A Lightweight Distributed Solution to Content Replication in Mobile Networks

C.-A. La*, P. Michiardi*, C. Casetti‡, C.-F. Chiasserini‡, M. Fiore‡
* EURECOM, Sophia Antipolis, France
‡ Politecnico di Torino, Torino, Italy

***Abstract*—Performance and reliability of content access in mobile networks is conditioned jointly by the number and location of content replicas deployed at the network nodes. The endeavour of this work is to address such an optimization problem with a *distributed, lightweight* solution that handles network dynamics. We devise a mechanism that lets nodes share the burden of storing and providing content, so as to achieve load balancing, and decide whether to replicate or drop the information so as to adapt to a dynamic content demand and time-varying topology. Simulation results show that our mechanism, which uses *local measurements only*, is: (i) extremely precise in approximating an optimal solution to content placement and replication; (ii) robust against network mobility; (iii) flexible in accommodating variation in time and space of the content demand.**

I. Introduction

Research and industrial activity in the networking field is pursuing the idea that networks should provide access to contents, rather than to hosts. Recently, this goal has been extended to wireless networks as well, as demonstrated by the tremendous growth of services and applications offered to users equipped with advanced mobile terminals, such as the iPhone.

The inexorable consequence of a steady increase in mobile data traffic exerted by mobile devices fetching content from the Internet is a drainage of mobile operators' network resources [1], [2]. A promising approach to solve this problem is *content replication*, which has been shown to be effective in enhancing performance and reliability of content access for end-users (see, e.g., [3] for a survey on the topic).

In this paper, we explore the concept of content replication in a cooperative wireless environment, where content demand and topology are dynamically changing. Nodes can potentially store data and serve other users through device-to-device communications (e.g., using IEEE 802.11 or Bluetooth). We consider that content has a validity time, after which it has to be discarded and a new version has to be downloaded from a server in the Internet. Furthermore, not all users in the network may be interested in a given content at a given time; hence, disseminating the information to the nodes according to an epidemic approach [9], or pushing the content to all users, might not be desirable.

Such a scenario introduces several problems to content replication. *Optimal replica placement* is one of those: selecting the location that is better suited to store content is difficult, especially when the network is dynamic. Another prominent issue is *how many content replicas* should be made available to mobile nodes. Clearly, decisions on the placement and number of replicas to be deployed in a network are tightly related problems: intuitively, the latter introduces a feedback loop to the former as every content replication triggers a new instance of the placement problem.

Traditionally, the above content replication problems have been studied through the lenses of classic Facility Location Theory [4]. Optimal placement can be cast as the *uncapacitated k-median* problem, whereas the joint optimization of placement and number of replicas can be studied as an *uncapacitated facility location* problem; both these problems are NP-hard for general network topologies.

In our previous work [5], we showed preliminary results indicating that a uniformly distributed replica placement can be well approximated using distributed store-and-forward mechanisms, in which nodes store content only temporarily. The endeavor of this work is to extend our previous study and target the *joint problem* (i) of establishing the number of replicas to deploy in a dynamic network and (ii) of finding their most suitable location, so as to achieve load balancing, that is, to let the network nodes evenly share the burden of storing and providing content.

Instead of designing distributed approximation algorithms of the optimal solution to facility location problems, which either require global (or extended) knowledge of the network [6], [7] or are unpractical [8], we extend our store-and-forward mechanism with a distributed replication algorithm that bases its decisions on local measurements only and aims at evenly distributing among nodes the demanding task of being a replica provider. As a result, we show that both optimal placement and content replication can be approximated through a lightweight, distributed scheme which adapts to different initial distributions of replicas and to variation in time and in space of content demand, while being robust to changes due to network dynamics.

II. Related work

Simple, widely used techniques for replication are gossiping and epidemic dissemination [9], [10], where the information is forwarded to a randomly selected subset of neighbors. Although our scheme may resemble this approach in that a replica node hands over the content to a randomly chosen neighbor, the mechanism we propose and the goals it achieves (i.e., approximation of the optimal number of replicas and their placement) are significantly different.

Another viable approach to replication is represented by quorum-based [11] and cluster-based protocols [12]. Both methods, although different, are based on the maintenance of quorum systems or clusters, which in mobile networks are likely to cause an exceedingly high overhead. Node grouping is also exploited in [13], where groups of nodes with stable links are used to cooperatively store contents and share information. The schemes in [13], however, require an a-priori knowledge of the query rate, which is assumed to be constant in time. Note that, on the contrary, our lightweight solution can cope with a dynamic demand, whose estimate by the replica nodes is used to trigger replication.

Threshold-based mechanisms for content replication are proposed in [14], [15]. In particular, in [14] it is the original server that decides whether to replicate content or not, and where. In [15], nodes have limited storage capabilities: if a node does not have enough free memory, it will replace a previously received content with a new one, only if it is going to access that piece of information more frequently than its neighbors up to h hops. Our scheme significantly differs from these works, since it is a totally distributed, extremely lightweight mechanism, which accounts for the content demand by other nodes and ensures a replica density that autonomously adapts to the changes in the query rate over time and space.

Finally, we point out that the RWD scheme was first proposed in our work [5]. That paper, however, besides being a preliminary study, focused on mechanisms for content handover only: no replication or content access were addressed.

III. Background and problem statement

Here, we inherit the problem of replication typical of the wired Internet and we discuss why the dynamic nature of wireless networks introduces new challenges with respect to the wireline counterpart.

We focus on *replication* and *replica placement* problems, i.e., we view content replication as a process of its own, rather than a by-product of a query/caching mechanism [3]. We investigate a scenario involving users equipped with devices offering Internet broadband connectivity as well as device-to-device communication capabilities (e.g., through IEEE 802.11). Although we do not concern ourselves with the provision of Internet access in wireless networks, we remark that broadband connectivity is where new content is fetched from (and updated).

In order to provide a basic description of the system, we focus on content being represented by a *single* information object. The mechanisms we describe can then be extended to multiple objects. We assume the object to be tagged with a *validity time* and originally hosted on a server in the Internet, which can only be accessed through the broadband access we hinted at. We then consider a *cooperative network environment* composed of a set $V = \{v(1), ..., v(N)\}$ of mobile nodes. A node $v(j)$ wishing to access the content first tries to retrieve it from other devices; if its search fails, the node downloads a fresh content replica from the Internet server and temporarily stores it for a period of time $\tau_{v(j)}$, termed *storage time*. For simplicity of presentation, in the following we assume $\tau_{v(j)} = \tau, \forall v(j) \in V$. During the storage time, $v(j)$ serves the content to nodes issuing requests for it and, possibly, downloads from the Internet server a fresh copy of the content if its validity time has expired. We assume that a node $v(i)$, which at a given time t does not store any copy of the content and which will later be referred to as "content consumer", issues queries at a rate $\lambda_{v(i)}(t)$.

To achieve load balancing, at the end of the storage time $v(j)$ has to decide whether (1) to hand the content over to another node, (2) to drop the copy, or (3) to replicate the content and hand over both copies. We refer to the nodes hosting a content copy at a given time instant as *replica nodes*, and we denote their set by $\mathcal{C}(t)$. Only replica nodes are responsible for updating the content and for injecting a new version in the wireless network.

Next, to highlight our contribution with respect to previous work, we relate our study to the formulation of the replication and replica placement problems typically used in the literature. Let us fix the time instant and drop the time dependency for ease of notation. Then, let $G = (V, E)$ represent the network graph at the given time, defined by a node set V and an edge set E. $\mathcal{C}$ is the set of facility nodes, i.e., nodes holding a replica. The following definition specifies the uncapacitated facility location problem, i.e., the joint optimization of the number of replicas to install and their placement.

Definition 1: Uncapacitated facility location. Given the node set V with pair-wise distance function d, service demand $\lambda_{v(j)}$ and cost for opening a facility at $v(j)$ equal to $f(v(j))$, $\forall v(j) \in V$, select a set of nodes to act as facilities so as to minimize the joint cost $C(V, \lambda, f)$ of acquiring the facilities and servicing the demand:

$$C(V, \lambda, f) = \sum_{\forall v(j) \in \mathcal{C}} f(v(j)) + \sum_{\forall v(j) \in V} \lambda_{v(j)} d(v(j), m(v(j)))$$

where $m(v(j)) \in \mathcal{C}$ is the facility that is *closer* to $v(j)$.

The uncapacitated k-median problem can be defined as above, but the number k of replicas to place is given as input and the cost function ignores the first additive term accounting for installation costs. For general graphs, both the above problems are NP-hard [16] and a variety of approximation algorithms have been developed, which however require global (or extended) knowledge of the network state [6].

Which new problems are introduced in the context of our work? (i) Node mobility introduces the problem of a dynamic graph G, requiring that the facility location problem be solved upon every network topology or demand rate change. (ii) Even under static topology and constant demand, solving the facility location problem does not yield load balancing among nodes: the optimal location of replicas inherently imposes the burdern of serving content to a specific and invariant set of nodes, that could be brought to energy depletion.

Our main contribution is therefore the design of a mechanism for content placement and replication that achieves load balancing as both the network topology and the query rate vary.

IV. Distributed mechanism for replication and placement problems

The workload experienced by a replica node is determined by the mechanism used by nodes to access the content through device-to-device communications. We identify two phases: a content query transmission, and a query reply transmission (by the replica node carrying the desired content). For the content query transmission, we assume that *perfect discovery* is adopted, as typically done in the literature[1]. According to perfect-discovery, nodes can access a centralized content-location service that returns the identity of the closest content replica in terms of euclidean distance. We do not address the problem of how the centralized service is updated, save by noting that it is certainly responsible for additional overhead and complexity, and that it can be managed through a separate protocol using unicast or multicast transmissions. A query is propagated using application-driven broadcast, but only the intended replica node (specified in the query) will serve the content; any other replica node will discard the request. Also, we assume that the identity of the nodes that have relayed the query is added to the query message itself. As the replica node with the desired content is reached, it will reply to the node issuing the query through a multihop transmission process that backtracks the path from the replica node to the querying node, exploiting the identity of relay nodes included in the query message. This backtracking, although possibly occurring through multiple hops, makes no use of ad hoc routing protocols, as it is completely application-driven.

Next, we examine the challenging problem of replica placement and discuss the behavior of replica nodes as a function of the system workload, in search of a cooperative, distributed content replication strategy in presence of changing demand.

A. Replica placement

We now review the *Random-Walk Diffusion (RWD)* mechanism we proposed in [5], which, given a fixed number k of replicas as input, can be used to approximate an optimal placement thereof. In Sec. IV-B, we build on RWD and extend it to address the joint optimization required by the facility location problem, which is the object of this work.

According to RWD, a mobile device, hosting a content replica, stores it for a storage time τ. At the end of its storage time, the replica node selects with equal probability one of its neighbors to store the content for the following storage period. Thus, content replicas roam the network by moving from one node to another, randomly, at each time step τ.

As discussed in Sec. III, at a fixed time instant, replica placement can be cast as the uncapacitated k-median problem. We thus construct a baseline replica placement, to be compared to that obtained with RWD, by applying the (centralized) approximation algorithm in [6] in presence of various network deployments.

[1] For sake of brevity, in this work we essentially focus on content replication and gloss over the important question of content access, by assuming perfect discovery. In our technical report [17] we explore alternative content access mechanisms, show their impact on replication and present application performance results.

In Sec. V we employ the well-known χ^2 goodness-of-fit test on the distribution of inter-distance between content replicas for the baseline and the RWD case, which allows to understand how well RWD approximates an optimal placement.

B. Content replication

We now focus on the more general problem of the uncapacitated facility location, defined in Sec. III, where the optimal number of replicas (facilities) to be placed in the network is to be determined along with their location. In particular, we want to answer the following questions.

1) Given a set of demand points that exhibit a homogeneous query rate λ, what is the optimal number of content replicas that should be deployed in the network to achieve load balancing?
2) Is it possible to design a lightweight distributed algorithm that approximates this optimal number of replicas in presence of a dynamic demand and time-varying topology?

We address these questions by extending the RWD mechanism described in Sec. IV-A. Again, we fix the time instant and, for simplicity, we drop the time dependency from our notation. Let the network be described by the graph $G = (V, E)$, with $|V| = N$ nodes deployed on an area $\mathcal{A}$. Also, recall that $\mathcal{C}$ and $V \setminus \mathcal{C}$ represent the sets of content replicas and of nodes issuing requests, respectively.

Given G and the query rate λ, the uncapacitated facility location problem amounts to the joint optimization of the number of replicas and their locations in the network. The original RWD mechanism achieves a good approximation of the optimal placement in mobile networks, but ignores the cost to deploy a content replica. Now, with reference to Def. 1, we define a *non-uniform* cost function, $f(v(j))$, $\forall v(j) \in \mathcal{C}$, to deploy content replicas in the network:

$$f(v(j)) = |s_{v(j)} - s_R| \tag{1}$$

where $s_{v(j)}$ is the workload expressed as number of queries served by replica node $v(j)$ during its storage time, and s_R is a reference value for the workload that node $v(j)$ is willing to support. We assume the case where all replica nodes are willing to serve the same amount of queries, although our study can be easily extended to the case of different values of s_R. Eq. (1) indicates that the cost for replica node $v(j)$ grows with the gap between its workload (function of the number of "closest" content consumers, hence the non-uniformity) and the reference value s_R.

Our replication mechanism only involves replica nodes, which are responsible to decide whether to replicate, hand over or drop content based on local measurements of their workload. During storage time τ, the generic replica node $v(j)$ *counts* the number of queries that it serves, i.e., $s_{v(j)}$. When the storage time expires, the replica node compares $s_{v(j)}$ to s_R. Decisions are taken as follows:

$$\text{if } s_{v(j)} - s_R \begin{cases} > \epsilon & \text{replicate} \\ < -\epsilon & \text{drop} \\ \text{else} & \text{hand over} \end{cases}$$

where ϵ is a tolerance value to avoid replication/drop decisions in case of small changes in the node workload.

The rationale of our mechanism is the following. If $s_{v(j)} > s_R$, replica node $v(j)$ presumes the current number of content replicas in the area to be insufficient to guarantee the expected workload s_R, hence the node replicates the content and hands the copies over to two of its neighbors (one each), following the RWD placement mechanism (Sec. IV-A). The two selected neighbors will act as replica nodes for the subsequent storage time. Instead, if $s_{v(j)} < s_R$, replica node $v(j)$ thinks that the current number of replicas in the area is exceeding the total demand, and just drops the content copy. Finally, if the experienced workload is (about) the same as the reference value, $v(j)$ selects one of its neighbors to hand over the current copy.

We stress that replication and placement are tightly related. For example, if content demand varies in time or in space (e.g., only a fraction of all nodes located in a sub-zone of the network area issue queries), both the number of replicas and their location must change. Thanks to the fact that replica nodes take decisions based on the measured workload, our solution can dynamically adapt to a time- or space-varying query rate, as will be shown by our simulation results. When instead the content demand is constant and homogeneous, our handover mechanism ensures load balancing among the network nodes.

In the following, we set up a simulation environment to evaluate the behavior of our mechanism when the wireless network is both static and dynamic; also, we characterize the time the system takes to approximate an optimal number of content replicas.

V. Simulation-based evaluation

We implemented our replica placement and content replication mechanism in the ns-2 simulator. For each experiment described in the following, we execute 10 simulation runs and report averaged results. Our statistics are collected after an initial warm-up period of 500 s, for ns-2 and the mobility model employed in our simulations to reach a steady state regime.

In our simulations, which lasted for almost 3 hours of simulated time (10000 s), we assume nodes to be equipped with a standard 802.11 interface, with an 11 Mbps fixed data transmission rate and a radio transmission range of 100 m. We consider a single content, whose size is of the order of 1 KB. In our evaluation we do not simulate cellular access. We point out that all standard MAC-layer operations are simulated, which implies that both queries and replies may be lost due to typical problems encountered in 802.11-based ad hoc networks (e.g., collisions or hidden terminals).

We place $N = 320$ nodes uniformly at random on a square area $\mathcal{A}$ of 1 km^2, with a resulting average node degree of about 10 neighbors. We simulate node mobility using the *stationary* random waypoint model where the average node speed is set to 1 m/s and the pause time is set to 100 s. These settings are representative, for example, of people using their mobile devices as they walk.

For the perfect discovery access scheme, if a query fails (i.e., no answer is received after 2 s), a new request is issued, up to a total of 5 times. Finally, the tolerance value ϵ used in the replication/drop algorithm is equal to 2, unless otherwise stated[2]; for all nodes, the storage time τ is set to 100 s, the user request rate is $\lambda = 0.01$ req/s, and the reference workload for a replica node is equal to $s_R = 10$.

A. Results

We present the main results of our work organized in a series of questions. We focus on the mobile scenario, but present results for a static network when the comparison is relevant.

1) How well does our replica placement approximate the optimal distribution? Here we assume a known number of content replicas to be deployed ($|\mathcal{C}|$=30), i.e., we consider the k-median problem discussed in Sec. III. We measure the accuracy of our distributed replica placement mechanism using the χ^2 goodness-of-fit test on the inter-distance between replicas, as explained in Sec. IV-A. Considering a mobile network, we compute the distribution of replica nodes as follows: every τ seconds we take a snapshot of the network in its current state, we compute the optimal replica placement, by solving the k-median problem through the centralized local-search algorithm in [6], and we use the χ^2 test against the distribution achieved by our mechanism.

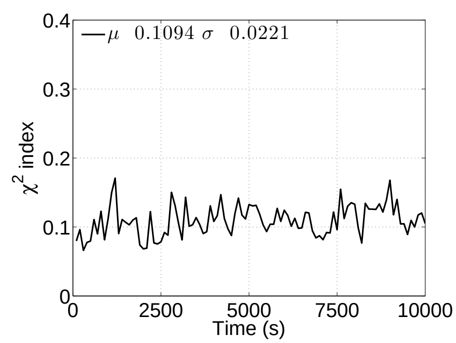

Fig. 1. Temporal evolution of the χ^2 index in a mobile scenario ($|\mathcal{C}|$=30 and τ=100 s).

Fig. 1 shows that our scheme does an excellent good job of approximating the optimal replica placement. In particular, the temporal evolution of the χ^2 index suggests that our replica placement mechanism is able to approximate very well the optimal solution[3], despite network dynamics.

2) Is the replication mechanism effective in reaching a target number of replicas? We now turn our attention to the *uncapacitated facility location* described in Sec. III and study how well the replication mechanism defined in Sec. IV-B approximates the joint problem of replication and placement.

Here we consider a scenario in which only one copy of the content is initially present in the network and we focus on the

[2]As shown later, the choice of ϵ has an impact on the time required for the system to reach an ideal number of replicas and to oscillate around this value. The choice of $\epsilon = 2$ was made after a careful analysis of our results.

[3]A $\chi^2 \approxeq 3$ is assumed to indicate a good match between two distributions [18].

evolution in time of the number of replicas in the system. We omit the temporal evolution of the χ^2 index, since our results are consistent with what we have observed for the placement scheme without replication.

Fig. 2 shows the temporal evolution of the total number of replicas $|\mathcal{C}|$ for the mobile scenario, against a reference line representing the optimal number of content replicas. Finding the optimal number of content replicas amounts to solving the uncapacitated facility location problem for a given network graph. To this end, we have implemented the *centralized* algorithm in [6] and computed an approximation to the optimal solution over several snapshots of the network graph. With reference to Def. 1, we set a non-uniform cost to open a facility as defined in Eq. 1. Intuitively, the cost to select a node to hold a content replica is proportional to its degree: a highly connected node will most likely attract more demand from content consumers.

For the parameters used in our simulations, the solution of the centralized algorithm indicates that the target number of replicas the system should reach is $|\mathcal{C}^*| = 30$.

Fig. 2 indicates that the number of content replicas we achieve with our scheme strikingly matches the target value: in steady state, the average relative error is less than 2%.

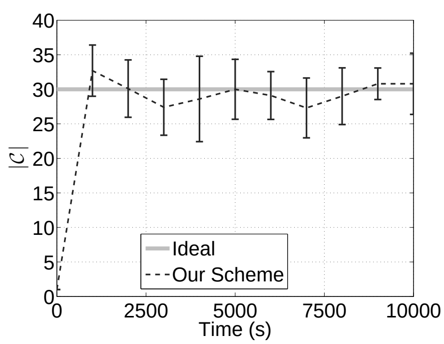

Fig. 2. Temporal evolution of the number of replicas, for a network bootstrapping with $|\mathcal{C}| = 1$ in a mobile scenario ($\lambda = 0.01$, $s_R = 10$, $\tau = 100$ s, $|\mathcal{C}^*| = 30$).

3) How is the total workload shared among replica nodes? As before, we study the joint placement and replication problem and we use the extreme scenario in which the network is initialized with only one content replica. Tab. I shows the 25%, 50% and 75% quantiles of the workload for each replica node, aggregated over the simulation time. As expected, the average load roughly matches the reference value $s_R = 10$, both in the static and mobile scenario.

TABLE I
AGGREGATE WORKLOAD DISTRIBUTION FOR REPLICAS FOR A NETWORK BOOTSTRAPPING WITH $|\mathcal{C}| = 1$ ($\lambda = 0.01$, $s_R = 10$, $\tau = 100$ S).

Percentile	25th	50th	75th	Mean
Static	4	8	14	9.73
Mobile	5	8	13	9.77

4) What is the convergence time of the replication mechanism? Convergence time should be carefully defined in our context: clearly, our mechanism cannot settle to a static, unique content replica placement, nor can it stabilize on a unique number thereof. For placement, it is not our intent to statically assign the role of content replica to a node and deplete nodal resources: we seek to balance the workload across all network nodes. We assume the network to have converged to a steady state when the difference between the reference value computed using the centralized local search algorithm and the experimental number of replicas is within 2%.

Again, we consider a scenario in which only one copy of the content is initially present in the network. Tab. II illustrates how convergence time (labelled t_s) varies with the storage time τ and the tolerance value ϵ. We also performed experiments to study the impact of the network size: we have observed a linear growth of the convergence time with N. Since the storage time τ is used to trigger replication/drop decisions, we expect to see a positive correlation between τ and convergence time: Tab. II confirms this intuition. We note that there is a trade-off between the convergence time and the message overhead: a small storage time shortens the convergence time at the cost of an increased number of content movements from a node to another. As for the impact of the tolerance parameter ϵ, our experiments indicate that a very reactive scheme would yield smaller convergence times, at the risk of causing frequent oscillations around a target value.

TABLE II
AVERAGE CONVERGENCE TIME t_S AS A FUNCTION OF THE STORAGE TIME τ ($\epsilon = 2$) AND THE TOLERANCE FACTOR ϵ ($\tau = 100$ S).

τ (s)	t_S (s)
20	800
100	1700
200	2300

ϵ	t_S (s)
0	700
2	1700
5	1900

5) What is the impact of variations in time and in space of the content demand? We now focus on the behavior of content replication in presence of a dynamic workload. We first examine workload variations in time. In a first phase, from time 0 to time 5000 s, we set the content request rate as $\lambda = 0.01$ req/s. In a second phase, from 5000 s to the end of the simulation, the request rate doubles, i.e., $\lambda = 0.02$ req/s.

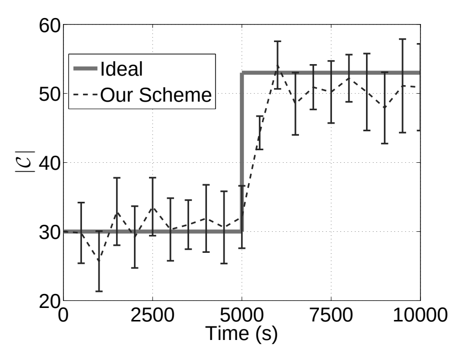

Fig. 3. Temporal evolution of the number of replica nodes in case of variations in time of the content demand, for a mobile network. $|\mathcal{C}^*|$ is equal to 30 and 53 in the first and second phase, respectively.

Fig. 3 shows the temporal evolution of the number of replicas in a mobile network. The figure is enriched with two reference values: in the first phase $|\mathcal{C}^*| = 30$, in the second phase $|\mathcal{C}^*| = 53$. Our mechanism achieves a very good approximation of the target number of replicas: despite node mobility, not only is our scheme able to correctly determine the number of replicas but also their target location. As a consequence, the load distribution is minimally affected by a variation in time of content demand. This result is reported in Tab. III, where we indicate the 25%, 50% and 75% quantiles of the workload, and the average load per replica node.

TABLE III
WORKLOAD DISTRIBUTION OF REPLICA NODES FOR VARIATIONS IN TIME OF THE CONTENT DEMAND, IN A MOBILE NETWORK.

Percentile	25th	50th	75th	Mean
1st Phase	4	8	13	9.98
2nd Phase	5	8	13	9.98

We now turn our attention to variations in space of content demand: we describe the behavior of the content replication mechanism with the following example. For the initial 5000 s of the simulation time, content queries are issued by all nodes deployed on the network area $\mathcal{A}$ of size 1 km^2. Subsequently, we select a smaller square area α of size 500 m^2 in the bottom left corner of $\mathcal{A}$ and instruct only nodes within that zone to issue content queries, while all other nodes exhibit a lack of interest.

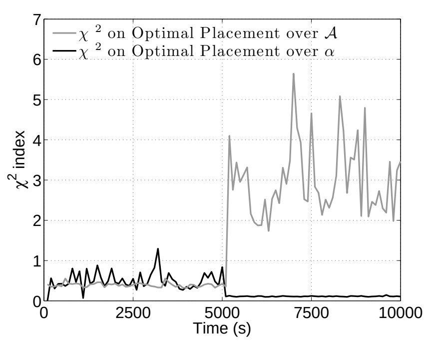

Fig. 4. Temporal evolution of the χ^2 index for variation in space of the content demand, in a mobile network.

Fig. 4 compares the empirical and the approximate optimal distributions using the temporal evolution of the χ^2 index. We observe a very good match (i.e., low values χ^2) over the network area $\mathcal{A}$ and on the sub-area α when content demand comes, respectively, from $\mathcal{A}$ for $t \leq 5000$ s and α for $t > 5000$ s. This suggests that when content demand varies in space, our scheme allows content replicas to migrate to the location where the demand is higher and meet a variation in the workload.

VI. CONCLUSIONS

We focused on content replication in mobile networks where users can access content through device-to-device communications, and we addressed the joint optimization problem of (i) establishing the number of content replicas to deploy in the network and (ii) finding their most suitable location. To achieve these goals, we proposed a distributed, lightweight scheme that approximates with high accuracy the solution obtained through centralized algorithms. We studied the flexibility of our scheme when content demand varies in time and in space: our experiments underlined the ability of our approach to adapt to such variations while maintaining accuracy in approximating an optimal solution.

Our next step will be to study the behavior of our scheme considering replication and placement problems of multiple information items, content popularity, and content sizes derived from actual traffic traces. We will explore alternative definitions of the cost of replication that will account for content access congestion in addition to the workload experienced by a replica node. Lastly, we will relax the assumption of a cooperative setting and analyze selfish replication with tools akin to game theory. By modelling the system as an anti-coordination game [19], we will extend the ideas presented in this work to achieve strategy-proofness.

REFERENCES

[1] www.nytimes.com/2009/09/03/technology/companies/03att.html?_r=1
[2] venturebeat.com/2009/05/11/iphone-users-eating-up-atts-network/
[3] A. Derhab and N. Badache, “Data Replication Protocols for Mobile Ad-hoc Networks: A Survey and Taxonomy,” *IEEE Communications Surveys & Tutorials*, vol. 11, no. 2, pp. 33–51, 2009.
[4] P. Mirchandani and R. Francis, *Discrete Location Theory, John Wiley and Sons*, 1990.
[5] C. Casetti, C.-F. Chiasserini, M. Fiore, C.-A. La, and P. Michiardi, “P2P Cache-and-Forward Mechanisms for Mobile Ad Hoc Networks,” *IEEE ISCC*, 2009.
[6] V. Arya, *et al.*, “Local Search Heuristic for k-median and Facility Location Problems,” *ACM STOC*, 2001.
[7] N. Laoutaris, G. Smaragdakis, K. Oikonomou, I. Stavrakakis, and A. Bestavros, “Distributed Placement of Service Facilities in Large-Scale Networks,” *IEEE Infocom*, 2007.
[8] T. Moscibroda and R. Wattenhofer, “Facility Location: Distributed Approximation,” *ACM PODC*, 2005.
[9] H. Hayashi, T. Hara, and S. Nishio, “On Updated Data Dissemination Exploiting an Epidemic Model in Ad Hoc Networks,” *BioADIT*, 2006.
[10] M. Hauspie, A. Panier, and D. Simplot-Ryl, “Localized Probabilistic and Dominating Set Based Algorithm for Efficient Information Dissemination in Ad Hoc Networks,” *IEEE MASS*, 2004.
[11] J. Luo, J.-P. Hubaux, and P. T. Eugster, “PAN: Providing Reliable Storage in Mobile Ad Hoc Networks with Probabilistic Quorum Systems,” *ACM MobiHoc*, 2003.
[12] H. Yu, H. Hassanein, and P. Martin, “Cluster-based Replication for Large-scale Mobile Ad-hoc Networks,” *International Conf. on Wireless Networks, Communications and Mobile Computing*, 2005.
[13] T. Hara, “Effective Replica Allocation in Ad Hoc Networks for Improving Data Accessibility,” *IEEE Infocom*, 2001.
[14] V. Thanedar, K. C. Almeroth, and E. M. Belding-Royer, “A Lightweight Content Replication Scheme for Mobile Ad Hoc Environments,” *Network*, 2004.
[15] M. Shinohara, H. Hayashi, T. Hara, and S. Nishio, “Replica Allocation Considering Power Consumption in Mobile Ad Hoc Networks,” *IEEE PERCOMW*, 2006.
[16] O. Kariv and S. Hakimi, “An Algorithmic Approach to Network Location Problems, Part II: p-medians,” *SIAM Journal on Applied Mathematics*, vol. 37, pp. 539–560, 1979.
[17] C.-A. La, P. Michiardi, C. Casetti, C.-F. Chiasserini, and M. Fiore, “A Lightweight Distributed Solution to Content Replication in Mobile Networks,” arxiv.org/abs/0909.2024, Sept. 2009.
[18] *NIST/SEMATECH e-Handbook of Statistical Methods*/, http://www.itl.nist.gov/div898/handbook/, 2009.
[19] P. Michiardi, C.-F. Chiasserini, C. Casetti, C.-A. La, and M. Fiore, “Brief Announcement : On a Selfish Caching Game,” *ACM PODC, 2009.*