Noname manuscript No.
(will be inserted by the editor)

Distributed Caching over Heterogeneous Mobile Networks

Stratis Ioannidis · Laurent Massoulié · Augustin Chaintreau

Received: date / Accepted: date

Abstract Sharing content over a mobile network through opportunistic contacts has recently received considerable attention. In proposed scenarios, users store content they download in a local cache and share it with other users they meet, *e.g.*, via Bluetooth or WiFi. The storage capacity of mobile devices is typically limited; therefore, identifying which content a user should store in her cache is a fundamental problem in the operation of any such content distribution system.

In this work, we propose Psephos, a novel mechanism for determining the caching policy of each mobile user. Psephos is fully distributed: users compute their own policies individually, in the absence of a central authority. Moreover, it is designed for a heterogeneous environment, in which demand for content, access to resources, and mobility characteristics may vary across different users. Most importantly, the caching policies computed by our mechanism are *optimal*: we show that Psephos maximizes the system's *social welfare*. To the best of our knowledge, our work is the first to address caching with heterogeneity in a fully distributed manner.

Keywords distributed caching · heterogeneity · mobile networks · content distribution

S. Ioannidis
Technicolor
Palo Alto, CA, USA
E-mail: stratis.ioannidis@technicolor.com

L. Massoulié
Technicolor
Paris, France
E-mail: laurent.massoulie@technicolor.com

A. Chaintreau
Columbia University
New York, NY, USA
E-mail: augustin@cs.columbia.edu

An earlier version of this work appeared at the ACM Sigmetrics 2010 conference.

1 Introduction

In this work, we consider a peer-to-peer content sharing system deployed over a network of mobile devices. The users of these devices download content from the Internet whenever they have access to a dedicated service infrastructure—*e.g.*, a wireless connection at their home or office environment. The downloaded content is subsequently stored and shared among users in an opportunistic fashion: the mobile users exchange their stored content, *e.g.*, via Bluetooth, whenever they meet.

Such content sharing systems have received considerable attention recently [1, 2, 5, 9, 13]. Their appeal can be attributed to two fundamental properties. First, through sharing, users gain access to content even when they are not within the infrastructure's coverage. Second, by utilizing the bandwidth available during opportunistic contacts, sharing can assist the distribution of content and reduce the overall load on the infrastructure. The above systems therefore extend the reach of the dedicated service infrastructure while also improving its scalability. As such, they are well suited for scenarios where access to the infrastructure is intermittent, as well as when the downlink bandwidth available for content distribution is limited.

Given that mobile devices typically have limited storage capacity, an important design challenge in the above content sharing systems is determining a user's *caching policy*. In short, a user's caching policy characterizes what content she should retrieve and store when accessing the infrastructure. From a system-wide perspective, a natural goal when selecting such policies is to minimize delay: the content stored at different users should be chosen so that, *e.g.*, the average delay for retrieving requested content is minimized.

Addressing this problem in the context of a mobile system is challenging for a variety of reasons. The most important one is heterogeneity. First, system resources may vary across users: any two users may have different storage capacities and access the infrastructure at a different rate. Second, users may not value content the same way: they may not necessarily be interested in the same content, and might be more or less sensitive to finding it quickly. Finally, they might follow diverse mobility patterns, thereby having different opportunities to share their cached content and to retrieve content from other users. All of the above three sources of heterogeneity (access to resources, user preferences and mobility) play an important role in determining what content a user should store, and imply that caching policies aiming to minimize delays may vary considerably among different users.

An additional challenge arises from the fact that user preferences and mobility may not be a priori known: no mechanism for determining a user's caching policy can readily access this information. As a result, determining caching policies in a centralized manner requires collecting data from users, such as mobility or content access traces. Such a data collection process may not scale, giving rise to the need for decentralization. Mobile users may also wish to share downloaded content while avoiding the use of any central authority—*e.g.*, due to "single point of failure" or privacy concerns. For

these reasons, distributed mechanisms for caching policy selection are of great interest.

The main contribution of this paper is to address the above issues by proposing PSEPHOS[1], a novel distributed mechanism for determining optimal user caching policies. In particular:

- PSEPHOS is designed for an environment in which *all of the above three sources* of heterogeneity (access to resources, user preferences and mobility) can occur. It is adaptive: it does not require a priori knowledge of user demands or mobility patterns, but adapts to them while constructing the caching policies. PSEPHOS is also distributed: each user computes its caching policy individually, by exchanging messages with other users she meets.
- PSEPHOS is simple and easy to implement. In short, users maintain a "vote" for each possible item they can store. These votes are computed using only local information, and reflect the amount of requests for content a user receives, as well as how much other users value this content. Determining which content to store is then straightforward: whenever a user accesses the infrastructure, she sorts these votes and fills her cache with the content items that have the top votes.
- Surprisingly, in spite of the inherent simplicity of the above voting scheme, the caching policies computed by PSEPHOS are *optimal*. We show that the caching policies selected by our mechanism maximize the aggregate utility (*i.e.*, the *social welfare*) of users participating in the system.

Our work is the first to address caching with heterogeneity in a fully distributed manner. Moreover, to the best of our knowledge, we are the first to propose the aforementioned voting scheme and rigorously show it converges to the solution of a convex optimization problem (here, maximizing the system's social welfare). This result can potentially be applied in solving convex optimization problems arising in different contexts than the one considered in this work.

The remainder of this paper is organized as follows. In Section 2, we overview related work in the area of content sharing in mobile networks. In Section 3, we give a detailed description of our system as well as our mechanism for determining caching policies. In Section 4 we present our main results, without proofs, and discuss the intuition behind them as well as their implications. The full analysis, and derivation of the above results, can be found in Sections 5 and 6. We discuss possible extensions of our methods and results in Section 7, and conclude in Section 8.

2 Related Work

Sharing content using opportunistic contacts between mobile devices has been recently proposed within several different contexts, including website down-

[1] From ψήφος, the Greek word for "pebble". Pebbles were used in ancient Greece as ballots during elections.

loading (7DS project [12]), the dissemination of podcasts (Podnet project [11]) and newsfeeds [1, 5, 9], publish-subscribe systems [6, 15], and file sharing [13]. With the proliferation of powerful mobile devices (*e.g.*, smartphones), the potential applications for such content sharing architectures are likely to grow.

The question of optimizing caching policies to minimize delay in a mobile network was first introduced in [13]; the present paper builds upon and extends this earlier work. The authors introduce utilities that are functions of the time required to retrieve a given item, thereby capturing user "impatience" [13]. They then seek item replication ratios that maximize the social welfare, a formulation and objective we also adopt here. They further prove that this problem is convex and can be solved by gradient descent, a result we reuse (see Lemma 1) and exploit to design Psephos.

A distributed replication mechanism that yields optimal replication ratios is proposed in [13] under the assumption of homogeneity: mobile users meet each other with the same rate, have identical storage capacities and demands for content and exhibit the same impatience towards delays. The mechanism works as follows: whenever a user retrieves an item, she pro-actively replicates this item throughout the system. The number of replicas generated is computed in terms of the time required to retrieve the item and the unique utility/impatience function, which is a priori known. This mechanism is shown to be optimal in equilibrium; however, convergence to an equilibrium point is not established.

Our work extends [13] by proposing a novel mechanism that deals with heterogeneity: we consider users having different contact rates, storage capacities, demands and utility functions. The mechanism proposed in [13] does not apply to this case, as it crucially relies on homogeneity and the a priori knowledge of the (unique) utility function. In contrast, Psephos uses only local information and operates under heterogeneity of all the aforementioned parameters. Moreover, we establish formal conditions guaranteeing the convergence of Psephos to optimal caching policies.

Recently, a distributed caching strategy based on stochastic gradient algorithms has been extended to content on channels created by sources within the network, under a general stationary ergodic process of contacts between the nodes [7]. As in our scenario, the dynamics of a distributed caching strategy can be shown to maximize locally a utility function, when each channel has a fixed constant delivery deadline. This requires to maintain additional states in each node (at least one per relay for each subscribed channel) and to establish beforehand the uniqueness of the optimal solution (which is shown in several cases). In contrast, Psephos, which was designed for memoryless contacts, is simpler and always converges to the optimal caching strategy for general utility functions of the delivery time.

The above problem has also strong ties to delay-tolerant broadcasting [1, 2, 5, 9]. In [1, 2], a scenario in which updates for a single content item are propagated in a homogeneous contact environment is considered. Optimal broadcasting decision policies aiming to minimize content age are designed. Optimal contact rates with the infrastructure, maximizing a utility of the

content age, are computed in [9] under heterogeneous mobility, while ODEs determining the content age distribution are derived in [5]. Our approach is orthogonal, optimizing the use of storage rather than bandwidth, and further differs from the above works by considering the joint propagation of multiple items.

Finally, our work has connections to data ferrying networks such as KioskNet [8, 10]. In such networks, designated mobile devices act as "ferries" carrying content and providing connectivity, *e.g.*, to remote villages. One possible use of Psephos is to determine optimal pre-fetching policies for such devices, driven by user demand.

3 System Description

3.1 Content Sharing over a Mobile Network

Our system consists of a set $\mathcal{N}$ of N mobile users, partitioned in L distinct classes $\mathcal{C}_1$, $\mathcal{C}_2$, ..., $\mathcal{C}_L$. Denote by $|\mathcal{C}_i|$ the size of class $\mathcal{C}_i$. Both of our main results (Theorems 1 and 2) are proved for a large system, *i.e.*, one in which N tends to infinity. Throughout this work, whenever considering such limits, we assume that the user population in each class grows proportionally to N, *i.e.*, for all i,

$$|\mathcal{C}_i| = r_i N, \tag{1}$$

where r_i is a constant (in N).

Users encounter each other according to independent Poisson processes. In particular, each user $m \in \mathcal{C}_i$ encounters users from class $\mathcal{C}_j$ according to a Poisson process with rate $\lambda_{i,j} \geq 0$. Conditioned on the fact that m meets a user from $\mathcal{C}_j$, the latter is selected uniformly at random from $\mathcal{C}_j$. Moreover, $\lambda_{i,j}$ are constants not depending on N. Contacts are symmetric, *i.e.*, m meets m' if and only if m' meets m. As a result,

$$r_i \lambda_{i,j} = r_j \lambda_{j,i}, \quad \text{for any } i, j. \tag{2}$$

To see this, observe that a given user $m \in \mathcal{C}_i$ meets a given user $m' \in \mathcal{C}_j$ with a rate $\lambda_{i,j}/|\mathcal{C}_j|$, and the latter should equal the rate with which m' meets m.

Each user $m \in \mathcal{C}_i$ accesses the infrastructure according to a Poisson process with rate $\mu_i > 0$, also constant in N. An access event may indicate, *e.g.*, that the user is within the vicinity of her home or office and can access the Internet through a wireless connection; as users are mobile, such access may be intermittent. Alternatively, in scenarios where users have uninterrupted access to the Internet, μ_i can be seen as an upper bound on the rate with which they send requests to the infrastructure, aimed at limiting the load they impose. Note that we require μ_i to be strictly positive: every user accesses the infrastructure infinitely often.

Users in the same class are statistically identical. In particular, they have the same contact characteristics and storage capacities, they exhibit the same

demand for content and accrue the same utility from locating certain content. Nonetheless, users are oblivious to the existence of these classes: a user does not know to which class she belongs and cannot distinguish other users based on their class. For this reason, decisions made by PSEPHOS on what to store in a user's cache do not rely on a priori knowledge of classes.

3.1.1 Caching Policies and Replication Ratios

Mobile users pre-fetch, store and share content they obtain from the Internet. For the sake of concreteness, we will assume that they share and access *websites*. In general, a user may request and access the same content more than once; hence, any type of dynamic content, such as a newsfeed or a blog, can be captured through our model.

We denote the set of all possible websites users may download by $\mathcal{W}$. Each user m in class $\mathcal{C}_i$ maintains a public cache in which she can store at most c_i websites, where $c_i \leq |\mathcal{W}|$. The user makes the contents of this cache publicly available: in particular, whenever m meets a user m', any websites stored in m's cache are made visible to m', and vice versa. For all $w \in \mathcal{W}$, let

$$y_{m,w} = \begin{cases} 1, & \text{if } m \text{ stores } w, \\ 0, & \text{o.w.} \end{cases} \tag{3}$$

be a 0-1 variable indicating if user $m \in \mathcal{N}$ stores website w. We refer to the vector

$$\mathbf{y}_m = [y_{m,w}]_{w \in \mathcal{W}}, \qquad m \in \mathcal{N},$$

as the *caching policy of* m. For all $w \in \mathcal{W}$, denote by

$$x_{i,w} = \sum_{m \in \mathcal{C}_i} y_{m,w} / |\mathcal{C}_i| \tag{4}$$

the fraction of users in class $\mathcal{C}_i$ storing website w—*i.e.*, $x_{i,w}$ is w's *replication ratio* in class $\mathcal{C}_i$. We refer to the vector

$$\mathbf{x}_i = [x_{i,w}]_{w \in \mathcal{W}}, \qquad i \in 1, \ldots, L,$$

as the *replication vector of class* $\mathcal{C}_i$.

Each user $m \in \mathcal{C}_i$ generates requests for a website $w \in \mathcal{W}$ according to a Poisson process with rate $d_{i,w}$. Generated requests for websites are satisfied as follows. If $y_{m,w} = 1$, *i.e.*, the user m stores the requested website in her public cache, the request is satisfied immediately. If not, the request is backlogged. It is subsequently satisfied when the first among the following two events occur: either m meets another user m' s.t. $y_{m',w} = 1$ or m accesses the infrastructure.

Note that, under the above assumptions, a request for a website w is satisfied when *any* copy of the website is retrieved. This is a simplification as website content is dynamic and, in reality, off-line content cached by users could become outdated. However, the above behavior is realistic when (a) websites are updated infrequently and (b) content is diffused quickly among

users that cache it. As described in Section 3.3, we aid this diffusion by always "refreshing" stale content when in contact with a user storing a more recent version.

While a user's request for a website is backlogged, a user may generate a second request for the same website. This is a consequence of our assumption of Poisson request arrivals. Though this may model persistent users or request arrivals from a higher network layer, in practice, it might be unrealistic; nevertheless, we focus on the Poisson assumption for the sake of tractability. In any case, all requests pending for a website are immediately satisfied upon contact with the infrastructure or a user storing the requested website.

Table 1 Summary of Notation

$\mathcal{N}$	Set of mobile users
N	Number of mobile users.
L	Number of user classes.
$\mathcal{W}$	Set of websites.
$\mathcal{C}_i$	i-th class, $i = 1, \ldots, L$.
r_i	Ratio of the size of class $\mathcal{C}_i$ over the size N of the user population.
c_i	Capacity of a cache of a user in class $\mathcal{C}_i$.
c_i'	Spare capacity of a cache of a user in class $\mathcal{C}_i$, not pre-allocated for permanent websites.
$\lambda_{i,j}$	Aggregate contact rate of a user $m \in \mathcal{C}_i$ with users in $\mathcal{C}_j$.
μ_i	Rate with which $m \in \mathcal{C}_i$ accesses the infrastructure.
$d_{i,w}$	Request rate of website w by a user $m \in \mathcal{C}_i$.
$\mathcal{PR}_i$	Set of permanent websites for users in class $\mathcal{C}_i$.
$\mathcal{BL}_i$	Set of blacklisted websites for users in class $\mathcal{C}_i$.
$\mathcal{F}_i$	Unrestricted websites in class $\mathcal{C}_i$.
$\mathbf{y}_m$	Caching policy of user m .
$\mathbf{x}_i$	Vector of replication ratios of websites in class $\mathcal{C}_i$.
$\mathbf{X}$	Matrix of vectors replication ratios $\mathbf{x}_i$.
$Y_{i,w}$	Delay of discovery of website w from a user in $\mathcal{C}_i$.
$U_{i,w}(t)$	Utility at class i when the delay of discovery of website w is equal to t.
$\rho_{i,w}$	Contact rate of a user in $\mathcal{C}_i$ with users storing w.
$F(\mathbf{X})$	The system social welfare.
D	The feasible domain of (10).
$\mathbf{v}_m$	Vote vector of user m.
$\alpha(N)$	Gain of cache policy selection mechanism.
$\beta(N)$	Gain of vote averaging mechanism.
$T_{m,w}$	Time elapsed since last encounter of m with the infrastructure or a user storing w.
$n_{m,w}$	Number of pending requests for w at user m.
$q_{m,w}$	Report provided by m w.r.t. website w.

3.1.2 Permanent and Blacklisted Websites

In general, the contents of a user's public cache are determined by Psephos, our distributed caching mechanism, with the global goal of minimizing delays. Intuitively, each user preallocates and commits this public cache to be used as deemed appropriate. As such, the contents of this public cache are part of the system design: the caching policies $\mathbf{y}_m$ are determined by the optimization

performed by PSEPHOS as described in Section 3.3. Moreover, when a user's request is satisfied, the website retrieved is *not* placed the public cache; it is stored by the user in a private storage space and viewed separately.

System performance in terms of delays can only improve if users share the contents of their private cache. However, we do not model this for two reasons. First, users may immediately view and discard websites in their private cache. As such, the benefit from sharing private caches might only be marginal. Second, focusing only on a system-managed caches allows the analysis to be tractable, as cache contents become a "free variable" of our design, rather than a parameter dependent on user demand.

Nonetheless, our model assumes users have partial control of what is prefetched in their cache: they can choose to always store certain sites they like in their cache and block others from ever being stored in it. The sites they choose to store permanently occupy part of the cache, so PSEPHOS can use only the remaining capacity to store any additional websites. Moreover, it cannot select to store any site that is explicitly blocked by a user.

More specifically, users in class $\mathcal{C}_i$ have a set $\mathcal{PR}_i \subseteq \mathcal{W}$ of preferred websites, which we call *permanent*, that are *always* stored in their cache, *i.e.*,

$$y_{m,w} = 1, \qquad \text{for all } m \in \mathcal{C}_i, w \in \mathcal{PR}_i.$$

Naturally, we assume that cache sizes are sufficient to store permanent sites, *i.e.*,

$$0 \leq |\mathcal{PR}_i| \leq c_i. \tag{5}$$

For each class $\mathcal{C}_i$ there is also a set of websites $\mathcal{BL}_i \subseteq \mathcal{W}$, which we call *blacklisted*, that may *never* be stored in the cache of a user in $\mathcal{C}_i$, *i.e.*,

$$y_{m,w} = 0, \qquad \text{for all } m \in \mathcal{C}_i, w \in \mathcal{BL}_i.$$

This could be the case if, *e.g.*, such content is deemed to be offensive or inappropriate. We assume that $\mathcal{PR}_i \cap \mathcal{BL}_i = \emptyset$ (permanent websites are not blacklisted) and

$$c_i \leq |\mathcal{W}| - |\mathcal{BL}_i|. \tag{6}$$

Inequality (6) states that the number of remaining websites after excluding blacklisted websites exceeds the capacity c_i of users in class $\mathcal{C}_i$. We can assume that this is true without loss of generality: if not, caching decisions are trivial, as all non-blacklisted websites can be placed in the cache. Note that the sets $\mathcal{PR}_i$ and $\mathcal{BL}_i$ need not be related in any way with the demand $d_{i,w}$: for example, we allow users to request websites that they have blacklisted.

The above imply that, whenever a user accesses the infrastructure, PSEPHOS can reshuffle at most

$$c_i' = c_i - |\mathcal{PR}_i|, \tag{7}$$

websites, as the rest are permanent. We will refer to c'_i as the *spare capacity* of a user in $\mathcal{C}_i$. Moreover, PSEPHOS can allocate the capacity c'_i to websites belonging to the following set:

$$\mathcal{F}_i = \mathcal{W} \setminus (\mathcal{PR}_i \cup \mathcal{BL}_i), \tag{8}$$

i.e., the set of websites that are neither permanent or blacklisted. We will refer to such websites as *unrestricted* because they are not restricted by prior user requirements. Note that (5)–(8) imply that $|\mathcal{F}_i| \geq c'_i$, though, in the case where $|\mathcal{F}_i| = c'_i$, the reshuffling decision is trivial: all websites in $\mathcal{F}_i$ are placed in the cache.

3.2 Maximizing the Social Welfare

The goal of our distributed caching mechanism is to determine the contents of the public cache of each user in order to maximize the average system utility—namely, the social welfare. Below, we define this objective formally.

Let $Y_{i,w} \geq 0$ be the time until a request for website w generated by a user chosen uniformly at random from class $\mathcal{C}_i$ is satisfied. Note that, if the user stores w, then a request for w is satisfied immediately, *i.e.*, $Y_{i,w} = 0$. If she does not, $Y_{i,w}$ is an exponentially distributed random variable with mean $1/\rho_{i,w}$, where

$$\rho_{i,w} = \mu_i + \sum_{j=1}^{L} \lambda_{i,j} x_{j,w}. \tag{9}$$

To see this, observe that the process with which $m \in \mathcal{C}_i$ comes into contact with other users storing w is Poisson with rate

$$\sum_{j=1}^{L} \sum_{m \in \mathcal{C}_j} \frac{\lambda_{i,j}}{|\mathcal{C}_j|} \cdot y_{m,w} = \sum_{j=1}^{L} \lambda_{i,j} \cdot \sum_{m \in \mathcal{C}_j} \frac{y_{m,w}}{|\mathcal{C}_j|} = \sum_{j=1}^{L} \lambda_{i,j} x_{j,w}.$$

We assume that users in $\mathcal{C}_i$ have utilities $U_{i,w} : \mathbb{R}_+ \to \mathbb{R}$ that are functions of the time required to find website w. These utilities capture the "impatience" [13] of a user, *i.e.*, how sensitive she is to getting a given website quickly. We make the following assumption about the utilities $U_{i,w}$:

Assumption 1 *For all i and w, $U_{i,w}$ are non-increasing, differentiable, and both $|U_{i,w}(t)|$ and $|U'_{i,w}(t)|$ are $o\left(e^{ct}\right)$, for all $c > 0$.*

The assumption that $U_{i,w}$ are non-increasing is natural (see also [9, 13]): the longer a user has to wait, the lower the utility she should obtain from finding the content. Note that we do not restrict ourselves to positive functions; *e.g.*, the function $U_{i,w}(t) = -t$ is of interest, as maximizing it amounts to minimizing the delay experienced by users. The boundedness assumption is of a rather mild nature. It holds for all positive non-increasing functions, and for negative functions it suffices that their absolute value grows slower than exponentially

with t. Finally, the differentiability assumption can be relaxed to account even for discontinuous functions. We discuss such a relaxation in Section 7.2.

The goal of our mechanism will be to choose the caching policy at each user so that the average utility per user (or, the social welfare) is maximized. In particular, for $\mathbf{X} \in \mathbb{R}^{L\times|\mathcal{W}|}$ the matrix whose i-th row is the replication vector $\mathbf{x}_i$, we wish to solve the following optimization problem:

$$\text{Maximize } F(\mathbf{X}) = \sum_{i,w} r_i d_{i,w} \mathbb{E}[U_{i,w}(Y_{i,w})] \tag{10a}$$

$$\text{subject to: } \sum_{w\in\mathcal{F}_i} x_{i,w} \leq c'_i, \quad \text{for all } i, \tag{10b}$$

$$0 \leq x_{i,w} \leq 1, \quad \text{for all } i, w, \tag{10c}$$

$$x_{i,w} = 1, \quad \text{for all } i, w \in \mathcal{PR}_i, \tag{10d}$$

$$x_{i,w} = 0, \quad \text{for all } i, w \in \mathcal{BL}_i. \tag{10e}$$

where $r_i = \frac{|\mathcal{C}_i|}{N}$, $d_{i,w}$ the rate of request of website w in class $\mathcal{C}_i$, and c'_i the spare capacity of a cache in class i. Note that the above optimization problem is stated in terms of the *replication ratios* $\mathbf{x}_i$, rather than the cache contents $\mathbf{y}_m$ of each user in the system. This is a consequence of the fact that the expected utilities are direct functions of the replication ratios. Posing (10) like this, rather than in terms of the cache contents directly, has several advantages. First, it significantly reduces the number of unknown variables, as we need $|\mathcal{W}|$ variables per class, rather than $|\mathcal{C}_i| \times |\mathcal{W}|$. Second, (10) has linear rather than integral constraints—in fact, as stated in Lemma 1 of Section 5.1, it is convex. As such, it is easier to solve.

Nevertheless, the mechanism we consider can only affect the replication ratios $\mathbf{X}$ indirectly: Psephos determines the caching policies $\mathbf{y}_m$ at individual users. In this sense, we wish to design a mechanism under which, as cache contents change, the (macroscopic) replication ratios converge to values that maximize the social welfare.

Denote by

$$D = \{\mathbf{X} \in \mathbb{R}^{L\times|\mathcal{W}|} \text{ satisfying (10b) to (10e)}\} \tag{11}$$

the feasible domain of (10). Note that, in reality, our mechanism cannot achieve *any* possible replication ratio in D. The achievable replication vectors in a class of $r_i N$ are "quantized", in the sense that the coordinates of $\mathbf{x}_i$ need to be integer multiples of $1/r_i N$. Because of this, it is possible that our mechanism cannot reach the optimal (real-valued) solution in D; instead, it may exhibit an error of a factor $O\left(\frac{1}{N}\right)$. However, as N increases, the above error becomes negligible (see also the discussion at the end of Section 4.1).

3.3 The Psephos Mechanism

In general, there are two operations that can be performed on a user's cache. First, a cache can be *refreshed*: more recent versions of the websites stored in it

can be retrieved. Second, the cache can be *reshuffled*: certain websites may be removed and replaced by others. The opportunities for both operations arise when a user accesses the infrastructure as well as when she encounters other users. The boundedness of c_i, μ_i and $\sum_j \lambda_{i,j}$ in N imply that both operations could be executed without affecting the system's scalability, as the bandwidth consumption incurred at each user would be constant.

Nonetheless, to simplify our analysis, we will assume that reshuffling may occur *only* when a user encounters the infrastructure. Note that the reshuffling operation is precisely the operation determined by Psephos: our distributed mechanism will dictate how cache contents change upon an encounter with the infrastructure. For the sake of concreteness, we will assume that refreshing occurs at *all* possible opportunities; this ensures that cached content is as fresh as possible. Note that, since the user utilities depend only on the delay until locating the desired content (*i.e.*, users are indifferent to how stale websites are), the refreshing operation has no effect on social welfare.

Formally, reshuffling cache contents under Psephos takes place as follows. To begin with, a user does not reshuffle at all encounters with the infrastructure (though she always refreshes her content). Instead, a reshuffling occurs with probability

$$\alpha = \alpha(N) > 0,$$

which we allow to be a function of the system size. Moreover, each user $m \in C_i$ maintains $|\mathcal{F}_i|$ real-valued processes

$$v_{m,w}(t), \quad w \in \mathcal{F}_i,$$

associated with each unrestricted website. We refer to $v_{m,w}(t)$ as the *vote* of website w at user m at time t, and to

$$\mathbf{v}_m(t) = [v_{m,w}(t)]_{w \in \mathcal{F}_i}$$

as the *vote vector* of user m.

These vote processes indicate how "important" a website is, and are used to select m's caching policy as follows: whenever m accesses the infrastructure and decides to reshuffle her cache, she sorts the vector $\mathbf{v}_m(t)$ in a decreasing order, and then caches the websites with the c_i' highest votes.

Obviously, the process determining the vote vector $\mathbf{v}_m$ is crucial: we describe it in detail in Section 3.3.1. We note that these votes are updated in a distributed manner through local decisions made by the mobile user. In particular, user m adjusts $\mathbf{v}_m$ whenever she meets other users or when she generates a request for a website.

As we will see, the introduction of α aims at "slowing down" the process with which users reshuffle their caches. Randomization is required by our analysis to maintain Poisson encounters with the infrastructure; in practice, reshuffling once in every α^{-1} encounters with the infrastructure would suffice.

3.3.1 The Vote Processes in Psephos

As discussed above, the vote vector $\mathbf{v}_m(t)$ plays a crucial role in selecting the caching policy at user m. In this section, we describe in detail how these vote processes are maintained and updated.

The vote at a user m is an Exponentially Weighted Moving Average (EWMA), constructed by "reports" she generates as well as "reports" she receives from other users she encounters. Such "reports" aim to indicate the effect that storing w at m has on the utility of the users that generate them. Below, we formally define how these reports are computed and exchanged among the users; some intuition into why these particular quantities are reported is presented in the next section.

To begin with, for every site w, m keeps track of the elapsed time between her last two consecutive encounters with another user that stores w or the infrastructure. We denote this quantity at time t by $T_{m,w}(t) > 0$. More formally, we define the "inter-service time" for item w w.r.t a user m to be the time elapsed between two successive opportunities to receive w. Note that such events constitute a Poisson process with rate $\rho_{i,w}$, given by (9), as m can obtain w either when it encounters the infrastructure or a user storing w. We define $T_{m,w}(t)$ to be the current estimate of the inter-service time, computed as the elapsed time between the last two consecutive epochs of this Poisson process, immediately preceding t.

Mobile user m also maintains a buffer of all pending requests; let $n_{m,w}(t)$ be the number of pending requests for w that m has at time t. Note that for every w stored in m's cache $n_{m,w}(t) = 0$.

Whenever $m \in \mathcal{C}_i$ meets another user, she reports to the latter the following quantity for every website w:

$$q_{m,w}(t) = -T_{m,w}(t) \cdot U'_{i,w}\left(T_{m,w}(t)\right) \cdot n_{m,w}(t), \tag{12}$$

where $U'_{i,w}$ the derivative of utility $U_{i,w}$. This quantity serves as a "report", capturing the effect that website w has on the utility of m. If user m meets a user m' that stores w, m first reports $q_{m,w}(t)$ and then updates the above values accordingly (by, *e.g.*, updating the value of $T_{m,w}$ and setting $n_{m,w}$ to zero).

For $m \in \mathcal{C}_i$, the average vote $v_{m,w}(t)$ (where $w \in \mathcal{F}_i$) is updated in the following way: between requests or encounters with other users, $v_{m,w}(t)$ decays exponentially with a rate $\beta = \beta(N) > 0$, a positive gain factor which we allow to depend on N. Whenever m receives a report, $v_{m,w}$ is incremented by the value in this report, weighted by $\beta(N)$. Moreover, whenever m generates a request, $v_{m,w}(t)$ is incremented by $\beta(N)(U_{i,w}(0) - U_{i,w}(T_{m,w}(t)))$.

More formally, for small $\delta > 0$, let $g_{m,w}(t, t+\delta)$ be such that

$$v_{m,w}(t+\delta) = (1-\beta(N)\delta)v_{m,w}(t) + \beta(N)g_{m,w}(t, t+\delta). \tag{13}$$

Then, with probability $1 - O\left(\delta^2\right)$,

$$g_{m,w}(t,t+\delta) = \begin{cases} q_{m',w}(t), \text{ if } m, m' \text{ meet in } (t,t+\delta] \\ U_{i,w}(0) - U_{i,w}(T_{m,w}(t)), \text{ if } m \\ \qquad \text{requests } w \text{ in } (t,t+\delta] \\ 0, \text{ o.w.} \end{cases} \tag{14}$$

Because of the exponential decay between increments $v_{m,w}(t)$, is called an EWMA. The gain factor $\beta(N)$ allows us to make the system more or less sensitive to newer "reports", thereby controlling the variability of $v_{m,w}(t)$. Note that the vote vector $\mathbf{v}$ is computed based only on local information: no knowledge of the rates $\lambda_{i,j}$, μ_i, $d_{i,w}$ or any distinction among users from different classes is required.

To assess the storage complexity of Psephos, note that a user needs to maintain a vote process only for websites in $\mathcal{F}_i$ for which either $d_{i,w} \neq 0$ or $d_{j,w} \neq 0$ for some j such that $\lambda_{i,j} \neq 0$. In other words, m stores $w \in \mathcal{F}_i$ if either m herself or a user she meets has requested w at some time in the past. For all other websites, the vote $v_{m,w}(t)$ is zero and need not be maintained. In addition, whenever a vote process converges to zero (indicating that the website it corresponds to has not been requested recently), the vote process again need not be stored any longer.

Similarly, the processes $T_{m,w}(t)$ and $n_{m,w}(t)$ need only be maintained for websites for which m is interested in (*i.e.*, $d_{i,w} > 0$). In practice, a user may keep track of these values only if she has pending requests for w.

Finally, regarding the message complexity of Psephos, note that $q_{m,w}(t)$ must be reported only for w for which m has pending requests, *i.e.*, $n_{m,w}(t) > 0$. For all other w, $q_{m,w}$ is zero and need not be reported. Put differently, users sends a report regarding a website w if and only if they also send a request for w; in this sense, the message complexity is of the same order as the complexity of querying other users whether they store w. If the latter is deemed excessive, and users need to "throttle" their request rate—*e.g.*, by transmitting requests at a lower rate—this can be modeled in our protocol through an appropriate rescaling of the contact rates. Alternatively, in such a scheme, only reports that are accompanied by requests for a website w need be transmitted, and encounters with users storing w will contribute to the computation of $T_{m.w}$ only if such requests are sent in parallel.

3.3.2 Psephos *as Gradient Descent*

Some intuition behind Psephos can be obtained by studying the "reports" exchanged by users. These reports are designed so that (14), the quantity that the EWMA averages over, is in fact an estimator of the gradient of F, the social welfare. In particular, we show that for $m \in \mathcal{C}_i$

$$\mathbb{E}[g_{m,w}(t,t+\delta)] \sim \delta \cdot \frac{1}{r_i} \frac{\partial F}{\partial x_{i,w}} \tag{15}$$

—see also Lemma 3 for a more formal statement.

Hence, the averaging performed by the EWMA $v_{m,w}(t)$ should yield, if sufficient time passes, a good estimate of $\frac{\partial F}{\partial x_{i,w}}$. Suppose now that for some $m \in \mathcal{C}_i$ the website w has the highest vote, but it is not stored by m. Then, (15) indicates that $\frac{\partial F}{\partial x_{i,w}}$ is the highest among all derivatives $[\frac{\partial F}{\partial x_{i,w'}}]_{w' \in \mathcal{F}_i}$. The latter suggests however that increasing the replication ratio of w within the class $\mathcal{C}_i$ will yield the highest possible increase on the social welfare F (compared to increasing the ratio of some other website). Adding w to m's cache accomplishes precisely this: it increases the replication ratio of w in its class, albeit infinitesimally.

Hence, our approach of selecting the sites with the highest votes is a form of gradient descent: its outcome changes replication ratios according to the largest components of gradient ∇F. Alternatively, provided that the vote processes correctly estimate this gradient, our distributed mechanism does "the right thing", by slightly increasing the social welfare with each website placed in the cache.

4 Main Results

Our main result is to show that the caching policies computed by Psephos lead to replication ratios per class that are optimal, *i.e.*, maximize the system's social welfare. We first show this formally under a time-scale separation assumption: we require that the cache policy selection preformed by Psephos occurs at a slower pace than the evolution of the vote processes $\mathbf{v}_m$. Subsequently, we provide some evidence as to why this time-scale separation might not be necessary in practice.

4.1 Convergence to Optimal Caching Policies

Our first main result is that, if the cache policy selection occurs at a much slower pace than the rate with which the vote processes evolve, our system is guaranteed to converge to an optimal caching policy. More formally,

Theorem 1 *Let $\{\mathbf{X}(t)\}_{t\geq 0}$ be the replication ratios at time t resulting from the caching polices selected by* Psephos*, as described in Section 3.3. Moreover, assume that*

$$\lim_{N\to\infty} \frac{N\alpha(N)}{\beta(N)} = 0, \tag{16}$$

Then, the steady state distribution of $\{X(t)\}_{t\geq 0}$ is such that

$$\lim_{t\to\infty} \mathbf{P}\big(|F(\mathbf{X}(t)) - \sup_{\mathbf{X}\in D} F(\mathbf{X})| > \epsilon(N)\big) \leq \epsilon(N),$$

where $\lim_{N\to\infty} \epsilon(N) = 0$. In other words,

$$\lim_{N\to\infty} \lim_{t\to\infty} F(\mathbf{X}(t)) = \sup_{\mathbf{X}\in D} F(\mathbf{X}), \quad \textit{in probability.}$$

To gain some intuition behind the above theorem, recall that $\alpha(N)$ is the probability with which PSEPHOS reshuffles a user's cache upon accessing the infrastructure. Moreover, recall that $\beta(N)$ is the rate with which the exponentially weighted moving average scheme in Section 3.3.1 adapts the current vote vector $\mathbf{v}_m$.

Eq. (16) states that $\alpha(N)$ is much smaller than $\beta(N)$—by a factor of N, at least. This implies that there is a clear separation between the time-scale at which the vote processes $\mathbf{v}_m(t)$ change and at the time-scale at which caching policies are changed. In particular, the latter process is much slower than the former.

Intuitively, this time separation implies that, between two consecutive updates of the caching policy of a user $m \in \mathcal{C}_i$, the vote process $\mathbf{v}_m$ has already converged close enough to the quantity that it is supposed to estimate — namely, to $\frac{1}{r_i}\left[\frac{\partial F}{\partial x_{i,w}}\right]_{w\in\mathcal{F}_i}$ (see also Lemma 6). The above convergence allows us to characterize the behavior of the system in terms of a system in which votes are almost exact in estimating the above quantity.

The quantity $\epsilon(N)$ can be used to characterize how close $\mathbf{X}(t)$ is to an optimizer of F. In particular, it is shown in Section 5.5 that it is of the following order:

$$\epsilon(N) = \sqrt{O\left(\sqrt{\frac{N\alpha(N)}{\beta(N)}}\right) + O\left(\frac{1}{N}\right)}. \tag{17}$$

The first term of the above formula is due to the "error" induced by the estimate of ∇F through the votes $\mathbf{v}_m$. The second term is due to the "quantization" effect we mentioned in Section 3.2. Roughly, the $\mathbf{X}$ at which our system converges may have an $O(\frac{1}{N})$ error compared to the optimal (real-valued) solution in D. Nonetheless, as N tends to infinity, this error term vanishes.

4.2 A Single Timescale Fluid Model

Maintaining a "fast" caching policy selection process is appealing from a practical perspective: we would like our mechanism to converge to the optimal policies as quickly as possible. However, our reliance on two time-scales to prove Theorem 1 restricts us from "speeding up" our mechanism, *e.g.*, by maintaining a high value of $\alpha(N)$, or by allowing the cache to be reshuffled upon contact with other users.

It is therefore interesting to understand whether a time-scale separation is truly necessary. Our second main result suggests that it is not. To show this, we consider the dynamics of a deterministic system in which votes *and* cache policies evolve jointly. Surprisingly, in spite of the lack of a separation between the time-scales of evolution of this system, the system converges where it is supposed to: the cache policies converge to a maximizer of F in D, and the votes converge to the gradient ∇F.

We first give a formal statement of our result, and then discuss how it applies within the context of our system. Consider a convex bounded set $C \subset \mathbb{R}_+^d$. Then, for $F : C \to \mathbb{R}$ a function which is strictly concave, non-decreasing and differentiable, we consider the following dynamics for the vectors $\mathbf{x} \in C$ and $\mathbf{v} \in \mathbb{R}_+^d$

$$\frac{d}{dt}\mathbf{x} = \alpha \left(G(\mathbf{v}) - \mathbf{x} \right), \tag{18a}$$

$$\frac{d}{dt}\mathbf{v} = \beta \left(\nabla F(\mathbf{x}) - \mathbf{v} \right), \tag{18b}$$

where α, β are positive constants, function G is defined as

$$G(\mathbf{v}) = \arg\max_{\mathbf{x} \in C} \langle \mathbf{x}, \mathbf{v} \rangle, \tag{19}$$

where $\langle \mathbf{x}, \mathbf{v} \rangle$ the inner product of $\mathbf{x}$ and $\mathbf{v}$ (in case of multiple maximisers, anyone can be chosen) and $\nabla F(\mathbf{x})$ denotes the gradient of F at $\mathbf{x}$. Then, the following theorem holds:

Theorem 2 *Assume that C contains two vectors $\mathbf{z}$, $\mathbf{z}'$ such that $z_i < z_i'$ for all $i = 1, \ldots, d$. Under the above assumptions, the dynamical system* (18) *converges to $(\mathbf{x}^*, \mathbf{v}^*)$ where $\mathbf{x}^*$ achieves the maximum of F over C, and $\mathbf{v}^* = \nabla F(\mathbf{x}^*)$.*

To see how the above applies to our system, assume for simplicity that $\mathcal{F}_i = \mathcal{W}$ (*i.e.*, there are no permanent or blacklisted sites), and let $d = L \times W$ and $C = D$. Note that the latter is indeed a convex set. As we will see (*c.f.* Lemma 1), our objective function F is indeed concave (though not necessarily strictly), non-decreasing and differentiable.

Then, the system (18) evolves in a similar way to our dynamical system under the following simplifications. First, all users in a class are assumed to have *exactly the same vote vector*, although this vector need not be "correct", in the sense that it may not necessarily be equal to the gradient ∇F. Second, the rate with which cache policies are changed by our mechanism is the same in each class: it can be verified (see Section 5.4) that this occurs if μ_i / r_i does not depend on i. Finally, the evolution of both votes and cache policies has been replaced with a "fluid limit", in which N is assumed to be infinite and the random changes in the system have been replaced with their expectations.

Under these simplifications, the votes in each class are represented by vector $\mathbf{v}$, which, as all users in the same class have the same votes, can be written as the concatenation of the vectors $\mathbf{v} = [\mathbf{v}_1; \mathbf{v}_2; \ldots; \mathbf{v}_L]$, where $\mathbf{v}_i$ the vote vector of class $\mathcal{C}_i$. Note that (18b) indeed evolves as a "fluid limit" of the votes in the system, where the random reports in the EWMA have been replaced with their expectation.

Similarly, (18a) captures the dynamic evolution of the replication ratios over all classes. To see this, observe that given the constraints C the inner product $\langle \mathbf{x}, \mathbf{v} \rangle$ is maximized in terms of $\mathbf{x} \in C$ when $\mathbf{x}$ takes the value 1 at the c_i top elements of each sub-vector $\mathbf{v}_i$ of $\mathbf{v}$, and the value 0 everywhere else.

This is indeed implemented by our cache policy selection mechanism at each user. In the fluid limit, each such change brings an infinitesimal change in the replication ratio of a class, and (18a) captures the system dynamics.

In spite of the simplifications involved, the convergence of (18) is far from obvious. In particular, as there is no time separation between the parameters α and β, the evolution of (18a) is not guaranteed to be supplied with a correct estimate of the gradient ∇F through the vote vector $\mathbf{v}$. From a technical standpoint, the proof requires a Lyapunov function that exploits a relationship between the function G and conjugate duality (see Section 6).

Nonetheless, Theorem 2 states that this system indeed has the desirable behavior: on one hand, $\mathbf{v}$ eventually converges to the gradient of F, and $\mathbf{x}$ converges to its (unique, due to strict convexity) maximizer. The above result, in spite of the simplifications involved, suggests that a time-scale separation between the two processes is not strictly necessary.

5 Proof of Theorem 1

To prove Theorem 1, we need to determine how the replication ratios $\mathbf{x}$ evolve over time. Determining the drift of the replication ratios amounts to determining the aggregate of changes across all the caches in a class. In turn, cache changes depend on the vote processes of individual users, and vote processes across users in the class are not identical. As a result, the challenge lies in understanding how the multiple, non-identical vote processes maintained by individual users, driving the changes in their cache contents, affect the "macroscopic" behavior of the system.

To address this, our proof of Theorem 1 is structured as follows. First, in Section 5.1, we establish that the objective function of the optimization problem (10) is a concave function of the replication ratios $\mathbf{x}$. As such (10) is a convex optimization problem. In Section 5.2, we give a more formal description of the change of the contents of a user's cache upon contact with the infrastructure. In particular, we describe a *threshold* such that websites with votes below this threshold are excluded from the cache while websites with votes above the threshold are included.

In Section 5.3, we use this definition to show that, for $m \in \mathcal{C}_i$, the expected value of the drift of a vote function $v_{m,w}$ is proportional to the derivative $\frac{\partial F}{\partial x_{i,w}}$. This allows us to establish in Section 5.4, that, under proper time-scale separation between the cache contents evolve approximately *as if all vote processes in a class were identical and, in particular, equal to the above derivatives*. In other words, although vote processes are different across users withing the same class, under time-scale separation they will be close enough to each other. This description greatly simplifies the dynamics of the system; in Section 5.5, we establish that, under such dynamics, the social welfare increases through time and, in particular, replication ratios converge to maximizers of (10), thereby proving the theorem.

5.1 Convexity and Differentiability

We begin our analysis by showing that maximizing social welfare is a convex optimization problem [4,14]. This was proved in a slightly different form in [13] (see Theorems 1 and 2 therein); we repeat the proof below for the sake of completeness.

Lemma 1 *The optimization problem* (10) *is convex. In particular, the objective function F is concave in $\mathbf{X} \in D$.*

Proof Our proof follows the same arguments as [13]. Note that $\mathbb{E}[U_{i,w}(Y_{i,w})]$ can be written as

$$x_{i,w}U_{i,w}(0) + (1 - x_{i,w}) \int_{t=0}^{\infty} U_{i,w}(t)\rho_{i,w}e^{-\rho_{i,w}t}dt$$

where $\rho_{i,w} = \mu_i + \sum_j \lambda_{i,j}x_{j,w}$. By Assumption 1, $U_{i,w}(t) = U_{i,w}(0) + \int_0^t U'_{i,w}(s)ds$, for all $t \geq 0$. This yields

$$\mathbb{E}[U_{i,w}(Y_{i,w})] = U_{i,w}(0) + (1 - x_{i,w}) \int_0^{\infty} U'_{i,w}(s)e^{-\rho_{i,w}s}ds \tag{20}$$

Since $U_{i,w}$ is non-increasing, $U'_{i,w}(s)$ is non-positive. Therefore, $(1 - x_{i,w})$ and $-\int_0^{\infty} U'_{i,w}(s)e^{-\rho_{i,w}s}ds$ are non-increasing, non-negative convex functions of $\mathbf{x}_i$; it can be verified that the product of such functions is convex [13]. Hence, $\mathbb{E}[U_i(Y_{i,w})]$ is concave as the sum of a concave and a constant function, and so is F as the positive sum of concave functions. □

Having established the concavity of F, we turn our attention to its gradient ∇F. The following lemma, which relies on Assumption 1, establishes the smoothness of ∇F.

Lemma 2 *The objective function F is twice continuously differentiable in D and*

$$\begin{aligned} \frac{\partial F}{\partial x_{i,w}} &= r_i d_{i,w} \mathbb{E}[U_{i,w}(0) - U_{i,w}(\hat{Y}_{i,w})] \\ &\quad - \sum_j r_j \frac{d_{j,w}}{\rho_{j,w}} (1 - x_{j,w}) \lambda_{j,i} \mathbb{E}[\hat{Y}_{j,w} U'_j(\hat{Y}_{j,w})] \end{aligned} \tag{21}$$

where $\hat{Y}_{i,w}$ is the time it takes to find the website conditioned on the event that $Y_{i,w} \neq 0$.

Proof From (20) we have that

$$\begin{aligned} \frac{\partial F}{\partial x_{i,w}} &= -r_i d_{i,w} \int_0^{\infty} U'_{i,w}(s)e^{-\rho_{i,w}s}ds \\ &\quad - \sum_j r_j d_{j,w}(1 - x_{j,w}) \int_0^{\infty} U'_{j,w}(s)e^{-\rho_{j,w}s} s\lambda_{j,i}ds. \end{aligned} \tag{22}$$

Moreover,

$$\int_0^\infty U'_{i,w}(s)e^{-\rho_{i,w}s}ds \overset{(20)}{=} \int_0^\infty \int_0^t U'_{i,w}(s)\rho_{i,w}e^{-\rho_{i,w}t}dsdt$$
$$= \int_0^\infty [U_{i,w}(t) - U_{i,w}(0)]\rho_{i,w}e^{-\rho_{i,w}t}dt$$

while

$$\int_0^\infty sU'_{j,w}(s)e^{-\rho_{j,w}s}ds = \frac{1}{\rho_{j,w}} \int_0^\infty sU'_{j,w}(s)\rho_{j,w}e^{-\rho_{j,w}s}ds$$

Eq. (21) therefore follows. Note that, $e^{-\rho_{j,w}s}$ is decreasing in $\rho_{j,w}$ for all j and w. Hence, by (22) and the monotone convergence theorem $\partial F/\partial x_{i,w}$ is continuous in $\rho_{j,w} \in [\mu_j, 1]$ (a limit in terms of $\rho_{j,w}$ and the integration appearing in (22) can be exchanged). Hence, $\rho_{j,w}$ is also continuous in $\mathbf{X}$. Note the following fact:

$$\text{If } |f(t)| = o(e^{ct}) \ \forall c > 0, \text{ then so is } |tf(t)|. \tag{23}$$

Hence, the quantity

$$\left|\frac{\partial F}{\partial x_{i,w}}\right| \overset{(22)}{\leq} r_i d_{i,w} \int_0^\infty (-U'_{i,w}(s))e^{-\mu_i s}ds + \sum_j r_j d_{j,w}\lambda_{j,w} \int_0^\infty (-sU'_{j,w}(s))e^{-\mu_j s}ds,$$

is bounded by Assumption 1 and (23). Differentiating (22) by $x_{i',w}$ gives a closed form for the Hessian of F, which can similarly be shown to be continuous and bounded; the finiteness of $\int_0^\infty |U'_{j,w}(s)|s^2e^{-\mu_j s}ds$, for all j and w, required for this result, is indeed implied by Assumption 1 and (23). □

An immediate implication of Lemma 2 is that ∇F is Lipschitz continuous, as F is twice continuously differentiable over the bounded domain D.

5.2 A Threshold Function and Mass Preservation

In this section, we give a more formal description of the cache policy selection process in terms of the vectors $\mathbf{v}_m$. In particular, we define a *threshold*, such that, if a vote $v_{m,w}$ exceeds this value, m will store w at the next encounter with the infrastructure while, if the vote is below this value, website w will stay out of the cache.

Indeed, let $m \in \mathcal{C}_i$ and assume that the vector $\mathbf{v}_m$ is sorted in decreasing order. Let $\tau_i(\mathbf{v})$ be the average of the c'_i-th and $c'_i + 1$-th entry in the sorted vector $\mathbf{v}_m$ (if the latter does not exist, then $|\mathcal{F}_i| = c'_i$, and the mechanism stores every unrestricted website). It is clear that, since the top c'_i items are placed in the cache, any website $w \in \mathcal{F}_i$ s.t. $v_{m,w} > \tau_i(\mathbf{v})$ is placed in m's cache while every website $w \in \mathcal{F}_i$ s.t. $v_{m,w} < \tau_i(\mathbf{v}_m)$ is removed. Hence, $\tau_i(\mathbf{v}_m)$ is a threshold indicating whether a website enters the cache or not.

Note that websites such that $v_{m,w} = \tau_i(\mathbf{v})$ exist if and only if the c'_i-th and $c'_i + 1$-th entry in the sorted vector $\mathbf{v}_m$ are equal. If such websites exist, we assume that ties among them are broken arbitrarily—any such site can be placed in the cache of user m, so that c'_i sites belonging to $\mathcal{F}_i$ are in the cache at all times.

The above imply that the cache of $m \in \mathcal{C}_i$ is updated according to the following process. For every $w \in \mathcal{F}_i$,

$$y_{m,w}(t+\delta) = y_{m,w}(t) + h_{m,w}(t, t+\delta), \quad \text{where} \tag{24}$$

$$h_{m,w}(t, t+\delta) = \begin{cases} 0, & \text{if } m \text{ does not reshuffle} \\ & \text{its cache in } (t, t+\delta] \\ \Delta_{m,w}(\mathbf{v}_m, \mathbf{y}_m), & \text{o.w.} \end{cases} \tag{25}$$

and $\Delta_{m,w}(\mathbf{v}_m, \mathbf{y}_m)$ is given by

$$\begin{aligned} \Delta_{m,w}(\mathbf{v}_m, \mathbf{y}_m) = {} & \mathbb{1}_{v_{m,w} > \tau_i(\mathbf{v}_m) \wedge y_{m,w}=0} - \mathbb{1}_{v_{m,w} < \tau_i(\mathbf{v}_m) \wedge y_{m,w}=1} \\ & + \mathbb{1}_{v_{m,w}=\tau_i(\mathbf{v}_m)} z_{m,w}(\mathbf{v}_m, \mathbf{y}_m). \end{aligned} \tag{26}$$

The quantity $z_{m,w}(\mathbf{v}_m, \mathbf{y}_m)$ may be either 0,1 or -1, depending on how m decides to break ties among websites with the same vote. However, because the total number of websites in its cache does not change, the number of websites entering the cache has to equal the number of websites exiting the cache. As a result, the following "mass preservation" rule must hold for all m:

$$\sum_{w \in \mathcal{F}_i} \Delta_{m,w}(\mathbf{v}_m, \mathbf{y}_m) = 0. \tag{27}$$

Eq. (27) places a constraint on the quantities $z_{m,w}(\mathbf{v}_m, \mathbf{y}_m)$: they may be arbitrary insofar as the total "mass" of websites moved around is preserved, and every user in class $\mathcal{C}_i$ stores exactly c'_i websites from $\mathcal{F}_i$.

5.3 Expected System Drift within Small Intervals

Given a user m, the function $g_{m,w}$, given by (14), characterizes the change of the vote process $v_{m,w}$ within a small time interval $(t, t+\delta]$. Similarly, the function $h_{m,w}$, given by (25), characterizes the change of the caching variable $y_{m,w}$ within a small time interval. In what follows, we characterize their expectations and their variances, in terms of δ.

Our first result is that, in expectation, $g_{m,w}(t, t+\delta)/\delta$ is equal to the partial derivative $\frac{\partial F}{\partial x_{i,w}}$, scaled by a factor $1/r_i$. This is a fundamental part of our proof, as it implies that the EWMA process defining the votes "averages out" a random process whose expectation is, in fact, a scaled version of the gradient ∇F.

Lemma 3 *For $m \in \mathcal{C}_i$, and $w \in \mathcal{F}_i$*

$$g_{m,w}(t, t+\delta) = \delta \left[\frac{1}{r_i} \frac{\partial F}{\partial x_{i,w}} + M_{m,w} \right] + O\left(\delta^2\right)$$

where $\mathbb{E}[M_{m,w}] = 0$ and $\mathbb{E}[|M_{m,w}|^2] < \infty$.

Proof Observe that, for any $m \in \mathcal{C}_i$, and any w s.t. $d_{i,w} > 0$, $T_{m,w}(t)$ is exponentially distributed with mean $1/\rho_{i,w}$; in other words, it follows the same distribution as $\hat{Y}_{i,w}$. By Little's theorem, for any $m \in \mathcal{C}_i$ s.t. $y_{m,w} = 0$, $\mathbb{E}[n_{m,w}(t)] = \frac{d_{i,w}}{\rho_{i,w}}$. Moreover, $n_{m,w}(t)$ is independent of $T_{m,w}(t)$. This is because $T_{m,w}(t)$ is updated at the last encounter with a user that stores w, at which point any pending requests for w are served and $n_{m,w}$ becomes zero; such events constitute renewal epochs of the process $\{n_{m,w}(t)\}$. As $n_{m,w}(t)$ reflects the requests accumulated since that last encounter, it is independent of $T_{m,w}(t)$.

Recall, from (14), that

$$g_{m,w}(t,t+\delta) = \begin{cases} q_{m',w}(t), \text{ if } m, m' \text{ meet in } (t,t+\delta] \\ U_{i,w}(0) - U_{i,w}(T_{m,w}(t)), \text{ if } m \\ \qquad \text{requests } w \text{ in } (t,t+\delta] \\ 0, \text{ o.w.} \end{cases}$$

Hence, the above observations give us that, for any $m \in \mathcal{C}_i$,

$$\begin{aligned}
\mathbb{E}[g_{m,w}(t,t+\delta)] &= O\left(\delta^2\right) + \delta d_{i,w} \mathbb{E}[U_{i,w}(0) - U_{i,w}(T_{m,w}(t))] \\
&\quad + \delta \sum_j \sum_{m' \in \mathcal{C}_j} \frac{\lambda_{i,j}}{r_j N} \mathbb{1}_{y_{m',w}=0} \mathbb{E}[n_{m',w}] \mathbb{E}[-T_{m',w}(t) U'_{j,w}(T_{m',w}(t))] \\
&= O\left(\delta^2\right) + \delta d_{i,w} \mathbb{E}[U_{i,w}(0) - U_{i,w}(\hat{Y}_{i,w}))] \\
&\quad + \delta \sum_j \sum_{m' \in \mathcal{C}_j} \frac{\lambda_{i,j}}{r_j N} \mathbb{1}_{y_{m',w}=0} \frac{d_{j,w}}{\rho_{j,w}} \mathbb{E}[-\hat{Y}_{j,w} U'_{j,w}(\hat{Y}_{j,w})] \\
&= O\left(\delta^2\right) + \delta d_{i,w} \mathbb{E}[U_{i,w}(0) - U_{i,w}(\hat{Y}_{i,w}))] \\
&+ \quad \delta \sum_j \lambda_{i,j} \frac{d_{j,w}}{\rho_{j,w}} \mathbb{E}[-\hat{Y}_{j,w} U'_{j,w}(\hat{Y}_{j,w})] \sum_{m' \in \mathcal{C}_j} \frac{\mathbb{1}_{y_{m',w}=0}}{r_j N} \\
&\overset{(2)}{=} O\left(\delta^2\right) + \delta d_{i,w} \mathbb{E}[U_{i,w}(0) - U_{i,w}(\hat{Y}_{i,w}))] \\
&\quad + \delta \frac{1}{r_i} \sum_j r_j \lambda_{j,i} (1 - x_{j,w}) \frac{d_{j,w}}{\rho_{j,w}} \mathbb{E}[-\hat{Y}_{j,w} U'_{j,w}(\hat{Y}_{j,w})] \\
&= \delta \frac{1}{r_i} \frac{\partial F}{\partial x_{i,w}} + O\left(\delta^2\right).
\end{aligned}$$

The number of pending requests for w and $m \in \mathcal{C}_i$, *i.e.*, $n_{m,w}(t)$, evolves as a queue with Poisson arrivals with rate $d_{i,w}$ and a single server with an exponential service time with parameter $\rho_{i,w}$ and with batch departures. In such a

queueing system, the steady state distribution of the size is geometric with parameter $\rho_{i,w}(\rho_{i,w}+d_{i,w})^{-1}$. Hence, in steady state $\mathbb{E}[(n_{m,w})^2] = \frac{2d_{i,w}^2 - d_{i,w}\rho_{i,w}}{\rho_{i,w}^2}$. The second moment of $g_{m,w}$ can thus be written as

$$\begin{aligned}
\mathbb{E}[g_{m,w}^2(t,t+\delta)] &\leq O\left(\delta^2\right) + \delta d_{i,w}\mathbb{E}[(U_{i,w}(0) - U_{i,w}(T_{m,w}(t)))^2] \\
&\quad + \delta \sum_j \sum_{m' \in \mathcal{C}_j} \frac{\lambda_{i,j}}{r_j N} \mathbb{1}_{y_{m',w}=0} \mathbb{E}[n_{m',w}^2]\mathbb{E}[\left(T_{m',w}(t)U'_{j,w}(T_{m',w}(t))\right)^2] \\
&= O\left(\delta^2\right) + \delta d_{i,w}\mathbb{E}\left[|U_{i,w}(0) - U_{i,w}(\hat{Y}_{i,w}))|^2\right] \\
&\quad + \delta O\left(\sum_j \frac{1}{\rho_{j,w}^2} \int_0^\infty (sU'_{j,w}(s))^2 \rho_{i,w} e^{-\rho_{i,w}s} ds\right)
\end{aligned}$$

Recall that $\rho_{i,w} \geq \mu_i > 0$. Moreover, note that if $|f(t)| = o\left(e^{-ct}\right)$ for all $c > 0$, then so is $|f(t)|^2$. Hence, Assumption 1, implies that $\mathbb{E}[(U_{i,w}(0) - U_{i,w}(T_{m,w}(t)))^2]$ is finite. On the other hand, the summands in the quantity summed over j is of the order of $\frac{1}{\rho_{j,w}} \int_0^\infty (sU'_{j,w}(s))^2 e^{-\mu_i s} ds$. The quantity $\int_0^\infty (sU'_{j,w}(s))^2 e^{-\mu_i s} ds$ is bounded by Assumption 1 and (23), and $1/\rho_{j,w}$ is bounded by $1/\mu_j$. The lemma therefore follows. □

We now turn our attention to $h_{m,w}$, which characterizes the infinitesimal change of the caching policy at m.

Lemma 4 *For $m \in \mathcal{C}_i$,*

$$h_{m,w}(t,t+\delta) = \delta \frac{\mu_i \alpha(N)}{r_i} \left\{ \Delta_{m,w}(\mathbf{v}_m, \mathbf{y}_m) + M'_{m,w} \right\} + O(\delta^2)$$

where $M'_{m,w}$ a r.v. with $\mathbb{E}[M'_{m,w}] = 0$ and $\mathbb{E}[|M'_{i,w}|^2] < 1$, and $\Delta_{m,w}(\mathbf{v}_m, \mathbf{y}_m)$ as in (26)*.*

Proof It is easy to see from (25) and the definition of the contact process among users and the infrastructure that

$$\begin{aligned}
\mathbb{E}[h_{m,w}(t,t+\delta)] &= O\left(\delta^2\right) + 0 \cdot (1 - \frac{\mu_i \alpha(N)}{r_i}\delta) + \frac{\mu_i \alpha(N)}{r_i}\delta \cdot \\
&\left[\mathbb{1}_{v_{m,w} > \tau_i(\mathbf{v}_m) \wedge y_{m,w}=0} - \mathbb{1}_{v_{m,w} < \tau_i(\mathbf{v}_m) \wedge y_{m,w}=1} + \mathbb{1}_{v_{m,w}=\tau_i(\mathbf{v}_m)} z_{m,w}(\mathbf{v}_m, \mathbf{y}_m)\right]
\end{aligned}$$

The boundedness of the variance follows easily from the fact that, conditioned on at most one contact with the infrastructure taking place, $|h_{m,w}(t,t+\delta)| \leq 1$. □

5.4 Objective Drift under Timescale Separation

In this section, we show the following three crucial facts. First, under time-scale separation, the vote processes of users in a class follow closely the gradient of the social welfare with respect to replication ratios in this class. Second, the

drift of the social welfare can be described succinctly in terms of (a) the drift of a system in which *all vote processes in a class* are equal to the above gradient and (b) an "error" term, which, under the time-scale separation, must be small. Finally, and most importantly, the drift under the above evolution is positive and becomes small *if and only if* the system is close enough to a maximizer of the social welfare; we will make use of this fact in the next section, to establish the proof of Theorem 1.

To begin with, Lemma 3 and Equations (13) and (14) immediately imply that the evolution of $\mathbf{v}_m$ can be described by the following stochastic difference equation:

$$v_{m,w}(t+\delta) = (1-\beta\delta)v_{m,w}(t)+\beta\delta\left[\frac{1}{r_i}\frac{\partial F(\mathbf{X})}{\partial x_{i,w}}+M_{m,w}+O(\delta)\right] \tag{28}$$

where $\mathbb{E}[M_{m,w}] = 0$ and $\mathbb{E}[M_{m,w}^2] < \infty$ second moments are bounded for all m and w. Similarly, Lemma 4 and Equations (4), (24) and (25) imply that the evolution of $\mathbf{x}_i$ can be described by the following stochastic difference equation:

$$x_{i,w}(t+\delta) = x_{i,w}(t)+\frac{\mu_i\alpha}{r_i}\delta\left[\frac{1}{r_iN}\sum_{m\in\mathcal{C}_i}\Delta_{m,w}(\mathbf{v}_m,\mathbf{y}_m)+M'_{i,w}+O(\delta)\right] \tag{29}$$

where $\mathbb{E}[M'_{i,w}] = 0$ and $\mathbb{E}[(M'_{i,w})^2] < \infty$ for all i and w.

Hence, the evolution of $\mathbf{v}_m$, $m \in \mathcal{C}_i$, occurs at a time-scale of the order of β, while the evolution of $\mathbf{x}_i$ occurs at a time-scale of the order of $\frac{\mu_i\alpha}{r_i}$. The following lemma states that if the above time-scales are separated, then, for large t, the vote processes follow closely the partial derivatives $\partial F/\partial x_{i,w}$:

Lemma 5 *For all i, for all $m \in \mathcal{C}_i$, and for all $w \in \mathcal{F}_i$,*

$$\limsup_{t\to\infty}\mathbb{E}[|v_{m,w}(t)-\frac{1}{r_i}\frac{\partial F(\mathbf{X}(t))}{\partial x_{i,w}}|^2] = O\left(\frac{\mu_i\alpha(N)}{r_i\beta(N)}\right)$$

Proof Fix $\delta(N) > 0$ to be an arbitrary function s.t.

$$\lim_{N\to\infty}\delta(N)\max(1,\beta(N)) = 0. \tag{30}$$

For $k \in \mathbb{N}$, take $v_{m,w}^k = v_{m,w}(\delta(N)\cdot k)$ and $\mathbf{X}^k = \mathbf{X}(\delta(N)\cdot k)$. Then, by (28) and (29), the evolution of $\{v_{m,w}^k, \mathbf{X}^k\}_{k\in\mathbb{N}}$ can be described as:

$$\begin{cases} v_{m,w}^{k+1} = v_{m,w}^k+\beta(N)\delta(N)\left[\frac{1}{r_i}\frac{\partial F(\mathbf{X})}{\partial x_{i,w}}-v_{m,w}^k+O(\delta(N))+M_{m,w}\right] \\ x_{j,w}^{k+1} = x_{j,w}^k, \text{ for } j \text{ s.t. } w \notin \mathcal{F}_j \\ x_{j,w}^{k+1} = x_{j,w}^k+\beta(N)\delta(N)\left[\epsilon(N)+M'_{j,w}(N)\right], \text{ for } j \text{ s.t. } w \in \mathcal{F}_j \end{cases}$$

where $\epsilon(N) = O\left(\frac{\mu_i\alpha(N)}{r_i\beta(N)}\right)$ and $M'_{i,w}(N) = \frac{\mu_i\alpha(N)}{r_i\beta(N)}M'_{i,w}$. Let $\lambda(\mathbf{X}) = \frac{1}{r_i}\frac{\partial F(\mathbf{X})}{\partial x_{i,w}}$ be the stationary point of the ODE $\dot{v}_{m,w} = \frac{1}{r_i}\frac{\partial F(\mathbf{X})}{\partial x_{i,w}} - v_{m,w}$. Note that, by Lemma 2, $\lambda(\mathbf{X})$ is Lipschitz. From Chapter 2, page 112 of Borkar [3],

$$\limsup_{k\to\infty}\mathbb{E}[|v_{m,w}^k-\lambda(\mathbf{X}^k)|^2] = O(\beta(N)\delta(N))+O\left(\frac{\mu_i\alpha(N)}{r_i\beta(N)}\right).$$

The lemma follows as δ can be any function s.t. (30) holds. □

Consider now the processes $\{\mathbf{X}(k)\}_{k\in\mathbb{N}}$, $\{v_{m,w}(k)\}_{k\in\mathbb{N}}$ defined *at the k-th epoch of a contact with the infrastructure* and let

$$\epsilon(k) = \max_{m\in\mathcal{C}_i,w} |v_{m,w}(k) - \frac{1}{r_i}\frac{\partial F(\mathbf{X}(k))}{\partial x_{i,w}}|. \tag{31}$$

This error amounts to the maximum difference between the vote process and the gradient within a class. Using the above lemma, it is easy to characterize its maginitude in terms of α and β.

Lemma 6

$$\limsup_{k\to\infty} \mathbb{E}[|\epsilon(k)|^2] = O\left(\frac{\mu_i N\alpha(N)}{r_i\beta(N)}\right). \tag{32}$$

Proof By Lemma 5,

$$\begin{aligned}\limsup_{k\to\infty} \mathbb{E}[|\epsilon(k)|^2] &= \limsup_{k\to\infty} \mathbb{E}[\sup_{m,w} |v_{m,w}(k) - \frac{1}{r_i}\frac{\partial F(\mathbf{X}(k))}{\partial x_{i,w}}|^2] \\ &\leq \limsup_{k\to\infty} \sum_{m,w} \mathbb{E}[|v_{m,w}(k) - \frac{1}{r_i}\frac{\partial F(\mathbf{X}(k))}{\partial x_{i,w}}|^2] = O\left(\frac{\mu_i N\alpha(N)}{r_i\beta(N)}\right). \qquad \square\end{aligned}$$

Finally, the following lemma states that the evolution of $F(\mathbf{X}(k))$ is well approximated by the evolution of a system in which (a) all users in the same class $\mathcal{C}_i$ have the same vote vector and (b) this vector equals $[\frac{1}{r_i}\frac{\partial F(\mathbf{X}(k))}{\partial x_{i,w}}]_{w\in\mathcal{F}_i}$. The error between this and the true evolution of $F(\mathbf{X}(k))$ can be bounded in terms of $\epsilon(k)$—which, by Lemma 6, is small when $\alpha(N)$ and $\beta(N)$ are well separated.

Lemma 7 *Let* $u_{i,w} \equiv \frac{1}{r_i}\frac{\partial F(\mathbf{X}(k))}{\partial x_{i,w}}$*, for all* $w \in \mathcal{F}_i$*. Then*

$$F(\mathbf{X}(k+1)) - F(\mathbf{X}(k)) = \frac{1}{N}\left[\sum_i \Delta_i(\mathbf{u}_i, \mathbf{x}_i) + M + O\left(\epsilon(k)\right) + O\left(\frac{1}{N}\right)\right]$$

where $\mathbb{E}[M] = 0$*,* $\mathbb{E}[|M|^2] \leq \infty$ *and*

$$\Delta_i(\mathbf{u}_i, \mathbf{x}_i) = \tilde{\mu}_i\left\{\sum_{\substack{w\in\mathcal{F}_i:\\ u_{i,w}>\tau_i(\mathbf{u}_i)}} (u_{i,w}-\tau_i(\mathbf{u}_i))(1-x_{i,w}) + \sum_{\substack{w\in\mathcal{F}_i:\\ u_{i,w}<\tau_i(\mathbf{u}_i)}} (u_{i,w}-\tau_i(\mathbf{u}_i))x_{i,w}\right\} \tag{33}$$

with $\tilde{\mu}_i = \mu_i / \sum_j \mu_j$*.*

Proof Conditioned on an encounter with the infrastructure taking place, this occurs at a given $m \in \mathcal{C}_i$ with probability $\frac{\tilde{\mu}_i}{r_i N}$. Hence, $\mathbf{X}(k)$ can be described as

$$x_{i,w}(k+1) = x_{i,w}(k) + \frac{1}{r_i N}\left\{\sum_{m\in\mathcal{C}_i} \frac{\tilde{\mu}_i}{r_i N}\Delta_{m,w}(\mathbf{v}_m, \mathbf{y}_m) + M''_{i,w}\right\}$$

for i, w s.t. $w \in \mathcal{F}_i$, and $x_{i,w}(k+1) = x_{i,w}(k)$, for i, w, s.t. $w \notin \mathcal{F}_i$. Note that since at each epoch $x_{i,w}$ changes by at most $\frac{1}{r_i N}$, $\|\mathbf{X}(k+1)-\mathbf{X}(k)\|_\infty = O\left(\frac{1}{N}\right)$. From the mean value theorem

$$F(\mathbf{X}(k+1))-F(\mathbf{X}(k))=\sum_{i,w}\frac{\partial F(\boldsymbol{\Xi})}{\partial x_{i,w}}(x_{i,w}(k+1)-x_{i,w}(k))$$

where $\boldsymbol{\Xi} = (1-\delta)\mathbf{X}(k+1)+\delta\mathbf{X}(k)$, for some $0 \le \delta \le 1$. By Lemma 2, $\frac{\partial F}{\partial x_{i,w}}$ is Lipschitz continuous, so we have that $|\frac{\partial F(\boldsymbol{\Xi})}{\partial x_{i,w}} - \frac{\partial F(\mathbf{X}(k))}{\partial x_{i,w}}| \le K\|\boldsymbol{\Xi} - \mathbf{X}(k)\|_\infty \le K\|\mathbf{X}(k+1) - \mathbf{X}(k)\|_\infty = O\left(\frac{1}{N}\right)$ Hence, for $\mathbf{u}_i \equiv [\frac{1}{r_i}\frac{\partial F(\mathbf{X}(k))}{\partial x_{i,w}}]_{w\in\mathcal{F}_i}$,

$$\begin{aligned}
&F(\mathbf{X}(k+1)) - F(\mathbf{X}(k)) = \\
&= \sum_{i,w}\left(\frac{\partial F(\mathbf{X}(k))}{\partial x_{i,w}}+O\left(\frac{1}{N}\right)\right)(x_{i,w}(k+1) - x_{i,w}(k)) \\
&= \sum_{i,w\in\mathcal{F}_i}\frac{u_{i,w}}{N}\left\{\sum_{m\in\mathcal{C}_i}\frac{\tilde{\mu}_i}{r_i N}\Delta_{m,w}(\mathbf{v}_m,\mathbf{y}_m)+M''_{i,w}\right\}+O\left(\frac{1}{N^2}\right) \\
&= \frac{1}{N}\sum_{i,w\in\mathcal{F}_i}\tilde{\mu}_i u_{i,w}\sum_{m\in\mathcal{C}_i}\frac{\Delta_{m,w}(\mathbf{v}_m,\mathbf{y}_m)}{r_i N}+\frac{M}{N}+O\left(\frac{1}{N^2}\right) \qquad (34)
\end{aligned}$$

where $\mathbb{E}[M] = 0$ and $\mathbb{E}[|M|^2] < \infty$. On the other hand, for any i,

$$\begin{aligned}
&\sum_{w\in\mathcal{F}_i}\tilde{\mu}_i u_{i,w}\left\{\sum_{m\in\mathcal{C}_i}\frac{\Delta_{m,w}(\mathbf{v}_m,\mathbf{y}_m)}{r_i N}\right\} = \\
&\overset{(26)}{=} \frac{\tilde{\mu}_i}{r_i N}\sum_{m\in\mathcal{C}_i}\left\{\sum_{\substack{w\in\mathcal{F}_i: v_{m,w}>\tau_i(\mathbf{v}_m)\\ \wedge y_{m,w}=0}} u_{i,w} - \sum_{\substack{w\in\mathcal{F}_i: v_{m,w}<\tau_i(\mathbf{v}_m)\\ \wedge y_{m,w}=1}} u_{i,w}\right. \\
&\qquad \left. + \sum_{w\in\mathcal{F}_i: v_{m,w}=\tau_i(\mathbf{v}_m)} u_{i,w} z_{m,w}(\mathbf{v}_m,\mathbf{y}_m)\right\}
\end{aligned}$$

Note that τ_i is Lipschitz continuous. In particular

$$|\tau_i(\mathbf{v}_m) - \tau_i(\mathbf{u}_i)| \le \|\mathbf{v}_m - \mathbf{u}_i\|_\infty \overset{(31)}{\le} \epsilon(k). \qquad (35)$$

Hence, for every $m \in \mathcal{C}_i$ and every $w \in \mathcal{F}_i$ s.t. $v_{m,w} = \tau_i(\mathbf{v}_m)$, we have that

$$|u_{i,w} - \tau_i(\mathbf{u}_i)| \le |u_{i,w} - v_{m,w}| + |\tau_i(\mathbf{v}_m) - \tau_i(\mathbf{u}_i)| \le 2\epsilon(k).$$

This, along with (27), implies that

$$\begin{aligned}
&\sum_{w\in\mathcal{F}_i}\tilde{\mu}_i u_{i,w}\left\{\sum_{m\in\mathcal{C}_i}\frac{\Delta_{m,w}(\mathbf{v}_m,\mathbf{y}_m)}{r_i N}\right\} = \\
&= \frac{\tilde{\mu}_i}{r_i N}\sum_{m\in\mathcal{C}_i}\left\{\sum_{\substack{w\in\mathcal{F}_i: v_{m,w}>\tau_i(\mathbf{v}_m)\\ \wedge y_{m,w}=0}}(u_{i,w}-\tau_i(\mathbf{u}_i)) + \sum_{\substack{w\in\mathcal{F}_i: v_{m,w}<\tau_i(\mathbf{v}_m)\\ \wedge y_{m,w}=1}}(\tau_i(\mathbf{u}_i)-u_{i,w})\right\} + O\left(2\tilde{\mu}_i W\epsilon(k)\right)
\end{aligned}$$

$$= \tilde{\mu}_i \Bigg\{ \sum_{\substack{w\in\mathcal{F}_i:\\ u_{i,w}>\tau_i(\mathbf{u}_i)}} (u_{i,w}-\tau_i(\mathbf{u}_i))(1-x_{i,w}) + \sum_{\substack{w\in\mathcal{F}_i:\\ u_{i,w}<\tau_i(\mathbf{u}_i)}} (u_{i,w}-\tau_i(\mathbf{u}_i))x_{i,w} \Bigg\}$$
$$+ E_1 + E_2 + O\left(2\tilde{\mu}_i W\epsilon(k)\right) \tag{36}$$

where

$$E_1 = \frac{\tilde{\mu}_i}{r_i N} \sum_{m\in\mathcal{C}_i} \sum_{w\in\mathcal{F}_i} [\mathbb{1}_{v_{m,w}>\tau_i(\mathbf{v}_m)\wedge y_{m,w}=0} - \mathbb{1}_{u_{i,w}>\tau_i(\mathbf{u}_i)\wedge y_{m,w}=0}](u_{i,w}-\tau_i(\mathbf{u}_i))$$
$$E_2 = \frac{\tilde{\mu}_i}{r_i N} \sum_{m\in\mathcal{C}_i} \sum_{w\in\mathcal{F}_i} [\mathbb{1}_{v_{i,w}<\tau_i(\mathbf{v}_i)\wedge y_{m,w}=1} - \mathbb{1}_{u_{i,w}<\tau_i(\mathbf{u}_i)\wedge y_{m,w}=1}](\tau_i(\mathbf{u}_i)-u_{i,w})$$

By (31) and (35), for every $w \in \mathcal{F}_i$ s.t. $u_{i,w} > \tau_i(\mathbf{u}_i) + 2\epsilon(k)$, $v_{m,w} > \tau_i(\mathbf{v}_m)$, while for every $w \in \mathcal{F}_i$ s.t. $u_{i,w} < \tau_i(\mathbf{u}_i) - 2\epsilon(k)$, $v_{i,m} < \tau_i(\mathbf{v}_m)$. This implies that the indicator functions in E_1 and E_2 (expressed in terms of $\mathbf{v}_m$ and $\mathbf{u}_i$) may only differ for $w \in \mathcal{F}_i$ such that $|u_{i,w} - \tau_i(\mathbf{u}_i)| \leq 2\epsilon(k)$. As such

$$|E_1| \leq \frac{\tilde{\mu}_i}{r_i N} \sum_{m\in\mathcal{C}_i} \sum_{w\in\mathcal{F}_i:|u_{i,w}-\tau_i(\mathbf{u}_i)|\leq 2\epsilon(k)} |u_{i,w} - \tau_i(\mathbf{u}_i)| \leq 2\tilde{\mu}_i W\epsilon(k)$$

and the same can be stated about E_2. The lemma therefore follows from (34) and (36). □

Lemma 7 implies that the mean drift of $F(\mathbf{X}(k))$ is determined by the quantity

$$\sum_i \Delta_i(\mathbf{u}_i, \mathbf{x}_i) \geq 0,$$

which is always non-negative. This indicates that, in expectation, $F(\mathbf{X}(k))$ will increase. By considering the Karush Kuhn Tucker (KKT) conditions of the optimization problem (10), one can show that the above mean drift is zero if and only if $\mathbf{X}$ is a maximizer of F. In fact, a stronger statement is true: if, for some $\mathbf{X}(k)$, the mean drift $\sum_i \Delta_i \mathbf{u}_i, \mathbf{x}_i$ is small, then $F(\mathbf{X}(k))$ is guaranteed to be close to the maximum value of F in D:

Lemma 8 *Consider a $\mathbf{X}^* \in D$ and denote by*

$$\mathbf{u}_i^* \equiv \left[r_i^{-1} \frac{\partial F(\mathbf{X}^*)}{\partial x_{i,w}} \right]_{w\in\mathcal{F}_i}.$$

If $\sum_i \Delta_i(\mathbf{u}_i^*, \mathbf{x}_i^*) \leq \epsilon$, for some $\epsilon > 0$, then

$$|F(\mathbf{X}^*) - \sup_{\mathbf{X}\in D} F(\mathbf{X})| \leq \epsilon \max_i \frac{r_i}{\tilde{\mu}_i}.$$

Proof The Lagrangian of (10) is

$$\mathcal{L}(\mathbf{X},\boldsymbol{\xi},\boldsymbol{\Psi},\boldsymbol{\Phi}) = F(\mathbf{X}) + \sum_i \xi_i(c_i' - \sum_{w\in\mathcal{F}_i} x_{i,w}) + \sum_{i,w\in\mathcal{F}_i} \psi_{i,w}(1-x_{i,w}) + \sum_{i,w\in\mathcal{F}_i} \phi_{i,w}x_{i,w} \tag{37}$$

Therefore, the KKT conditions of (10) are

$$\sum_{w\in\mathcal{F}_i} x_{i,w} \le c_i',\ \xi_i(\sum_{w\in\mathcal{F}_i} x_{i,w} - c_i') = 0,\ \xi_i \ge 0,\quad \forall i,\ \text{and}$$

$$\begin{cases} 0 \le x_{i,w} \le 1,\ \psi_{i,w} \ge 0,\ \psi_{i,w}(x_{i,w}-1) = 0, \\ \phi_{i,w} \ge 0,\ \phi_{i,w}x_{i,w} = 0 & \forall i, w \in \mathcal{F}_i. \\ \frac{\partial F}{\partial x_{i,w}} - \xi_i - \psi_{i,w} + \phi_{i,w} = 0, \end{cases}$$

Let $f(\boldsymbol{\xi},\boldsymbol{\Psi},\boldsymbol{\Phi}) = \sup_{\mathbf{X}\in D} \mathcal{L}(\mathbf{X},\boldsymbol{\xi},\boldsymbol{\Psi},\boldsymbol{\Phi})$ and define

$$\begin{aligned}
&\xi_i^* = r_i\tau_i(\mathbf{u}_i^*), && \forall i \\
&\psi_{i,w}^* = r_i(u_{i,w}^* - \tau_i(\mathbf{u}_i^*)), \phi_{i,w}^* = 0, && \forall w \in \mathcal{F}_i \text{ s.t } u_{i,w}^* > \tau_i(\mathbf{u}_i^*) \\
&\psi_{i,w}^* = 0, \phi_{i,w}^* = r_i(\tau_i(\mathbf{u}_i^*) - u_{i,w}^*), && \forall w \in \mathcal{F}_i \text{ s.t } u_{i,w}^* < \tau_i(\mathbf{u}_i^*) \\
&\psi_{i,w}^* = 0, \phi_{i,w}^* = 0, && \forall w \in \mathcal{F}_i \text{ s.t } u_{i,w}^* = \tau_i(\mathbf{u}_i^*)
\end{aligned}$$

Then,

$$f(\boldsymbol{\xi}^*,\boldsymbol{\Psi}^*,\boldsymbol{\Phi}^*) = L(\mathbf{X}^*,\boldsymbol{\xi}^*,\boldsymbol{\Psi}^*,\boldsymbol{\Phi}^*). \tag{38}$$

To see this, observe that,

$$\frac{\partial\mathcal{L}(\mathbf{X}^*,\boldsymbol{\xi}^*,\boldsymbol{\Psi}^*,\boldsymbol{\Phi}^*)}{\partial x_{i,w}} = \frac{\partial F(\mathbf{X}^*)}{\partial x_{i,w}} - \xi_i^* - \psi_{i,w}^* + \phi_{i,w}^* = 0$$

by the definition of $\boldsymbol{\xi}^*, \boldsymbol{\Psi}^*, \boldsymbol{\Phi}^*$ and $\mathbf{u}_i^*$. Moreover, since F is concave in $\mathbf{X} \in D$, so is $\mathcal{L}$. The above imply that the supremum of $\mathcal{L}(X,\boldsymbol{\xi}^*,\boldsymbol{\Psi}^*,\boldsymbol{\Phi}^*)$ in D is attained at $\mathbf{X}^*$, which yields (38). The max-min principle implies that

$$\sup_{\mathbf{X}\in D} \inf_{\substack{\boldsymbol{\xi}\ge 0,\\ \boldsymbol{\Psi}\ge 0,\boldsymbol{\Phi}\ge 0}} \mathcal{L}(\mathbf{X},\boldsymbol{\xi},\boldsymbol{\Psi},\boldsymbol{\Phi}) \le \inf_{\substack{\boldsymbol{\xi}\ge 0,\\ \boldsymbol{\Psi}\ge 0,\boldsymbol{\Phi}\ge 0}} \sup_{\mathbf{X}\in D} \mathcal{L}(\mathbf{X},\boldsymbol{\xi},\boldsymbol{\Psi},\boldsymbol{\Phi})$$

In fact, because F is concave the above inequality is an equality. In any case however,

$$\sup_{\mathbf{X}\in D} \inf_{\substack{\boldsymbol{\xi}\ge 0,\\ \boldsymbol{\Psi}\ge 0,\boldsymbol{\Phi}\ge 0}} \mathcal{L}(\mathbf{X},\boldsymbol{\xi},\boldsymbol{\Psi},\boldsymbol{\Phi}) = \sup_{\mathbf{X}\in D} F(\mathbf{X})$$

as all coefficients of $\boldsymbol{\xi}, \boldsymbol{\Psi}$, and $\boldsymbol{\Phi}$ in (37) are positive, and

$$\inf_{\substack{\boldsymbol{\xi}\ge 0,\\ \boldsymbol{\Psi}\ge 0,\boldsymbol{\Phi}\ge 0}} \sup_{\mathbf{X}\in D} \mathcal{L}(\mathbf{X},\boldsymbol{\xi},\boldsymbol{\Psi},\boldsymbol{\Phi}) \le f(\boldsymbol{\xi},\boldsymbol{\Psi},\boldsymbol{\Phi})$$

for all non-negative $\boldsymbol{\xi}, \boldsymbol{\Psi}$, and $\boldsymbol{\Phi}$. We therefore have that

$$\begin{aligned}
&\sup_{\mathbf{X}\in D} F(\mathbf{X}) \leq f(\boldsymbol{\xi}^*, \boldsymbol{\Psi}^*, \boldsymbol{\Phi}^*) \overset{(38)}{=} L(\mathbf{X}^*, \boldsymbol{\xi}^*, \boldsymbol{\Psi}^*, \boldsymbol{\Phi}^*) \\
&= F(\mathbf{X}^*) + \sum_i \xi_i^*(c_i - \sum_w x_{i,w}^*) + \sum_{i,w} \psi_{i,w}^*(1 - x_{i,w}^*) + \sum_{i,w} \phi_{i,w}^* x_{i,w}^* \\
&= F(\mathbf{X}^*) + \sum_i r_i \Big[\sum_{w: u_{i,w}^* > \tau_i(\mathbf{u}_i^*)} \left(u_{i,w}^* - \tau_i(\mathbf{u}_i^*)\right)(1 - x_{i,w}^*) + \sum_{w: u_{i,w}^* < \tau_i(\mathbf{u}_i^*)} \left(\tau_i(\mathbf{u}_i^*) - u_{i,w}^*\right) x_{i,w}^* \Big] \\
&\overset{(33)}{\leq} F(\mathbf{X}^*) + \epsilon \max_i \frac{r_i}{\tilde{\mu}_i}
\end{aligned}$$

and the lemma follows. □

5.5 Convergence to Maximizer of Social Welfare

We are now ready to prove Theorem 1. In short, from Lemma 8 we know that a system that evolves under vote processes that equal the gradient of the objective will come to rest at a maximizer of the social welfare. On the other hand, from Lemma 7, the evolution of our system is off from the above dynamics by an $\epsilon(k)$ factor; under time-scale separation however, this quantity is small, so we can guarantee that our system will converge close to a maximizer of (10).

More formally, from Lemma 7,

$$\begin{aligned}
&\mathbb{E}[F(\mathbf{X}(k+1))] - \mathbb{E}[F(\mathbf{X}(k))] = \\
&\quad = \frac{1}{N}\left[\mathbb{E}[\sum_i \Delta_i(\mathbf{u}_i(k), \mathbf{x}_i(k))] + O\left(\mathbb{E}[\epsilon(k)]\right) + O\left(\frac{1}{N}\right)\right]
\end{aligned}$$

In steady state, we have that $\lim_{k\to\infty} \mathbb{E}[F(\mathbf{X}(k+1))] = \lim_{k\to\infty} \mathbb{E}[F(\mathbf{X}(k))]$. As a result,

$$\begin{aligned}
\limsup_{k\to\infty} \mathbb{E}[\sum_i \Delta_i(\mathbf{u}_i, \mathbf{x}_i)] &= \limsup_{k\to\infty} O\left(\mathbb{E}[|\epsilon(k)|]\right) + O\left(\frac{1}{N}\right) \\
&\overset{(31)}{=} O\left(\sqrt{\frac{N\alpha(N)}{\beta(N)}}\right) + O\left(\frac{1}{N}\right)
\end{aligned}$$

Observe that, from Lemma 8, for any $\epsilon > 0$,

$$\mathbf{P}\left(|F(\mathbf{X}) - \sup_{\mathbf{X}\in D} F(\mathbf{X})| > \epsilon\right) \leq \mathbf{P}\left(\sum_i \Delta_i(\mathbf{u}_i, \mathbf{x}_i) > \epsilon \min \frac{\tilde{\mu}_i}{r_i}\right).$$

By Markov's inequality, the above implies that the steady state distribution of $\mathbf{x}_i$ is such that for any $\epsilon > 0$,

$$\limsup_{t\to\infty} \mathbf{P}\left(|F(\mathbf{X}(t)) - \sup_{\mathbf{X}\in D} F(\mathbf{X})| > \epsilon\right) = \frac{1}{\epsilon}\left(O\left(\sqrt{\frac{N\alpha(N)}{\beta(N)}}\right) + O\left(\frac{1}{N}\right)\right).$$

and the theorem therefore follows by taking ϵ as in (17). □

6 Proof of Theorem 2

The theorem will be established by exhibiting a Lyapunov function for the dynamics under consideration. To this end, we introduce some auxiliary functions: we define $J(\mathbf{v})$ as

$$J(\mathbf{v}) := \max_{\mathbf{x}\in C} \langle \mathbf{x}, \mathbf{v} \rangle. \tag{39}$$

It then follows from Theorem 23.5, p. 218 in Rockafellar [14] that

$$G(\mathbf{v}) = \nabla J(\mathbf{v}). \tag{40}$$

We also introduce the function $F^*(\mathbf{v})$, defined as

$$F^*(\mathbf{v}) := \inf_{\mathbf{x}\in C} \langle \mathbf{x}, \mathbf{v} \rangle - F(\mathbf{x}). \tag{41}$$

We then define the candidate Lyapunov function $L(\mathbf{x}, \mathbf{v})$ as

$$L(\mathbf{x}, \mathbf{v}) := J(\mathbf{v}) - F^*(\mathbf{v}) - \frac{\beta}{\alpha} \left(F(\mathbf{x}) + F^*(\mathbf{v}) - \langle \mathbf{x}, \mathbf{v} \rangle \right). \tag{42}$$

Upon taking time derivatives, in view of (18a),(18b) and (40), one obtains after term cancellation that

$$\frac{d}{dt} L(\mathbf{x}, \mathbf{v}) = \beta \left(\frac{\beta}{\alpha} + 1 \right) \langle \mathbf{x} - \nabla F^*(\mathbf{v}), \nabla F(\mathbf{x}) - \mathbf{v} \rangle.$$

Setting $\mathbf{y} = \nabla F^*(\mathbf{v})$, it readily follows from the theory of convex function duality [14] that $\mathbf{v} = \nabla F(\mathbf{y})$. Thus the time derivative of $L(\mathbf{x}, \mathbf{v})$ is proportional to $\langle \mathbf{x} - \mathbf{y}, \nabla F(\mathbf{x}) - \nabla F(\mathbf{y}) \rangle$. Now strict concavity of F entails that this scalar product is non-positive, and equals zero if and only if $\mathbf{x} = \mathbf{y}$, or equivalently $\mathbf{v} = \nabla F(\mathbf{x})$.

Thus $L(\mathbf{x}, \mathbf{v})$ is strictly decreasing unless $\mathbf{v} = \nabla F(\mathbf{x})$. We further argue that it is strictly decreasing also whenever $\mathbf{x} \neq G(\mathbf{v})$. To see this, consider a pair of points $(\mathbf{x}, \mathbf{v})$ such that $\mathbf{v} = \nabla F(\mathbf{x})$, but $\mathbf{x} \neq G(\mathbf{v})$. Then after some small time $\epsilon > 0$, $\|\mathbf{v}(\epsilon) - \mathbf{v}\|$ is of order $o(\epsilon)$ since the time derivative $(d/dt)(\mathbf{v})$ is of order $o(1)$, while $\|\mathbf{x}(\epsilon) - \mathbf{x}\|$ is of order $\Omega(\epsilon)$ since the time derivative $(d/dt)\mathbf{x}$ is non-zero. Thus the condition $\mathbf{v}(\epsilon) = \nabla F(\mathbf{x}(\epsilon))$ is violated for all small enough positive ϵ, as the gradient of a strictly concave function is one-to-one. Hence by the previous analysis, the time derivative of $L(\mathbf{x}, \mathbf{v})$ is negative for all small enough positive ϵ.

We now show that L is a proper Lyapunov function, *i.e.*, it goes to $+\infty$ as its arguments increase in an unbounded fashion. To this end, we first remark that the bracketed term $\langle \mathbf{x}, \mathbf{v} \rangle - F(\mathbf{x}) - F^*(\mathbf{v})$ in the definition of L is non-negative. Indeed, it follows from the definition of F^* that $F^*(\mathbf{v}) \leq \langle \mathbf{x}, \mathbf{v} \rangle - F(\mathbf{x})$ for all $\mathbf{x} \in C$. We then consider the remaining term, $J(\mathbf{v}) - F^*(\mathbf{v})$. Recalling the

definitions of J and F^*, and the assumption that both $\mathbf{z}$, $\mathbf{z}'$ belong to the convex set C, with $\mathbf{z}' > \mathbf{z}$, it is readily seen that

$$J(\mathbf{v}) - F^*(\mathbf{v}) \geq \langle \mathbf{v}, \mathbf{z}' \rangle + F(\mathbf{z}) - \langle \mathbf{v}, \mathbf{z} \rangle = \langle \mathbf{v}, \mathbf{z}' - \mathbf{z} \rangle + F(\mathbf{z})$$

and hence must diverge to infinity as $\mathbf{v}$ does so (recall that v_i are non-negative, which follows from the monotonicity of F). The proof will then be concluded by arguing that the conditions $\mathbf{x} = G(\mathbf{v})$ and $\mathbf{v} = \nabla F(\mathbf{x})$ uniquely characterize the optimal pair $(\mathbf{x}^*, \mathbf{v}^*)$. To this end, let $\mathbf{x} \neq \mathbf{x}^*$, and $\mathbf{v} = \nabla F(\mathbf{x})$. Noting that the function $t \in [0,1] \to F(t\mathbf{x}^* + (1-t)\mathbf{x})$ is strictly concave and maximal at $t = 1$, its derivative at $t = 0+$ must be positive; since this derivative reads $\langle \mathbf{v}, \mathbf{x}^* - \mathbf{x} \rangle$, it readily follows that $\mathbf{x} \neq G(\mathbf{v})$ since $\langle \mathbf{v}, \mathbf{x}^* \rangle$ is strictly larger than $\langle \mathbf{x}, \mathbf{v} \rangle$. The result follows. □

7 Extensions

7.1 A Slotted-Time Model

Our analysis can be directly extended to a slotted-time model. The quantity $\lambda_{i,j}$ would then indicate the probability of an encounter between users in classes $\mathcal{C}_i$ and $\mathcal{C}_j$ at a given timeslot. Similarly, μ_i would indicate the probability of accessing the infrastructure at a given timeslot.

Is this case, the delay until a request from w originating from a user in $\mathcal{C}_i$ is satisfied will be geometrically distributed with parameter $\rho_{i,w}$, given again by (9). Our results can be applied to utilities $U_{i,w} : \mathbb{N} \to \mathbb{R}$ expressed in terms of time slots required to retrieve a website, that satisfy the monotonicity and boundedness conditions in Assumption 1 (differentiability is no longer necessary). Our results hold, *mutatis mutandis*, if one replaces $U'_{i,w}(t)$ with the quantity $U_{i,w}(t+1) - U_{i,w}(t)$. In particular, our mechanism remains essentially the same, the only difference being that user m reports the following quantity instead of the one in (12).

$$-T_{m,w}(t) \cdot [U_{i,w}(T_{m,w}(t) + 1) - U_{i,w}(T_{m,w}(t))] \cdot n_{m,w}(t).$$

7.2 Non-Differentiable Utilities

The assumption that $U_{i,w}$ are differentiable could be replaced by the assumption that they are *càdlàg* (*i.e.*, right continuous with left limits). In such a case, the monotonicity of $U_{i,w}$ implies that it be written as $U_{i,w}(t) = -\int_0^t d\nu_{i,w}$ where $\nu_{i,w}$ a positive measure on $\mathbb{R}_+$. It can then be shown that the problem (10) remains convex. Moreover, the quantities reported by users $m \in \mathcal{C}_i$ would change as follows: $q_{m,w}(t) = -\hat{d}_{i,w}(t) \cdot G_{i,w}(T_{m,w}(t))$ where $T_{m,w}(t)$ is computed as before, G is the function $G_{i,w}(t) = \int_0^t s d\nu_{i,w}(s)$ and $\hat{d}_{i,w}(t)$ is an estimator of the request rate of user m. The latter can be obtained, *e.g.*, as the inverse of the mean time between consecutive requests for website w. This

is less satisfactory than the mechanism employed for differentiable functions, as it requires recovering the request rates of users.

8 Conclusions

We proposed Psephos, a distributed mechanism for computing optimal caching policies in a mobile network. Contrary to earlier work, Psephos is designed to operate in a heterogeneous environment. Caching decisions under Psephos are simple: only items receiving the highest "votes" are stored.

We formally demonstrated that Psephos maximizes social welfare under two main assumptions: the existence of user classes and the time separation between the cache reshuffling and vote processes. Theorem 2 suggests that time separation may not be necessary. Moreover, although our proofs relied on the existence of user classes, Psephos does not make class-dependent caching decisions; in fact, we did not assume a priori and explicit knowledge of classes. In light of the above, the characterization of Psephos's performance, even numerically or empirically, in the absence of these assumptions is an interesting open problem.

Psephos is naturally suitable for a closed system with a limited number of different contents whose sizes are of the same magnitude, but not for an open system like YouTube, *etc.* Extending this work to open systems is interesting future work. Moreover, though our analysis allowed for the personalization of caching strategies through blacklisted and permanent websites, it did not evaluate the effect that user selection of such websites has on the system. Understanding the impact of selfishness, through such selections, on both the social welfare as well as the utilities of individual users, is also an interesting open problem.

References

1. Altman, E., Nain, P., Bermond, J.C.: Distributed storage managements of evolving files in delay tolerant ad hoc networks. In: IEEE INFOCOM (2009)
2. Altman, E., Pellegrini, F.D., Miorandi, D., Neglia, G.: Decentralized stochastic control of delay tolerant networks. In: IEEE INFOCOM (2009)
3. Borkar, V.S.: Stochastic Approximation: A Dynamical Systems Viewpoint. Cambridge University Press (2008)
4. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press (2004)
5. Chaintreau, A., Boudec, J.Y.L., Ristanovic, N.: The age of gossip: Spatial mean-field regime. In: ACM SIGMETRICS (2009)
6. Costa, P., Mascolo, C., Musolesi, M., Picco, G.: Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks. IEEE Jsac **26**(5), 748–760 (2008)
7. Gunawardena, D., Karagiannis, T., Proutiere, A., and Santos-Neto, E., and Vojnovic, M.: Scoop: decentralized and opportunistic multicasting of information streams. In ACM MobiCom (2011)
8. Guo, S., Keshav, S.: Fair and efficient scheduling in data ferrying networks. In: ACM CoNEXT (2007)
9. Ioannidis, S., Chaintreau, A., Massoulié, L.: Optimal and scalable distribution of content updates over a mobile social network. In: IEEE INFOCOM (2009)

10. Isaacman, S., Martonosi, M.: Potential for collaborative caching and prefetching in largely-disconneced villages. In: Wireless Networks and Systems for Developing Regions Workshop (2008)
11. Lenders, V., May, M., , Karlsson, G.: Wireless ad hoc podcasting. In: IEEE SECON (2007)
12. Papadopouli, M., Schulzrinne, H.: Effects of power conservation, wireless coverage and cooperation on data dissemination among mobile devices. In: ACM MobiHoc (2001)
13. Reich, J., Chaintreau, A.: The age of impatience: Optimal replication schemes for opportunistic networks. In: ACM CoNext (2009)
14. Rockafellar, R.T.: Convex Analysis. Princeton Univesity Press (1996)
15. Yoneki, E., Hui, P., Chan, S., Crowcroft, J.: A socio-aware overlay for pub/sub communication in DTN. In: ACM MSWiM (2007)