and **Fuzzy Decision**. The simulation is carried out in a created single cell of the RAN environment with a number of helpers. We pick up the user request traces by randomly selecting from the Youtube requests (specified in the previous section) which were filed during the four *hottest* hours (i.e., hours with the most incoming requests) on January 29, 2008 - a total of 5252 requests filed for 3888 different videos. Each requested video is assumedly with resolution = 640 × 360, video length = three minutes and video size = 30MB. To attain fair simulation, we suppose the popularity of each video is known in advance and meanwhile follow the environment settings of the Greedy strategy in [3] - with cell radius = 400 meters and a random set of helpers in the cell = 32. The performance of the four target strategies is collected and compared in terms of *video request hits* and *complexity*.

Fig. 1 depicts the numbers of video request hits versus the helper storage (in GBs) for different strategies. When the helper storage is small, as the result shows, the **Popular** strategy yields fewer video request hits - because it caches only the popular videos. For **Fuzzy Decision** which considers no popularity degrees of the videos, increasing the helper storage does not bring apparent increase in request hits. It yields better request hits than **Popular** only at smaller cache storage - likely because of caching more different videos. **Our strategy** depicts the best hits between cache storage 30 ~ 60GB because we need not only the hits for popular videos but also that for the non-popular videos. The **Greedy** strategy is shown to generate more desirable video hits than our strategy between cache storage 60 ~ 90GB. This happens because it makes predictions on user locations and when there is sufficient helper storage, it can make sure videos be cached in the neighboring helpers based on the prediction results.

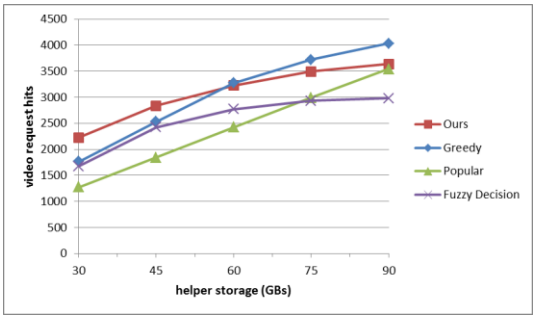

Fig. 1. Video request hits vs. the helper storage (GBs).

In Fig. 2, we gives the computational complexity of these strategies, denoted by the number of requested video lookups in the helpers. As the figure shows, the complexity

of **Popular** equals the total number of videos to be cached by the helpers. **Our strategy**, which randomly caches the non-popular videos, produces slightly higher complexity than **Popular**. **Fuzzy Decision** produces higher complexity than both **Popular** and **our strategy** because it needs 10 runs of random caching before reaching caching decisions. The **Greedy** strategy needs to search all user locations in order to decide the appropriate helper for caching each of the videos. It therefore needs the most complexity among all strategies.

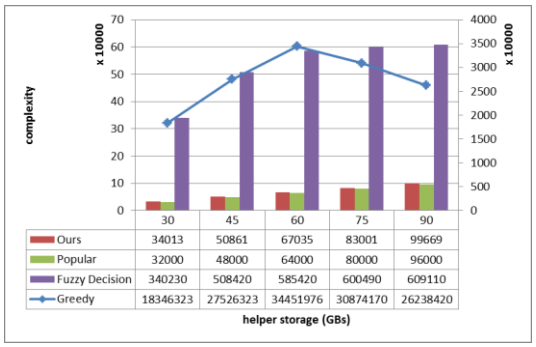

Fig. 2. Complexity vs. helper storage.

4 Conclusions

This paper presents a new caching strategy on the wireless CDN architecture to enhance the practice of RANs. Our new strategy, built following analytical results of real user video request traces, adopts different caching approaches to handle videos with different popularity degrees. It caches videos with high popularity to all helpers first and those with low popularity to the remaining storage later, and by doing so enhances the request hits of those less popular but numerous videos (which other caching strategies have failed to achieve). Simulation results show that, when compared with related strategies, our strategy can achieve higher request hits with reasonable complexity.

References

1. Buyya, R., et al. (eds.): Content Delivery Networks. Springer-Verlag, Berlin Heidelberg (2008)
2. Golrezaei, N., Shanmugam, K., Dimakis, A., Molisch, A., Caire, G.: Femtocaching: Wireless Video Content Delivery through Distributed Caching Helpers. In: 2012 IEEE INFOCOM, pp. 1107--1115 (2012)
3. Golrezaei, N., Dimakis, A. G., Molisch, A. F., Caire, G.: Femtocaching and Device-to-device Collaboration: A New Architecture for Wireless Video Distribution. IEEE Communications Magazine 51(4), 142--149 (2013)

4. Chen, J. B.: Efficient Content Placement on Multimedia CDN Using Fuzzy Decision Algorithm. Applied Mathematics & Information Sciences 6(2), 471—477 (2012)
5. YouTube Traces From the Campus Network (2008), http://traces.cs.umass.edu/index.php/Network/Network

