



HAL
open science

MobiTrade: Trading Content in Disruption Tolerant Networks

Amir Krifa, Chadi Barakat, Thrasyvoulos Spyropoulos

► **To cite this version:**

Amir Krifa, Chadi Barakat, Thrasyvoulos Spyropoulos. MobiTrade: Trading Content in Disruption Tolerant Networks. [Technical Report] 2011, pp.9. inria-00568235

HAL Id: inria-00568235

<https://inria.hal.science/inria-00568235>

Submitted on 23 Feb 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

MobiTrade: Interest driven content dissemination architecture for Disruption Tolerant Networks

Amir Krifa*, Chadi Barakat*, Thrasyvoulos Spyropoulos[†]

*Project-Team Planète, INRIA Sophia-Antipolis, France

[†]EURECOM, Sophia-Antipolis, France

Emails: {Amir.Krifa, Chadi.Barakat}@inria.fr, Thrasyvoulos.Spyropoulos@eurecom.fr

Abstract—The rapid proliferation of advanced mobile devices has created a growing demand for data content. Existing approaches (e.g. relying on cellular infrastructures) cannot keep up with the large volume of content generated and requested, without the deployment of new expensive infrastructure. Exchanging content of interest *opportunistically*, when two nodes are in range, presents a low cost and high bandwidth alternative for popular, bulky content. Yet, efficiently collecting, storing, and sharing the content while preventing selfish users from impairing collaborative ones, poses major challenges.

In this paper, we present **MobiTrade**, a collaborative content dissemination system on top of a delay tolerant network. It allows users to head out in the real world, express locally their interests, and wait to get notified whenever an encountered device has content(s) matching these interests. Even though interactions are done between neighboring wireless devices (*locally*), **MobiTrade** implements a *trading* scheme that motivates mobile devices to act as *merchants* and carry content across the network to satisfy each other's interests. Users continuously *profile* the type of content requested and the collaboration level of encountered devices. Based on this knowledge, an appropriate utility function is used to rank these requests and collect an optimal inventory of data that maximizes the expected *value* of stored content for future encounters. Using NS3 simulations based on synthetic and real mobility traces, we show that **MobiTrade** achieves *up to 2×* higher query success rates compared to other content dissemination schemes. Furthermore, we show that **MobiTrade** successfully isolates selfish, non-collaborative devices. Finally, using a simple game theoretic framework we show that turning on our **MobiTrade** mechanism is an efficient Nash Equilibrium.

I. INTRODUCTION

Mobile networking is quickly reaching a tipping point. While data has been a second-class customer for cellular networks until recently, the wide spread of smart phones, and the access these provide to existing (e.g. social networking) and novel applications, are generating unprecedented amounts of mobile data. The capacity of current cellular infrastructures (e.g. GPRS and 3G) has already been pushed to the limit by even a small number of eager data plan users [1]. To support the increasing number of devices generating data at high rates, ISPs will inevitably be pushed towards either lowering bandwidth quotas [1], adopting non flat rate plans, or deploying (expensive) next generation equipment (e.g. LTE). This has lead many researchers (and industry) to explore alternative or hybrid architectural solutions [12], [5].

To this end, direct mobile-to-mobile communication can be leveraged to harvest the large amounts of unused bandwidth between wireless devices in proximity. Mobile devices with multiple wireless interfaces (e.g. Bluetooth and WiFi) allow two users in range to exchange data at much higher speeds, lower power consumption per bit, and essentially no (direct) monetary cost [25]. At the same time, increasing user demand for content is creating a shift in focus towards content and

data centric systems [16], in both wired and wireless Internet. According to statistics published by ComScore [2], content dissemination through social networking ranks as the fastest-growing mobile content category. In such a new environment, the primary focus will be on content rather than on devices. This raises an opportunity for a content dissemination overlay over the large numbers of mobile devices *in the wild*, meeting with each other in passing.

Nevertheless, node mobility and the much shorter range of high speed interfaces makes *contacts* between devices inherently *intermittent* and *time-limited*. As the topology is very unstable, content providers and content consumers might be completely unaware of each other and never connected at the same time to the same part of the network. Therefore, content should be moved towards users in a *store-carry-and-forward* manner. While a large number of store-carry-and-forward routing schemes has been recently proposed in the context of *Opportunistic* and *Disruption Tolerant Networking (DTN)*, (see e.g. [28], [24] for a survey), the vast majority of these schemes are only end-to-end routing solutions.

To efficiently handle content dissemination, a new architecture is needed, where people express their interests, head out in the real world and wait to get notified whenever a content that matches their interests is carried by any of the encountered devices. To achieve this, a candidate architecture should not only take care of the network and device resources, but also carefully consider: (i) the propagation of interests of participating users, (ii) the matching of these interests to individual node mobility patterns, and (iii) the willingness of involved users to collaborate. This latter point can be a major deal-breaker in any envisioned architecture (as is the case for example in traditional MANETs [10]). Indeed, mobile devices are controlled by rational people and we should expect them to behave selfishly by attempting to maximize their revenues and conserve their resources (content storage and energy), unless cooperation is somehow incentivized and free-riders penalized.

In this paper, we present **MobiTrade**, an interest driven content dissemination architecture for opportunistic networks. **MobiTrade** has no notion of content source or content destination at the device level, but mainly deals with labeled content and content *channels* (a channel is a generic set of labels describing a type of content [22]). Practically, to download content, a user only needs to load the **MobiTrade** software into his device, express his interests in joining a set of channels (locally on his mobile), publish any content he wants to share, and then turn the application into the background while heading out into the real world. As he walks around, the user device meets other devices and exchanges autonomously with them content matching their specified interests. Unlike many related works [7] [22] [14] [23], **MobiTrade** supposes that

some users might have a selfish behavior. To cope with this, it proposes a *trading* scheme to enforce collaboration among mobile devices, to block selfish behaviors and to convey the right content towards the right consumer in a *store-carry-and-forward* mode, without having the source and interested devices explicitly communicating.

There are two main components in this trading algorithm. First, Tit-For-Tat is directly employed during individual device meetings. If Alice meets Bob and needs X units of content from him, she must give him X units back (one by one). This is a strict policy that isolates selfish nodes trying to receive content without giving anything back. It also motivates nodes to carry content for "foreign" channels (not interested in personally), that they can use as *trading currency*. Second, MobiTrade gives the ability to associate a utility per channel and to identify the most lucrative ones for each device. From a trading point of view, this utility can be seen as the expected reward of storing and carrying a piece of content, by "selling" it to devices encountered in the future. In other words, MobiTrade turns any participating device into a *merchant* with the sole goal of *collecting in its buffer an inventory of data that maximizes its future trading capacity and its resulting revenue (content of interest received)*.

The rest of this paper is organized as follow. The following section presents the related work. Section III describes the MobiTrade architecture main building blocks. In Section IV, we detail the approach we follow to infer the utility of a given channel. Then, we provide a detailed simulation analysis in Section V based on both synthetic [6] and real mobility traces [3] and we compare MobiTrade to different content dissemination policies. Finally, we summarize our conclusions and discuss future work in Section VI.

II. RELATED WORK

A significant share of research on opportunistic networks has focused on unicast (see e.g. [28] or [24]) and multicast routing [29], [11]. In this paper, we consider the problem of content dissemination, where content publishers and consumers are not aware of each other. Content dissemination systems have been proposed for the Internet, and also for conventional MANETs [19] [17]. In general, these systems assume that network paths are rather stable, and often generate a significant amount of traffic to maintain knowledge of other devices' caches. Therefore, they are not suitable for opportunistic networks.

MobiTrade shares similarities with pub/sub systems for opportunistic networks. TACO-DTN [26] is a publish/subscribe system mainly designed to distribute temporal events, whereas our approach is implemented as a pure receiver driven system. Peoplenet [23] is hybrid system propagating and matching queries over, first, infrastructure, and, second, using DTN device-to-device communication in the wireless "last hop" (e.g. inside a cell) to forward further. Finally, compared to SocialCast [14], MobiTrade uses a considerably more sophisticated utility that considers both content demand/popularity and the collaboration level of any user it encounters.

To our best knowledge, the only other work looking at pure content dissemination for opportunistic networks is the research thread first initiated by the PodNet project [20], [22]. This work proposes a DTN Podcasting architecture, built around the concept of *content channels*, that we also use in

this paper. In the first version of PodNet [22], users only store and share channels they are interested in. In a later version [20], simple strategies to cache other channels as well are considered, in order to improve the overall performance. ContentPlace [7] by Boldrini et al. attempts to improve Podcasting using explicit knowledge of social networking links of participants. Compared to it, MobiTrade does not require such user reported social information and does not make any hard assumptions regarding node mobility. Finally, the most recent work in this thread, by Hu et al [15], attempts a rigorous formulation of the problem of optimally matching channels (to store) to a population of devices. A distributed algorithm is then proposed based on the framework of Markov Chain Monte Carlo optimization. While we find this framework particularly interesting, it also comes at the expense of high complexity, long convergence delays (known in MCMC), and a need for carefully tuned simulated annealing [8].

A major difference of MobiTrade, is that we consider users to be inherently selfish, instead of inherently collaborative as in all the aforementioned studies. Experience teaches us that selfish behavior is often the norm, unless incentives are provided, and can be a major impediment to any such peer-to-peer system in the wild [10]. The only proposal we are aware of, dealing with selfish users in the context of DTNs is [9], where a Tit-For-Tat mechanism ("bartering") is also used between nodes to exchange content. While Tit-For-Tat (TFT) ensures selfish users are blocked, it does not answer itself *how collaborative nodes should optimally (re-)act in the presence of TFT*. This is answered in MobiTrade by a personalized inventory management mechanism, key to almost all the system's functions and good performance.

As a final note, MobiTrade is not a reputation system, as e.g. [27]. In reputation systems, nodes collect and share their opinions about peers with others. In our case, each node forms a personal opinion of peers used to only optimize her actions. Our system is more similar to a *market* of independent traders. As a result, a bad customer for node X might be a good customer for node Y.

III. MOBITRADE ARCHITECTURE

A. MobiTrade Data Records

In a content dissemination architecture, nodes need first to somehow express their interests for different (types of) content and advertise these interests. To this end, we borrow the concept of *channels*, introduced in [22]. Specifically, the MobiTrade architecture relies on two data records: content and channel records (Fig. 1).

Channel Record: A user asks for a set of contents by creating locally a channel record that encapsulates the set of keywords the user thinks they better describe the contents she is looking for. Channels can be added or deleted by the user, at any time. In contrast to the CCN model [16], the channel record in MobiTrade is *one-for-many* (not one-for-one). A desirable content is identified based on a match between the channel keywords and the content description (also characterized by a set of keywords). We think that such a match process is more appropriate for the communication model we are proposing, where the user might not know the exact content he is looking for, or might anyway be interested

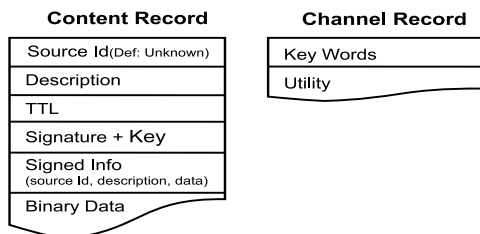


Fig. 1. MobiTrade data records and channel storage

in all contents matching the description (e.g. Madonna songs, photos of Nice, sports tickets for sale).

A lot more can be said about this channel structure (e.g. hierarchies, merging and splitting of channels, semantic content matching, etc.), but this is beyond the scope of this paper. Instead, we choose to use a simple channel structure here and focus on the algorithmic part of the system. Finally, each channel record contains a *utility* entry. This is a key quantity for MobiTrade, allowing our system to optimize various important functions (scheduling, inventory management, collaboration profiling, etc.). For now, we will assume this as given, but Section IV is devoted to how this utility is derived.

Content Record: In addition to the content description, a content record contains a number of additional fields. First, we associate to each MobiTrade content a *TTL* (Time to Live). By the end of this time, records can be removed. The cases requiring this *TTL* field are many, for example someone could be interested in selling something today. To ensure devices respect this *TTL* field, MobiTrade devices do not reward each other for expired content.

Furthermore, to provide users with a full control over their privacy (i.e., the contents they publish and their center of interests), we choose to keep anonymous any generated channel record (no user or device ID are stored). Nevertheless, for contents, one still has the possibility to associate to them a *canonical* source name that refers in some way to the content publisher. This field does not correspond necessarily to the user device address, but it can hold the publisher postal address or its phone number, which can be used for distinguishing between contents, feedback mechanisms on separate communication mediums or authentication purposes.

Finally, to provide for some MobiTrade security, we choose to use a CCN like *content-based security* model [16]. With this model, protection and trust travel with the content itself, rather than being a property of the connections over which it is transmitted. MobiTrade devices authenticate via the signature field (Fig. 1), the binding between the content source ID, its description and some parts of its data. In addition to this signature, each signed MobiTrade content carries with it the public key necessary for its verification by other devices. The signature algorithm can be selected to meet the performance requirements of that particular published data.

B. MobiTrade Protocol

Communication within the MobiTrade architecture is driven by the consumers of contents. A user asks for contents by “joining” a channel to which these might belong, storing the respective record(s) locally on the device until it becomes out of date. Then, each time a new meeting opportunity arises with another mobile device, both devices initiate the MobiTrade communication protocol that has two main functions:

- (*Fair Content Exchange*) to help nodes identify content

of interest in their peer’s inventory (buffer) and provide a set of rules to exchange such content in a *fair* manner.

- (*Inventory Management*) to allow nodes to profile each exchange (*learning*) and use this knowledge to improve the outcome of future interactions (*prediction*).

The MobiTrade protocol is summarized in Fig. 2. Each device starts by sending its list of channels to the other device. Based on it, each device decides on the set of contents to forward, i.e. content in its buffer matching one of the channels requested by its peer. These contents are scheduled for transmission and passed to the *Tit-For-Tat (TFT)* trading algorithm (Section III-D) that ensures an equitable exchange.

In addition to channels the node is personally interested in, if extra space is available, it might choose to join “foreign” channels. Content for these channels is not stored for personal consumption, but can be used as exchange currency during the TFT phase, in order to acquire additional content of interest. This provides the incentive to nodes to act as *merchants*, collecting and carrying content to be used only for trading. This also allows content to *propagate efficiently across the network and between remote producer-consumer pairs, without any explicit routing mechanism*.

After content starts being exchanged (one-by-one), a node receiving a content might need to perform some inventory management. First, if it already has this piece of content (or the content is expired), it will drop it¹. If the content is new, and there is available buffer space for the channel this belongs to, the content is stored in the buffer. The exchange finishes once the transfer of the requested contents ends or the two devices get out of the range of each other. At this point, both devices update the set of channels they are keeping track of as well as their corresponding utilities based on the (profiled) outcome of the session (as explained in Section IV).

C. Proportional Storage and Bandwidth Allocation

In a context with many nodes, various channels, and lots of multimedia content, node buffers will be operated most of the time at capacity and contact duration between nodes might not suffice to exchange all intended content. This highlights the need for efficient resource allocation algorithms. Unlike related work [7] [22], *MobiTrade allocates both buffer space and contact bandwidth in an equitable way among the different channels*.

First, MobiTrade implements a mechanism of *proportional soft quotas* to share available buffer space among channels. Let a node carry N channels with channel utilities $U(i), i \in \{1, N\}$, and have a total buffer capacity of B . Then, the *quota* $B(i)$ of the buffer space channel i is entitled to is

$$B(i) = \frac{U(i)}{\sum_i U(i)} B.$$

The proportion of storage allocated to a channel is proportional to its utility. Quotas are updated whenever one or more of the channel utilities change or channels are added/removed.

¹This is possible, since nodes only send to each other lists of channels and not a detailed list of contents. This is a coding trade-off that tries to avoid large amounts of meta-data being exchanged before any actual content is sent. On the other hand, it might also lead to some wasted bandwidth if lots of duplicate content is transmitted. A possible middle-ground solution to this problem could be the use of Bloom filters.

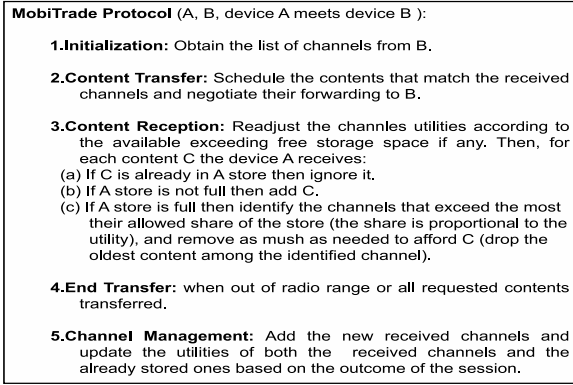


Fig. 2. MobiTrade protocol

Based on these quotas and the amount of storage channel i is *currently* occupying, let $S(i)$, a node receiving a content (of W bytes) for channel i will perform the following actions:

- if $S(i) + W < B(i)$, then store the content.
- if $S(i) + W > B(i)$ and $\sum_i S(i) + W < B$, then store content.
- if $S(i) + W > B(i)$ and $\sum_i S(i) + W > B$, then pick the channel j maximizing $\max_j (S(j) - B(j))$ and drop the oldest content for this channel.

Points (2) and (3) above imply that the quotas $B(i)$ are soft. Channels can exceed their share and take over free space, if any is available. However, as soon as the buffer is full, the policy pushes shares back to their just proportion.

Finally, in the presence of limited contact durations, a device cannot simply forward contents by decreasing order of the utilities of their channels since a channel can match more than one content. For example, a MobiTrade device can face a situation where many contents match a popular channel, and hence it keeps forwarding only those contents at the expense of other channels which are less popular. To remedy this, MobiTrade applies the *Weighted Fair Queuing* policy which prevents starvation of channels and ensures that contents are forwarded proportionally to the utility value of the channel they match.

As a final note, when the scheduling policy decides to forward a set of contents from the same channel, MobiTrade sends the *youngest* contents first. This decision is motivated by our findings in [18] and is complementary to the drop oldest policy applied among the contents of the same channel in case of congestion. More sophisticated policies could be also applied [18], but this is beyond the scope of this paper.

D. Tit-For-Tat Trading

One major difference of our system compared to other content dissemination solutions for DTNs (e.g. [22], [7], [26]) is that we assume that participating users are selfish by default. Thus they might act as free-riders: during a meeting they receive content they want but don't give something back (even if they have), in order to save transmission power or to save some bandwidth. Or, they might not collect content their peers are requesting. Experience teaches us that even in the absence of power or storage concerns (e.g. wired users of BitTorrent), additional concerns (legal, malice), might motivate users to circumvent collaborative mechanisms [21]. We believe these are *powerful incentives that could sabotage any collaborative content dissemination system, if not properly handled*.

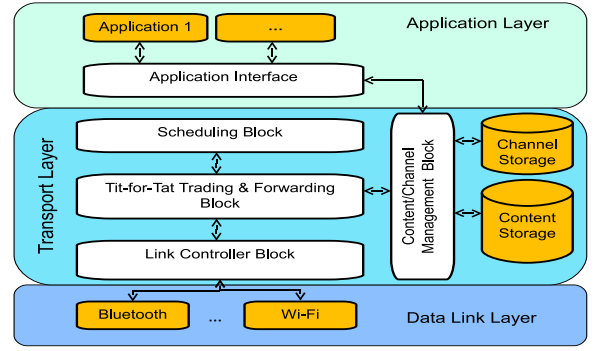


Fig. 3. MobiTrade building blocks

In order to remedy this and isolate selfish users, a *strict* Tit-For-Tat (TFT) trading is enforced during meetings. Content scheduled for exchange is forwarded one-by-one (or in equally sized blocks/pieces, if contents are not of equal size), i.e. node A sends some, then node B sends some, then node A again, etc. Forwarding of content stops when the other device cannot (or chooses not to) reciprocate the amount of bytes it received².

In addition to this TFT mechanism ensuring that selfish nodes are not serviced when content for them is available, a utility maintenance mechanism (to be described in Section IV) further ensures that well-intended nodes will not waste their resources collecting content for selfish nodes from which they will not receive any reward.

E. MobiTrade Device Model

Having described the various mechanisms of our system, we give here some more details of the MobiTrade architecture. Fig. 3 is a schematic of the core MobiTrade device, which has five building blocks. We are currently implementing this architecture on Android phones in order to conduct small scale experiments.

The *Application Interface* is designed so that developers can easily extend MobiTrade and create new content sharing applications that use the MobiTrade functionalities. Each MobiTrade application has a unique ID that it registers at the application interface. The application then simply communicates with the MobiTrade system using event handlers.

The *Scheduling Block* defines the order to follow while forwarding a set of requested contents within a limited contact opportunity. As explained before, this ordering is done based on the utility values.

The *Tit-For-Tat Trading and Forwarding Block* takes in charge (i) the forwarding of the stored channels records whenever a new contact is established, and (ii) the negotiation and forwarding of the requested contents.

The *Content/Channel Management Block* manages the channel and content storage space. It provides an API for storing and retrieving channel and content records that hides the storage technology specifics. This block also takes in charge utility updates for the stored channels and buffer management (as described in Section III-C).

The *Link Controller Block* is the lowest layer in the MobiTrade architecture. It provides a common interface for sending and receiving data across the different available wireless

²In order to solve the bootstrapping issue, when two well-intended devices meet for the first time, MobiTrade enables generous cooperation up to a certain amount as described in Section IV.

TABLE I
NOTATION

Variable	Description
$U_{CH}(n)$	In bytes. Models the utility of the channel CH at the n_{th} meeting.
$X(CH)$	0 or 1. Expresses whether the met device is also interested in channel CH or not.
$CL(CH)$	In bytes. Expresses the collaboration level of the met device during a given meeting with respect to the channel CH .
$SC(CH)$	In bytes. Expresses the number of bytes sent to the met device during a given meeting with respect to the channel CH .
ω	Between 0 and 1. A weight that decides on the elapsed time window over which we average the utility of channels.
α	In bytes. Expresses MobiTrade device generosity level, used to de-block the situation when two devices meet for the first time and request new channels.
β	In bytes. Controls the speculation that a MobiTrade device makes regarding the expected future reward from a given channel.

interfaces. It is also responsible for periodically scanning the neighborhood for devices, for establishing the contacts whenever meeting opportunities arise and for triggering the Tit-For-Tat trading and forwarding block. The link controller provides an API that hides network technology specifics (Bluetooth and Wi-Fi) from the rest of the upstream blocks.

IV. INFERENCE OF CHANNEL UTILITY

MobiTrade aims at maximizing the number of collected contents while providing incentives for devices to collaborate. To have their requests satisfied, MobiTrade devices have to propose contents to other devices in counterpart of what they are looking for. As said before, each device is equipped with a storage space that is filled with contents from the different existing channels, which are used later as trading currency. This storage space is filled so as to satisfy the future demand: each channel occupies a proportion of the storage equal to the reward it is expected to bring upon future meetings. Due to Tit-For-Tat trading, the reward from carrying a channel is the amount of data from this channel a device will sell upon future meetings (to get both data for its own usage and data used for later trading). Hence, *to optimize performance, MobiTrade devices should be able to quantify the expected reward from the channels they carry.*

In MobiTrade, the expected reward from each channel is modeled through a utility function used for ranking contents upon a meeting and for dropping them upon saturation of the storage. In this section, we detail how these utilities are calculated. Table I summarizes some useful notation.

For each channel CH , MobiTrade defines its utility $U_{CH}(n)$ at the n^{th} meeting (counted over all devices) in a way to reflect the *expected* number of bytes the device will sell from this channel (and thus the bytes of interest these will buy back) with any random device it will meet³. This is clearly a complex interplay of mobility patterns, available channels and content, and node interests. *We choose to keep our framework as assumption free as possible about mobility and interest patterns*, and take the following approach.

³Note that MobiTrade does not differentiate between own and foreign channels (these could also overlap) at the level of calculating utilities. However, foreign content doesn't interfere with content for own consumption. If a peer has content of interest for own consumption, this will always be retrieved, provided it can be "bought", and passed to the application. MobiTrade only decides whether this should be further cached in the content storage for future trading.

Calculation and update of channel utilities: Assuming stationarity of the network over at least a time window of $2T$, the expected reward from carrying a channel CH can be calculated by averaging all experienced rewards over all meetings in the past time window T . Meetings that do not request channel CH count as zero. To facilitate implementation, an Exponential Weighted Moving Average (low pass) filter is used for averaging.

Let's consider a device A and let's suppose that an $(n + 1)^{th}$ meeting opportunity arises. After the establishment of the physical connection, both devices exchange matching contents while applying the Tit-For-Tat trading algorithm (as in Fig. 2). Once done, they both record the volume of exchanged data and update each the utilities of the channels they carry. For device A and channel CH , this update is done as follows:

$$U_{CH}(n + 1) = \omega.U_{CH}(n) + (1 - \omega).X(CH).CL(CH),$$

where $\omega = (t_n - t_{n-1})/T$ is the weight associated to the low pass filter⁴. Concerning the term on the right hand side, it models the amount of data exchanged with the met device B from channel CH . $X(CH)$ is a binary variable that expresses whether B is interested or not in CH . This variable captures the popularity of a given channel over all MobiTrade devices met by A . As for $CL(CH)$, it captures the volume of contents that could be sold to device B in the future (i.e. a prediction of B 's future demand for CH).

This calculation leads to several nice properties. First, this utility calculation is per channel and does not account for the physical addresses of encountered devices. The same device met several times or different devices met the same number of times count the same from the viewpoint of MobiTrade, as long as the amount of data sold was the same. This avoids tracking individual devices which improves the scalability of MobiTrade. Second, it does not try to over-optimize for the next meeting only (as could be the case with detailed mobility prediction) but rather optimizes the inventory over a larger time horizon, enough to absorb prediction inaccuracies at individual meetings. Finally, coupled with the Tit-For-Tat algorithm, our utility (through $CL(CH)$) accounts for the collaboration of devices in addition to the popularity of channels. In lack of this feature, a channel widely requested among a group of users, who do not collaborate by bringing back interesting contents for A , could force A to keep collecting contents matching CH , only to discover later that this content buys him nothing. The storage of A could have been better used by carrying contents for less popular channels but more collaborative devices.

Collaboration and Bootstrapping: The $CL(CH)$ term above expresses the collaboration level of the device B with respect to the channel CH . If a channel however is requested for the first time at the $(n + 1)^{th}$ meeting, its $U_{CH}(n)$ would be initialized to zero. A new node that asks for a channel CH , would see its request being ignored, as no content for CH was exchanged in this round. Clearly, an appropriate bootstrapping mechanism is needed, in order to avoid this chicken and egg problem for new nodes or channels. This can be implemented as some *slack* or *generosity* in the $CL(CH)$ calculation and the TFT mechanism. At the same time, this generosity should be such that it *cannot be exploited* by selfish nodes. The

⁴By making it function of the elapsed time between the current meeting and the previous one, one can ensure an averaging over a time window T .

calculation of $CL(CH)$ below is inspired from TCP slow start, and attempts to best satisfy the above two (conflicting) goals:

$$CL(CH) = \begin{cases} \text{Max}(\alpha, 2 \cdot SC(CH)) & \text{if } SC(CH) < \beta, \\ SC(CH) + \alpha & \text{otherwise.} \end{cases}$$

The collaboration level $CL(CH)$, that is, the prediction of future demand, is thus a function of the actual (last exchange) demand $SC(CH)$. If $SC(CH)$ is less than some threshold β , we allow $SC(CH)$ to double, to accelerate the collaboration process at its beginning; otherwise, devices will have to meet more often to reach a satisfactory collaboration level. After β^5 , the generosity of device A switches into a linear mode when it believes it has successfully approximated the steady-state demand, and only speculates an additional α to $SC(CH)$.

The same factor α is equally used as minimal value for $CL(CH)$ to unblock the situation when device B asks for a new channel ($SC(CH)$ equals zero in this latter case). If a channel does not bring the expected reward for any reason (lack of collaboration, oldness of the contents carried, etc.), this will be reflected by a decrease in $SC(CH)$, which automatically leads to a decrease in the utility value we associate to this channel. Similarly, α also serves to keep selfish nodes in control. If some selfish device asks for a long list of new channels, MobiTrade will associate initially a small utility value to them. A selfish/malicious user is then obliged to collaborate in order to increase the utilities of his channels and thus the portion of content storage these are given.

From the perspective of a collaborative trader node, a community of non collaborative users is equivalent to a community of users not requesting channels. We believe this improves the robustness of the system and allows it to scale to large networks, without the need for explicit blacklisting and reputation systems (at least for selfish nodes).

V. PERFORMANCE EVALUATION

We move now to performance evaluation of our system. We first describe our experimental setup, and then present simulations results for two main types of scenarios: collaborative scenarios and scenarios including selfish users.

A. Experimental Setup

Protocols: We have implemented the MobiTrade content dissemination protocol in the NS3 simulator [4]. Throughout our simulations we will be considering two versions of MobiTrade, with (**MobiTrade + TFT**) and without Tit-For-Tat (**MobiTrade - TFT**). Note that this corresponds only to the forwarding module, described in Section III-D. The channel utility maintenance, described in Section IV, is kept on in all scenarios. We have also implemented two different versions of the *PodNet* scheme as a baseline for comparison⁶, as described, to our best understanding in [22] and [20]: (i) non-collaborative Podcasting, where users just carry and share their own channels [22] (**Podcasting**); (ii) collaborative

⁵We take an optimistic approach and choose it equal to the maximum utility value over all channels of A .

⁶We choose the PodNet framework, as it is the most directly comparable to our scheme, and consists of simple enough mechanisms that we consider practical for implementation. For example, [15] is considerably more complex and based on an MCMC framework that requires careful simulated annealing and might take a long time to converge. Furthermore, [7] requires explicit knowledge of social network links, not available in our framework.

Podcasting with the *Uniform* channel sharing strategy, where, a device records all channels it has seen in the past and solicits contents for these channels randomly [20] (**Podcasting + Uniform**). This latter strategy was shown to perform best in [20], compared to other heuristics taking into account channel popularity. As a final note, we stress that MobiTrade's first goal is not to compete with optimal collaborative schemes, but rather to efficiently deal with selfish nodes, without compromising the socially optimal (collaborative) performance.

Mobility Models: To evaluate the different protocols, we use two type of mobility scenarios, a state-of-the-art synthetic mobility model (HCMM) [6] and a real mobility trace (KAIST) [3]. HCMM is a mobility model inspired by Watts' Caveman model that was shown to reproduce statistics of human mobility, such as inter-contact times and contact duration. In HCMM, the Caveman model is used to define a graph (overlay) with nodes divided into (well connected) groups and each group is assigned to a physical home location. Also, some users belonging to different groups can have links to each other (bridges). These (intra- and inter-group) links are used in HCMM to drive movements: each user moves towards a given group's home location with a probability proportional to the weight of its links towards the group. In our scenario, we consider 50 users distributed into 5 groups. The plane is divided into a 10*10 grid of cells (5000 m wide), and each cell can serve as a home location for a group.

The KAIST scenario consists of real human mobility traces collected from a university campus (KAIST) in South Korea [3]. We consider a sample of the KAIST campus traces taken from 50 students, where the GPS receivers log their position at every 30 seconds. We integrated both mobility models in NS3. Both case studies consist of simulations that last 24 hours where devices use the 802.11b protocol to communicate with a transmission range around 60 meters.

Traffic Model: Unless otherwise stated, each user joins randomly 2 channels at the beginning of the simulation. For simplicity, we assume that all generated contents have the same size⁷. However, different channels do not need to have the same size (the size of a channel is equal to the sum of its contents' sizes). Some channels might have lots of content, and others less. Finally, we consider that each user generates contents periodically that match one of the channels that were requested by users from other groups⁸.

B. Collaborative scenarios

We first evaluate MobiTrade, assuming all nodes are collaborative, using the following four scenarios (described in Table II) (there are 50 channels in total): **SC₁** implements a *homogeneous traffic* pattern, i.e. each channel has the same size and each user joins the same number of channels. In **SC₂**, users choose a *random number of channels* to join, but channels still have the same size. In **SC₃**, users ask for the same number of channels but these have *random sizes*. Finally, **SC₄** introduces some *churn*, where 10 of the users join the simulation after 8 hours, while existing sessions are ongoing, and leave again 8 hours later. Due to space limitations, for all scenarios, we show plots for the HCMM

⁷Variable sized content could still be split to equal sized pieces or blocks. We defer the study of more complex content structures to future work.

⁸The content generation interval depends on the number of contents for a channel and the duration of the simulation.

mobility model, accompanied with respective results for the KAIST trace summarized in Tables.

TABLE II
COLLABORATIVE SIMULATION SCENARIOS

Sim. Scenario:	SC ₁	SC ₂	SC ₃	SC ₄
Nbr. of Users:	50	50	50	40 + 10 transient
Requested CH(s) per User	2	random [1, 20]	2	2
Size of CH(s) (# of contents)	20	20	Random [1, 20]	20

Effect of TTL: Before we proceed, we take a quick look first into the impact of content TTL. As explained in Section III, our buffer drop and scheduling policies give priority to younger messages⁹. This is not only in accordance with our daily experience (for many types of content, e.g. news feeds, we prefer to have the most recent version), but has also been shown to be an efficient resource allocation policy in the context of a single channel [18] (intuitively, older content has a higher chance to have been delivered already). Fig. 4 depicts the MobiTrade average delivery rate as a function of the content *TTL* (for scenario SC₁), with and without prioritizing younger messages (per channel). It is evident that the higher the (application chosen) *TTL* the more old content “hogs” node buffer and contact bandwidth, not allowing new content to reach its audience. When prioritizing younger messages, not only does performance stabilize, but with an infinite TTL the gain from just this mechanism is up to 76%.

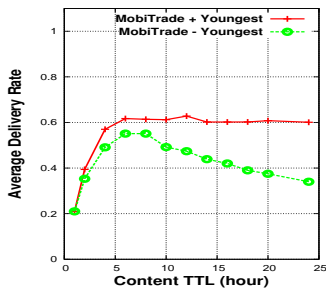


Fig. 4. Drop and Scheduling policy inside the same channel (SC₁).

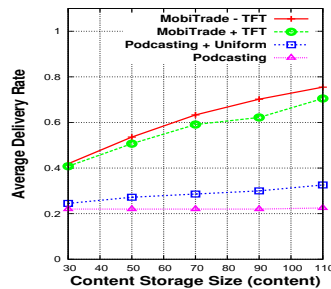


Fig. 5. Fixed number of channels per user and fixed channel size (SC₁).

Scenario SC₁: Fig. 5 compares the performance of MobiTrade with and without TFT and the two versions of Podcasting described in Section V-A. The figure of merit is, again, the average delivery rate, defined as the *amount of content received for channels a node requested* divided by the *total amount of content generated for these channels* (throughout the simulation). This is averaged over all nodes. Delivery rate is plotted as a function of node storage.

There are three main observations to be made in Fig. 5. *First*, collecting and sharing foreign channels (**MobiTrade and Podcasting + Uniform**) improves performance compared to only storing own channel. This confirms the findings of [20]. *Second*, the uniform sharing policy [20] is clearly not optimal (as suggested also in [15]), and is significantly outperformed by MobiTrade’s inventory management (by up to 2×). This is more pronounced as storage is increased. *Third*, using Tit-For-Tat (TFT) in a context where *all* nodes are well-intended results in a small drop of the average delivery rate

⁹Note that this age *only* corresponds to the time the content was inserted in the network by its publisher, and does not (directly) relate to the actual age (e.g. an old vs. a new rock song).

by about 6%, compared to the case without TFT. Using game theoretic terms, in Section V-D, we show that *rational users will choose to pay this price to secure themselves from selfish users*. Finally, Table III, summarizes the respective results for the KAIST trace (we only show values for a buffer of 110 contents; the plot trend is similar to Fig. 5). The results for the KAIST trace (row 1) corroborate the above findings.

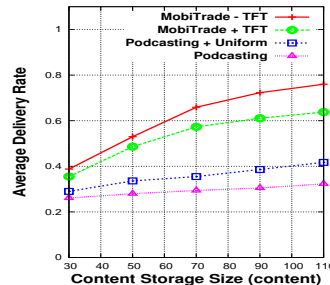


Fig. 6. Scenario SC₂.

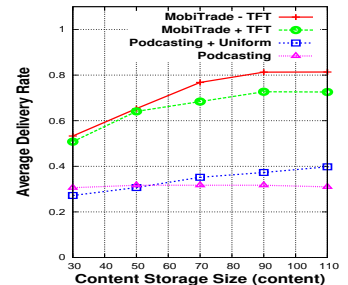


Fig. 7. Scenario SC₃.

TABLE III
AVG. DELIVERY RATE BASED ON THE REAL KAIST TRACE
(COLLABORATIVE SCENARIO, CONTENT STORAGE SIZE = 110 CONTENTS).

Policy:	MobiTrade + TFT	MobiTrade - TFT	Podcasting	Podcasting Uniform
Scenario SC ₁	0.83	0.89	0.6	0.72
Scenario SC ₂	0.78	0.86	0.75	0.69
Scenario SC ₃	0.79	0.88	0.68	0.74

Scenarios SC₂ and SC₃: These two scenarios consider the effect of heterogeneity with respect to channel demand (SC₂) and channel size (SC₃). The goal is to examine whether *asymmetry* of demand or supply of content (common in practice) could give rise to *deadlocks* due to the *inherent symmetry* of the Tit-For-Tat mechanism.

Figures 6 and 7 show the respective delivery rate for these two scenarios, as a function of storage space. As we can see, traffic asymmetry does not affect the main observations made in scenario SC₁. Interestingly, for (SC₂), Podcasting only own channels seems to outperform uniform sharing of foreign channels. Results for the KAIST trace are again in agreement (rows 2 and 3 of Table III)). We conclude that, even in the presence of asymmetric traffic, MobiTrade performs up to almost 2× better even without selfish nodes. Finally, while it is clear that these two scenarios do not suffice to *exclude* every probability of a deadlock, they constitute positive evidence to the robustness of MobiTrade. We defer an analytical treatment of deadlocks to future work.

Scenarios SC₄: The objective of this last scenario is to study the impact of *node churn* and the ability of MobiTrade to efficiently bootstrap new nodes. Here, 10 new users join the simulation after 8 hours, each one of them asks for 2 already existing channels, then, it leaves the simulation 8 hours later. Fig. 8 plots the average delivery rate among the 10 new users and the 40 existing ones as a function of time. It is evident there, that when the new set of users join already existing channels, they are not blocked. Instead, they are able to collaborate and quickly scale up their performance¹⁰.

¹⁰The important thing in this plot is the slope of the curves for new and old nodes, which matches, implying that both types get similar service. They differences in absolute value are only because nodes joining late have already missed part of the content already generated for this channel (and possibly dropped due to congestion).

Delay: Finally, in Fig. 9, we look at the average delivery delay of different schemes (scenario SC_1), measured as *time a matching content is received – time it was inserted in the channel*. We can clearly see that the ranking of schemes in terms of delay is similar to the one for delivery ratio (Fig. 5).

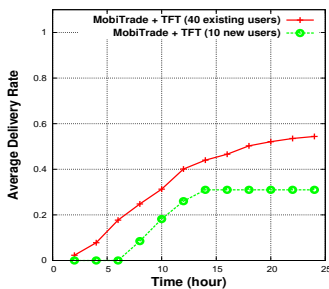


Fig. 8. Scenario SC_4 , 10 new users ask for already existing channels.

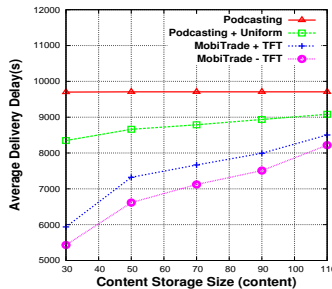


Fig. 9. Scenario SC_1 , studying the average delivery delay.

C. Scenarios with selfish users (SU)

Having established the good performance of MobiTrade in the absence of selfish nodes, we now turn our attention to scenarios where few or more nodes might not reciprocate for content they receive. We deem such scenarios as the norm rather than the exception in the real world. As mentioned earlier, most related proposals do not deal (explicitly) with such users. We consider two such scenarios, as described in Table IV: In SS_1 , we consider 10 selfish users among 50 that ask for different channels than those requested by the remaining collaborative users. In SS_2 , we consider the same number of selfish users which ask randomly for channels already requested by collaborative users. Selfish users and Collaborative users are denoted with “SU” and “CU”, respectively.

TABLE IV
SIMULATION SCENARIOS INCLUDING SELFISH USERS.

Sim. Scenario:	SS_1	SS_2
Nbr. of Users:	40 CU + 10 SU	40 CU + 10 SU
Nbr. of CH(s)	CU: 2/20 - SU: 2/10 (SU and CU channels differ)	CU, SU: 2/20 (among same channels)
Size of CH(s)	CU: 20 - SU: 40	CU, SU: 20

Scenario SS_1 : Fig. 10 depicts the average delivery rate (for different user strategies, CU and SU) with and without the TFT mechanism enabled. At high congestion (storage of 50 contents), enabling the TFT mechanism increases the average delivery rate among collaborative users by 15% (16% using the KAIST trace, Table V) and decreases it among selfish users by 63%. Indeed, enabling the TFT mechanism blocks selfish users and makes MobiTrade re-dispatch/reuse the saved resources among the channels shared by collaborative users. For a storage of 110, collaborative users are able to reach 73% higher throughput than selfish ones, by using TFT. The latter see a 3–4 \times drop in performance. In the same context, as shown in Table VI, the Podcasting scheme cannot control selfish nodes, as expected, and as their numbers increase, the latter end up outperforming collaborative ones.

Scenario SS_2 : Here, the 10 selfish nodes ask for channels already requested and carried by collaborative ones. This means that the utility management mechanism cannot affect them, allowing more opportunities to “scrape” content. Fig. 11 plots the average delivery rate of (MobiTrade + TFT) among

collaborative users in two cases: first (i), when selfish users are active and second (ii) when they are inactive. Clearly, when TFT is used, the performance of collaborative users is not harmed (verified also for the KAIST trace, Table V), while the one of selfish users drops severely, by up to 2.1 \times for a storage of 110 contents¹¹. This result consolidates our findings in Section IV regarding the impact of selfish users on the performance of collaborative ones once they both join the same channels. Indeed, selfish users are simply considered by MobiTrade as users which don’t ask for the channels. The system resources are kept safe and only dispatched among collaborative users.

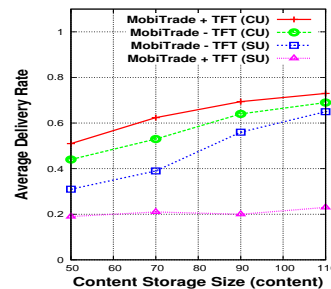


Fig. 10. Scenario SS_1 , impact on collaborative users.

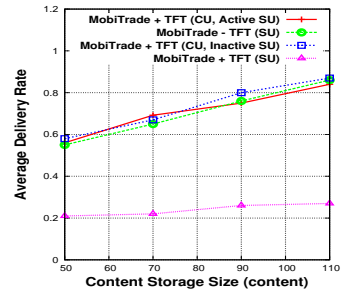


Fig. 11. Scenario SS_2 , impact on collaborative users.

TABLE V
AVG. DELIVERY RATE BASED ON THE REAL KAIST TRACE (SCENARIO INCLUDING SELFISH USERS, CONTENT STORAGE SIZE = 50 CONTENTS).

Policy:	MobiTrade (CU)	MobiTrade (CU)	MobiTrade + TFT(SU)	MobiTrade - TFT(SU)
SS_1 :	0.79 (+TFT)	0.68 (-TFT)	0.21	0.57
SS_2 :	0.81 (+TFT, Inactive SU)	0.78 (+TFT, Active SU)	0.24	0.77

TABLE VI
AVG. DELIVERY RATE (HCCM MOBILITY MODEL, CONTENT STORAGE SIZE = 110 CONTENTS, CU AND SU ASK FOR DIFFERENT CHANNELS).

Nbr. SU(s):	5	10	15	20
MobiTrade + TFT(CU):	0.8	0.76	0.71	0.62
MobiTrade + TFT(SU):	0.25	0.22	0.2	0.17
Podcasting + Uniform(CU):	0.46	0.4	0.37	0.34
Podcasting + Uniform(SU):	0.29	0.33	0.36	0.39

D. Choosing Strategies in MobiTrade

We have so far considered scenarios with x collaborative nodes or y selfish ones. But why would any node choose to be selfish or collaborative? Our results suggest that selfish behavior pays off if other nodes have TFT off, but hurts when TFT is on. And why would a collaborative node choose to employ our TFT mechanism? Our results suggest that *if all nodes are collaborative* they might get more content by turning TFT off. When multiple nodes have with these choices, where would the system converge? We *sketch* below a game theoretic framework that tries to answer these questions (due to space limitations, we don’t include here a more rigorous treatment).

¹¹We observe that in this, as well as the previous scenario, selfish users are not 100% isolated. This is *only due* to the generosity mechanism described in Section IV and the fact that we chose the minimum unit of transmission α to be one content, for simplicity. Increasing the amount of content in the network or reducing the value of α (note that this does not affect collaborative users much, due to the multiplicative increase), further isolates selfish nodes.

Let's consider a simple case of two nodes with contents to exchange. The set of strategies \mathcal{A} for each node is to choose from is: (i) being selfish (SU), (ii) being collaborative while activating TFT (CU + TFT) or (iii) being collaborative but not activating TFT (CU - TFT). Let M be the amount of content these nodes get if they normally use MobiTrade (CU+TFT). Let finally γ be a *discount factor*, capturing the cost to a node (e.g. energy) of reciprocating a piece of content ($0 \leq \gamma \leq 1$). The *total payoff* to each node from getting M contents is

$$payoff = \gamma M.$$

The following matrix describes the ‘‘MobiTrade game’’ and the respective payoffs in normal form. It is not a zero-sum game.

TABLE VII
MOBITRADE GAME AND PAYOFFS.

	CU + TFT	CU - TFT	SU
CU + TFT	$[\gamma M, \gamma M]$	$[\gamma M^+, \gamma M]$	$[0, 0]$
CU - TFT	$[\gamma M, \gamma M^+]$	$[\gamma M^+, \gamma M^+]$	$[(\gamma - 1)M^+, M^+]$
SU	$[0, 0]$	$[M^+, (\gamma - 1)M^+]$	$[0, 0]$

We note that M^+ is the somewhat higher payoff a node gets if both nodes are not selfish and disable TFT (see e.g. Fig. 5) and $(\gamma - 1)M^+ (\leq 0)$ is the cost to a node of sending M contents without getting something back. We can see that if $\gamma = 0$ (cost per content is equal to gain), then no user has an incentive to participate in the system (i.e. users will be selfish). This however, as well as $\gamma = 1$, are unrealistic cases.

The more interesting cases are for $0 < \gamma < 1$. In this case, it is easy to see that the game has a single *Nash equilibrium* at (CU + TFT, CU + TFT) with payoffs $(\gamma M, \gamma M)$. That is, none of the two nodes can *strictly* improve its payoff by a *unilateral* change of strategy [13]. In other words, for nodes participating in our network choosing to use MobiTrade with TFT is the optimal (selfish or rational) strategy, which is a very desirable outcome for our system. This analysis can be easily extended to N nodes. The main difference there is that (CU+TFT) has a non-zero reward even if all but one other users are not SU (making the equilibrium stronger).

Finally, deactivating TFT, i.e. point (CU-TFT, CU-TFT), is the socially optimal operating point and is also *Pareto optimal*. However, it is not an equilibrium except for the limiting case of $\gamma = 1$ (no cost). For all other cases, the *price of anarchy* per node is equal to $\gamma(M^+ - M)$, which as we saw in the results of this section, is only a small price to pay.

VI. CONCLUSIONS AND FUTURE WORK

In this work, we investigated the content dissemination problem over disruption tolerant networks while considering the possible existence of selfish users. Inspired from real life trading behavior, we proposed MobiTrade, a complete framework that incites users to collaborate, profiles their needs and manages their device resources optimally towards maximizing their revenues in terms of contents. Using NS3 simulations based on a synthetic mobility model (HCMM), and a real mobility trace (KAIST), we show that selfish users are isolated and system resources are only allocated among collaborative users. Finally, using a game theoretic framework, we show that turning on MobiTrade is the optimal strategy to use for selfish, rational users. In future work, we intend to consider more complex content structures (e.g. hierarchical channels,

semantic matching, etc.) and their effect on our system, as well as to perform a rigorous analytical treatment of our framework, that will allow us to gain further insight into the complex interactions (mobility, interests, and detailed resource models) underlying it.

REFERENCES

- [1] AT&T: Improving 3G network. <http://gigaom.com/2008/06/08/3g-network-iphone/>.
- [2] Comscore statistics. <http://www.comscore.com>.
- [3] KAIST mobility traces. <http://research.csc.ncsu.edu/netsrv/?q=node/4>.
- [4] The NS-3 network simulator. <http://www.nsnam.org>.
- [5] A. Balasubramanian, R. Mahajan, and A. Venkataramani. Augmenting mobile 3g using wifi. In *Proc. of ACM MobiSys*, 2010.
- [6] C. Boldrini, M. Conti, and A. Passarella. Users mobility models for opportunistic networks: the role of physical locations. In *Proc. of IEEE WRECOM*, 2007.
- [7] C. Boldrini, M. Conti, and A. Passarella. Design and performance evaluation of contentplace, a social-aware data dissemination system for opportunistic networks. In *Elsevier Computer Networks*, 2010.
- [8] P. Brémaud. *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*. Springer, Berlin, Germany, 2001.
- [9] L. Buttyan, M. Felegyhazi, and I. Vajda. Barter-based cooperation in delay-tolerant personal wireless networks. In *Proc. of IEEE AOC*, Helsinki, Finland, 2007.
- [10] M. Felegyhazi, J. P. Hubaux, and L. Buttyan. Nash equilibria of packet forwarding strategies in wireless ad hoc networks. *Transactions on Mobile Computing*, 2006.
- [11] W. Gao, Q. Li, B. Zhao, and G. Cao. Multicasting in delay tolerant networks: A social network perspective. In *Proc. of ACM MobiHoc*, 2009.
- [12] B. Han, P. Hui, M. Marathe, G. Pei, A. Srinivasan, and A. Vullikanti. Cellular traffic offloading through opportunistic communications: A case study. In *Proc. of ACM CHANTS*, 2010.
- [13] J. C. Harsanyi and R. Selten. *A General Theory of Equilibrium Selection in Games*. The MIT Press, 1988.
- [14] O. Helgason, E. Yavuz, S. Kouyoumdjieva, L. Pajevic, and G. Karlsson. A mobile peer-to-peer system for opportunistic content-centric networking. In *Proc. of ACM MobiHeld*, 2010.
- [15] L. Hu, J. L. Boudec, and M. Vojnovic. Optimal channel choice for collaborative ad-hoc dissemination. In *Proc. of IEEE INFOCOM*, San Diego, CA, USA, 2010.
- [16] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard. Networking named content. In *Proc. of ACM CoNEXT*, 2009.
- [17] S. Jung, U. Lee, A. Chang, D. Cho, and M. Gerla. Bluetorrent: Cooperative content sharing for bluetooth users. In *Proc. of IEEE PerCom*, 2007.
- [18] A. Krifa, C. Barakat, and T. Spyropoulos. Message drop and scheduling in DTNs: Theory and practice. Technical report, HAL INRIA, 2010.
- [19] A. Krifa, M. K. Sbai, C. Barakat, and T. Turletti. Bithoc: A content sharing application for wireless ad hoc networks. In *Proc. of IEEE PerCom*, 2009.
- [20] V. Lenders, G. Karlsson, and M. May. Wireless ad hoc podcasting. In *Proc. of IEEE SECON*, New York, NY, USA, 2007.
- [21] T. Locher, P. Moor, S. Schmid, and R. Wattenhofer. Free riding in bittorrent is cheap. In *Proc. of ACM HotNets*, Irvine, CA, USA, 2006.
- [22] M. May, C. Wacha, V. Lenders, and G. Karlsson. Wireless opportunistic podcasting: Implementation and design tradeoffs. In *Proc. of ACM CHANTS*, 2007.
- [23] M. Motani, V. Srinivasan, and P. Nuggehalli. Peoplenet: engineering a wireless virtual social network. In *Proc. of ACM MobiCom*, 2005.
- [24] L. Pelusi, A. Passarella, and M. Conti. Opportunistic networking: Data forwarding in disconnected mobile ad hoc networks. *IEEE Commun. Mag.*, 44(11), 2006.
- [25] M. R. Ra, J. Paek, A. B. Sharma, R. Govindan, M. H. Krieger, and M. J. Neely. Energy-delay tradeoffs in smartphone applications. In *Proc. of MobiSys*, San Francisco, California, USA, 2010.
- [26] G. Sollazzo, M. Musolesi, and C. Mascolo. TACO-DTN: A time-aware content-based dissemination system for delay tolerant networks. In *Proc. of ACM MobiOpp*, 2007.
- [27] B. Sonja and J. Y. L. Boudec. A robust reputation system for peer-to-peer and mobile ad-hoc networks. In *Proc. of P2PEcon*, 2004.
- [28] T. Spyropoulos, N. B. Rais, T. Turletti, K. Obraczka, and T. Vasilakos. Routing for disruption tolerant networks: Taxonomy and design. In *ACM/Kluwer Wireless Networks (WINET)*, 2010.
- [29] Z. Wenrui, A. Mostafa, and Z. Ellen. Multicasting in delay tolerant networks: semantic models and routing algorithms. In *Proc. of ACM SIGCOMM WDTN*, 2005.