Replication Schemes for Opportunistic Networks with Impatient users

Joshua Reich and Augustin Chaintreau

***Abstract*— As the number of mobile users increases, and cell phones become more powerful, delivering multimedia content to them using a centralized infrastructure becomes both expensive and inadequate. Here we study an alternative solution, which leverages local dedicated caches on these devices to opportunistically fulfill other user requests, in a peer-to-peer manner. The problem we study is that of choosing the *allocation* of content items to these caches, to fulfill the demand of all users in a timely manner. We show that the allocation's efficiency itself is determined by a previously overlooked factor - the *impatience* of content requesters - which leads to completely different allocations being optimal as content requesters impatience follows different *delay-utility* functions. Moreover, although no global cache state can be maintained in such opportunistic environment, we show for a homogeneous network that the optimal cache allocation can be attained by simple reactive replication algorithms that use only local knowledge.**

I. Introduction

Using smartphones to access the Internet is quickly becoming a commodity. Although users expect the same data speed in a nomadic settings using a centralized infrastructure, the recent growth of content access demand raises concern, in particular to distribute rich multimedia content. An alternative solution is to disseminate this content, especially the popular one, reusing in a peer-to-peer (p2p) manner the vast amount of untapped bandwidth, from ad-hoc 802.11 or Bluetooth, that is available to exchange data locally between the nodes.

We assume that some nodes dedicate a cache, where content item can be stored to serve other users request as they meet. The primary question that we address is how to fill these caches with content so that user requesting to access a given item are best satisfied. As opposed to connected p2p, fulfillment of content request may incur non-negligible delay, which lead this system to be sensitive to user-impatience, or generally to how users behave as they spend time waiting for a piece of content. This can be described as a *delay-utility* function mapping delay to utility, which is only assumed to satisfy a monotonic non-increasing behavior. Minimizing the impact of delayed fulfillment on user satisfaction leads to select the allocation of content to caches so as to maximize the aggregate expected utility, or social welfare, in the network. This also depends on the popularity of items and rates of meeting between the nodes.

By selectively replicating local content as node meetings provide the opportunity, the global cache can be driven towards a more efficient allocation. However, since no global knowledge of the network's state can be used, attaining the optimal should be the result of a series of independent decision made by the nodes based on local information only.

We have shown [1] that the following property holds:

- User impatience plays a critical role in determining the optimal allocation for disseminating content. We define the social welfare of a mobile p2p caching system for any delay-utility and global cache allocation.The optimal allocation can be computed efficiently in a centralized manner. These results indicate that, as the user population becomes increasingly impatient, the optimal allocation changes radically: it varies steadily between a uniform allocation dividing the global cache between all content items, and a highly-skewed allocation in which popular items receive a disproportionate share of the global cache.
- Inspired by these results, we develop a reactive distributed algorithm, *Query Counting Replication (QCR)* that for any delay-utility function drives the global cache towards the optimal allocation. Moreover QCR does so without use of any explicit estimators or control channel information.

This invited paper presents a short version of [1].

II. Related Work

The performance of many P2P applications can benefit greatly from opportunistic contacts between the nodes of a mobile network: such solution was proposed for website prefetching [2], or for disseminating podcasts [3]. The performance of some of these systems have been analyzed from a hit-rate or delay standpoint [4], [5] for the case of a persistent demand.

A. The impact of delay on p2p mobile networks

Representing the effect of delay through a utility function has been applied to different areas of networking, including congestion control[6], and wireless scheduling[7]. In the context of opportunistic networks, Utility functions are primarily used as local states variables. The opportunistic routing protocol PROPHET [8] uses past information to predict delivery probability. The RAPID protocol generalizes this principle into an inference algorithm which accounts for several metrics related to delay [9], while CAR [10] proposes to use Kalman filter to improve the prediction's accuracy. In the same context, the impact of using different utility functions has been analyzed for single-copy routing scheme [11], to optimize buffer management [12], or the use of error-correcting code [13]. In the context of publish-subscribe applications, the same use of utility function was

J. Reich is with Department of Computer Science, Columbia University, NY, USA. reich@columbia.edu

A. Chaintreau is with Thomson, Paris Research Lab, France. augustin.chaintreau@thomson.net

introduced to either predict user future demands [14], or leverage uneven distribution of demands and proximity between users [15], [16]. Other advanced cache management protocols includes using filters [17] and social relationships between mobile users in community [18].

In general, such use of utility function helps the system to distinguish on the fly which intermediate nodes is the most likely to succeed (*i.e.,* for the unicast case, progressing towards the destination, or, for the pub-sub application, facilitating dissemination to subscribing nodes). The performance of all these schemes are in general difficult to analyze due to their complexity, and the interaction between local decisions using estimated utility and the global effect on network performance. Our work significantly departs from this closely-related work in two ways. The first is that instead of using (local) utility as an *intermediate* quantity used to estimate one or several parameters informing protocols, we take (global) utility as an *end-measure* for network efficiency (*i.e.,* the system's performance as it is perceived by users in aggregate). At no time during the course of the protocols is the utility estimated, but we rather wish to follow the effect of using different simple replication protocols on the global utility of the network, that is the objective to maximize. The second difference is that we account for a general behavior of users with regard to delay, as the global utility (or social welfare) is a function of any individually experienced delay-utilities (previous work either ignores user impatience or implicitly accounts for it using a fixed step function). This approach permits us to precisely define the optimal cache allocation as a function of user impatience, and show for the first time that simple reactive protocols may approach this optimal.

B. Replication protocols

Replication protocols were first introduced for unstructured peer-to-peer systems deployed on wired networks, as a way to increase data availability and hence to limit search traffic [19], [20]. Assuming that nodes search for files in random peers, it was shown [19] that for each fulfilled request, creating replicas in the set of nodes used for the search (*i.e., path-replication*) achieves a square root allocation: a file i requested with probability p_i has a number of replicas proportional to $\sqrt{p_i}$ at equilibrium. This allocation was shown to lead to an optimal number of messages overall exchanged in the system. Assuming that nodes use an expanding ring search, an allocation where each file is replicated in proportion of its probability p_i was shown to be optimal [20]. The meeting between unpredictable mobile nodes can in some sense be compared to a random search, however, we are not aware of any previous work analytically studying the performance of replication algorithms in this context.

Our results indicate that similar replication techniques can be used for a peer-to-peer system deployed on top of opportunistic contacts between mobile devices. Indeed we even show that replication can be tuned to approach the optimal utility. However, our results also indicate that this should be done wisely. Firstly, because the impatience of users (which arise because search delays are not negligible) greatly impact which replication strategy is the best choice. Secondly, if replication actions are unevenly delayed the system may fail to converge to the correct allocation, if at all. Consequently mechanisms are needed to ensure all replication delays occur evenly.

We heard recently about a parallel on-going effort to characterize a related channel selection problem [21]. The algorithm proposed in this case uses an estimate of dissemination time and a Metropolis-Hasting adaptive scheme. One difference between the two approaches is that we show, because the optimal allocation satisfies a simple balance condition, that even simple algorithms which do no maintain any estimates of dissemination time or current cache allocation are optimal for a known delay-utility function. Another difference is that we also prove that the submodularity property for the cache allocation can be established even when contacts and delay-utility functions are not homogeneous.

III. Efficiency of P2P caching

Each node in the P2P system may be a *client*, a *server*, or both. The set of client nodes is denoted by $\mathcal{C}$, we generally denote its size by N. Each client demands and consumes content as described in Section III-B. The set of all server nodes is denoted by $\mathcal{S}$. Servers maintain a cache in order to make it available to interested clients (when such clients are met). This includes in particular the two following scenarios:

- Dedicated nodes: server and client populations are separate (*i.e.,* $\mathcal{C} \cap \mathcal{S} = \emptyset$).
- Pure P2P: all nodes are server and client (*i.e.,* $\mathcal{C} = \mathcal{S}$).

The main variable of interest in the system is the content of the cache in all server nodes. In this section we assume it to be fixed; in practice this dynamically evolves through a replication protocol, as will be seen later in section V.

For any item i and m in $\mathcal{S}$, we define $x_{i,m}$ to be one if server node m possesses a copy of item i, and zero otherwise. The matrix $\mathbf{x} = (x_{i,m})_{i \in I, m \in \mathcal{S}}$ represents the state of the global distributed cache. We denote the total number of replicas of item i present in the system by $x_i = \sum_{m \in \mathcal{S}} x_{i,m}$.

In the rest of this paper, we assume that all servers have the same cache size so that they can contain up to ρ content items.It follows that a content allocation $\mathbf{x}$ in server nodes is feasible if and only if:

$$\forall m \in \mathcal{S}\,, \sum_{i \in I} x_{i,m} \leq \rho\,.$$

A. Representing Impatience as Delay-utility

The term *impatience* refers to the phenomenon that users become decreasingly satisfied (or increasingly dis-satisfied) with the delays they experience. A *delay-utility* function can be used to characterize this phenomenon of user impatience in analytic terms, where the value of this function is monotonically decreasing with time (as increasing delay will not translate into increasing satisfaction). The value $h_i(t)$ denotes the gain for the network resulting from delayed fulfillment of a request for item i when this occurs t time

units after the request was created. This value can be negative, which denotes that this delayed fulfillment generates a disutility, or a cost for the network. Note that t is related here to user's waiting time, not to the time elapsed since the creation of the item (which we do not account for).

We now present several examples of delay-utility functions corresponding to different perceptions of the performance of a P2P caching system by the users.

Advertising Revenue Assuming content items are videos starting with embedded advertisements, and that the network provider receives a constant unit revenue each time a commercial is watched by a user (a potential business plan for the VideoForU startup scenario mentioned in introduction). In this case, the delay-utility function simply denotes the probability that a user watches a given video when she receives the content t time after it was requested. Two possible function families modeling this situation are:

- Step function: $h_\tau^{(\mathrm{s})} : t \mapsto \mathbb{I}_{\{t \leq \tau\}}$.
- Exponential function: $h_\nu^{(\mathrm{e})} : t \mapsto \exp(-\nu t)$.

The former models a case where all users stop being interested in seeing the item after waiting for the same amount of time. In the second case, the population of users is more mixed: at any time, a given fraction of users is susceptible to losing interest in the content.

Time-Critical Information Assuming the content exchanged by nodes deals with an emergency, or a classified advertisement for a highly demanded and rare product (*i.e.,* a well located apartment). In such cases, as opposed to the previous model the value of receiving this piece can start from a high value but very quickly diminish. It is possible to capture such a behavior by a delay-utility presenting a large reward for a prompt demand fulfillment.

- Inverse power: $h_\alpha^{(\mathrm{p})} : t \mapsto \dfrac{t^{1-\alpha}}{\alpha - 1}$. with $\alpha > 1$

Note that the value of delivering an item immediately in this case is arbitrarily large ($h(0^+) = \infty$). Such immediate delivery can occur when a node is both a server and a user, as the local cache may already contain the item requested. To exclude this case, we restrict the use of such delay-utility functions to the Dedicated node case.

Waiting Cost In some situations, such as a patch needed to use or update a particular application, users may request for an item and insist on receiving it no matter how long it takes, becoming with time increasing upset because of tardy fulfillment. As an example, the time a user spent with an outdated version of a software application may be related with the risk of being infected by a new virus, and hence incurring a high cost. One can consider to represent such cases a delay-utility function that grows increasingly more negative with time, corresponding to a cost for the user and the network. The linear penalty delay-utility function (*i.e.,* waiting time) is the most intuitive of these. However, this function is but one in a family of penalizing delay-utility functions:

- Negative power: $h_\alpha^{(\mathrm{p})}$ as above with $\alpha < 1$
- Negative Logarithm: $h_1^{(\mathrm{p})} : t \mapsto -\ln(t)$.

The negative logarithm corresponds to the limit as α approaches 1. It features both a high value for fast fulfillment of request and a negative cost becoming unbounded as waiting time grows.

We plot on Figure 1 illustration of delay-utility functions for the three cases presented above.

B. Client Demand

Clients register their demand for content in the form of *requests*. As in previous work, we assume that the process of demand for different items follows different rates, reflecting differing content popularity. We denote by d_i the total rate of demand for item i. The probability $\pi_{i,n}$ reflects the relative likeliness of demand arising at node n, where $\sum_{n \in \mathcal{C}} \pi_{i,n} = 1$. In other words, node n creates a new request for item i with a rate equal to $d_i \pi_{i,n}$. One can generally assume that different populations of nodes have different popularity profile, generally captured in the values of $\pi_{i,n}$. Otherwise, we can assume to simplify that items, especially the ones with the highest demand, are popular equally among all network nodes. This corresponds to the case where $\pi_{i,n} = 1/|\mathcal{C}|$.

An examples of demand distributions is

- Pareto: with parameter $\omega > 0$: $d_i \propto i^{-\omega}$ for all $i \in I$.

In the rest of this paper, we will assume any arbitrary values of d_i.

C. Node Mobility

As all nodes (whether client or server) move in a given area, they occasionally meet other nodes - these meetings provide the opportunity for replication of cache content and fulfillment of outstanding requests. For simplicity and as a way to compare different P2P caching schemes, we focus on a case where contacts between clients and server nodes follow independent and memory less processes. In other words, we neglect the time dependence and correlation between meeting times of different pairs which may arise due to complex properties of mobility. In that case the process of contacts between two nodes m and n is entirely characterized by their contact intensity (the number of contacts between them per unit of time), which we denote by $\mu_{m,n}$.

Hence we will consider the following simple contact model:

- Continuous time: The system evolves in an asynchronous manner, so that events may occur in continuous time. We assume that node contacts occur according to a Poisson Process with rate $\mu_{m,n}$ (for $m \in \mathcal{S}$, $n \in \mathcal{C}$).

The system is said to follow *homogeneous contacts* if we have $\mu_{m,n} = \mu$ for all nodes $m \in \mathcal{S}$ and $n \in \mathcal{C}$. This case corresponds to a population of nodes with similar characteristics where all meeting are equally likely, as for instance it may be between the participants of a special event.

D. Content allocation objective

Demand arises in our P2P system according to content popularity, and is served as a function of mobility and

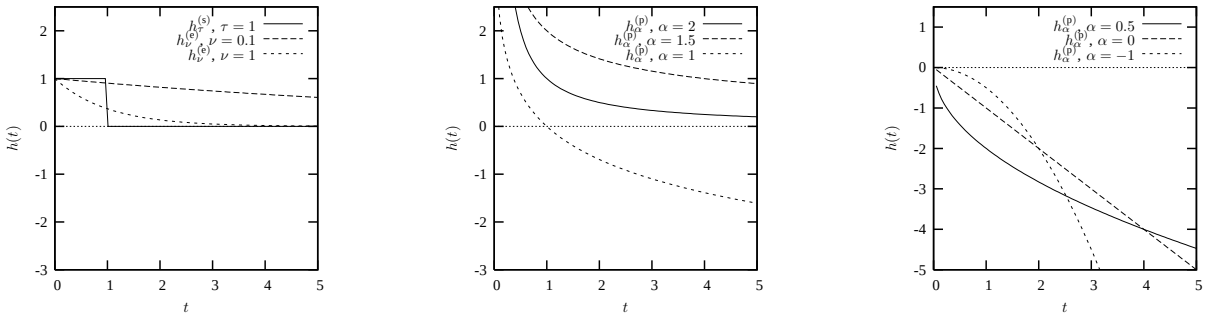

Fig. 1. Different delay-utility functions

content availability in caches, captured through variables $\mathbf{x} = (x_{i,m})_{i\in I, m\in\mathcal{S}}$.

We define $U_{i,n}(\mathbf{x})$ to be the expected gain generated by a request for item i created by client node n. Following our model of users's impatience, this expected gain is equal to $\mathbb{E}[h_i(Y)]$ where Y denotes the time needed to fulfill this request, which itself critically depends on the availability of item i in servers's caches.

The total utility perceived by all clients in the system, also called *social welfare*, may then be written as:

$$U(\mathbf{x}) = \sum_{i\in I} d_i \sum_{n\in\mathcal{C}} \pi_{i,n} U_{i,n}(\mathbf{x})\,. \tag{1}$$

A good allocation of content to cache a choice of $\mathbf{x}$ that results in a high social welfare. Note that this objective combines the effect of delay on gain perceived by users, the popularity of files, as well as the cache allocation.

In the remaining of this section, we derive expressions for $U_{i,n}(\mathbf{x})$ and $U(\mathbf{x})$ using the *differential delay-utility function*.

Differential delay-utility function We denote this function by c_i. It is defined as

$$c_i(t) = -\frac{dh_i}{dt}(t)\,.$$

The values of $c_i(t)$ is always non-negative as h_i is a non-increasing function. It represents the additional loss of utility, which is incurred per additional unit of time spend waiting (resp. the loss of utility incurred for waiting another time slot).

General expression for $U_{i,n}(\mathbf{x})$

Following a slight abuse of notation, we set by convention $x_{i,n} = 0$ when n is not a server node (*i.e.,* $n \notin \mathcal{S}$). With this notation, we find the following expressions for $U_{i,n}$.

Lemma 1: $U_{i,n}(\mathbf{x})$ may be expressed as

$$h_i(0^+) - (1 - x_{i,n}) \int_0^\infty \exp\left(-t \sum_{m\in\mathcal{S}} x_{i,m}\mu_{m,n}\right) c_i(t)dt\,.$$

The proof follows from the memory less property of contacts and the expectation as obtained in integration by part. The term $(1 - x_{i,n})$ deals with possible immediate fulfillment (*i.e.,* request created by a node that already contains this item in its local cache). For more details, see [1].

Homogeneous contact case If we assume homogeneous node contacts (*i.e.,* when $\mu_{m,n} = \mu$), the general expressions above lead to simpler closed form expressions. In particular, the gain or utility depends on $(x_{i,n})_{i\in I, n\in\mathcal{S}}$ only via the number of copies present in the system for each item $(x_i)_{i\in I}$.

First, in the dedicated node case (*i.e.,* $\mathcal{S}\cap\mathcal{C} = \emptyset$), we have:

$$U(\mathbf{x}) = \sum_{i\in I} d_i \left(h(0^+) - \int_0^\infty e^{-t\mu x_i} c_i(t)dt\right)\,. \tag{2}$$

Similarly, for the pure P2P case, if we further assume that all $N = |\mathcal{C}| = |\mathcal{S}|$ nodes follow the same item popularity profile (*i.e.,* $\pi_{i,n} = 1/N$), we have:

$$U(\mathbf{x}) = \sum_{i\in I} d_i \left(h(0^+) - \left(1 - \frac{x_i}{N}\right) \int_0^\infty e^{-t\mu x_i} c_i(t)dt\right)\,. \tag{3}$$

All these expressions follows from a simple application of Lemma 1 (see [22] for complete details).

IV. Optimal Cache Allocation

The *social welfare* defined in the previous section offers a measure of the efficiency of cache allocation which captures users's requests and impatience behavior. In this section we wish to solve the following social welfare optimization:

$$\max\left\{ U(\mathbf{x}) \;\middle|\; x_{i,n} \in \{0,1\}\,, \forall n \in \mathcal{S}, \sum_{i\in I} x_{i,n} \le \rho \right\}\,. \tag{4}$$

A. Submodularity, Centralized computation

A function f that maps subset of $\mathcal{S}$ to a real number is said *sub-modular* if it satisfies the following property: $\forall A \subseteq B \subseteq \mathcal{S}\,, \forall m \in \mathcal{S}\,,\ f(A\cup\{m\}) - f(A) \ge f(B\cup\{m\}) - f(B)$. This property generalizes to set functions the concavity property defined for continuous variable. It may be called "diminishing return" as, if the function f is non-decreasing, it states that in order to maximize the value of f, the relative improvement obtained when including new element diminishes as the set grows.

Theorem 1: For any item i and node n, $U_{i,n}$ is submodular. As a consequence U is submodular.

This result can be interpreted intuitively. On the one hand, in order to increase the value of $U_{i,n}$, creating a new copy of item i (*i.e.,* including a new element in the set of servers containing a copy of i) always reduce delays and hence increase utility. On the other hand the *relative* improvement obtained when creating this copy depends on the number of

copies of i already present, and it diminishes as the item is more frequently found. What is perhaps less obvious is that this result holds to any mixed client/server population of nodes, heterogeneous contact processes, and any arbitrary popularity profile.

The proof of this result uses the general expression for $U_{i,n}$ found in Lemma 1 and a few observations: First, that the expression inside the integral multiplying the differential impatience function is a supermodular non-increasing and non-negative function of the set of server containing i. Second, that since the differential impatience function is positive, all these properties apply to the integral itself. Finally, that the product with $(1 - x_{i,n})$ preserve the supermodular non-increasing and non-negative properties. A complete formal proof can be found in [22].

In the case of homogeneous contact rates, we can obtain an even stronger results, as the social welfare only depends on the amount of replicas for each item, and not on the actual subset of nodes that possess it.

Theorem 2: In the homogeneous contact case, $U(\mathbf{x})$ is a concave function of $\{\ x_i \mid\ i \in I\ \}$.

The optimal values of $\{\ x_i \in \{0, 1, \cdots, |\mathcal{S}|\} \mid\ i \in I\ \}$ are found by a greedy algorithm that uses at most $O(|I| + \rho|\mathcal{S}|\ln(|I|))$ computation steps.

Moreover, the solution of the *relaxed* social welfare maximization (*i.e.,* maximum value of $U(\mathbf{x})$ when $(x_i)_{i\in I}$ are allowed to take real value) can be found by gradient descent algorithm.

The concavity property is here not surprising, as it corresponds to submodularity when the function is defined using continuous variables rather than a set. Formally, the arguments used to prove this result are quite similar to the previous proof: one should take advantage of previous expressions which feature the product with the differential impatience function, and then use the fact that the family of convex non-negative non-increasing functions is closed under product.

The greedy algorithm follows a simple operation repeated once for each copy that can be cached ($\rho|\mathcal{S}|$ steps in total): at each time step from the current cache allocation, it adds a copy for the item that brings the most significant relative increase in utility (assuming there does not exist already $|\mathcal{S}|$ copies of this item). By doing so, the algorithm is likely to select first popular items. As the popular items fill the cache with copies, the relative improvement obtained for each additional copy diminishes, and the greedy rule will choose to create copies for other less popular items. The diminishing return property ensures that this greedy algorithm selects the optimal cache allocation. A formal proof of these results can be found in [22].

B. Characterizing the optimal allocation

We show in this section that the solution of the relaxed optimization problem satisfies a simple equilibrium condition.

Property 1: We consider the continuous time contact and dedicated node case. Let $\tilde{x}$ be the solution of the relaxed social welfare maximization (as defined in Theorem 2). Then

$$\forall i, j\ ,\ \tilde{x}_i = |\mathcal{S}|\ \text{ or }\ \tilde{x}_j = |\mathcal{S}|\ \text{ or }\ d_i \cdot \varphi(\tilde{x}_i) = d_j \cdot \varphi(\tilde{x}_j)\ .$$

where φ is defined as $\varphi : x \mapsto \displaystyle\int_0^\infty \mu t e^{-\mu t x} c(t) dt$.

This property states that, at the optimal solution of the relaxed cache allocation problem, the amount of copies created for all items depends on their popularity exactly in the same way: via a unique function φ defined *independently* of i. This equality holds only when the number of copies is not limited by the number of servers, otherwise it becomes an inequality.

V. Distributed Optimal Schemes

In this section, we demonstrate that one does not need to maintain global information, or know the demand of items *a priori*, to approach an optimal cache allocation. We show that a simple reactive protocol, generalizing replication techniques introduced in the P2P literature, are able to approach the optimal allocation using only local information.

A. Query Counting Replication

We propose a general class of distributed schemes, that we call *Query Counting Replication (QCR)*. QCR *implicitly adapts* to the current allocation of data and collection of requests, without storing or sharing explicit estimators. QCR achieves this by keeping a *query count* for each new request made by the node. Whenever a request is fulfilled for a particular item, the final value of the query counter is used to regulate the number of new replicas made of that item. The function ψ that maps the value of the query counter to the amount of replicas produced is called the *reaction* function.

This principle generalizes path replication [19] where $\psi(y)$ was a linear function of y.

B. Tuning replication for optimal allocation

We now describe how to choose the reaction function ψ depending on users's impatience. We first observe that the expected value of the query counter for different item i is proportional to $1/x_i$, since whenever a node is met there is roughly a probability $x_i/|\mathcal{S}|$ that it contains item i in its cache. Hence, we can assume as a first order approximation that approximately $\psi(|\mathcal{S}|/x_i)$ replicas are made for each request of that items. Inversely, as a consequence of random replacement in cache, each new replicas being produced for any items erases a replica for item i with probability $x_i/(\rho|\mathcal{S}|)$. As a consequence, the number of copies for each item follows the system of differential equations:

$$\forall i \in I\ ,\ \frac{dx_i}{dt} = d_i \cdot \psi(\frac{|\mathcal{S}|}{x_i}) - \frac{x_i}{\rho|\mathcal{S}|} \cdot \sum_{j\in I} d_j \psi(\frac{|\mathcal{S}|}{x_j})\ . \quad (5)$$

Assuming the system converges to a stable steady state, the creation of copies should compensate exactly for their deletion by replacement. In other words a stable solution of this equation satisfies

$$\forall i \in I\ ,\ d_i \frac{1}{x_i} \cdot \psi(\frac{|\mathcal{S}|}{x_i}) = \frac{1}{\rho|\mathcal{S}|} \cdot \sum_{j\in I} d_j \psi(\frac{|\mathcal{S}|}{x_j})\ .$$

Note that the RHS is a constant that does not depend on i anymore, so that this implies

$$\forall i, j \in I\,,\; d_i \frac{1}{x_i} \cdot \psi(\frac{|\mathcal{S}|}{x_i}) = d_j \frac{1}{x_j} \cdot \psi(\frac{|\mathcal{S}|}{x_j})\,.$$

In other words, the steady state of this algorithm satisfies the equilibrium condition of Property 1 if and only if we have: $\forall x > 0\,,\; \frac{1}{x}\psi(\frac{|\mathcal{S}|}{x}) = \varphi(x)$ where φ is defined as in Property 1. Equivalently,

$$\forall y > 0\,,\; \psi(y) = \frac{|\mathcal{S}|}{y}\varphi(\frac{|\mathcal{S}|}{y})\,.$$

Property 2: The steady state of QCR satisfies the equilibrium condition of Property 1 if and only if

$$\psi(y) \propto |\mathcal{S}|/y \int_0^\infty \mu t e^{-\mu \frac{t|\mathcal{S}|}{y}} c(t) dt\,.$$

The upshot of this result is that as long as the delay-utility function representing user impatience is known, we can always determine the number of copies QCR must make to drive the allocation towards its optimal.

VI. Conclusion

Our results focus on a specific feature which makes P2P caching in opportunistic network unique: users' impatience. From a theoretical standpoint, we have shown that optimality is affected by impatience but can be computed and moreover satisfies an equilibrium condition. From a practical standpoint, we have seen that it directly affects which replication algorithm should be used by a P2P cache. Passive replication, ending in proportional allocation, can sometimes perform very badly, but one can tune an adaptive replication scheme to approach the performance of the optimal, based only on local information.

We believe these results may serve as a stepping stone to address other unique specific characteristics of P2P caching in opportunistic system, in particular they offer a reference case from which one can study (1) the impact of heterogeneity and complex mobility property more systematically, (2) clustered and evolving demands in peers, as distributed mechanism like QCR naturally adapts to a dynamic demand. Another important aspect that remains to be addressed is how to estimate the impatience function implicitly from user feedback, instead of assuming that it is known.

VII. Acknowledgment

We would like to gratefully acknowledge Nikodin Ristanovic, Stratis Ioannidis and Laurent Massoulié, for their insightful comments and their help during the preparation of this work.

References

[1] J. Reich and A. Chaintreau, "The age of impatience: optimal replication schemes for opportunistic networks," in *Proc. of ACM CoNEXT*, 2009.

[2] M. Papadopouli and H. Schulzrinne, "Effects of power conservation, wireless coverage and cooperation on data dissemination among mobile devices," in *Proc. ACM MobiHoc*, 2001.

[3] V. Lenders, M. May, and G. Karlsson, "Wireless ad hoc podcasting," in *Proc. IEEE SECON*, 2007.

[4] C. Lindemann and O. P. Waldhorst, "Modeling epidemic information dissemination on mobile devices with finite buffers," in *Proc. ACM Sigmetrics*, 2005.

[5] G. Karlsson, V. Lenders, and M. May, "Delay-tolerant broadcasting," *IEEE Transactions on Broadcasting*, vol. 53, pp. 369–381, 2007.

[6] S. Kunniyur and R. Srikant, "End-to-end congestion control schemes: utility functions, random losses and ecn marks," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 689–702, 2003.

[7] P. Liu, R. Berry, and M. Honig, "Delay-sensitive packet scheduling in wireless networks," *Proc. of WCNC 2003*, vol. 3, 2003.

[8] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," *SIGMOBILE Mobile Computing and Communication Review*, vol. 7, no. 3, 2003.

[9] A. Balasubramanian, B. Levine, and A. Venkataramani, "DTN Routing as a Resource Allocation Problem," in *Proc. ACM SIGCOMM*, August 2007. [Online]. Available: http://prisms.cs.umass.edu/brian/pubs/balasubramanian.sigcomm.2007.pdf

[10] M. Musolesi and C. Mascolo, "Car: Context-aware adaptive routing for delay-tolerant mobile networks," *IEEE Transactions on Mobile Computing*, 2009.

[11] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Efficient routing in intermittently connected mobile networks: the single-copy case," *IEEE/ACM Trans. on Netw.*, vol. 16, no. 1, Feb 2008. [Online]. Available: http://portal.acm.org/citation.cfm?id=1373452.1373458

[12] A. Krifa, C. Barakat, and T. Spyropoulos, "Optimal buffer management policies for delay tolerant networks," *Proc. of IEEE SECON*, 2008.

[13] S. Jain, M. Demmer, R. Patra, and K. Fall, "Using redundancy to cope with failures in a delay tolerant network," *ACM SIGCOMM Computer Communication Review*, Jan 2005. [Online]. Available: http://portal.acm.org/citation.cfm?id=1080106&dl=GUIDE,

[14] G. Sollazzo, M. Musolesi, and C. Mascolo, "Taco-dtn: a time-aware content-based dissemination system for DTN," in *Proc. ACM MobiOpp*, 2007.

[15] C. Boldrini, M. Conti, and A. Passarella, "Contentplace: social-aware data dissemination in opportunistic networks," in *Proc. ACM MSWiM*, 2008.

[16] P. Costa, C. Mascolo, M. Musolesi, and G. Picco, "Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks," *IEEE Jsac*, vol. 26, no. 5, pp. 748–760, June 2008.

[17] J. Greifenberg and D. Kutscher, "Efficient publish/subscribe-based multicast for opportunistic networking with self-organized resource utilization," in *Proc. AINAW*, 2008.

[18] E. Yoneki, P. Hui, S. Chan, and J. Crowcroft, "A socio-aware overlay for pub/sub communication in DTN," in *Proc. ACM MSWiM*, 2007.

[19] E. Cohen and S. Shenker, "Replication strategies in unstructured peer-to-peer networks," *SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 4, pp. 177–190, 2002.

[20] S. Tewari and L. Kleinrock, "Proportional replication in peer-to-peer networks," in *Proc. INFOCOM*, 2006.

[21] L. Hu, J.-Y. L. Boudec, and M. Vojnovic, "Optimal channel choice for collaborative ad-hoc dissemination," MSR, Tech. Rep. MSR-TR-2009-26, 2009.

[22] J. Reich and A. Chaintreau, "The age of impatience: optimal replication schemes for opportunistic networks," Thomson, Tech. Rep. CR-PRL-2009-06-0001, 2009, available at: www.thlab.net/~chaintre/pub/reich09age.TR.pdf.