The Age of Impatience: Optimal Replication Schemes for Opportunistic Networks

Joshua Reich[♯], Augustin Chaintreau[♭]

[♯]Department of Computer Science, Columbia U., NY, email: reich@columbia.edu
[♭]Thomson, Paris, email: augustin.chaintreau@thomson.net

Thomson Technical Report
Number: **CR-PRL-2009-06-0001**
Date: June 30th 2009

Abstract: We study mobile P2P content dissemination schemes which leverage the local dedicated caches provided by hand-held devices (*e.g.,* smart-phones, PDAs) and opportunistic contacts between these devices. In such a distributed environment, each opportunistic contact represents a *current* opportunity to selectively replicate local cache content to fulfill *future* demand. The efficiency of current replication decisions, though, depends on both the arrival of new demand, as well as users's *impatience* (*i.e.,* as time passes without fulfillment, the demanding peer loses interest with increasing probability). Consequently, we measure the efficiency of a replication scheme based on how well its replication decisions enable the timely fulfillment of demand. Moreover, in many scenarios this must be achieved solely through use of locally available information.

Our work makes several important contributions: (1) We are the first to consider the impact of users's impatience on optimal content dissemination schemes; we characterize the optimal allocation using a generic impatience function and show it to be efficiently computable. In some cases, this optimal allocation can be known in closed form. (2) Furthermore, we develop a reactive distributed algorithm, *Query Counting Replication (QCR)* that produces the optimal allocation for any impatience function, based on local information and no knowledge of users's demands. We discover that when QCR is implemented naively in the opportunistic context, pathological instability of the global cache allocation will result. In response, we provide a novel mechanism, *Mandate Routing*, that resolves these pathologies. (3) Finally, we validate these techniques on real-world contact traces, demonstrating the robustness of our results in the face of heterogeneous meeting rates and bursty contacts.

THOMSON

1. INTRODUCTION

As smartphones capable of displaying, storing, and transmitting media content continue to proliferate, the problem of how to distribute content becomes ever more timely. A common approach to content dissemination leverages a centralized model in which content providers send content directly to users through infrastructure. In fact, that is how practically all existing mobile content dissemination systems operate today. However, this approach is both *expensive* for the content provider and *inefficient* from a networking perspective as it leaves unused the potentially very large quantity of bandwidth available for content exchange between smartphones within transmission range of their short range radios (e.g., Bluetooth, 802.11). Consequently, in this work we explore *opportunistic* protocols for content dissemination that selectively replicate local cache content when peers meet - in order to efficiently fulfill demand, while harnessing the full power of the mobile network.

To model this problem, we consider a set of mobile nodes carried by users, each equipped with a finite dedicated cache. The contents of such a dedicated cache can be used to fulfill the requests of other peers when the opportunity arises. The challenge is to allocate the caches of all peers (equivalently, manage the *distributed global cache*) so as to maximize the expected efficiency of the system.

However, in contrast with traditional Peer-to-Peer (P2P) networks, the time elapsed between demand arrival and its fulfillment is far from negligible. Consequently, it is possible that users may exhibit *impatience* - as time passes without a user's demand being fulfilled, there is an increasingly greater likelihood that the user will become impatient and lose interest or even become angry. Our work is the first to consider the impact of user impatience on the efficiency of a content dissemination strategy. We do so by incorporating a generic impatience function, whose only constraint is that it is monotone non-increasing, into the network's objective measure.

While our analytic aim is to describe the optimal allocation of the distributed global cache (i.e., the allocation that maximizes the efficiency of demand fulfillment in face of user impatience), we are also concerned with developing mechanisms capable of achieving or approximating such allocations. The same factors that form the basis for our allocation problem (unpredictable mobility and resultant sporadic contacts) also imply that it may be difficult to gather global knowledge of the network's state. Consequently, we seek to develop distributed mechanisms capable of producing optimal or approximately optimal allocations using as little knowledge as possible.

Moreover, unpredictable peer contacts also impact the deployment of such distributed reactive mechanisms. In the opportunistic context, we find that when we naively deploy a distributed mechanism that replicates cache items in response to demand and fulfillment events, the network's unpredictable contact structure causes some pieces of content to be favored over others. Thus a mechanism that approaches the correct allocation in more traditional peer-to-peer (P2P) settings becomes pathological in the opportunistic context. Along with discovering this phenomenon, we also develop a technique for resolving it, that is lightweight and easy to implement.

In this paper, we make the following contributions:

1. We are the first work to consider the impact of impatience on optimal content dissemination schemes. We propose a general model to capture this impact and show that under very general assumptions on the impatience function (namely, that is is monotone) one can define an optimal cache allocation (Section 3). Furthermore, we demonstrate that this optimal is unique and can be computed efficiently in a centralized manner. We prove that is may vary, depending on the users's impatience function, between a uniform and a highly skewed allocation. Under the simplified assumption of homogeneous meeting rates, we show that the corresponding optimal cache allocation is known in closed form for a general class of impatience utility (Section 4).

2. Inspired by these results, we study the behavior of distributed cache algorithms, that do not know the popularity of content items or the current cache allocation. We propose a reactive algorithm, *Query Counting Replication (QCR)*, which generalizes protocols found in previous work. We prove that, if we only assume that the impatience function and the contact rate is known, QCR can be tuned to approach the optimal allocation at equilibrium. We also discover that when QCR is implemented naively in the opportunistic context, pathological instability of the global cache allocation will result. In response, we provide a novel mechanism, *Mandate Routing (MR)*, that resolves these pathologies and may have applicability to other opportunistic mechanisms (Section 5).

3. Finally, we validate these techniques on real-world contact traces, demonstrating the robustness of our analytic results (obtained under the simplified assumption of homogeneity) in the face of heterogeneous meeting rates and bursty contacts (Section 6).

These results demonstrate that users's impatience impacts the performance of P2P mobile opportunistic system, both from a theoretical standpoint (optimality) as well as for practical choices (replication protocols).

For clarity, we do not include the proof of the analytical result in the main body of the text, they can be found in Appendix A.

2. RELATED WORK

Networks that leverage local connection opportunities to communicate in a delay tolerant manner can be classified into two categories. The first category, featuring networks such as DieselNet[1] or KioskNet[15], involves nodes with scheduled or controlled routes, and routing protocols designed to communicate critical information with predictable latency. The second category contains network featuring unpredictable mobility [9, 5] that may be used in an opportunistic manner. In this case, it is infeasible to provide strict guarantees on message delivery time. However, the performance of many P2P applications may still benefit greatly from opportunistic contacts between the nodes. It is into this second category that the content dissemination problem we investigate here falls. Motivated by the need to understand the capability of such networks, we measure how such opportunistic systems can disseminate content while coping with user impatience.

Content dissemination through opportunistic contacts was first proposed in the 7DS peer-to-peer architecture[14]. PodNet [11] is another opportunistic system that focuses on the dissemination of podcasts, or series of content items on a channel. Publish/subscribe applications over opportunistic networks recently became an active research area. The performance of some of these systems have been analyzed from a hit-rate or delay standpoint [12, 10] for a persistent demand. Advanced cache management protocols proposed to replicate content using either filtering schemes [8], heuristic utility values based on prediction of user future demands [16] and/or graphs that leverage social relationships between mobile users [2, 18, 7]. The performance of these schemes are difficult to analyze due to their complexity.

Our work departs from previous formulations in that we account for the effect of user impatience. In other words, 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). This approach permits to precisely define the optimal cache allocation, and show for the first time that simple reactive protocols may approach this optimal.

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 [6, 17]. Assuming that nodes search for files in random peers, it was shown [6] 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 [17]. The meeting between unpredictable mobile nodes can in some sense be compared to a random search, however, we are not aware of any work who studied analytically 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 prove that this should be done wisely. Firstly, because the impatience of users (which arise because search delay are not negligible) greatly impacts which replication strategy is the best choice. Secondly, because 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. These new findings complement previous work and may be used as a stepping stone to design the performance of replication algorithm in more general network with heterogeneous meeting rates.

3. EFFICIENCY OF P2P CACHING

In this section, we describe a model for a mobile P2P caching system with impatient users. Nodes are occasionally connected as they come within wireless range; Some of them, possibly all, are able to store items to serve others when possible. The main result of this section is that the efficiency of this system as a whole can be measured in an objective function that allows a general model of user impatience.

Clients and Server nodes.

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 3.1. The set of all server nodes is denoted by $\mathcal{S}$. Servers cache content in order to make it available to interested clients (when such clients are met). Note that the same node can be both a client and a server, (*i.e.,* it maintains a dedicated local cache for serving other nodes' requests).

Our model includes in particular the two following scenarios of interest

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 dedicated nodes case resembles a managed P2P system, where delivery of content is assisted by special type of nodes (*e.g.,* buses or throwboxes[1], kiosks[15]). The pure P2P case denotes a cooperative setting where all nodes (*e.g.,* users's cell-phones[14, 11]) request content as well as help deliver data to others.

3.1 Client demand and Impatience

We implicitly assume that the efficiency of a P2P cache is going to be affected primarily by user behaviors. There are two important aspects to consider: firstly, which content is accessed with which frequency, and secondly, how users react to delays experienced while searching for this content.

We denote by I the set of items that can be requested by users in the system, they are indexed by i.

Demand, Popularity.

Demand for content items are created by clients in the form of *requests*. As in previous work, we assume that the process of demand for different items follow 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 $\pi_{i,n} = 1/|\mathcal{C}|$.

Without loss of generality we assume that items are sorted according to demand in decreasing order (*i.e.,* $d_i \geq d_j$ for $i \leq j$). Examples of demand distributions are

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

Uniform $d_i = d$ for all $i \in I$ (this is a special case of the above where $\omega = 0$).

The uniform demand case is unsuprisingly quite simple to deal with, as all items should receive an equal rate of the cache. In the rest of this paper, we will assume any arbitrary values of d_i. In simulation we will use a Pareto popularity distribution, generally considered as representative of content popularity.

Impatience function.

In contrast with previous work in P2P networks, P2P content dissemination over an opportunistic mobile network induces a non-negligible *delay* between the time a request is made by a client node and the time that it is fulfilled. This delay depends on the current cache allocation, as the request is fulfilled the next time the node has an opportunistic contact with a server possessing a replica of this item. Its main consequence is that it is important to account for the behavior of users with regard to this delay, or equivalently for users's *impatience*: by the time an item is received by a client, it may very well happen that this item is not relevant to that client any more (or even worse, tardy fulfillment has made that client actively angry).

Since users may follow several possible behaviors in different scenarios, we want a simple yet flexible model to account for their impatience. This is why we introduce for each item i a general *impatience function* h_i: the value $h_i(t)$ denotes the gain for the network resulting from delayed fulfillment of demand when the request was fulfilled t time units after it was created. Note that this value can be negative, which denotes that this delayed fulfillment generates a disutility, or a cost for the network. Since it is always preferable to fulfill demand as soon as possible, we can generally assume that h_i is a non-increasing function of time. In the rest of this paper, we derive results applying to any arbitrary non-increasing impatience function h_i.

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

One example of impatience function can be seen as an *advertising revenue*. Assume content items are videos starting with a commercial, and that the network provider receives a constant unit revenue each time a commercial is watched by a user. In this case, the impatience function simply denotes the probability that a user watches a given video when he/she receives the content t time after it was requested. Two examples of functions modeling this situation are:

Step function $h^{(\mathrm{s})}_{\tau} : t \mapsto \mathbb{I}_{\{t\leq\tau\}}$.

Exponential function $h^{(\mathrm{e})}_{\nu} : t \mapsto \exp(-\nu t)$.

The former models a case where all users stop being interested to see 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 lose interest for this content.

Another example of impatience function corresponds to *critical time information*, like information about an emergency, or advertisement for a few highly demanded and rare goods (well located apartments) that are likely to be sold very quickly. In this case, the impatience function presents a large reward for a prompt demand fulfillment. One example of functions reflecting this are:

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

As a third example, similar to a *waiting cost*, one can consider an impatience function that grows increasingly more negative with time. This corresponds to the fact that users get increasingly upset because of tardy fulfillment. Two possible choices of the impatience function for this case are

Negative Logarithm $h_1^{(\mathrm{p})} : t \mapsto -\ln(t)$.

Negative power $h_\alpha^{(\mathrm{p})}$ as above with $\alpha < 1$

We plot on Figure 1 illustration of impatience functions for the three cases presented above.

To simplify the presentation below, we consider first the case where h admits a finite limit at time $t = 0$, (*i.e.,* $h(0^+) < \infty$). This applies to all cases above except for the inverse power and the negative logarithm function. These functions can be considered as well, but they pose some particular technical challenges due to the singularity of the function in $t = 0$. In Appendix A.4, we prove that all the results of this paper hold for these functions as well, as long as $\alpha \leq 2$, in the dedicated node case.

3.2 Caches and Mobility

Content of server cache.

The main variable of interest in the system is the content of the cache in all server nodes. 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 of ρ identically sized content items. This is not a critical assumption and most of the following results can be extended to caches of differing sizes. From this assumption we deduce that a content allocation $\mathbf{x}$ in server nodes is feasible if and only if:

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

Mobility.

As nodes 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 simple 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. Our model can be more generally defined for any contact processes, as will be done in Section 6 for comparison. The memoryless assumption helps us to understand what are optimal strategies in a simple case before evaluating them using real traces for a complete validation of these trends. We consider two contact models:

Discrete time The system evolves in a synchronous manner, in a sequence of time slots with duration δ. For each time slot, we assume node contacts occur independently with probability $\mu_{m,n} \cdot \delta$.

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 $\mu_{m,n}$.

Note that when δ is small compared to any other time in the system, the discrete time model approaches the continuous time model. In this paper, we later focus on the analysis of the continuous time case, all our results apply to the discrete time case as well. Whenever space permits we will write results in both cases. Simulations results, which are based on discrete event processes, confirm the good match between our continuous time analysis and the discrete time dynamics of a real system (see Sec. 6).

The system is said to follow *homogeneous contacts* if we have $\mu_{m,n} = \mu$ for all nodes m and n. 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. Under the assumption of homogeneous contacts, we show later that the target optimal allocation and the behavior of distributed algorithm can be precisely understood. Simulation confirm that the trends we identify precisely in this simple case generalize to heterogeneous population.

3.3 Content allocation objective

Demand arises in this P2P system according to content popularity, and is served as a function of mobility and 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 $\mathbb{E}\left[h_i(Y)\right]$ 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 expressed in the following way:

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

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.

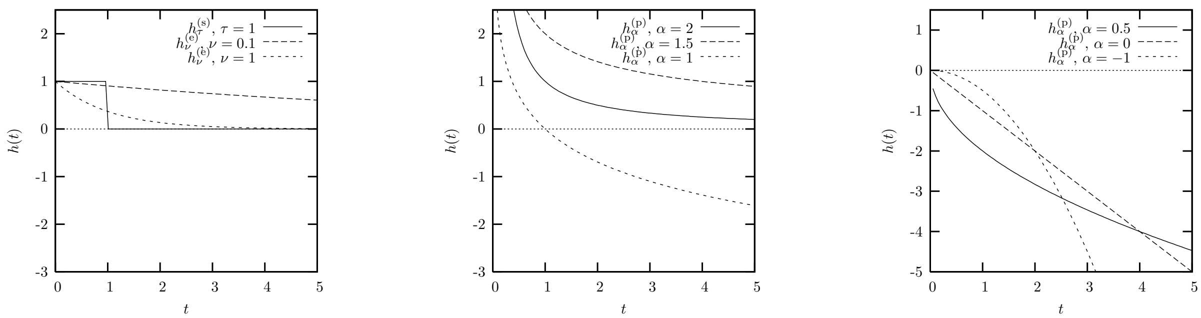


Figure 1: Different impatience functions

In the remaining of this section, we derive an expression for $U_{i,n}(\mathbf{x})$, based on the *differential impatience function*, which will be instrumental in deriving some of its properties.

Differential impatience function.

We denote this function by c_i for the continuous time contact model (resp. Δc_i for the discrete time contact model). These functions are defined by

$$c_i(t) = -\frac{dh_i}{dt}(t)\,, \text{ and } \Delta c_i(k\delta) = h_i(k\delta) - h_i\left((k+1)\,\delta\right)\,.$$

The values of $c_i(t)$ and $\Delta c_i(k\delta)$ are always positive as h_i is a non-increasing function. The value of c_i (resp. Δc_i) 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).

We present in the second line of Table 1 the expression for c_i for all the impatience functions introduced above. Note that when h_i is not derivable (like for the step function), it may happen that c_i is not defined as a function but as the derivative measure in the sense of the distribution.

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 $U_{i,n}$. the proof relies on memory less property of contacts and may be found in Appendix A.1.

LEMMA 1. *In the discrete time contact model, $U_{i,n}(\mathbf{x})$*

$$\text{is } \; h_i(\delta) - (1 - x_{i,n}) \sum_{k\geq 1} \prod_{m\in\mathcal{S}} \left(1 - x_{i,m}\mu_{m,n}\delta\right)^k c_i(k\cdot\delta)\,,$$

For the continuous time contact model, $U_{i,n}(\mathbf{x})$ is

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

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 (see Appendix A.1 for exact arguments). 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, respectively for the discrete time contact model and the continuous time contact model:

$$U(\mathbf{x}) = \sum_{i\in I} d_i \left(h(\delta) - \sum_{k\geq 1} (1 - \mu\delta)^{x_i k}\, c_i(k\cdot\delta) \right)\,. \quad (2)$$

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

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 for the two different models of contact process:

$$U(\mathbf{x}) = \sum_{i\in I} d_i \left(h(\delta) - \left(1 - \frac{x_i}{N}\right) \sum_{k\geq 1} (1 - \mu\delta)^{x_i k}\, c_i(k\cdot\delta) \right)\,. \quad (4)$$

$$U(\mathbf{x}) = \sum_{i\in I} d_i \left(h(0^+) - \left(1 - \frac{x_i}{N}\right) \int_0^\infty e^{-t\mu x_i} c_i(t)dt \right)\,. \quad (5)$$

4. 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:

$$\max \left\{ \; U(\mathbf{x}) \;\middle|\; x_{i,n} \in \{0,1\}\,, \forall n \in \mathcal{S}, \sum_{i\in I} x_{i,n} \leq \rho \; \right\}\,. \quad (6)$$

4.1 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 "diminishing returns" property: $\forall A \subseteq B \subseteq \mathcal{S}\,, \forall m \in \mathcal{S}\,,$ $f(A \cup \{m\}) - f(A) \geq f(B \cup \{m\}) - f(B)\,.$

The function $U_{i,n}(\mathbf{x})$ can be interpreted as a function that maps subset of $\mathcal{S}$ (*i.e.,* the subset of servers that possess a replica for item i) to a real number (the expected value of a request for item i created in client n). Similarly, U may be seen as a function that maps subset in $\mathcal{S} \times I$ (subsets denoting which servers possess which replica), to a real value (the social welfare).

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

Note that this result applies to any mixed client/server population of nodes, heterogeneous contact processes, and any arbitrary popularity profile. The proof (see Appendix A.2) is a consequence of the general expression for $U_{i,n}$ found in the previous subsection.

One can take in general advantage of submodularity to show that a greedy procedure builds a $(1-1/e)$-approximation of the maximum social welfare under capacity constraints (see [13]). Such an algorithm is used in Section 6 to find a cache allocation for heterogeneous contact traces.

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. We have (see Appendix A.2 for the proof).

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.*

4.2 Characterizing the optimal allocation

In the homogeneous contact case, whenever x_i only takes integer values, it can be difficult to grasp a simple expression for the allocation maximizing social welfare, as it is subject to boundary and rounding effect. However, when the number of servers is large, x_i may take larger values, in particular for popular items. Hence, the difference between the optimal allocation and the solution of the *relaxed* optimization (where $(x_i)_{i\in I}$ may take real values, as defined in Theorem 2) tends to become small. The latter is then a good approximation of the former. In addition, when the number of clients N becomes large, the difference between the dedicated node case and the pure P2P case tends to become negligible, as the correcting terms $(1 - \frac{x_i}{N})$ in Eq.(4) and (5) approaches 1.

We show in this section that the solution of the relaxed optimization problem satisfies a simple equilibrium conditions. Although we only derive this condition in the continuous time contact model, a similar condition can be found in the discrete case model.

THEOREM 3. *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). We have*

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

where we define φ as $\varphi : x \mapsto \int_0^\infty \mu t e^{-\mu t x} c(t) dt\,.$

The proof (see Appendix A.3) relies on the observation that, if the condition is not satisfied then it is possible to modify $\tilde{x}$ to remain in the capacity constraint and obtain an even larger social welfare.

As a consequence, if all items exhibit the same power impatience function ($h_i = h_\alpha^{(\mathrm{p})}$), the condition implies, for all item i that are within the boundary conditions (*i.e.,* $x_i < |\mathcal{S}|$), that $(\tilde{x}_i)^{2-\alpha} d_i$ is a constant independent of i. We deduce that the optimal solution of this relaxed problem resembles the distribution where $x_i \propto d_i^{\frac{1}{2-\alpha}}$.

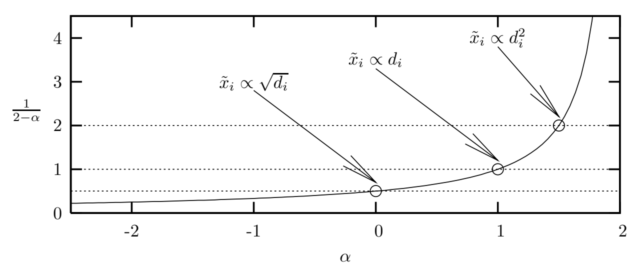


Figure 2: Coefficient of the optimal allocation for different power impatience functions.

5. DISTRIBUTED OPTIMAL SCHEMES

The previous section establishes that the cache allocation problem admits an optimal operating point, which may in some cases be known in closed form, and can always be computed in a centralized manner. When a highly available control channel is available, using such a centralized approach is feasible. However, in the absence of such a control channel, this approach might only be replicated noisily through the use of distributed estimators at each server node.

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 demonstrate that simple reactive protocols, generalizing replication techniques introduced in the P2P literature, are able to approach the optimal allocation using only local information. Interestingly, these results provide both a method that improves performance of P2P caching systems and can be deployed easily, but also explain why sometimes very simple replication strategies can perform quite well.

In order to build such low-overhead reactive protocol for the opportunistic setting, two particular challenges need be overcome:

- We must understand how to construct a replication strategy that reacts naturally to the demand for and availability of content (Section 5.1), while also properly adapting our replication strategy to impatience of users (Section 5.2). A successful strategy will allow us to approach the optimal efficiency when the system reaches equilibrium.
- We must ensure that the replication technique is implemented in such a way that ensures the convergence towards the equilibrium. This challenge proves to be non-trivial in the opportunistic context for the strategies we examine (Section 5.3).

5.1 Query Counting Replication

Distributed techniques for achieving optimal allocations are beneficial because they move the burden of data and message exchange off the centralized infrastructure and onto the nodes themselves. This can result in significant reduction of cost for provisioning service. Moreover this can make such provisioning *possible* where infrastructure isn't available.

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. We describe in Section 5.2 precisely how it should be set, given knowledge of user impatience.

As an example, Node a begins requesting a copy of item i. Each time node a subsequently meets a node, node a queries the node met for a copy of item i and increments the query counter associated with i. If after nine meetings a finally meets a node possessing a copy of item i, the request is fulfilled and, according to the above rule, a will create $\psi(9)$ replicas of this item and transmit them proactively to other nodes met when the opportunity arise. This principles generalizes path replication [6] where $\psi(y)$ was a linear function of y.

5.2 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:

$$\forall i \in I\,,\ \frac{dx_i}{dt} = d_i \cdot \psi(\frac{|\mathcal{S}|}{x_i}) - \frac{x_i}{\rho|\mathcal{S}|} \cdot \sum_{j\in I} d_j \psi(\frac{|\mathcal{S}|}{x_j})\,. \quad (7)$$

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 Theorem 3 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 Theorem 3. Equivalently,

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

THEOREM 4. *The steady state of QCR satisfies the equilibrium condition of Theorem 3 if and only if*

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

Table 1 summarizes the values of the different functions, in particular the optimal reaction function to use, for the different models of impatience introduced before. This table was computed for the continuous time and dedicated node case. A similar table can be found for the pure P2P case, it can be found in Appendix A.5 together with how to derive its exact expression. It is approximately equivalent to this one whenever N is large.

5.3 Mandate routing

Up to this point we have worked under the assumption that copies can be made more or less immediately,

Model	Step function	Exponential decay	Inv. Power ($\alpha < 1$)	Neg. Power ($1 < \alpha < 2$)	Neg. logarithm ($\alpha = 1$)
Impatience $h(t)$	$\mathbb{I}_{\{t \le \tau\}}$	$\exp(-\nu t)$	$\dfrac{t^{1-\alpha}}{\alpha - 1}$		$-\ln(t)$.
Diff. Impat. c	Dirac at $t = \tau$	density $t \mapsto \nu \exp(-\nu t)$	density $t \mapsto t^{-\alpha}$		
Gain $U(\mathbf{x})$	$\sum_i d_i(1 - e^{-\mu\tau x_i})$	$\sum_i d_i(1 - \frac{1}{1+\frac{\mu}{\nu}x_i})$	$\sum_i d_i x_i^{\alpha-1} \frac{\mu^{\alpha-1}\Gamma(2-\alpha)}{\alpha-1}$		$\sum_i d_i \ln(x_i) - \texttt{cst}$
Cond. φ (Thm 3)	$d_i \cdot \mu\tau e^{-\mu\tau x_i}$	$d_i \cdot \frac{\mu}{\nu}\left(1 + \frac{\mu}{\nu}x_i\right)^{-2}$	$d_i \cdot x_i^{\alpha-2}\mu^{\alpha-1}\Gamma(2-\alpha)$		
Reaction ψ (Thm 4)	$(\mu\tau\lvert\mathcal{S}\rvert/y)\, e^{-\frac{\mu\tau\lvert\mathcal{S}\rvert}{y}}$	$\left(2 + \frac{\nu}{\mu\lvert\mathcal{S}\rvert}y + \frac{\mu\lvert\mathcal{S}\rvert}{\nu}\frac{1}{y}\right)^{-1}$	$y^{1-\alpha}\left(\mu^{\alpha-1}\lvert\mathcal{S}\rvert^{\alpha-1}\Gamma(2-\alpha)\right)$		
Opt. allocation $\tilde{x}_i$	no closed form	no closed form	$x_i \propto d_i^{\frac{1}{2-\alpha}}$.		

Table 1: Several usual impatience functions and associated equilibrium and reaction functions.

as in classical wired P2P networks. However, in an opportunistic context this is far from true. Particularly:

- Copies can only be made when another node is met, which happens only sporadically. Creating a replica may also takes additional time as the node met may already have a replica of that message.
- Since cache are replaced randomly, it could be that, when a replica of the item needs to be produced, this item is no longer in the possession of the node desiring to replicate it.

Mandates & Pathologies.

Because replicas cannot be simply generated immediately, QCR mechanism deployed in an opportunistic context must inherently make (either implicitly or explicitly) a set of instructions for *future* replication of item i (*i.e.,* instructions to be used later, when the possibility for execution arises). We call such an instruction a *replication mandate* or *mandate* for short.

When a meeting occurs the mandate attempts to execute itself, but as we have already discussed, the circumstances may often not allow for its execution. This dependence of mandate execution on the state of the distributed cache may throw a monkey wrench in the dynamics outlined in Section 5.2 - for if the cache deviates to much from its expected state, the rates at which a given replica population evolves may be higher or lower than expected as well. As an example, if there are many fewer than expected copies of item i in the cache, and item i was erased by later random replacement, item i may rarely be present again, so that mandates may not be executed soon in the future. An item i that, in contrast, is more frequently found, will execute its mandate more quickly and hence continue to dominate. Consequently, if mandates are simply left at their node of origin the allocation produced by any given run of QCR can diverge significantly from the target allocation, resulting in a loss of social welfare.

Our solution.

In order to address the above pathology, we need to ensure that the number of replication actions taken for each message is proportionally the same as the number of mandates produced for that message. This could be done in several ways, which all boil down to one of the following two approaches: (1) Keep replicas near mandates, or (2) Keep mandates near replicas.

The former approach (*e.g.,* protecting items with current mandates from being erased by random replacement) violates the dynamics we are trying to protect and introduces significant implementation-level complexity - as we must now either replicate or protect against deletion particular messages based on locally existing mandates. While in practice these effects may be more or less severe, the second option of keeping mandates near replicas provides us with a way of solving the problem *that involves no addition biasing of the overwrites, nor requires any adjustment to the mechanism of cache adjustment.* Additionally mandates are by nature quite small pieces of data, so moving them about introduces little additional overhead in terms of communication and storage.

The mandate routing scheme used for the experiments shown in Section 6 is simple but can have significant impact as will be seen later. We assume that when two nodes meet, mandates are transfered with preference to the nodes possessing copies of the messages to be replicated. This ensures that most of the mandates that cannot be executed are soon transferred to appropriate nodes. Otherwise mandates are simply spread around - split evenly between the nodes. We demonstrate empirically that this simple modifications avoids divergence of QCR and is sufficient to converge towards an optimal point.

6. VALIDATION

We now conduct an empirical study of different replications algorithms in a homogeneous contact setting, as well as several traces in various mobility scenario. The goal of this study is threefold: first, we wish to validate empirically that the rationale behind our distributed scheme do actually converge close to the optimal value we predict, second, we wish to see quantitatively how it improves over simple heuristics, and last, we prove that the same scheme adapts well to contact heterogeneity present in the traces, as well as complex time statistics and dependencies between contacts.

6.1 Simulation settings

We have built a custom discrete-event, discrete-time simulator in C++ which given any input contact trace simulates demand arrival and the interactions of node meetings.

Data plots present below are the average of 15 or more trials with confidence interval corresponding to 5% and 95% percentiles. As said in Section 3.1 items are requested following Pareto distribution, here with parameter $\omega = 1$. By default we assume $\rho = 5$ To keep this section short, we present the most representative results we found below, other values of ω and ρ can be found in Appendix B.

Implementation of QCR.

When two nodes meet they first exchange meta data. If either nodes have outstanding requests for messages to be found in the other's cache, then each of those requests is fulfilled. For each fulfillment a gain is recorded by the simulator. As in the model this value is based on the age of this request and the impatience function. Nodes maintain query counting counter and makes a set of new mandates for each message fulfilled (as specified in Section 5.2). After fulfillment, the nodes then execute and route all of their eligible mandates.

Each item i has one *sticky* replica which cannot be erased. This implementation detail has the effect of ensuring that we do not enter an absorbing state in which certain messages have been lost through discrete stochastic effects. We believe it is a reasonable assumption for a fielded system, given that the initial seeder of a content item will likely keep that item permanently.

Competitor Algorithms.

We compare the performance of QCR against some with *perfect* control-channel information and the ability to set the cache *precisely and without restriction* to their desired allocation: **OPT** an approximation of the optimal obtained by a greedy algorithm optimizing Pb.(6). It is exactly optimal in the homogeneous case; **UNI**: memory is evenly allocated amongst all items; **SQRT**: memory allocation proportionally to the square root of the demand; **PROP**: memory allocation proportional to demand; **DOM**: all nodes contain the ρ most popular items.

6.2 Homogeneous contacts

We simulate a network of 50 nodes with 50 content items, meeting according to a rate $\mu = 0.05$. In this case the absolute value of μ plays no role in the comparison between different replication algorithm. As we wish to validate our analysis, we focus on the pure p2p case, which is the furthest from the analysis we conducted. In particular we tune the reaction function ψ according to Table 1 which has been derived from the dedicated node case. We did do because we expect all correcting terms to be small when N is not too small.

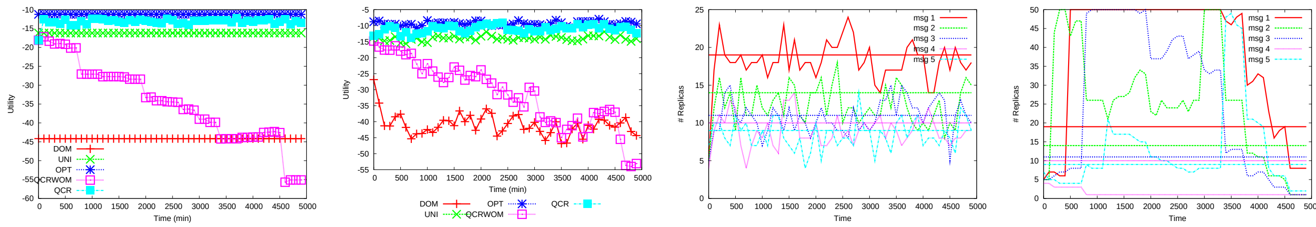
QCR with and without mandate routing.

Figure 3 illustrates the need to implement mandate routing in query based replication. It was obtained for the power impatience function with $\alpha = 0$, and a moderate cache size (5 items per node). This result is representative of all comparison where mandate routing was turned on and off. As the time of the simulation evolves we see that the utility (as estimated in expectation on (a), and observed from real fulfillment in (b)) dramatically decreases with time when QCR does not implement mandate routing. Further investigations have shown that simultaneously the amount of mandate diverges for item less frequently requested. We see on (d), where the number of replicas is shown for the five most requested items, that QCR without mandate routing systematically overestimates their share and sometimes. In contrast, the number of replicas with mandate routing fluctuates around the targeted value, and QCR quickly converges and stay near optimal utility.

Comparison with fixed allocations.

Figure 4 presents the utility obtained with different algorithm including QCR. For any algorithm, we plot in the y-axis $(U - U_{\text{opt}})/|U_{\text{opt}}|$ where U is the utility obtained on average during the simulation by this algorithm and U_{opt} is the value obtained with the optimal allocation. Hence it is always negative. Value $y = -1$ corrresponds to a utility 1% smaller than the optimal social welfare. Due to large variation of this quantity among parameters and algorithms, we used a logarithmic scale in the y-axis to present these results. For each of the algorithms, we consider two models of impatience function (power impatience and step function) with different parameters, varied along the x-axis.

We observe that extreme strategies (*i.e.,* uniform and dominated replication) fail to approach the optimal in general. In particular it is the case for small value of α, when users are sensitive to waiting delay and the decrease in social welfare can be high, and small value of τ where quick response is essential. Demand aware offline strategies (*i.e.,* proportional and square-root) perform similarly to QCR, although the latter does not require any global information. We even observe that QCR



(a) Expected Utility (b) Observed Util. (c) Cache: Mandate Routing (d) no Mandate Routing

Figure 3: Effect of Mandate Routing $\rho = 5, I = 50, N = 50, \omega = 1, \alpha = 0$

outperform proportional in many cases, sometimes very significantly. Among all cases, QCR does not incur a loss of utility beyond 5% (for step function) and 60% in the worst case of power function. One unexpected result is that the square root allocation is performing reasonably well in most case studied, which indicates that path replication may be a good replication algorithm for general impatience function, however this is an ideal performance observed when the allocation is fixed with *a priori* knowledge. In contrast, proportional allocation leads to much worse performance, in particular for power impatience function. Proportional allocation resembles a passive demand based replication where a fixed number of replicas (*e.g.,* one replica) are created whenever a request is fulfilled (as found in [11] and many other works). These results illustrate that such passive replication simply gives too much weight to popular items, and that compensating for this effect is both needed and doable.

6.3 Real Contact Traces

We now consider contact as measured between mobile nodes in two real scenarios.

Conference scenario.

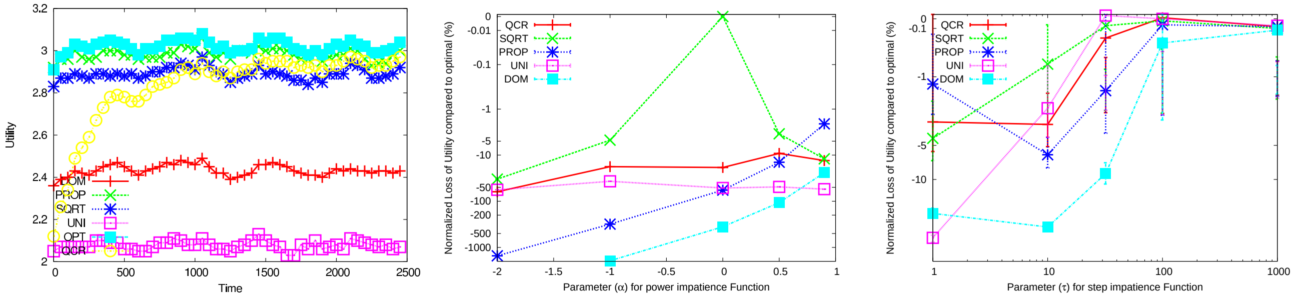
We use the Infocom06 data set which measures Bluetooth sightings between 73 participants of the Infocom conference (see [5] for more details) during 3 days. We included in this simulation contacts among 50 participants (numbered from 21 to 71 in the original data sets) that were selected as the one for which the measurement period was the longest, as a way to remove bias from poorly connected nodes.

Figure 5 (a) presents the utility as seen over time (time averaged over an hour) for different allocation and QCR (with mandate routing). We clearly observe the alternation of daytime and nighttime during the trace. For this case, the dominated and proportional allocation performs the best, QCR being very close to the latter, in spite of heterogeneity and complex time statistics. Square root and uniform performs poorly as the delay requirement is too stringent to bring significant improvement for non-popular items.

Figure 5 (b) and (c) presents the relative loss of utility for different algorithms (over the optimal allocation, as defined before) as a function of τ. We separate the impact of heterogeneity *per se* by presenting the actual traces and a synthetic trace where contact rates of all pairs are identical but contacts are assumed to follow memoryless time statistics. Heterogeneity *per se* does not seem to impact much the performance of QCR, indeed it looks like QCR is even performing better, perhaps because its implicit reaction to content availability adapts well to heterogeneous cases. The most notable difference with the homogeneous case is that the square root allocation is not a clear winner anymore. Proportional and dominated seems to improve their relative performance. The results from actual traces show that time statistics greatly impacts the performance of fixed allocation. First, since the optimal was computed under approximation of memory less contact, some algorithm actually provides higher utility that this reference benchmark. We also observe that the dominated allocation greatly improves due to bursty statistics. However, the performance of QCR remains quite comparable, it is generally within 15% of the optimal or approaches it even more closely.

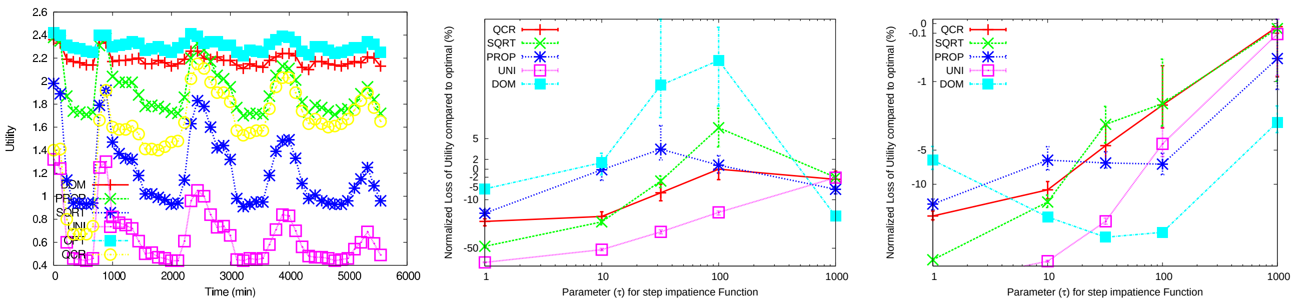
Vehicular networks.

We use contacts recorded between taxicabs gathered from the Cabspotting project. The data sets was extracted from a day of data and assumed that taxicab are in contacts whenever they are less than 200m apart (see [4] for more details). Results, shown as relative utility loss from optimal allocation, may be found in Figure 6 (a) (b) (c). Again, we observe that the optimal, which is based on a memoryless assumption, can be outperformed by some allocation (as in (b) for the step function case). Just like for the Infocom data set, we see that square root tends to produce degraded performance, while dominated allocation improves as heterogeneity and complex time statistics are included in the contact trace. The performance of QCR, the only schemes based on local information, appears less affected by this change.



(a) Evolution of Utility ($\tau = 1$) (b) Power Impatience (c) Step Function Impatience

Figure 4: Comparison of several algorithm (as loss of utility with regard to the optimal allocation) for two impatience model



(a) Evolution of Utility ($\tau = 1$) (b) Actual Trace (c) Synthesized Trace

Figure 5: Utility for Infocom06 Trace and step function model of impatience

7. CONCLUSION

Our results focus on a specific feature which makes P2P caching in opportunistic network unique: users's 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, as the optimal allocation may vary with different impatience model. We have seen that passive replication, ending in proportional allocation, can perform very badly, but that one can tune an adaptive replication scheme to approach the performance of the optimal in a large number of settings, 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. It offers a reference case from which one can study (1) the impact of heterogeneity and complex mobility property, (2) clustered and evolving demands in peers, as distributed mechanism like QCR naturally adapts to spatio-temporal variations. Another important aspect to address is how to estimate the impatience function implicitly from user feedback, instead of assuming that it is known.

8. ACKNOWLEDGMENT

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

9. REFERENCES

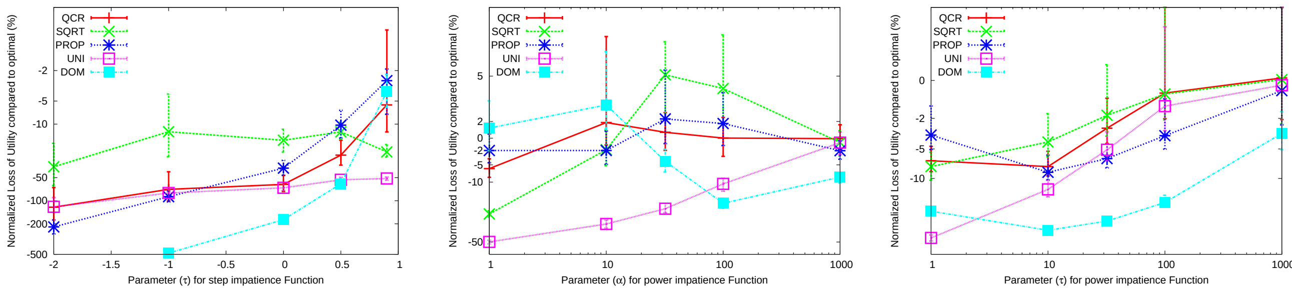
[1] A. Balasubramanian, B. Levine, and A. Venkataramani. DTN Routing as a Resource Allocation Problem. In *Proc. ACM SIGCOMM*, August 2007.

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

[3] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[4] A. Chaintreau, J.-Y. L. Boudec, and N. Ristanovic. The age of gossip: spatial mean field regime. In *Proc. of ACM SIGMETRICS*, 2009.

[5] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, J. Scott, and R. Gass. Impact of human mobility on opportunistic forwarding algorithms. *IEEE Trans. Mob. Comp.*, 6(6):606–620, 2007.



(a) Power function, Actual Trace (b) Step function, Actual Trace (c) Step function, Synthetic Trace

Figure 6: Cabspotting, Utility (in normalized comparison w.r.t. optimal allocation) for different impatience functions

[6] E. Cohen and S. Shenker. Replication strategies in unstructured peer-to-peer networks. *SIGCOMM Comput. Commun. Rev.*, 32(4):177–190, 2002.

[7] P. Costa, C. Mascolo, M. Musolesi, and G. Picco. Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks. *IEEE Jsac*, 26(5):748–760, June 2008.

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

[9] M. Grossglauser and D. Tse. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Trans. on Net.*, 10(4):477–486, 2002.

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

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

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

[13] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of the approximations for maximizing submodular set functions. *Mathematical Programming*, 14, 1978.

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

[15] A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav. Low-cost communication for rural internet kiosks using mechanical backhaul. In *Proc. ACM MobiCom*, 2006.

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

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

[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.

APPENDIX

A. PROOFS

A.1 General Expression for U

Proof of Lemma 1.

We use the following two facts:

Let X be a geometric random variables in $\{\ 1, 2, \cdots\ \}$ with probability of success r, then for any functions f

$$\mathbb{E}\left[f(X)\right] = f(1) - \sum_{k\geq 1}(1-r)^k(f(k)-f(k+1))\,. \quad (8)$$

It follows a simple Abel transformation of the series defining the expectation of this variable.

Similarly, let X be an exponential random variable with parameter λ, then we have, for any derivable function f defined on $[0, \infty[$ which admits a limit in 0^+:

$$\mathbb{E}\left[f(X)\right] = f(0^+) + \int_0^\infty \exp\left(-\lambda t\right) f'(t)dt\,. \quad (9)$$

It follows from a simple integration by part.

Whenever a demand for item i is created in client node n, two possible cases occur: Either the node is also a server and it contains a version of this item (*i.e.,* $n \in \mathcal{S}$ and $x_{i,n} = 1$), or the item has to be request from a server node. In the former case, the demand is fulfilled immediately in the next time slot, and the demand creates a gain $h(\delta)$ for this user. Otherwise, it may be fulfilled whenever a node meets a server node that stores this item. For any time slot, this client node's request may be fulfilled by server m with probability $(x_{i,m} \cdot \mu_{m,n} \cdot \delta)$. Hence the number of trials before this request is fulfilled follows a geometric random variable in $\{\ 1, 2, \cdots\ \}$ with a probability of success

$$r_{i,n} = 1 - \prod_{m\in\mathcal{S}}\left(1 - x_{i,m}\mu_{m,n}\delta\right)\,.$$

We may then write that the expected gain for an item i requested at node n is

$$\mathbb{E}\left[U_{i,n}\right] = x_{i,n}h(\delta) + (1-x_{i,n})\mathbb{E}\left[h(\delta\cdot X)\right]\,,$$

where X is a geometric random variable with success probability $r_{i,n}$, and $x_{i,n} = 0$ by convention whenever $n \notin \mathcal{S}$. Combining the above inequality with Eq.(8) we have:

$$\mathbb{E}\left[U_{i,n}\right] = h(\delta) - (1-x_{i,n})\sum_{k\geq 1}(1-r)^k\left(h(\delta k) - h(\delta\,(k+1))\right).$$

After replacing r and using the differential impatience function c, we obtain the expression of the lemma.

In a continuous time contact model, when the node i does not possess a copy of item i, the time elapsed before this request can be fulfilled is an exponential random variable with parameter:

$$\sum_{m\in\mathcal{S}} x_{i,m}\mu_{m,n}\,.$$

The result then follows from the exact same argument and Eq.(9).

Application to Homogeneous contact case.

In the homogeneous contact case, the expression for U simplifies: Let us start with the **dedicated node case** first. In the discrete time case, one can see

$$\prod_{m\in\mathcal{S}}\left(1 - x_{i,m}\mu_{m,n}\delta\right) = \left(1-\mu\delta\right)^{x_i}\,.$$

Moreover, in the dedicated node case, for any $n \in \mathcal{C}$, $n \notin \mathcal{S}$ and hence $x_{i,n}$ is null by convention. The expression of $U_{i,n}$ from Lemma 1 hence can be rewritten:

$$U_{i,n}(\mathbf{x}) = h_i(\delta) - \sum_{k\geq 1}\left(1-\mu\delta\right)^{k\cdot x_i} c_i(k\cdot\delta)\,.$$

This expression does not depend on n anymore, and we have

$$\begin{aligned} U(\mathbf{x}) &= \sum_i d_i \sum_{n\in\mathcal{C}} \pi_{i,n} U_{i,n} \\ &= \sum_i d_i \left(h_i(\delta) - \sum_{k\geq 1}\left(1-\mu\delta\right)^{k\cdot x_i} c_i(k\cdot\delta)\right)\,, \end{aligned}$$

which implies Eq.(2).

In the continuous time case, we first observe that

$$\sum_{m\in\mathcal{S}} x_{i,m}\mu_{m,n} = x_i\cdot\mu\,.$$

As a consequence, following the same argument as before,

$$U_{i,n}(\mathbf{x}) = h_i(0^+) - \int_0^\infty \exp\left(-t x_i\cdot\mu\right) c_i(t) dt\,.$$

This expression does not depend on n any more, and implies that $U(\mathbf{x})$ can be written as Eq.(3).

We now consider the **pure P2P case**, where all nodes are server, and we assume that profile are uniform among users (*i.e.,* $\pi_{i,n} = 1/N$).

$$U_{i,n}(\mathbf{x}) = h_i(\delta) - (1-x_{i,n})\sum_{k\geq 1}\left(1-\mu\delta\right)^{k\cdot x_i} c_i(k\cdot\delta)\,.$$

For n such that $x_{i,n} = 1$ we have $U_{i,n} = h_i(\delta)$, this case occurs exactly x_i times. For all other value of n we have the same expression as in the dedicated node case above, this case occurs exactly $N - x_i$ times. Hence we deduce:

$$\begin{aligned} \sum_{n\in\mathcal{C}} \pi_{i,n} U_{i,n} &= \sum_{n\in\mathcal{C}} \frac{1}{N} U_{i,n} \\ &= h_i(\delta) - \frac{N-x_i}{N}\sum_{k\geq 1}\left(1-\mu\delta\right)^{k\cdot x_i} c_i(k\cdot\delta)\,. \end{aligned}$$

This proves Eq.(4). A similar argument for the continuous case yields Eq.(5).

A.2 Submodularity and Concavity property

Proof of Theorem 1.

It is sufficient to prove that, for a given item i and a client node n, $U_{i,n}$ is a submodular function of the set $\{ m \in \mathcal{S} \mid x_{i,m} = 1 \}$, since the sum of supermodular functions is supermodular.

Let us fix an item i and a client node n. For any subset A of $\mathcal{S}$, we introduce the following set function:

$$\sigma : A \mapsto \prod_{m \in A} (1 - \mu_{m,n} \cdot \delta) ,$$

We may redefine $U_{i,n}$ as a set function such that $U_{i,n}(A)$ is given by:

$$h_i(\delta) - (1 - \mathbb{I}_{\{n \in A\}}) \sum_{k \geq 1} \sigma(A)^k c_i(k \cdot \delta) .$$

We first observe that, for any $k \geq 1$, the function $f : A \mapsto \sigma(A)^k$ is supermodular. Indeed, we have for any subset $A \subseteq \mathcal{S}$ and $m \in \mathcal{S}$:

$$f(A \cup \{m\}) - f(A) = \mathbb{I}_{\{m \notin A\}} \cdot \sigma(A)^k \left((1 - \mu_{m,n})^k - 1\right) .$$

For $A \subseteq B$, we have

$$\mathbb{I}_{\{m \notin A\}} \left(\sigma(A)\right)^k \geq \mathbb{I}_{\{m \notin B\}} \left(\sigma(B)\right)^k \geq 0 ,$$

since both terms are positive non-increasing set function (w.r.t. set inclusion). As $\left((1 - \mu_{m,n})^k - 1\right)$ is always non-positive, we deduce for $A \subseteq B$

$$\begin{aligned} f(A \cup \{m\}) - f(A) &= \left((1 - \mu_{m,n})^k - 1\right) \mathbb{I}_{\{m \notin A\}} \cdot \sigma(A) \\ &\leq \left((1 - \mu_{m,n})^k - 1\right) \mathbb{I}_{\{m \notin B\}} \cdot \sigma(B) \\ &\leq f(B \cup \{m\}) - f(B) . \end{aligned}$$

We deduce for any k, f is a supermodular function, which is also positive. A weighted sum, with positive weights, of supermodular function is a supermodular function. Hence, since $c_i(k \cdot \delta)$ is positive as the impatience function h_i is non-increasing, we deduce

$$A \mapsto \sum_{k \geq 1} \sigma(A)^k c_i(k \cdot \delta) ,$$

is a supermodular function. We also deduce that it is positive. We now observe that $g : A \mapsto (1 - \mathbb{I}_{\{n \in A\}})$ is a positive supermodular function. Indeed, we may write

$$g(A \cup \{m\}) - g(A) = -\mathbb{I}_{\{m \notin A\}} \mathbb{I}_{\{m = n\}} .$$

which is, again, a non-decreasing function of A (w.r.t. the set inclusion).

Finally as a product of positive supermodular function is positive and supermodular, we deduce that $U_{i,n}$ may be written as a constant minus a positive supermodular function. It is hence by definition submodular.

The same argument holds for a continuous time model: We first introduce the following set function defined on all subset A of $\mathcal{S}$:

$$\tilde{\sigma} : A \mapsto \sum_{m \in A} \mu_{m,n} ,$$

We may redefine $U_{i,n}$ for the continuous time model as a set function such that $U_{i,n}(A)$ is given by:

$$h_i(0^+) - (1 - \mathbb{I}_{\{n \in A\}}) \int_0^\infty \exp\left(-t \cdot \tilde{\sigma}(A)\right) c_i(t) dt .$$

We first observe that, for any $t \geq 0$, the function $f : A \mapsto \exp(-t \cdot \tilde{\sigma}(A))$ is supermodular. Indeed, we have for any subset $A \subseteq \mathcal{S}$ and $m \in \mathcal{S}$:

$$f(A \cup \{m\}) - f(A) = \mathbb{I}_{\{m \notin A\}} \cdot e^{-t\tilde{\sigma}(A)} \left(e^{-t\mu_{m,n}} - 1\right) .$$

For $A \subseteq B$, we have $\mathbb{I}_{\{m \notin A\}} \cdot e^{-t\tilde{\sigma}(A)} \geq \mathbb{I}_{\{m \notin B\}} \cdot e^{-t\tilde{\sigma}(B)}$, since both are positive non-increasing function of the set. As, for any m $(e^{-t\mu_{m,n}} - 1)$ is non-positive, we deduce for $A \subseteq B$

$$\begin{aligned} f(A \cup \{m\}) - f(A) &= \mathbb{I}_{\{m \notin A\}} \cdot e^{-t\tilde{\sigma}(A)} \left(e^{-t\mu_{m,n}} - 1\right) \\ &\leq \mathbb{I}_{\{m \notin B\}} \cdot e^{-t\tilde{\sigma}(B)} \left(e^{-t\mu_{m,n}} - 1\right) \\ &\leq f(B \cup \{m\}) - f(B) . \end{aligned}$$

We deduce that, whatever be the value of t, f is a supermodular function, which is also positive. A weighted sum, with positive weights, of supermodular function is a supermodular function, which also generalizes to integral of function multiplied by a positive term. Hence, since $c_i(t)$ is by definition positive for any value of t we deduce that

$$A \mapsto \int_0^\infty \exp\left(-t \cdot \tilde{\sigma}(A)\right) c_i(t) dt ,$$

is a supermodular function. We also deduce that it is positive. Recall that $A \mapsto (1 - \mathbb{I}_{\{n \in A\}})$ is a positive supermodular function. Hence, again, as a product of positive supermodular function is supermodular and positive, we deduce that $U_{i,n}$ may be written as a constant minus a positive super-linear function. It is hence by definition submodular.

Proof of Theorem 2.

With homogeneous contacts between nodes, the social welfare only depends on the total number of copies x_i of each item i. Let us start with the dedicated node case under the discrete time contact model. In this case, we have

$$U(\mathbf{x}) = \sum_{i \in I} d_i \left(h(\delta) - \sum_{k \geq 1} (1 - \mu\delta)^{x_i k} c_i(k \cdot \delta) \right) .$$

We first observe that, if we allow x_i to take real number value, U is a concave function of the $\{x_i | i \in I\}$. It comes from the fact that, for any $k \geq 1$, the function

$$x \mapsto (1 - \mu\delta)^{xk} ,$$

is convex. U is then a weighted sum of concave functions with positive coefficient (since $c(k \cdot \delta)$ is non-negative for all k). The same argument applies to show that the relaxed optimization for continuous time, as well as for the pure P2P case, satisfy the same property, from Eq.(3)-(5).

This proves the second half of theorem and directly implies that there exists a unique maximum of the relaxed optimization which can be found through gradient descent algorithm [3].

The only remaining point to prove the theorem is to show that, for the original problem where x_i only takes integer values, the maximum can be obtained by a greedy procedure

For any $\mathbf{x} = \{x_i | i \in I\}$, we denote by $\frac{\Delta U}{\Delta x_i}(\mathbf{x})$ the *marginal improvement* obtained when another copy of item i is created. It is defined by

$$\frac{\Delta U}{\Delta x_i}(\mathbf{x}) = U(x_1, \ldots, x_{i-1}, x_i+1, x_{i+1}, \ldots, x_M) - U(\mathbf{x}) .$$

One key observation is that, for a given $i \in i$, the marginal improvement $\frac{\Delta U}{\Delta x_i}(\mathbf{x})$ is independent of x_j for all $j \neq i$. Note that this is intuitive, as for a fixed number of copies of a given item, the delay or waiting time to find this item is not impacted by how much copies of *other* items are available. In other words, we may write $\frac{\Delta U}{\Delta x_i}(\mathbf{x}) = f_i(x_i)$, where

$$f_i(x) = d_i \sum_{k \geq 1} (1 - \mu\delta)^{xk} \left(1 - (1 - \mu\delta)^k\right) c_i(k \cdot \delta) .$$

In summary, the value of the social welfare can be decomposed, for each item i, as a sum of x_i terms that are values of a function f_i:

$$U(\mathbf{x}) = \sum_{i \in I} \left(f_i(1) + f_i(2) + \cdots + f_i(x_i)\right) . \qquad (10)$$

Another important observation is that, owing to the concavity of the function U (established above), the function f_i above are all non-increasing.

We will now prove that a greedy procedure can find the allocation with maximum social welfare. Let us first define two optimization problems.

$$\mathbf{OPT}(C): \quad \max \sum_{i \in I} \sum_{l \in \{1,2,\cdots,x_i\}} f_i(l) \ \text{ such that} \quad 1 \leq x_i \leq |\mathcal{S}| , \sum_{i \in I} x_i \leq C$$

$$\mathbf{SETOPT}(C): \quad \max \sum_{i \in I} \sum_{l \in A_i} f_i(l) \ \text{ such that} \quad A_i \subseteq \mathbb{N}^* , 1 \leq |A_i| \leq |\mathcal{S}| , \sum_{i \in I} |A_i| \leq C$$

The problem $\mathbf{OPT}(C)$ with $(C = \rho \cdot |\mathcal{S}|)$ is exactly the problem we wish to solve. The problem $\mathbf{SETOPT}(C)$ is only defined for the need of the proof: it is a slightly more general problem in the sense that it does not require to look at sum of values of f_i on contiguous integers, but can consider values of functions f_i on any collection of subsets.

We start by the two following simple results:

Lemma 2. *When the functions f_i are non-increasing for all i, the two optimization problems are equivalent:*

$$\text{We have } \ \forall i \in I \, , \, A_i = \{1, 2, \ldots, x_i\} .$$

where $\{ \ x_i \ | \ i \in I \ \}$ (resp. $\{ \ A_i \ | \ i \in I \ \}$) denotes the solution of $\boldsymbol{OPT}(C)$ (resp. $\boldsymbol{SETOPT}(C)$).

One can easily show that the A_i should be made of contiguous integers starting from 1 (If that is not the case, is is easy to construct an even better choice of A_i to maximize the sum). Owing to this fact, the two problems are equivalent and the result above holds.

Lemma 3. *Let $\{ \ A_i \ | \ i \in I \ \}$ and $\{ \ B_i \ | \ i \in I \ \}$ be the solutions of $\boldsymbol{SETOPT}(C)$ and $\boldsymbol{SETOPT}(C+1)$.*

- *for any $i \in I$, $A_i \subseteq B_i$.*
- *The only j such that $B_j \neq A_j$ satisfies*

$$j = \operatorname{argmax} \left\{ \max_{l \notin A_i} f_i(l) \ \middle| \ i \in I, |A_i| < |\mathcal{S}| \right\} .$$

Since A cannot contain B due to size constraint, there exists j and a such that $a \in B_j$ and $a \notin A_j$. Let $B'_j = B_j \setminus \{a\}$ and $B'_i = B_i$ for all $i \neq j$. Then the collection of all subsets B'_j for all $j \in I$ satisfies the conditions of problem $\mathbf{SETOPT}(C)$. By optimality of B we should have $B'_i = A_i$ for all $i \in I$ (otherwise one could always construct an even better choice than B. This concludes the proof, as again, if the second property does not hold, one can construct an even better solution starting from the subsets A_i and adding one element.

As a consequence of the two lemmas, when all functions f_i for all i are non-increasing, we deduce that if $\{ \ x_i \ | \ i \in I \ \}$ and $\{ \ y_i \ | \ i \in I \ \}$ are the solutions of $\mathbf{OPT}(C)$ and $\mathbf{OPT}(C+1)$ then

$$\begin{cases} y_j = x_j + 1 & \text{if } \ j = \operatorname{argmax}_i \{f_i(x_i+1) | x_i < |\mathcal{S}|\} \\ y_j = x_j & \text{otherwise} \ . \end{cases}$$

This can be used as a recursive rules to deduce the optimal allocation for any cache size C. This is what Algorithm 1, defined below, implements in at most $O(|I| + \rho|\mathcal{S}| \cdot \ln(|I|))$ steps (since the search for the maximum improvement can be run in at most $O(\ln(|I|)$ with a priority queue).

The proof was conducted here in the discrete time for the dedicated node case, the same exact argument applies to pure P2P case, where f_i is defined as

$$f_i(x) = d_i(1 - \frac{x}{N}) \sum_{k \geq 1} (1 - \mu\delta)^{xk} \left(1 - (1 - \mu\delta)^k\right) c_i(k \cdot \delta) .$$

Algorithm 1 Max Welfare (Homogeneous contact).

$x_i \leftarrow 1$;$\texttt{sum} \leftarrow M$;
$A = \{\ 1, 2, \cdots, M\ \}$;
for all $i \in A$ **do**
  $\texttt{imp}_i \leftarrow \frac{\Delta U}{\Delta x_i}(\mathbf{x})$;
end for
while $\texttt{sum} \le \rho|\mathcal{S}|$ **do**
  pick $j = \arg\max_i \{\ \texttt{imp}_i\ \mid i \in A\ \}$;
  $x_j \leftarrow x_j + 1$; $\texttt{sum} \leftarrow \texttt{sum} + 1$;
  $\texttt{imp}_j \leftarrow \frac{\Delta U}{\Delta x_i}(\mathbf{x})$;
  if $x_j = |\mathcal{S}|$ **then**
    $A \leftarrow A \setminus \{j\}$;
  end if
end while

Similarly, we can prove the same result for a continuous time contact model, where f_i is defined in the dedicated node case as

$$f_i(x) = d_i \int_0^\infty e^{-\mu t \cdot x} \left(1 - e^{\mu t}\right) c_i(t) dt\,,$$

and in the pure p2p case as

$$f_i(x) = d_i (1 - \frac{x}{N}) \int_0^\infty e^{-\mu t \cdot x} \left(1 - e^{\mu t}\right) c_i(t) dt\,.$$

All these functions are positive and non-increasing, which allows to use the exact same proof.

A.3 Proof of Theorem 3

In the continuous time contact model an dedicated node we have:

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

Note that it implies $\dfrac{\partial U}{\partial x_i}(\mathbf{x}) = d_i \varphi(x_i)\,.$ where φ is defined as in Theorem 3.

Let us assume that $\tilde{x}_i < |\mathcal{S}|$, $\tilde{x}_j < |\mathcal{S}|$. If we have $d_i \cdot \varphi(\tilde{x}_i) < d_j \cdot \varphi(\tilde{x}_j)$, then there exists $\varepsilon > 0$ such that $U(x') > U(\tilde{x})$ where we define x' as

$$\begin{cases} x'_j = \tilde{x}_j + \varepsilon \\ x'_i = \tilde{x}_i - \varepsilon \\ x'_k = \tilde{x}_k & \text{for } \ k \neq i, k \neq j\,. \end{cases}$$

This contradicts the optimality of $\tilde{x}$, and proves that the equilibrium condition holds.

A.4 Singularity of h at $t = 0$

As shown in Lemma 1, $U_{i,n}(\mathbf{x})$ and $U(\mathbf{x})$ may be expressed in a simple form when $h(0^+) < \infty$. In the dedicated node case, where servers and clients are in disjoint sets, the same expression generalizes to other types of impatience function where $h(0^+) = \infty$. As an example, we treat here the case of the negative logarithm (*i.e.*, $h_\alpha^{(\mathrm{p})}$ with $\alpha = 1$) and the inverse powers (*i.e.*, $h_\alpha^{(\mathrm{p})}$ with $\alpha > 1$).

Since we are in the dedicated node case, we have

$$U_{i,n}(\mathbf{x}) = \mathbb{E}\left[h(X)\right]\,,$$

where X is an exponential random variable with parameter $\sum_{m \in \mathcal{S}} x_{i,m}\mu_{m,n}$, that we can denote by $\mu_{i,n}$ for short.

Expression of $U_{i,n}$ for the Negative logarithm.
When $\alpha = 1$, the impatience function is defined as $h_1^{(\mathrm{p})} : t \mapsto -\ln(t)$. Hence we have :

$$U_{i,n}(\mathbf{x}) = \int_0^\infty \mu_{i,n} \exp\left(-t\mu_{i,n}\right) \left(-\ln(t)\right) dt\,.$$

A simple change of variable leads to

$$\begin{aligned} U_{i,n}(\mathbf{x}) &= \int_0^\infty \left(\ln(\mu_{i,n}) - \ln(u)\right) e^{-u} du \\ &= \ln\left(\sum_{m \in \mathcal{S}} x_{i,m}\mu_{m,n}\right) - \int_0^\infty \ln(u) e^{-u} du\,. \end{aligned}$$

This allows to compute U thanks to

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

This expression simplifies for the case of homogeneous contact as follows

$$\begin{aligned} U(\mathbf{x}) &= \sum_{i \in I} d_i \ln\left(x_i\right) + \texttt{cst}, \quad \text{where} \\ &\texttt{cst} = \sum_{i \in I} d_i \left(\ln\left(\mu\right) - \int_0^\infty \ln(u) e^{-u} du \right)\,. \end{aligned}$$

Consequence on Theorem 1 and 2.
We first note that it proves that $U_{i,n}$ is, as a set function, submodular (which extends the results of Theorem 1). Indeed, with similar notation than in Section A.2, we can write:

$$U_{i,n}(A) = \ln\left(\tilde{\sigma}(A)\right) - \int_0^\infty \ln(u) e^{-u} du\,.$$

which proves this result. We also deduce that the argument for the proof of Theorem 2 holds, since we can write $\frac{\Delta U}{\Delta x_i}(\mathbf{x}) = f_i(x_i)$ with

$$f_i(x) = d_i \ln\left(1 + \frac{1}{x_i}\right)\,.$$

which is a positive non-increasing function.

Expression of $U_{i,n}$ for the inverse powers.
When $1 < \alpha < 2$, we have that $h_\alpha^{(\mathrm{p})} : t \mapsto \frac{t^{1-\alpha}}{\alpha - 1}$, and we may write in the dedicated node case:

$$U_{i,n}(\mathbf{x}) = \int_0^\infty \mu_{i,n} \exp\left(-t\mu_{i,n}\right) \frac{t^{1-\alpha}}{\alpha - 1} dt\,.$$

Again, a simple change of variable yields

$$\begin{aligned} U_{i,n}(\mathbf{x}) &= (\mu_{i,n})^{\alpha-1} \int_0^{\infty} \exp(-u) \frac{u^{1-\alpha}}{\alpha-1} du \\ &= \frac{\Gamma(2-\alpha)}{\alpha-1} \left(\sum_{m\in\mathcal{S}} x_{i,m}\mu_{m,n} \right)^{\alpha-1} . \end{aligned}$$

This equation can be used to derive $U(\mathbf{x})$ in the general case. For homogeneous contact case, it simplifies to

$$U(\mathbf{x}) = \frac{\Gamma(2-\alpha)}{\alpha-1} \mu^{\alpha-1} \sum_{i\in I} d_i (x_i)^{\alpha-1} .$$

Consequence on Theorem 1 and 2.

We have, with similar notation than in Section A.2,

$$U_{i,n}(A) = \frac{\Gamma(2-\alpha)}{\alpha-1} (\tilde{\sigma}(A))^{\alpha-1} .$$

For any $1 < \alpha < 2$, the function $x \mapsto x^{\alpha-1}$ is concave. The formula above proves that $U_{i,n}$ is again in this case, a submodular function, extending again the results of Theorem 1 to a new class of function.

Again the argument for the proof of Theorem 2 holds, as we have $\frac{\Delta U}{\Delta x_i}(\mathbf{x}) = f_i(x_i)$ with

$$f_i(x) = d_i \frac{\Gamma(2-\alpha)}{\alpha-1} \left((x+1)^{\alpha-1} - x^{\alpha-1} \right) .$$

which is a positive non-increasing function.

As a conclusion, although the exact computation may differ, the exact same arguments why all the results of this paper applies to a general impatience function with a limit in 0^+ holds for a general family of impatience function even when it diverges in 0. It should be possible to prove this result more generally, as well as to obtain similar expressions for the discrete time case, but this is beyond the scope of this current work.

A.5 Table condition in the pure P2P case

In the continuous time contact model an pure p2p case we have:

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

Note that it implies $\dfrac{\partial U}{\partial x_i}(\mathbf{x}) = d_i \varphi'(x_i)$ where

$$\varphi'(x) = (1 - \frac{x}{N})\varphi(x) + \frac{1}{N} \int_0^{\infty} e^{-t\mu x} c(t) dt .$$

In other words, just like Theorem 3 applies to the dedicated node case, a similar equilibrium condition exists for the optimality in the pure p2p case. We just need to define another function φ' which contains a correcting term, that is small when N is large.

The main consequence is that, just like Theorem 4, there exists a demand reaction function which is exactly optimal in the pure p2p case. This reaction function is shown in Table 2. As it can be seen on the table, most of the time the correction term is small, which is why we did not use this version in the simulations.

B. FULL NUMERICAL RESULTS

In this section we present, for completeness, the result we obtain across all parameters we used for validation.

B.1 Impact of cache size and popularity distribution

Figure 7 presents the difference in Utility as taken from optimal, for the step impatience function with different value of τ (on the x-axis), for the same strategies as in Section 6. The difference with previous results is that results are also shown for a smaller cache size $\rho = 2$ and a larger cache size $\rho = 10$. Note that we also reproduce the result for $\rho = 5$ (the default value) to allow for an easy comparison. As it can be seen, although there are small numerical differences, it plays almost no role in the qualitative behaviors, across all contact traces.

Figure 8, presents the same result as above, except that we consider an highly skewed distribution (*i.e.,* $\omega = 2$) and a moderately skewed distribution (*i.e.,* $\omega = 1/2$), compared with the default case where ω is equal to 1. We recall that ω corresponds to the skewness of the popularity of file as: $d_i \propto i^{-\omega}$. As expected, a moderately skewed distribution tends to improve the performance of uniform and decreases that of the dominated allocation. A highly skewed distribution seems to even out most of the strategies across all data sets, except for uniform which performs badly.

B.2 Full results for the homogeneous case

We consider in this section a network with homogeneous contact among the nodes. Figure 4 already presented for different family of impatience functions the respective performance of the different algorithms. Figures 9 and 10 present in more details the content of the cache, and the utility obtained as a function of time for one run of each of this impatience functions.

B.3 Full results for real contact traces

We now present the same full results (cache and utility perceived as a function of time for the contact traces (as well as the synthetic contact traces, which represents the heterogeneity of contact only).

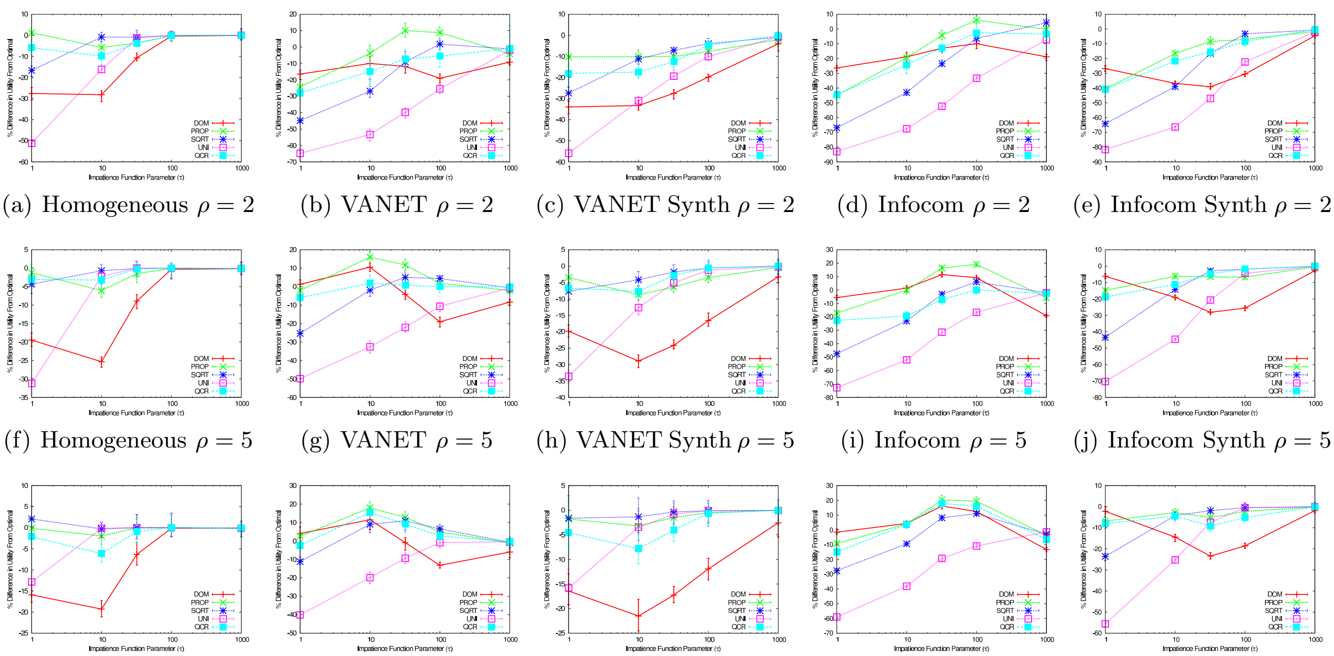
Figures 11 and 12 present the result for the Infocom datasets and the step impatience function (with $\rho = 5$.

Figure 13 and 14 present the same results for the Vanet data sets with power impatience functions.

Finally 15-20 present the results for the Vanet data sets, with step impatience functions and several values of ρ.

Model	Step function	Exponential decay	Neg. Power ($\alpha < 1$)
Impatience $h(t)$	$\mathbb{I}_{\{t \leq \tau\}}$	$\exp(-\nu t)$	$\dfrac{t^{1-\alpha}}{\alpha - 1}$
Diff. impat. c	Dirac at $t = \tau$	density $t \mapsto \nu \exp(-\nu t)$	density $t \mapsto t^{-\alpha}$
Gain $U_{i,n}(\mathbf{x})$	$1 - (1 - \frac{x_i}{N})e^{-\mu\tau x_i}$	$1 - (1 - \frac{x_i}{N})\left(1 + \frac{\mu}{\nu} x_i\right)^{-1}$	$\mu^{\alpha-1}\Gamma(1-\alpha)\left(-(x_i)^{\alpha-1} + \frac{1}{N}(x_i)^{\alpha}\right)$
Cond. φ'	$d_i \cdot \left(\mu\tau + \frac{1-\mu\tau x_i}{N}\right) e^{-\mu\tau x_i}$	$d_i \cdot \left(\frac{\mu}{\nu} - \frac{1}{N}\right)\left(1 + \frac{\mu}{\nu} x_i\right)^{-2}$	$d_i \cdot \mu^{\alpha-1}\Gamma(1-\alpha)\left((1-\alpha)x_i^{\alpha-2} + \frac{1}{N}\alpha x_i^{\alpha-1}\right)$
Reaction ψ	$\dfrac{\mu\tau + \frac{1-\mu\tau/y}{N}}{y} e^{-\frac{\mu\tau}{y}}$	$\left(1 - \frac{\nu}{\mu}\frac{1}{N}\right)\left(2 + \frac{\nu}{\mu} y + \frac{\mu}{\nu}\frac{1}{y}\right)^{-1}$	$\mu^{\alpha-1}\Gamma(1-\alpha)\left((1-\alpha)y^{1-\alpha} + \frac{1}{N}\alpha y^{-\alpha}\right)$

Table 2: Optimal demand reaction and cache allocation for several usual impatience functions (dedicated cache case).

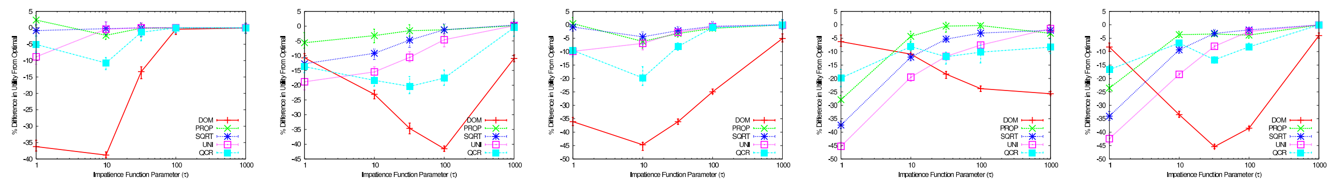


(a) Homogeneous $\rho = 2$ (b) VANET $\rho = 2$ (c) VANET Synth $\rho = 2$ (d) Infocom $\rho = 2$ (e) Infocom Synth $\rho = 2$

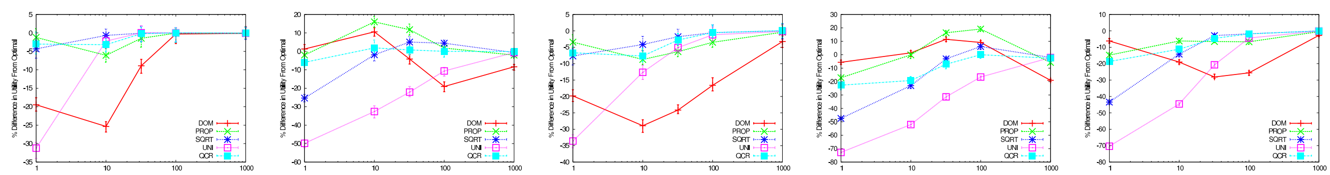
(f) Homogeneous $\rho = 5$ (g) VANET $\rho = 5$ (h) VANET Synth $\rho = 5$ (i) Infocom $\rho = 5$ (j) Infocom Synth $\rho = 5$

(k) Homogeneous $\rho = 10$ (l) VANET $\rho = 10$ (m) VANET Synth $\rho = 10$ (n) Infocom $\rho = 10$ (o) Infocom Synth $\rho = 10$

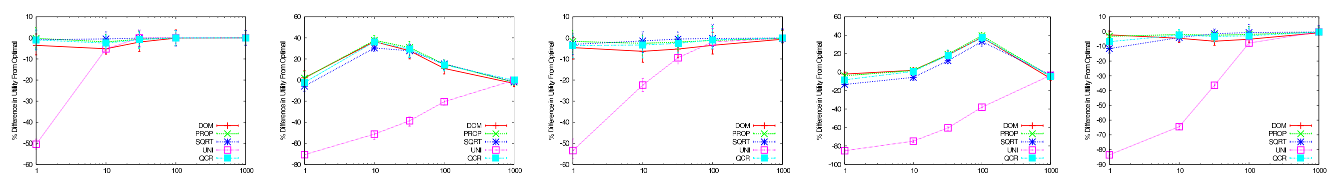
Figure 7: Impact of the cache size ρ (I=10, N=50, $\omega = 1$).



(a) Homogeneous $\omega = 1/2$ (b) VANET $\omega = 1/2$ (c) VANET Synth $\omega = 1/2$ (d) Infocom $\omega = 1/2$ (e) Infocom Synth $\omega = 1/2$

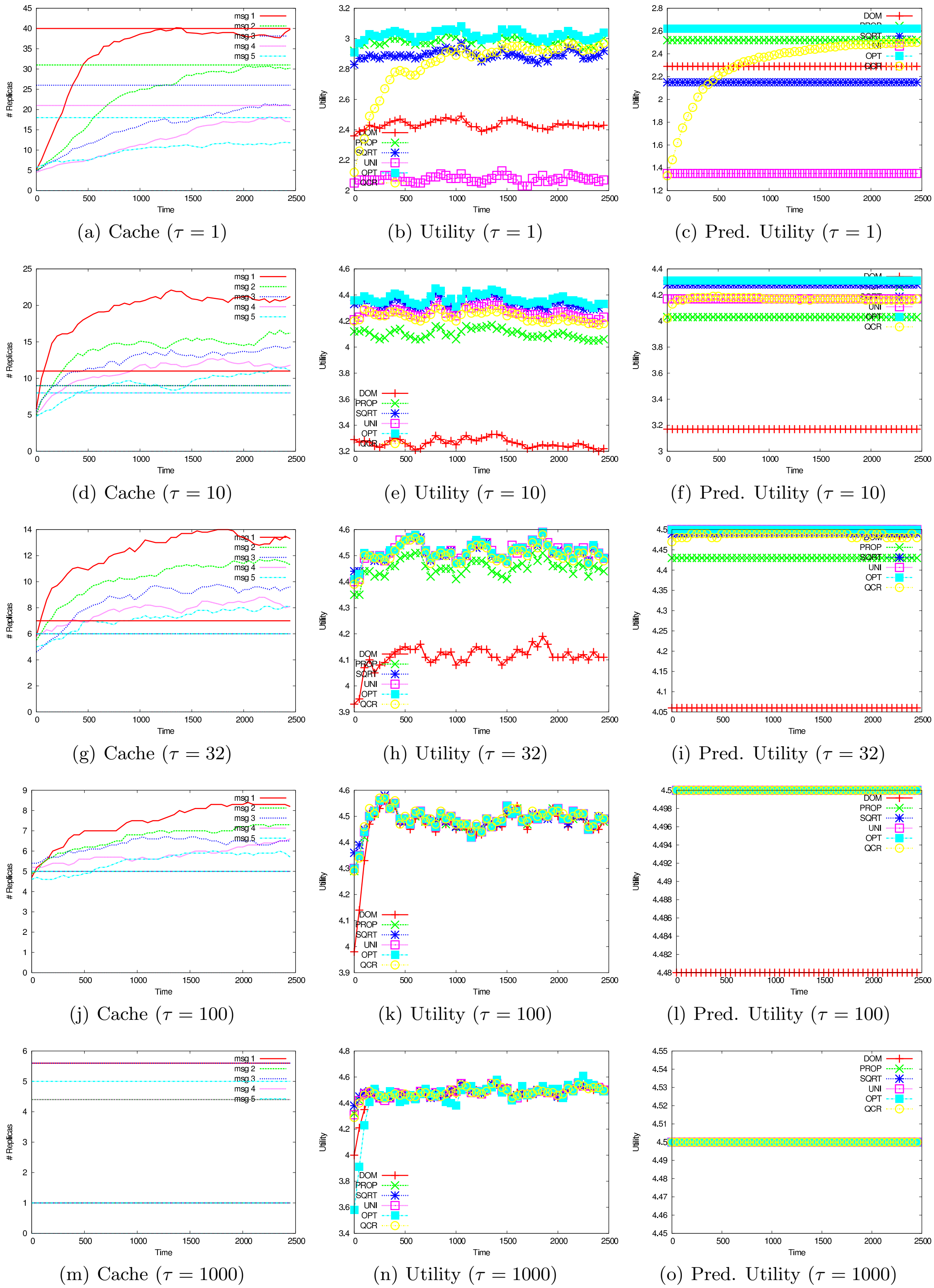


(f) Homogeneous $\omega = 1$ (g) VANET $\omega = 1$ (h) VANET Synth $\omega = 1$ (i) Infocom $\omega = 1$ (j) Infocom Synth $\omega = 1$



(k) Homogeneous $\omega = 2$ (l) VANET $\omega = 2$ (m) VANET Synth $\omega = 2$ (n) Infocom $\omega = 2$ (o) Infocom Synth $\omega = 2$

Figure 8: Impact of the popularity distribution of items ω (I=10, N=50, $\rho = 5$).



(a) Cache ($\tau = 1$) (b) Utility ($\tau = 1$) (c) Pred. Utility ($\tau = 1$)

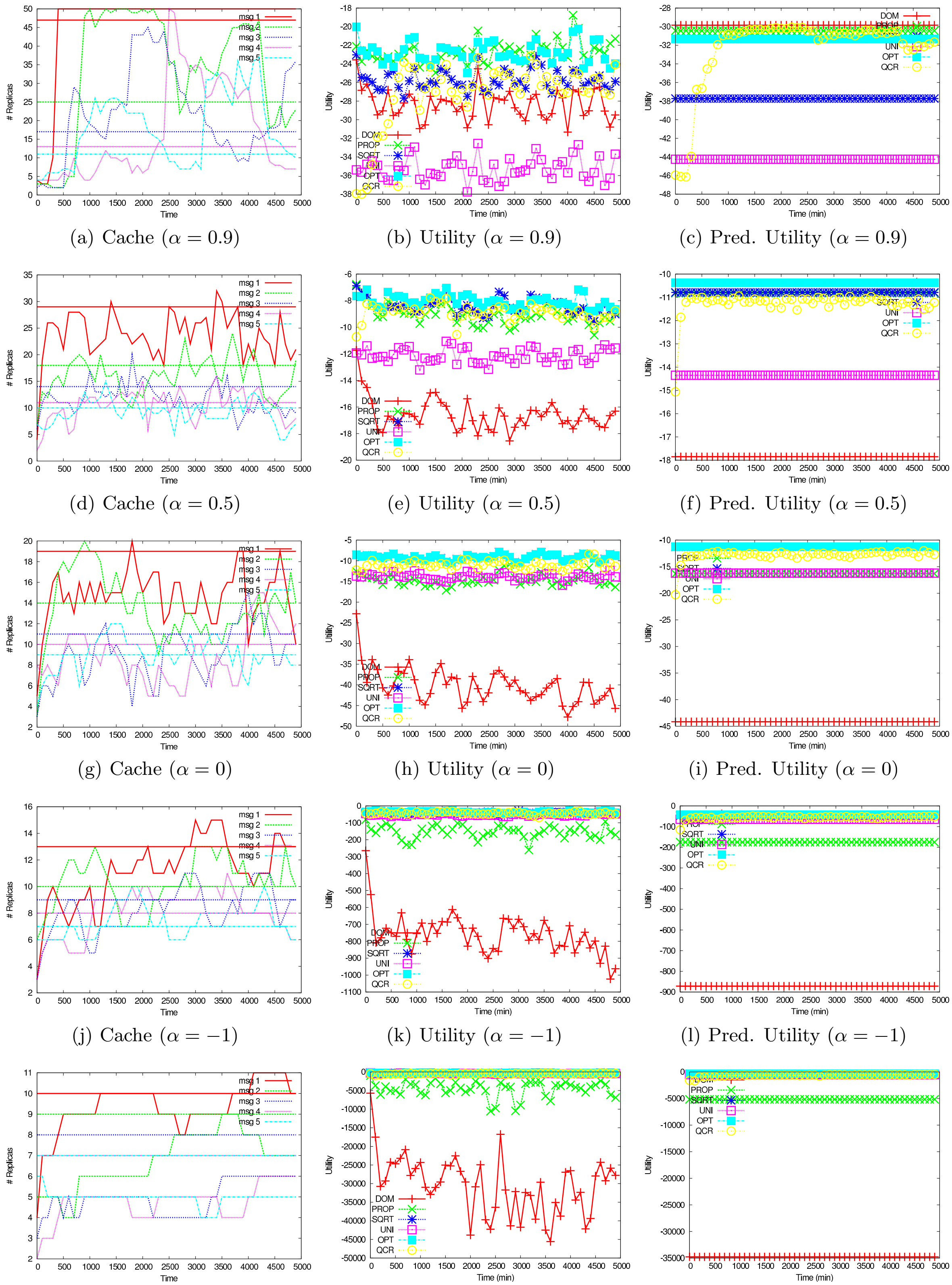
(d) Cache ($\tau = 10$) (e) Utility ($\tau = 10$) (f) Pred. Utility ($\tau = 10$)

(g) Cache ($\tau = 32$) (h) Utility ($\tau = 32$) (i) Pred. Utility ($\tau = 32$)

(j) Cache ($\tau = 100$) (k) Utility ($\tau = 100$) (l) Pred. Utility ($\tau = 100$)

(m) Cache ($\tau = 1000$) (n) Utility ($\tau = 1000$) (o) Pred. Utility ($\tau = 1000$)

Figure 9: Homogenous Mixing, Step Impatience: Experienced Utility, Predicted Utility, and Cache Evolution Over Time for $\mu = 0.05, \rho = 5, I = 50, N = 50, \omega = 1$



(a) Cache ($\alpha = 0.9$) (b) Utility ($\alpha = 0.9$) (c) Pred. Utility ($\alpha = 0.9$)

(d) Cache ($\alpha = 0.5$) (e) Utility ($\alpha = 0.5$) (f) Pred. Utility ($\alpha = 0.5$)

(g) Cache ($\alpha = 0$) (h) Utility ($\alpha = 0$) (i) Pred. Utility ($\alpha = 0$)

(j) Cache ($\alpha = -1$) (k) Utility ($\alpha = -1$) (l) Pred. Utility ($\alpha = -1$)

(m) Cache ($\alpha = -2$) (n) Utility ($\alpha = -2$) (o) Pred. Utility ($\alpha = -2$)

Figure 10: Homogenous Mixing, Power Impatience: Utility, Predicted Utility, and Cache Evolution Over Time for $\mu = 0.05, \rho = 5, I = 50, N = 50, \omega = 1$

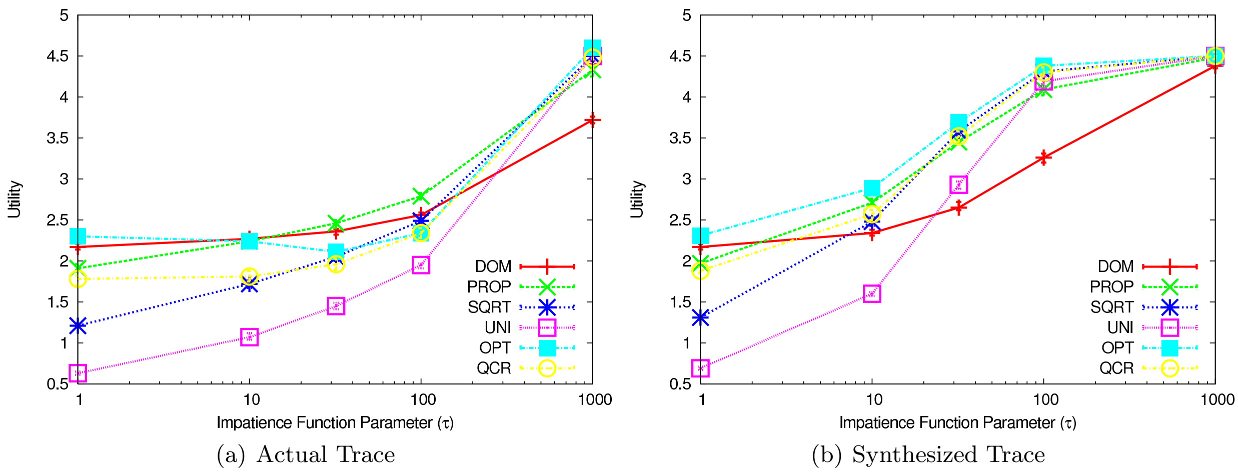
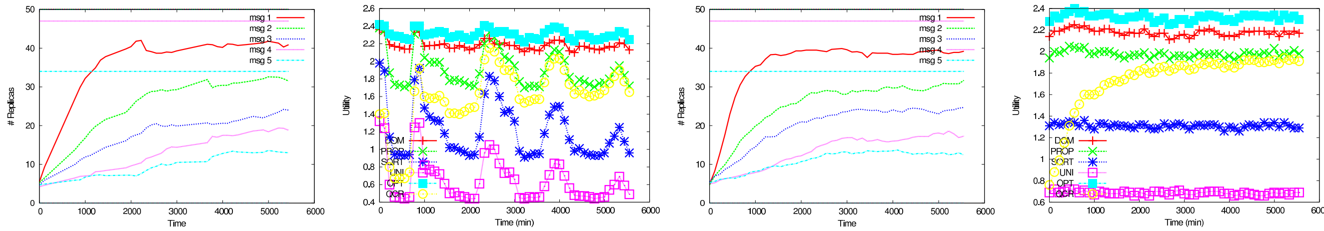
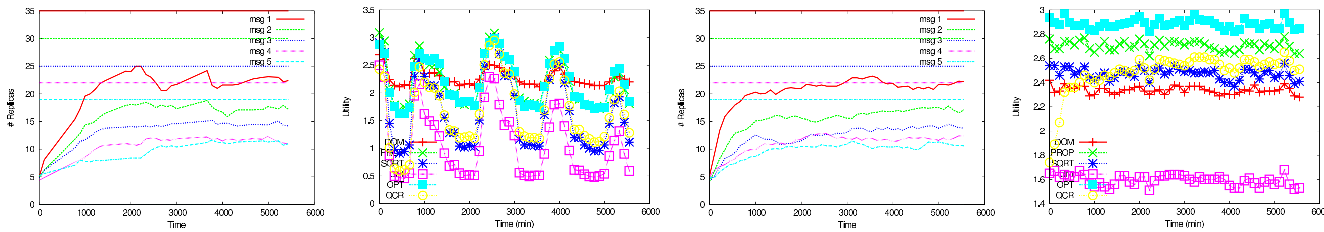


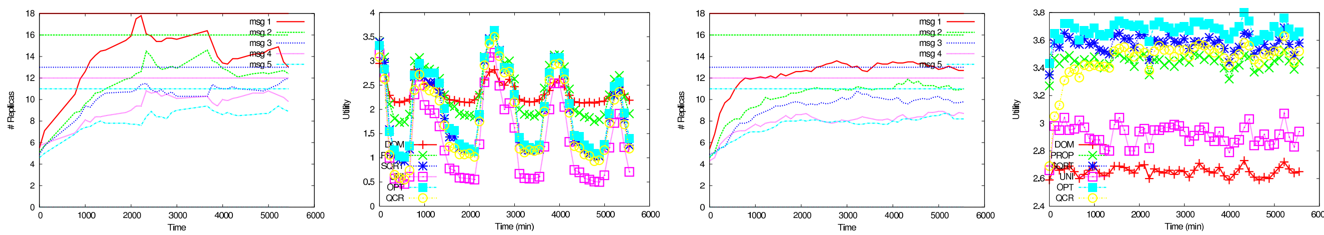
Figure 11: Infocom '06 Trace, Step Impatience - Understanding Impact of Time Statistics on Avg Behavior



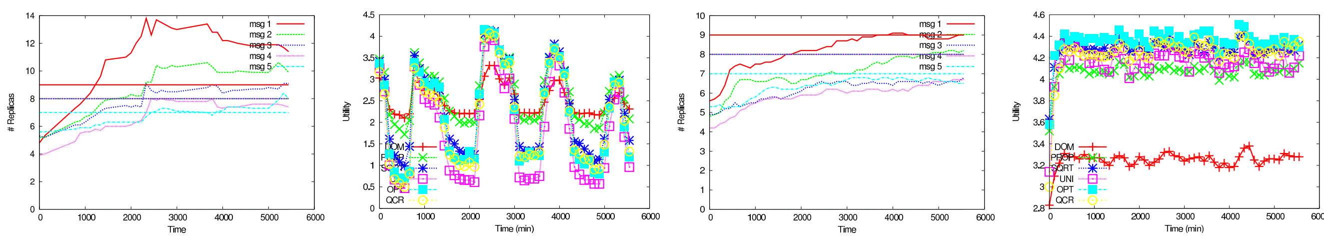
(a) Cache ($\tau = 1$) (b) Utility ($\tau = 1$) (c) (Synthetic) Cache ($\tau = 1$) (d) (Syntetic) Utility ($\tau = 1$)



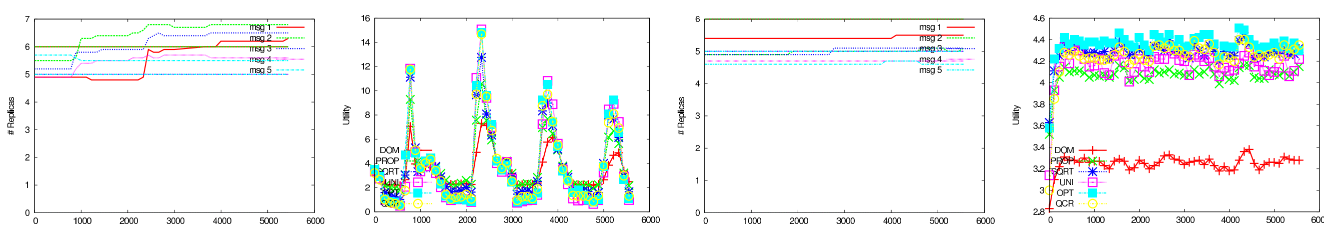
(e) Cache ($\tau = 10$) (f) Utility ($\tau = 10$) (g) (Synthetic) Cache ($\tau = 10$) (h) (Syntetic) Utility ($\tau = 10$)



(i) Cache ($\tau = 32$) (j) Utility ($\tau = 32$) (k) (Synthetic) Cache ($\tau = 32$) (l) (Syntetic) Utility ($\tau = 32$)



(m) Cache ($\tau = 100$) (n) Utility ($\tau = 100$) (o) (Synthetic) Cache ($\tau = 100$) (p) (Syntetic) Utility ($\tau = 100$)



(q) Cache ($\tau = 1000$) (r) Utility ($\tau = 1000$) (s) (Synthetic) Cache ($\tau = 1000$) (t) (Syntetic) Utility ($\tau = 1000$)

Figure 12: Infocom '06, Step Impatience - Timewise Results for $\rho = 5, I = 50, N = 50, \omega = 1$

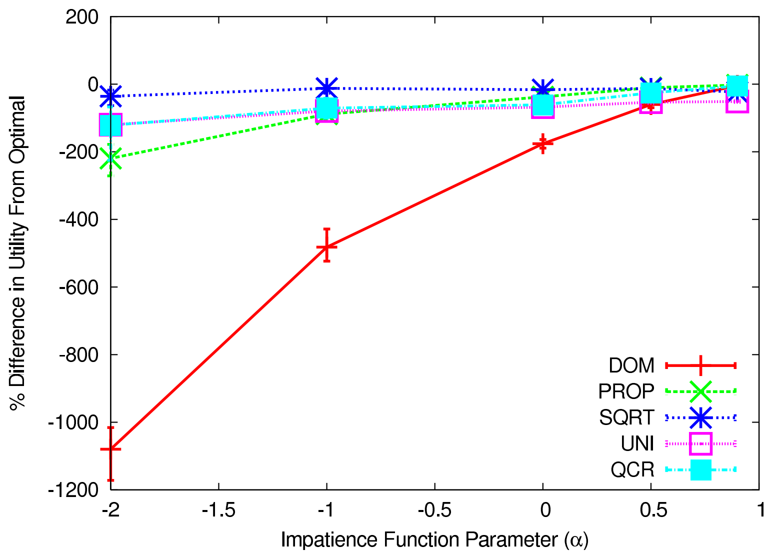
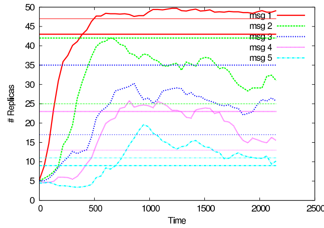
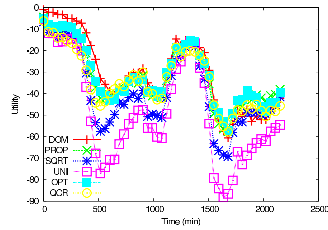
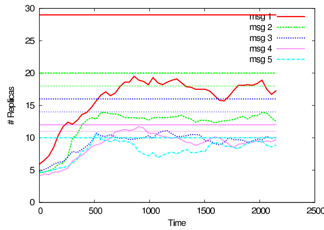


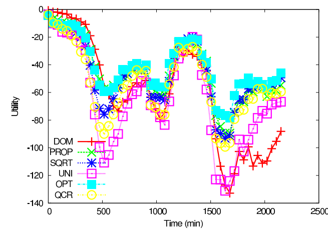
Figure 13: VANET Trace, Power Impatience, Util vs. α



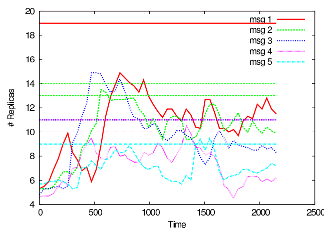


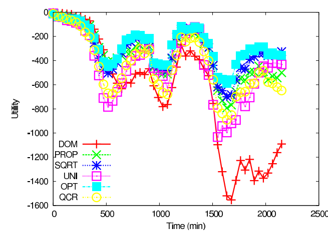
(a) MA VANET ($\alpha = 0.9$) (b) Cache VANET ($\alpha = 0.9$)



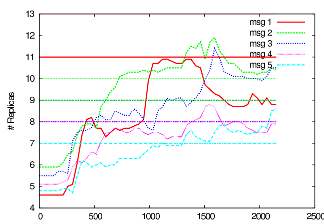


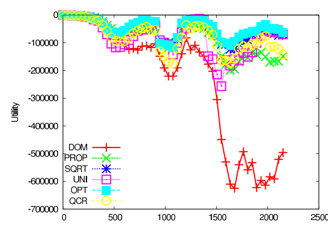
(c) MA VANET ($\alpha = 0.5$) (d) Cache VANET ($\alpha = 0.5$)



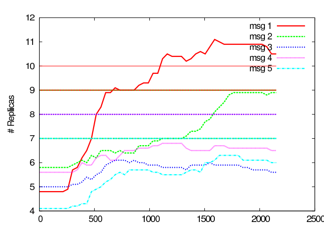


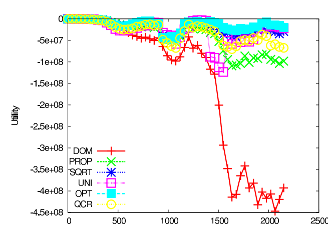
(e) MA VANET ($\alpha = 0$) (f) Cache VANET ($\alpha = 0$)





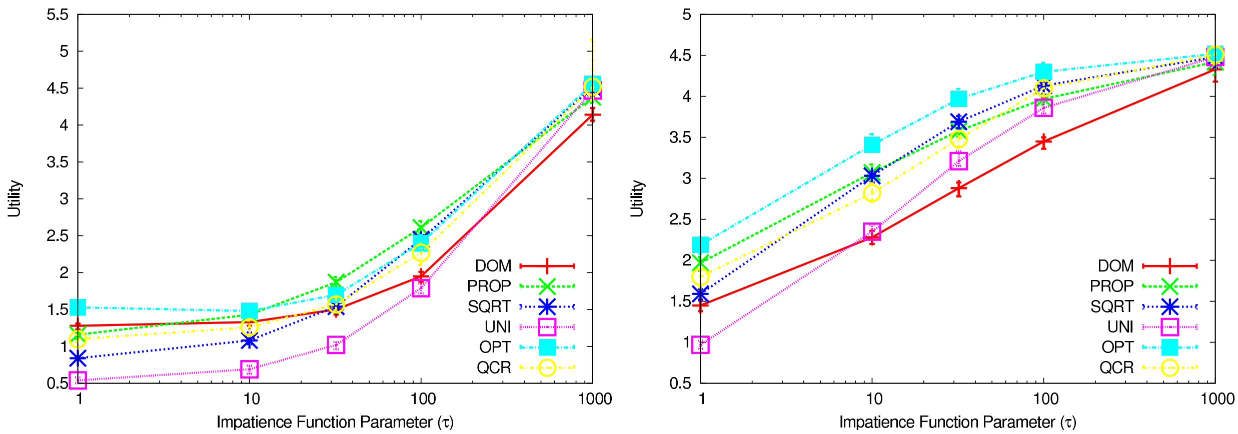
(g) MA VANET ($\alpha = -1$) (h) Cache VANET ($\alpha = -1$)





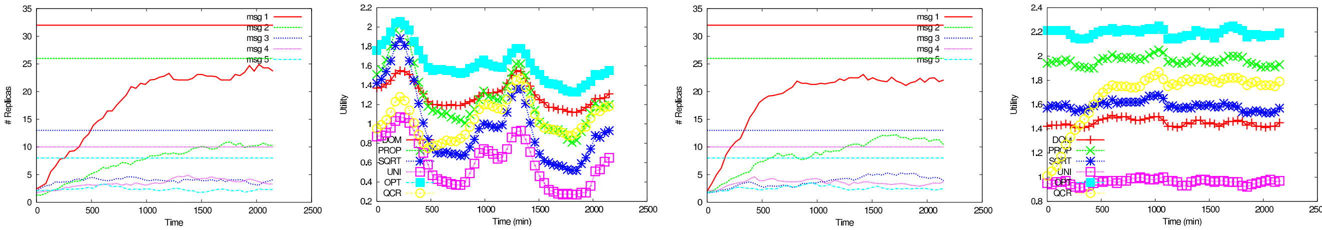
(i) MA VANET ($\alpha = -2$) (j) Cache VANET ($\alpha = -2$)

Figure 14: VANET Trace, Power Impatience for $\rho = 5, I = 50, N = 50, \omega = 1$

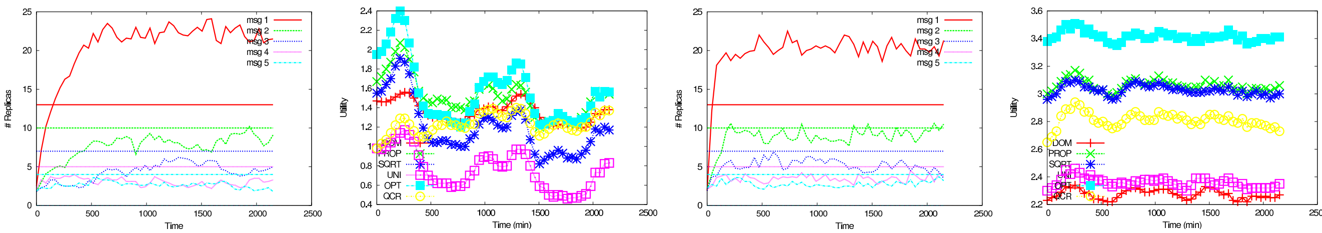


(a) Actual Trace (b) Synthesized Trace

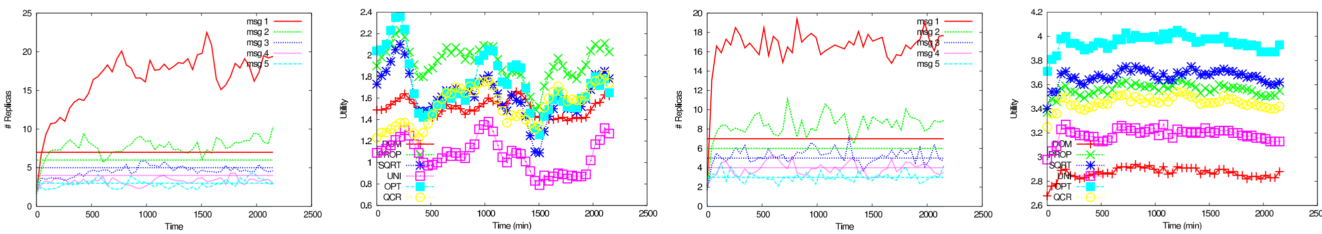
Figure 15: VANET Trace, Step Impatience ($\rho = 2$) - Understanding Impact of Time Statistcs on Avg Behavior



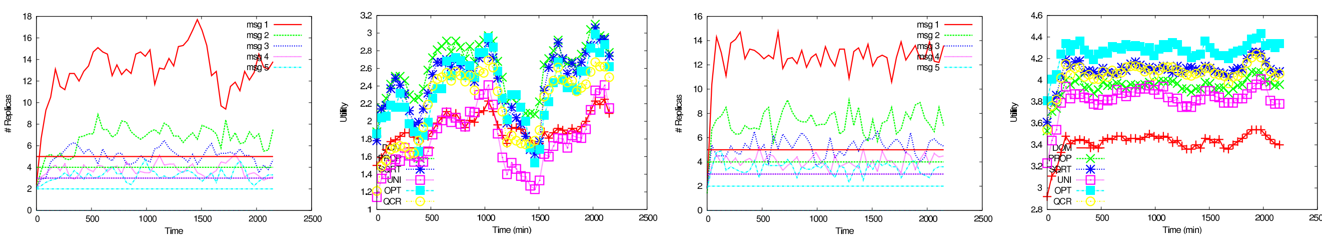
(a) Cache ($\tau = 1$) (b) Utility ($\tau = 1$) (c) (Synthetic) Cache ($\tau = 1$) (d) (Syntetic) Utility ($\tau = 1$)



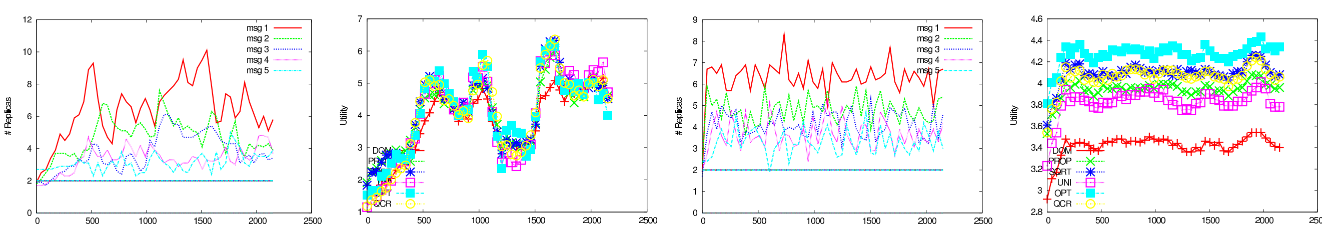
(e) Cache ($\tau = 10$) (f) Utility ($\tau = 10$) (g) (Synthetic) Cache ($\tau = 10$) (h) (Syntetic) Utility ($\tau = 10$)



(i) Cache ($\tau = 32$) (j) Utility ($\tau = 32$) (k) (Synthetic) Cache ($\tau = 32$) (l) (Syntetic) Utility ($\tau = 32$)

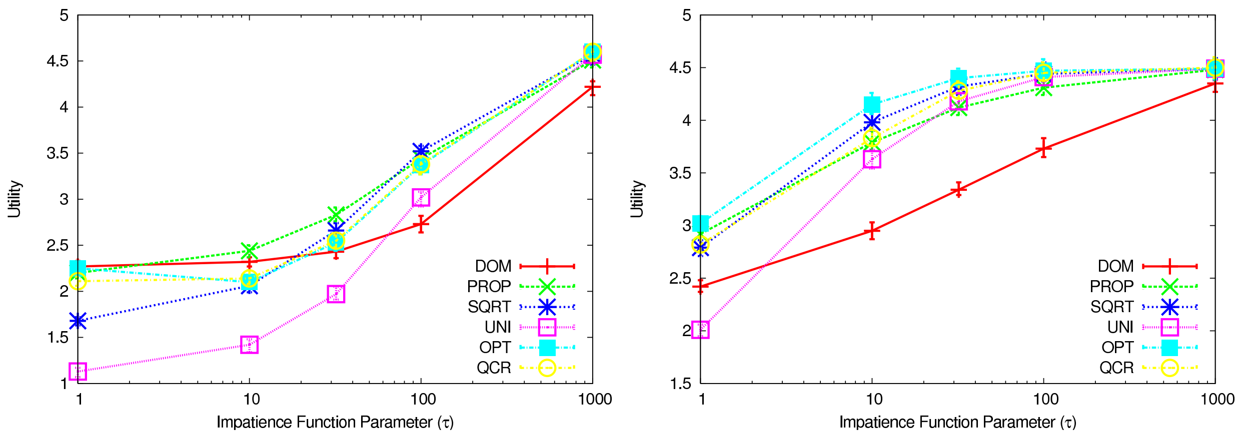


(m) Cache ($\tau = 100$) (n) Utility ($\tau = 100$) (o) (Synthetic) Cache ($\tau = 100$) (p) (Syntetic) Utility ($\tau = 100$)



(q) Cache ($\tau = 1000$) (r) Utility ($\tau = 1000$) (s) (Synthetic) Cache ($\tau = 1000$) (t) (Syntetic) Utility ($\tau = 1000$)

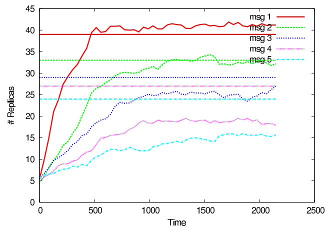
Figure 16: VANET, Step Impatience - Timewise Results for $\rho = 2, I = 50, N = 50, \omega = 1$



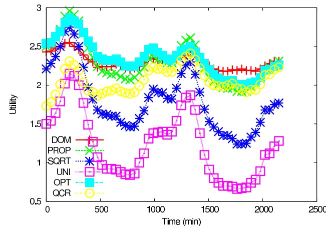
(a) Actual Trace

(b) Synthesized Trace

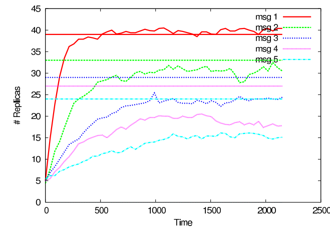
Figure 17: VANET Trace, Step Impatience - Understanding Impact of Time Statistcs on Avg Behavior



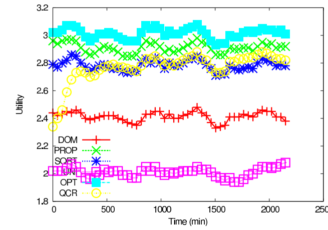
(a) Cache ($\tau = 1$)



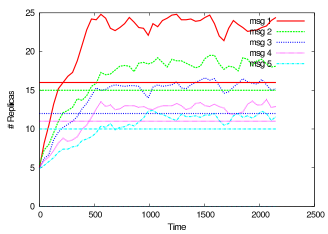
(b) Utility ($\tau = 1$)



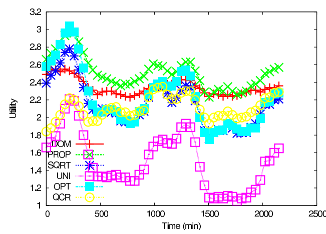
(c) (Synthetic) Cache ($\tau = 1$)



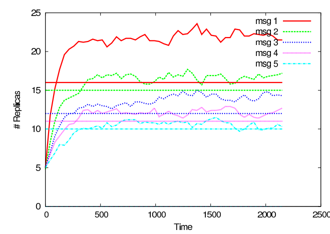
(d) (Syntetic) Utility ($\tau = 1$)



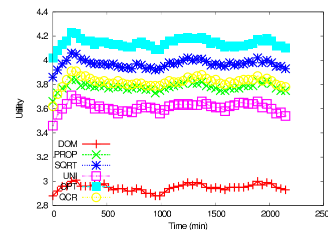
(e) Cache ($\tau = 10$)



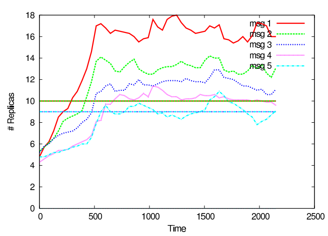
(f) Utility ($\tau = 10$)



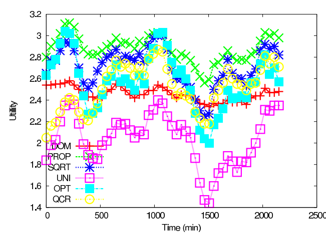
(g) (Synthetic) Cache ($\tau = 10$)



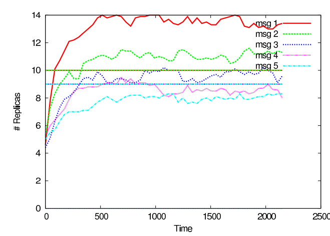
(h) (Syntetic) Utility ($\tau = 10$)



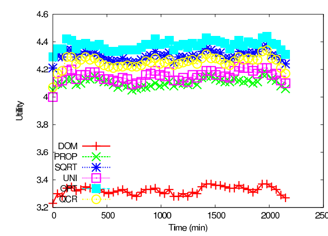
(i) Cache ($\tau = 32$)



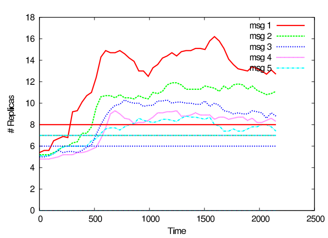
(j) Utility ($\tau = 32$)



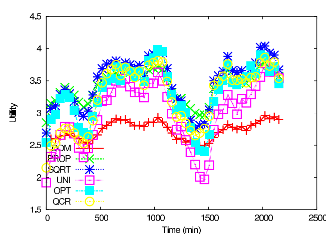
(k) (Synthetic) Cache ($\tau = 32$)



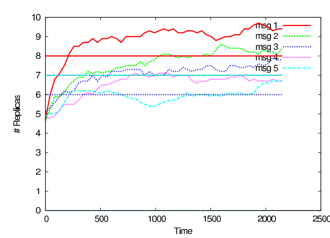
(l) (Syntetic) Utility ($\tau = 32$)



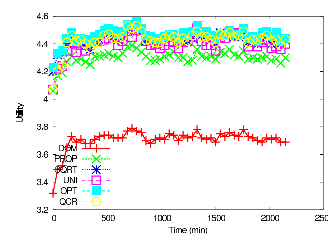
(m) Cache ($\tau = 100$)



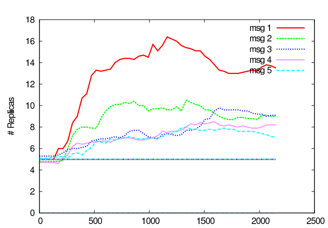
(n) Utility ($\tau = 100$)



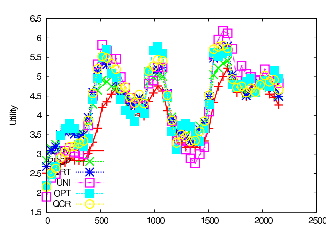
(o) (Synthetic) Cache ($\tau = 100$)



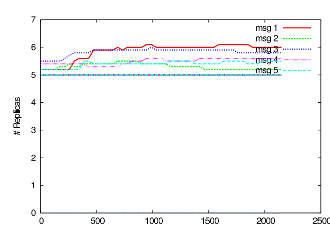
(p) (Syntetic) Utility ($\tau = 100$)



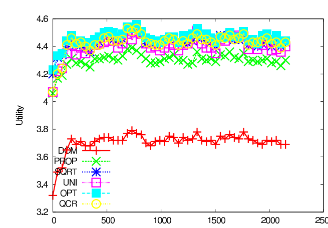
(q) Cache ($\tau = 1000$)



(r) Utility ($\tau = 1000$)

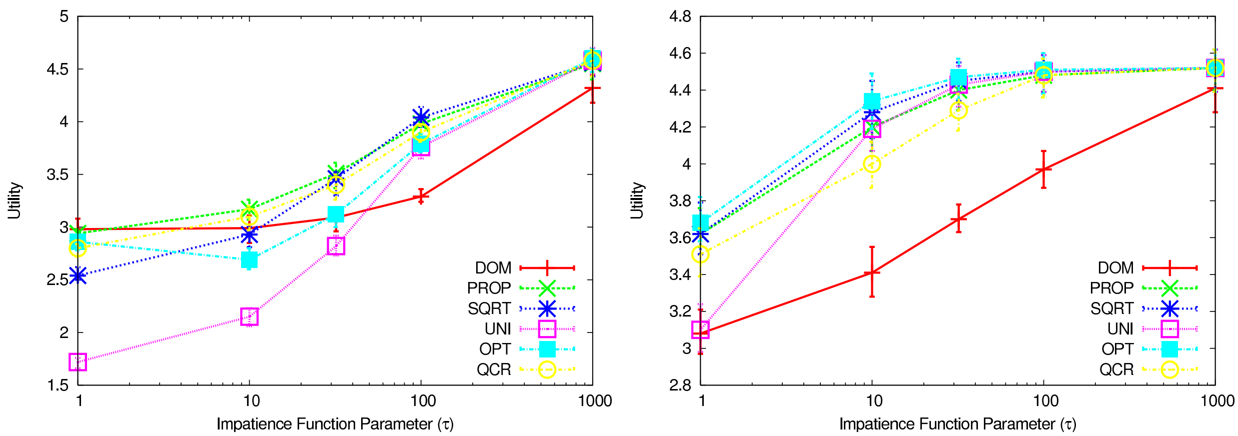


(s) (Synthetic) Cache ($\tau = 1000$)



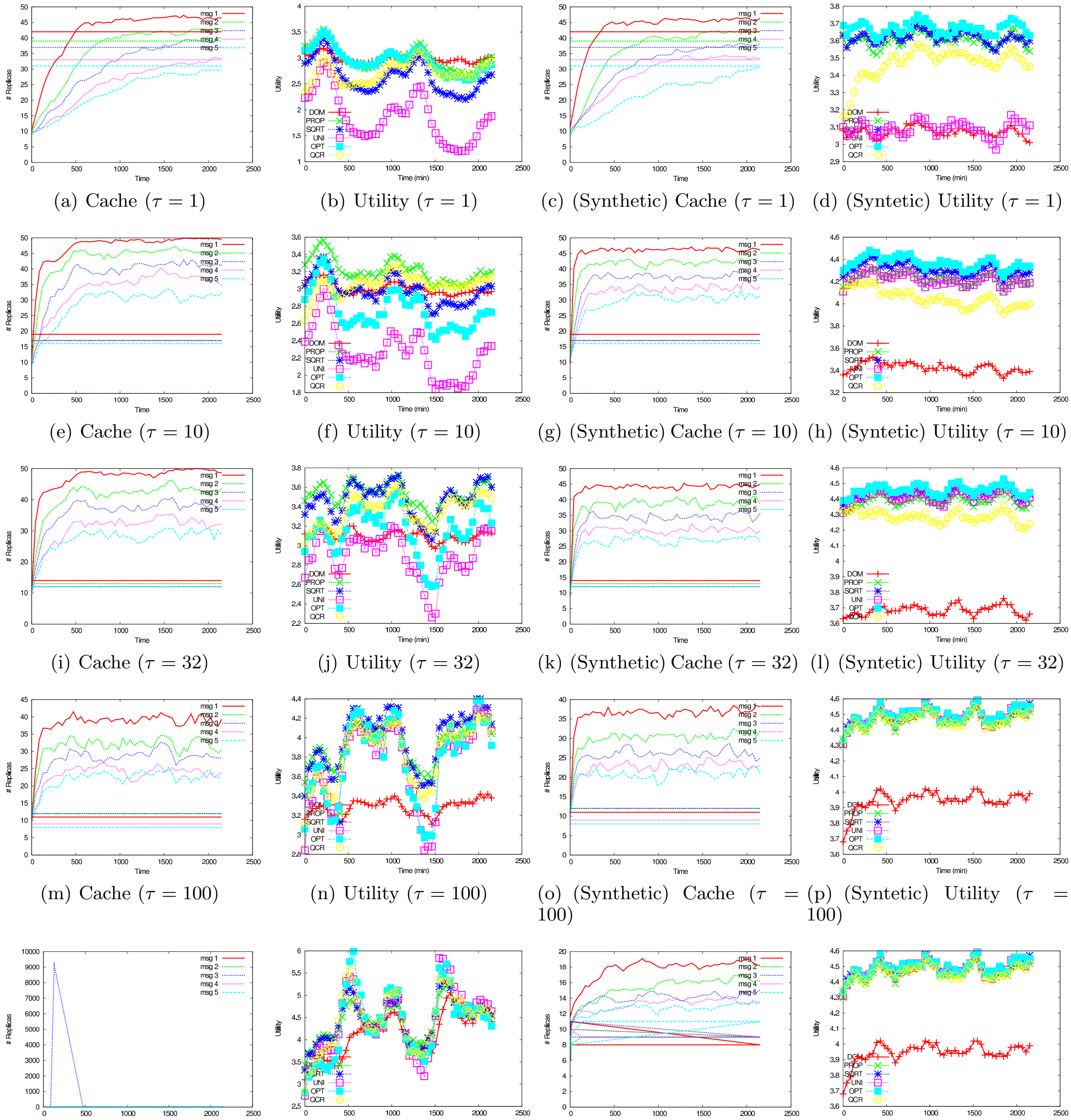
(t) (Syntetic) Utility ($\tau = 1000$)

Figure 18: VANET, Step Impatience - Timewise Results for $\rho = 5, I = 50, N = 50, \omega = 1$



(a) Actual Trace (b) Synthesized Trace

Figure 19: VANET Trace, Step Impatience ($\rho = 10$) - Understanding Impact of Time Statistcs on Avg Behavior



(a) Cache ($\tau = 1$)
(b) Utility ($\tau = 1$)
(c) (Synthetic) Cache ($\tau = 1$)
(d) (Syntetic) Utility ($\tau = 1$)

(e) Cache ($\tau = 10$)
(f) Utility ($\tau = 10$)
(g) (Synthetic) Cache ($\tau = 10$)
(h) (Syntetic) Utility ($\tau = 10$)

(i) Cache ($\tau = 32$)
(j) Utility ($\tau = 32$)
(k) (Synthetic) Cache ($\tau = 32$)
(l) (Syntetic) Utility ($\tau = 32$)

(m) Cache ($\tau = 100$)
(n) Utility ($\tau = 100$)
(o) (Synthetic) Cache ($\tau = 100$)
(p) (Syntetic) Utility ($\tau = 100$)

(q) Cache ($\tau = 1000$)
(r) Utility ($\tau = 1000$)
(s) (Synthetic) Cache ($\tau = 1000$)
(t) (Syntetic) Utility ($\tau = 1000$)

Figure 20: VANET, Step Impatience - Timewise Results for $\rho = 10, I = 50, N = 50, \omega = 1$