Hierarchical Video Caching in Wireless Cloud: Approaches and Algorithms

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

***Abstract*— We had introduced video caching techniques in the Radio Access Network (RAN) in [1] as a way to reduce the need to bring requested videos from Internet CDNs, thus reducing overall backhaul traffic, improving video quality of experience and increasing network capacity to support more simultaneous video requests. In this paper, we investigate supplementing the resulting wireless cloud with a hierarchical caching scheme, where the gateways in the Core Network (CN) also have video caches. The hierarchical caching approach further improves network capacity by enabling multiple cell sites to share caches at higher levels of the hierarchy, thereby improving overall cache hit ratio, without increasing the total cache size used. In addition, we exploit hierarchical caching to better accommodate mobility, so that when a user with an active video session moves from one cell to a neighboring cell, it is likely that the video currently being downloaded is already in a cache within the RAN or CN network associated with the new cell. To achieve the goal of improving capacity and supporting mobility, we extend our User Preference Profile (UPP) based caching policies [1] to accommodate the hierarchical caching structure introduced in this paper. For all the videos that miss the cache in any layer of hierarchy, we propose a scheduling approach to allocate RAN and CN backhaul resources judiciously so as to maximize the capacity of the wireless network. We extend our discrete event statistical simulation framework developed in [1] to study the performance of the proposed hierarchical caching approach. Our simulation results show that using hierarchical caching can enhance cache hit ratio by 24% and network capacity by up to 45% compared to caching only in the RAN. Significant capacity gains are also observed when additionally considering user mobility.**

***Index Terms*—Hierarchical Caching, Wireless Radio Access and Core Network, Wireless Network Capacity**

I. Introduction

With the rapid growth in the number of smart phone and tablet users year over year, it is expected that the global mobile data traffic will grow by 92 percent annually from 2010 to 2015, of which up to two thirds is expected to be video [2]. While Content Delivery Networks (CDNs) have been recently enhanced to reduce Internet bandwidth consumption and associated delay/jitter of online video, such video consumed by mobile devices must additionally travel through the wireless carrier Core Network (CN) and Radio Access Network (RAN) before reaching the User Equipment (UE). To facilitate this tremendous growth in mobile video consumption without risking running out of wireless network capacity and the associated problems of congestion and delay, we recently introduced a video aware wireless cloud, where the base stations in the RAN have video caches, with caching policies which are aware of the video preferences of users in cell sites [1]. We demonstrated that the proposed RAN caching techniques can significantly increase network capacity while reducing video latency and thereby improving user experience.

In this paper, we enhance the wireless video cloud, further distributing the RAN caches to include network elements within the CN, resulting in a hierarchical video caching structure, but without increasing the total cache size used. Adding caches within the CN can supplement RAN caches, enable multiple cell sites share caches at higher levels of the cache hierarchy, and help eliminate bandwidth bottlenecks between the UE and CDN. The result can be improved overall cache hit ratio, and increased network capacity to support simultaneous video requests.

Additionally, the proposed hierarchical caching approach can be beneficial to support for mobility, which is a challenge for the RAN caching. When a user moves from one cell to another cell, the associated RAN cache of the new cell may not have the video, leading to a cache miss and the video to be downloaded from the Internet CDNs, resulting in increased latency and reduced capacity. However, in hierarchical caching, proper caching of the video at the CN caches can help provide seamless mobility.

In this paper, we extend our User Preference Profile (UPP)-based caching policies introduced in [1] to support hierarchical caching. Our policies also implicitly anticipate mobility and prepare for the eventuality that the video downloads have to be migrated to a neighboring cell cache. As with RAN caching, even with hierarchical caching, some cache misses are inevitable in each layer of the hierarchy, and each video download needs to go through the backhaul of all network elements in the hierarchy up to the level where the video is found, so some backhaul traffic must be scheduled throughout the network. We propose a scheduling approach that improves the total number of concurrently admitted videos while maintaining the user's required Quality of Experience (QoE) by first scheduling the videos based on the video codec's Leaky Bucket Parameters (LBP) [3] and assigning any spare backhaul bandwidth using Linear Programming (LP) optimization. Our simulation results show that the proposed hierarchical caching approach, together with the scheduling technique, can improve the capacity of the wireless network significantly over the results that we presented in [1].

A. Related Work and Paper Outline

Significant amount of work has been done to develop CDNs for the Internet [4][5]. As explained earlier, Internet CDNs, and caching at Internet CDNs, do not address the problems of latency and capacity for video delivery in wireless networks. Recently, traffic off-loading to available Wi-Fi networks or Femto cells, and providing incentives for peer-to-peer sharing, have been proposed as ways to improve video QoE and capacity in wireless networks [6][7]. However, the above

978-1-4577-2053-6/12/$31.00 ©2012 IEEE

techniques do not utilize the power of video caching, which has traditionally proved to be very beneficial for Internet CDNs. In [1], we proposed an approach to move video caching to the edge of the wireless networks – at (e)NodeBs in RAN. Since conventional CDN video caching techniques are not as effective for smaller sized RAN caches, we proposed new caching policies which are sensitized to user preferences in corresponding cells. We demonstrated that RAN caching, together with the proposed RAN backhaul scheduling technique, can significantly reduce congestion, and improve the capacity of wireless networks to support concurrent video requests while satisfying desired QoE. In this paper, we propose a hierarchical caching approach, extending the RAN caching approach in [1] to include caching also in CN elements.

The remainder of the paper is organized as follows. In section II, we first review the wireless system architecture and justify the applicability of the "cache hierarchy tree" that we propose in this paper as a means to further improve the capacity and provide mobility support. In section III, we first introduce the hierarchical caching approaches and later explain our hierarchical version of the caching policies that we introduced in [1]. Subsequently, in section IV, we introduce our video scheduling approach. Section V outlines our simulation framework and experimental results demonstrating the superior cache hit ratio and system capacity that can be achieved using hierarchical caching vs. RAN-only caching and scheduling techniques. We conclude the paper in section VI.

II. Wireless Network Abstraction

In this section, we first briefly review the system architecture of 3G and 4G wireless networks. Subsequently, we introduce hierarchical caching at different nodes in the wireless network and abstract the wireless network with caches by using a tree topology.

A. Wireless Network Architecture Overview

In the previous 3G wireless standards, e.g. 1xEV-DO and UMTS, only limited radio functionality was placed in the NodeB and the Radio Network Controller (RNC) was responsible for resource management, controlling the NodeBs, as well as session or connection setup. Every soft or hard handover needed to go through the RNC. In such an architecture, the requests first traversed through the NodeB to the RNC and then from the RNC to the SGSN and GGSN and would follow the same path in the reverse direction to the UE. No inter-NodeB communication was in place, and the network was circuit-switched oriented. A NodeB was homed to an RNC and RNC was connected to a SGSN and so on. Fig. 1(a) shows an overview of the 3G architecture along with our proposed caches in NodeBs, RNCs, and GGSNs. We do not do any caching at the SGSNs.

In 3GPP Long Term Evolution (LTE) and System Architecture Evolution (SAE) wireless standard, the main data path is from the Packet Data Gateway (PGW) to Service Gateway (SGW) to eNodeB – i.e. a top down flow, although control and minimal data transactions can be done between nodes within the same level (e.g. between two eNodeBs or two SGWs). From 3GPP release 6 to release 8 the functionality of the RNC has been consolidated into the eNodeB containing all the network-side radio functionality. SAE was developed with the goal to accommodate the high capacity LTE radio interface, optimize for packet-switched operation, improve the experienced delay and support the higher user throughput provided by the physical layer, along with inter-operability with the other 3GPP and wireless standards [8][9].

In 4G, eNodeBs can be inter-connected over the X2 interface, a high capacity interface designed in SAE for transferring control information or UE's data buffer during handover; here no RNC is used. SAE supports handovers at the eNodeB level over the X2 or S1 interface. Although, this X2 interface is available for limited data transfer, it cannot be used for the long term data transfer between two eNodeBs, so it cannot be exploited for inter-cache data transfer; for this reason, we assume that nodes at the eNodeB layer cannot share their cache contents directly. Fig. 1(a) shows a high-level system architecture for 4G along with our proposed caches located at each eNodeB, SGW, and PGW. Mobility Management Entity (MME) keeps track of UE locations in its service area and once the UE first registers in the network, it allocates resource in the eNodeB and SGW for the UE. The SGW is responsible for relaying the data between eNodeB and PGW. A set of MMEs and SGWs are assigned to serve a particular set of eNodeBs. An eNodeB may connect to many MMEs and SGWs, for instance if there is congestion or one of the elements cannot be reached because the route is not available. However, each UE will be served by only one MME and SGW at a time. Because in the normal operation one eNodeB is connected to one MME and SGW, without loss of generality, we simplify our caching structure to a tree based hierarchy similar to the one we propose for 3G. In the next sub-section, we discuss our tree topology.

B. Tree Structure Abstraction and Video Request Flow

We construct a network that has a tree topology to model data flow in a 3G or 4G network. In this tree structure, leaf nodes (1^{st} layer nodes) are (e)NodeBs where users attach. 2^{nd} layer nodes are the RNCs/SWGs, which do not have users directly connected to them, but cover a group of (e)NodeBs and their associated users. Similarly, 3^{rd} layer nodes are GGSN/PGW with RNC/SGWs attached. In this paper we limit ourselves to a single GGSN/PGW (3^{rd} layer node), which forms the root node of the tree and is connected to the Internet CDN via the Internet backhaul. The caches associated with the 1^{st}, 2^{nd} and 3^{rd} layer nodes are referred to as the 1^{st}, 2^{nd} and 3^{rd} layer caches respectively. A video may be present at the 1^{st}, 2^{nd} or 3^{rd} layer caches, but is *guaranteed* to be found in the Internet CDN connected to the backhaul of the 3^{rd} layer node. Fig. 1(c) shows an example tree architecture where nodes of the tree represent the caches and edges represent the backhaul links with bandwidth, C_i, which sets the upper bound on the total number of concurrent video downloads possible on the link, which we will discuss further in section IV.

Regardless of the caching policy used with this hierarchical model, if a user requests a video, and the video is found in the lower layer cache, the video is delivered from that cache and the backhaul connecting to a higher layer node is kept available for other downloads. If the request results in a cache miss in the lower layer, then the request goes to the higher

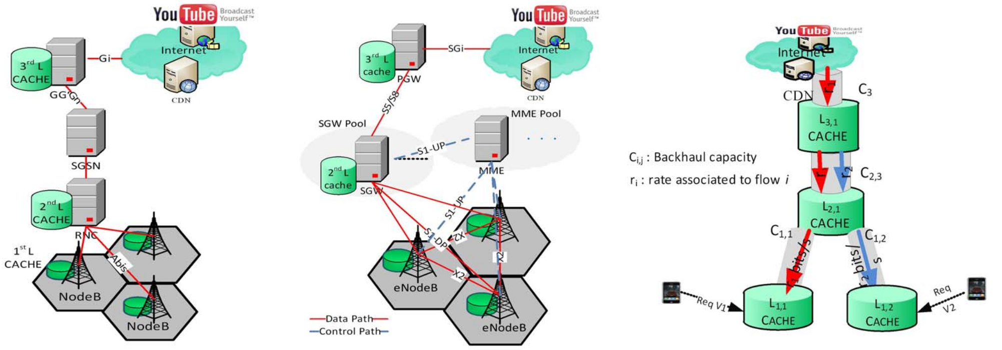

Fig. 1. Wireless system architecture and associated caches, (a) 3G Architecture (b) 4G Architecture (c) Hierarchical Cache Model

layer cache only if there is enough backhaul bandwidth to accommodate the video download. Consequently, the finding of a video in a lower layer cache results in lower latency and higher capacity. More importantly, finding the video within the wireless cloud helps us to lower the traffic in the backhaul connecting the wireless cloud to the CDN (Internet backhaul). In the next section, we explain the hierarchical caching policies that achieve this.

III. Cell Site Aware Hierarchical Caching Policies

In this section, we first explain the overall hierarchical cache approach to improve the video cache hit ratio within the wireless cloud as well as to provide support for mobility. We next describe the design of our hierarchical UPP based caching policies, including the required modifications to the UPP based caching policies proposed in [1].

A. Hierarchical Caching Policies

An important property of hierarchical caching relates to the amount of video redundancy in the caches of different layers of the hierarchy, impacting cache hit ratio of users in a cell, as well as providing support for mobility between cells. Different cache hierarchy architectures have been proposed in the literature [10]; such as Inclusive Cache Hierarchy where a higher layer cache includes all of the videos that exist in the associated lower layer caches conditioned upon the storage availability of the higher layer cache, or Exclusive Cache Hierarchy where a higher layer cache will not cache videos that are present in the associated lower layer caches. Inclusive cache hierarchy can be very effective to support mobility between cells, whereas exclusive cache hierarchy can be more effective to improve cache hit ratio of more static users. In an inclusive cache hierarchy, a user traveling from cell site A to cell site B while receiving a video stream, can continue to receive the video almost seamlessly during the hand-off, even if the video that is being watched is not found in the cell site B cache, because the video can be found in the 2^{nd} layer cache that is connected to both A and B. In an exclusive cache hierarchy, where the 2^{nd} layer supplements the 1^{st} layer cache, the cache hit ratios of users in cells A and B will be improved, as long as the mobility of users in A and B is low. One problem with such caching schemes is that they require extensive cache coordination; maintaining coherency between caches may result in high levels of overhead.

In this paper, we propose a hybrid and partially distributed hierarchical caching policy to increase cache hit ratio and provide support for high mobility. In this approach, each layer independently caches the video contents according to its caching policy and the only coordination required is that each node relays its Active User Set (AUS) information [1], i.e. the active video users served by the node, whenever AUS is updated, to its higher layer node in the cache hierarchy tree. The AUS of the higher layer node is then defined as the union of all AUS of the 1^{st} layer nodes connected to it. To further optimize for mobility, in this paper we use the implied inclusivity of our hierarchical caching algorithms to improve support for mobility. In other words, we do not perform Exclusive Cache Hierarchy as explained earlier in this sub-section.

In our proposed hierarchical UPP-based algorithm, each cache in the cache hierarchy makes its decision independently, and as a result, videos may be redundantly cached at multiple layers in the cache hierarchy. This is inconsequential if the assumption is that the cache size grows by a significant factor for each layer as we get closer to the root node because the redundant part of the cache will only be a small part of the total cache. However, if the cache sizes are limited it becomes more important to conserve space and an exclusive policy may be needed. To further optimize for storage, it is possible to remove the intersection of all the 1^{st} layer caches, from the 2^{nd} layer and 3^{rd} layer caches, and so on.

We modified the caching policies studied in [1] to support hierarchical caching described later in this section: Most Popular Videos (MPV), Least Recently Used (LRU), Reactive-User Preference Profile (R-UPP), and Proactive-User Preference Profile (P-UPP). In the RAN-only model – where only (e)NodeBs have caches – P-UPP and R-UPP cache candidates are calculated based on the AUS of the individual (e)NodeBs. The reactive caching policies, i.e. LRU or R-UPP, fetch the video from the video source if there is a cache miss and cache it if the conditions of the replacement policy are met. If there is a cache miss in the 1^{st} layer cache, the request propagates to the 2^{nd} layer cache, up the tree until there is a cache hit or it reached the root node of the hierarchy. Subsequently, the video is fetched, and while traversing down the tree hierarchy, each cache in the hierarchy chooses whether to cache the content based on its cache replacement policy. In the next section, we first briefly discuss changes required for the traditionally used MPV and LRU caching policies to support the proposed hierarchical caching approach. Next we describe in details the new hierarchical R-UPP and P-UPP caching algorithms.

B. Cache Policies within the cache hierarchy

1) Hierarchical MPV: MPV is a proactive caching policy, which caches the "most popular videos" using the (nation-wide) video popularity distribution [11]. In Hierarchical MPV, each layer in the cache hierarchy caches the same "most popular videos" to the degree the cache size permits.

2) Hierarchical LRU: LRU [10] is a reactive caching policy that caches contents as they are being fetched from the backhaul following a cache miss. If the cache is full, LRU replaces the video in the cache that has been used least recently. Hierarchical LRU is a straight-forward extension of the single-layer LRU, but it is noteworthy that this scheme has a built-in exclusivity mechanism. Consider a video request that occurs frequently at all (e)NodeBs associated with a 2^{nd} layer cache: Initially the video is being fetched by a user at a single (e)NodeB which results in the video being cached at that (e)NodeB and in the 2^{nd} layer cache. When a user at another (e)NodeB requests the same video, it is delivered directly from the 2^{nd} layer cache and stored in that new (e)NodeB's cache as well. Eventually the video will be stored at all the (e)NodeBs and all future user requests for that video will be served from the 1^{st} layer caches, i.e. the 2^{nd} layer cache will no longer see any requests for that video and it will eventually be evicted by the LRU policy at the 2^{nd} layer to free up the space for other videos.

3) Hierarchical R-UPP: R-UPP is a reactive cache policy which replaces the videos based on the UPPs of the active users in a cell [1]. Upon a cache miss, R-UPP fetches the video from the backhaul and caches it if the UPP of the AUS indicates it is more likely to be requested again than any video currently cached. When applying R-UPP to hierarchical caching, similarly to LRU, if the request to the 1^{st} layer cache is a cache miss, the request is progressively passed to the next layer in the cache hierarchy tree until either there is a cache hit or it has reached the root of the tree meaning the video needs to be fetched from the Internet CDN. While the fetched video is traversing towards the UE in the hierarchy tree, each cache on the way to the 1^{st} layer cache decides whether to cache the video. The replacement policy for this algorithm has been explained in [1]: After each new video request, we calculate the request probability, P_R, of the videos in the cache as well as that of the newly requested video. Using these probabilities we form the Least Likely Request (LLR) set, which is the smallest set of videos that need to be evicted to fit in the newly requested video and may consist one or multiple cache entries depending on the size of the requested video. Then we replace the LLR set with the requested video only if the P_R of the new video is higher than the aggregate P_R of the LLR. The details of the hierarchical R-UPP caching algorithm are shown below: $AUS(L_{i,j})$ represents the AUS that is associated with the j^{th} cache in the i^{th} layer. $UPP(AUS(L_{i,j}))$ is the aggregate UPP of the AUSs associated with the cache $L_{i,j}$. Based on our definition of the cache tree structure, each cache in the 1^{st} layer is associated with one cache in the 2^{nd} layer and one cache in the 3^{rd} layer. We use $L_{i,L_{1,j}}$ to refer to the i^{th} layer cache that is associated with the j^{th} 1^{st} layer cache. In the cellular network, there are three layers of cache ($n = 3$) as explained in section II.A.

```
Hierarchical R-UPP
For each new request for Video $V$ to $L_{1,j}$ (cache of the $j^{th}$ eNodeB)
Initialize $\mathbf{Counter}$ to zero
For $i = 1$ to $n$ (each layer in the cache hierarchy)
   If $V \in L_{i,L_{1,j}}$ Schedule download of $V$ from $L_{i,L_{1,j}}$
     $\mathbf{Counter} = i$
End If, End For
If $\mathbf{Counter} == 0$ ($V$ not found in any cache)
  Schedule download of $V$ from the Internet CDN
  $\mathbf{Counter} = n + 1$
End If
If download scheduling successful for $i = \mathbf{Counter} - 1$ down to 1
  If there is space in cache $L_{i,L_{1,j}}$
    Update cache: $L_{i,L_{1,j}} = L_{i,L_{1,j}} + V$
  Else
    Find UPP for cache $L_{i,L_{1,j}}$ based on $AUS(L_{i,L_{1,j}})$
    Calculate $P_R$ for $V$ and the videos in $L_{i,L_{1,j}}$ and generate $LLR_i$
    If $P_R(LLR_i) > P_R(V)$
      Do not cache $V$
    Else
      $L_{i,L_{1,j}} = L_{i,L_{1,j}} + V - LLR_i$
End If, End If, End For, End If
```

4) Hierarchical P-UPP: Hierarchical P-UPP caching algorithm is based on the P-UPP cache policy [1], which pre-loads the cache with the videos that are most likely to be requested given the UPP of the AUS of the associated (e)NodeBs. When the AUS of any of the (e)NodeBs change due to user arrival or departure (including user mobility), video request probabilities are recalculated as proposed in [1], and the cache contents are updated with the videos belonging to the Most Likely Requested set, MLR [1]. MLR is a subset of videos, with the highest aggregate request probability, that fits into the cache. In order to avoid excessive update overhead, each cache replacement can be associated with a probability threshold (T_ϵ), so that the replacement only takes place if there is a significant improvement in request probability. The algorithm for P-UPP is shown below:

```
Hierarchical P-UPP
Cache Update:
If AUS changed for the $i^{th}$ (e)NodeB, $L_{1,i}$
  Find UPP of $L_{1,i}$ and any higher layer cache in the path to $L_{1,i}$
  Calculate $P_R$ based on $UPP(AUS(L_{i,L_{1,j}}))\forall i = 1, .., n$
  Calculate $MLR_{i,L_{1,j}}$ and $LLR_{i,L_{1,j}} \forall i = 1, .., n$
  For each video $k$ in sorted list of $MLR_{i,L_{1,j}}$ set, $MLR_{i,L_{1,j}}(k)$
    $LLR_{i,L_{1,j}}(t)$: subset of LLR videos with least $P_R$ to be
                 evicted from cache to fit $MLR_{i,L_{1,j}}$
    if $P_R(MLR_{i,L_{1,j}}(k)) - \sum P_R(LLR_{i,L_{1,j}}(t)) > T_\epsilon$
      Update the cache with $MLR_{i,L_{1,j}}(k)$ and evict $LLR_{i,L_{1,j}}(t)$;
      Update $MLR_{i,L_{1,j}}$ and $LLR_{i,L_{1,j}}$;
End If, End For, End If
Video Request:
Initialize $\mathbf{Counter}$, $i$ to zero
If new Video Request $V$ to the cache of the $j^{th}$ (e)NodeB, $L_{1,j}$
For i =1 to n (each layer in the cache hierarchy)
If $V \in i^{th}$ Layer Cache associated with cache $L_{1,j}$, $L_{i,L_{1,j}}$
  Download $V$ from $L_{i,L_{1,j}}$
  $\mathbf{Counter} = i$
  Return $V$, $\mathbf{Counter}$
End If, End For
If $V$ not found in any cache, $\mathbf{Counter} > n$
  download $V$ from the Internet CDN
End If
```

Although we showed in [1] that UPP based cache policies, R-UPP and P-UPP, result in higher cache hit ratios than conventional MPV and LRU policies, still all videos not found in the (e)NodeB caches need to be brought from a higher layer cache or from the Internet CDNs, traversing through the CN and RAN backhaul. For all the videos that cause a miss in the RAN caches, including compulsory misses (i.e. the first time a reactive cache accesses a video) and cache maintenance traffic, we propose a scheduling approach that coordinates with the requesting video clients and uses backhaul resources judiciously to increase the overall capacity of the system.

IV. Scheduling Approach For Capacity and Delay

Whenever a video is downloaded from one layer of the hierarchy to the next, the successful scheduling of the download in that layer is conditioned upon the availability of sufficient backhaul bandwidth; otherwise the request will be blocked.

LBP [3] consist of N 3-tupples (R, B, F) corresponding to N sets of transmission rates and buffer size parameters for a given bit stream and are generated based on the video coding structure and allocated channel rate, and are used both in this paper and [1] for allocating the minimum required rate for each video. This rate corresponds to the maximum acceptable initial delay (a QoE parameter) that a user can tolerate, and if it cannot be satisfied because of lack of available bandwidth in the backhaul of any layer of the hierarchy, the request is blocked. In addition, to avoid stalling, the scheduling algorithm needs to ensure that the download rate does not decrease below this minimum rate any time during the transmission; for this reason, the scheduler refrains from admitting new video request if there is not enough spare bandwidth to maintain the minimum required rate of the scheduled requests. Further, once all the requested videos have been scheduled according to the LBPs, there may parts of the network operating at less than 100% capacity for a period of time. This spare capacity can be used to accelerate the ongoing downloads with the intent to finish the downloads faster and free up bandwidth for later use. To utilize the spare capacity, we introduce flow maximization using linear programming. The idea is to think of each download (e.g. between 2^{nd} layer node and eNodeB, or between 3^{rd} and 2^{nd} layer nodes) as being part of a flow that spans the distance between the video source and the end user. For instance in Fig. 1(c), the first video request, V_1, spans all the way from the Internet CDN to the 1^{st} layer cache, and the minimum allocated rate is based on the maximum delay that a user can tolerate and the available bandwidth at each level of the hierarchy. The same rate, R_1, has to be allocated for all the backhauls that V_1 should be downloaded through. The 2^{nd} video request, V_2, results in a cache hit in the 3^{rd} layer so the flow (video download) only spans from 3^{rd} layer to 1^{st} layer. The bandwidth of the i^{th} flow is identified by b_i and subsequently maximized under the constraint that the sum of the bandwidth of all scheduled flows that go through each backhaul must not exceed its capacity limits, C_n, and should be greater than the initially scheduled (minimum) rates, r_i:

$$\begin{aligned} \text{Maximize: } & \sum_i b_i \\ \text{Subject to: } & b_i \geq r_i \forall i \\ & \sum_{i \in F_n} b_i \leq C_n, n = 1, ..., N \end{aligned}$$

This optimization problem is solved for the entire network, so all caches and backhauls are numbered from 1 to N, where N is the total number of nodes in the network. F_n is the set of flows that go through the nth backhaul and r_i is the minimum allocated rate of the i^{th} video request. This optimization is being executed only after all the initial video bandwidth assignments (r_i) were decided based on LBP. Meaning that after any new video request, we first make sure that the new video request can be admitted based on its LBP and minimum required rate of all existing video downloads (rate obtained using LBP), and then we run the scheduling algorithm again to further optimize the rate by using the spare capacity. Unlike the distributed scheduling algorithm that we proposed in [1], with LP reallocation of spare capacity we can relax the minimum rate requirements during peak load periods by asking the users (mobile clients) to recalculate their minimum rate requirements. This can be done because the buffer levels during the download sessions may be higher than anticipated at the time of initial scheduling. Due to space constraints, this method is not explored further in this paper.

V. Simulation Framework and Results

We extended the MATLAB Monte Carlo simulation framework that was developed in [1] to assess the benefits of hierarchical caching and present the results in this section. As explained in section III.B, we model the network as a tree and assume a backhaul bandwidth of 100Mbps between eNodeB and SGW, 200Mbps between SGW and PGW, and 220Mbps between PGW and the Internet CDN. The above selection of backhaul bandwidths, in particular between PGW and Internet CDN, while lower than in a real carrier network, allows us to study a fully loaded network with only few eNodeBs and SGWs instead of a real network with hundreds of such nodes. We assume a network consisting of 9 nodes: 2 sets of 3 eNodeBs are connected to 2 SGWs which are connected to one PGW. The size of the 2^{nd} layer cache is 3 times of the size of the 1^{st} layer cache and the size of the 3^{rd} layer cache is 10 times that of the 1^{st} layer cache. The total number of video requests simulated per trial is 100,000 and the requests originate uniformly from the users of all eNodeBs. The total number of videos available for download is 20,000, distributed uniformly across 250 video categories, and following a Zipf popularity distribution [11] with parameter of -0.8. The video duration is exponentially distributed with mean of 8 minutes and truncated to a maximum of 30 minutes and a minimum of 2 minutes. We assume the video codec bit rate is uniformly distributed between 200kbps (QVGA quality) and 2Mbps (HD quality). The simulation assumes 5000 potential mobile users with Poisson arrival and departure with mean inter-arrival time of 100 seconds and user active time of 2700 seconds (time the user is present whether actively downloading video or not). Video requests are generated independently per active user and follow a Poisson process with mean of 480 seconds. For all the simulations we assume the same total cache size for hierarchical and RAN-only caches but for the hierarchical case the cache has been distributed across three layers and for the RAN-only only across the 1^{st} layer caches. All variables are randomly generated for each simulation trial and all results presented here include 4 trials.

Fig. 2(a) shows the performance of non-hierarchical (MPV, LRU, R-UPP, P-UPP) and hierarchical (H-MPV, H-LRU, H-R-UPP, H-P-UPP) cache policies in terms of cache hit ratio achieved, for different total cache sizes of 50, 100, and

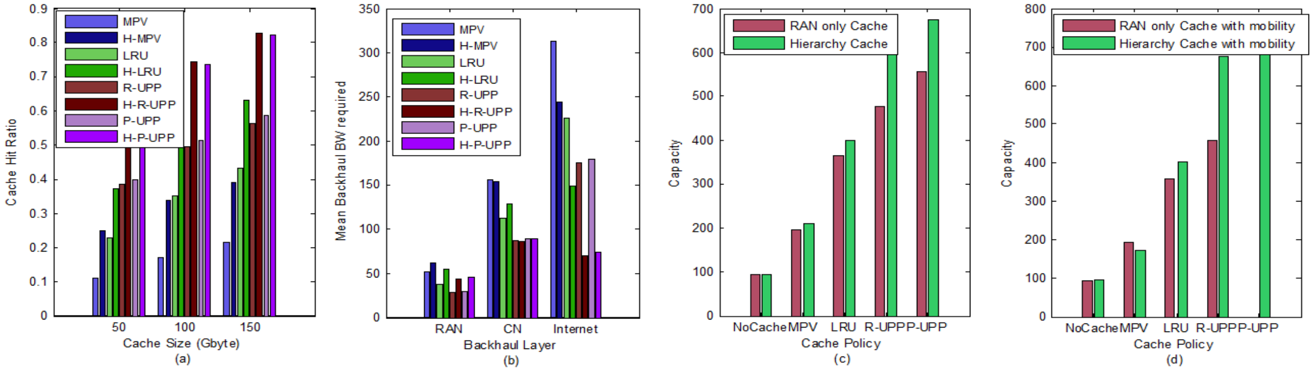

Fig. 2. Performance of the hierarchical caching policies: (a) Cache Hit Ratio vs. Cache Size (b) Mean backhaul BW required per layer of the hierarchy (c) Capacity vs. cache policy with no mobility (d) Capacity vs. cache policy with mobility

150Gbytes. This simulation assumes users do not move from one cell site to another (no mobility). Also, no bandwidth limitation is in effect. In all cases hierarchical caching results in higher overall cache hit ratio compared to caching only at the edge of the RAN although the total cache size is unchanged. For cache size of 150Gbytes, H-P-UPP and H-R-UPP both result in cache hit ratios of 0.82, about 24 and 22 percentage point better than the RAN-only versions respectively.

Fig. 2(b) shows the mean backhaul bandwidth required in RAN, CN, and Internet backhaul when the total cache size is 150Gbytes. The mean required RAN bandwidth for the H-P-UPP is 45Mbps while for the P-UPP it is 30Mbps. The required CN backhaul bandwidth is the same both for H-P-UPP and P-UPP at 90Mbps, while the required Internet backhaul bandwidth is 74 and 180 Mbps for H-P-UPP and P-UPP respectively. We see similar trends for the other cache policies: hierarchical caching results in significantly increased RAN backhaul traffic, but significantly lower Internet backhaul traffic (less data fetched from the CDN), which should result in lower operating costs for the network operator.

Next, we quantify the advantage of caching both at the RAN and CN compared with caching only at the RAN in terms of capacity of the wireless network when considering the bandwidth limitations described in the beginning of the section. Here, capacity is defined as the maximum number of concurrent video sessions that result in a blocking probability of less than 1% [1]. Fig. 2(c) compares capacity of the hierarchical and RAN-only cache policies when the total cache size is 150Gbytes. With the chosen bandwidth configuration, hierarchical caching performs better than RAN-only caching because it addresses congestion in the links between 2^{nd} layer and 3^{rd} layer nodes and 3^{rd} layer node and CDN. For example, network capacity improves by 21% and 30% using hierarchical P-UPP and R-UPP policies compared with RAN-only P-UPP and R-UPP respectively. Capacity using the hierarchical LRU and MPV is improved by 9% and 8% respectively compared to RAN-only versions.

Finally we study the effect of mobility where, in addition to having users added to and removed from cell sites, the users move between cell sites while continuing with their video downloads. In our simulation, cell site migration follows a Poisson process with mean active cell time of 100 seconds – i.e. the mean time a user stays in a cell site before moving to another cell site. An ongoing video session is blocked (terminated) if the eNodeB that the UE migrates to cannot support the new session. Fig. 2(d) compares hierarchical caching with RAN-only caching under mobility condition. We did not present capacity results for the RAN-only P-UPP because this configuration is not suitable for mobility in its native form (i.e. without exchanging neighbor (e)NodeB AUS). From Fig. 2(d), we observe that UPP based hierarchical policies perform significantly better in the case of mobility: hierarchical R-UPP performs 47% better than the RAN-only R-UPP.

VI. Conclusion

In this paper, we proposed hierarchical caching of video contents in the CN to supplement the caches at the edge of the RAN. We extended caching policies proposed in [1] to support hierarchical caching. Our simulation results show that the hierarchical caching of videos in the CN to supplement RAN micro-caching can significantly decrease the required Internet backhaul bandwidth while maintaining the end user's video QoE leading to a significant capacity increase in existing networks. In the future, we plan to extend our approach to consider bandwidth constraints in the RAN RF links.

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

References

[1] H.Ahlehagh and S.Dey, "Video caching in radio access network: Impact on delay and capacity," *Proceedings of the 2012 IEEE Wireless Communications and Networking Conference (WCNC 2012)*, 2012.

[2] White Paper, "Cisco visual networking index: Global mobile data," 2010-2015.

[3] 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.

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

[5] M. Pathan and R. Buyy, "A taxonomy of CDNs, Content Delivery Networks," *Springer-Verlag, Germany*, 2008.

[6] N. Amram *et al.*, "QoE-based transport optimization for video delivery over next generation cellular networks," *In Proceedings of IEEE Symposium on Computer and Communications (SCC)*, 2011.

[7] J. Dyaberi *et al.*, "Scholastic streaming: Rethinking mobile video-on-demand in a campus environment," *ACM Multimedia 2010 - Mobile Video Delivery (MoViD) Workshop, October*, 2010.

[8] H. Holma and A. Toskala, *LTE for UMTS: Evolution to LTE Advanced.*

[9] G. T. 23.401, "General Packet Radio Service (GPRS) enhancements for evolved universal terrestrial radio access network (e-utran) access," *Release 9*.

[10] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, 2007.

[11] M. Cha *et al.*, "Analyzing the video popularity characteristics of large-scale user generated content systems," *IEEE/ACM Transactions on networking*, October 2009.