

Caching Encrypted Content via Stochastic Cache Partitioning

Andrea Araldo*, György Dán†, Dario Rossi*

* Telecom ParisTech, Paris, France – {andrea.araldo,dario.rossi}@telecom-paristech.fr

† School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden – gyuri@ee.kth.se

Abstract—In-network caching is an appealing solution to cope with the increasing bandwidth demand of video, audio and data transfer over the Internet. Nonetheless, in order to protect consumer privacy and their own business, Content Providers (CPs) increasingly deliver encrypted content, thereby preventing Internet Service Providers (ISPs) from employing traditional caching strategies, which require the knowledge of the objects being transmitted.

To overcome this emerging tussle between security and efficiency, in this paper we propose an architecture in which the ISP partitions the cache space into slices, assigns each slice to a different CP, and lets the CPs remotely manage their slices. This architecture enables transparent caching of encrypted content, and can be deployed in the very edge of the ISP’s network (i.e., base stations, femtocells), while allowing CPs to maintain exclusive control over their content. We propose an algorithm, called SDCP, for partitioning the cache storage into slices so as to maximize the bandwidth savings provided by the cache. A distinctive feature of our algorithm is that ISPs only need to measure the *aggregated miss rates* of each CP, but they need not know of the individual objects that are requested. We prove that the SDCP algorithm converges to a partitioning that is close to the optimal, and we bound its optimality gap. We use simulations to evaluate SDCP’s convergence rate under stationary and non-stationary content popularity. Finally, we show that SDCP significantly outperforms traditional reactive caching techniques, considering both CPs with perfect and with imperfect knowledge of their content popularity.

I. INTRODUCTION

It is widely known that content delivery over the Internet represents a sizeable and increasing fraction of the overall traffic demand. Furthermore most of the content, including video, is carried over HTTP: this evolution of the last decade was not among those forecasted for the IP hourglass model evolution [1], and is rather a choice of practical convenience. This evolution has a tremendous practical relevance, to the point that HTTP was overtly recognized [2] and proposed [3] as the new de facto “thin waist” of the TCP/IP protocol family.

Recently, we are on the verge of yet another shift of the thin waist: we observe that the fraction of traffic delivered through HTTPS has already passed 50% [4], and it is expected to increase, as the IETF Internet Architecture Board (IAB) recommends “protocol designers, developers, and operators to make encryption the norm for Internet traffic” [5]. Besides the IAB recommendation, Content Providers (CP) are already heavily relying on encryption to both protect the privacy of their users, as well as sensitive information (related to user preferences) of their own business.

This evolution toward an all-encrypted Internet creates a tussle between security and efficiency. Today’s Internet heavily relies on middleboxes such as NATs (to combat the scarcity of IPv4 addresses) and transparent or proxy caches [6] (to relieve traffic load). However, some of these middleboxes simply fail to operate in today’s Internet with end-to-end encryption: for example, end-to-end encryption renders caching useless, since (i) ISPs just observe streams of seemingly random bytes with no visibility of objects and cannot select what to cache and (ii) multiple transfers of the same object generate different streams since the same object is encrypted with different keys, thus destroying redundancy that caching would exploit to relieve traffic load. At times where the design of the new 5G architecture strives to reduce latency, increase the available bandwidth and better handle mobility, this tradeoff is especially unfortunate, as distributed caches represent a natural way to reduce latency, reduce bandwidth usage and to cope with mobility avoiding long detours to anchor points [7], [8].

This architectural evolution calls for a redesign of the current operations involving both Internet Service Providers (ISP) and Content Providers (CP): by design, the solutions should preserve business-critical CP information (e.g., content popularity, user preferences) on the one hand, while allowing for a deeper integration of caches in the ISP architecture (e.g., in 5G femto-cells) on the other hand [9].

In this paper we address this issue by making the following contributions:

- We propose an architecture in which the ISP partitions its cache storage into slices and each CP manages its slice by caching its most popular content, without having to reveal sensitive information, such as content popularity, to the ISP.
- We propose an algorithm called *Stochastic Dynamic Cache Partitioning* (SDCP) based on stochastic sub-gradient descent, which *dynamically partitions* the cache storage among various CPs so as to maximize the cache hit rate (hence, the bandwidth savings). To protect the business-critical information of the CPs, the algorithm only needs to measure the *aggregate miss rates* of the individual CPs.
- We prove that the proposed SDCP algorithm converges close to the allocation that maximizes the overall cache hit rate, and provide a bound on the optimality gap.
- We show that the computational complexity of each step of the algorithm is polynomial in the number of CPs, which makes the algorithm easy to implement in practice.

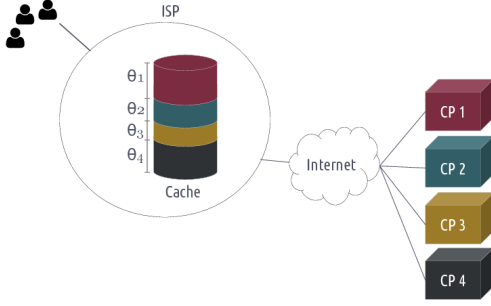


Figure 1. System model. The ISP partitions the cache space of K slots among $P = 4$ CPs.

- By means of an extensive performance evaluation via simulation, we contrast our algorithm to classic caching schemes, showing that gains maintain even in realistic cases that include (i) time-varying content popularity as well as (ii) imperfect knowledge of content popularity.

The rest of the paper is organized as follows. Sec. II describes the system model and objectives. Sec. III introduces our cache partitioning algorithm, of which we provide proof of convergence in Sec. IV and a complexity analysis in Sec. V. Sec. VI evaluates the performance of the algorithm through simulations. Sec. VII reviews related work, and Sec. VIII concludes the paper.

II. SYSTEM MODEL AND PROBLEM FORMULATION

In this section we first describe the system model and introduce the main notation (Sec. II-A). We then formulate the objective of the ISP that owns the cache in terms of miss-rate minimization (Sec. II-B).

A. System Model

We consider a cache with a storage size of K slots (e.g., in units of MB) maintained by an operator, likely an ISP. As illustrated in Fig. 1, the operator uses the cache to offer cache-as-a-service to P Content Providers (CPs) by partitioning the set of slots and assigning each partition slice to a different CP.

We denote by $\theta_p \in \mathbb{Z}_{\geq 0}$ the number of cache slots allocated to CP p , which it can use for caching its most popular content. We define the set of feasible cache allocation vectors

$$\Theta \triangleq \{\boldsymbol{\theta} \in \mathbb{Z}_{\geq 0}^P \mid \sum_{p=1}^P \theta_p \leq K\} \subset \mathbb{Z}_{\geq 0}^P. \quad (1)$$

We consider that the arrival of requests for content can be modeled by a stationary process, and the number of arrivals over a time interval of length T can be *bounded* by some positive constant $A(T)$. This assumption is reasonable as requests are generated by a finite customer population, and each customer can generate requests at a bounded rate in practice. Upon reception of a request for a content of the CPs that share the cache, the request can either generate a cache hit (for content stored in the CP partition at time of the request)

or a cache miss (otherwise). Formally, we denote by $L_p(\theta_p)$ the *expected* cache miss rate (i.e., expected number of misses per time unit) of CP p when allocated θ_p slots of storage. We make the reasonable assumption that L_p is decreasing and strictly convex on $[0 \dots K]$, which corresponds to that having more storage decreases the miss intensity (in expectation) with a decreasing marginal gain, and each CP would in principle have enough content to fill the entire storage. As an example, if CP caches its θ_p most popular objects (ideal LFU) then $L_p(\theta_p)$ would be the tail distribution (i.e., the complementary cumulative distribution function) of the content popularity distribution (e.g., Zipf). As another example, convexity also holds if the CP uses probabilistic caching combined with LRU, which has been shown to approximate the LFU cache eviction policy as the admission probability approaches zero. We investigate these (and other) examples having practical relevance in the numerical results section.

For convenience, we define the expected cache miss intensity vector

$$\vec{L}(\boldsymbol{\theta}) \triangleq (L_1(\theta_1), \dots, L_P(\theta_P))^T. \quad (2)$$

Finally, we define the overall expected cache miss intensity

$$L(\boldsymbol{\theta}) \triangleq \sum_{p=1}^P L_p(\theta_p). \quad (3)$$

We assume that (i) the content in each cache slice is encrypted by the respective CP so that the ISP cannot read it, that (ii) the ISP cannot observe what content an individual request is for, but that (iii) it can observe the number of content requests received by a CP and the corresponding number of cache misses. By doing so, the CP retains precious business information related to individual content, whereas the ISP retains the ability to measure network-related performance, as it only measures the miss rate of aggregated content. The exact mechanisms for CP and ISP interaction are outside of the scope of our work, but the ongoing standardization effort at the ETSI Industry Specification Group (ISG) on Mobile Edge Computing (MEC) [9] is set out to define a set of APIs to facilitate interactions such as the one we consider in this paper.

Furthermore, we assume that cache misses do not significantly affect the users' behavior, and thus the sequence of cache allocation vectors $\boldsymbol{\theta} \in \Theta$ and the arrival of requests for content is jointly stationary. This is to say that users' taste in terms of content is not affected by the QoS/QoE metrics (e.g., increased delay as a consequence of a cache miss), which reasonably holds in practice. Finally, we assume that each CP p uses a cache policy that results in a stationary placement of files placed in its storage, given the allocated storage size θ_p , e.g., the CP could place its θ_p most popular objects (as with an ideal Least Frequently Used (LFU) policy) or could use one of the many cache eviction and replacement policies (as with a classic Least Recently Used (LRU) policy). As a consequence, the sequence of cache allocation vectors $\boldsymbol{\theta} \in \Theta$ and the sequence of cache misses are pairwise jointly stationary.

Table I
FREQUENTLY USED NOTATION (WITH PLACE OF DEFINITION)

P	Number of content providers (CP)
K	Available cache slots
K'	Allocated cache slots (6)
θ	Cache configuration
Θ	Set of feasible cache allocations (1)
\mathcal{C}	Set of allowed virtual cache allocations (5)
$L(\theta)$	Expected cache miss intensity (3)
\bar{L}	Miss intensity vector (2)
$\bar{L}(\theta)$	Interpolant of the miss intensity (Lemma 8)
θ^*	Unique minimizer of \bar{L} (Lemma 8)
$\mathbf{D}^{(k)}$	Perturbation vector (8)
$+\bar{\mathbf{y}}^{(k)}, -\bar{\mathbf{y}}^{(k)}$	Measured miss stream vector lines 7 and 9 of Alg. 1
T	Time slot length
ϕ	Euclidean projection (10)
$\hat{\mathbf{g}}^{(k)}$	Stochastic subgradient (line 11 of Alg. 1)
$\bar{\mathbf{g}}(\theta)$	Subgradient of \bar{L} (24)
$\Gamma(\theta)$	Center-point function (7)
ρ	Rank Estimation Accuracy (Sec. VI-E)

B. Problem Formulation

Given that the management of each slice is delegated to the respective CP, the only task of the ISP is to decide how to allocate the overall cache space among the different CPs. At a high level, one can approach the problem of cache storage allocation from the perspective of the CPs or from the perspective of the cache owner. Approaching the problem from the perspective of the CPs one can formulate a plethora of objective functions (including abstract notions of utility, based on quality of experience and considering fairness and/or business preferences, etc.). Approaching the problem from the cache owner's perspective, the choice of the objective function is instead much restrained: indeed, considering that caches at the network edge are typically installed for reducing the bandwidth requirements in the backhaul network (i.e., between the edge and the core network), a reasonable objective is maximize the cache hit rate, as that leads to the minimization of bandwidth requirements.

In this work, we adopt the perspective of the cache owner and consider that the ISP is interested in maximizing the bandwidth savings achievable by the available cache storage, and thus its objective is to find the optimal allocation θ^{OPT} that minimizes the overall *expected cache miss intensity*, i.e.,

$$\theta^{OPT} \in \arg \min_{\theta \in \Theta} L(\theta), \quad (4)$$

based on the measured cache miss intensity. In what follows we propose an algorithm that iteratively approximates the optimal allocation θ^{OPT} in case of stationary popularities. Given its iterative nature, the algorithm is also amenable to be run continuously, so to track popularity changes in scenarios with a dynamic catalog.

III. STOCHASTIC DYNAMIC CACHE PARTITIONING

We tackle the miss-rate minimization problem via Stochastic Dynamic Cache Partitioning (SDCP). Our SDCP proposal is an iterative algorithm that is executed over time slots of fixed duration T . The pseudo code of the algorithm is shown in Alg. 1. For simplicity we present the algorithm assuming that P is even, but the case of an odd number of CPs can

Algorithm 1: Stochastic Dynamic Cache Partitioning

```

1 Choose an initial allocation  $\theta_0 \in \mathcal{C} \cap \mathbb{R}_{\geq 0}^P$ 
2 for  $k = 0$ ;  $k++$  do
3   Generate  $\mathbf{D}^{(k)}$ 
4    $+\theta^{(k)} = \Gamma(\theta^{(k)}) + \frac{1}{2}\mathbf{D}^{(k)}$ 
5    $-\theta^{(k)} = \Gamma(\theta^{(k)}) - \frac{1}{2}\mathbf{D}^{(k)}$ 
6   Set the configuration to  $+\theta^{(k)}$  for time  $T/2$ 
7   Measure  $+\bar{\mathbf{y}}^{(k)}$ 
8   Set the configuration to  $-\theta^{(k)}$  for a time  $T/2$ 
9   Measure  $-\bar{\mathbf{y}}^{(k)}$ 
10   $\delta\bar{\mathbf{y}}^{(k)} = +\bar{\mathbf{y}}^{(k)} - -\bar{\mathbf{y}}^{(k)}$ 
11   $\hat{\mathbf{g}}^{(k)} = \delta\bar{\mathbf{y}}^{(k)} \circ \mathbf{D}^{(k)} - \frac{1}{P} \cdot (\delta\bar{\mathbf{y}}^{(k)T} \cdot \mathbf{D}^{(k)})\mathbf{1}_P$ 
12   $\theta^{(k+1)} = \phi(\theta^{(k)} - a^{(k)}\hat{\mathbf{g}}^{(k)})$ 
13 end

```

be handled by introducing a fictitious CP with zero request rate. Unless otherwise stated, we treat the case of stationary popularities, and defer the case of dynamic popularities to the performance evaluation section.

At time slot k the algorithm maintains a virtual cache allocation $\theta^{(k)}$. The virtual allocation is an allocation of K' storage slots among the CPs, i.e.,

$$\theta^{(k)} \in \mathcal{C} \triangleq \left\{ \theta \in \mathbb{R}^P \mid \mathbf{1}_P^T \cdot \theta \triangleq K' \right\}, \quad (5)$$

where

$$K' = K - P/2. \quad (6)$$

We will justify the introduction of K' and of \mathcal{C} in the proof of Lemma 6.

In order to obtain from $\theta^{(k)}$ an integral allocation that can be implemented in the cache, we define the center-point function $\Gamma: \mathbb{R}^P \rightarrow \mathbb{R}^P$, which assigns to a point in Euclidean space the center of the hypercube containing it, i.e.,

$$\begin{aligned} \gamma(x) &\triangleq \lfloor x \rfloor + 1/2, & \forall x \in \mathbb{R}, \\ \Gamma(\theta) &\triangleq (\gamma(\theta_1), \dots, \gamma(\theta_P))^T, & \forall \theta \in \mathcal{C}, \end{aligned} \quad (7)$$

where we use $\lfloor \cdot \rfloor$ to denote the floor of a scalar or of a vector in the component-wise sense. Furthermore, we define the perturbation vector $\mathbf{D}^{(k)} = (D_1^{(k)}, \dots, D_P^{(k)})^T$ at time slot k , which is chosen independently and uniform at random from the set of $-1, +1$ valued zero-sum vectors

$$\mathbf{D}^{(k)} \in \mathcal{Z} \triangleq \{ \mathbf{z} \in \{-1, 1\}^P \mid \mathbf{z}^T \cdot \mathbf{1}_P = 0 \}. \quad (8)$$

Given Γ and $\mathbf{D}^{(k)}$ the algorithm computes two cache allocations to be implemented during time slot k ,

$$\begin{aligned} +\theta^{(k)} &\triangleq \Gamma(\theta^{(k)}) + \frac{1}{2}\mathbf{D}^{(k)}, \\ -\theta^{(k)} &\triangleq \Gamma(\theta^{(k)}) - \frac{1}{2}\mathbf{D}^{(k)}. \end{aligned} \quad (9)$$

The algorithm first applies allocation $+\theta^{(k)}$ for $T/2$ amount of time and measures the cache miss rate $+y_p^{(k)}$ for each provider $p = 1, \dots, P$, i.e. the amount of objects that are not found in the cache and retried from the CP server. It then applies allocation $-\theta^{(k)}$ during the remaining $T/2$ amount of time in slot k and measures the cache miss rates $-y_p^{(k)}$. The vectors of measured cache misses $-\bar{\mathbf{y}}^{(k)} \triangleq (-y_1^{(k)}, \dots, -y_P^{(k)})^T$ and $+\bar{\mathbf{y}}^{(k)} \triangleq (+y_1^{(k)}, \dots, +y_P^{(k)})^T$ are used to compute the impact

$\delta y_p^{(k)} \triangleq y_p^{(k)} - y_p^{(k)}$ of the perturbation vector on the cache miss intensity of CP p , or using the vector notation $\delta \bar{y}^{(k)} \triangleq \bar{y}^{(k)} - \bar{y}^{(k)}$.

Based on the measured miss rates, the algorithm then computes the allocation vector $\theta^{(k+1)}$ for the $(k+1)$ -th step. Specifically, it first computes (line 11, where \circ denotes the Hadamard product) the update vector $\hat{\mathbf{g}}^{(k)}$, which we show in Cor. 12 to match in expectation a subgradient of the miss-stream interpolant \bar{L} , defined in Lemma 8. The $(k+1)$ -th allocation moves from the k -th allocation in the direction of the update vector $\hat{\mathbf{g}}^{(k)}$, opportunely scaled by a *step size* $a^{(k)} > 0$. Additionally, denoting with $\mathbb{R}_{\geq 0}$ the set of non-negative numbers, $\theta^{(k+1)}$ is computed using the Euclidean projection $\varphi : \mathcal{C} \rightarrow \mathcal{C} \cap \mathbb{R}_{\geq 0}^P$, defined as

$$\varphi(\theta) \triangleq \arg \min_{\theta' \in \mathcal{C} \cap \mathbb{R}_{\geq 0}^P} \|\theta - \theta'\|. \quad (10)$$

Several remarks are worth making. First, we will show in Lemma 5 that the equation above admits a unique solution and thus the definition is consistent. Second, we will show in Lemma 6 that $\hat{\mathbf{g}}$ computed as in line 11 guarantees that the update $\theta^{(k)} - a^{(k)} \hat{\mathbf{g}}^{(k)}$ at line 12 lies inside \mathcal{C} . Nonetheless, this update may have some negative components and we need to project it into $\mathcal{C} \cap \mathbb{R}_{\geq 0}$ by applying φ , to ensure that the subsequent virtual allocation $\theta^{(k+1)}$ is valid. Third, the *step size* $a^{(k)}$ must be chosen to satisfy

$$a^{(k)} > 0, \forall k > 0 \quad (11)$$

$$\sum_{k=1}^{\infty} a^{(k)} = \infty \quad (12)$$

$$\sum_{k=1}^{\infty} (a^{(k)})^2 < \infty \quad (13)$$

in order to guarantee convergence (see Theor. 14). Fourth, although the convergence of the proposed algorithm is guaranteed, for stationary content popularity, irrespectively of the choice of $a^{(k)}$ satisfying the above conditions, we point out that the step size plays an important role in determining the convergence speed, which we will numerically investigate in Sec. VI.

IV. CONVERGENCE ANALYSIS OF SDCP

We first provide definitions and known results (Sec.IV-A) that are instrumental to prove important properties of the proposed algorithm: consistency (Sec.IV-B), convergence (Sec.IV-C) and a bound on the optimality gap (Sec.IV-D).

A. Preliminaries

Let us start by introducing the forward difference defined for functions on discrete sets.

Definition 1. For a function $\mathbf{F} : \mathbb{Z}^{q_1} \rightarrow \mathbb{R}^{q_2}$, $q_1, q_2 \geq 1$ the forward difference is

$$\Delta_n \mathbf{F}(\mathbf{x}) \triangleq \mathbf{F}(\mathbf{x} + n \cdot \mathbf{1}_{q_1}) - \mathbf{F}(\mathbf{x}), \forall \mathbf{x} \in \mathbb{Z}^{q_1}, n \in \mathbb{Z} \setminus \{0\},$$

where $\mathbf{1}_q$ is the column vector of all ones of dimension q . By abuse of notation, we will simply use $\Delta \mathbf{F}(\mathbf{x})$ to denote $\Delta_1 \mathbf{F}(\mathbf{x})$.

The forward difference is convenient for characterizing convexity using the following definition [10].

Definition 2. A discrete function $F : \mathbb{Z} \rightarrow \mathbb{R}$ is strictly convex iff $x \rightarrow \Delta F(x)$ is strictly increasing.

Furthermore, for a class of functions of interest we can establish the following.

Lemma 3. Let $F : \mathbb{Z} \rightarrow \mathbb{R}$ be decreasing and strictly convex, $x \in \mathbb{Z}$ and $n \in \mathbb{Z} \setminus \{0\}$, then we have

$$\Delta_n F(x) > n \Delta F(x). \quad (14)$$

Proof:

We first show that $\forall x, y \in \mathbb{Z}$ such that $y > x$, the following holds

$$\Delta_n F(y) > \Delta_n F(x) \quad \text{if } n > 0, \quad (15)$$

$$\Delta_n F(y) < \Delta_n F(x) \quad \text{if } n < 0. \quad (16)$$

For $n > 0$ we can use Def. 2 to obtain

$$\begin{aligned} \Delta_n F(y) &= \sum_{i=0}^{n-1} [F(y+i+1) - F(y+i)] = \sum_{i=0}^{n-1} \Delta F(y+i) \\ &> \sum_{i=0}^{n-1} \Delta F(x+i) = \Delta_n F(x), \end{aligned} \quad (17)$$

which proves (15).

For $n < 0$ algebraic manipulation of the definition of the forward difference and (17) gives

$$\Delta_n F(y) = -\Delta_{|n|} F(y - |n|) < -\Delta_{|n|} F(x - |n|) = \Delta_n F(x),$$

which proves (16). To prove (14) for $n > 0$, observe that thanks to Def. 2, each of the n terms of the last summation in (15) is lower bounded by $\Delta F(x)$. For $n < 0$ via algebraic manipulation we obtain

$$\Delta_n F(x) = -\sum_{i=1}^{|n|} \Delta F(x-i) > -\sum_{i=1}^{|n|} \Delta F(x) = -|n| \cdot \Delta F(x),$$

which proves (14) as $|n| = -n$. ■

Since SDCP generates virtual configurations whose components are not necessarily integer, we have to extend the discrete functions L_p to real numbers. Thanks to Theor. 2.2 of [11], we have the following existence result.

Lemma 4. Given a discrete decreasing and strictly convex function $F : \mathbb{Z} \rightarrow \mathbb{R}$, there exists a continuous and strictly convex function $\bar{F} : \mathbb{R} \rightarrow \mathbb{R}$ that extends F , i.e., $F(x) = \bar{F}(x), \forall x \in \mathbb{Z}$. We call \bar{F} the interpolant of F .

Finally, we formulate an important property of the Euclidean projection φ .

Lemma 5. There is a unique function φ satisfying (10). Furthermore, φ satisfies

$$\|\varphi(\theta) - \theta'\| \leq \|\theta - \theta'\|, \forall \theta \in \mathcal{C}, \theta' \in \mathcal{C} \cap \mathbb{R}_{\geq 0}^P, \quad (18)$$

i.e., $\varphi(\theta)$ is no farther from any allocation vector than θ .

Proof: Observe that $\mathcal{C} \cap \mathbb{R}_{\geq 0}^P$ is a simplex, and thus closed and convex. Hence, the Euclidean projection φ is the unique

solution of (10) [12]. Furthermore, the Euclidean projection is non-expansive (see, e.g., Fact 1.5 in [13]), i.e., for $\theta, \theta' \in \mathcal{C}$ it satisfies $\|\varphi(\theta) - \varphi(\theta')\| \leq \|\theta - \theta'\|$. Observing that if $\theta' \in \mathcal{C} \cap \mathbb{R}_{\geq 0}^P$ then $\varphi(\theta') = \theta'$ proves the result. ■

B. Consistency

We first have to prove that during each time slot the configurations $-\theta^{(k)}, +\theta^{(k)}$ that SDCP imposes on the cache are feasible. This is non-trivial, as the operators used in computing the allocations are defined on proper subsets of \mathbb{R}^P . The following lemma establishes that the allocations computed by SDCP always fall into these subsets.

Lemma 6. *The allocations $\theta^{(k)}$ are consistent in every time slot, as they satisfy*

- (a) $\theta^{(k)} - a_k \hat{\mathbf{g}}^{(k)} \in \mathcal{C}$,
- (b) $\theta^{(k+1)} \in \mathcal{C} \cap \mathbb{R}_{\geq 0}^P$,
- (c) $+\theta^{(k)}, -\theta^{(k)} \in \Theta$.

Proof: Recall that $\theta_0 \in \mathcal{C} \cap \mathbb{R}_{\geq 0}^P$. To show (a) observe that

$$\hat{\mathbf{g}}^{(k)} \cdot \mathbf{1}_P = \sum_{p=1}^P \delta y_p^{(k)} \cdot D_p^{(k)} - \sum_{p=1}^P \delta y_p^{(k)} \cdot D_p^{(k)} = 0, \quad (19)$$

and thus if $\theta^{(k)} \in \mathcal{C}$, then $\theta^{(k)} - a^{(k)} \hat{\mathbf{g}}^{(k)} \in \mathcal{C}$. The definition of the Euclidean projection (10) and (a) together imply (b). Finally, observe that if $\mathbf{D}^{(k)} = (z_1, \dots, z_P)^T \in \mathcal{Z}$, as in (8), then since half of the z_p equal -1 and half of them equal $+1$, we can write

$$\begin{aligned} \mathbf{1}^T \cdot +\theta^{(k)} &= \sum_{p=1}^P \left[\lfloor \theta_p^k \rfloor + \frac{1+z_p}{2} \right] = \\ &= \mathbf{1}^T \cdot \lfloor \theta \rfloor + \frac{P}{2} \leq \mathbf{1}^T \cdot \theta + \frac{P}{2} \leq K' + \frac{P}{2} = K, \end{aligned} \quad (20)$$

which proves (c). Note that the above motivates the choice of K' in the definition of the set of virtual allocations \mathcal{C} , as if $K' > K - \frac{P}{2}$ then $+\theta^{(k)}, -\theta^{(k)} \in \Theta$ may be violated due to the use of the mapping γ and $\mathbf{D}^{(k)}$ in (9). ■

C. Convergence

To prove convergence of SDCP, we first consider the relationship between the measured miss rates $+y_p^{(k)}$ and $-y_p^{(k)}$ and the expected miss intensities $L_p(+\theta_p^{(k)})$ and $L_p(-\theta_p^{(k)})$, respectively. We define the measurement noise

$$\begin{aligned} +\boldsymbol{\varepsilon}^{(k)} &\triangleq +\bar{\mathbf{y}}^{(k)} - \bar{\mathbf{L}}(+\boldsymbol{\theta}^{(k)}), \\ -\boldsymbol{\varepsilon}^{(k)} &\triangleq -\bar{\mathbf{y}}^{(k)} - \bar{\mathbf{L}}(-\boldsymbol{\theta}^{(k)}) \end{aligned} \quad (21)$$

and the corresponding differences

$$\begin{aligned} \delta \boldsymbol{\varepsilon}^{(k)} &\triangleq +\boldsymbol{\varepsilon}^{(k)} - -\boldsymbol{\varepsilon}^{(k)}, \\ \delta \bar{\mathbf{L}}^{(k)} &\triangleq \bar{\mathbf{L}}(+\boldsymbol{\theta}^{(k)}) - \bar{\mathbf{L}}(-\boldsymbol{\theta}^{(k)}). \end{aligned} \quad (22)$$

Observe that $\mathbf{D}^{(k)}, +\bar{\mathbf{y}}^{(k)}$ and $-\bar{\mathbf{y}}^{(k)}$ are random variables and form a stochastic process. Using these definitions we can formulate the following statement about the measured miss rates.

Lemma 7. *The conditional expectation of the measurement noise and its difference satisfy*

$$\mathbb{E}[\delta \boldsymbol{\varepsilon}^{(k)} | \boldsymbol{\theta}^{(k)}] = \mathbf{0}_P. \quad (23)$$

Proof: Observe that due to the stationarity of the request arrival processes we have $\mathbb{E}[+\boldsymbol{\varepsilon}^{(k)} | \boldsymbol{\theta}^{(k)}] = \mathbf{0}$ and $\mathbb{E}[-\boldsymbol{\varepsilon}^{(k)} | \boldsymbol{\theta}^{(k)}] = \mathbf{0}$, which due to the additive law of expectation yields the result. ■

Intuitively, this is equivalent to saying that the sample averages provide an unbiased estimator of the miss rates. In what follows we establish an analogous result for the update vector $\hat{\mathbf{g}}^{(k)}$ with respect to a subgradient of the interpolant \bar{L} of the expected miss intensity L , which itself is a discrete function. We define and characterize \bar{L} in the following lemma, which recalls known results from convex optimization.

Lemma 8. *Given the interpolants \bar{L}_p of the expected miss intensities L_p of the CPs and defining the interpolant of L as $\bar{L}(\boldsymbol{\theta}) \triangleq \sum_{p=1}^P \bar{L}_p(\theta_p), \forall \boldsymbol{\theta} \in \mathbb{R}_{\geq 0}^P$, \bar{L} is strictly convex and admits a unique minimizer $\boldsymbol{\theta}^* \in \mathcal{C} \cap \mathbb{R}_{\geq 0}^P$.*

Proof: Recall that each interpolant \bar{L}_p of L_p is strictly convex as shown in Lemma 4. The strict convexity of \bar{L} can then be obtained applying Proposition 1 of [14]. Then, we observe that $\boldsymbol{\theta}^*$ is the solution to a convex optimization problem with a strictly convex objective function, which is unique (Sec. 4.2.1 of [15]). ■

For completeness, let us recall the definition of a subgradient of a function from (see, e.g., [14]).

Definition 9. Given a function $\bar{L} : \mathbb{R}^P \rightarrow \mathbb{R}$, a function $\bar{\mathbf{g}} : \mathcal{C} \subseteq \mathbb{R}^P \rightarrow \mathbb{R}^P$ is a subgradient of \bar{L} over \mathcal{C} iff

$$\bar{L}(\boldsymbol{\theta}') - \bar{L}(\boldsymbol{\theta}) \geq \bar{\mathbf{g}}(\boldsymbol{\theta})^T \cdot (\boldsymbol{\theta}' - \boldsymbol{\theta}), \forall \boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathcal{C}.$$

We are now ready to introduce a subgradient $\bar{\mathbf{g}}(\boldsymbol{\theta})$ for the interpolant of the expected cache miss intensity \bar{L} .

Lemma 10. *The function*

$$\bar{\mathbf{g}}(\boldsymbol{\theta}) \triangleq \Delta \bar{\mathbf{L}}^{(k)}(\lfloor \boldsymbol{\theta} \rfloor) - \frac{1}{P} \cdot \Delta L(\lfloor \boldsymbol{\theta} \rfloor) \cdot \mathbf{1}_P \quad (24)$$

is a subgradient of \bar{L} over $\mathcal{C} \cap \mathbb{R}_{\geq 0}^P$.

Proof: Observe that for $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathcal{C}$

$$\begin{aligned} \bar{\mathbf{g}}(\boldsymbol{\theta})^T \cdot (\boldsymbol{\theta}' - \boldsymbol{\theta}) &= \Delta \bar{\mathbf{L}}^{(k)}(\lfloor \boldsymbol{\theta} \rfloor)^T \cdot (\boldsymbol{\theta}' - \boldsymbol{\theta}) \\ &\quad - \frac{1}{P} \cdot \Delta L(\lfloor \boldsymbol{\theta} \rfloor) \cdot [\mathbf{1}_P^T \cdot (\boldsymbol{\theta}' - \boldsymbol{\theta})]. \end{aligned}$$

At the same time, for $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathcal{C}$ we have

$$\mathbf{1}_P^T \cdot (\boldsymbol{\theta}' - \boldsymbol{\theta}) = (\mathbf{1}_P^T \cdot \boldsymbol{\theta}' - \mathbf{1}_P^T \cdot \boldsymbol{\theta}) = K' - K' = 0.$$

Therefore, for any $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathcal{C}$

$$\bar{\mathbf{g}}(\boldsymbol{\theta})^T \cdot (\boldsymbol{\theta}' - \boldsymbol{\theta}) = \Delta \bar{\mathbf{L}}^{(k)}(\lfloor \boldsymbol{\theta} \rfloor) \cdot (\boldsymbol{\theta}' - \boldsymbol{\theta}). \quad (25)$$

Thus, according to Def. 9, in order to show that $\bar{\mathbf{g}}$ is a subgradient of \bar{L} it suffices to show that

$$\sum_{p=1}^P [\bar{L}_p(\theta'_p) - \bar{L}_p(\theta_p)] \geq \sum_{p=1}^P \Delta L_p(\lfloor \theta_j \rfloor) \cdot (\theta'_p - \theta_p). \quad (26)$$

We now show that this holds component-wise. If $\lfloor \theta'_p \rfloor - \lfloor \theta_p \rfloor = 0$, then the above clearly holds. Otherwise, if $n = \lfloor \theta'_p \rfloor - \lfloor \theta_p \rfloor \neq 0$ we apply a well known property of convex functions (Theor. 1.3.1 of [16]) to obtain:

$$\begin{aligned} \frac{\bar{L}_p(\lfloor \theta'_p \rfloor) - \bar{L}_p(\lfloor \theta_p \rfloor)}{(\lfloor \theta'_p \rfloor - \lfloor \theta_p \rfloor)} &\leq \frac{\bar{L}_p(\theta'_p) - \bar{L}_p(\theta_p)}{(\theta'_p - \theta_p)} \\ &\leq \frac{\bar{L}_p(\lfloor \theta'_p \rfloor + 1) - \bar{L}_p(\lfloor \theta_p \rfloor + 1)}{(\lfloor \theta'_p \rfloor + 1 - (\lfloor \theta_p \rfloor + 1))}, \end{aligned}$$

which, by Def. 1, can be rewritten as:

$$\frac{\Delta_n L_j(\lfloor \theta_j \rfloor)}{n} \leq \frac{\bar{L}_p(\theta'_p) - \bar{L}_p(\theta_p)}{\theta'_p - \theta_p} \leq \frac{\Delta_n L_j(\lfloor \theta_j + 1 \rfloor)}{n}. \quad (27)$$

For $n > 0$ we can use the first inequality of (27) and Lemma 3 to obtain

$$\bar{L}_p(\theta'_p) - \bar{L}_p(\theta_p) \geq \Delta L_j(\lfloor \theta_j \rfloor) \cdot (\theta'_p - \theta_p). \quad (28)$$

For $n < 0$ we can use the second inequality of (27) and Lemma 3 to obtain

$$\frac{\bar{L}_p(\theta'_p) - \bar{L}_p(\theta_p)}{\theta'_p - \theta_p} \leq \Delta L_j(\lfloor \theta_j + 1 \rfloor) \leq \Delta L_j(\lfloor \theta_j \rfloor) \quad (29)$$

and by multiplying the first and the second term of (29) by $\theta'_p - \theta_p$ (which is negative since $n = \lfloor \theta'_p \rfloor - \lfloor \theta_p \rfloor$ is negative), we obtain the result. \blacksquare

The subgradient $\bar{\mathbf{g}}$ will be central to proving the convergence of SDCP, but it cannot be measured directly. The next proposition establishes a link between the update vector $\hat{\mathbf{g}}^{(k)}$, which we compute in every time slot, and the subgradient $\bar{\mathbf{g}}$.

Proposition 11. *The update vector $\hat{\mathbf{g}}^{(k)}$ is composed of the subgradient $\bar{\mathbf{g}}$ plus a component due to the noise,*

$$\hat{\mathbf{g}}^{(k)} = \bar{\mathbf{g}}(\boldsymbol{\theta}^{(k)}) + \delta \boldsymbol{\varepsilon}^{(k)} \circ \mathbf{D}^{(k)} - \frac{1}{P} \cdot \left[\delta \boldsymbol{\varepsilon}^{(k)T} \cdot \mathbf{D}^{(k)} \right] \mathbf{1}_P.$$

Proof: We first apply (22) to obtain

$$\begin{aligned} \hat{\mathbf{g}}^{(k)} &= \delta \bar{\mathbf{L}}^{(k)} \circ \mathbf{D}^{(k)} - \frac{1}{P} \cdot (\delta \bar{\mathbf{L}}^{(k)T} \cdot \mathbf{D}^{(k)}) \mathbf{1}_P \quad (30) \\ &+ \delta \boldsymbol{\varepsilon}^{(k)} \circ \mathbf{D}^{(k)} - \frac{1}{P} \cdot (\delta \boldsymbol{\varepsilon}^{(k)T} \cdot \mathbf{D}^{(k)}) \mathbf{1}_P. \end{aligned}$$

Consider now a particular realization of the random variable $\mathbf{D}^{(k)}$. We can express component p of $\delta \bar{\mathbf{L}}^{(k)} \circ \mathbf{D}^{(k)} = \left[\bar{\mathbf{L}}(+\boldsymbol{\theta}^{(k)}) - \bar{\mathbf{L}}(-\boldsymbol{\theta}^{(k)}) \right] \circ \mathbf{D}^{(k)}$ as

$$\begin{aligned} &\left[L_p \left(\Pi \left(\theta_p^{(k)} \right) + \frac{1}{2} D_p^{(k)} \right) - L_p \left(\Pi \left(\theta_p^{(k)} \right) - \frac{1}{2} D_p^{(k)} \right) \right] \\ &\quad \cdot D_p^{(k)} \\ &= \left[L_p \left(\Pi \left(\theta_p^{(k)} \right) + \frac{1}{2} \right) - L_p \left(\Pi \left(\theta_p^{(k)} \right) - \frac{1}{2} \right) \right] \\ &= \left[L_p \left(\lfloor \theta_p^{(k)} \rfloor + 1 \right) - L_p \left(\lfloor \theta_p^{(k)} \rfloor \right) \right], \end{aligned}$$

where the first equality can be easily verified assuming that $D_p^{(k)} = -1$ and then assuming that it is $D_p^{(k)} = 1$. We thus obtain

$$\delta \bar{\mathbf{L}}^{(k)} \circ \mathbf{D}^{(k)} = \Delta \bar{\mathbf{L}}(\lfloor \boldsymbol{\theta}^{(k)} \rfloor)$$

and in scalar form

$$\delta \bar{\mathbf{L}}^{(k)T} \cdot \mathbf{D}^{(k)} = \Delta L(\lfloor \boldsymbol{\theta}^{(k)} \rfloor).$$

By substituting these in (30)

$$\begin{aligned} \hat{\mathbf{g}}^{(k)} &= \Delta \bar{\mathbf{L}}(\lfloor \boldsymbol{\theta}^{(k)} \rfloor) - \frac{1}{P} \cdot \Delta L(\lfloor \boldsymbol{\theta}^{(k)} \rfloor) \cdot \mathbf{1}_P \\ &+ \delta \boldsymbol{\varepsilon}^{(k)} \circ \mathbf{D}^{(k)} - \frac{1}{P} \cdot (\delta \boldsymbol{\varepsilon}^{(k)T} \cdot \mathbf{D}^{(k)}) \mathbf{1}_P \end{aligned}$$

and using (24), we obtain the result. \blacksquare

Furthermore, thanks to Lemma 7, the second term of (30), which is due to the noise, is zero in expectation, which provides the link between the update vector $\hat{\mathbf{g}}^{(k)}$ and the subgradient $\bar{\mathbf{g}}(\boldsymbol{\theta}^{(k)})$.

Corollary 12. *The conditional expectation of $\hat{\mathbf{g}}^{(k)}$ is $\mathbb{E}[\hat{\mathbf{g}}^{(k)} | \boldsymbol{\theta}^{(k)}] = \bar{\mathbf{g}}(\boldsymbol{\theta}^{(k)})$ and thus $\hat{\mathbf{g}}^{(k)}$ is a stochastic subgradient of \bar{L} , i.e. $\mathbb{E}[\hat{\mathbf{g}}^{(k)}] = \bar{\mathbf{g}}(\boldsymbol{\theta}^{(k)})$.*

Before we prove the main theorem, we formulate the following basic result.

Lemma 13. *Consider a sequence of allocations $\hat{\boldsymbol{\theta}}^{(k)} \in \mathcal{C}$ such that the sequence $\|\hat{\boldsymbol{\theta}}^{(k)} - \boldsymbol{\theta}^*\|^2$ converges to some $z^* > 0$. Then $\sum_{k=0}^{\infty} [\bar{L}(\hat{\boldsymbol{\theta}}^{(k)}) - \bar{L}(\boldsymbol{\theta}^*)] = +\infty$.*

Proof: By hypothesis, $\hat{\boldsymbol{\theta}}^{(k)}$ is distant from the unique minimizer $\boldsymbol{\theta}^*$, and thus

$$\exists \epsilon > 0 : \|\hat{\boldsymbol{\theta}}^{(k)} - \boldsymbol{\theta}^*\|^2 \geq \epsilon \text{ for } k \text{ sufficiently large.}$$

Let us define $\mathcal{C}' = \{\boldsymbol{\theta} \in \mathcal{C} | \|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|^2 \geq \epsilon\}$, which is closed and bounded. According to the Bolzano-Weierstrass theorem there exists a subsequence $\check{\boldsymbol{\theta}}^{(k)}$ of $\hat{\boldsymbol{\theta}}^{(k)}$ that converges to an allocation $\boldsymbol{\theta}' \in \mathcal{C}'$. By construction, \bar{L} is continuous, whence $\lim_{k \rightarrow \infty} \bar{L}(\check{\boldsymbol{\theta}}^{(k)}) = \bar{L}(\boldsymbol{\theta}')$ and, since \bar{L} has a unique minimizer (Lemma 8), we can define $2\epsilon' = \bar{L}(\boldsymbol{\theta}') - \bar{L}(\boldsymbol{\theta}^*) > 0$. By the definition of limit, for k sufficiently large, $|\bar{L}(\check{\boldsymbol{\theta}}^{(k)}) - \bar{L}(\boldsymbol{\theta}')| < \epsilon'/2$, and thus $\bar{L}(\check{\boldsymbol{\theta}}^{(k)}) - \bar{L}(\boldsymbol{\theta}^*) > \epsilon'/2$. Therefore

$$\sum_{k=0}^{\infty} [\bar{L}(\hat{\boldsymbol{\theta}}^{(k)}) - \bar{L}(\boldsymbol{\theta}^*)] \geq \sum_{k=0}^{\infty} [\bar{L}(\check{\boldsymbol{\theta}}^{(k)}) - \bar{L}(\boldsymbol{\theta}^*)] = +\infty,$$

where, in the first inequality, we use the fact that $[\bar{L}(\check{\boldsymbol{\theta}}^{(k)}) - \bar{L}(\boldsymbol{\theta}^*)]$ is a subsequence of the non-negative sequence $[\bar{L}(\hat{\boldsymbol{\theta}}^{(k)}) - \bar{L}(\boldsymbol{\theta}^*)]$ and the last equality is a well known property of lower-bounded positive series. \blacksquare

We can now prove the main theorem.

Theorem 14. *The sequence $\boldsymbol{\theta}^{(k)}$ generated by SDCP converges in probability to the unique minimizer $\boldsymbol{\theta}^*$ of \bar{L} , i.e., for arbitrary $\delta > 0$*

$$\lim_{k \rightarrow \infty} Pr\{\|\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^*\| > \delta\} = 0.$$

Proof: The proof of convergence is similar to (Theor. 46 in [14]), with the difference that our proof holds for Euclidean projection-based stochastic subgradients. Let us compute

$$\begin{aligned} \|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^*\|^2 &= \|\boldsymbol{\varphi}(\boldsymbol{\theta}^{(k)} - a^{(k)}\hat{\mathbf{g}}^{(k)}) - \boldsymbol{\theta}^*\|^2 \\ &\leq \|\boldsymbol{\theta}^{(k)} - a^{(k)}\hat{\mathbf{g}}^{(k)} - \boldsymbol{\theta}^*\|^2 \\ &= \|\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^*\|^2 - 2a^{(k)} \cdot (\hat{\mathbf{g}}^{(k)})^T \cdot (\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^*) \\ &\quad + (a^{(k)})^2 \cdot \|\hat{\mathbf{g}}^{(k)}\|^2, \end{aligned} \quad (31)$$

where the first inequality is due to Lemma 5. Thanks to Cor. 12 and Def. 9, respectively

$$\begin{aligned} w^{(k)} &\triangleq \left(\mathbb{E}[\hat{\mathbf{g}}^{(k)} | \boldsymbol{\theta}^{(k)}] \right)^T \cdot (\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^*) \\ &= \left(\bar{\mathbf{g}}(\boldsymbol{\theta}^{(k)}) \right)^T \cdot (\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^*) \geq \bar{L}(\boldsymbol{\theta}^{(k)}) - \bar{L}(\boldsymbol{\theta}^*). \end{aligned} \quad (32)$$

Recall that the number of arriving requests per time slot $A(T)$ is bounded, and thus $\|\hat{\mathbf{g}}^{(k)}\|^2$ is bounded, i.e., $\|\hat{\mathbf{g}}^{(k)}\|^2 \leq c$ for some $0 < c < \infty$. Hence, applying the expectation to (31)

$$\mathbb{E} \left[\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^*\|^2 \mid \boldsymbol{\theta}^{(k)} \right] \leq \|\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^*\|^2 + c(a^{(k)})^2. \quad (33)$$

Defining the random variable

$$z_k \triangleq \|\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^*\|^2 + c \sum_{s=k}^{\infty} (a^{(s)})^2,$$

it can be easily verified that (33) is equivalent to the inequality $\mathbb{E}[z_{k+1} | z_k, \dots, z_1] \leq z_k$. Consequently, $\{z_k\}_{k=1}^{\infty}$ is a supermartingale and converges almost surely to a limit z^* . Recalling now that the step size sequence by definition satisfies $\lim_{k \rightarrow \infty} \sum_{s=k}^{\infty} (a^{(s)})^2 = 0$, we have that the sequence $\{\|\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^*\|^2\}$ also converges to z^* with probability one.

What remains is thus to show that $z^* = 0$, which we do by contradiction. If this were not true, then there are $\epsilon > 0$ and $\delta > 0$ such that, with probability $\delta > 0$, $\|\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^*\| \geq \epsilon$ for all sufficiently large k . Thanks to (32) and Lemma 13, $\sum_{k=0}^{\infty} a^{(k)} \cdot w^{(k)} = +\infty$, with probability δ , which, in turn, would imply

$$\mathbb{E} \left[\sum_{k=0}^{\infty} a^{(k)} \cdot \left(\bar{\mathbf{g}}(\boldsymbol{\theta}^{(k)}) \right)^T \cdot (\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^*) \right] = +\infty.$$

However, this would contradict the following relation (which is obtained by a recursion on (31) and then applying the expectation)

$$\begin{aligned} \mathbb{E}[\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^*\|^2] &\leq \\ &\|\boldsymbol{\theta}^{(0)} - \boldsymbol{\theta}^*\|^2 - 2\mathbb{E} \left[\sum_{s=0}^k a^{(s)} \cdot (\hat{\mathbf{g}}^{(s)})^T \cdot (\boldsymbol{\theta}^{(s)} - \boldsymbol{\theta}^*) \right] + \\ &\mathbb{E} \left[\sum_{s=0}^k a^{(s)} \cdot \|\hat{\mathbf{g}}^{(s)}\|^2 \right], \end{aligned}$$

as the left hand side cannot be negative. ■

D. Optimality gap

It is worthwhile to note that the minimizer $\boldsymbol{\theta}^*$ of \bar{L} over $\mathcal{C} \cap \mathbb{R}_{\geq 0}^P$ may not coincide with its minimizer $\boldsymbol{\theta}^{OPT}$ over Θ for two reasons: i) $K' < K$ and ii) $\boldsymbol{\theta}^{OPT}$ is forced to have integer components while $\boldsymbol{\theta}^*$ can be a real vector. In what follows we show that the optimality gap $\|\boldsymbol{\theta}^{OPT} - \boldsymbol{\theta}^*\|_{\infty}$ is bounded by a small number, compared to the number of cache slots available.

Lemma 15. *The gap between the optimal solution $\boldsymbol{\theta}^{OPT}$ and the configuration $\boldsymbol{\theta}^*$ to which SDCP converges is $\|\boldsymbol{\theta}^* - \boldsymbol{\theta}^{OPT}\|_{\infty} \leq (3/2)P$*

Proof: We observe that $\boldsymbol{\theta}^*$ is the optimal solution of the continuous Simple Allocation Problem (SAP), expressed as $\max \left(-\sum_{p=1}^P L_p(\theta_p) \right)$, subject to $\sum_{p=1}^P \theta_p \leq K'$ with $\boldsymbol{\theta} \in \mathbb{R}_{\geq 0}^P$. K' usually referred to as *volume* and we denote with $\text{SAP}_{\text{cont}}(K')$ the problem above. The integer version of the SAP, which we denote by $\text{SAP}_{\text{int}}(K')$, is obtained from the problem above with the additional constraint $\boldsymbol{\theta} \in \mathbb{Z}^P$. According to Cor. 4.3 of [17] there exists a solution $\hat{\boldsymbol{\theta}}$ of $\text{SAP}_{\text{int}}(K')$ such that $\|\boldsymbol{\theta}^* - \hat{\boldsymbol{\theta}}\|_{\infty} \leq P$. The solution of the integer SAP can be constructed via the greedy algorithm presented in Sec. 2 of [17]. In our case, it consists of iteratively adding storage slots, one by one, each time to the CP whose miss intensity is decreased the most by using this additional slot. Based on this, it is easy to verify that a solution $\boldsymbol{\theta}^{OPT}$ can be obtained starting from $\hat{\boldsymbol{\theta}}$ and adding the remaining $K - K'$ slots. Therefore, $\|\boldsymbol{\theta}^{OPT} - \hat{\boldsymbol{\theta}}\|_{\infty} \leq P/2$, which implies

$$\|\boldsymbol{\theta}^{OPT} - \boldsymbol{\theta}^*\|_{\infty} \leq \|\boldsymbol{\theta}^* - \hat{\boldsymbol{\theta}}\|_{\infty} + \|\boldsymbol{\theta}^{OPT} - \hat{\boldsymbol{\theta}}\|_{\infty} \leq (3/2)P. \quad \blacksquare$$

V. IMPLEMENTATION ASPECTS

After proving the convergence of SDCP (which makes the algorithm interesting), we now turn our attention to aspects having practical relevance from implementation and operational standpoints (which makes the algorithm feasible). Notably, we tackle the issue of efficiently computing an Euclidean projection $\boldsymbol{\varphi}$ (Sec. V-A). We next investigate the choice of the step size $a^{(k)}$ on the last line of Alg. 1, that has practical relevance so as the convergence rate of SDCP is concerned (Sec. V-B). Finally, we analyze the space and time complexity of SDCP (Sec. V-C).

A. Euclidean Projection

The efficient implementation of the Euclidean projection $\boldsymbol{\varphi}$ onto a simplex, as defined in (10), has been widely studied in the literature. A convenient implementation is the one shown in Alg. 2, which is a modified version of the algorithm proposed in [12]. Its computational complexity is $O(P \log P)$, which is not expected to be a computational burden since the number of CPs P is expected to be moderate. While we choose Alg. 2 for its simplicity, we point out that algorithms with lower complexity $O(P)$ exist, such as [18], that could be considered shall ever the Euclidean projection $\boldsymbol{\varphi}$ become a system bottleneck. ■

Algorithm 2: Euclidean Projection computation based on [12]

Input : $\theta \in \mathcal{C}, K' > 0$
2 Sort the components of θ in (u_1, \dots, u_P) such that $u_1 \geq \dots \geq u_P$
4 Find $w = \max\{1 \leq p \leq P : u_p + \frac{1}{p} (K' - \sum_{j=1}^p u_j) > 0\}$
6 Define $z = \frac{1}{w} (K' - \sum_{j=1}^w u_j)$
Output: $\varphi(\theta)$ s.t. its p -th component is $\max\{\theta_p + z, 0\}, p = 1, \dots, P$

Algorithm 3: Conditional Step Size Sequence Computation

1 $a = \frac{1}{\|\hat{\mathbf{g}}^{(1)}\|_1} \cdot \frac{K'}{P}$
2 $b = a/10$
3 **if** $k \leq k_{BS}$; // Bootstrap Phase
4 **then**
5 | $a^{(k)} = a$
6 **else if** $k \leq M$; // Adaptive Phase
7 **then**
8 | Compute the miss-ratio $m^{(k)}$ during the current iteration
9 | Compute m_{5th} , i.e. the 5th percentile of the previous miss ratios $m^{(1)}, \dots, m^{(k-1)}$
10 | $\hat{a}^{(k)} = a^{(k-1)}/2$
11 | $\tilde{a}^{(k)} = a^{(k-1)} - \frac{a^{(k-1)} - b}{M - k + 1}$
12 | $a^{(k)} = \begin{cases} (\min(\hat{a}^{(k)}, \tilde{a}^{(k)}), b) & m^{(k)} \leq m_{5th} \\ \tilde{a}^{(k)} & \text{otherwise} \end{cases}$
13 **else**
14 | $a^{(k)} = a^{(k-1)} \cdot \left(1 - \frac{1}{(1+k+M)}\right)^{\frac{1}{2} + \epsilon}$; // Moderate Phase
15 **end**

B. Step Size Sequence

Recall that, in order for SDCP to converge, the step size sequence must satisfy equations (11)-(13). There can be several compliant step size sequences and choosing the right one is critical, as it affects the speed of convergence, as we investigate via simulation in Sec. VI-B. Considering the case of catalog with stationary popularity first, in what follows we propose three step size sequences: namely, *Reciprocal*, *Moderate* and *Conditional*.

In the *Reciprocal* scheme, the step size is

$$a^{(k)} = a/k$$

where

$$a = \frac{1}{\|\hat{\mathbf{g}}^{(1)}\|} \cdot \frac{K'}{P}$$

and $\hat{\mathbf{g}}^{(1)}$ is the update factor (Proposition 11) computed at the first time slot.

Observe that, with this choice, the Euclidean norm of the first update $a^{(1)} \cdot \hat{\mathbf{g}}^{(1)}$ is $\frac{K'}{P}$, which implies a broad exploration of the allocation space at the very beginning.

In the *Moderate* scheme, step sizes decrease slowly, to avoid confining the exploration only to the beginning. Motivated by [19], we define the step size as

$$a^{(k)} = a^{(k-1)} \cdot \left(1 - \frac{1}{(1+M+k)}\right)^{\frac{1}{2} + \epsilon} \quad (34)$$

where a is computed as above and M, ϵ are positive constants that can be used to tune the slope of decrease.

We finally propose in Alg. 3 a third step size sequence, to which we refer as *Conditional*. It consists of a *Bootstrap phase* (up to iteration k_{BS}) in which the step size remains constant, thus allowing broad exploration. Then an *Adaptation phase* follows, up to iteration M , in which the step size decreases, by default, linearly from an initial value a to a final value b . This decrease is steeper than linear when the miss ratio measured at the current iteration is smaller than the 5-th percentile of the miss ratio values observed so far. In this case the step size is halved, unless it already equals b . The intuition behind this phase is that we try to reduce the extent of exploration every time we encounter a “good” allocation, i.e., an allocation that shows a small miss ratio compared to what we experienced so far. Note that we do not start immediately with the Adaptation phase, since we need to collect enough samples during the Bootstrap phase in order to correctly evaluate the quality of the current allocation. Finally, we continue with *Moderate phase*, in which step sizes are updated as in (34) and are asymptotically vanishing, thus guaranteeing convergence.

These step size choices can be extended to the case of non-stationary catalog popularity laws by simply reinitializing the step sequence after a configurable time period (whose settings depends on the timescale of the popularity evolution, which is expected to be quite slow in practice). We consider scenarios with non-stationary catalog in Sec. VI-D.

C. Computational and Space Complexity

The computational complexity of our algorithm depends on the implementation of the projection φ just analyzed and on the step size sequence. Since the algorithm is iterative, in what follows we analyze the complexity per iteration.

Recall that the computation complexity of the projection using Alg. 2 is $O(P \log P)$. As for the step sizes, it is easy to verify that $\hat{\mathbf{g}}^{(1)}$ and K need to be computed only once, with a complexity $O(P)$. Then, the per-iteration computational complexity of *Reciprocal* and *Moderate* is $O(1)$. As for *Conditional*, we have to take into account the computation of the percentile, which is linear with the amount of samples [20]. Since at iteration k there are $k - 1$ measured miss ratio samples, the complexity is $O(k)$. Considering that all the other operations of Alg. 1 are linear in time, the overall complexity at time slot k is

$$\begin{cases} O(P \log P) & \text{for Reciprocal and Moderate} \\ O(P \log P + k) & \text{for Conditional} \end{cases} \quad (35)$$

We now turn to the space complexity. It is easy to show that the space complexity of Alg. 2 is linear in P . The space complexity of the *Reciprocal* and of the *Moderate* step size computation is $O(P)$. In the case of *Conditional* step sizes, we need space for the k miss ratio measurement samples. All the other operations in Alg. 1 require a space that is linear in P . Therefore, the overall space complexity at time slot k is

$$\begin{cases} O(P) & \text{for Reciprocal and Moderate} \\ O(P + k) & \text{for Conditional} \end{cases} \quad (36)$$

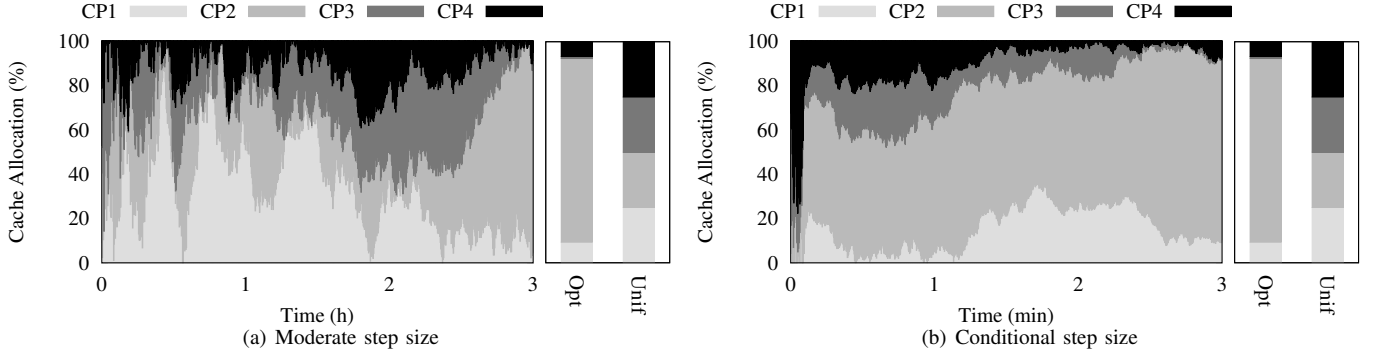


Figure 2. Evolution of the allocation of cache slots across CPs, with cache size $K = 10^5$ and (a) *Moderate* vs (b) *Conditional* step sizes. For reference purposes, bars on the right side of each figure reports the Optimal (Opt) and Uniform (Unif) allocations.

To estimate the overall complexity (i.e., over all iterations until convergence), a further observation is worth sharing. As we just introduced, in cases with dynamic content popularity (see Sec. VI-D), it is reasonable to periodically reinitialize *Conditional* step sizes after a fixed amount of iterations: on the one hand, this makes adaptation to possible changes in the popularity distributions faster; on the other hand, this further limits the complexity as a function of k .

VI. PERFORMANCE EVALUATION

In what follows we evaluate the performance of SDCP through simulations performed in Octave under various patterns of content popularity and we compare its performance to that of reactive caching. We first describe the evaluation scenario (Sec. VI-A) and show how the convergence speed is affected by the choice of the step size sequence (Sec. VI-B). We then evaluate the sensitivity of SDCP to various system parameters (Sec. VI-C). Recognizing that content catalogs are rarely static in the real world, we investigate the expected performance in the case of changing content catalogs (Sec. VI-D). Finally, we confirm SDCP bring consistent gains over reactive caching, both in case of perfect and imperfect knowledge of content popularity (Sec. VI-E).

A. Evaluation Methodology

We consider a content catalog of 10^8 objects, in line with the literature [21] and recent measurements [22]. We partition the catalog in disjoint sub-catalogs, one per each CP. We assume that the content popularity in each sub-catalog follows Zipf's law with exponent $\alpha = 0.8$, as usually done in the literature [23]. We use a cache size of $K \in \{10^4, 10^5, 10^6\}$ objects (which corresponds to cache/catalog ratios of 10^{-4} , 10^{-3} and 10^{-2} respectively), adopting $K = 10^6$ as default value. We set the request arrival rate to $\lambda = 10^2$ req/s, according to recent measurements performed on ISP access networks [22]. We compare the performance of SDCP to that of the optimal allocation θ^{OPT} (Opt), and to that of the naive solution in which the cache space K is equally divided among all the CPs and is unchanged throughout the simulation (Unif).

When using *Conditional* step sizes, after a preliminary evaluation, we set $\epsilon = 1/1000$ as in [19] and $b = a/10$.

We set k_{BS} and M , i.e., the duration of the bootstrap and adaptive phases, to the number of iterations in 6 minutes and 1 hour, respectively. While the optimal duration of these phases would depend on the arrival rate, the performance achieved with these choices was satisfactory for the different scenarios we considered. Unless otherwise stated, the time-slot duration T is set to 10s.

B. Convergence

We start with evaluating the convergence of SDCP, for a cache size of $K = 10^5$ and 4 CPs, receiving 13%, 75%, 2% and 10% of requests, respectively. Fig. 2 shows the evolution of the cache allocation as a function of time, for the *Moderate* (left) and for the *Conditional* (right) step size sequences, and illustrates important properties of the SDCP algorithm. First, after an initial exploration phase the algorithm starts to converge towards a stable allocation. It is important to note that convergence happens in a stochastic sense, i.e., the allocation is a random variable driven by random perturbations (the content request arrival process), but the number of allocated slots per CP converges in the sense of Theor. 14. Third, it is also immediate to notice that the rate of convergence depends on the step size sequence. Interestingly, while convergence is guaranteed only asymptotically, already for short timescales of practical relevance, such as those reported in the figure, the allocation vectors for *Moderate* and *Conditional* steps are close to the optimal one. In particular, the most relevant metric from an ISP viewpoint is represented by the miss-rate, which directly relates to the amount of backhauling bandwidth that a specific allocation incurs: at the end of the 3 hours observation window in Fig. 2, the *Moderate* and *Conditional* miss rates are respectively only 3% and 2% far from the optimal one.

To better assess the impact of the step size sequence, we next consider a larger scenario with cache size $K = 10^6$ and $p = 10$ CPs, one of which is a popular CP, to which 70% of requests are directed, followed by a second one receiving 24% of requests, another 6 CPs accounting for 1% each, and the remaining two CPs receiving no requests. Fig. 3 shows the step sizes and the inaccuracy of the algorithm, i.e. the distance

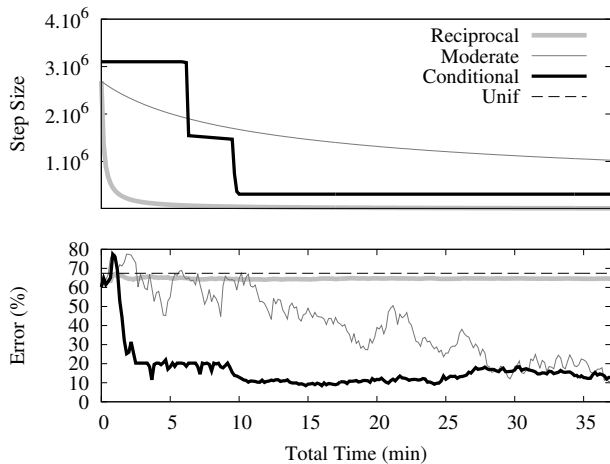


Figure 3. Error and step size sequence with cache size $K = 10^6$.

to the optimal allocation, measured as:

$$Error(\theta) \triangleq \frac{\|\theta^{OPT} - \theta\|_{\infty}}{K} = \frac{\max_{j=1\dots p} |\theta_j - \theta_j^{OPT}|}{K}. \quad (37)$$

We can observe that the step size under *Reciprocal* decreases too fast, and immediately limits the adaptation of the allocation, significantly slowing down convergence. Conversely, *Moderate* results in step sizes remaining large for a long time, preventing the algorithm from keeping the allocation in regions that guarantee good performance. *Conditional* shows the best performance since in the Adaptation phase the step sizes are sharply decreased whenever the current allocation is providing a small miss ratio.

C. Sensitivity Analysis

We next study how the performance of SDCP is affected by the algorithm parameters and the scenario. We first focus on the time slot duration T . On the one hand, a small T implies that only few requests are observed in each time slot, which may result in a high noise $+\varepsilon^{(k)}, -\varepsilon^{(k)}$, and ultimately affects the accuracy of the update. On the other hand, a large T decreases the measurement noise, but makes the allocation updates less frequent, which possibly slows down convergence.

To evaluate the impact of T , Fig. 4 shows the miss ratio measured over 1h for the default scenario. We consider SDCP with the three step size sequences of Sec. V-B, and compare it to the Uniform and to the Optimal allocations as benchmarks. The figure shows that SDCP with the Conditional step size sequence enhances the cache efficiency significantly. We also observe that a time slot duration of $T = 10s$ (corresponding to 100 samples on average per CP) represents a good compromise between a more accurate miss ratio estimation based on more samples (with large T) and a larger number of iterations at the cost of lower accuracy (with small T).

Since Conditional step sizes guarantees the best performance, we will use it in the sequel.

Fig. 5 shows the cache miss rate measured over 1h for a time slot length of $T = 10s$ and for various cache sizes

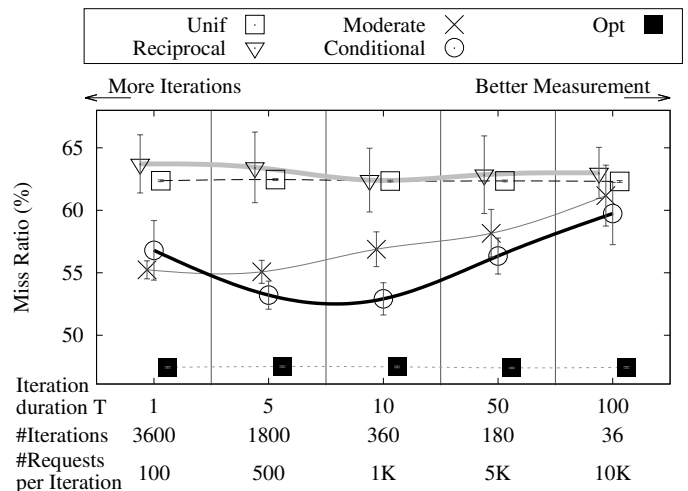


Figure 4. Impact of time-slot duration T on the average miss ratio (bars represent the 95% confidence interval over 20 runs).

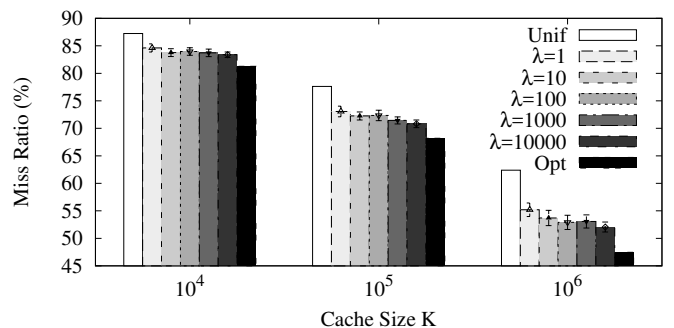


Figure 5. Miss rate measured over 1h for various average request rates λ and cache sizes K . Note that the miss ratio of Unif and Opt does not depend on the request rate. Bars represent the 95% confidence interval over 20 runs.

$K \in \{10^4, 10^5, 10^6\}$ and arrival rates $\lambda \in [1, 10^4]$. The figure confirms that the gains of SDCP hold for different cache sizes, and shows that the gain increases for large caches. To interpret the results for different arrival rates, recall that for any given time slot duration T , the average request rate affects the measurement noise. Fig. 5 confirms that the miss ratio increases when the measurement noise is higher, i.e., for lower λ , but it also shows a very limited impact: the number of time slots in a relatively short time (in 1h, there are 360 time slots of duration $T = 10s$) allows SDCP to converge to a good cache configuration, in spite of the noise and the consequent estimation errors.

D. Changing Content Popularity

Motivated by recent studies that showed that content popularities change over time [24], we next investigate the performance of SDCP under time-varying content popularity.

For each object we generate requests using a discrete-time interrupted Poisson process (IPP), i.e., a discrete-time two state Markov modulated Poisson process. At each time slot an object can be either in ON state or OFF state. Requests for an object are only generated during a time slot if the object is in the ON state. We denote by $p_{ON \rightarrow OFF}$ and

$p_{OFF \rightarrow ON}$ the transition probability from ON to OFF and from OFF to ON states, respectively. It is easy to show (Sec. 2.2 of [25] and page 201 of [26]) that the number of time slots N_{ON} (N_{OFF}) an object stays in the ON (OFF) state is a random variable with geometric distribution and mean $1/p_{ON \rightarrow OFF}$ ($1/p_{OFF \rightarrow ON}$). We note that this model is a discrete time version of the one in [24], in which objects alternate between ON and OFF states, whose duration is exponentially distributed. As in [24], we set the catalog size to $3.5 \cdot 10^6$, the cache size to $K = 10^4$ objects and $\mathbb{E}[N_{ON}]/\mathbb{E}[N_{OFF}] = 1/9$. Observe that a smaller $\mathbb{E}[N_{ON}]$ value means that the content popularity varies more frequently. To provide a conservative evaluation, we consider a pessimistic case of a very fast dynamic, setting $\mathbb{E}[T_{ON}] = 1$ day, below the fastest dynamic reported in the literature (Tab. I of [24]). On average, we maintain the overall request rate equal to our default value $\lambda = 100req/s$, i.e., an object i in ON state receives requests at rate $\lambda \cdot q_i \cdot \frac{\mathbb{E}[T_{ON}] + \mathbb{E}[T_{OFF}]}{\mathbb{E}[T_{ON}]}$, where q_i is the popularity of i , computed using Zipf's law.

Fig. 6 shows the performance of *Unif*, which statically partitions the cache in slices of equal size, and of *Conditional*(τ), which reinitializes the step sequence every τ amount of time ($\tau \in \{3h, 1d, \infty\}$), i.e., 8 times per day, once per day, and never, respectively, and, within each period of time τ updates the step sequence according to Alg. 3. The results show that reinitialization helps improve cache efficiency under changing content popularities. Indeed, the evolution of the content popularity is detrimental for *Conditional*(∞) and *Conditional*(1d) already after a few hours, causing them to have performance worse than *Unif*. To understand this behavior, we can interpret the sequence of allocations computed by the algorithm as a trajectory in the P -dimensional space, over time. If the popularity does not change, the optimal allocation is always the same point and the trajectory of *Conditional*(∞) approaches this point by decreasing the extent of the variation, thanks to the decreasing step size sequence. On the contrary, if popularity changes, the optimal allocation changes over time. Therefore, if we keep decreasing the step size sequence, as in *Conditional*(∞), the trajectory cannot follow the optimal point fast enough. Nonetheless, reinitializing the step sequences as in *Conditional*(3h) is sufficient to respond to fast catalog dynamics as the results show. Furthermore, notice that the timescale of the reinitialization of the step sequence leading to “good enough” results in the non-stationary case, happens to be of the same order of magnitude of the timescale where SDCP was already yielding to “good enough” results in the stationary case (recall Fig. 2).

Clearly, it would be interesting to further test our algorithm under real traffic traces, i.e., with simple trace-driven simulation. Sadly, publicly available datasets [27] are not fit for our purpose, and thus the scientific community as a whole lacks access to real datasets. Yet, we point out that this simple strategy is expected to give even better performance in realistic situations, since dynamics are slower [24] than the ones considered here.

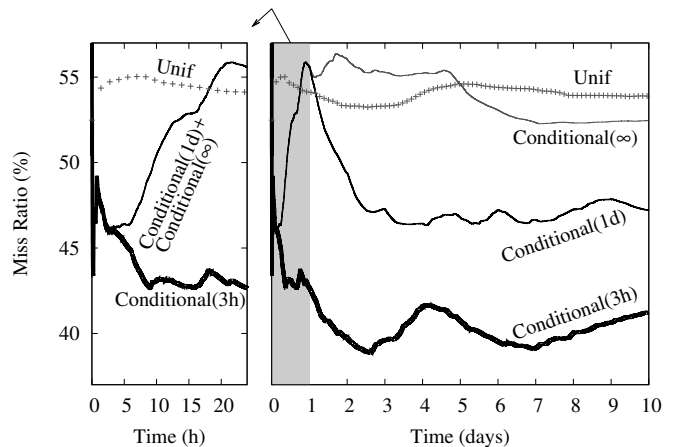


Figure 6. Miss ratio with varying content popularity. Step size sequences in *Conditional*(τ) are reset every τ interval. On the left the first day is zoomed, where *Conditional*(∞) and *Conditional*(1d) correspond, since none of them have reinitialized the step sequence.

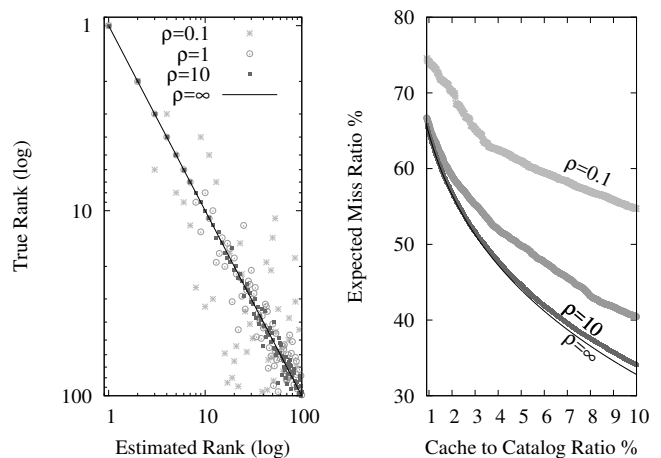


Figure 7. Error due to rank mis-estimation, for an experiment with a catalog of 5000 objects. In the left figure, we plot the true rank corresponding to each estimated rank. It means that for each i -th object in the x axis (which is the object that is estimated to be the i -th most popular object), we plot what is its true rank. In the right plot, we depict the expected miss ratio when storing in cache the first estimated most popular objects instead of the first true most popular object.

E. SDCP vs. Reactive Caching

Finally, we compare SDCP to reactive caching. To allow for a fair comparison, we have to consider that in practice the CPs may have a noisy prediction of the content popularities, and thus they would inevitably cache objects in the allocated storage that are not among the most popular objects. We model the popularity prediction of CP p by the empirical frequency of object requests in a sequence of $\rho \cdot |\mathcal{O}_p|$ requests for the CP's objects, where $|\mathcal{O}_p|$ is the cardinality of the catalog of the CP p and $\rho \in [0, +\infty[$ is a parameter that we call the *estimation accuracy*.

Fig. 7 (left) shows the error in the rank estimation of a CP with different values of ρ for an example run. Clearly, increasing ρ increases the accuracy of the rank estimation and for $\rho = \infty$ the estimate is noiseless. It is important to note that

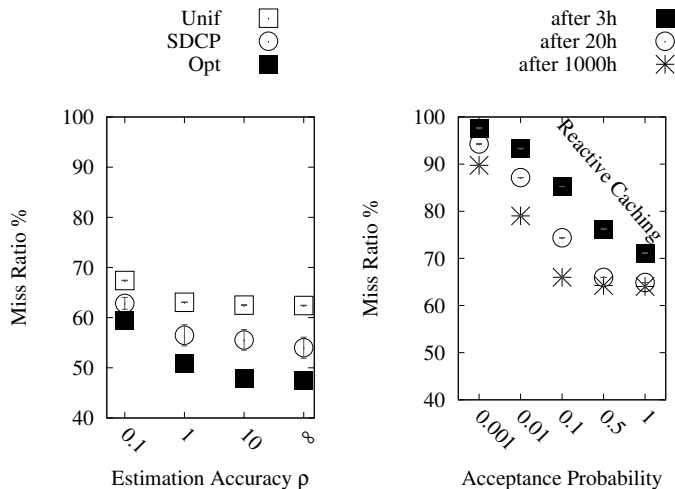


Figure 8. Miss ratio obtained with SDCP (left) and reactive caching (right). The average values over 20 runs are shown, as well as 95% confidence intervals (when visible). Default scenario is assumed. In the left plot, the optimal allocation is computed based on the true popularity distributions, even if, during the simulation, the objects that are saved in each cache slice are selected based on the estimated rank.

a rank mis-estimation is more likely for less popular objects, as the ratio between the popularity of objects i and $i+1$ tends to 1 for high i , and thus the popularity of objects in the tail is practically indistinguishable. This can be observed in the bottom-right corner of Fig. 7 (left).

The mis-estimation of the rank naturally increases the miss ratio that a CP can achieve with its allocated cache space. Fig. 7 (right) shows the cache miss rate for the same values of ρ as Fig. 7 (left) and shows that the cache miss rate increases significantly, especially for high values of the cache/catalog ratio, when less popular objects have a chance to be cached. Note that rank mis-estimation may cause that the miss rate becomes a non-convex function of the cache storage, but if the estimator is unbiased then the expected miss ratio is a convex function.

As an alternative to SDCP, we consider a reactive strategy called *Probabilistic Caching*. Under this strategy when a cache miss occurs the newly fetched object is inserted into the cache with a certain probability, referred to as the *admission probability* [28]. If the object is inserted then the least recently used object is evicted. It was recently shown that Probabilistic caching is asymptotically optimal as the admission probability goes to zero [28]. We use ccnSim [29], an event driven simulator, to evaluate the reactive strategy. Observe that we consider reactive caching for the sake of comparison, but in reality it would be impossible to implement such a strategy in the scenario we consider, as it assumes that the ISP can observe the objects transmitted.

Fig. 8 (left) shows the miss rate as a function of the estimation accuracy obtained with SDCP and, as a comparison, with Unif and the theoretically optimal allocation, after 3h of simulations. Fig. 8 (right) shows the corresponding miss rate for reactive caching as a function of the admission probability, evaluated over a period of 3h, 20h and 100h. The results show that SDCP is superior to reactive caching,

even conservatively considering the worst cases for SDCP (low estimation accuracy and short time of 3h to improve the performance) and the best case for reactive caching (best admission probability and long experiment time).

To get an insight into the reason for the superior performance of SDCP, Fig. 9 shows the number of objects stored in cache that do not belong to the first K most popular objects, i.e., objects that would ideally not be cached, in absence of rank mis-estimation. We refer to these objects as *trash* and to the other objects as *good*. While SDCP is able to select good objects and reduce the trash after 3h, in the case of reactive caching the cache is filled mostly with trash after 3h. A low acceptance probability effectively decreases the trash, but since it lets the cache content change very slowly, even though newly inserted objects are likely to be popular ones, it takes too much time for them to enter the cache.

VII. RELATED WORK

Recent years have seen an increasing interest in cooperative cache management (Sec. VII-A) and in cache partitioning (Sec. VII-B), both of which are related to our work.

A. Cooperative Cache Management

The literature on cooperative cache management focuses on ISP/CDN/CP cooperation. These works show that ISPs have a strong incentive in investing in caching to reduce the traffic on their critical paths, but also show that CPs and users would also benefit from ISP in-network caching. The game theoretical study in [30] shows that caches are inefficient when operated by CPs, since CP content placement and ISP traffic engineering are often not compatible. Solutions are proposed in [31], [32], which however require ISPs to share with the CP confidential information, such as topology, routing policies or link states, and as such are arguably practical, since ISPs are typically not willing to disclose information about the topology and the state of their network, i.e., congestion, available bandwidth, etc. [33]. Conversely, [34], [33], [35] foster an ISP-operated cache system, but requires the ISP to be able to observe every object requested by the users, which is arguably equally impractical since CPs purposely hide this confidential information, and makes these approaches incompatible with encryption. In contrast with these previous works, our solution allows for encryption and is not limited to a single CP, unlike [31].

At the time of writing there is not an established technological standard yet, however there are at least two important industry fora our work is aligned with. The mission of the Open Caching Working Group (OCWG) [36] is to develop standards, policies and best practices for a new layer of content caching within ISP networks, which can coexist with HTTPS and provide shared access to storage for many CPs.

Another relevant normalization forum is represented by the ETSI Industry Specification Group (ISG) on Mobile Edge Computing (MEC)[7], whose purpose is to create a standardized, open environment, which will allow the efficient and seamless integration of applications from vendors, service providers, and third-parties across multi-vendor Mobile-edge

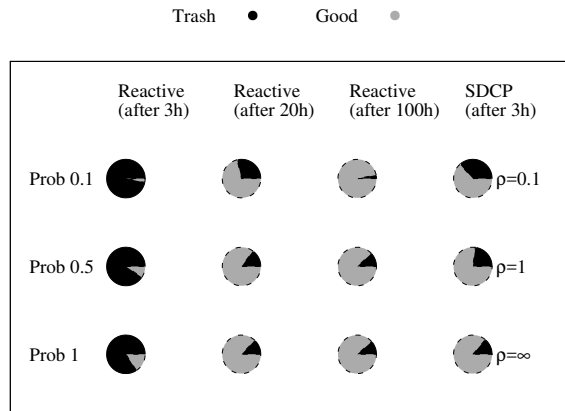


Figure 9. Percentage of trash and unused cache space. Every 10 seconds the number of trash objects is measured. The average over all these measurement is depicted. Default scenario is assumed.

Computing platforms. Recently, MEC entered its phase 2 with the goal to offering standard APIs [9], and interfaces between MEC hosts and transport networks, which will enable the cooperation we envision in this paper. Quoting the ETSI MEC whitepaper, “the aim is to reduce latency, ensure highly efficient network operation and service delivery, and offer an improved user experience” [7]: clearly, since *caching* reduces latency, it is one of the use cases considered by the ETSI MEC. 5G Americas goes even further and identifies caching and Information Centric Networking as a crucial technology to enable in 5G networks, at the mobile edge and beyond [8], reinforcing the soundness of our assumptions.

Our work fits the OCWG requirements as well as the ETSI MEC, and as such is, we believe, of high practical relevance.

B. Cache Partitioning

Cache partitioning has a long history in CPU cache optimization. One example is the stochastic optimization based algorithm of [37], which partitions a CPU cache among competing processes. Clearly, the cache workload created by CPU jobs, the cache size and all the other characteristics of the CPU cache scenario are fundamentally different from the in-network cache case we address. The literature on in-network content caching usually treats the cache as a whole, with the following recent exceptions [38], [39], [40], [41], [42], [43]. In particular, [38], [39] assign slices to different services, to favor the ones with more stringent requirements. In [40], each slice is used for a different video quality. Closer to our work, are [41] and [42] that consider a cache owner (corresponding to the ISP in our case) sharing storage among different CPs.

In particular, [41] proposes a pricing scheme based on the value that each CP gives to cache space, which on the one hand requires settling, on the other hand it may or may not lead to the minimization of the cache miss rate. Unlike [41], our solution allows to minimize the cache miss rate without the need for payments, which may make its adoption less controversial considering the disputes among ISPs and CPs in recent years [33].

Authors of [42] show in an optimization framework that partitioning a cache among multiple CPs is more efficient than

using a single LRU cache. Unlike our work, [42] assumes that the cache owner knows the content popularity and uses Che’s approximation [44] for formulating an optimization problem, which requires CPs to reveal business critical information periodically. This assumption implies an important difference in terms of methodology, since in our framework learning of the cache miss rate as a function of the storage size happens simultaneously to finding the optimal allocation, whereas the optimization framework in [42] assumes the hit rate is a known function of the storage size. Finally, [42] assumes that each CP manages its cache slice using LRU, which does not allow to exploit the a-priori knowledge that CPs may have about their content popularity. Our work overcomes these limitations, and extends [43] by discussing implementation aspects of the proposed SDCP algorithm, and by analyzing the sensitivity of the algorithm and its adaptation to non-stationary workloads. To the best of our knowledge, our work is the first to consider the dynamic partitioning of in-network caches that allows CPs to manage their cache content, with the aim of minimizing the overall ISP cache miss rate, without disclosing nor leaking content-related information to the ISP at the same time.

VIII. CONCLUSION

One of the main challenges of in-network caching nowadays is its incompatibility with encrypted content. Our work represents a first step in solving this challenge by proposing a simple and therefore appealing system design: Stochastic Dynamic Cache Partitioning requires solely the knowledge of aggregated cache miss-intensities, based on which it provably converges to an allocation with a small optimality gap. SDCP assigns cache slots to CPs and allows CPs choose what to cache. Since CPs may be able to predict content popularity, they can take informed caching decisions that would be impossible in classic caching schemes, in which the cache owner is a third party with little knowledge of content popularity. Therefore, not only does SDCP make cache sharing feasible in case of encrypted content, but it can also has the potential to make it more efficient than traditional reactive caching policies, as our comparison shows. Simulation results confirm the benefits of the proposed algorithm to hold under various scenarios, which includes scenarios where the CPs has non-perfect estimates of the object popularity. Additionally, numerical results confirm that performance maintains also in scenarios with complex content catalog dynamics, that thus make SDCP applicable in scenarios of high practical relevance.

ACKNOWLEDGEMENTS

This work benefited from support of the Swedish Foundation for Strategic Research through the Modane project and of NewNet@Paris, Cisco’s Chair “NETWORKS FOR THE FUTURE” at Telecom ParisTech (<http://newnet.telecom-paristech.fr>). Any opinion, findings or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of partners of the Chair.

REFERENCES

- [1] S. Deering, “Watching the Waist of the Protocol Hourglass,” in *IETF51 Plenary Talk*, 2001.

- [2] D. Thaler, "Evolution of the IP Model," in *IETF73 Plenary Talk*, 2008.
- [3] I. Popa, Lucian and Ghodsi, Ali and Stoica, "HTTP as the Narrow Waist of the Future Internet," in *ACM SIGCOMM HotNets Workshop*, 2010.
- [4] D. Naylor, A. Finamore, I. Leontiadis, Y. Grunenberger, M. Mellia, M. Munafo, K. Papagiannaki, and P. Steenkiste, "The Cost of the "S" in HTTPS," in *ACM CoNEXT*, 2014.
- [5] "IAB Statement on Internet Confidentiality," IETF, Tech. Rep., 2014.
- [6] G. Barish and K. Obraczke, "World Wide Web caching: trends and techniques," *IEEE Commun. Mag.*, vol. 38, no. 5, pp. 178–184, 2000.
- [7] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing – a key technology towards 5g," *ETSI White Paper*, no. 11, pp. 1–16, 2016.
- [8] 5G Americas, "Understanding information centric networking and mobile edge computing," http://www.5gamericas.org/files/3414/8173/2353/Understanding_Information_Centric_Networking_and_Mobile_Edge_Computing.pdf, 2016.
- [9] www.etsi.org/news-events/news/1180-2017-03-news-etsi-multi-access-edge-computing-starts-second-phase-and-renews-leadership-team.
- [10] Ü. Yüceer, "Discrete convexity: convexity for functions defined on discrete spaces," *Discrete Applied Mathematics*, vol. 119, no. 3, pp. 297–304, 2002.
- [11] J. Carnicer and W. Dahmen, "Characterization of Local Strict Convexity Preserving Interpolation Methods by C1 Functions," *Journal of Approximation Theory*, vol. 77, no. 1, pp. 2–30, 1994.
- [12] W. Wang and M. A. Carreira-Pepinan, *Projection onto the probability simplex : An efficient algorithm with a simple proof and an application*. arXiv:1309.1541v1, 2013.
- [13] H. H. Bauschke and J. M. Borwein, "On Projection Algorithms for Solving Convex Feasibility Problems," *SIAM Review*, vol. 38, no. 3, pp. 367–426, 1996.
- [14] N. Z. Shor, *Nondifferentiable Optimization and Polynomial Problems*. Springer, 1998.
- [15] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [16] C. P. Niculescu and L.-E. Persson, *Convex Functions and Their Applications: A Contemporary Approach*. Springer, 2004.
- [17] D. S. Hochbaum, "Lower and Upper Bounds for the Allocation Problem and Other Nonlinear Optimization Problems," *Mathematics of Operation Research*, vol. 19, no. 2, pp. 390–409, 1994.
- [18] N. Maculan and G. G. J. de Paula, "A linear-time median-finding algorithm for projecting a vector on the simplex of R^n ," *Operations Research Letters*, vol. 8, no. 4, pp. 219–222, 1989.
- [19] Q. Wang, "Optimization with Discrete Simultaneous Perturbation Stochastic Approximation Using Noisy Loss Function Measurement," Ph.D. dissertation, John Hopkins University, 2013.
- [20] M. Blum, R. W. Floyd, V. Pratt, R. L. Rivest, and R. E. Tarjan, "Time bounds for selection," *Journal of Computer and System Sciences*, vol. 7, no. 4, pp. 448–461, 1973.
- [21] C. Fricker, P. Robert, J. Roberts, and N. Sbihi, "Impact of Traffic Mix on Caching Performance in a Content-Centric Network," in *IEEE INFOCOM*, 2012.
- [22] C. Imbrenda, L. Muscariello, and D. Rossi, "Analyzing Cacheable Traffic in ISP Access Networks for Micro CDN Applications via Content-Centric Networking," in *ACM SIGCOMM ICN*, 2014.
- [23] K. Poularakis, G. Iosifidis, and L. Tassiulas, "Joint Caching and Base Station Activation for Green Heterogeneous Cellular Networks," in *IEEE ICC*, 2015.
- [24] M. Garetto, E. Leonardi, and S. Traverso, "Efficient analysis of caching strategies under dynamic content popularity," *IEEE INFOCOM*, 2015.
- [25] T. Bonald and M. Feuillet, "Exponential Distribution," in *Network Performance Analysis*. Wiley, 2011.
- [26] W. J. Stewart, *Probability, Markov Chains, Queues and Simulation*, 1st ed. Princeton University Press, 2009.
- [27] W. Bellante, R. Vilaridi, and D. Rossi, "On netflix catalog dynamics and caching performance," in *IEEE CAMAD*, 2013.
- [28] V. Martina, M. Garetto, and E. Leonardi, "A unified approach to the performance analysis of caching systems," in *IEEE INFOCOM*, 2014.
- [29] R. Chiocchetti, D. Rossi, and G. Rossini, "ccnSim: a Highly Scalable CCN Simulator," in *IEEE ICC*, 2013.
- [30] D. DiPalantino and R. Johari, "Traffic Engineering vs. Content Distribution: A Game Theoretic Perspective," in *IEEE INFOCOM*, 2009.
- [31] W. Jiang, R. Zhang-Shen, J. Rexford, and M. Chiang, "Cooperative content distribution and traffic engineering in an ISP network," *ACM SIGMETRICS Perf. Eval. Review*, vol. 37, no. 1, pp. 239–250, 2009.
- [32] B. Frank, I. Poesche, Y. Lin, G. Smaragdakis, A. Feldmann, B. M. Maggs, J. Rake, S. Uhlig, and R. Weber, "Pushing CDN-ISP Collaboration to the Limit," *ACM SIGCOMM CCR*, vol. 43, no. 3, pp. 35–44, 2013.
- [33] A. Gravey, F. Guillemin, and S. Moteau, "Last Mile Caching of Video Content by an ISP," in *European Teletraffic Seminar*, 2013.
- [34] K. Cho, H. Jung, M. Lee, D. Ko, T. T. Kwon, and Y. Choi, "How Can an ISP Merge with a CDN?" *IEEE Commun. Mag.*, vol. 49, no. 10, pp. 156–162, 2011.
- [35] G. Dan and N. Carlsson, "Dynamic content allocation for cloud-assisted service of periodic workloads," in *IEEE INFOCOM*, 2014.
- [36] "Open Caching: Problem Statement and Guiding Principles," Streaming Video Alliance, Tech. Rep., 2015.
- [37] I. Megory-Cohen and G. Ela, *Dynamic Cache Partitioning by Modified Steepest Descent*. U.S. Patent No. 5,357,623, 1994.
- [38] Y. Lu, T. F. Abdelzaher, and A. Saxena, "Design, implementation, and evaluation of differentiated caching services," *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, no. 5, pp. 440–452, 2004.
- [39] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino, "Evaluating per-application storage management in content-centric networks," *Comput. Comm.*, vol. 36, no. 7, pp. 750–757, 2013.
- [40] A. Araldo, F. Martignon, and D. Rossi, "Representation Selection Problem : Optimizing Video Delivery through Caching," in *IFIP Netw.*, 2016.
- [41] S. Hoteit, M. E. Chamie, D. Saucez, and S. Secci, "On Fair Network Cache Allocation to Content Providers," *Comput. Netw.*, 2016.
- [42] W. Chu, M. Dehghan, D. Towsley, and Z.-l. Zhang, "On Allocating Cache Resources to Content Providers," in *ACM SIGCOMM ICN*, 2016.
- [43] A. Araldo, G. Dan, and D. Rossi, "Stochastic Dynamic Cache Partitioning for Encrypted Content Delivery," in *ITC*, 2016.
- [44] H. Che, Y. Tung, and Z. Wang, "Hierarchical Web caching systems: modeling, design and experimental results," *IEEE JSAC*, vol. 20, no. 7, pp. 1305–1314, sep 2002.



Andrea Araldo (S'14) is a postdoc at MIT. He received his MSc in Computer Science from Università di Catania in 2012 and his PhD in Computer Networks from Univ. ParisSud and Télécom ParisTech in 2016. He was a visiting researcher at KTH in 2016 and worked in FP7 European research projects, such as Ofelia and mPlane and for the Italian academic consortium CNIT in 2012-13. His interests include Network Optimization, Content Distribution in the Internet, Intelligent Transportation systems.



György Dán (SM'17) (M.Sc. '99 and '03, PhD '06) is a Professor at KTH Royal Institute of Technology, Stockholm, Sweden. He was a visiting researcher at the Swedish Institute of Computer Science in 2008, a Fulbright research scholar at University of Illinois at Urbana-Champaign in 2012-2013, and an invited professor at EPFL in 2014-2015. He has been associate editor of Computer Communications since 2014, and TPC members of some 40 conferences including ACM ICN, ACM e-Energy and IEEE Infocom (Distinguished member 2015,2017). He has coauthored over 100 papers receiving 4 best paper awards.



Dario Rossi (SM'13) is a Professor at Telecom ParisTech and Ecole Polytechnique, and is the holder of Cisco's Chair NewNet@Paris. He served on the board of several IEEE Transactions, and in the program committees of over 50 conferences including ACM ICN, ACM CoNEXT, ACM SIGCOMM and IEEE INFOCOM (Distinguished Member 2015, 2016 and 2017). He has coauthored 9 patents and over 150 papers, receiving 4 best paper awards, a Google Faculty Research Award (2015) and an IRTF Applied Network Research Prize (2016).