Collaborative Caching in Wireless Video Streaming Through Resource Auctions

Jie Dai, *Student Member, IEEE,* Fangming Liu, *Member, IEEE,* Bo Li, *Fellow, IEEE,* Baochun Li, *Senior Member, IEEE,* and Jiangchuan Liu, *Senior Member, IEEE*

***Abstract*—Recent advances in wireless communications and mobile networking have dramatically increased the popularity of multimedia services for mobile users, with wireless video streaming at their fingertips. To facilitate efficient acquisition of video content, proxy caching has been widely used by wireless service providers (WSPs), which typically deploy cache servers at mobile switching centers (MSCs). However, capacity provisioning of cache servers is challenging, given the dynamic user demands and the limited cache server resources. With increased densities of wireless service deployment, it is increasingly common that mobile users are covered by more than one WSP within an area. This brings opportunities of a collaborative caching paradigm among the cache servers deployed at different MSCs. In this paper, we explore the benefits of collaborative caching in wireless streaming services, addressing both challenges of incentives and truthfulness of selfish WSPs. We propose a collaborative mechanism that maximizes the social welfare in the context of Vickrey-Clarke-Groves (VCG) auctions, in which cache servers cooperate in the trading of their resources in a self-enforcing manner. Experimental results demonstrate that superior performance can be achieved with respect to the quality of video streaming.**

***Index Terms*—Collaborative Caching, VCG Auction, Incentive Engineering, Truthfulness.**

I. Introduction

The popularity of on-demand video streaming has substantially changed the landscape of wireless multimedia applications, with an increasing number of content providers, such as YouTube and Netflix, offering streaming video content to mobile users. Meanwhile, a diverse range of wireless access technologies, including HSPA+ cellular access, Femtocells, and recent advances in 4G deployment such as LTE [1], have made broadband wireless connections a near-term reality.

Thanks to the availability of last-mile wireless bandwidth, wireless service providers (WSPs) have increasingly focused on the quality of wireless video streaming, with *proxy caching* being extensively utilized to improve the streaming quality [2]. Cache servers have been deployed in Mobile Switching Centers (MSCs) [3] of WSPs, so as to locate video content closer to end users. For instance, Verizon has announced its network optimization deployment [4], in which video caching is applied to improve the experience of wireless users.

Nevertheless, the inherent dynamics of mobile users have presented a daunting challenge to resource provisioning at these cache servers. Mobile users frequently join and leave streaming sessions in the presence of mobility, and consequently the load on deployed cache servers becomes unpredictable. The intuitive solution of over-provisioning resources would lead to inevitable but unnecessary waste of resources, given the bursty nature of aggregated demand.

Considering the fact that mobile users are often covered by multiple WSPs in the same area, all of which deliver video content to their own clients, a better way to offload their cache servers is to incentivize these *autonomous* and *selfish* WSPs to *collaborate* with one another. The cache servers belonging to different WSPs may assist one another to stream video content to end users by using their own bandwidth and storage resources. Such a collaboration would improve the degree of multiplexing available resources in WSPs, as well as the general availability of video content in proxy servers.

Existing works in the literature have not demonstrated how these independent cache resources owned by WSPs can be organized in a collaborative way, so as to achieve more efficient multiplexing of resources. The main challenge is the inherently *selfish* nature of autonomous WSPs, often belonging to different for-profit corporations. Though WSPs are rational in their strategic behavior, they will not cooperate with others if their gains do not outweigh their costs.

In this paper, we focus on engineering the *incentives* to promote and encourage the cache servers owned by different WSPs to *truthfully* cooperate with one another. Inspired by the theory of Vickrey-Clarke-Groves (VCG) auctions [5], we design and analyze a new collaborative caching mechanism that can be employed by co-locating WSPs. VCG auctions are known as non-cooperative games, in which any decision about cooperation is "self-enforced." We treat server bandwidth as commodities in these auctions, with different valuations based on the dynamics of the streaming systems. Virtual payments associated with the valuation of bandwidth resources ensure that the contribution of cache servers is acknowledged by other participants. *Truthfulness* is also guaranteed which enables cache servers to faithfully reveal their true valuation when

Manuscript received 15 February 2011; revised 20 July 2011. The research was support in part by a grant from Huawei Technologies Co. Ltd. under the contract HUAW18-15L0181011/PN, by a grant from HKUST under the contract RPC11EG29, by a grant from The National Natural Science Foundation of China (NSFC) under grant No.61103176.

J. Dai and B. Li are with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong (e-mail: {jdai, bli}@cse.ust.hk).

F. Liu is with the Services Computing Technology and System Lab, Cluster and Grid Computing Lab in the School of Computer Science and Technology, Huazhong University of Science and Technology (e-mail: fmliu@mail.hust.edu.cn).

B. Li is with the Department of Electrical and Computer Engineering, University of Toronto (e-mail: bli@eecg.toronto.edu).

J. Liu is with the School of Computing Science, Simon Fraser University (e-mail: jcliu@cs.sfu.ca).

Digital Object Identifier 10.1109/JSAC.2012.120226

"bids" are submitted. Our bidding strategy is far simpler than conventional cooperative networking mechanisms that require sophisticated strategies to coordinate the behavior across participants. Our simulation results further show that the performance of streaming systems can be noticeably improved by maximizing the social welfare in the auctions, in which the bandwidth units are used to serve more valuable demands.

The remainder of the paper is organized as follows. In Sec. II, we discuss our contribution in the context of related works. In Sec. III, we formulate the basic model of resource auctions in collaborative caching. In Sec. IV, we present the design of VCG-based bandwidth resource trading mechanisms. Sec. V discusses how storage resources are utilized in our proposed mechanisms. Sec. VI proceeds to present an extensive simulation study to evaluate the effectiveness of collaborative caching. Finally, we conclude the paper in Sec. VII.

II. Related Work

Proxy caching has long been utilized in wireless streaming to improve its performance. Zhang *et al.* [2] present a cost-based cache replacement algorithm for a single cache server and a server selection algorithm for multiple cache servers in wireless multimedia proxy caching. They further analyze the Quality of Service (QoS) requirement in wireless streaming while introducing the corresponding QoS-adaptive proxy caching mechanisms [6]. Tan *et al.* [7] introduce a smart caching design that duplicates detected video content at access points, to effectively reduce any redundant traffic in the WLAN while improving the response delay of video streaming. Our work differs from these studies by implementing a collaboration mechanism among cache servers. Our collaborative caching encourages the cache servers to actively deliver content across different domains, while both fairness and truthfulness are guaranteed.

The potential benefit of collaborative caching in video streaming systems has been discussed in our previous work [8] and several other related works [9] [10]. Chen *et al.* [9] consider a collaborative caching mechanism in an Internet Protocol Television (IPTV) system with a hierarchical architecture. They suggest that central offices with limited storage space should cooperatively exchange video content that is temporarily unavailable. Borst *et al.* [10] develop distributed caching algorithms that aim to maximize the traffic volume served from a cache, while minimizing bandwidth costs. Their target application scenario is a video-on-demand (VoD) system with tree topologies. These existing studies generally assume a single authority that owns all of the existing resources; the cache servers could then spontaneously cooperate with one another under certain sophisticated and enforced regulations. In contrast, we consider the more practical scenario where cache servers are distributed over multiple domains. They may be arbitrarily connected and exhibit selfish behavior to maximize their own benefits, which presents significant challenges to the design of efficient and truthful cooperation.

Game theoretic models have also been applied in video streaming systems to provide incentives among cache servers.

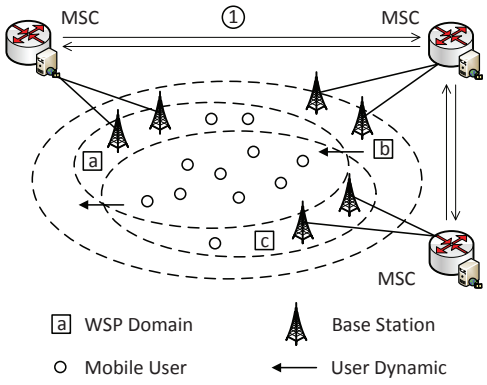


Fig. 1. An illustrative example of collaborative caching.

Ip *et al.* [11] propose an incentive scheme to motivate cache servers to share their caches. The revenue is rewarded based on the amount of cache spaces provided and on a price function. In contrast, rather than solely considering byte hit rates and storage costs [11] that are more suitable for traditional file caching systems, our collaborative caching design focuses on improving the quality of wireless streaming quality, with a particular emphasis on bandwidth provisioning.

III. Resource Auctions in Collaborative Caching

We start with an illustrative example to describe basic principles in our collaborative caching mechanisms for WSPs, in order to achieve better social efficiency.

A. An Example and Basic Features

In wireless video streaming, cache servers are deployed by WSPs to improve the overall system performance. Each cache server utilizes its constrained bandwidth and storage capacity to serve video content requests from mobile users, which consequently results in better streaming quality for a group of video programs, also referred to as *channels*. In conventional caching deployed by each individual WSP, these resources are always allocated to local users within its own domain. We refer to this as *independent caching* in this paper. Our proposed collaborative caching mechanism further explores the cooperation among cache servers across different administrative domains to improve the streaming quality.

We present a scenario in Fig. 1 to describe how such a collaboration can be achieved. WSP a and WSP b deploy cache servers at their own MSCs which are connected to base stations. A server in domain b only has a limited storage capacity. With independent caching, requests to unavailable content in b have to be relayed to remote servers. However, this problem can be alleviated through collaboration across different domains. For instance, after receiving content requests as indicated by arrow 1, the cache server with available content in domain a can allocate certain bandwidth to fulfill requests from mobile users belonging to domain b. Intuitively, such collaboration helps alleviate the sudden request surge in a domain often encountered when there is a popular channel. As such, the dynamics of the overall video streaming system can be smoothly dispersed across multiple WSPs. The efficient

utilization of bandwidth resources across cache servers is critical in dealing with the dynamics of the system.

Despite the potential benefits, there is a lack of a systematic and practical coordination mechanism among cache servers deployed by different WSPs. Given the autonomous nature of WSPs, one distinctive feature is the *selfishness* among deployed cache servers. This implies that, (1) cache servers within one WSP are primarily concerned with the potential benefits through collaborations. Thus, there needs to be a proper *incentive* mechanism for them to collaborate. (2) selfish WSPs may not faithfully conform to regulations of the collaboration if "dishonest" behavior can benefit them. Therefore, the *truthfulness* that forces WSPs to reveal their true information is also crucial in the design. (3) given the distributed nature of the content distribution, a fully decentralized coordination is required. Our primary objective is to design such a collaborative caching mechanism that improves the social welfare, while guaranteeing both incentives and truthfulness.

B. Model Formulation

We consider a WSP set $\mathcal{I} = \{1, 2, ..., N_i\}$ in collaborative caching with a total of N_i WSPs that provide streaming services in the same region. We integrate the servers belonging to the same WSP into one cache server for the sake of simplicity. The notation $i \in \mathcal{I}$ can be used interchangeably to represent WSP i or the cache server of WSP i. Let the set of video channels be denoted by $\mathcal{K} = \{1, 2, ..., N_k\}$, where N_k denotes the total number of channels that are available over multiple domains. The bandwidth and storage capacity at server i are denoted by W_i and S_i.

The VCG auction from game theory fits the need to design our collaboration mechanism such that, (1) each buyer has to *pay* after each trade, which serves as the incentive to contribute. (2) the payment method of the VCG auction ensures *truthfulness*, in which buyers are willing to truthfully reveal their bidding information in the auction. (3) auctions can be organized in a fully decentralized fashion. We now present the basic model and mechanism design inspired by the VCG auction theory, in order to achieve efficient allocations that benefit users of different WSPs.

When auction is applied in the context of collaborative caching, auction participants become cache servers, with bandwidth resources being commodities. Each cache server acts as both a seller or bidder that trades resources in multiple rounds. When acting as a seller, cache server i provides bandwidth W_i^r to remote bidders. We define the bandwidth unit w as the basic unit to be traded in VCG auctions. Therefore, W_i^r has been divided into W_i^r/w homogeneous bandwidth units. Note that this forms a multi-unit auction with the VCG mechanism [12]. The assigned bandwidth can be implemented as an external source residing in a remote MSC, which adaptively adjusts its uplink capacity. The following equation then holds, in which W_i^k denotes the local bandwidth assignment to channel $k \in \mathcal{K}$:

$$W_i^r = W_i - \sum_{k \in \mathcal{K}} W_i^k \tag{1}$$

Auctions are conducted when bidders need a certain amount of bandwidth from sellers to fulfill content requests for locally unavailable contents or to alleviate the pressure on heavily loaded channels. For the buyer server i, the number of bandwidth units required for channel k from server j is denoted as n_{ij}^k. Buyer i will submit bids that include n_{ij}^k and the corresponding valuation information to seller j. After collecting all bids from the cache servers in other WSPs, the winner determination process at the seller side can decide the bandwidth allocation scheme. Consequently, buyers have to pay by *virtual payments* that can later be used for bandwidth resource demands, which serves as an incentive for contributions. The truthfulness will be guaranteed by the payment method with any bids deviating from the true valuation not benefiting in the VCG auction.

Besides the bandwidth allocation scheme derived from the above auctions, storage resources on cache servers also require periodic updates to keep valuable content that benefit their own payoffs and social welfare. We show that storage updates in collaborative caching help achieve a desirable level of performance with a comparatively lower overhead, given different settings of caching mechanisms.

IV. VCG-based Bandwidth Trading

In this section, we describe the VCG-based bandwidth trading mechanism employed by cache servers. We introduce the concept of *virtual bidder* b_i^k which submits bids to request n_{ij}^k bandwidth units on behalf of users on channel k in domain i. b_i^k has a privately known valuation function: $v_i^k : \{0, 1, ..., n_{ij}^k\} \rightarrow \Re, \forall i \in \mathcal{I}, k \in \mathcal{K}$, in which $v_i^k(n)$ denotes the valuation of the benefit of receiving n bandwidth units. b_i^k then submits bids $\mathbf{b}_i^k = \{(0,0), (1, v_i^k(1)), ..., (n_i^k, v_i^k(n_{ij}^k))\}$ to seller j since submitting one's true valuation is the dominant strategy in VCG auctions.

We now define the valuation function in VCG auctions. When we consider a video streaming system, the user streaming quality can be used to reveal the actual benefit of receiving a certain amount of bandwidth units. We also consider important observations inside real-world peer-assisted streaming systems through extensive measurements [13] [14]. According to [13], there exists a positive correlation between the per-user server bandwidth provisioned and the proportion of users that experience a smooth playback. In the context of independent caching, the streaming quality can therefore be defined as:

$$q_i^k = \gamma(\frac{W_i^k}{r_k \cdot x_i^k})^\alpha \tag{2}$$

In this definition, γ is an adjustable scaling parameter. The channel with streaming rate r_k has x_i^k concurrent users. The value of α satisfies the constraint $0 < \alpha < 1$ [14], which indicates that the streaming quality could be improved when the provisioned bandwidth increases, but with a decreasing marginal gain. This formulation is also consistent with observations in [13], in which the streaming quality negatively correlates to channel populations.

Definition 1: Valuation Function. Based on the formulation of the streaming quality under independent caching, we then propose the valuation function in collaborative caching as:

$$v_i^k(n_{ij}^k) = \gamma(x_i^k)^{1-\alpha}((\frac{W_i^k + e_{ij} \cdot w \cdot n_{ij}^k}{r_k})^\alpha - (\frac{W_i^k}{r_k})^\alpha) \tag{3}$$

The valuation function reflects the streaming quality improvement over channel k by receiving n_{ij}^k units of bandwidth from server j. The notation e_{ij} $(0 < e_{ij} \leq 1)$ represents the degradation of benefits due to the differentiation of coverage areas among different domains of server i and j.

Definition 2: Winner Determination. Each cache server j needs to determine the allocation of W_j^r after receiving bids from other servers. The winner determination is considered to be an *efficient allocation* if it maximizes the social welfare as the following:

$$\begin{aligned} \text{Maximize} \quad & \sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{I}} v_i^k(\tilde{n_{ij}^k}) \\ \text{Subject to:} \quad & \sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{I}} \tilde{n_{ij}^k} \leq \frac{W_j^r}{w} \\ & 0 \leq \tilde{n_{ij}^k} \leq n_{ij}^k \quad \forall i \in \mathcal{I}, k \in \mathcal{K} \end{aligned} \tag{4}$$

$\tilde{n_{ij}^k}$ is the optimization variable in the formulation. Since the valuation function is privately known, the receiving bids only include discrete values for each possible number of acknowledged units. Therefore, this problem becomes an integer programming problem, which is in general NP-hard. It has been further shown that truthfulness is no longer guaranteed in a VCG auction if approximation algorithms are applied [5]. Fortunately, we show in Sec. IV-A that an optimal solution can be achieved here by exploring the unique structure of the valuation function.

Definition 3: VCG Payments. The VCG-based auction mechanism results in a payment for bidder b_i^k as:

$$p_i^k = v_i^k(\tilde{n_{ij}^k}^*) + \max_{\substack{\mathbf{b}_i^k = \\ \{(0,0)\}}} \sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{I}} v_i^k(\tilde{n_{ij}^k}) - \sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{I}} v_i^k(\tilde{n_{ij}^k}^*) \tag{5}$$

The notation $\tilde{n_{ij}^k}^*$ denotes the optimal solution to the winner determination problem (4). The implications of virtual payment are two-fold. First, it acts as an incentive to resource contributors which incur costs in delivering content. Second, the truthfulness of collaborative caching is also guaranteed which is proven in Sec. IV-A.

Definition 4: The utility gained by virtual bidders in a VCG auction is $u_i^k = v_i^k(\tilde{n_{ij}^k}^*) - p_i^k$.

A. Optimal Bandwidth Allocation Strategies

The conventional winner determination in the form of integer programming results in a high computational complexity [5]. However, the valuation function in our proposed mechanisms satisfies a downward sloping property [12], which leads to a polynomial-time optimal allocation.

Theorem 1: The valuation function given in (3) satisfies the downward sloping property.

Proof: The first order derivative of (3) is given as:

$$\frac{\partial v_i^k}{\partial n_{ij}^k} = \frac{\gamma \cdot \alpha \cdot e_{ij} \cdot w (x_i^k)^{1-\alpha}}{{r_k}^{\alpha}(W_i^k + e_{ij} \cdot w \cdot n_{ij}^k)^{1-\alpha}} > 0 \tag{6}$$

This ensures that the valuation monotonically increases with a growing number of provisioned bandwidth units. The second order derivative of the valuation function is given as:

Algorithm 1 A Collaborative Caching Framework through VCG Resource Auctions

```
n_j^r ← ⌊W_j^r / w⌋
if n_j^r ≥ Σ_{k∈K} Σ_{i∈I} n_ij^k then
    ñ_ij^k* ← n_ij^k, ∀i ∈ I, k ∈ K
else
    ñ_ij^k* ← 0, ∀i ∈ I, k ∈ K
    for n_j^r > 0 do
        {i, k} = arg max_{i∈I,k∈K} v_i^k(ñ_ij^k* + 1) − v_i^k(ñ_ij^k*)
        ñ_ij^k* ← ñ_ij^k* + 1
        n_j^r ← n_j^r − 1
    end for
end if
```

$$\frac{\partial^2 v_i^k}{\partial {n_{ij}^k}^2} = \frac{\gamma(\alpha^2 - \alpha)(e_{ij} \cdot w)^2 (x_i^k)^{1-\alpha}}{{r_k}^{\alpha}(W_i^k + e_{ij} \cdot w \cdot n_{ij}^k)^{2-\alpha}} < 0 \tag{7}$$

The second derivative is constantly smaller than 0, which indicates that the valuation function given in (3) satisfies the downward sloping property. ∎

Thus, the seller can then achieve the optimal solution to problem (4) in polynomial time without obtaining any details of the valuation function. The procedure to determine the winner is given in Algorithm 1 with a computational complexity of $\mathcal{O}(|\mathcal{I}||\mathcal{K}|n_j^r)$ for each auction.

Theorem 2: Utility u_i^k gained by virtual bidder b_i^k is a non-negative value.

Proof: Suppose $\tilde{n_{ij}^k}^*$ is determined in the winner determination process. The resulting payment is p_i^k given by Eq. (5). In the optimal solution to the problem of $\max \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{I}} v_i^k(\tilde{n_{ij}^k})$ in case of $\mathbf{b}_i^k = \{(0,0)\}$, $\tilde{n_{ij}^k} = 0 \leq n_{ij}^k$. Therefore, this solution also satisfies the constraints in Problem (4) if remaining bids are not changed. Therefore, its social welfare is smaller than or equal to $\sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{I}} v_i^k(\tilde{n_{ij}^k}^*)$. Then $p_i^k - v_i^k(\tilde{n_{ij}^k}^*) \leq 0$, $u_i^k \geq 0$. ∎

Theorem 3: Bidding on one's true valuation $v_i^k(\tilde{n_{ij}^k}^*)$ maximizes the utility obtained by virtual bidders.

Proof: The utility gained from the true valuation is $u_i^k = \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{I}} v_i^k(\tilde{n_{ij}^k}^*) - \max_{\mathbf{b}_i^k=\{(0,0)\}} \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{I}} v_i^k(\tilde{n_{ij}^k})$. Consider that b_i^k submits a false valuation such that ${v'}_i^k(\tilde{n_{ij}^k}) \neq v_i^k(\tilde{n_{ij}^k})$. In this case, the utility gained from the false valuation is ${u'}_i^k = \max \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{I}} {v'}_i^k(\tilde{n_{ij}^k}) - {v'}_i^k(\tilde{n'_{ij}^k}^*) - \max_{\mathbf{b}_i^k=\{(0,0)\}} \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{I}} v_i^k(\tilde{n_{ij}^k}) + v_i^k(\tilde{n'_{ij}^k}^*)$, where $\tilde{n'_{ij}^k}^*$ is the optimal allocation to $\max \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{I}} {v'}_i^k(\tilde{n_{ij}^k})$. The utility differentiation between two scenarios is denoted as $u_i^k - {u'}_i^k = (\sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{I}} v_i^k(\tilde{n_{ij}^k}^*) - v_i^k(\tilde{n'_{ij}^k}^*)) - (\max \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{I}} {v'}_i^k(\tilde{n_{ij}^k}) - {v'}_i^k(\tilde{n'_{ij}^k}^*))$. Since $\tilde{n_{ij}^k}^*$ is an efficient allocation to problem (4) in the context of true valuations, the authenticity of remaining bids ensures that $u_i^k - {u'}_i^k \geq 0$. Therefore, we have $u_i^k \geq {u'}_i^k$. ∎

All bids submitted by virtual bidders are therefore truthful according to Theorem 2 and 3. Meanwhile, the demands with higher valuations are satisfied with higher priorities, which optimize the social welfare. The bidding strategy is simplified, which benefits the practical deployment of the mechanisms.

B. Implementation Issues

First, local resource allocation. In each trading round, a fraction of β ($0 \leq \beta \leq 1$) of the total bandwidth capacity is used for the local content dissemination such that $\sum_{k\in\mathcal{K}} W_i^k = \beta W_i$. Since the status of local channels is always known to the respective WSP, the decision of W_i^k is also based on the valuation function (3) to optimize the performance.

Second, the determination of n_{ij}^k. The decision of n_{ij}^k is determined by the remaining budget in auctions. The cache server formulates an optimization problem that minimizes the number of requested units $\sum_{k\in\mathcal{K}} n_{ij}^k$, where the constraint is given as the total valuation of requested units $\sum_{k\in\mathcal{K}} v(n_{ij}^k)$ being greater than the remaining budget. Demands with higher valuations can be found through a local search in polynomial time, and requests are therefore issued.

Third, server selection for issuing requests. This decision is made based on the external bandwidth availability and the content availability, which can be acquired through information exchange. Those cache servers with more available bandwidth W_j^r are preferred to be selected.

Fourth, the frequency of trading. If the trade is performed in each time slot, the resource allocation is achieved with finer granularity but excessive computational overhead. To mitigate such overhead, the trading result can be potentially used for a longer period of time. The effect of changing the trading frequency is evaluated in Sec. VI.

V. Collaborative Caching Storage

One of the main benefits of applying collaborative caching in wireless video streaming is the ability of requesting locally unavailable contents which improves the storage utilization through resource multiplexing. Given bandwidth resource auctions introduced in Sec. IV, the limited cache storage should also be carefully utilized. Without loss of generality, cache servers are required to keep full copies of video channels. The storage decision of cache server i can then be denoted as $a_i^k = \{0,1\}, \forall i \in \mathcal{I}, k \in \mathcal{K}$. In this section, we analyze the effect of different cache storage strategies under both independent and collaborative caching mechanisms.

A. Performance Metric

The bandwidth resource of cache servers cannot be assumed to be abundant. Even if a video is cached, there is no guarantee that the currently available bandwidth is sufficient to satisfy all the content requests. Therefore, the video request can be counted as a cache hit only when the video is cached and served with allocated bandwidth. Accordingly, we define the performance metric of *achieved cache hit ratio* as:

$$\mathcal{R} = \frac{\sum_{k\in\mathcal{K}} \phi \cdot x_i^k}{\sum_{k\in\mathcal{K}} x_i^k} \quad \forall i \in \mathcal{I} \tag{8}$$

in which ϕ represents a binary value such that:

$$\phi = \begin{cases} 1, & \text{If } (\frac{W_i^k}{w} + \sum_{j\in\mathcal{I}} \tilde{n_{ij}^k}^*) > 0, \\ 0, & \text{Otherwise.} \end{cases} \tag{9}$$

The achieved cache hit ratio defined in Eq. (8) is the proportion of online users whose content requests are served either by a local WSP through local resource allocation, or external WSPs through resource auctions.

B. Storage Update Strategies

Under both independent and collaborative caching mechanisms, the periodical storage update aims to improve the overall streaming quality in the system. It can be observed from Eq. (3) that it is more beneficial to assign bandwidth resource to popular channels. Therefore, conventional independent caching prefers to cache highly popular videos as:

$$\begin{aligned} \text{Maximize} \quad & \sum_{k\in\mathcal{K}} a_i^k x_i^k \\ \text{Subject to:} \quad & \sum_{k\in\mathcal{K}} a_i^k f_k \leq S_i \\ & a_i^k = \{0,1\} \quad 1 \leq k \leq N_k \end{aligned} \tag{10}$$

Here f_k denotes the file size of video k and S_i represents the storage capacity of the cache server i. In practice, the statistical result of video popularity over the last storage update interval is used for the update decision.

The storage update decision needs to further coordinate with resource auctions with the proposed collaborative caching mechanism. Cache servers update their contents based on the local requests and the demands in the current market, as both the locally and socially valuable content should be stored. The local demands are reflected by the local resource allocation W_i^k and the content requests n_{ij}^k. The prediction of market demands is to take historical trading information into consideration. Thus, the socially valuable content can be represented by the number of bandwidth units satisfied to other WSPs $\tilde{n_{ji}^k}^*$. The basic utility maximization formulation is then given as:

$$\begin{aligned} \text{Maximize} \quad & \sum_{k\in\mathcal{K}} (\lambda a_i^k (\frac{W_i^k}{w} + \sum_{j\in\mathcal{I}} n_{ij}^k) + (1-\lambda) a_i^k \sum_{j\in\mathcal{I}} \tilde{n_{ji}^k}^*) \\ \text{Subject to:} \quad & \sum_{k\in\mathcal{K}} a_i^k f_k \leq S_i \\ & a_i^k = \{0,1\} \quad 1 \leq k \leq N_k \end{aligned} \tag{11}$$

Here λ ($0 \leq \lambda \leq 1$) specifies the design choice for the storage update strategy in the collaborative caching mechanism. If we set $\lambda = 1$, the storage decision is made solely based on the local demands. If we set $\lambda = 0$, the storage update strategy is fully dominated by demands in the current market. When we set λ as a certain value between 0 and 1, it represents a combined strategy that benefits both demands of the local WSP and content requests from other WSPs.

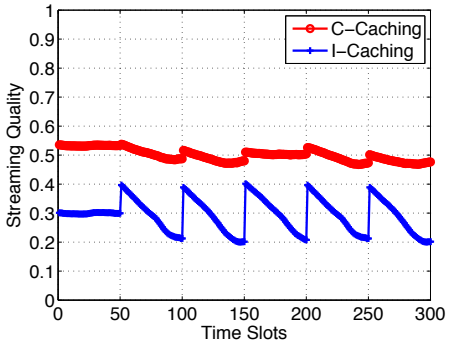


Fig. 2. The streaming quality of collaborative caching versus independent caching.

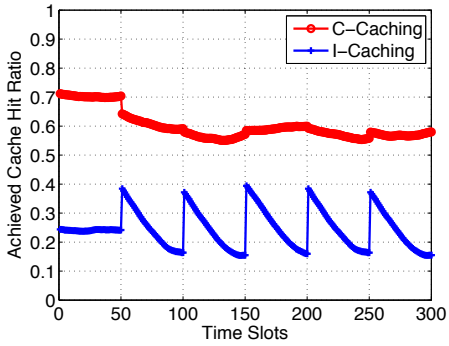


Fig. 3. The achieved cache hit ratio of collaborative caching versus independent caching.

C. Impact of Storage Update Interval

The video popularity throughout the entire system dynamically changes over time. In this case, the cache server with an independent caching mechanism needs to frequently update its storage to cache the most popular content, in order to achieve an efficient use of bandwidth resources. However, frequent updates would cause considerable cost in distributing content from the centralized content server to cache servers.

Such costs can be significantly reduced in our proposed collaborative caching by enabling content delivery among different WSPs. The diversified contents among multiple WSPs can help satisfy requests for videos that are not locally available even with a long interval of storage update. This cost reduction is illustrated in our performance evaluation section.

VI. Performance Evaluation

In this section, we examine the performance of the proposed collaborative caching scheme, in comparison with the conventional independent caching mechanism. Our evaluation is based on a time-slotted simulator implemented using Python. The performance of different caching mechanisms is analyzed in terms of the overall streaming quality, *i.e.*, the percentage of online users that experience a smooth playback, and the achieved cache hit ratio, defined in Sec. V. We also take a close look at the potential cost and overhead of collaborative caching, which is critical in practical system design.

We simulate four WSPs that deploy cache servers to assist wireless video streaming. There are 500 video channels and the peak popularity is distributed over 10 to 500. The bandwidth resource auctions are carried out periodically according to a predefined allocation update interval. The storage space of cache servers is randomly filled at the initial phase, and then periodically updated with different strategies in Sec. V.

A. Performance Improvement through Collaborative Caching

We first plot the overall streaming quality improvement brought forth by the proposed collaborative caching (denoted as C-Caching) in Fig. 2. The allocation update interval is one time slot, and the storage update interval is 50 time slots. The size of the bandwidth unit in the auction is the same as the video streaming rate of 500 kbps. Each cache server can store 25% of all the existing video channels, and the bandwidth capacity is set to $16,000$ bandwidth units which account for 20% of the total bandwidth requirement in the system. The value of parameter β and λ is set to 0.6 and 0.5, respectively.

We first observe that the overall streaming quality of C-Caching consistently outperforms that of the independent caching (denoted as I-Caching). Due to the random initialization of each cache storage, the streaming quality of C-Caching and I-Caching stay stable at around 50% and 30%, respectively. After the first storage update, the streaming quality of I-Caching increases to 40%, as more bandwidth units are utilized for supporting channels with high valuation. However, the streaming quality of C-Caching does not explicitly increase since the resource auctions already fulfill the demands for those unavailable popular videos before the storage update. In subsequent time slots, the streaming quality of I-Caching fluctuates between 20% and 40%, while C-Caching always maintains a satisfactory level of performance. This demonstrates the stability of the proposed collaborative caching mechanism.

Similar performance patterns can be observed in the achieved cache hit ratio as plotted by Fig. 3. On average, the achieved cache hit ratio of C-Caching is almost 3 times higher than that of I-Caching. The rationale is that I-Caching prefers to assign bandwidth resources to the most popular videos to achieve a better overall performance. As such, only a smaller portion of videos can be satisfied by deployed cache servers. In comparison, the diversified content exchanges under C-Caching and the dedicated bandwidth resource in resource auctions could help maintain a higher cache hit ratio and relatively stable streaming quality.

Fig. 4 compares different scenarios of the bandwidth and storage capacity settings. Along with an increase in storage capacity, the streaming quality under I-Caching increases almost linearly since the negative impact of the content unavailability brought forth by system dynamics can be alleviated. In comparison, the streaming quality improvement under C-Caching is initially significant and becomes less significant if the storage capacity exceeds 40%. This clearly demonstrates that collaborative caching through multiplexing can substantially reduce the overall storage requirement to achieve a desirable streaming quality.

Fig. 5 depicts the impact of the portion of bandwidth resources that are allowed to be utilized by other WSPs.

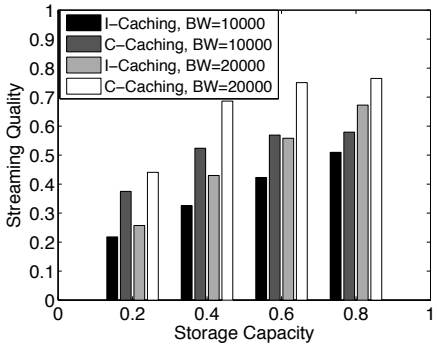


Fig. 4. The streaming quality of collaborative caching versus independent caching, under different settings of bandwidth and storage capacity.

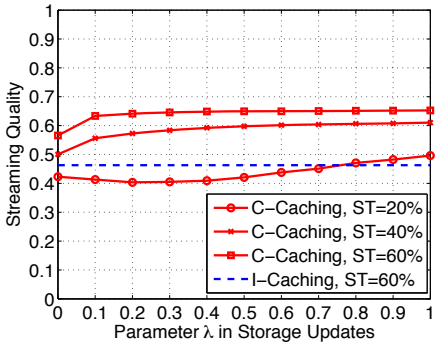


Fig. 6. The streaming quality versus the parameter λ in the cache storage update, under different settings of storage capacities and caching mechanisms.

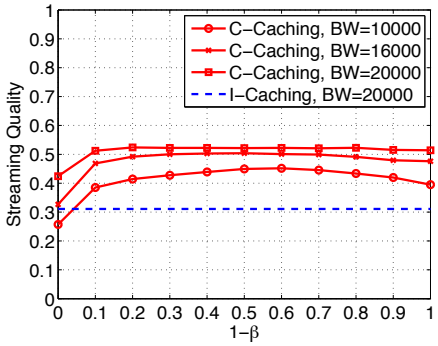


Fig. 5. The streaming quality versus the percentage of bandwidth provisioned to other WSPs, under different settings of bandwidth capacities and caching mechanisms.

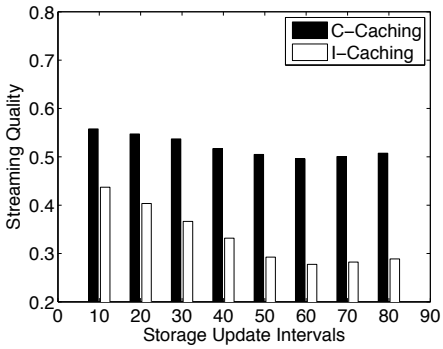


Fig. 7. The streaming quality versus the cache storage update interval, under different settings of caching mechanisms.

It can be observed that the streaming quality is maximized when an adequate portion of bandwidth is allocated for other WSPs (*i.e.*, for auctions), and both excessive local and external bandwidth provision in C-Caching cannot yield the maximal streaming quality. The reason is that when β is small, most of content requests from other WSPs can be satisfied even with a low valuation of the bandwidth units. On the contrary, if the value of β is as large as 1, cache servers can no longer benefit from resource multiplexing across different WSPs, in cases of popularity variation or content unavailability. Fig. 5 suggests that we can properly design the β to achieve better streaming quality in practical engineering.

Fig. 6 denotes the effect of parameter λ in the storage update under different storage capacity settings. We observe that a local demands first strategy is preferred when the storage capacity is limited (ST$=20\%$) in C-Caching. However, the overall content diversity is satisfactory by caching respective popular contents, which can be concluded from Fig. 3 that exhibits a satisfactory level of the achieved cache hit ratio. The sensitivity of λ decreases when the storage capacity is relatively abundant, since both locally preferred and socially preferred contents could be kept with a larger storage space, regardless of specific settings of λ.

B. Overhead with Collaborative Caching

We next examine the overhead incurred by the proposed C-Caching by analyzing the impact of the update interval of cache storage and that of bandwidth allocation. Fig. 7 illustrates the tradeoff between the streaming quality and the frequency of cache storage updates. It can be seen that the performance gap between the two mechanisms becomes more significant when the update interval gradually gets larger. The increasing gap is caused by fluctuations in channel popularity, which can easily be resolved with inter-WSP resource auctions. This shows that I-Caching has to frequently update its storage which brings a higher cost associated with the cache replacement. In sharp contrast, the performance of C-Caching remains stable despite specific cache update strategies.

We then analyze the tradeoff between the streaming quality and the update interval of bandwidth resource allocation in Fig. 8. We evaluate several scenarios in which decisions of bandwidth allocation from resource auctions are utilized over multiple rounds. Fig. 8 clearly shows that the overall streaming quality only slightly decreases along with an increase of the allocation update interval. This implies that in practice, it is feasible to reuse existing bandwidth allocation results, which further reduces the overhead of auction mechanisms.

C. Fairness among WSPs

When the auction is conducted in practice among multiple WSPs, it is indispensable to further evaluate the fairness of the system. Fig. 9 plots the respective streaming quality of different WSPs in our simulation. It shows that the variation of streaming quality at WSPs exhibits a similar pattern over time. This demonstrates that the proposed resource auction can improve the social welfare by mutual assistance. Fig. 10

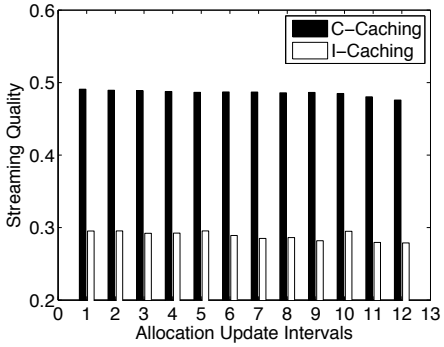


Fig. 8. The streaming quality versus the update interval of bandwidth allocation, under different settings of caching mechanisms.

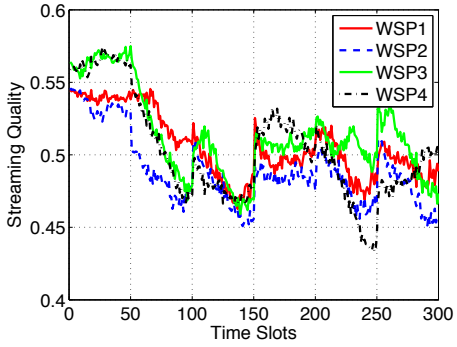


Fig. 9. The streaming quality of different WSPs in collaborative caching over time.

further plots the cumulative amount of received packets from other WSPs. It can be observed that the trend of all curves remains similar overall, which also reflects the fairness of the collaborative caching mechanism.

VII. Conclusion

In this paper, we have proposed a collaborative caching mechanism for wireless video streaming that coordinates cache resource provisioning among selfish WSPs. In particular, we focus on engineering the incentives to promote and encourage cache servers from different WSPs to truthfully cooperate with one another in the context of VCG auctions. We have derived solutions to allocate server bandwidth and analyzed the utilization of server storage space. Simulation results demonstrate that the performance of streaming systems can be significantly improved, with maximized social welfare in auctions conducted by the collaborative caching mechanism.

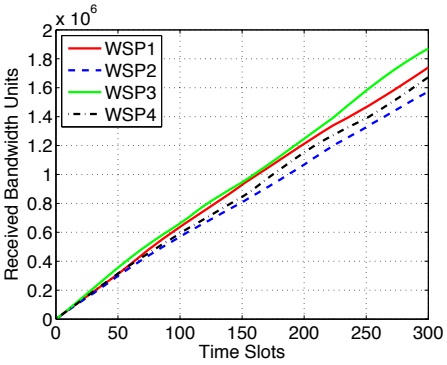
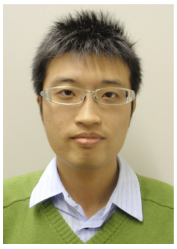


Fig. 10. The number of received packets from other WSPs in collaborative caching over time.

References

[1] F. Khan, *LTE for 4G Mobile Broadband: Air Interface Technologies and Performance*. Cambridge University Press, 2009.
[2] Q. Zhang, Z. Xiang, W. Zhu, and L. Gao, "Cost-Based Cache Replacement and Server Selection for Multimedia Proxy across Wireless Internet," *IEEE Transactions on Multimedia*, vol. 6, no. 4, 2004.
[3] H. Chen and Y. Xiao, "Cache Access and Replacement for Future Wireless Internet," *IEEE Communications Magazine*, vol. 44, no. 5, 2006.
[4] "Explanation of Optimization Deployment," http://www.verizonwireless.com/vzwoptimization/.
[5] N. Nisan and A. Ronen, "Computationally Feasible VCG Mechanisms," *Journal of Artificial Intelligence Research*, vol. 29, no. 1, 2007.
[6] Q. Zhang, F. Yang, and W. Zhu, "Cross-layer QoS Support for Multimedia Delivery over Wireless Internet," *EURASIP J. Appl. Signal Process.*, vol. 2005, no. 2, 2005.
[7] E. Tan, L. Guo, S. Chen, and X. Zhang, "SCAP: Smart Caching in Wireless Access Points to Improve P2P Streaming," in *Proc. IEEE ICDCS*, Jun. 2007.
[8] J. Dai, B. Li, F. Liu, B. Li, and H. Jin, "On the Efficiency of Collaborative Caching in ISP-aware P2P Networks," in *Proc. IEEE INFOCOM*, Apr. 2011.
[9] L. Chen, M. Meo, and A. Scicchitano, "Caching Video Contents in IPTV Systems with Hierarchical Architecture," in *Proc. IEEE ICC*, Jun. 2009.
[10] S. Borst, V. Gupta, and A. Walid, "Distributed Caching Algorithms for Content Distribution Networks," in *Proc. IEEE INFOCOM*, Mar. 2010.
[11] A. T. S. Ip, J. C. S. Lui, and J. Liu, "A Revenue-rewarding Scheme of Providing Incentive for Cooperative Proxy Caching for Media Streaming Systems," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 4, no. 1, 2008.
[12] S. Dobzinski and N. Nisan, "Mechanisms for Multi-Unit Auctions," *Journal of Artificial Intelligence Research*, vol. 37, no. 1, 2010.
[13] C. Wu, B. Li, and S. Zhao, "Diagnosing Network-wide P2P Live Streaming Inefficiencies," in *Proc. IEEE INFOCOM*, Apr. 2009.
[14] C. Wu and B. Li, "Multi-channel Live P2P Streaming: Refocusing on Servers," in *Proc. IEEE INFOCOM*, Apr. 2008.

Jie Dai is currently a Ph.D. student in the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. Prior to that, he received his B.Eng. degree in computer science from Huazhong University of Science and Technology, Wuhan, China, in 2007. From April to August 2011, he was a visiting student at the School of Computing Science, Simon Fraser University, Canada. His research interests include collaborative caching systems, ISP networks, cloud computing, peer-to-peer networks, and multimedia systems. He is a student member of IEEE.



Fangming Liu received his B.Engr. degree in 2005 from Department of Computer Science and Technology, Tsinghua University, Beijing, China; and his Ph.D. degree in Computer Science and Engineering from the Hong Kong University of Science and Technology in 2011. He is currently an Associate Professor in the Services Computing Technology and System Lab, Cluster and Grid Computing Lab, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. From Aug 2009 to Feb 2010, he was a visiting student at the Computer Engineering Group, Department of Electrical and Computer Engineering, University of Toronto, Canada. His research interests are in the area of peer-to-peer networks, rich-media distribution, cloud computing and large-scale datacenter networking. He is a member of IEEE and IEEE Communications Society.

Bo Li (F'11) is a professor in the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. He is a Cheung Kong Chair Professor in Shanghai Jiao Tong University, China, and the Chief Technical Advisor for China Cache Corp. (NASDAQ:CCIH). He was previously with IBM Networking System Division, Research Triangle Park, and an adjunct researcher at Microsoft Research Asia. His recent interests include: large-scale content distribution in the Internet, Peer-to-Peer media streaming, the Internet topology, cloud computing, green computing and communications.

He received his B. Eng. Degree in the Computer Science from Tsinghua University, Beijing, and his Ph.D. degree in the Electrical and Computer Engineering from University of Massachusetts at Amherst.

Baochun Li received the B.Engr. degree from the Department of Computer Science and Technology, Tsinghua University, China, in 1995 and the M.S. and Ph.D. degrees from the Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, in 1997 and 2000. Since 2000, he has been with the Department of Electrical and Computer Engineering at the University of Toronto, where he is currently a Professor and the Bell Canada Endowed Chair in Computer Engineering. His research interests include cloud computing, peer-to-peer networks, applications of network coding, and wireless networks. He was the recipient of the IEEE Communications Society Leonard G. Abraham Award in the Field of Communications Systems in 2000. In 2009, he was a recipient of the Multimedia Communications Best Paper Award from the IEEE Communications Society, and a recipient of the University of Toronto McLean Award. He is a member of ACM and a senior member of IEEE.

Jiangchuan Liu (S'01-M'03-SM'08) received the BEng degree (cum laude) from Tsinghua University, China, in 1999, and the PhD degree from The Hong Kong University of Science and Technology in 2003. He is a co-recipient of the Best Student Paper Award of IWQoS'2008 and the Multimedia Communications Best Paper Award from the IEEE Communications Society.

He is currently an Associate Professor at Simon Fraser University, British Columbia, Canada, and was an Assistant Professor at The Chinese University of Hong Kong from 2003 to 2004. His research interests include cloud computing, peer-to-peer systems, multimedia communications, and wireless networking. He is an Associate Editor of IEEE Transactions on Multimedia, and an editor of IEEE Communications Surveys and Tutorials.