On the Interaction between Content Caching and Request Assignment in Cellular Cache Networks

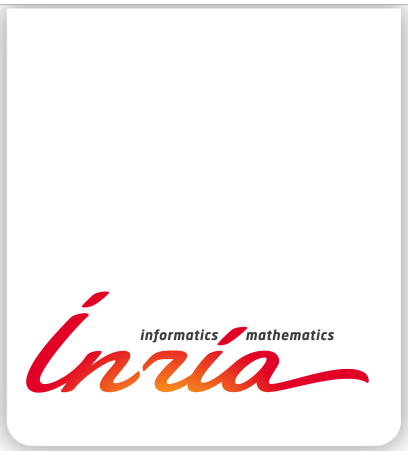
Naveen Kolar Purushothama, Laurent Massoulie, Emmanuel Baccelli, Aline Carneiro Viana, Don Towsley

On the Interaction between Content Caching and Request Assignment in Cellular Cache Networks

Naveen Kolar Purushothama, Laurent Massoulie, Emmanuel Baccelli, Aline Carneiro Viana, Don Towsley

RESEARCH REPORT
N° 8707
March 2015
Project-Team Infine

ISSN 0249-6399 ISRN INRIA/RR--8707--FR+ENG



On the Interaction between Content Caching and Request Assignment in Cellular Cache Networks

Naveen Kolar Purushothama, Laurent Massoulie, Emmanuel Baccelli, Aline Carneiro Viana, Don Towsley

Équipe-Projet Infine

Rapport de recherche n° 8707 — version — version initiale March 2015 — version révisée March 2015 — 21 pages

Résumé :

Mots-clés :

RESEARCH CENTRE
SACLAY – ÎLE-DE-FRANCE

1 rue Honoré d'Estienne d'Orves
Bâtiment Alan Turing
Campus de l'École Polytechnique
91120 Palaiseau

On the Interaction between Content Caching and Request Assignment in Cellular Cache Networks

Abstract: The potential availability of storage space at cellular and femtocell base-stations (BSs) raises the following question: How should one optimize performance through both load balancing and content replication when requests can be sent to several such BSs? We formally introduce an optimization model to address this question and propose an online algorithm for dynamic caching and request assignment. Crucially our request assignment scheme is based on a server price signal that jointly reflects content and bandwidth availability. We prove that our algorithm is optimal and stable in a limiting regime that is obtained by scaling the arrival rates and content chunking. From an implementation standpoint, guided by the online algorithm we design a light-weight scheme for request assignments that is based on load and cache-miss cost signals; for cache replacements, we propose to use the popular LRU (Least Recently Used) strategy. Through simulations, we exhibit the efficacy of our joint-price based request assignment strategy in comparison to the common practices of assigning requests purely based on either bandwidth availability or content availability.

Key-words: Cellular cache networks, fluid approximations, Lyapunov stability, LRU caches.

1 Introduction

Mobile data traffic is expected to increase by more than 57% annually to 24.3 exabytes per month by 2019 [1]. This increase will likely result in durably congested wireless access at the edge despite advances in radio technologies. Congestion is also anticipated at the core, driven not only by mobile Internet access but also by the emergence of new cloud services and Machine-to-Machine (M2M) communications.

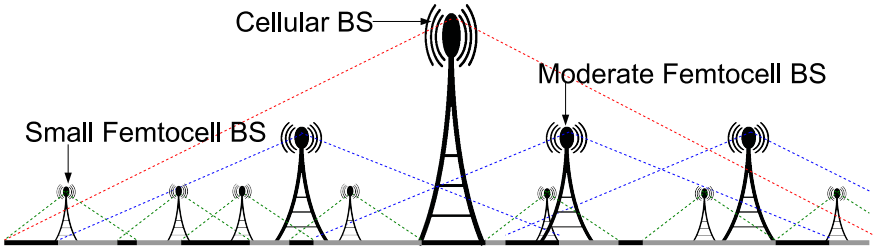


FIGURE 1 – A cellular network with a dense deployment of heterogeneous base-stations (BS).

To alleviate congestion at the Internet edge, one promising approach is to target denser deployments of wireless base-stations (BSs), such as cache enabled femtocells in addition to existing cellular BSs [2, 3]. Such a scenario is depicted in Figure 1, where the alternating black and gray horizontal sections represent regions that are within the range of distinct subset of BSs. Thus, mobile users, depending on their respective locations, will be potentially connected to several BSs from which content may be directly downloaded. In this context, BSs can be heterogeneous in terms of bandwidth and memory capacities. For instance, a nearby femtocell BS will provide more bandwidth than the heavily loaded cellular BS ; however, the chance of finding a desired content at the femtocell is lower due to its lower cache capacity.

Thus, both bandwidth and content availability critically affect "by which BS" and "with which performance" incoming requests for contents can be served. In turn, content request assignment strategies impact targeted BSs through both their bandwidth loads and cache contents as they trigger cache update decisions. The goal of this paper is to understand this intricate interplay between content request assignment strategies, bandwidth load, and cache management in cellular cache networks. Specifically, we strive to determine routing mechanisms and associated cache update rules that together achieve a target trade-off between cache-miss probability and bandwidth loads.

Related Work : The problem of caching at femtocell base stations or heterogeneous caches has been recently studied in [2–4]. Authors in [3] address content placement at the femtocell base stations to maximize the probability of cache hit. A similar setup, but with mobile users, is studied in [2]. In both these works, request routing is purely based on content availability (so that cache hits are maximized), while ignoring the bandwidth costs at the base stations. Further, the authors focus on static content placement algorithms. For instance, in [3] the content placement problem is formulated as that of maximizing a monotone submodular function under matroid constraints ; a static greedy algorithm that is within a constant factor of the optimal is proposed.

Borst et al. in [5] address the problem of content placement and routing to optimize the total bandwidth cost, but their formulation is limited to linear cost functions, in contrast to our work where we allow for general convex costs. Moreover, the focus in [5] is again on designing static algorithms. Similarly, there are other works in the literature addressing the problem of static content placement so as to optimize some linear system cost [6–8].

There is an extensive literature on online cache management algorithms and in particular on LRU [9–11]. There is also an abundant literature on dynamic request routing [12, 13], but very few works address the design of online algorithms for both routing and cache management, and the interaction between the two.

Notable exceptions are the works on cache networks [14,15], which focus on stabilizing queues of pending requests by joint management of bandwidth and caches. In contrast we instead focus on loss-based models and on the objective of cost minimization rather than queue stabilization.

With a similar focus as ours, the authors in [16] study the problem of cache replication and request forwarding in CDNs comprising distributed caches, e.g., household set-top boxes. However, the objective is again limited to linear traffic costs. Moreover, they focus on scenarios with many servers with identical characteristics. This is in contrast with our work as we consider a system with few potentially heterogeneous caches; our limiting regime is instead obtained by scaling the arrival rates and the chunk size. There are other work in the literature, e.g., [17, 18], where proportional placement strategies and matching algorithms for routing requests to content available caches are studied. These differ again from our present work by focusing on scenarios with many identical servers.

The remainder of the paper is organized as follows. We detail our system model in Section 2. In Section 3 we describe an online algorithm for dynamic caching and request assignments, and state our theoretical result on the algorithm's performance. Simulation results are presented in Section 4. Proof of our main result is available in Appendix A.

2 System Model

Consider a system comprising a finite collection of *base-stations (BSs)* $\mathcal{S}$ and *locations* $\mathcal{L}$ where a location $\ell \in \mathcal{L}$ can be thought of as representing the set of all physical locations within the range of a subset of BSs. For instance, in Figure 1 the alternating black and grey shaded horizontal segments represents different locations. The interconnection between the BSs and the locations can be represented using an adjacency matrix A, where $A_{\ell s} = 1$ implies that location ℓ is within the range of BS s; $A_{\ell s} = 0$ otherwise. We assume a finite set $\mathcal{C}$ of *contents* in the system with the size of each content being one unit. We allow *content chunking* whereby a content can be completely reconstructed using k independently coded packets of the content. Each BS $s \in \mathcal{S}$ includes a cache that can store up to M_s units of contents (*memory constraints*). We allow the BSs to store chunks of each content instead of complete copies. We assume that the chunks stored at different servers are always independent so that it only suffices to download k chunks from any subset of BSs to completely reconstruct the content.

Requests for content c arrive at location ℓ according to a Poisson process with rate $\lambda^{(k)}_{\ell c} = k\lambda_{\ell c}$, which can be thought as the aggregate rate for c from of all users stationed at ℓ. The number of chunks of content c, available at server s, restricts the rate at which s can serve incoming requests for c (*content-availability constraints*). Content download requests, if accepted, result in service at unit rate. In this sense we consider a loss model (in contrast to a queuing model).

The bandwidth availability at a BS $s \in \mathcal{S}$ is reflected using a *bandwidth-cost function*, $C^{(k)}_s(r)$, which represents the cost incurred by s for serving requests at a total rate of r. An example bandwidth-cost function can be of the form

$$C^{(k)}_s(r) = \exp\Big\{a\Big(\frac{r - R^{(k)}_s}{k}\Big)\Big\}, \tag{1}$$

the possible motivations for which are the following :

(M1) One motivation is to assume that $R_s^{(k)} = kR_s$ is the recommended rate at which server s is designed to operate; $C_s^{(k)}(r)$ is then the price charged by s for serving at rate r. Thus, the price remains low if the BS is underloaded (i.e., $r < R_s^{(k)}$), while increasing exponentially in r if the BS operates in the overloaded regime ($r > R_s^{(k)}$).

(M2) A more formal motivation can be deduced in terms of incurred transmission costs : Let $C_s^{(k)}(r)$ denote the transmission power required by BS s to achieve a transmission rate of r units. Assume that BS s is at a fixed distance d_s from all connected locations. Then, we have the following (large SNR) relation [19] : $r = W^{(k)} \log(\frac{C_s^{(k)}(r) d_s^{-\eta}}{N_0})$ where $W^{(k)} = kW$ is the available bandwidth at BS s, $\eta > 2$ is the pathloss exponent, and N_0 is the receiver noise variance. Thus, $C_s^{(k)}(r)$ can now be expressed as in (1) with $a = \frac{1}{W}$ and $R_s^{(k)} = -kW \log(N_0 d_s^{\eta})$.

Note that the recommended rate, $R_s^{(k)}$ in (M1) and the available bandwidth $W^{(k)}$ in (M2) scale with the arrival rates, $\lambda_{\ell c}^{(k)}$. We do not restrict ourselves to the example cost function in (1) and consider general bandwidth-cost functions satisfying We assume that $C_s^{(k)}$, $k \geq 1$, can be expressed as $C_s^{(k)}(r) = C_s(\frac{r}{k})$, where C_s, $s \in \mathcal{S}$, is strictly convex and increasing. We refer to the above system as the *k-th system*.

To obtain a benchmark for performance comparison, we consider a system where the caching and routing variables are relaxed to be reals. Let $\theta_{sc} \in [0,1]$ denote the fraction of content c stored at BS s, and $kr_{s\ell c}$ the rate at which s serves requests for c from ℓ. Then, the memory constraint at BS $s \in \mathcal{S}$ is given by $\sum_c \theta_{sc} \leq M_s$. Next, since only a fraction θ_{sc} of a request for c can be served by s, we have the following content availability (CA) constraints : $kr_{s\ell c} \leq A_{\ell s} k \lambda_{\ell c} \theta_{sc}$, $s \in \mathcal{S}, \ell \in \mathcal{L}, c \in \mathcal{C}$, where we have approximated the number of ongoing requests (of content c at ℓ) by the steady state arrival rate, $k\lambda_{\ell c}$ (assuming an average service time of one).

In order to obtain an unconstrained optimization formulation, we replace the above hard memory and CA constraints with strictly convex and increasing penalty functions, $\widehat{C}_s$ and $C_{s\ell c}$, in the cost to be minimized (formulation directly involving the hard constraints is studied in Appendix B). $\widehat{C}_s(\theta)$ denotes the cost of storing θ units of contents at BS s. For the CA constraint, denoting $x := kr_{s\ell c} - k\lambda_{\ell c}\theta_{sc}$, $C_{s\ell c}(\frac{x}{k})$ represents the cost of respecting (x negative) the CA constraint, or exceeding the constraint (x positive) by x units. Since the BSs are physically constrained to always satisfy the CA constraints, a positive x represents the content cache-miss rate from BS s for requests from ℓ for c. Thus, $C_{s\ell c}(\frac{x}{k})$ represents the cost incurred for fetching the missed content rate from the main data center at the backend. Another possibility is to let $C_{s\ell c}(\frac{x}{k}) = b \max(\frac{x}{k}, 0)$ for some $b > 0$, which can be interpreted as a cost of b per cache miss.

We now define the *total cost* incurred as,

$$C(\{\theta_{sc}\}, \{r_{s\ell c}\}) = \sum_s C_s\Big(\sum_{\ell c} A_{\ell s} r_{s\ell c}\Big) + \sum_s \widehat{C}_s\Big(\sum_c \theta_{sc}\Big) + \sum_{s\ell c} A_{\ell s} C_{s\ell c}\Big(r_{s\ell c} - \lambda_{\ell c}\theta_{sc}\Big) \qquad (2)$$

and propose the following optimization problem :

Total Cost (TC) :

$$\begin{aligned}
\text{Minimize :} \quad & C(\{\theta_{sc}\}, \{r_{s\ell c}\}) && \text{(3a)}\\
\text{Over :} \quad & \theta_{sc} \in [0,1], s \in \mathcal{S}, c \in \mathcal{C} && \text{(3b)}\\
& r_{s\ell c} \geq 0, s \in \mathcal{S}, \ell \in \mathcal{L}, c \in \mathcal{C} && \text{(3c)}\\
\text{Subject to :} \quad & \sum_s A_{\ell s} r_{s\ell c} = \lambda_{\ell c}, \ell \in \mathcal{L}, c \in \mathcal{C}. && \text{(3d)}
\end{aligned}$$

The equality constraint (3d) is used to ensure that all incoming requests are handled by the BSs.

In Section 3, we propose an *online algorithm* to solve this optimization problem. We show that our algorithm converges to the optimal solution of (3) as the scale of the system in terms of arrival rate and number of chunks per content (i.e., k) increases.

3 Online Algorithm

The main objective of this section is to propose an algorithm for assigning incoming requests to BSs based on their current "prices" ; the key idea is that these prices reflect both the load as well as the content availability at the respective BSs (which is in contrast to the common practice of assigning requests purely based on either load or content availability).

3.1 Cache Update and Request Assignment

Recall the k-th system described in Section 2, with arrival rates $k\lambda_{\ell c}$ of requests for c at ℓ, and k chunks per content. Let $\Theta_{sc}(t)$ denote the number of chunks of content c stored at BS s at time t, and $G_{slc}(t)$ the total number of chunks of content c served (possibly for more than one request) from s to ℓ at time t. Thus, $\Theta_{sc}(t)/k$ is the fraction of content c stored at s at time t, and $R_{slc}(t) := G_{slc}(t)/k$ is the rate at which s serves requests for c from ℓ. We refer to $X_k(t) := (\{\Theta_{sc}(t)\}, \{R_{s\ell c}(t)\})$ as the system state at time t. The chunk service times are i.i.d (independent and identically distributed) exponential random variables with unit rate.

Cache Updates : Updating a BS's cache for a content involves either adding new chunks or removing some existing chunks. In the k-th system, let $k\nu_{sc}$ denote the rate at which BS s updates the cache for content c. Suppose BS s initiates a cache update for c at time t. Then, the number of chunks that are added or removed is proportional to

$$\Delta\Theta_{sc}(t) \quad = \quad -Q_s(t) + \sum_{\ell} A_{\ell s}\lambda_{\ell c}P_{s\ell c}(t), \tag{4}$$

where (using F' to denote the derivative of F),

$$Q_s(t) \quad := \quad \widehat{C}'_s\Big(\sum_c \frac{\Theta_{sc}(t)}{k}\Big) \tag{5}$$

$$P_{s\ell c}(t) \quad := \quad C'_{s\ell c}\Big(\frac{R_{s\ell c}(t)}{k} - \lambda_{\ell c}\frac{\Theta_{sc}(t)}{k}\Big). \tag{6}$$

Formally, $\Theta_{sc}(t)$ is updated as follows : $\Theta_{sc}(t^+) = \Theta_{sc}(t) + [\epsilon\Delta\Theta_{sc}(t)]$, where t^+ denotes the time instant just after t, $\epsilon > 0$ is a constant, and the notation $[x]$ represents the integer closest to x. We refer to $Q_s(t)$ and $P_{s\ell c}(t)$, respectively, as the *memory price* and the *content availability (CA) price* (w.r.t (ℓ, c)) of BS s at time t.

Remark : For the sake of analysis, we assume that the caches are updated instantaneously at the update instants, although in practice a small time is required to download new chunks from a backend data center. We further assume that the backend data center always supplies a new batch of independently coded chunks, so that it is possible to reconstruct a content by downloading k chunks from any subset of BSs, without worrying about chunks being dependent.

Note that $\Delta\Theta_{sc}(t)$ is simply the negative of the gradient (w.r.t θ_{sc}) of the cost function in (2). Thus, our cache update strategy is essentially the gradient descent algorithm. The novelty of our work appears next, where we propose a request assignment strategy that couples the update instants of $\{R_{s\ell c}(t)\}$ variables with the request arrival times. Further, incoming requests are assigned to download chunks from a (state dependent) selected subset of BSs, thus always

increasing the current $R_{s\ell c}(t)$ values of the selected BSs. This differs from the gradient descent algorithm where, depending on the gradient, an update could also recommend decreasing the value of some rate variables. We formally discuss our request assignment strategy next.

Request Assignments : Suppose a request for content c arrives at location ℓ at time t. Then, the decision as to how many chunks should be downloaded from different BSs is based on the "prices" associated with the BSs connected to ℓ. Formally, for each (ℓ, c), we define the price of BS s (such that $A_{\ell s} = 1$) at time t as,

$$\Pi_{s\ell c}(t) \quad = \quad P_s(t) + P_{s\ell c}(t) \tag{7}$$

where, $P_{s\ell c}(t)$ is the CA price (recall (6)) and

$$P_s(t) \quad := \quad C'_s\Big(\sum_{\ell' c'} A_{\ell' s}\frac{R_{s\ell' c'}(t)}{k}\Big)$$

represents the *bandwidth price* of BS s.

Remark : Although the price depends on various caching and routing variables, its calculation in practice can be based on, for instance, an estimate of the current load at a BS to compute its bandwidth price; The content availability price can be simplified to reflect the cost of fetching the unavailable chunks from the backend data center. We will study the performance of such a light-weight scheme in Section 4.

Now, given the BS prices, let $\mathcal{S}_{\ell c}(t)$ denote the set of all minimum price BSs reachable from ℓ, i.e.,

$$\mathcal{S}_{\ell c}(t) = \underset{s}{\operatorname{argmin}}\Big\{\Pi_{s\ell c}(t) : A_{\ell s} = 1\Big\}. \tag{8}$$

Our request assignment strategy centers on downloading sufficient chunks of c from the BSs in $\mathcal{S}_{\ell c}(t)$ to construct the content at ℓ. However, if there are an insufficient number of chunks at these BSs, signals in the form of *dummy chunk* assignments are issued to increase their CA price and thus trigger future replication of content c chunks at these BSs. Formal details follow, where we have chosen to distribute the dummy chunks equally among all the BSs in $\mathcal{S}_{\ell c}(t)$ (see (9b) and (9c)) ; however, it is important to note that any other dummy chunk assignment would work equally well, as long as the total number of chunks allocated (actual+dummy) is k.

Let n denote the number of BSs in $\mathcal{S}_{\ell c}(t)$ (i.e., $n = |\mathcal{S}_{\ell c}(t)|$). Now, arrange the BSs in $\mathcal{S}_{\ell c}(t)$ in any arbitrary order, for instance, e.g., in increasing order of the BS IDs (assuming that each BS is given a unique integer ID). Suppose $(s_1, s_2, \cdots, s_n)$ is such an arrangement, then BS s_i is assigned to serve $\Gamma_{s_i\ell c}(t) = \Gamma^{(1)}_{s_i\ell c}(t) + \Gamma^{(2)}_{s_i\ell c}(t)$ chunks (actual+dummy, resp.) of content c, where (using $\lfloor x \rfloor$ to denote the largest integer smaller than x), for $i = 1, \cdots, n$,

$$\Gamma^{(1)}_{s_i\ell c}(t) \quad = \quad \min\Big\{k - \sum_{j=1}^{i-1}\Gamma^{(1)}_{s_j\ell c}(t), \Theta_{s_i c}(t)\Big\}, \tag{9a}$$

$$\Gamma^{(2)}_{s_i\ell c}(t) \quad = \quad \left\lfloor \frac{k - \sum_{j=1}^{n}\Gamma^{(1)}_{s_j\ell c}(t)}{n} \right\rfloor, i = 1, \cdots, n-1 \tag{9b}$$

$$\Gamma^{(2)}_{s_n\ell c}(t) \quad = \quad k - \sum_{j=1}^{n}\Gamma^{(1)}_{s_j\ell c}(t) - \sum_{j=1}^{n-1}\Gamma^{(2)}_{s_j\ell c}(t). \tag{9c}$$

No chunks are downloaded from the BSs not in $\mathcal{S}_{\ell c}(t)$, i.e., we have $\Gamma_{s\ell c}(t) = 0$ for $s \notin \mathcal{S}_{\ell c}(t)$.

Thus, $\Gamma^{(1)}_{s_i\ell c}(t)$ is the actual number of chunks that s_i can serve, while $\sum_{j=1}^{n}\Gamma^{(2)}_{s_j\ell c}(t)$ represents the total number of chunks that are not present in the caches, i.e., results in cache misses. Although, physically it is not possible for s_i to serve $\Gamma^{(2)}_{s_i\ell c}(t)$ number of additional chunks, we proceed to allocate these as dummy chunks served by s_i. As mentioned earlier, the idea is to increase the CA price $P_{s_i\ell c}(t)$ (recall (6)), so that more chunks of c will be replicated by s during its subsequent cache updates (see (4)). Practically, the dummy chunks are downloaded from the backend data center.

Finally, the associated rate variables are updated according to the following expression : $R_{s\ell c}(t^+) = R_{s\ell c}(t) + \Gamma_{s\ell c}(t)$.

3.2 Theoretical Guarantees

Under the above cache update and request assignment strategy, the system can be modeled as a CTMC (continuous Time Markov Chain) with system state $X_k(t)$. Let $X = (\{\Theta_{sc}\}, \{R_{s\ell c}\})$ denote a generic state, and let e_{sc} and $e_{s\ell c}$ be unit vectors with the same dimension as X consisting of all zeros except for one at positions Θ_{sc} and $R_{s\ell c}$, respectively. Then, the state transitions are given by

$$X \rightarrow X + [\epsilon\Delta\Theta_{sc}]e_{sc} \text{ with rate } k\nu_{sc} \tag{10a}$$

$$X \rightarrow X - \frac{1}{k}e_{s\ell c} \text{ with rate } kR_{s\ell c} \tag{10b}$$

$$X \rightarrow X + \sum_{s:A_{\ell s}=1}\frac{\Gamma_{s\ell c}}{k}e_{s\ell c} \text{ with rate } k\lambda_{\ell c}, \tag{10c}$$

where $\Gamma_{s\ell c}$ is computed using (9a) and (9b), but with $X_k(t)$ replaced by X. Transition (10a) corresponds to the event of a cache update at s for c. Transition (10b) occurs when a chunk completes its service (recall that the chunk service times are unit mean i.i.d exponentials). Finally, (10c) represents the transition that occurs whenever a new request for c arrives at location ℓ.

As the scaling parameter $k \rightarrow \infty$, it can be shown that the scaled state process, $X_k(t)/k$, converges almost surely to $x(t) = (\{\theta_{sc}(t)\}, \{r_{s\ell c}(t)\})$ on any finite interval $[0, T]$, where $x(t)$ is the solution of the following ODE (ordinary differential equation) [20, 21] :

$$\dot{\theta}_{sc}(t) = \epsilon\nu_{sc}\Delta\theta_{sc}(t) \tag{11a}$$

$$\dot{r}_{s\ell c}(t) = -r_{s\ell c}(t) + \lambda_{\ell c}\gamma_{s\ell c}(t). \tag{11b}$$

In the above expression, $\Delta\theta_{sc}(t)$ is given by

$$\Delta\theta_{sc}(t) = -q_s(t) + \sum_{\ell} A_{\ell s}\lambda_{\ell c}p_{s\ell c}(t), \tag{12}$$

where $q_s(t) = \widehat{C}'_s(\sum_c \theta_{sc}(t))$, $p_{s\ell c}(t) = C'_{s\ell c}(r_{s\ell c}(t) - \lambda_{\ell c}\theta_{sc}(t))$, and $\gamma_{s\ell c}(t)$ satisfies the following : First define the price of BS s (such that $A_{\ell s} = 1$) as

$$\pi_{s\ell c}(t) = p_s(t) + p_{s\ell c}(t), \tag{13}$$

where $p_s(t) = C'_s(\sum_{\ell' c'} A_{\ell' s'} r_{s\ell' c'}(t))$; again, as in (8), define $\mathcal{S}_{\ell c}(t)$ as

$$\mathcal{S}_{\ell c}(t) = \underset{s}{\text{argmin}}\Big\{\pi_{s\ell c}(t) : A_{\ell s} = 1\Big\}, \tag{14}$$

and arrange the $n = |\mathcal{S}_{\ell c}(t)|$ BSs in $\mathcal{S}_{\ell c}(t)$ as $(s_1, s_2, \cdots, s_n)$ (following the same criterion as used in the finite system). Then, $\gamma_{s_i \ell c}(t) = \gamma^{(1)}_{s_i \ell c}(t) + \gamma^{(2)}_{s_i \ell c}(t)$ is given by, for $i = 1, \cdots, n$,

$$\gamma^{(1)}_{s_i \ell c}(t) = \min\left\{1 - \sum_{j=1}^{i-1} \gamma^{(1)}_{s_j \ell c}(t), \theta_{s_i c}(t)\right\} \tag{15a}$$

$$\gamma^{(2)}_{s_i \ell c}(t) = \frac{1 - \sum_{j=1}^{n} \gamma^{(1)}_{s_j \ell c}(t)}{n}. \tag{15b}$$

For any $s \notin \mathcal{S}_{\ell c}(t)$, we have $\gamma_{s \ell c}(t) = 0$.

The following is the main result of this section.

Theorem 1. *The dynamics $x(t) = (\{\theta_{sc}(t), r_{s\ell c}(t)\})$ described by the ODE in (11) converges to the optimal point of the total cost minimization problem in (3). Thus, for large system size k, our cache update and request assignment algorithm, described by the process $(\{\Theta_{sc}(t), R_{s\ell c}(t)\})$, operates close to the optimal point.*

Outline. We first show that the stationary point, $x^* = (\{\theta^*_{sc}\}, \{r^*_{s\ell c}\})$, of the dynamics in (11) is optimal for the TC problem in (3) : since all the cost functions are convex, the problem in (3) is a convex optimization problem; thus, to prove optimality, it suffices to identify dual variables that, along with $(\{\theta^*_{sc}\}, \{r^*_{s\ell c}\})$, satisfy the KKT conditions. We prove convergence (stability) using Lyapunov functions method. This is challenging because the usual procedure of using a quadratic Lyapunov function (e.g., in [22, Chapter 3]) does not work in our case. This is because, the rate dynamics, $\dot{r}_{s\ell c}$, in (11b) are driven by the chunk assignment variables, $\gamma_{s\ell c}$, instead of the cost gradients $(\partial C/\partial r_{s\ell c})$. Hence, using the Lagrangian function (of the problem in (3)) along with a quadratic term, we carefully construct a Lyapunov function that enables us to establish the stability of x^*. Thus, our results can be of independent interest towards Lyapunov stability analysis. Formal details of the proof are available in Appendix A. □

4 Simulation Experiments

We first consider a simple setting comprising 2 locations and 3 BSs; the content catalog size is $|\mathcal{C}| = 10$. After demonstrating the efficacy of the joint-price based request assignment schemes for this setting, we present results for a larger system.

We assume that BSs 1 and 3 are *femtocells* serving locations 1 and 2, respectively, while BS 2 is a common *cellular BS* connected to both the locations. Thus, the requests arriving at a location can be assigned either to the respective femtocell or to the cellular BS. The total (unscaled) content request rate from each location is 1, i.e., $\sum_c \lambda_{\ell c} = 1$ for $\ell = 1, 2$. The content popularity at each location follows a Zipf distribution with parameter 0.8, i.e., the probability that an incoming request is for the i-th most popular content is proportional to $1/i^{0.8}$. In order to obtain a scenario where the same content can have different popularities at different locations, we randomly shuffle the distributions for the two locations. The (unscaled) cache update rate of each BS for each content is 1, i.e., $\nu_{sc} = 1$.

The various costs (recall (2)) used in the simulations are as follows :

• Bandwidth cost : $C_s(r) = \exp(2(r - R_s))$, where $R_1 = R_3 = 0.5$ and $R_2 = 0.1$; thus, the cellular BS is costlier than the femtocell BSs.

• Memory cost : $\widehat{C}_s(\theta) = \exp(5(\theta - M_s))$ where $M_1 = M_3 = 2$ while $M_2 = 8$; thus, the cellular BS has more memory compared with the femtocells.

• Content availability (CA) cost : $C_{s\ell c}(x) = \exp(bx)$; we are particularly interested in studying the effect of b on performance. A large value of b, which yields a steep cost function, results

in a system where cache misses are costlier than bandwidth. Thus, b can be used to trade off bandwidth cost and cache-miss probability. We refer to b as the *cache-miss exponent*.

Performance of the online algorithm : In Figure 2 we first plot the optimal performance curve, obtained by directly solving the problem in (3) for different values of b (curve labeled 'Optimal'). As highlighted in the figure, the penalty exponent b increases from left to right along all the curves. Also shown in the figure is the performance of the online algorithm, described in Section 3, for a system with $k = 10$ chunks per content. Although the online algorithm is proven to converge to the optimal performance as $k \to \infty$, it is interesting to observe that its performance is comparable with the optimal already for $k = 10$.

Before proceeding further, let us briefly discuss the trade-off curves. Note that, as b increases the cache-miss probability improves, however at the expense of an increased bandwidth cost. This is because, for a larger b, the cellular BS offers a lower price for the incoming requests as it holds more contents. Thus, most of the requests are forwarded to the cellular BS, increasing its (and the overall) bandwidth cost. In contrast, when b is small, bandwidth cost is the key component in deciding the BSs' prices. Since femtocells are less expensive in this regard, more requests are now forwarded to the femtocells, resulting in more cache misses as they only hold few contents.

Drawbacks of the online algorithm : The implementation of the online algorithm however requires the BSs to maintain routing variables in the form of number of chunks of each content downloaded by requests from each location. This would require identifying each chunk with its request's location, which is not possible in practice when the number of ongoing requests are large. Further, estimates of the request arrival rates, $\lambda_{\ell c}$, are required by the BSs to compute their content availability prices. Finally, the cache update in the online algorithm is *server-driven* requiring the BSs to update their caches regularly. This would result in unnecessary updates whenever the request rate is less.

A light-weight signaling scheme : In view of the above drawbacks, we modify the online algorithm to obtain a light-weight signaling scheme for request assignments. We refer to it as the *joint-price* scheme since, as in the online algorithm, the request assignment here is based jointly on the content and bandwidth availability. The bandwidth price, as before, remains to be the same convex increase function of the current load on the BS ; thus, by using the congestion signals from the BS, it is possible to estimate the bandwidth cost at a location. However, for the content availability price, we now simply use the cost of fetching unavailable chunks if an insufficient number of chunks of the requested content is available at the BS. Formally, the content availability price of BS s for request c (from any location) at time t is given by $b(k - \Theta_{sc}(t))/k$, where $\Theta_{sc}(t)$ is the number of chunks of content c at BS s at time t and b now actually represents the cost of fetching the unavailable fraction of content c. Thus, as with the online algorithm, a larger b implies that the cache misses are costly.

We use the above joint price signaling scheme in conjunction with the LRU (Least Recently Used) cache eviction algorithm [9], which is a *request driven* cache management strategy. Under LRU, cache replacements are initiated by incoming requests that do not find the desired content in the cache (a cache-miss event). As a consequence, the least recently used content is removed from the cache to free memory for storing the new content. Since we deal with contents at the chunk level, we have implemented a variation of LRU, referred to as the *CLRU (Chunk LRU)* strategy. In CLRU, the incoming request at a cache for m chunks of a content triggers the relocation of such m chunks to the head of the cache queue to be served. The placement of extra cached chunks at the cache (i.e., if more than m chunks are cached) remain unchanged. In the opposite case, if chunks are missing (i.e., if less than the requested m chunks are cached) and the cache is full, new chunks of the content are downloaded and placed at the head of the queue by removing least recently used chunks at the tail of the queue.

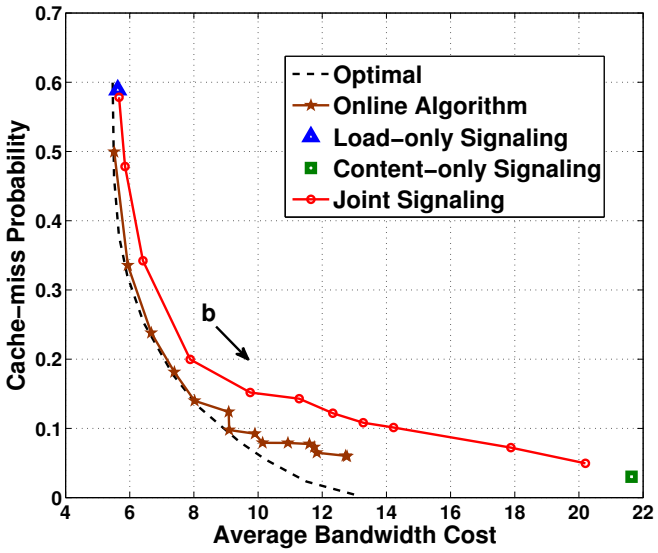


FIGURE 2 – Performance curves.

Performance of joint-price scheme : In Figure 2 we have depicted the performance of the joint-price signaling scheme. Also shown in the figure are the performance of two heuristic policies (commonly used in practice) where the request assignment is purely based on bandwidth availability or content availability signals (points 'Load-only Signaling' and 'Content-only Signaling', respectively). The joint-price scheme provides a favorable trade off between bandwidth cost and cache-miss probability, encompassing the heuristic policies as its extreme end points.

Although the joint-price scheme suffers some degradation in performance when compared with the online algorithm, it is useful to emphasize that the simplicity of the former scheme would render it more favorite than the latter when it comes to considerations for practical implementation. Moreover, the performance degradation is less for low to moderate values of cache-miss cost b, which is typically the case in practice.

Large network : Motivated by the observations made in the previous section, we proceed to study the performance of the joint-price scheme for large networks. We now consider a content catalog of size 1000. The number of locations are increased to 10. It will be useful to think of a location $\ell \in \{1, 2, \cdots, 10\}$ as the segment $[0, \ell]$ on the real line (recall Figure 1 for illustration). A cellular base station is located at the center (i.e., at 5) so that, as in the earlier case, it can cover all the locations. However, we now increase the number of (*small-size*) femtocells to 9, which are located at positions $1, 2, \cdots, 9$; each small-size femtocell has a range of 1. Thus, each location (except the end locations) is served by two adjacent femtocells. To model the heterogeneity in the femtocell base stations, we introduce 5 *moderate-size femtocells*. The moderate femtocells have a larger range of 2 and are located at $3, 4, \cdots, 7$. Note that, the inner locations are covered by more femtocells (small+moderate) than the outer ones.

Small-size femtocells have smaller caches (of size 10) compared to moderate-size femtocells (whose cache capacity is 100), and the latter's capacity is smaller than the cellular BS which can store up to $M_s = 800$ contents. Bandwidth cost increases as we move from the small to moderate femtocell and to cellular BS ; recalling the cost functions from the previous section, the respective values of R_s are 5, 2.5 and 1.

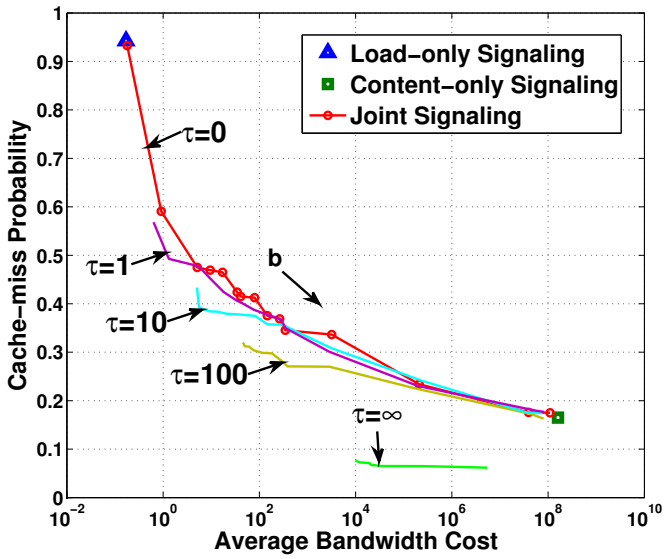


FIGURE 3 – Performance of the light-weight schemes for a large network.

In Figure 3 we first plot the performance achieved by the joint-price signaling scheme along with the performance of the two heuristic schemes. Again, we observe that the joint-price scheme can achieve a large range of performance values, which is in contrast to the heuristic schemes whose performance is fixed at either optimizing bandwidth cost or cache-miss probability, without regarding for the other metric.

In the thus far implemented joint-price schemes, we have been assigning requests to download chunks *strictly* from the set of min-priced servers. We now relax this strict consideration by regarding any BS, whose price is within a *tolerance value* of τ from the minimum price, as *eligible* to serve chunks for an incoming request (provided as before the eligible BSs are connected to the request's location) ; chunks are however downloaded by sorting the eligible BSs in the increasing order of their prices. In Figure 3 we have depicted the performance of the joint-price scheme for different values of τ, where the earlier strict min-price approach now corresponds to $\tau = 0$; the case where all connected BSs (irrespective of their prices) are eligible corresponds to $\tau = \infty$.

We observe that as τ increases the trade-off curves shift downwards, implying that it is possible to achieve a lower cache-miss probability for a given target bandwidth cost. However, the range of trade-off that is possible reduces as τ increases. For instance, using $\tau = 100$ it is not possible to reduce the bandwidth cost to less than 40 ; In fact, $\tau = \infty$ case achieves the lowest cache-miss probability, however at the expense of being constrained to operate at a higher range of bandwidth cost. This shift in performance is because, as τ increases, more BSs become eligible to serve a request so that the cache-miss probability decreases ; however, more BSs are now loaded, resulting in a bandwidth scarce system. Thus, further trade-off in performance can be achieved by suitably choosing the value of τ.

In summary, the joint-price scheme, in conjunction with the CLRU policy for cache management, can be a practical candidate for request assignments in futuristic cellular networks where a dense deployment of heterogeneous femtocell base-stations are expected. The cache-miss cost b, along with the tolerance value τ, can be used to serve as "tunable knobs" to trade-off one metric for the other.

5 Conclusion

We proposed an optimization framework to study the problem of cost minimization in cellular cache networks. Towards this direction, we proposed an online algorithm for caching and request assignments which is shown to be optimal and stable in a limiting regime that is obtained by scaling the arrival rates and the content chunking. Based on the online algorithm, we proposed a light-weight joint-price scheme for request assignment that can be used in conjunction with the LRU cache management strategy. Through simulations we found that our joint-price based request assignment strategy outperforms the common practices of routing purely based on either load or content availability. Our proposed joint-price routing mechanisms are thus an appealing candidate combining sound theoretical guarantees with good experimental performance while being simple to implement.

Références

[1] "Cisco Visual Networking Index : Global Mobile Data Traffic Forecast Update, 2013-2018," Feb. 2014.

[2] K. Poularakis and L. Tassiulas, "Exploiting User Mobility for Wireless Content Delivery," in *IEEE International Symposium on Information Theory Proceedings, ISIT '13*.

[3] N. Golrezaei, K. Shanmugam, A. Dimakis, A. Molisch, and G. Caire, "FemtoCaching : Wireless Video Content Delivery Through Distributed Caching Helpers," in *INFOCOM '12*.

[4] M. Dehghan, A. Seetharam, B. Jiang, T. He, T. Salonidis, J. Kurose, D. Towsley, and R. Sitaraman, "On the Complexity of Optimal Routing and Content Caching in Heterogeneous Networks," in *to appear at INFOCOM 2015*.

[5] S. Borst, V. Gupta, and A. Walid, "Distributed Caching Algorithms for Content Distribution Networks," in *INFOCOM '10*.

[6] J. Dai, Z. Hu, B. Li, J. Liu, and B. Li, "Collaborative Hierarchical Caching with Dynamic Request Routing for Massive Content Distribution," in *INFOCOM '12*.

[7] I. Baev, R. Rajaraman, and C. Swamy, "Approximation Algorithms for Data Placement Problems," *SIAM J. Comput.*, vol. 38, no. 4, pp. 1411–1429, August 2008. [Online]. Available : http://dx.doi.org/10.1137/080715421

[8] E. Jaho, M. Karaliopoulos, and I. Stavrakakis, "Social Similarity Favors Cooperation : The Distributed Content Replication Case," *IEEE Transactions on Parallel and Distributed Systems*, March 2013.

[9] G. Bianchi, A. Detti, A. Caponi, and N. Blefari Melazzi, "Check Before Storing : What is the Performance Price of Content Integrity Verification in LRU Caching?" *SIGCOMM CCR*, July 2013.

[10] C. Fricker, P. Robert, and J. Roberts, "A Versatile and Accurate Approximation for LRU Cache Performance," in *Proceedings of the 24th International Teletraffic Congress, ITC '12*.

[11] A. Dan and D. Towsley, "An Approximate Analysis of the LRU and FIFO Buffer Replacement Schemes," in *ACM SIGMETRICS '90*.

[12] M. Jonckheere and J. Virtamo, "Optimal Insensitive Routing and Bandwidth Sharing in Simple Data Networks," in *ACM SIGMETRICS '05*.

[13] T. Bonald, M. Jonckheere, and A. Proutiére, "Insensitive Load Balancing," in *ACM SIGMETRICS /PERFORMANCE '04*.

[14] M. Amble, P. Parag, S. Shakkottai, and L. Ying, "Content-Aware Caching and Traffic Management in Content Distribution Networks," in *INFOCOM '11.*

[15] N. Abedini and S. Shakkottai, "Content Caching and Scheduling in Wireless Networks With Elastic and Inelastic Traffic," *IEEE/ACM Transactions on Networking*, 2013.

[16] W. Jiang, S. Ioannidis, L. Massoulié, and F. Picconi, "Orchestrating Massively Distributed CDNs," in *ACM CoNEXT '12.*

[17] B. Tan and L. Massoulié, "Optimal Content Placement for Peer-to-Peer Video-on-Demand Systems," *IEEE/ACM Transactions on Networking*, April 2013.

[18] M. Leconte, M. Lelarge, and L. Massoulié, "Bipartite Graph Structures for Efficient Balancing of Heterogeneous Loads," in *ACM SIGMETRICS/ PERFORMANCE '12.*

[19] D. Tse and P. Viswanath, *Fundamentals of wireless communication.* New York, NY, USA : Cambridge University Press, 2005.

[20] T. G. Kurtz, "Solutions of Ordinary Differential Equations as Limits of Pure Jump Markov Processes," *Journal of Applied Probability*, vol. 7, no. 1, pp. pp. 49–58, 1970.

[21] F. P. Kelly and R. J. Williams, "Fluid Model for a Network Operating under a Fair Bandwidth-Sharing Policy," *The Annals of Applied Probability*, vol. 14, no. 3, 2004.

[22] R. Srikant, *The Mathematics of Internet Congestion Control (Systems and Control : Foundations and Applications).* SpringerVerlag, 2004.

[23] H. Khalil, *Nonlinear Systems.* Prentice Hall, 2002.

A Proof of Theorem 1

Let $x^* = (\{\theta^*_{sc}\}, \{r^*_{s\ell c}\})$ be a stationary point of the ODE in (11). Then, $(\{\theta^*_{sc}\}, \{r^*_{s\ell c}\})$ satisfies

$$q^*_s = \sum_\ell A_{\ell s}\lambda_{\ell c}p^*_{s\ell c} \tag{16}$$

$$r^*_{s\ell c} = \lambda_{\ell c}\gamma^*_{s\ell c}, \tag{17}$$

where $q^*_s = \widehat{C}'_s(\sum_c \theta^*_{sc})$, $p^*_{s\ell c} = C'_{s\ell c}(r^*_{s\ell c} - \lambda_{\ell c}\theta^*_{sc})$, and $\gamma^*_{s\ell c}$ is obtained as in Section 3.2 by first defining $\pi^*_{s\ell c}$ and $\mathcal{S}^*_{\ell c}$ (recall (13) and (14), respectively), but with $(\{\theta_{sc}(t)\}, \{r_{s\ell c}(t)\})$ replaced by $(\{\theta^*_{sc}\}, \{r^*_{s\ell c}\})$; then, arranging the $n = |\mathcal{S}^*_{\ell c}|$ BSs in $\mathcal{S}^*_{\ell c}$ as $(s_1, s_2, \cdots, s_n)$, we compute $\gamma^*_{s_i\ell c}$ using (15). Note that, since, $\sum_s A_{\ell s}\gamma^*_{s\ell c} = 1$, from (17) we have

$$\sum_s A_{\ell s}r^*_{s\ell c} = \lambda_{\ell c}. \tag{18}$$

Proof of Optimality. Recalling the total cost in (2), the Lagrangian of the problem in (3) can be written as

$$L = L(\{\theta_{sc}\}, \{r_{s\ell c}\}, \{\beta_{s\ell c}\}, \{\delta_{\ell c}\}) \tag{19}$$

$$= C(\{\theta_{sc}\}, \{r_{s\ell c}\}) - \sum_{s\ell c} A_{\ell s}\beta_{s\ell c}r_{s\ell c} + \sum_{\ell c}\delta_{\ell c}\Big(\lambda_{\ell c} - \sum_s A_{\ell s}r_{s\ell c}\Big), \tag{20}$$

where $\{\beta_{s\ell c}\}$ and $\{\delta_{\ell c}\}$ are the dual variables associated with the constraints (3c) and (3d), respectively. For convenience, define $p_s := C'_s(\sum_{\ell c} A_{\ell s}r_{s\ell c})$, $q_s := \widehat{C}'_s(\sum_c \theta_{sc})$, and $p_{s\ell c} :=$

Symbol	Description
M_s	Cache capacity of BS s
p_s	Bandwidth price; $p_s = C'_s(\sum_{\ell c} A_{\ell s} r_{s\ell c})$
$p_{s\ell c}$	CA price; $p_{s\ell c} = C'_{s\ell c}(r_{s\ell c} - \lambda_{\ell c}\theta_{sc})$
q_s	Memory price; $q_s = \widehat{C}'_s(\sum_c \theta_{sc})$
$r_{s\ell c}$	Rate at which BS s serves requests for content c from location ℓ
g_s	Total service rate of s : $g_s = \sum_{\ell c} A_{\ell s} r_{s\ell c}$
$\epsilon\Delta\theta_{sc}$	Fraction of chunks updated : $\epsilon > 0$ and $\Delta\theta_{sc} = -q_s + \sum_\ell A_{\ell s}\lambda_{\ell c} p_{s\ell c}$
$\gamma_{s\ell c}$	Fraction of chunks of content c for c
$\lambda_{\ell c}$	Request arrival rate for c at ℓ
ν_{sc}	Rate at which s updates its cache for c
$\phi_{\ell c}$	Min price : $\phi_{\ell c} = \min\{\pi_{s\ell c} : A_{\ell s} = 1\}$
$\pi_{s\ell c}$	BS price : $\pi_{s\ell c} = p_s + p_{s\ell c}$
θ_{sc}	Fraction of content c stored at BS s

TABLE 1 – Glossary of symbols.

$C'_{s\ell c}(r_{s\ell c} - \lambda_{\ell c}\theta_{sc})$. Then, the KKT conditions are given by, for all $s \in \mathcal{S}, \ell \in \mathcal{L}, c \in \mathcal{C}$ (such that $A_{\ell s} = 1$),

$$q_s - \sum_\ell A_{\ell s}\lambda_{\ell c} p_{s\ell c} = 0 \tag{21}$$

$$p_s + p_{s\ell c} - \beta_{s\ell c} - \delta_{\ell c} = 0 \tag{22}$$

along with the primal feasibility ((3b) to (3d)), dual feasibility ($\beta_{s\ell c} \geq 0$), and the complementary slackness conditions ($\beta_{s\ell c} r_{s\ell c} = 0$).

Recalling (16), we see that $(\{\theta^*_{sc}\}, \{r^*_{s\ell c}\})$ immediately satisfies (21). Next, define $\beta^*_{s\ell c}$ and $\delta^*_{\ell c}$ as follows : $\delta^*_{\ell c} := \min_s\{\pi^*_{s\ell c} : A_{\ell s} = 1\}$ and $\beta^*_{s\ell c} := \pi^*_{s\ell c} - \delta^*_{\ell c}$. Thus, we have

$$\beta^*_{s\ell c} + \delta^*_{\ell c} = \pi^*_{s\ell c} = p^*_s + p^*_{s\ell c}. \tag{23}$$

It is easy to check that (22) is satisfied. Further, for $s \in \mathcal{S}^*_{\ell c}$ we have $\beta^*_{s\ell c} = 0$, so that the slackness condition $\beta^*_{s\ell c} r^*_{s\ell c} = 0$ is met. Similarly, the slackness condition is also satisfied for $s \notin \mathcal{S}^*_{\ell c}$ since then $\gamma^*_{s\ell c} = 0 \implies r^*_{s\ell c} = 0$ (recall (17)).

Thus, we have exhibited dual variables, $(\{\beta^*_{s\ell c}\}, \{\delta^*_{\ell c}\})$, that, along with $(\{\theta^*_{sc}\}, \{r^*_{s\ell c}\})$, satisfies the KKT conditions. Since the problem in (3) is convex, it follows that $(\{\theta^*_{sc}\}, \{r^*_{s\ell c}\})$ is optimal. □

Proof of Convergence. Consider the following Lyapunov function :

$$V(x(t)) = L^*(x(t)) + S(x(t)) \tag{24}$$

where, $L^*(x(t)) := L(x(t), \{\beta^*_{s\ell c}\}, \{\delta^*_{\ell c}\})$ is the Lagrangian function in (19) evaluated at the dual optimal, and $S(x(t))$ is a quadratic term, $S(x(t)) := \sum_{sc} \frac{1}{2\epsilon\nu_{sc}}(\theta_{sc}(t) - \theta^*_{sc})^2$. To prove that the stationary point $x^* = (\{\theta^*_{sc}\}, \{r^*_{s\ell c}\})$ is globally asymptotic stable, it suffices to ensure that our candidate Lypunov function V satisfies the conditions in [23, Theorem 4.2] ($\dot{F}$ denotes the time derivative of a function F) : (i) $V(x^*) < V(x(t))$, for all $x(t) \neq x^*$, (ii) $\| x(t) \| \to \infty \implies V(x(t)) \to \infty$, and (iii) $\dot{V}(x(t)) < 0$, for all $x(t) \neq x^*$.

Since all the cost functions (C_s, $\widehat{C}_s$, and $C_{s\ell c}$) are strictly convex and increasing, it follows that x^* is the unique minimizer of L^*. Thus, L^* along with the quadratic term S easily ensures (i) and (ii). The challenging part is to show (iii), which is the remainder of the proof.

For simplicity, from here on we will drop the time variable t from our notation. Thus, taking the time derivative of L^* and rearranging, we obtain

$$\begin{aligned}\dot{L^*}(x) &= \sum_{s\ell c} A_{\ell s}\dot{r}_{s\ell c}(p_s + p_{s\ell c} - \beta^*_{s\ell c} - \delta^*_{s\ell c}) \\ &\quad + \sum_{sc}\dot{\theta}_{sc}(q_s - \sum_{\ell} A_{\ell s}\lambda_{\ell c}p_{s\ell c}). \end{aligned} \tag{25}$$

From (11a) we see that the second term is non-positive; further, using (11b) and (23) the above expression can be simplified to obtain

$$\dot{L^*}(x) \leq \sum_{s\ell c} A_{\ell s}(-r_{s\ell c} + \lambda_{\ell c}\gamma_{s\ell c})(p_s - p^*_s + p_{s\ell c} - p^*_{s\ell c}). \tag{26}$$

For simplicity, define $g_s := \sum_{\ell c} A_{\ell s}r_{s\ell c}$ (similarly define g^*_s). Then, since $p_s = C'_s(g_s)$ is increasing in g_s, we can write $(p_s - p^*_s)(g_s - g^*_s) \geq 0$, or equivalently

$$(p_s - p^*_s)g_s \quad \geq \quad (p_s - p^*_s)g^*_s, \tag{27}$$

with equality if and only if (iff) $g_s = g^*_s$. Similarly, $p_{s\ell c}$ is increasing in its argument $(r_{s\ell c} - \lambda_{\ell c}\theta_{sc})$, so that we have

$$(p_{s\ell c} - p^*_{s\ell c})(r_{s\ell c} - \lambda_{\ell c}\theta_{sc}) \geq (p_{s\ell c} - p^*_{s\ell c})(r^*_{s\ell c} - \lambda_{\ell c}\theta^*_{sc}). \tag{28}$$

Using (27) and (28) in (26), and simplifying yields

$$\dot{L^*}(x) \leq \sum_{s\ell c} A_{\ell s}(-r^*_{s\ell c} + \lambda_{\ell c}\gamma_{s\ell c})(\pi_{s\ell c} - \pi^*_{s\ell c}) + T \tag{29}$$

with equality iff $x = x^*$, where T is given by

$$\begin{aligned} T &= \sum_{s\ell c} A_{\ell s}(p_{s\ell c} - p^*_{s\ell c})(\lambda_{\ell c}\theta^*_{sc} - \lambda_{\ell c}\theta_{sc}) \\ &= \sum_{sc}(\theta^*_{sc} - \theta_{sc})\Big(\sum_{\ell} A_{\ell s}\lambda_{\ell c}(p_{s\ell c} - p^*_{s\ell c})\Big) \\ &\overset{o}{=} \sum_{sc}(\theta^*_{sc} - \theta_{sc})(q_s + \Delta\theta_{sc} - q^*_s) \\ &= \sum_{s}(q_s - q^*_s)\Big(\sum_{c}(\theta^*_{sc} - \theta_{sc})\Big) + \sum_{sc}(\theta^*_{sc} - \theta_{sc})\Delta\theta_{sc} \\ &\overset{*}{\leq} \sum_{sc}(\theta^*_{sc} - \theta_{sc})\Delta\theta_{sc}. \end{aligned} \tag{30}$$

In the above expression, o is obtained by using (12) and (16), and $*$ is because $q_s = \widehat{C}_s(\sum_c \theta_{sc})$ is increasing in its argument so that we have an inequality similar to (27). Now, combining (29) and (30) along with $\dot{S}(x) = \sum_{sc}(\theta_{sc} - \theta^*_{sc})\Delta\theta_{sc}$, we obtain,

$$\dot{V}(x) \leq \sum_{s\ell c} A_{\ell s}(-r^*_{s\ell c} + \lambda_{\ell c}\gamma_{s\ell c})(\pi_{s\ell c} - \pi^*_{s\ell c}), \tag{31}$$

with equality iff $x = x^*$.

Define, $\phi_{\ell c} := \min_s\{\pi_{s\ell c} : A_{\ell s} = 1\}$ (similarly define $\phi^*_{\ell c}$). Now, since our chunk assignment policy is such that $\gamma_{s\ell c} > 0$ implies $\pi_{s\ell c} = \phi_{\ell c}$, it follows that,

$$\pi_{s\ell c}(\lambda_{\ell c}\gamma_{s\ell c} - r^*_{s\ell c}) \leq \phi_{\ell c}(\lambda_{\ell c}\gamma_{s\ell c} - r^*_{s\ell c}), \forall s.$$

Similarly, since $r^*_{s\ell c} > 0$ implies $\pi^*_{s\ell c} = \phi^*_{\ell c}$, we have,

$$\pi^*_{s\ell c}(r^*_{s\ell c} - \lambda_{\ell c}\gamma_{s\ell c}) \leq \phi^*_{\ell c}(r^*_{s\ell c} - \lambda_{\ell c}\gamma_{s\ell c}), \forall s.$$

Thus, (31) becomes,

$$\dot{V}(x) \quad \leq \quad \sum_{\ell c}(\phi_{\ell c} - \phi^*_{\ell c}) \sum_s A_{\ell s}(\lambda_{\ell c}\gamma_{s\ell c} - r^*_{s\ell c}), \tag{32}$$

with equality iff $x = x^*$. Finally, the proof is complete by noting that $\sum_s A_{\ell s}\lambda_{\ell c}\gamma_{s\ell c} = \sum_s A_{\ell s} r^*_{s\ell c} = \lambda_{\ell c}$ (recall (15) and (18)). □

B Formulation with Hard Constraints

The total cost minimization problem in (3) includes the memory and the content availability constraints using respective penalty functions. In this section we propose a bandwidth cost minimization problem where these constraints are directly included as follows :
Bandwidth Cost (BC) :

$$\begin{aligned}
\text{Minimize :} \quad & \sum_s C_s\Big(\sum_{\ell c} A_{\ell s} r_{s\ell c}\Big) && \text{(33a)}\\
\text{Over :} \quad & \theta_{sc} \in [0,1], s \in \mathcal{S}, c \in \mathcal{C} && \text{(33b)}\\
& r_{s\ell c} \geq 0, s \in \mathcal{S}, \ell \in \mathcal{L}, c \in \mathcal{C} && \text{(33c)}\\
\text{Subject to :} \quad & \sum_c \theta_{sc} \leq M_s, s \in \mathcal{S} && \text{(33d)}\\
& r_{s\ell c} \leq A_{\ell s}\lambda_{\ell c}\theta_{sc}, s \in \mathcal{S}, \ell \in \mathcal{L}, c \in \mathcal{C} && \text{(33e)}\\
& \sum_s A_{\ell s} r_{s\ell c} = \lambda_{\ell c}, \ell \in \mathcal{L}, c \in \mathcal{C}. && \text{(33f)}
\end{aligned}$$

Expressions (33d) and (33e) represent the memory and the content availability constraints, respectively. As in (3), the equality constraint (33f) is used to ensure that all incoming requests are handled by the BSs. Throughout this work we assume that the above problem is feasible.

Using a primal-dual update algorithm, we design an analogue of the cache replacement and request assignment strategy developed in Section 3. We again prove the optimality and stability of the proposed algorithm.

B.1 Primal-Dual Updates

For simplicity we reuse the same notation introduced in the previous section. Thus, $\Theta_{sc}(t)$ denotes the number of chunks of content c stored at s, and $R_{s\ell c}(t)$ is the rate at which s is serving requests for c from ℓ. We refer to $(\{\Theta_{sc}(t)\}, \{R_{s\ell c}(t)\})$ as the *primal variables*. Additionally, we introduce *dual variables* (to be defined shortly) $Q_s(t)$ and $P_{s\ell c}(t)$ corresponding to the memory and bottleneck constraints, respectively. These dual variables are analogous to the memory and content availability prices ((5) and (6), respectively) used in the previous section. The total system state is now comprised of both primal and dual variables :

$X_k(t) = (\{\Theta_{sc}(t)\}, \{R_{s\ell c}(t)\}, \{Q_s(t)\}, \{P_{s\ell c}(t)\})$. Note that, as before we are assuming that each content can be constructed by downloading k chunks from different BSs. Again, the chunk service times are i.i.d exponential random variables of mean one.

Dual Updates : The dual variable $Q_s(t)$ associated with the memory constraint is updated at rate $k\zeta_s$. Suppose BS s chooses to update $Q_s(t)$ at time t, then the amount of change is given by

$$\Delta Q_s(t) \quad = \quad \Big(\frac{\sum_c \Theta_{sc}(t)}{k} - M_s\Big)^+_{Q_s(t)} \tag{34}$$

where (borrowing notation from [22])

$$(y)^+_x = \begin{cases} y, \text{ if } x > 0 \\ \max\{y, 0\}, \text{ if } x = 0. \end{cases}$$

We thus have the following update : $Q_s(t^+) = Q_s(t) + \Delta Q_s(t)$.

We see that $Q_s(t)$ associated with BS s increases or decreases, respectively, depending on whether the memory constraint M_s is exceeded or not; however, if $Q_s(t) = 0$ then there is no further change until the constraint is exceeded again. Later we will see that such an update reduces replication whenever the constraint is exceeded and increases replication otherwise, while ensuring that complementary slackness is respected.

The other dual variable, $P_{s\ell c}(t)$, is similarly updated at a rate $k\xi_{s\ell c}$, with the amount of change at time t given by

$$\Delta P_{s\ell c}(t) = \Big(\lambda_{\ell c}\frac{\Gamma_{s\ell c}(t)}{k} - \lambda_{\ell c}\frac{\Theta_{sc}(t)}{k}\Big)^+_{P_{s\ell c}(t)} \tag{35}$$

where $\Gamma_{s\ell c}(t)$ is computed exactly as in Section 3.1 : First we define the set of minimum price BSs as $\mathcal{S}_{\ell c}(t) = \arg\min_s\{P_s(t) + P_{s\ell c}(t) : A_{\ell s} = 1\}$. Then, $\Gamma_{s\ell c}(t)$ is obtained using (9), but with the memory and CA prices in (5) and (6) now replaced by the corresponding dual variables, $Q_s(t)$ and $P_{s\ell c}(t)$, respectively.

Primal Updates : The update of the primal variables is similar to the update of $\Theta_{sc}(t)$ and $R_{s\ell c}(t)$ in Section 3.1, again with the exception that the memory and CA prices in (5) and (6) are replaced by the corresponding dual variables, $Q_s(t)$ and $P_{s\ell c}(t)$, respectively.

B.2 Theoretical Guarantees

It is possible to show that the rescaled state process $X_k(t)/k$ converges almost surely as $k \to \infty$, on finite intervals of time, to $x(t) = (\{\theta_{sc}(t)\}, \{r_{s\ell c}(t)\}, \{q_s(t)\}, \{p_{s\ell c}(t)\})$ satisfying the following : for all s, ℓ (such that $A_{\ell s} = 1$) c,

$$\dot{q}_s(t) \quad = \quad \zeta_s\Big(\sum_c \theta_{sc}(t) - M_s\Big)^+_{q_s(t)} \tag{36a}$$

$$\dot{p}_{s\ell c}(t) \quad = \quad \xi_{s\ell c}\Big(\lambda_{\ell c}\gamma_{s\ell c}(t) - \lambda_{\ell c}\theta_{sc}(t)\Big)^+_{p_{s\ell c}(t)} \tag{36b}$$

$$\dot{\theta}_{sc}(t) \quad = \quad \epsilon\nu_{sc}\Big(-q_s(t) + \sum_\ell A_{\ell s}\lambda_{\ell c}p_{s\ell c}(t)\Big) \tag{36c}$$

$$\dot{r}_{s\ell c}(t) \quad = \quad -r_{s\ell c}(t) + \lambda_{\ell c}\gamma_{s\ell c}(t), \tag{36d}$$

where $\gamma_{s\ell c}(t)$ are defined exactly as in Section 3.2, except that the memory and the bottleneck prices are now replaced by the corresponding dual variables, $q_s(t)$ and $p_{s\ell c}(t)$.

Theorem 2. *The dynamics $x(t)$ described by the ODE in (36) converges to the optimal point of the bandwidth cost minimization problem in (33). Thus, for large system size k, our primal-dual update algorithm, described by the process $X_k(t)$, operates close to the optimal point.*

Proof. Let $x^* = (\{\theta^*_{sc}\}, \{r^*_{s\ell c}\}, \{q^*_s\}, \{p^*_{s\ell c}\})$ be a stationary point of the above system of ODE. Note that, x^* satisfies

$$q^*_s\Big(\sum_c \theta^*_{sc} - M_s\Big) = 0 \tag{37a}$$

$$p^*_{s\ell c}\Big(\lambda_{\ell c}\gamma^*_{s\ell c} - \lambda_{\ell c}\theta^*_{sc}\Big) = 0 \tag{37b}$$

$$-q^*_s + \sum_\ell A_{\ell s} p^*_{s\ell c} = 0 \tag{37c}$$

$$-r^*_{s\ell c} + \lambda_{\ell c}\gamma^*_{s\ell c} = 0. \tag{37d}$$

Proof of Optimality : The Lagrangian of the problem in (33) is given by

$$\begin{aligned} L &= L(\{\theta_{sc}\}, \{r_{s\ell c}\}, \{q_s\}, \{p_{s\ell c}\}, \{\beta_{s\ell c}\}, \{\delta_{\ell c}\}) \qquad (38)\\ &= \sum_s C_s\Big(\sum_{\ell c} A_{\ell s} r_{s\ell c}\Big) + \sum_s q_s\Big(\sum_c \theta_{sc} - M_s\Big) \\ &\quad + \sum_{s\ell c} A_{\ell s} p_{s\ell c}\Big(r_{s\ell c} - \lambda_{\ell c}\theta_{sc}\Big) - \sum_{s\ell c} A_{\ell s}\beta_{s\ell c} r_{s\ell c} \\ &\quad + \sum_{\ell c}\delta_{\ell c}\Big(\lambda_{\ell c} - \sum_s A_{\ell s} r_{s\ell c}\Big), \qquad (39)\end{aligned}$$

As in the optimality proof of Theorem 1, let us define $\beta^*_{s\ell c}$ and $\delta^*_{\ell c}$, the dual variables associated with constraints (33c) and (33f), as follows : $\delta^*_{\ell c} = \min_s\{\pi^*_{s\ell c} : A_{\ell s} = 1\}$ and $\beta^*_{s\ell c} = \pi^*_{s\ell c} - \delta^*_{\ell c}$. Thus, we have

$$\beta^*_{s\ell c} + \delta^*_{\ell c} = \pi^*_{s\ell c} = p^*_s + p^*_{s\ell c} \tag{40}$$

where, $p^*_s = C'_s(\sum_{\ell c} A_{\ell s} r^*_{s\ell c})$. With the above definition the slackness condition, $\beta^*_{s\ell c} r^*_{s\ell c} = 0$, is satisfied : since $r^*_{s\ell c} > 0$ implies $\gamma^*_{s\ell c} > 0$ (see (37d)) which means that s is a minimum price BS, so that we have $\pi^*_{s\ell c} = \delta^*_{\ell c}$, and hence $\beta^*_{s\ell c} = 0$.

Let $\{q^*_s\}$ and $\{p^*_{s\ell c}\}$ be the dual variables associated with the memory and the content availability constraints in (33d) and (33e), respectively. From (37a) note that q^*_s already satisfies the slackness condition. Using (37d) in (37b), we similarly see that $p^*_{s\ell c}$ also satisfies the slackness condition.

Finally, the proof is complete by observing that $(\{\theta^*_{sc}\}, \{r^*_{s\ell c}\})$ along with the dual variables $(\{q^*_s\}, \{p^*_{s\ell c}\}, \{\beta^*_{s\ell c}\}, \{\delta^*_{\ell c}\})$ satisfies the remaining KKT conditions.

Proof of Convergence : We neglect the time variable t from our notation as before. Recalling that $x = (\{\theta_{sc}\}, \{r_{s\ell c}\}, \{q_s\}, \{p_{s\ell c}\})$, consider the following Lyapunov function :

$$V(x) = L^*(x) + S(x) \tag{41}$$

where,

$$\begin{aligned} L^*(x) &= L(\{\theta_{sc}\}, \{r_{s\ell c}\}, \{q^*_s\}, \{p^*_{s\ell c}\}, \{\beta^*_{s\ell c}\}, \{\delta^*_{\ell c}\}) \qquad (42)\\ S(x) &= \sum_{sc}\frac{1}{2\epsilon\nu_{sc}}(\theta_{sc} - \theta^*_{sc})^2 + \sum_s \frac{1}{2\zeta_s}(q_s - q^*_s)^2\end{aligned}$$

$$+\sum_{s\ell c}\frac{1}{2\xi_{s\ell c}}A_{\ell s}(p_{s\ell c}-p^*_{s\ell c})^2. \tag{43}$$

Thus, we are using the Lagrangian function (evaluated at the dual optimal) along with quadratic terms as our candidate Lyapunov function. As in the convergence proof of Theorem 1, the proposed Lyapunov function easily satisfies conditions (i) and (ii) from [23, Theorem 4.2] ; the challenge remains in proving condition (iii).

First, the time derivative of L^* can be bounded as follows :

$$\begin{aligned}\dot{L}^*(x) &= \sum_{s\ell c}A_{\ell s}\dot{r}_{s\ell c}\Big(p_s+p^*_{s\ell c}-\beta^*_{s\ell c}-\delta^*_{\ell c}\Big)\\ &\quad+\sum_{sc}\dot{\theta}_{sc}\Big(q^*_s-\sum_{\ell}A_{\ell s}p^*_{s\ell c}\Big).\end{aligned} \tag{44}$$

Recalling (36d), (37c) and (40), the above expression can be simplified to obtain

$$\dot{L}^*(x) = \sum_{s\ell c}A_{\ell s}(-r_{s\ell c}+\lambda_{\ell c}\gamma_{s\ell c})(p_s-p^*_s). \tag{45}$$

As in the convergence proof of Theorem 1, defining $g_s := \sum_{\ell c}A_{\ell c}r_{s\ell c}$, we have (since $p_s = C_s'(g_s)$ is strictly increasing; recall (27))

$$(p_s-p^*_s)g_s \geq (p_s-p^*_s)g^*_s,$$

with equality iff $g_s = g^*_s$. Using the above in (45) and rearranging we obtain

$$\dot{L}^*(x) \leq \sum_{s\ell c}A_{\ell s}(p_s-p^*_s)\Big(\lambda_{\ell c}\gamma_{s\ell c}-r^*_{s\ell c}\Big), \tag{46}$$

with equality iff $x = x^*$.

Next, consider the quadratic term $S(x)$. Its time derivative is given by

$$\begin{aligned}\dot{S}(x) &= \sum_{sc}\frac{1}{\epsilon\nu_{sc}}(\theta_{sc}-\theta^*_{sc})\dot{\theta}_{sc}+\sum_{s}\frac{1}{\zeta_s}(q_s-q^*_s)\dot{q}_s\\ &\quad+\sum_{s\ell c}\frac{1}{\xi_{s\ell c}}A_{\ell s}(p_{s\ell c}-p^*_{s\ell c})\dot{p}_{s\ell c}\\ &= T_1+T_2+T_3,\end{aligned} \tag{47}$$

where T_1, T_2 and T_3 are the respective terms in the RHS of the above expression. T_2 can be bounded as follows :

$$\begin{aligned}T_2 &= \sum_{s}\frac{1}{\zeta_s}(q_s-q^*_s)\dot{q}_s\\ &= \sum_{s}(q_s-q^*_s)\Big(\sum_{c}\theta_{sc}-M_s\Big)^+_{q_s}\\ &\leq \sum_{s}(q_s-q^*_s)\Big(\sum_{c}\theta_{sc}-M_s\Big)\\ &= \sum_{s}(q_s-q^*_s)\Big(\sum_{c}\theta_{sc}-\sum_{c}\theta^*_{sc}\Big)\end{aligned}$$

$$+\sum_s (q_s - q_s^*)\Big(\sum_c \theta_{sc}^* - M_s\Big)$$
$$\leq \sum_s (q_s - q_s^*)\Big(\sum_c \theta_{sc} - \sum_c \theta_{sc}^*\Big). \quad (48)$$

Similarly, T_3 can be bounded as

$$T_3 \leq \sum_{s\ell c} A_{\ell s}(p_{s\ell c} - p_{s\ell c}^*)\Big(\lambda_{\ell c}\gamma_{s\ell c} - \lambda_{\ell c}\theta_{sc} - \lambda_{\ell c}\gamma_{s\ell c}^* + \lambda_{\ell c}\theta_{sc}^*\Big). \quad (49)$$

Using (48) and (49) in (47), and rearranging, we obtain :

$$\dot{S}(x) \leq \sum_{s\ell c} A_{\ell s}(p_{s\ell c} - p_{s\ell c}^*)(\lambda_{\ell c}\gamma_{s\ell c} - \lambda_{\ell c}\gamma_{s\ell c}^*)$$
$$+\sum_{sc}(\theta_{sc} - \theta_{sc}^*)\Big(\frac{\dot{\theta}_{sc}}{\epsilon\nu_{sc}} + q_s - q_s^* + \sum_l A_{\ell s}\lambda_{\ell c}(p_{s\ell c}^* - p_{s\ell c})\Big)$$

The second term in the RHS of the above expression is 0; to see this, substitute for $\dot{\theta}_{sc}$ from (36c), and then use the fact that the stationary point satisfies $q_s^* = \sum_l A_{\ell s}\lambda_{\ell c}p_{s\ell c}^*$. Further, using (37d), we have

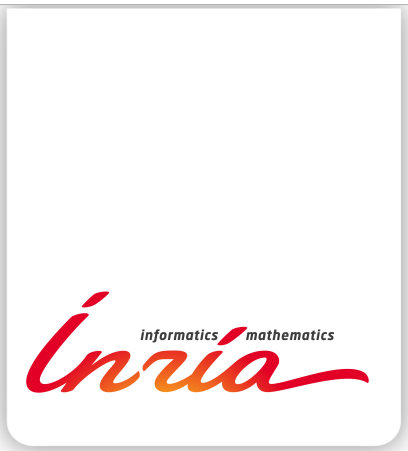
$$\dot{S}(x) \leq \sum_{s\ell c} A_{\ell s}(p_{s\ell c} - p_{s\ell c}^*)(\lambda_{\ell c}\gamma_{s\ell c} - r_{s\ell c}^*). \quad (50)$$

Combining (46) and (50) we can write (recall that $\pi_{s\ell c} = p_s + p_{s\ell c}$)

$$\dot{V}(x) \leq \sum_{s\ell c} A_{\ell s}(\pi_{s\ell c} - \pi_{s\ell c}^*)(\lambda_{\ell c}\gamma_{s\ell c} - r_{s\ell c}^*),$$

with equality iff $x = x^*$. The proof is complete by recalling the argument following (31) in the proof of Theorem 1.

□



RESEARCH CENTRE
SACLAY – ÎLE-DE-FRANCE

1 rue Honoré d'Estienne d'Orves
Bâtiment Alan Turing
Campus de l'École Polytechnique
91120 Palaiseau

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399