Distributed Caching over Heterogeneous Mobile Networks

Stratis Ioannidis, Laurent Massoulié, and Augustin Chaintreau
Technicolor, Paris, France
stratis.ioannidis@technicolor.com, laurent.massoulie@technicolor.com,
augustin.chaintreau@technicolor.com

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 rigorously show that PSEPHOS maximizes the system's *social welfare*. Our results are derived formally using techniques from stochastic approximation and convex optimization; to the best of our knowledge, our work is the first to address caching with heterogeneity in a fully distributed manner.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*distributed networks, store and forward networks*; C.4 [**Performance of Systems**]: Performance Attributes

General Terms

Algorithms

Keywords

caching, heterogeneity, opportunistic networks, content distribution

1. INTRODUCTION

In this work, we consider a peer-to-peer content sharing system built 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, 8, 12]. 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 ideal for the distribution of content when access to the infrastructure is intermittent, as well as when the downlink bandwidth available for content distribution is limited. Their inherent drawback is delay: users can retrieve content only when they have access to the infrastructure or when they encounter other users storing it. For this reason, the content shared should be *delay-tolerant*, in the sense that prompt delivery is not a strict requirement.

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. In general, users should store content that is likely to be of use either to themselves or other users they encounter. 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. Heterogeneity implies that caching policies aiming to minimize delays may vary considerably among different users. An additional challenge arises from the fact that at least two of the above parameters, namely user preferences and mobility, may *not be a priori known*. In this sense, any mechanism for determining a user's caching policy cannot readily access this information.

Another challenge posed by determining caching policies is the need for *decentralization*. To begin with, as important system parameters are not a priori known, 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. Moreover, as in traditional, wired peer-to-peer systems, mobile users may 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. Finally, our mechanism 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 rigorously 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. We note that our focus is on a theoretical study of the optimality of PSEPHOS; we leave possible experimental or empirical studies of the mechanism as future 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 Section 5. We discuss possible extensions of our methods and results in Section 6, and conclude in Section 7.

2. RELATED WORK

Sharing content using opportunistic contacts between mobile devices has been recently proposed within several different contexts, including website downloading (7DS project [11]), the dissemination of podcasts (Podnet project [10]) and newsfeeds [1,5,8], publish-subscribe systems [6,14], and file sharing [12]. 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 [12]; 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" [12]. 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 [12] 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 [12] 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 [12] 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.

The above problem has also strong ties to delay-tolerant broadcasting. In [1] and [2], updates for a single content item are propagated in a homogeneous contact scenario, and 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 [8] under heterogeneous mobility, while ODEs

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

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, the application we consider has connections to data ferrying networks such as KioskNet [7, 9]. 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 prefetching policies for such devices, driven by user demand.

3. SYSTEM DESCRIPTION

3.1 Content Sharing over a Mobile Network

3.1.1 Mobile User Classes

Our system consists of N mobile users, partitioned in L distinct classes $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_L$. Denote by $|\mathcal{C}_i|$ the size of class $\mathcal{C}_i$. Some of our results will be proved for a large system, *i.e.*, one in which N tends to infinity; whenever considering such limits, we will assume that the user population in each class grows proportionally to N, *i.e.*, for all i,

$$|\mathcal{C}_i| = r_i N, \qquad (1)$$

where r_i is a constant (in N).

We assume that 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, we assume that 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 our mechanism (*e.g.*, on what to store in a user's cache) will not depend on a priori knowledge of classes.

3.1.2 User Contacts and Access to the Infrastructure

We will assume that users encounter each other according to independent Poisson processes. In particular, we assume that 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 $\mathcal{C}_j$ (*i.e.*, each member of $\mathcal{C}_j$ is equally likely to meet m). We assume that $\lambda_{i,j}$ are constants that do not depend on N. Hence, any given user meets—and shares content—with only a constant number of other users per second; this is important, as users typically have limited bandwidth.

Contacts are symmetric, *i.e.*, m meets m' if and only if m' meets m. As a result,

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

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.

Moreover, we will assume that 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 indicates, *e.g.*, that the user is within the vicinity of her home or office and can access the Internet through a wireless connection. Note that we require μ_i to be strictly positive: every user accesses the infrastructure, eventually.

3.1.3 Content Sharing

The 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.

Furthermore, a given 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 the user m stores the requested website in her public cache, the request is satisfied immediately. If not, the request is stored. It is subsequently satisfied when the user meets some other user that stores the requested website, or when the user accesses the infrastructure.

Under the above scenario, users care simply about retrieving any copy of a website and are indifferent to how stale it might be. Note that this is a simplification as, in reality, off-line content cached by users could become outdated. However, the above behavior is realistic website contents are updated infrequently.

The purpose of the public cache of a user is to store prefetched content in order to serve future requests. As a result, its contents will be the outcome of the optimization performed by PSEPHOS, our distributed caching mechanism, as described in Section 3.3. In particular, 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.

3.1.4 Refreshing vs. Reshuffling

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. Since users are indifferent to how stale websites are, the refreshing operation has no effect in how requests for content are satisfied. For the sake of concreteness, we will assume that refreshing occurs at *all* possible opportunities; this also ensures that cached content is as fresh as possible.

3.1.5 Permanent, Blacklisted and Unrestricted Sites

In general, the contents of a user's cache are determined by PSEPHOS with the global goal of minimizing delays. Nonetheless, the mechanism does give users partial control of what is pre-fetched in their cache. This control has the following form: users 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 of preferred websites, which we call *permanent*, that are *always* stored in their cache. We denote this set by $\mathcal{PR}_i \subseteq \mathcal{W}$, where

$$0 \leq |\mathcal{PR}_i| \leq c_i. \tag{3}$$

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 $\mathcal{C}_i$; 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

$$c_i \leq |\mathcal{W}| - |\mathcal{BL}_i| \leq |\mathcal{W}|. \tag{4}$$

The first inequality states that after removing the blacklisted sites, there are at least c_i websites available. We can assume that this is true without loss of generality: if not, then the capacity of the caches in class i can be considered equal to $W - |\mathcal{BL}_i|$, as any excess capacity remains unused. Note that the sets $\mathcal{PR}_i$ and $\mathcal{BL}_i$ need not be correlated 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

$$c'_i = c_i - |\mathcal{PR}_i|, \tag{5}$$

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

$$\mathcal{F}_i = \mathcal{W} \setminus (\mathcal{PR}_i \cup \mathcal{BL}_i), \tag{6}$$

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 (3)–(6) 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.

3.2.1 Caching Policies and Replication Ratios

For all $w \in \mathcal{W}$, let

$$y_{m,w} = \begin{cases} 1, & \text{if } m \text{ stores } w, \\ 0, & \text{o.w.} \end{cases} \tag{7}$$

be a 0-1 variable indicating if user $m \in \mathcal{C}_i$ stores website w. We will refer to the vector $\vec{y}_m = [y_{m,w}]_{w\in\mathcal{W}}$ as the *caching policy of mobile user m*. Note that, by definition of permanent and blacklisted websites, for every $m \in \mathcal{C}_i$,

$$y_{m,w} = 1,\ \forall w \in \mathcal{PR}_i, \quad \text{and} \quad y_{m,w} = 0,\ \forall w \in \mathcal{BL}_i.$$

For all $w \in \mathcal{W}$, denote by

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

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 will refer to the vector $\vec{x}_i = [x_{i,w}]_{w\in\mathcal{W}}$ as the *replication vector of class $\mathcal{C}_i$*.

3.2.2 Social Welfare

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, $Y_{m,w}$ is a exponentially distributed random variable with mean $1/\rho_{i,w}$, where

$$\rho_{i,w} = \mu_i + \sum_j \lambda_{i,j} x_{j,w}. \tag{8}$$

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

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

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" [12] 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 $|U_{i,w}(t)| = o\left(e^{ct}\right)$, for all $c > 0$.*

The assumption that $U_{i,w}$ are non-increasing is natural (see also [8, 12]): 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 utilities, and for negative utilities it suffices that $|U_{i,w}(t)|$ 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 6.2.

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 $\vec{x}_i$, we wish to solve the following optimization problem:

$$\text{Maximize } F(\mathbf{X}) = \sum_{i,w} r_i d_{i,w} \mathbb{E}[U_{i,w}(Y_{i,w})] \tag{9a}$$

$$\text{subject to: } \sum_{w\in\mathcal{F}_i} x_{i,w} \leq c'_i, \quad \text{for all } i, \tag{9b}$$

$$0 \leq x_{i,w} \leq 1, \quad \text{for all } i, w, \tag{9c}$$

$$x_{i,w} = 1, \quad \text{for all } i, w \in \mathcal{PR}_i, \tag{9d}$$

$$x_{i,w} = 0, \quad \text{for all } i, w \in \mathcal{BL}_i. \tag{9e}$$

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 actual capacity of a cache in class i. Note that our mechanism can only affect the replication ratios $\mathbf{X}$ indirectly: its decisions determine the caching policies $\vec{y}_m$ at individual users. We denote by

$$D = \{\mathbf{X} \in \mathbb{R}^{L\times|\mathcal{W}|} \text{ satisfying (9b) to (9e)}\} \quad (10)$$

the feasible domain of (9). Note that, in reality, our mechanism cannot achieve *any* possible replication ratio in D. The achievable replication vectors in a class of r_iN are "quantized", in the sense that the coordinates of $\vec{x}_i$ need to be integer multiples of $1/r_iN$. 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

Under these constraints set by permanent and blacklisted sites, 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

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

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 $\vec{v}_m(t)$ in a decreasing order, and then caches the websites with the c'_i highest votes.

Obviously, the process determining the vote vector $\vec{v}_m$ 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 $\vec{v}_m$ 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 $\vec{v}_m(t)$ 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 essentially 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 node that stores w or the infrastructure. We denote this quantity at time t by $T_{m,w}(t) > 0$. 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 $n_{m,w}(t) = 0$.

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 has issued a request for w at some point in the past.

Whenever m meets another user, she reports to the latter the following quantity for every website w:

$$rep_{m,w}(t) = -T_{m,w}(t) \cdot U'_{i,w}\left(T_{m,w}(t)\right) \cdot n_{m,w}(t), \quad (11)$$

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.

It is important to note that if user m meets a user m' that stores w, m first reports $rep_{m,w}(t)$ and then updates these values accordingly (by, *e.g.*, updating the value of $T_{m,w}$ and setting $n_{m,w}$ to zero). Moreover, in practice, $rep_{m,w}(t)$ need only be reported for all w for which there are pending requests, *i.e.*, $n_{m,w}(t) > 0$. For all other w, $rep_{m,w}$ is zero and need not be reported.

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

$$v_{m,w}(t+\delta) = (1-\beta(N)\delta)v_{m,w}(t)+\beta(N)g_{m,w}(t,t+\delta). \quad (12)$$

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

$$g_{m,w}(t,t+\delta)=\begin{cases} rep_{m',w}(t), \text{ if } m, m' \text{ meet in } (t,t+\delta] \\ U_{i,w}(0)-U_{i,w}(T_{m,w}(t)), \text{ if } m \\ \qquad \text{requests } w \text{ in } (t,t+\delta] \\ 0, \text{ o.w.} \end{cases} \quad (13)$$

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 $v_{m,w}(t)$.

We note that, in practise, a user may 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 s.t. $\lambda_{i,j} \neq 0$. In other words, either m herself or a user she meets has requested site $w \in \mathcal{F}_i$ at some time in the past. For all other sites, the vote $v_{m,w}(t)$ can be assumed to be equal to zero and need not be maintained.

Note that the vote vector $\vec{v}$ 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.

3.3.2 Psephos as Gradient Descent

Some intuition behind Psephos can be obtained by studying the "reports" exchanged by users. These reports are de-

Table 1: Summary of Notation

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	Relative size of class $\mathcal{C}_i$.
c_i	Total capacity of a cache in class $\mathcal{C}_i$.
c_i'	Actual capacity of a cache in class $\mathcal{C}_i$.
$\mathcal{PR}_i$	Permanent websites in class $\mathcal{C}_i$.
$\mathcal{BL}_i$	Blacklisted websites in class $\mathcal{C}_i$.
$\mathcal{F}_i$	Unrestricted websites in class $\mathcal{C}_i$.
$\vec{y}_m$	Caching policy of user m .
$\vec{x}_i$	Replication vector of class $\mathcal{C}_i$.
$\mathbf{X}$	Matrix of replication vectors $\vec{x}_i$.
$\lambda_{i,j}$	Contact rate of $m \in \mathcal{C}_i$ with users in $\mathcal{C}_j$.
μ_i	Contact rate of $m \in \mathcal{C}_i$ with the infrastructure.
$d_{i,w}$	Request rate of w by a given user $m \in \mathcal{C}_i$.
$U_{i,w}(t)$	Utility at class i when finding website w within time t.
$Y_{i,w}$	Delay of discovery of website w from a user in $\mathcal{C}_i$.
$\rho_{i,w}$	Contact rate of a user in $\mathcal{C}_i$ with users storing w.
$F(\mathbf{X})$	The social welfare.
D	The feasible domain of (9).
$\alpha(N)$	Gain of cache policy selection mechanism.
$\beta(N)$	Gain of vote averaging mechanism.
$\vec{v}_m$	Vote vector of user m.

signed so that (13), the quantity that the EWMA averages over, is in fact an estimator of the gradient of F. In particular, for $m \in \mathcal{C}_i$,

$$\mathbb{E}[g_{m,w}(t, t+\delta)] \sim \delta \cdot \frac{1}{r_i} \frac{\partial F}{\partial x_{i,w}} \tag{14}$$

—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, (14) 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 $\vec{v}_m$.

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*

$$\lim_{N\to\infty} \frac{N\alpha(N)}{\beta(N)} = 0, \tag{15}$$

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 our mechanism 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 $\vec{v}_m$.

Eq. (15) 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 timescale at which the vote processes $\vec{v}_m(t)$ change and at the timescale 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 $\vec{v}_m$ has already converged close enough to the quantity that it is suppose 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:

$$\epsilon(N) = \sqrt{O\left(\sqrt{\frac{N\alpha(N)}{\beta(N)}}\right) + O\left(\frac{1}{N}\right)}. \tag{16}$$

The first term of the above formula is due to the "error" induced by the estimate of ∇F through the votes $\vec{v}_m$. The second term is due to the "quantization" effect we mentioned in Section 3.2.2. 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 timescales 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 timescale separation is truly necessary. Our second main result indicates 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 timescales 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 $\vec{x} \in C$ and $\vec{v} \in \mathbb{R}_+^d$

$$\frac{d}{dt}\vec{x} = \alpha\left(G(\vec{v}) - \vec{x}\right), \tag{17a}$$

$$\frac{d}{dt}\vec{v} = \beta\left(\nabla F(\vec{x}) - \vec{v}\right), \tag{17b}$$

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

$$G(\vec{v}) = \arg\max_{\vec{x}\in C}\langle \vec{x}, \vec{v}\rangle, \tag{18}$$

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

THEOREM 2. *Assume that C contains two vectors $\vec{z}$, $\vec{z}'$ such that $z_i < z_i'$ for all $i = 1, \ldots, d$. Under the above assumptions, the dynamical system* (17) *converges to $(\vec{x}^*, \vec{v}^*)$ where $\vec{x}^*$ achieves the maximum of F over C, and $\vec{v}^* = \nabla F(\vec{x}^*)$.*

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 (17) 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 $\vec{v}$, which, as all users in the same class have the same votes, can be written as the concatenation of the vectors $\vec{v} = [\vec{v}_1; \vec{v}_2; \ldots; \vec{v}_L]$, where $\vec{v}_i$ the vote vector of class $\mathcal{C}_i$. Note that (17b) 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, (17a) captures the dynamic evolution of the replication ratios over all classes. To see this, observe that given the constraints C the inner product $\langle \vec{x}, \vec{v}\rangle$ is maximized in terms of $\vec{x} \in C$ when $\vec{x}$ takes the value 1 at the c_i top elements of each sub-vector $\vec{v}_i$ of $\vec{v}$, and the value 0 everywhere else. This is indeed implemented at 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 (17a) captures system dynamics.

In spite of the simplifications involved, the convergence of (17) is far from obvious. In particular, as there is no time separation between the parameters α and β, the evolution of (17a) is not guaranteed to be supplied with a correct estimate of the gradient ∇F through the vote vector $\vec{v}$. From a technical standpoint, the proof requires a Lyapunov function that exploits a relationship between the function G and conjugate duality (see Section 5.6).

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

5. ANALYSIS

5.1 Convexity and Differentiability

We begin our analysis by showing that maximizing social welfare is a convex optimization problem [4, 13]. This was proved in a slightly different form in [12] (see Theorems 1 and 2 therein); we repeat the proof below for the sake of completeness.

LEMMA 1. *The optimization problem* (9) *is convex. In particular, the objective function F is concave in $\mathbf{X} \in D$.*

PROOF. Our proof follows the same arguments as [12]. 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

$$\mathbb{E}[U_{i,w}(Y_{i,w})] = U_{i,w}(0) + (1 - x_{i,w})\int_0^{\infty} U'_{i,w}(s)e^{-\rho_{i,w}s}ds \tag{19}$$

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 $\vec{x}_i$; it can be verified that the product of such functions is convex [12]. 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*

$$\frac{\partial F}{\partial x_{i,w}} = r_i d_{i,w}\mathbb{E}[U_{i,w}(0) - U_{i,w}(\hat{Y}_{i,w})] - \sum_j r_j \frac{d_{j,w}}{\rho_{j,w}}(1 - x_{j,w})\lambda_{j,i}\mathbb{E}[\hat{Y}_{j,w}U'_j(\hat{Y}_{j,w})] \tag{20}$$

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 (19) we have that

$$\frac{\partial F}{\partial x_{i,w}} = -r_i d_{i,w} \int_0^\infty U'_{i,w}(s) e^{-\rho_{i,w} s} ds$$
$$- \sum_j r_j d_{j,w} (1 - x_{j,w}) \int_0^\infty U'_{j,w}(s) e^{-\rho_{j,w} s} s \lambda_{j,i} ds. \quad (21)$$

Moreover,

$$\int_0^\infty U'_{i,w}(s) e^{-\rho_{i,w} s} ds \overset{(19)}{=} \int_0^\infty \int_0^t U'_{i,w}(s) \rho_{i,w} e^{-\rho_{i,w} t} ds dt$$
$$= \int_0^\infty [U_{i,w}(t) - U_{i,w}(0)] \rho_{i,w} e^{-\rho_{i,w} t} dt$$

while

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

Eq. (20) therefore follows. Note that, by (21), $\partial F / \partial x_{i,w}$ is decreasing in $\rho_{j,w}$ for all j. Hence, by the monotone convergence theorem it is continuous in $\rho_{j,w} \in [\mu_j, 1]$, and is therefore also continuous in $\mathbf{X}$. Note the following fact:

$$\text{If } |f(t)| = o(e^{ct}) \ \forall c > 0, \text{ then so is } |f'(t)| \text{ and } |tf(t)|. \quad (22)$$

Hence, the quantity

$$\left| \frac{\partial F}{\partial x_{i,w}} \right| \overset{(21)}{\leq} r_i d_{i,w} \int_0^\infty (-U'_{i,w}(s)) e^{-\mu_i s} ds$$
$$+ \sum_j r_j d_{j,w} \lambda_{j,w} \int_0^\infty (-s U'_{j,w}(s)) e^{-\mu_j s} ds,$$

is bounded by Assumption 1 and (22). Differentiating (21) 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 $\int_0^\infty |U'_{j,w}(s)| s^2 e^{-\mu_j s} ds$, for all j and w, required for this result, is indeed implied by Assumption 1 and (22). □

Note that 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 $\vec{v}_m$.

Let $m \in \mathcal{C}_i$ and assume that the vector $\vec{v}_m$ is sorted in decreasing order. Let $\tau_i(\vec{v})$ be the average of the c'_i-th and $c'_i + 1$-th entry in the sorted vector $\vec{v}_m$ (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. $v_{m,w} > \tau_i(\vec{v})$ is placed in m's cache while every website $w \in \mathcal{F}_i$ s.t. $v_{m,w} < \tau_i(\vec{v}_m)$ is removed. Hence, $\tau_i(\vec{v}_m)$ is a threshold indicating whether a website enters the cache or not.

Note that websites such that $v_{m,w} = \tau_i(\vec{v})$ exist if and only if the c'_i-th and $c'_i + 1$-th entry in the sorted vector $\vec{v}_m$ 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 at all times equal c'_i sites belonging to $\mathcal{F}_i$ are in the cache.

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

$$y_{m,w}(t + \delta) = y_{m,w}(t) + h_{m,w}(t, t + \delta), \quad \text{where} \quad (23)$$

$$h_{m,w}(t, t+\delta) = \begin{cases} 0, & \text{if } m \text{ does not reshuffle} \\ & \quad \text{its cache in } (t, t+\delta] \\ \Delta_{m,w}(\vec{v}_m, \vec{y}_m), & \text{o.w.} \end{cases} \quad (24)$$

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

$$\Delta_{m,w}(\vec{v}_m, \vec{y}_m) = \mathbb{1}_{v_{m,w} > \tau_i(\vec{v}_m) \wedge y_{m,w} = 0} - \mathbb{1}_{v_{m,w} < \tau_i(\vec{v}_m) \wedge y_{m,w} = 1}$$
$$+ \mathbb{1}_{v_{m,w} = \tau_i(\vec{v}_m)} z_{m,w}(\vec{v}_m, \vec{y}_m). \quad (25)$$

The quantity $z_{m,w}(\vec{v}_m, \vec{y}_m)$ 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:

$$\sum_{w \in \mathcal{F}_i} \Delta_{m,w}(\vec{v}_m, \vec{y}_m) = 0. \quad (26)$$

Eq. (26) places a constraint on the quantities $z_{m,w}(\vec{v}_m, \vec{y}_m)$: 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 System Drift within Small Intervals

The functions $g_{m,w}$ and $h_{m,w}$, given by (13) and (24), respectively, characterize the change of the votes and the caching strategies within a small time interval $(t, t+\delta]$. 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 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 $T_{m,w}(t)$.

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

$$\mathbb{E}[g_{m,w}(t, t+\delta)] = O\left(\delta^2\right) + \delta d_{i,w} \mathbb{E}[U_{i,w}(0) - U_{i,w}(\hat{Y}_{i,w}))]$$
$$+ \delta \sum_j \sum_{m' \in \mathcal{C}_j} \frac{\lambda_{i,j}}{r_j N} \mathbb{1}_{y_{m',w}=0} \frac{d_{j,w}}{\rho_{j,w}} \mathbb{E}[-\hat{Y}_{m',w} U'_{j,w}(\hat{Y}_{m',w})]$$

$$\overset{(2)}{=} O\left(\delta^2\right) + \delta d_{i,w} \mathbb{E}[U_{i,w}(0) - U_{i,w}(\hat{Y}_{i,w}))]$$
$$+ \delta \frac{1}{r_i} \sum_j r_j \lambda_{j,i} (1 - x_{i,w}) \frac{d_{j,w}}{\rho_{j,w}} \mathbb{E}[-\hat{Y}_{m',w} U'_{j,w}(\hat{Y}_{m',w})]$$

$$= \delta \frac{1}{r_i} \frac{\partial F}{\partial x_{i,w}} + O\left(\delta^2\right).$$

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

$$\mathbb{E}[g_{m,w}(t,t+\delta)] \leq O\left(\delta^2\right) + \delta d_{i,w}\mathbb{E}\left[|U_{i,w}(0) - U_{i,w}(\hat{Y}_{i,w}))|^2\right] + \delta O\left(\sum_j \frac{1}{\rho_{j,w}^2} \int_0^\infty (sU'_{j,w}(s))^2 \rho_{i,w} e^{-\rho_{i,w}s} ds\right)$$

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 (22), and $1/\rho_{j,w}$ is bounded by $1/\mu_j$. The lemma therefore follows. $\square$

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

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

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

where $M'_{m,w}$ a r.v. with $\mathbb{E}[M'_{m,w}] = 0$ and $\mathbb{E}[|M'_{i,w}|^2] < 1$, and $\Delta_{m,w}(\vec{v}_m, \vec{y}_m)$ as in (25).

PROOF. It is easy to see from (24) and the definition of the contact process among users and the infrastructure that

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

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

5.4 System Evolution In Terms of Timescale Separation

Lemmas 3 and 4 imply that the evolution of $\vec{v}_m$ and $\vec{x}_i$ can be described by the following stochastic difference equations:

$$\begin{cases} v_{m,w}(t+\delta) = (1-\beta\delta)v_{m,w}(t) + \beta\delta\left[\frac{1}{r_i}\frac{\partial F(\mathbf{X})}{\partial x_{i,w}} + M_{i,w} + O(\delta)\right] \\ x_{i,w}(t+\delta) = x_{i,w}(t) + \frac{\mu_i \alpha}{r_i}\delta \cdot \\ \qquad \left\{\frac{1}{r_i N}\sum_{m \in \mathcal{C}_i} \Delta_{m,w}(\vec{v}_m, \vec{y}_m) + M'_{i,w} + O(\delta)\right\} \end{cases}$$

where $\mathbb{E}[M_{i,w}] = \mathbb{E}[M'_{i,w}] = 0$ and their second moments are bounded for all m, i and w.

Hence, the evolution of $\vec{v}_m$, $m \in \mathcal{C}_i$, occurs at a timescale of the order of β, while the evolution of $\vec{x}_i$ occurs at a timescale of the order of $\frac{\mu_i \alpha}{r_i}$. The following lemma states that if the above timescales 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.

$$\lim_{N\to\infty} \delta(N)\max(1, \beta(N)) = 0. \tag{27}$$

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, the evolution of $\{v_{m,w}^k, \mathbf{X}^k\}_{k\in\mathbb{N}}$ can be described as:

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

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 Borkhar [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. (27) holds. $\square$

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

$$\epsilon(k) = \sup_{m,w} |v_{m,w}(k) - \frac{1}{r_i}\frac{\partial F(\mathbf{X}(k))}{\partial x_{i,w}}|. \tag{28}$$

LEMMA 6.

$$\limsup_{k\to\infty} \mathbb{E}[|\epsilon(k)|^2] = O\left(\frac{\mu_i N \alpha(N)}{r_i \beta(N)}\right). \tag{29}$$

PROOF. By Lemma 5,

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

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*

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

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

$$\Delta_i(\vec{u}_i, \vec{x}_i) = \tilde{\mu}_i \left\{ \sum_{w\in\mathcal{F}_i: u_{i,w} > \tau_i(\vec{u}_i)} (u_{i,w} - \tau_i(\vec{u}_i))(1 - x_{i,w}) + \sum_{w\in\mathcal{F}_i: u_{i,w} < \tau_i(\vec{u}_i)} (u_{i,w} - \tau_i(\vec{u}_i)) x_{i,w} \right\} \tag{30}$$

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

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

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

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

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

$$\begin{aligned}
&F(\mathbf{X}(k+1)) - F(\mathbf{X}(k)) = \\
&= \sum_{i,w}\left(\frac{\partial F(\mathbf{X}(k))}{\partial x_{i,w}} + O\left(\frac{1}{N}\right)\right)(x_{i,w}(k+1) - x_{i,w}(k)) \\
&= \sum_{i,w\in\mathcal{F}_i}\frac{u_{i,w}}{N}\left\{\sum_{m\in\mathcal{C}_i}\frac{\tilde{\mu}_i}{r_i N}\Delta_{m,w}(\vec{v}_m,\vec{y}_m) + M''_{i,w}\right\} + O\left(\frac{1}{N^2}\right) \\
&= \frac{1}{N}\sum_{i,w\in\mathcal{F}_i}\tilde{\mu}_i u_{i,w}\sum_{m\in\mathcal{C}_i}\frac{\Delta_{m,w}(\vec{v}_m,\vec{y}_m)}{r_i N} + \frac{M}{N} + O\left(\frac{1}{N^2}\right) \quad (31)
\end{aligned}$$

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

$$\begin{aligned}
&\sum_{w\in\mathcal{F}_i}\tilde{\mu}_i u_{i,w}\left\{\sum_{m\in\mathcal{C}_i}\frac{\Delta_{m,w}(\vec{v}_m,\vec{y}_m)}{r_i N}\right\} = \\
&\stackrel{(25)}{=} \frac{\tilde{\mu}_i}{r_i N}\sum_{m\in\mathcal{C}_i}\Bigg\{\sum_{\substack{w\in\mathcal{F}_i: v_{m,w} > \tau_i(\vec{v}_m)\\ \wedge y_{m,w}=0}} u_{i,w} - \sum_{\substack{w\in\mathcal{F}_i: v_{m,w} < \tau_i(\vec{v}_m)\\ \wedge y_{m,w}=1}} u_{i,w} \\
&\quad + \sum_{w\in\mathcal{F}_i: v_{m,w} = \tau_i(\vec{v}_m)} u_{i,w} z_{m,w}(\vec{v}_m,\vec{y}_m)\Bigg\}
\end{aligned}$$

Note that τ_i is Lipschitz continuous. In particular

$$|\tau_i(\vec{v}_m) - \tau_i(\vec{u}_i)| \le \|\vec{v}_m - \vec{u}_i\|_\infty \stackrel{(28)}{\le} \epsilon(k). \quad (32)$$

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

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

This, along with (26), implies that

$$\begin{aligned}
&\sum_{w\in\mathcal{F}_i}\tilde{\mu}_i u_{i,w}\left\{\sum_{m\in\mathcal{C}_i}\frac{\Delta_{m,w}(\vec{v}_m,\vec{y}_m)}{r_i N}\right\} = \\
&= \frac{\tilde{\mu}_i}{r_i N}\sum_{m\in\mathcal{C}_i}\Bigg\{\sum_{\substack{w\in\mathcal{F}_i: v_{m,w} > \tau_i(\vec{v}_m)\\ \wedge y_{m,w}=0}}(u_{i,w} - \tau_i(\vec{u}_i)) \\
&\quad + \sum_{\substack{w\in\mathcal{F}_i: v_{m,w} < \tau_i(\vec{v}_m)\\ \wedge y_{m,w}=1}}(\tau_i(\vec{u}_i) - u_{i,w})\Bigg\} + O(2\tilde{\mu}_i W\epsilon(k)) \\
&= \tilde{\mu}_i\Bigg\{\sum_{\substack{w\in\mathcal{F}_i:\\ u_{i,w} > \tau_i(\vec{u}_i)}}(u_{i,w} - \tau_i(\vec{u}_i)(1 - x_{i,w}) + \sum_{\substack{w\in\mathcal{F}_i:\\ u_{i,w} < \tau_i(\vec{u}_i)}}(u_{i,w} - \tau_i(\vec{u}_i)x_{i,w}\Bigg\} \\
&\quad + E_1 + E_2 + O(2\tilde{\mu}_i W\epsilon(k)) \quad (33)
\end{aligned}$$

where

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

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

By (28) and (32), for every $w \in \mathcal{F}_i$ s.t. $u_{i,w} > \tau_i(\vec{u}_i) + 2\epsilon(k)$, $v_{m,w} > \tau_i(\vec{v}_m)$, while for every $w \in \mathcal{F}_i$ s.t. $u_{i,w} < \tau_i(\vec{u}_i) - 2\epsilon(k)$, $v_{i,m} < \tau_i(\vec{v}_m)$. This implies that the indicator functions in E_1 and E_2 (expressed in terms of $\vec{v}_m$ and $\vec{u}_i$) may only differ for $w \in \mathcal{F}_i$ such that $|u_{i,w} - \tau_i(\vec{u}_i)| \le 2\epsilon(k)$. As such

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

and the same can be stated about E_2. The lemma therefore follows from (31) and (33). $\square$

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

$$\sum_i \Delta_i(\vec{u}_i, \vec{x}_i) \ge 0,$$

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 (9), 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 $\vec{X}(k)$, the mean drift $\sum_i \Delta_i \vec{u}_i, \vec{x}_i$ is small, then $F(\vec{X}(k))$ is guaranteed to be close to the maximum value of F in D:

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

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

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

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

PROOF. The Lagrangian of (9) is

$$\begin{aligned}
\mathcal{L}(\mathbf{X}, \vec{\xi}, \mathbf{\Psi}, \mathbf{\Phi}) &= F(\mathbf{X}) + \sum_i \xi_i(c'_i - \sum_{w\in\mathcal{F}_i} x_{i,w}) \\
&\quad + \sum_{i,w\in\mathcal{F}_i}\psi_{i,w}(1 - x_{i,w}) + \sum_{i,w\in\mathcal{F}_i}\phi_{i,w}x_{i,w} \quad (34)
\end{aligned}$$

Therefore, the KKT conditions of (9) are

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

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

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

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

Then,

$$f(\vec{\xi}^*,\mathbf{\Psi}^*,\mathbf{\Phi}^*) = L(\mathbf{X}^*,\vec{\xi}^*,\mathbf{\Psi}^*,\mathbf{\Phi}^*). \tag{35}$$

To see this, observe that,

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

by the definition of $\vec{\xi}^*,\mathbf{\Psi}^*,\mathbf{\Phi}^*$ and $\vec{u}_i^*$. Moreover, since F is concave in $\mathbf{X}\in D$, so is $\mathcal{L}$. The above imply that the supremum of $\mathcal{L}(X,\vec{\xi}^*,\mathbf{\Psi}^*,\mathbf{\Phi}^*)$ in D is attained at $\mathbf{X}^*$, which yields (35). The max-min principle implies that

$$\sup_{\mathbf{X}\in D}\inf_{\substack{\vec{\xi}\geq 0,\\ \mathbf{\Psi}\geq 0,\mathbf{\Phi}\geq 0}}\mathcal{L}(\mathbf{X},\vec{\xi},\mathbf{\Psi},\mathbf{\Phi}) \leq \inf_{\substack{\vec{\xi}\geq 0,\\ \mathbf{\Psi}\geq 0,\mathbf{\Phi}\geq 0}}\sup_{\mathbf{X}\in D}\mathcal{L}(\mathbf{X},\vec{\xi},\mathbf{\Psi},\mathbf{\Phi})$$

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

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

as all coefficients of $\vec{\xi},\mathbf{\Psi}$, and $\mathbf{\Phi}$ in (34) are positive, and

$$\inf_{\substack{\vec{\xi}\geq 0,\\ \mathbf{\Psi}\geq 0,\mathbf{\Phi}\geq 0}}\sup_{\mathbf{X}\in D}\mathcal{L}(\mathbf{X},\vec{\xi},\mathbf{\Psi},\mathbf{\Phi}) \leq f(\vec{\xi},\mathbf{\Psi},\mathbf{\Phi})$$

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

$$\begin{aligned}
&\sup_{\mathbf{X}\in D}F(\mathbf{X}) \leq f(\vec{\xi}^*,\mathbf{\Psi}^*,\mathbf{\Phi}^*) \overset{(35)}{=} L(\mathbf{X}^*,\vec{\xi}^*,\mathbf{\Psi}^*,\mathbf{\Phi}^*)\\
&= F(\mathbf{X}^*) + \sum_i \xi_i^*(c_i - \sum_w x_{i,w}^*) + \sum_{i,w}\psi_{i,w}^*(1 - x_{i,w}^*)\\
&\quad + \sum_{i,w}\phi_{i,w}^* x_{i,w}^*\\
&= F(\mathbf{X}^*) + 0 + \sum_i r_i\Big[\sum_{w:u_{i,w}^*>\tau_i(\vec{u}_i^*)}\left(u_{i,w}^* - \tau_i(\vec{u}_i^*)\right)(1 - x_{i,w}^*)\\
&+ \sum_{w:u_{i,w}^*<\tau_i(\vec{u}_i^*)}\left(\tau_i(\vec{u}_i^*) - u_{i,w}^*\right)x_{i,w}^*\Big] \overset{(30)}{\leq} F(\mathbf{X}^*) + \epsilon\max_i\frac{r_i}{\tilde{\mu}_i}
\end{aligned}$$

and the lemma follows. □

5.5 Proof of Theorem 1

We are now ready to prove Theorem 1. From Lemma 7,

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

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

$$\begin{aligned}
\limsup_{k\to\infty}\mathbb{E}[\sum_i\Delta_i(\vec{u}_i,\vec{x}_i)] &= \limsup_{k\to\infty}O\left(\mathbb{E}[|\epsilon(k)|]\right) + O\left(\frac{1}{N}\right)\\
&\overset{(28)}{=} O\left(\sqrt{\frac{N\alpha(N)}{\beta(N)}}\right) + O\left(\frac{1}{N}\right)
\end{aligned}$$

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

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

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

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

and the theorem therefore follows by taking ϵ as in (16). □

5.6 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 $J(\vec{v})$ as

$$J(\vec{v}) := \max_{\vec{x}\in C}\langle\vec{x},\vec{v}\rangle. \tag{36}$$

It then follows from Theorem 23.5, p. 218 in Rockafellar [13] that

$$G(\vec{v}) = \nabla J(\vec{v}). \tag{37}$$

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

$$F^*(\vec{v}) := \inf_{\vec{x}\in C}\langle\vec{x},\vec{v}\rangle - F(\vec{x}). \tag{38}$$

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

$$L(\vec{x},\vec{v}) := J(\vec{v}) - F^*(\vec{v}) - \frac{\beta}{\alpha}\left(F(\vec{x}) + F^*(\vec{v}) - \langle\vec{x},\vec{v}\rangle\right). \tag{39}$$

Upon taking time derivatives, in view of (17a),(17b) and (37), one obtains after term cancellation that

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

Setting $\vec{y} = \nabla F^*(\vec{v})$, it readily follows from the theory of convex function duality [13] that $\vec{v} = \nabla F(\vec{y})$. Thus the time derivative of $L(\vec{x},\vec{v})$ is proportional to $\langle\vec{x} - \vec{y}, \nabla F(\vec{x}) - \nabla F(\vec{y})\rangle$. Now strict concavity of F entails that this scalar product is non-positive, and equals zero if and only if $\vec{x} = \vec{y}$, or equivalently $\vec{v} = \nabla F(\vec{x})$.

Thus $L(\vec{x},\vec{v})$ is strictly decreasing unless $\vec{v} = \nabla F(\vec{x})$. We further argue that it is strictly decreasing also whenever $\vec{x} \neq G(\vec{v})$. To see this, consider a pair of points $(\vec{x},\vec{v})$ such that $\vec{v} = \nabla F(\vec{x})$, but $\vec{x} \neq G(\vec{v})$. Then after some small time $\epsilon > 0$, $\|\vec{v}(\epsilon) - \vec{v}\|$ is of order $o(\epsilon)$ since the time derivative $(d/dt)(\vec{v})$ is of order $o(1)$, while $\|\vec{x}(\epsilon) - \vec{x}\|$ is of order $\Omega(\epsilon)$ since the time derivative $(d/dt)\vec{x}$ is non-zero. Thus the condition $\vec{v}(\epsilon) = \nabla F(\vec{x}(\epsilon))$ 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 $L(\vec{x},\vec{v})$ 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 unboundedly. To this end, we first remark that the bracketed term $\langle\vec{x},\vec{v}\rangle - F(\vec{x}) - F^*(\vec{v})$ in the definition of L is non-negative. Indeed, it follows from the definition of F^* that $F^*(\vec{v}) \leq \langle\vec{x},\vec{v}\rangle - F(\vec{x})$ for all $\vec{x}\in C$. We then consider the remaining term, $J(\vec{v}) - F^*(\vec{v})$. Recalling the definitions of J and F^*, and the assumption that both $\vec{z}$, $\vec{z}'$ belong to the convex set C, with $\vec{z}' > \vec{z}$, it is readily seen that

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

and hence must diverge to infinity as $\vec{v}$ 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 $\vec{x} = G(\vec{v})$ and $\vec{v} = \nabla F(\vec{x})$ uniquely characterize the optimal pair $(\vec{x}^*, \vec{v}^*)$. To this end, let $\vec{x} \neq \vec{x}^*$, and $\vec{v} = \nabla F(\vec{x})$. Noting that the function $t \in [0,1] \rightarrow F(t\vec{x}^* + (1-t)\vec{x})$ is strictly concave and maximal at $t = 1$, its derivative at $t = 0+$ must be positive; since this derivative reads $\langle \vec{v}, \vec{x}^* - \vec{x} \rangle$, it readily follows that $\vec{x} \neq G(\vec{v})$ since $\langle \vec{v}, \vec{x}^* \rangle$ is strictly larger than $\langle \vec{x}, \vec{v} \rangle$. The result follows. $\square$

6. EXTENSIONS

6.1 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 (8). Our results can be applied to utilities $U_{i,w} : \mathbb{N} \rightarrow \mathbb{R}$ 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 (11).

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

6.2 Non-Differentiable Utilities

The assumption that $U_{i,w}$ are differentiable could be replaced by the assumption that they are *càd làg* (*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 (9) remains convex. Moreover, the quantities reported a users $m \in \mathcal{C}_i$ would change as follows: $rep_{m,w}(t) = -\hat{d}_{i,w}(t) \cdot G_{i,w}(T_{m,w}(t))$ 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 that the mechanism employed for differentiable functions, as it requires recovering the request rates of users.

7. CONCLUSIONS

We proposed PSEPHOS, a distributed mechanism for computing optimal caching policies in a mobile netwrk. 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 demonstrate 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 indicates 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.

Finally, 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.

8. REFERENCES

[1] ALTMAN, E., NAIN, P., AND BERMOND, J. C. Distributed storage management of evolving files in delay tolerant ad hoc networks. In *IEEE INFOCOM* (2009).

[2] ALTMAN, E., PELLEGRINI, F. D., MIORANDI, D., AND 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., AND VANDENBERGHE, L. *Convex Optimization*. Cambridge University Press, 2004.

[5] CHAINTREAU, A., BOUDEC, J.-Y. L., AND RISTANOVIC, N. The age of gossip: Spatial mean-field regime. In *ACM SIGMETRICS* (2009).

[6] COSTA, P., MASCOLO, C., MUSOLESI, M., AND PICCO, G. Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks. *IEEE Jsac 26*, 5 (2008), 748–760.

[7] GUO, S., AND KESHAV, S. Fair and efficient scheduling in data ferrying networks. In *ACM CoNEXT* (2007).

[8] IOANNIDIS, S., CHAINTREAU, A., AND MASSOULIÉ, L. Optimal and scalable distribution of content updates over a mobile social network. In *IEEE INFOCOM* (2009).

[9] ISAACMAN, S., AND MARTONOSI, M. Potential for collaborative caching and prefetching in largely-disconneced villages. In *Wireless Networks and Systems for Developing Regions Workshop* (2008).

[10] LENDERS, V., MAY, M., , AND KARLSSON, G. Wireless ad hoc podcasting. In *IEEE SECON* (2007).

[11] PAPADOPOULI, M., AND SCHULZRINNE, H. Effects of power conservation, wireless coverage and cooperation on data dissemination among mobile devices. In *ACM MobiHoc* (2001).

[12] REICH, J., AND CHAINTREAU, A. The age of impatience: Optimal replication schemes for opportunistic networks. In *ACM CoNext* (2009).

[13] ROCKAFELLAR, R. T. *Convex Analysis*. Princeton Univesity Press, 1996.

[14] YONEKI, E., HUI, P., CHAN, S., AND CROWCROFT, J. A socio-aware overlay for pub/sub communication in DTN. In *ACM MSWiM* (2007).