An Analysis of Two Cooperative Caching Techniques for Streaming Media in Residential Neighborhoods

Shahram Ghandeharizadeh, Shahin Shayandeh, Yasser Altowim
shahram@usc.edu, shayande@usc.edu, altowim@usc.edu
Computer Science Department
University of Southern California
Los Angeles, California 90089

Abstract

Domical is a recently introduced cooperative caching technique for streaming media (audio and video clips) in wireless home networks. It employs asymmetry of the available link bandwidths to control placement of data across the caches of different devices. A key research question is what are the merits of this design decision. To answer this question, we compare Domical with DCOORD, a cooperative caching technique that ignores asymmetry of network link bandwidths in its caching decisions. We perform a qualitative and quantitative analysis of these two techniques. The quantitative analysis focuses on startup latency defined as the delay incurred from when a device references a clip to the onset of its display. Obtained results show Domical enhances this metric significantly when compared with DCOORD inside a wireless home network. The qualitative analysis shows DCOORD is a scalable technique that is appropriate for networks consisting of many devices. While Domical is not appropriate for such networks, we do not anticipate a home network to exceed more than a handful of wireless devices.

1 Introduction

Advances in mass-storage, networking, and computing have made streaming of continuous media, audio and video clips, in residential neighborhoods feasible. Today, the last-mile limitation has been resolved using a variety of wired solutions such as Cable, DSL, and fiber. Inside the home, computers and consumer electronic devices have converged to offer plug-n-play devices without wires. It is not uncommon to find a Plasma TV with wireless connectivity to a DVD player, a time shifted programming device (DVR) such as Tivo, a cable set-top box, a game console such as Xbox, and a computer or a laptop. The primary constraint of this home network[1] is the radio range of devices and the available network bandwidth connecting devices.

The wireless in-home networks are attributed to consumer demand for no wires, ease of deploying a wireless network, and the inexpensive plug-n-play components that convert existing wired devices into wireless ones. A device might be configured with an inexpensive[2] magnetic disk drive and provide hybrid functionalities. For example, a cable box might be accompanied with a magnetic disk drive and provide DVR functionalities [21]. A device may use its storage to cache content.

[1]Power becomes a constraint when a mobile device is removed from the network for use outside the home.
[2]The cost per Gigabyte of magnetic disk is less than 10 cents for 1.5 Terabyte disk drives.

	DCOORD	Domical
Scalable	Yes	No
Limited Network Bandwidth	No	Yes
Employs object size	No	Yes
Data granularity	Clip	Clip/Block

Table 1. A qualitative analysis.

DCOORD [2] and Domical [12] are two cooperative caching techniques for residential neighborhoods. While DCOORD is designed for home gateways in a neighborhood, Domical targets devices inside the wireless home. A qualitative comparison of these two techniques is shown in Table 1. This table shows DCOORD assumes abundant network bandwidth and employs a decentralized hash table to scale to hundreds and thousands of home gateways in a residential neighborhood. Domical, on the other hand, targets an in-home network consisting of a hand-full of devices.

Both DCOORD and Domical partition the available storage space of a device into two areas: a) private space, and b) cache space. The private space is for use by the client's applications. Both techniques manage the cache space of participating devices and their contents. A parameter, α, controls what fraction of cache space is managed in a greedy manner. When α=0, the device is fully cooperative by contributing all of its cache space for collaboration with other devices. When α=1, the device acts greedy by using a technique such as LRU [8] or DYNSimple [9] to enhance a local optimization metric such as cache hit rate. Both DCOORD and Domical support these extreme and intermediate α values.

DCOORD and Domical have different objectives. While Domical strives to minimize the likelihood of bottleneck link formation in a wireless network, DCOORD strives to maximize both the cache hit rate of each node and the number of unique clips stored across the nodes of a cooperative group. In addition, their design decisions are different. DCOORD caches data at the granularity of a clip while Domical supports caching at the granularity of both clips and blocks. (Section 5.1 shows block caching enhances the startup latency observed with Domical.) Finally, DCOORD chooses victim objects using a recency metric while Domical considers both the frequency of access to objects and their size.

Since Domical was designed for use with a handful of devices, it may not substitute for DCOORD outside the home when the neighborhood consists of hundreds of household. This raises the following interesting question: Is it possible for DCOORD to substitute for Domical inside a wireless home? The short answer is a "No" because of the asymmetric bandwidth of the wireless links between devices. To elaborate, a recent study [20] analyzed deployment of six wireless devices in different homes in United States and England. It made two key observations. First, the bandwidth of wireless connections between devices is asymmetric. For example, with a realistic wireless home network corresponding to a deployment of six nodes employing 802.11a networking cards in a British household [20] with asymmetric link bandwidths, the bandwidth from device 5 to device 0 is 9 Mbps while the reverse bandwidth is 1 Mbps.

Second, this study observed that an ad hoc communication provides a higher bandwidth when compared with a deployment that employs an access point because it avoids the use of low bandwidth connection(s). For example, if node 5 is the access point, all communications from other devices to device 0 would observe a bandwidth of 1 Mbps. With an ad hoc communication, a device may communicate with those devices in its radio range directly. Moreover, a device may send its data to a destination device by using other devices as routers. In our example, ad hoc communication enables device 5 to stream a clip to device 0 at bandwidths significantly higher than 1 Mbps by using device 3 (or 2) as an intermediary to route its data.

The primary contribution of this study is to quantify the merits of a cooperative caching technique such as Domical that controls placement of data across devices by considering the asymmetry of their wireless link bandwidths. We use DCOORD as a comparison yard-stick because it is the only cooperative caching technique that is comparable to Domical, see Section 2. Obtained results show Domical enhances startup latency observed by different devices significantly. This implies that for a wireless home network an appropriate cooperative caching technique should consider bandwidth configurations between different devices.

A secondary contribution is to highlight caching of data at the granularity of block when network bandwidth and storage are abundant. With a cooperative technique such as Domical, block (instead of clip) caching enhances startup latency.

The rest of this paper is organized as follows. Section 2 surveys research related to cooperative caching techniques. Sections 3 and 4 provide an overview of Domical and DCOORD, respectively. Section 5 provides a quantitative comparison of these two techniques. We conclude with future research directions in Section 6.

2 Related Work

To the best of our knowledge no study quantifies the performance of two different cooperative caching techniques for streaming media in a wireless home network. In [11], we showed Domical enhances average startup latency of a wireless home network when compared with a greedy caching technique. Below, we survey other cooperative caching techniques. A key conceptual difference between these techniques and DCOORD/Domical is the following: Prior techniques do not enable a node to support both greedy and cooperative modes of cache management simultaneously using the parameter α.

A group of proxies may cooperate to increase the aggregate cache space, balance load, and improve system scalability [1, 5, 11, 12]. MiddleMan [1] segments clips into equi-sized segments and employs a centralized coordinator to balance load evenly across proxies. Cont-Coop [11] and Domical [12] cache data with the objective to balance the load of streaming media across the asymmetric wireless connections of a home network. While these techniques impose more work on a device with abundant network bandwidth in order to enhance startup latency, MiddleMan strives to balance the load across proxy servers. Both Cont-Coop and Domical cache data at the granularity of a clip with Domical outperforming Cont-Coop [12]. Our study is novel because we quantify the tradeoffs associated with two different cooperative caching techniques, Domical and DCOORD [2]. One may implement our proposed techniques using the centralized coordinators of MiddleMan [1] and Middleware of [16], and proxy cluster of Silo [5].

Cooperative proxy caching for streaming media has been explored in overlay networks [15, 18]. COPACC [15] assumes an architecture of a two level cache: at both proxies and clients. It partitions a clip into three segments: Prefix, Prefix-of-suffix, and Suffix. While proxies cache the prefix of a clip, clients cache the Prefix-of-suffix. This layout is shown to minimize the startup latency and facilitate multicast delivery with dynamic clients. OCS [18] assumes a similar architecture as COPACC and presents a caching technique at the granularity of a clip. It specifically notes that caching at the granularity

Term	Definition
Startup latency (δ)	Delay from when a user references a clip to the onset of display.
Cache hit rate	The percentage of clip requests satisfied using the cache.
Byte-hit rate	Percentage of bytes retrieved from the local cache of a device
μ	Mean of the distribution of access
$B_{Display_i}$	Bandwidth required to display clip i
$d_{i,j}$	Display time of j^{th} block of clip i
S_{DB}	Size of the repository
S_T	Total cache capacity of nodes in the network
f_i	Frequency of access to clip i
S_i	Size of clip i
α	Parameter indicating size of Collaborative Vs Elite space

Table 2. Terms and their definitions

of a block provides inferior average startup latency than clip-based caching (without either presenting comprehensive results or elaborating on the reasons). Our study is different because, in our architecture, the proxy and client caches are the same and one.

3 Domical

Domical [12] constructs dependencies between the caches of different devices with the objective to prevent the possibility of a wireless link from becoming a bottleneck. A device may contribute a fraction of its available cache space for cooperation with other devices. Data items occupying the cache space might be either clips or blocks of different clips. To simplify the discussion and without loss of generality, in the following we assume the granularity of cached data is a clip. In Section 3.1, we describe modifications to support block caching.

The cache space of N_i is made dependent[3] on the state of caches managed by devices N_0, N_1, ..., N_{i-1}. This impacts how N_i victimizes clips as follows. N_i constructs a list of those clips with a replica in the cache space of N_0, N_1, ..., N_{i-1}. These are named common clips. The remaining clips are named rare clips. Domical victims the common clips with the lowest byte-hit ratio first. If this does not release sufficient space for the incoming clip X and there are no common clips left, then Domical victimizes the rare clips starting with those that have the lowest byte-hit ratio.

Domical abstracts the available cache space of a device into two parts named Collaborative, and Elite. The Collaborative and Elite portions of the storage space constitute the available cache space and are managed by Domical. Domical manages the Elite cache space with the objective to optimize a local metric such as cache hit rate. Candidate algorithms include LRU-K [19], GreedyDual-Size [17] or DYNSimple [9] to name a few. Domical renders the Collaborative cache space of different devices dependent on one another (without compromising autonomy of different devices) in order to enhance a global metric such as average startup latency.

[3]Towards the end of this section, we describe how Domical computes dependencies between different nodes.

The parameter α dictates what fraction of the available cache space is assigned to the Collaborative and Elite portions. Assuming S_{N_i} dictates the cache space of device N_i, the size of Elite space is $\alpha \times S_{N_i}$. The remainder, $(1 - \alpha) \times S_{N_i}$, is the Collaborative space. In [12], we show Domical enhances the average startup latency significantly with $\alpha = 0$ when compared with $\alpha = 1$.

One may implement Domical in different ways. Assuming devices in the home network are single user devices with abundant amount of processing capability, a simple implementation would be as follows. When a clip X is admitted into the cache and Domical has insufficient space to store X, it victimizes existing clips as follows. First, it sorts[4] those clips occupying N_i's cache in descending order using their byte-hit ratio, $\frac{f_j}{S_j}$, where f_j is the frequency of access to object j and S_j is the size of that object. Next, it marks the first k clips with a total size exceeding the size of the Elite space of the cache. These clips maximize the byte-hit ratio of the Elite space. The remaining clips occupy the Collaborative space. Domical identifies the common and rare clips in the Collaborative space, removing as many common clips with a low byte-hit ratio to provide space for the incoming clip X. Once common clips are exhausted, it deletes the rare clips starting with the ones that have the lowest byte-hit ratio.

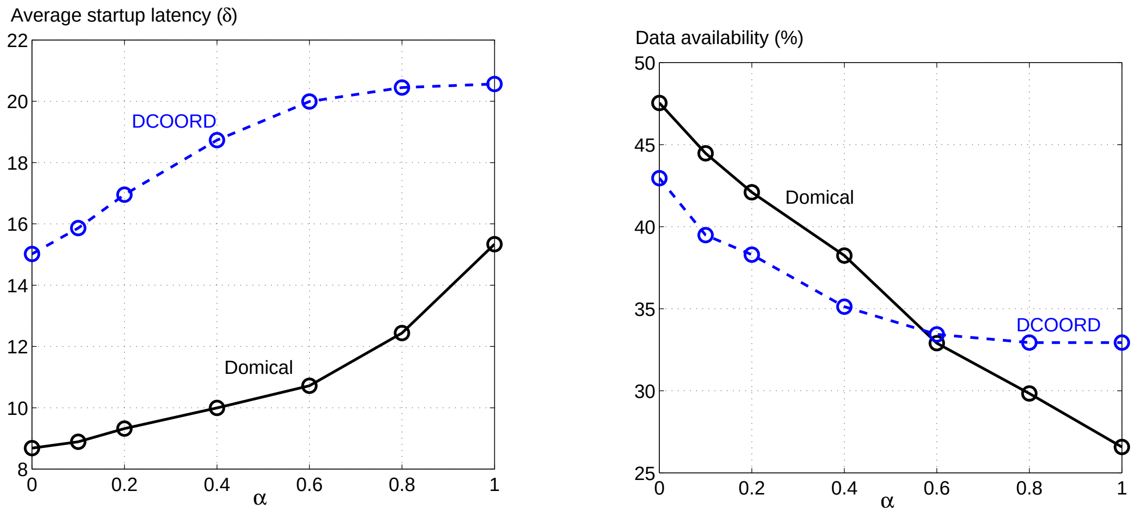
N_i may identify common and rare clips by listening on the traffic generated by different devices referencing different clips. Moreover, it may periodically exchange the identity of clips occupying its cache with other devices.

Domical computes dependencies between shared caches of different nodes using the bandwidth contention ratio metric [12]. This metric estimates the amount of imbalance across the network links when N_i streams a clip to every other node in the network. Intuitively, the node with the smallest bandwidth contention value (N_{min}) results in the lowest imbalance, avoiding formation of hot spots and bottleneck links. N_{min} manages its entire cache space (both Elite and Collaborative) using a greedy caching technique. The state of Collaborative space of every other node in the cooperative group depends on N_{min}. Moreover, the remaining nodes are sorted in ascending order using their bandwidth contention ratio metric. The state of the Collaborative space of a node i is dependent on the cache of every other node (both Elite and Collaborative space) before it and N_{min}.

3.1 Domical Block-based Replacement (DBR)

To change the granularity of data placement from a clip to a block, we modify the local replacement policy used by each device to manage its Elite and Collaborative cache space. Instead of byte hit ratio, each device employ the urgency worthiness metric [10] to sort the list of blocks in order to choose victims. This metric is defined as $\frac{f_i}{B_{Display_i} \times d_{i,j}}$ where $B_{Display_i}$ is the bandwidth required to display clip i and $d_{i,j}$ denotes the display time of j^{th} block of clip i relative to its start. The intuition behind this metric is to cache sufficient number of the first few blocks of a clip (termed prefetch portion) to enable a client to display these blocks while streaming the remaining blocks of a clip without the client starving for data. This complements streaming to minimize observed startup latency close to zero, similar to a clip being cached in its entirety. The dependencies between Collaborative cache of different devices minimizes the likelihood of duplicate prefetch portions cached across multiple devices.

[4] Sorting is not appropriate for devices with multiple simultaneous users due to its computational overhead.



1.a) Startup latency

1.b) Data availability

Figure 1. Different α values, Domical Vs DCOORD, $UK1X$, $\mu = 0.73$, $\frac{S_T}{S_{WS}} = 0.5$

4 DCOORD

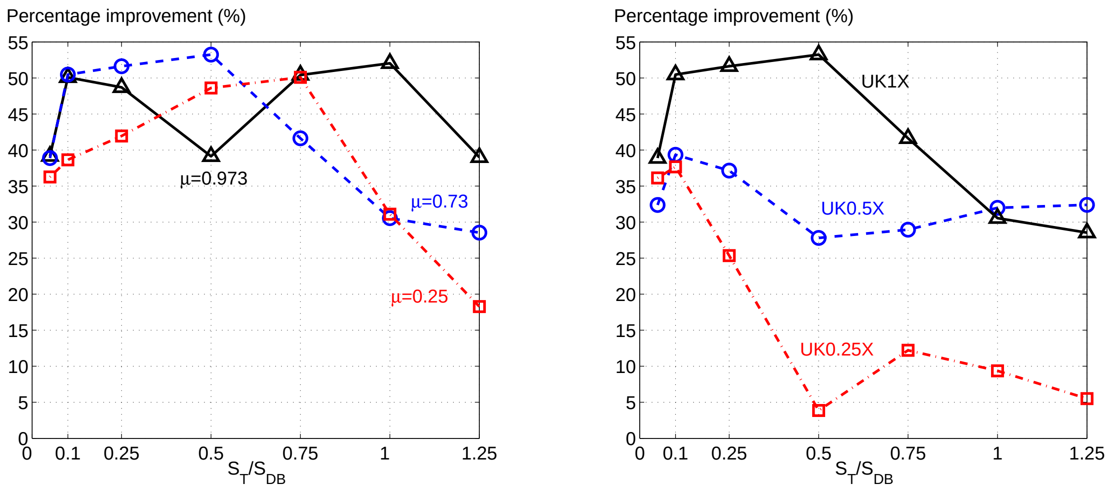
DCOORD employs a hash function to assign clips to the different nodes in a cooperative group. Similar to Domical, the available cache space is partitioned into two sets: 1) WS, working set, and 2) SS, shared set. The parameter α dictates the size of each set. Assuming total storage capacity of a node i is S_{N_i}, size of WS is $\alpha \times S_{N_i}$ and size of SS is $(1 - \alpha) \times S_{N_i}$.

DCOORD strives to maximize both the cache hit rate of each node and the number of unique clips stored across the nodes of a cooperative group. This is realized by each node N_i maintaining a directory and a list. The directory of N_i maintains the list of clips that hash to N_i with a copy on some other node in the cooperative group. The list of N_i implements an LRU replacement technique to manage the identity of clips occupying N_i's cache. The most recently referenced clip by node N_i is at the "tail" of the list and is a member of the WS. The clips at the "tail" of the list whose total size approximates the size of N_i's WS constitute N_i's WS.

The "head" of the list points to those clips that are candidates for eviction. When a clip X is referenced by N_i's neighboring device N_j, DCOORD moves this clip to the head of N_i's list because it assumes the neighboring device will cache the referenced clip. This increases the number of unique clips across the cooperative caches because the referenced clip will be victimized from N_i's cache. If the hash function assigns clip X to N_i, then N_i maintains N_j in its directory structure. Future requests by neighboring devices that reference clip X are directed to N_j by looking up this directory.

5 A simulation study

When one compares Domical and DCOORD, the following natural questions arise: Is it possible for DCOORD to substitute for Domical? And, if Domical is better then how much better is it? To answer these questions, we built a simulation model of both DCOORD and Domical. This model assumes a household consisting of six wireless devices with wireless



2.a) $UK1X$ 2.b) $\mu = 0.73$

Figure 2. Percentage improvement in startup latency provided by Domical when compared with DCOORD, α=0.

network bandwidths identical to those of a United Kingdom household reported in [20]. This household is denoted as UK1X. We scale down the link bandwidths by a factor of 2 and 4 to construct two hypothetical households, UK0.5X and UK0.25X.

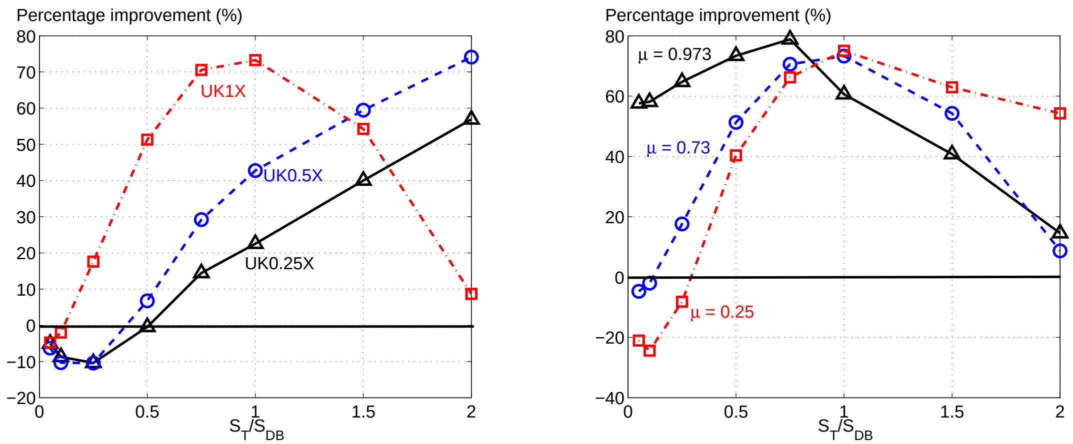
We assumed a heterogeneous repository consisting of 864 clips. All are video clips belonging to two media types with display bandwidth requirements of 2 and 4 Mbps. The 432 clips that constitute each media type are evenly divided into those with a display time of 30, 60, and 120 minutes. The total repository size, S_{DB}, is fixed at 1.29 Terabytes. Each device is configured with the same amount of cache space and the total size of this cache in the network is S_T. In our experiments, we manipulate the value of S_T by reporting the ratio $\frac{S_T}{S_{DB}}$.

We use a Zipf-like distribution [3] with mean of μ to generate requests for different clips. A larger value for exponent μ makes the distribution more skewed by increasing the frequency of access to a few popular clips. On the other hand, a smaller value of μ makes the distribution more uniform. Zipf has been used to model the distribution of web page requests [14, 3, 22], and sale of movie tickets[5] in the United States [7].

One node in the system is designated to admit requests in the network by reserving link bandwidth on behalf of a stream. This node, denoted N_{admit}, implements the Ford-Fulkerson algorithm [6] to reserve link bandwidths. When there are multiple paths available, N_{admit} chooses the path to minimize startup latency.

The simulator conducts ten thousand rounds. In each round, we select nodes one at a time in a round-robbin manner, ensuring that every node has a chance to be the first to stream a clip in the network. A node (say N_1) references a clip using a random number generator conditioned by the assumed Zipf-like distribution. If this clip resides in N_1's local storage then its display incurs a zero startup latency. Otherwise, N_1 identifies those nodes containing its referenced clips, termed candidate servers. Next, it contacts N_{admit} to reserve a path from one of the candidate servers. N_{admit} provides N_1 with the amount of

[5]In [7], a Zipf-like distribution is defined as $\frac{\chi}{i^{(1-\omega)}}$ where ω is 0.27. In this paper, μ equals $1-\omega$. To be consistent with [7], we analyze 0.73 as a possible value for μ in Section 5.



3.a $\mu = 0.73$ 3.b $UK1X$

Figure 3. Percentage improvement with block-based caching when compared with clip-based caching using Domical.

reserved bandwidth, the paths it must utilize, and how long it must wait prior to streaming the clip. This delay is the incurred startup latency.

In each iteration, we measure the following parameters local to a node: startup latency, byte-hit and cache hit ratios, and unutilized cache space. In addition, we measure the following global parameters: average startup latency, average hop distance to stream a clip, and average amount of bytes transmitted across the network. To minimize the impact of a cold-start on the observed parameters, the most popular clips are stored in the cache of every device at the start of the experiment. The number of cached clips changes for different $\frac{S_T}{S_{DB}}$ values.

5.1 Performance results

Figure 1.a shows the average startup latency with Domical and DCOORD as a function of different α values. When compared with one another, Domical enhances average startup latency by approximately 40% to 50%. It is interesting to note that Domical results in higher availability of data for α values less than 0.6, see Figure 1.b. This means the dependencies between the caches of different devices (constructed by Domical) is effective in maximizing the number of unique clips in the home network. With $\alpha = 1$, DCOORD provides a higher availability because (a) it employs a hash function to assign clips to nodes, and (b) when a clip assigned to N_i is referenced by a neighboring device, N_i places this clip as the next to be evicted from N_i's local storage. Such a mechanism does not exist with Domical.

Domical provides a lower startup latency than DCOORD because it assigns the frequently accessed clips to the device with the highest out-going link bandwidths. This minimizes the formation of bottleneck links in the wireless network, reducing the possibility of a device waiting for an active display of a clip to end.

In almost all our experiments, Domical outperforms DCOORD. In Figure 2.a, we show the percentage improvement in startup latency observed by Domical when compared with DCOORD for different distributions of access to clips and $\alpha = 0$

using the network bandwidth observed from the UK household of [20]. In this experiment, we vary the total cache size (S_T) on the x-axis. Even with an access distribution that resembles a uniform distribution ($\mu = 0.25$), Domical outperforms DCOORD because it materializes a larger number of unique clips across the cooperative cache.

The bandwidth of the wireless links has an impact on the margin of improvement provided by Domical. This is shown in Figure 2.b where we analyze the impact of scaling down wireless link bandwidths: Factor of two and four relative to the original observed link bandwidths, termed UK0.5x and UK0.25X, respectively. The percentage improvement observed by Domical drops because the bandwidth of wireless links are so low that formation of bottlenecks is very high.

One may improve the startup latencies observed with Domical by changing the granularity of caching from clip to block. This is because Domical pre-stages the first few blocks of different clips across the network strategically in order to minimize the startup latency. This is shown in Figure 3 where we report on the percentage improvement observed with block caching when compared with clip caching (for Domical). Note that when either the available cache space or bandwidth of wireless network connections is scarce (low $\frac{S_T}{S_{DB}}$ ratios in Figure 3.a with UK0.25X), caching at the granularity of a clip is the right choice. This is because, with block-based caching, the remainder of each clip referenced by every device may involve the infrastructure outside the home, exhausting the wireless network bandwidth of the home gateway [13].

6 Future research directions

The asymmetric and limited bandwidth of wireless connections between devices in a household make a compelling case for a cooperative caching technique such as Domical. This is because Domical assigns data to the available cache space of different devices with the objective to minimize the likelihood of bottleneck links in the network. In this paper, we did a qualitative and quantitative comparison of Domical with DCOORD. The qualitative analysis shows Domical is not a substitute for DCOORD outside the home. The quantitative analysis shows Domical enhances average startup latency significantly when compared with DCOORD inside the home.

Our comparison of DCOORD with Domical raises several interesting future research directions. First, should DCOORD be extended to cache data at the granularity of a block (instead of a clip)? Such an extension may combine DCOORD's data structures with design elements from segment-based [23] caching and Domical's urgency-worthiness metric [13, 10]. Second, one may design a new cooperative caching technique by combining concepts of Domical with DCOORD. Such a technique may strive to enhance both average startup latency and availability of the data. Third, both DCOORD and Domical can be tailored based on the requirements of an application. For example, if an application consists of users who display a fraction of their referenced clips (as assumed by Silo [4]), one may employ variable sized logical blocks. We intend to extend our analysis to better understand the impact of such these design decisions. Finally, we are exploring a possible implementation of Domical. This implementation may use middlewares such as [16] as its basis. Other obvious candidates include the centralized coordinator of MiddleMan [1] and the proxy cluster of Silo [4].

References

[1] S. Acharya and B. Smith. MiddleMan: A Video Caching Proxy Server. In *Proceedings of the International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, pages 162–172, June 2000.

[2] H. Bahn. A Shared Cache Solution for the Home Internet Gateway. *IEEE Transactions on Consumer Electronics*, 50(1):168–172, February 2004.

[3] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web Caching and Zipf-like Distributions: Evidence and Implications. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, pages 126–134, 1999.

[4] Y. Chae, K. Guo, M. Buddhikot, S. Suri, and E. Zegora. Silo, Rainbow, and Caching Token: Schemes for Scalable, Fault Tolerant Stream Caching. *IEEE Journal on Selected Areas in Communications on Internet Proxy Services*, Sept. 2002.

[5] Y. Chae, K. Guo, M. Buddhikot, S. Suri, and E. Zegura. Silo, Rainbow, and Caching Token: Schemes for Scalable, Fault Tolerant Stream Caching. *IEEE Journal on Selected Areas in Communications*, 20(7):1328–1344, Sep 2002.

[6] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, editors. *Introduction to Algorithms*, chapter 26.2. MIT Press, 2001.

[7] A. Dan, D. Sitaram, and P. Shahabuddin. Scheduling Policies for an On-Demand Video Server with Batching. In *Proceedings of the ACM Multimedia Conference*, pages 15–23, October 1994.

[8] P. J. Denning. The Working Set Model for Program Behavior. *Communications of the ACM, Vol. 11, No. 5*, pages 323–333, 1968.

[9] S. Ghandeharizadeh and S. Shayandeh. Greedy Cache Management Techniques for Mobile Devices. In *Proceedings of the IEEE International Conference on Data Engineering*, pages 39–48, April 2007.

[10] S. Ghandeharizadeh and S. Shayandeh. An Evaluation of Two Domical Block Replacement Techniques for Streaming Media in Wireless Home Networks. In *Proceedings of the IEEE International Symposium on Multimedia (ISM)*, pages 372–377, Dec. 2008.

[11] S. Ghandeharizadeh and S. Shayandeh. Cooperative Caching Techniques for Continuous Media in Wireless Home Networks. In *Proceedings of the international conference on Ambient media and systems (Ambi-Sys)*, pages 1–8. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), February 2008.

[12] S. Ghandeharizadeh and S. Shayandeh. Domical Cooperative Caching: A Novel Caching Technique for Streaming Media in Wireless Home Networks. In *Proceedings of International Conference on Software Engineering and Data Engineering (SEDE)*, pages 274–279, June 2008.

[13] S. Ghandeharizadeh and S. Shayandeh. Cache Replacement Techniques for Streaming Media in Wireless Home Networks. Technical Report 2008-05, USC Computer Science Department, Los Angeles, CA, 2008.

[14] S. Glassman. A caching relay for the world wide web. *Computer Networks and ISDN Systems*, 27(2):165–173, 1994.

[15] A. T. S. Ip, J. Liu, and J. C.-S. Lui. COPACC: An Architecture of Cooperative Proxy-Client Caching System for On-Demand Media Streaming. *IEEE Trans. Parallel Distrib. Syst.*, 18(1):70–83, 2007.

[16] W. J. Jeon and K. Nahrstedt. QoS-aware Middleware Support for Collaborative Multimedia Streaming and Caching Service. *Microprocessors and Microsystems*, 27(2):65–72, 2003.

[17] S. Jin and A. Bestavros. Popularity-Aware GreedyDual-Size Web Proxy Caching Algorithms. In *Proceedings of the 20th International Conference on Distributed Computing Systems (ICDCS)*, pages 254–261, April 2000.

[18] T. M, W. Tavanapong, and W. Putthividhya. OCS: An Effective Caching Scheme for Video Streaming on Overlay Networks. *Multimedia Tools Appl.*, 34(1):25–56, 2007.

[19] E. J. O'Neil, P. E. O'Neil, and G. Weikum. The LRU-K Page Replacement Algorithm for Database Disk Buffering. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 413–417, 1993.

[20] K. Papagiannaki, M. Yarvis, and W. S. Conner. Experimental characterization of home wireless networks and design implications. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, pages 1–13, April 2006.

[21] J. R. Quain. Cable Without a Cable Box, and TV Shows Without a TV. The New York Times, Technology Section, July 26 2007.

[22] A. Wolman, M. Voelker, N. Sharma, N. Cardwell, A. Karlin, and H. M. Levy. On the Scale and Performance of Cooperative Web Proxy Caching. *SIGOPS Opereting Systems Review*, 33(5):16–31, April 1999.

[23] K. Wu, P. S. Yu, and J. L. Wolf. Segment-based Proxy Caching of Multimedia Streams. In *WWW*, 2001.