Video Caching in Radio Access Network: Impact on Delay and Capacity

Hasti Ahlehagh, Sujit Dey
Mobile System Design Lab, Dept. of Electrical and Computer Engineering
University of California, San Diego
{hahlehag, dey}@ucsd.edu

Abstract— **In this paper, we introduce distributed caching of videos at the base-stations of the Radio Access Network (RAN) as a way to reduce the need to bring requested videos from Internet CDNs, thereby reducing backhaul transmission, improving video quality of experience – delay and video stalling – and increasing overall network capacity to support more number of simultaneous video requests. Unlike Internet CDNs that can store millions of videos in a relatively few large sized caches, our proposed caching architecture consists of a very large number of micro-caches, with each base-station micro-cache being able to store only 1000s of videos, and hence may not be able to have high cache hit ratio. To address this challenge, we propose two new caching policies based on the User Preference Profile (UPP) of users in a cell: R-UPP (Reactive UPP) and P-UPP (Proactive UPP). Further, for videos that result in cache misses and need to be fetched from Internet CDNs, we develop a video scheduling approach that allocates the RAN backhaul resources to the video requests so as to reduce video latency and increase network capacity. We develop a discrete event statistical simulation framework using MATLAB to study the performance of RAN caching. Our simulation results show that RAN micro-caches with the proposed UPP-based caching policies, together with the proposed scheduling approach, can improve the probability of video requests that can meet initial delay requirements by almost 60%, and the number of concurrent video requests that can be served by up to 100%. The results also show that UPP based policies can enhance network capacity by up to 30% compared to conventional caching policies.**

Keywords: User Preference Profile, Proactive/Reactive Caches, Wireless Network Capacity, Video Quality of Experience

I. INTRODUCTION

With the world-wide growth in the adoption of smart phones and tablets, access to Internet video and video applications from mobile devices is projected to grow very significantly [1]. When Internet video is accessed by a mobile device, the video has to be fetched from the servers of a content delivery network (CDN)[2][3]. CDNs help reduce Internet bandwidth consumption and associated delay/jitter, but the video must additionally travel through the wireless carrier Core Network (CN) and Radio Access Network (RAN) before reaching the mobile device. Besides adding to video latency, bringing each requested video from the Internet CDNs can put significant strain on the carrier's CN and RAN backhaul, leading to congestion, significant delay, and constraint on the network's capacity to serve large number of concurrent video requests.

To facilitate the tremendous growth of mobile video consumption without the associated problems of congestion, delay, and lack of capacity, in this paper we introduce caching of videos at (e)NodeBs at the edge of the RAN, shown in Fig. 1, such that most video requests can be served from the RAN caches, instead of having to be fetched from the Internet CDNs. However, since the proposed approach will lead to thousands of caches, with each (e)NodeB in the carrier RAN having a cache (and may be Access Points in Wi-Fi hot spots), we need to use much smaller sized "micro-caches" for RAN caching, capable of storing 1000s of videos, compared to the much larger sized caches used in Internet CDNs capable of holding millions of videos. Hence, there may be a problem with enabling high cache hit ratio for the RAN micro-caches, which may erode the benefits of caching at the edge of the wireless network.

To address the above challenge, we propose novel caching policies, based on new concepts we introduce in the paper: the preference of current video users in a cell, and what videos are least likely and most likely to be watched by the cell users. For those video requests that cannot be found in RAN caches, and hence need to be fetched from Internet CDNs, we propose a video scheduling approach that allocates the RAN backhaul resources to the video requests such that the overall capacity of the network in terms of the number of concurrent video requests can be enhanced, while satisfying video Quality of Experience (QoE) – meeting an initial delay and ensuring no stalling during playback. Our numerical results show that RAN caching with the proposed caching policies can lead to significant improvements in terms of video delay and system capacity.

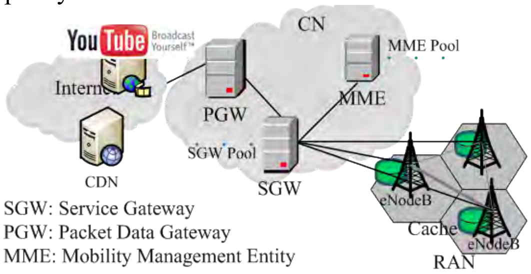


Figure 1: Video Micro-caches at the edge of the RAN

A. Related Work and Paper Outline

Significant work has been done in developing content delivery networks (CDNs) for Internet content [2][3], as well as caching techniques and locations suitable for Internet content delivery [4][5][6][7]. As explained earlier, Internet CDNs, and caching at Internet CDNs, do not address the problems of delay and capacity for video delivery in wireless networks. Moreover, as explained earlier, and shown in Section V, conventional caching techniques, which assume large caches, may not be effective for the smaller and distributed RAN micro-caches proposed in this paper.

There has been some research in caching web content in wireless networks [8] and on mobile devices [9]. However, these techniques do not consider the challenges of video caching and delivery. Caching techniques have been also developed for ad-hoc networks, like [10][11], which are not applicable to the problem of video caching and delivery in cellular networks. Recently, carriers have started caching online video content in as a way to address the delay and capacity problems arising from growing video consumption from mobile devices. However, to the best of our knowledge, video caching has not yet been attempted in the RANs. Similarly, we are not aware of published research in RAN video caching and delivery, including caching policies aware of the preferences of users in a RAN cell, and scheduling of videos to maximize video capacity while satisfying QoE, which are the problems we address in this paper.

The remainder of the paper is organized as follows: In section II, we first review relevant prior results on popularity of online video and video categories. Based on these results, we define the video category preferences of active users in a cell, and most likely requested and least likely requested videos of such users. In section III, we provide an overview of the conventional caching policies, and the user preference based policies we introduce. Subsequently, in section IV, we introduce our video scheduling approach, which allocates the RAN backhaul resources to the video requests that result in cache misses, such that the overall number of concurrent video requests that can be served by the RAN is maximized, while meeting desired video QoE. Section V outlines our simulation framework, and provides experimental results. We conclude the paper in section VI.

II. Video Preference of Users In A Cell Site

In this section, we first review previous research on video popularity characteristics and users' video access patterns. We highlight the conclusions that lead us to define our user preference based approach for video caching. Towards that end, we identify the most likely and least likely requested videos in a cell site, given the current set of active users in the cell.

A. Popularity of Online Videos and Video Categories

Recently, there have been several studies done on the popularity of online videos. Using empirical analysis, [12] studied several characteristics of videos from popular online video sites YouTube and Daum, such as the overall popularity distribution and distribution within each video category, correlation between age of a video and its popularity, and temporal locality of the videos. One of the relevant results from the study is that video popularity follows a Zipf distribution: 10% of the online videos account for nearly 80% of the views, while the remaining 90% of the videos account for only a total 20% of views [12]. In [13], the authors studied users' access patterns in video traffic from a campus network. Among other interesting conclusions, their results showed that local video popularity can differ significantly from national video popularity. Separately, market research [14] has shown that some video categories can be significantly more popular than others. For example, popular video categories such as "Auto" and "Entertainment" have a 90 days average cumulative views number that is 10 times more than less popular video categories such as Travel, revealing a strong bias towards some video categories.

From the above research, we conclude that: 1) video popularity follows a Zipf distribution, 2) national video popularity does not reflect local video popularity, so the video popularity in different cell sites may be different from each other and the national popularity distribution and 3) users may have strong preferences towards specific video categories. These results motivate us to define and identify video preferences of the active users in a cell in terms of video categories that they prefer to watch.

B. Cell Site Video Preference

To understand local video popularity in a cell site, we define Active User Set (AUS) of a cell as a set of mobile users in the cell who either have an active video session, or have watched and likely to again watch video when present in the cell. AUS changes as users enter or leave the cell site. For example, in LTE, (e)NodeBs know the location of the UEs in connected mode, or the general location of UEs in idle mode. We associate a User Preference Profile, UPP, with each individual user, which we define as the probability that the user, u_k, requests videos of a specific video category vc_j, $p(vc_j|u_k)$, for all available video categories. The probability that a video belonging to video category vc_j is requested by the active users in a cell, AUS, is the sum of probabilities that vc_j is being selected by each user in the AUS, and is given by:

$$p_{AUS}(vc_j) = \sum_{k=1}^{|U|} p(u_k) p(vc_j|u_k) \qquad (1)$$

In the above equation, |U| is the cardinality of AUS, and $p(u_k)$ is the probability that user, u_k, generates a video request. Following this definition, we define the UPP of an AUS as the selection probability of each available video category by the AUS: $\{ p_{AUS}(vc_j) | \forall j = 1{:}|VC| \}$. We assume all users are equally likely to generate a video request, so we can rewrite equation (1) as: $p_{AUS}(vc_j) = \frac{1}{|U|}\sum_{k=1}^{|U|} p(vc_j|u_k)$.

Next, given the overall video popularity distribution, and the category of each video, we identify the video popularity distribution within each category. Let $p(v_i)$ be the overall popularity of video v_i across all videos and video categories. Let $p_{vc_j}(v_i) = p(v_i)$ if v_i belongs to category vc_j, else $p_{vc_j}(v_i) = 0$. We can then express popularity of video v_i within video category, vc_j, by:

$$p(v_{i,j}) = \frac{p_{vc_j}(v_i)}{\sum_{i=1}^{|V|} p_{vc_j}(v_i)} \qquad (2)$$

where |V| is the total number of videos, and the denominator is the sum of the probabilities of all videos belonging to vc_j. Note that video popularity distribution may be available for each category [12], else it can be calculated using (2). Knowing the probability of request of different video categories in a cell corresponding to the current AUS (1), and the popularity of videos in each category, we can now derive $P_R(v_i)$, which is the probability that video v_i is requested given the AUS of the cell:

$$P_R(v_i) = \sum_{j=1}^{|VC|} p(v_{i,j}) p_{AUS}(vc_j) \tag{3}$$

We next define two sets that we use for the UPP based caching policies that we define in the next section: Most Likely Requested (MLR) and Least Likely Requested (LLR) sets. MLR is a subset of videos with P_R values greater than a threshold; and LLR is a subset of videos from the cache with the least P_R value.

III. Cell Site Aware Caching Algorithms

In this section, we outline four different caching algorithms; two that are conventionally used by Internet CDNs, MPV and LRU, and two that we propose based on preferences of active users in the cell, P-UPP and R-UPP.

A. MPV

MPV is a proactive caching policy, which caches the "most popular videos" using the (nation-wide) video popularity distribution described before. MPV neither updates the caches based on the user requests nor implements any cache replacement policy. The only changes that require cache update are changes in the video popularity distribution. Since the number of videos that are cached depends on the cache size, the performance of MPV in terms of cache hit ratio can be high if implemented for large caches possible for Internet CDNs. However, because of the limited size of the RAN micro-caches proposed in this paper, and because videos requested by active users of a cell may be very different from nation-wide most popular videos, the cache hit ratio achieved by MPV policy may not be high when used for RAN micro-caches.

B. LRU

LRU [15] is a reactive caching policy, which fetches the video from the Internet CDN and caches it if there is a cache miss. If the cache is full, LRU replaces the video in the cache that has been least recently used. The cache hit ratio of a micro-cache associated with a cell that uses LRU policy depends on the overlap of the video requests of the active users in the cell, and influenced by the degree of overlap of their UPP. The backhaul bandwidth and delay needed to bring videos to the cache will depend on the cache hit ratio, since there is no pre-fetching bandwidth.

C. R-UPP

We propose R-UPP as a reactive caching policy based on the UPPs of the active users in a cell. For a video requested that is not present in the cache, R-UPP fetches the video from the Internet CDN and caches it. If the cache is full, R-UPP replaces videos in the cache depending on the UPP of the active users using LLR set introduced in section II.B, and in case of ties according to the LRU replacement policy. More specifically, when there is a cache miss, we calculate the request probabilities of the videos in the cache as well as the requested video, forming a LLR subset. We replace the least likely requested video of the cache with the requested video only if the newly requested video is not the one with least P_R. If there are multiple videos that have the same min P_R value in the LLR (|LLR|>1), we use the LRU policy to select the one to be replaced. The above approach ensures that the cached videos have the highest probability of being requested again by the current active users of the cell. The proposed R-UPP algorithm is shown below.

```
R-UPP
For new Video Request V
If V ϵ Cache
  Download from Cache
Else
  Find cell site UPP based on AUS
  Calculate $P_R$ for V and the cached videos and generate LLR subset
  If |LLR| >1,
    LLR = LRU(LLR)
  End
  If LLR==V
    Do not cache V
  Else
    Cache = Cache + V – {LLR}
  End, End
```

D. P-UPP

We propose P-UPP as a proactive caching policy, which preloads the cache with videos that are most likely to be requested, based on the UPP of the active users of the cell. At the beginning, and every time the AUS changes due to user arrival or departure, video request probabilities are calculated using (3), and videos belonging to the Most Likely Requested set, MLR, are loaded in the cache. However, if the AUS changes frequently, this proactive policy may lead to high computational complexity, and more importantly, high backhaul bandwidth. Hence, we propose a hybrid solution where the cache is only updated if the expected cache hit ratio improvement due to replacement exceeds a preset threshold. More specifically, for each video *i* from the MLR set to be added to the cache, we calculate the difference between its request probability and the request probability of the subset of LLR videos from the cache with least $P_R$values that need to be evicted to free up space for the new video; only if the difference is greater than a threshold, we effectuate the cache update. The proposed P-UPP algorithm is shown below.

```
P-UPP
If AUS changed
   Find cell site UPP based on AUS
   Calculate request probability $P_R$ based on cell site UPP
   Calculate MLR and LLR sets based on cell site UPP
   for each video $i$ in sorted list of MLR set, $MLR_i$
      $LLR_j$: subset of LLR videos with least $P_R$ that has to be evicted from cache to fit $MLR_i$
      if $P_R(MLR_i) - \sum P_R(LLR_j) > Threshold$
         Update the cache with $MLR_i$ and evict $LLR_j$; update MLR and LLR;
      End, End, End
For Video Request V:
 If V ϵ {Cache}
    Download V from Cache
Else
   Download from Backhaul
End
```

While we expect the UPP based cache policies, R-UPP and P-UPP, to result in higher cache hit ratios than conventional MPV and LRU policies, still all videos not found in the cell cache need to be brought from the Internet CDNs, traversing through the core and backhaul network. In the next section, we will discuss the implication on video delay, and hence video

QoE. We propose a scheduling approach that coordinates with requesting video clients and allocates backhaul resources in a way that increases the overall capacity of the system.

IV. Scheduling Approach For Delay and capacity

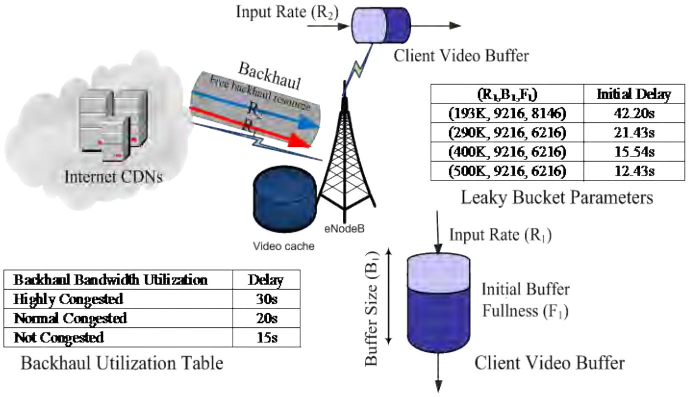
For each video request that results in a cache miss, the corresponding video needs to be fetched from an Internet CDN. For proactive policies, MPV and P-UPP, bringing the missed videos is in addition to the videos that need to be fetched proactively to the cache. Depending on the number of concurrent video requests, the number of cache misses, and the number of proactively fetched videos and frequency of pre-fetching, the backhaul bandwidth may not be sufficient for all the videos that need to be brought through the backhaul. There can be various possible ways of scheduling the video fetches and allocating the backhaul bandwidth. One approach is to satisfy all the pending fetches, which may result in some fetches getting significantly delayed, resulting in unacceptable video playback delay. In this paper, we take an alternative approach. Our proposed scheduling approach aims to maximize the number of videos that can be served, while ensuring each served video meets certain QoE requirements, including initial delay. In this section, we first define video QoE requirements and capacity. Next, we describe our proposed scheduling approach.

A. Video QoE and Capacity

We consider *video QoE* as consisting of two aspects: the initial delay the player has to wait before it can start playing, and the number of stalls during the video session. The initial delay is needed to fill the client buffer to a certain level so to absorb any variations in the network's data transmission rate, and the decoding process can proceed smoothly without any stalls once playback has started.

In this paper, we use Leaky Bucket Parameters (LBPs) to determine the initial delay requirement. In most video coding standards [16][17], a compliant bit stream must be decoded by a HRD (Hypothetical Reference Decoder) connected to the output of the encoder emulating a decode buffer, a decoder and a display unit. The HRD [16] generates LBPs that consist of *N* 3-tupples (R, B, F) corresponding to *N* sets of transmission rates and buffer size parameters for a given bit stream. An LBP tuple guarantees that as long as the average transmission rate is maintained at R bits/second, the client has a buffer size of B bits, and the buffer is initially filled with F bits before video playback starts, the video session will proceed without any stalling. Consequently, F/R is the initial delay that the decoder needs to wait to guarantee a stall free playback. Fig. 2 shows example LBPs associated with a video client buffer, and the resulting initial delays. For example, if the transmission rate is 400Kbps, the initial delay is 15.54 seconds.

A video client, at the beginning of a video session can use the LBPs for the requested video to request a data rate, and select the corresponding initial delay. As shown in Fig. 2, the higher the data rate requested, the less the initial delay. However, if all the video clients greedily select the highest data rates, there may be more congestion in the RAN backhaul, leading to fewer requests that can be served. Consequently, we RAN backhaul bandwidth constraint, and a distribution of video requests, we define *capacity* as the number of concurrent requests that can be served while meeting each request's QoE requirement (maximum acceptable initial delay, and no stalling). Our scheduling approach is to maximize capacity by allocating to each requesting video client the lowest valid LBP bit rate that satisfies its maximum acceptable initial delay, and hence also ensuring no stalling during the video session.



(R_1,B_1,F_1)	Initial Delay
(193K, 9216, 8146)	42.20s
(290K, 9216, 6216)	21.43s
(400K, 9216, 6216)	15.54s
(500K, 9216, 6216)	12.43s

Backhaul Bandwidth Utilization	Delay
Highly Congested	30s
Normal Congested	20s
Not Congested	15s

Figure 2: Example scheduling scenarios: LBPs for video requests, Backhaul congestion states, and Initial Delays

B. Backhaul Scheduling Approach

The goal of our backhaul scheduling approach is to support as many concurrent videos served as possible while ensuring initial delay below an acceptable threshold. We define three backhaul utilization states: not congested, normal congested, and highly congested, associating a maximum delay to each of these states. At any given time, the backhaul is in one of the above utilization states, depending on the videos that need to be fetched through the backhaul (including videos that need to be pre-fetched). Fig. 2 shows an example of backhaul utilization states and associated delays. For a video request which results in a cache miss, depending on the current utilization state, the scheduler sends the corresponding maximum delay as part of the initial handshaking to the client. From the LBPs available, the client selects a transmission bit rate R that results in an initial delay F/R right below the backhaul delay threshold, and communicates it back to the scheduler. Subsequently, the scheduler allocates the RAN backhaul resource at the transmission rate selected by the client, only if enough space bandwidth is available. For example, consider a scenario where the backhaul is in normal utilization state and a video request needs to be served which has an acceptable initial delay threshold of 25 seconds. According to the example in Fig. 2, the scheduler asks the requesting video client to select a data rate that corresponds to a delay less than 20 seconds. According to the client's LBP table, the video client will select the date rate of 400 kbps, which will yield an initial delay of 15.54 seconds. The 400kpbs is also the backhaul data rate that will be allocated by the scheduler to this client if enough backhaul bandwidth is available. Note different video requests will have different LBPs and hence the data rate selected will be different. If the scheduler cannot allocate the requested bandwidth, the request will be blocked.

Note that the LBP should span useful delay/rate pairs which are the delays that the scheduler could be interested in achieving for the initial delay. Intermediate values not directly available in the table may be derived using interpolation of existing table values.

In the next section, we report on experimental studies we have performed using a simulation framework to assess the performance of the caching policies described in section III, and the RAN backhaul scheduling approach outlined in this section.

V. Simulation Framework and Results

A statistical simulation framework was developed using MATLAB to compare the relative performance of the caching policies. We use Monte Carlo simulation, where the implementation consists of a number of iterations where the innermost loop corresponds to one video request per iteration which is being evaluated for all the cache policies. There is an outer loop over a set of different cache sizes, and finally the outermost loop repeats the entire simulation using a new set of inputs for increased statistical significance. Next, we explain our simulation parameters and present the results.

Table 1 lists the parameters used for our simulation results. We discuss briefly some aspects of the parameters used, and then proceed to report the results. Though the results are based on a Zipf distribution with parameter 0.8 (to model video distribution according to [12]), our experiments with other Zipf parameter values confirm the trends and conclusions reported here. Similarly, to ensure simulation speed, though we restrict the total number of videos available for request to 20,000, and the total number of mobile users to 5000, we expect the trends reported here to hold for higher values. User arrival and departure follow a Poisson process, and we use an M/M/ ∞ queuing model [18] to find the total number of concurrent active users. To generate a video request, a user is selected randomly from the AUS, and a video request is generated based on the user's UPP and the popularity ranking of videos. For the results reported below, we assume a backhaul bandwidth of 100Mbps, and the micro-cache size varies between 50 to 400Gbits.

Fig. 3(a) shows the performance of the different cache policies in terms of cache hit ratio achieved for a given cell, for different cache sizes. It is evident that the UPP-based cache policies perform significantly better than the conventional cache policies for all cache sizes. For example, when the cache size is 250Gbit, P-UPP and R-UPP achieve cache hit ratios of 0.67 and 0.65 respectively, compared to the LRU and MPV policies achieving cache hit ratios of 0.50 and 0.25 respectively.

A Cache hit ratio of 0.75 is achieved by P-UPP when the cache size is 400Gbit. Note that though Fig. 3(a) shows P-UPP and R-UPP achieving similar cache hit ratios, from other simulation results not presented in this paper due to space limitation, with different parameters like lower P-UPP update threshold (Section III.D), we observe that P-UPP can perform up to 10 percentage point better than R-UPP in terms of cache hit.

Fig. 3(b) shows the mean RAN backhaul bandwidth required by the different policies. For example, with cache size of 250Gbits, we require 62Mbps backhaul bandwidth for R-UPP, 81Mbps for P-UPP, 79Mbps for LRU, and around 94Mbps for MPV cache policy. Note that if there was no video caching at the edge of the RAN (no cache in Fig. 3(b)), the backhaul bandwidth needed to bring all the requested videos would be 98Mbps.

Table 1: Simulation Parameters

Variable	Distribution/Parameters Value
Total Number of Video Requests	Const, 100,000
Total Number of Videos	Const, 20,000
Video Popularity Distribution	Zipf, 0.8
Video Frame Size Distribution	As proposed in [19]
Total Number of Video Categories	Const, 250
Video Size	Exp, min=2, mean=8, max=30(minutes)
Video Bit Rate	Uniform, 200kbps (QVGA), 2Mbps (HD)
Total number of mobile users	Const, 5,000
UPP Distribution Across VCs	Exponential, mean=2
User Arrival/Departure Model	Poisson: User inter-arrival time=100s, User active time= 2800s
Video Request Arrival	Poisson, Mean inter-arrival time per user = 120s
Backhaul Delay Thresholds	Const, [10,20,30] seconds
Max Cache Size	Const, Up to 400Gbits
Backhaul Bandwidth	Const, 100 Mbps

Fig. 3(c) shows the blocking probability (probability that requested videos could not be scheduled) when the cache size varies from 50 to 400Gbits. An ideal system should achieve a low blocking probability while satisfying the desired initial video delay, here 30 seconds, for all users. For cache size of 250Gbits, R-UPP and P-UPP achieve blocking probabilities of less than 0.0001 and 0.004 respectively, while the blocking probability for LRU is around 0.02 and 0.15 for MPV. While the previous results demonstrate the superiority of the UPP-based cache policies in terms of cache hit ratios, and reduced backhaul traffic overhead and thereby higher chances of being successfully served, we next study their performances in terms of the initial delay needed by the scheduled videos, a key contributor to QoE. Specifically, Fig. 3(d) shows the probability that the delay of a successfully scheduled video is below a certain value when the cache size is 200GBits, and also when no RAN cache is used. For example, the probability of achieving an initial delay of 5 second or less is about 0.56 when no RAN cache is available, 0.66 for MPV, 0.74 for LRU, 0.80 for R-UPP, and 0.82 for P-UPP. Fig. 3(d) clearly shows that using micro-caching at the RAN significantly improves the probability of video requests that can meet initial delay requirements, in particular when the desired initial delay is low. The results also show the superiority of the UPP based policies, compared to MPV and LRU policies, in achieving better initial delay.

To better understand the impact of RAN caching and our proposed policies on the capacity of the wireless network, we performed a set of experiments to measure the capacity for different cache sizes and initial buffering delays. Fig. 3(e) shows capacity vs. cache size; note that each point in this graph captures the case where the blocking probability is exactly 0.01, which is achieved by changing the user inter-arrival rate such that the steady state target blocking rate is achieved and noting the number of concurrent video requests generated at that specific user inter-arrival rate. For cache size of 150Gbits, the capacity is 84 concurrent videos served in the cell without RAN caching, 113 with MPV, 151 with LRU, 175 with P-UPP, and 206 with R-UPP. We can infer from Fig. 3(e) that R-UPP performs about 18% better than P-UPP, about 36% better than LRU, about 82% better than MPV, and 145% better than when there is no RAN caching. The superiority of the R-UPP in terms of capacity is due to the high cache hit ratio that it can achieve

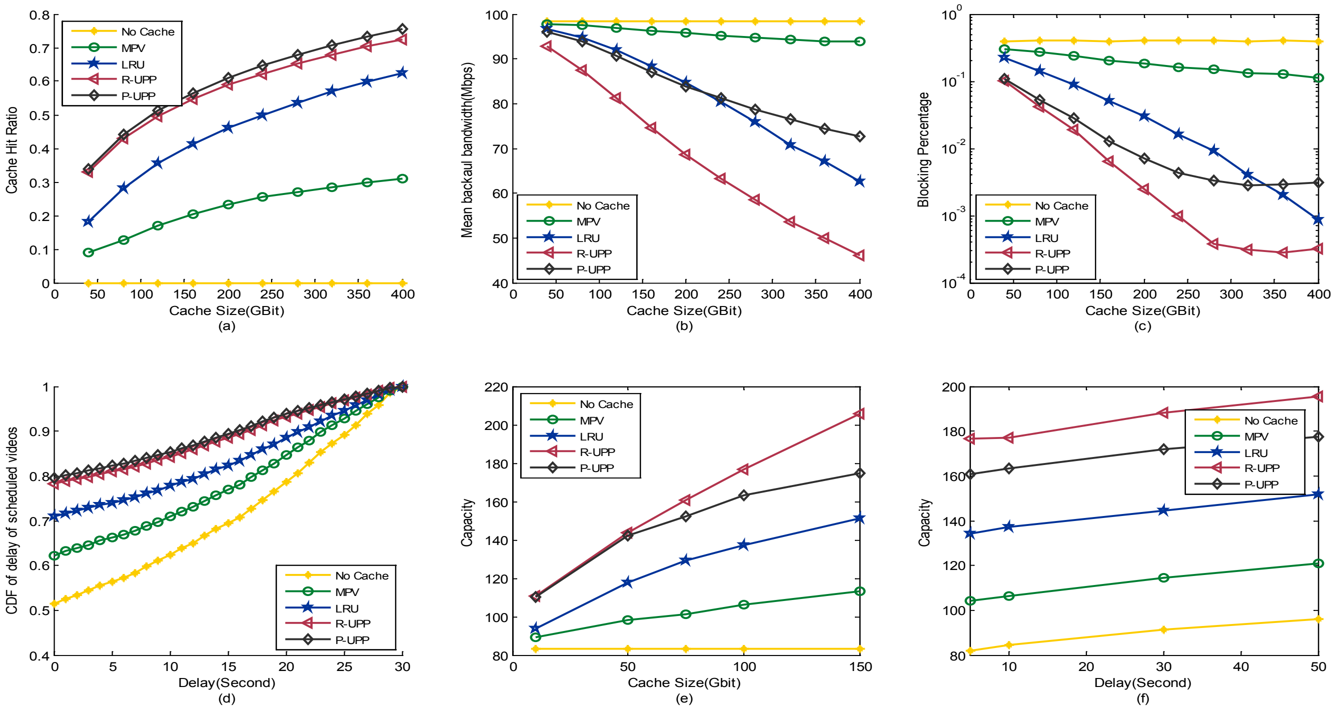


Figure 3: Performance of the caching policies (a) Cache Hit Ratio vs. Cache Size (b) Mean Backhaul BW Required vs. Cache Size (c) Blocking Probability vs. Cache Size (d) CDN of the Delay of Scheduled Video Requests when Cache Size=200Gbits (e) Capacity vs. Cache Size (f) Capacity vs. Delay when cache size=100Gbits, blocking probability=0.01

(comparable to P-UPP) while having no overhead (like proactively filling the cache as done by P-UPP and MPV).

Fig. 3(f) shows the capacity when we change the target delay of the not-congested region (Section IV.B). As the target delay increases, the capacity increases for all caches. For delay of 30 seconds, the total number of concurrent videos served without cache is 91, with MPV 114, with LRU 144, with P-UPP around 172, and with R-UPP 188. We can infer from the figure that R-UPP performs about 10% better than P-UPP, about 30% better than LRU, and more than 100% better than when no caching is performed at the RAN edge.

VI. Conclusion

In this paper, we demonstrated the feasibility and effectiveness of using micro-caches at the edge of the RAN, coupled with new caching policies based on video preference of users in the cell and a new scheduling technique that allocates RAN backhaul bandwidth in coordination with requesting video clients. Our simulation results show that the new RAN micro-caching based video delivery approach can significantly increase the number of concurrent video requests that can be served while meeting initial delay requirements. In the future, we plan to extend our approach to consider mobility of users across cells. We also plan to expand our approach to consider bandwidth constraints in the RAN RF links.

Acknowledgements: This research was supported by Cisco Systems, Inc, by Intel Corp. and by the UC Discovery Grant program.

References

[1] White Paper,"Cisco Visual Networking Index: Global Mobile Data", 2010-2015.

[2] G. Pallis, A. Vakali,"Insight and perspectives for content delivery networks", Communications of the ACM, vol. 49, issue 1, January 2006.

[3] M. Pathan, R. Buyya,"A Taxonomy of CDNs, Content Delivery Networks", Springer-Verlag, Germany, 2008.

[4] S. Sen et. al,"Proxy prefix caching for multimedia streams", in Proc. of IEEE INFOCOM, March 1999.

[5] A. Wierzbicki,"Internet Cache Location and Design of Content Delivery Networks", In Web Engineering and Peer-to-Peer Computing, Lecture Notes in Computer Science, vol. 2376/2010, 2002.

[6] A. Balamash and M. Krunz,"An Overview of Web Caching Replacement Algorithms", IEEE Commun. Surveys and Tutorials, vol. 6, no. 2, 2004.

[7] Mitch Cherniack, et al., "Profile-Driven Cache Management", In Proceedings of ICDE'2003.

[8] J. Z. Wang, et al.,"Network Cache Model for Wireless Proxy Caching", In Proceedings of the 13th IEEE International Symposium on Modeling, MASCOTS, Sept 2005.

[9] Hui Chen and Yang Xiao,"Cache Access and Replacement for Future Wireless Internet", IEEE Communications Magazine, May 2006.

[10] W.H.O. Lau, et al.,"Cooperative cache architecture in support of caching multimedia objects in MANETs", In Proceedings of International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM, 2002.

[11] N. Wakamiya, et al.,"Video Streaming Systems with Cooperative Caching Mechanisms", in Proceedings of SPIE International Symposium, 2002.

[12] M. Cha et. al.,"Analyzing the Video Popularity Characteristics of Large-Scale User Generated Content Systems", IEEE/ACM Transactions on Networking, Vol. 17, No. 5, Oct. 2009.

[13] Michael Zink, et al.,"Watch Global Cache Local: YouTube Network Traces at a Campus Network - Measurements and Implications." In Proceedings of MMCN 2008, San Jose, CA, USA, Jan 2008.

[14] Reelseo. Available: http://www.reelseo.com/most-popular-video-sites-categories/

[15] N. Laoutaris,"A Closed-Form Method for LRU Replacement under Generalized Power-Law Demand", presented at CoRR, 2007.

[16] J. Ribas-Corbera, et al.,"A Generalized Hypothetical Reference Decoder for H.264/AVC", IEEE Transactions on Circuits and Systems, vol. 13, no. 7, July 2003.

[17] JM Codec. [Online]. Available: http://iphome.hhi.de/suehring/tml/

[18] R. Gallager and Bertsekar, "Data Networks", Prentice Hall, 1992.

[19] D. M. B. Masi, et al.,"Video Frame Size Distribution Analysis", The Telecommunications Review 2008, Volume 19, Sept 2008.