# MobiTrade: Trading Content in Disruption Tolerant Networks

Amir Krifa
Project-Team Planète, INRIA
Sophia Antipolis, France
Amir.Krifa@inria.fr

Chadi Barakat
Project-Team Planète, INRIA
Sophia Antipolis, France
Chadi.Barakat@inria.fr

Thrasyvoulos Spyropoulos
EURECOM, France
Spyropou@eurecom.fr

## ABSTRACT

In this paper, we present MobiTrade, a utility driven trading system for efficient content dissemination on top of a disruption tolerant network. While simple tit-for-tat (TFT) mechanisms can force nodes to "give one to get one", dealing with the inherent tendency of peers to take much but give back little, they can quickly lead to deadlocks when some (or most) of interesting content must be somehow fetched across the network. To resolve this, MobiTrade proposes a *trading* mechanism that allows a node ("merchant") to buy, store, and carry content *for other nodes* (its "clients") so that it can later trade it for content it is personally interesting in. To exploit this extra degree of freedom, MobiTrade nodes continuously *profile* the type of content requested and the collaboration level of encountered devices. An appropriate utility function is then used to collect an optimal inventory that maximizes the expected value of stored content for future encounters, *matched* to the observed mobility patterns, interest patterns, and collaboration levels of encountered nodes. 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 devices.

## 1. 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 they provide to existing and novel applications, are generating unprecedented amounts of mobile data. The capacity of current cellular infrastructures has already been driven to the limit [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. This has lead many researchers (and industry) to explore alternative or hybrid architectural solutions [13].

To this end, direct mobile-to-mobile communication can be leveraged to harvest the large amounts of unused bandwidth between wireless devices in proximity. While multi-hop communication over mobile devices has been recently dealt with in the context of Delay Tolerant Networks (DTNs), increasing user demand for content is creating a shift in focus towards content and data centric systems (e.g. the CCN project [11]), in both wired and wireless Internet. As a result, a number of content dissemination systems have been recently proposed for mobile devices *in the wild* to exchange content of interest in a peer-to-peer manner [19, 18, 16, 17, 7, 15, 14, 6].

In addition to dealing with the challenging networking conditions, content sharing systems for DTNs have two main functions to perform: *(i)* propagation of interests and content discovery; *(ii)* delivery of matching content (over one or more hops); A number of architectural decisions can be made to achieve these goals, leading to publish/subscribe systems [19, 18], query-based and broker-based [16, 20, 17, 7, 15, 14, 6], etc. These systems aim to maximize the amount of useful content users can receive from the network. Nevertheless, distributed (or peer-to-peer) content sharing systems have one more important goal: *(iii) to ensure enough nodes collaborate to make the system interesting to participants.* This latter goal is often *conflicting* with optimal algorithms for *(i)* and *(ii)*, and has been a major *deal-breaker* in most envisioned architectures for mobile ad hoc networks [12]. 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, unless cooperation is somehow incentivized and free-riders penalized.

The following *architectural dilemma* arises then when considering a content sharing architecture over *non-altruistic* mobile devices. Nodes can choose to only store and share content they personally consume (thus somewhat mitigating selfish *inclinations*) [16]. This greatly simplifies content discovery and delivery. To further protect nodes against free-riders, a tit-for-tat (TFT) mechanism could be enforced. Yet, this approach is very restrictive: content of interest can be retrieved *only if the set of encountered nodes are also interested in (and thus carry) the requested content.* This can lead to long delays and a *suboptimal* query success rate *even if TFT is not used*, if nodes with common interest do not meet each other often.

To improve hit rates, nodes could use their spare resources (contact bandwidth, disk space) to collect, store, and relay additional content, not meant to be consumed locally [7, 14, 15, 14, 6]. An interesting optimization problem then arises: how should the total spare bandwidth in the network be optimally allocated to available content so as to maximize the overall network hit rate? Answers include randomized or popularity-based local heuristics [17], using the available buffer space only for *friends* and social peers [7, 14, 6], as well as optimal distributed algorithms [15]. Unfortunately, none of these solutions answers *why* participating nodes should collaborate implementing the policy of choice. In fact, we argue that *the above optimization problem needs to be turned on its head, in light of the non-altruistic nature of users.*

MobiTrade optimizes the content sharing strategy from the perspective of each individual participant. First, we argue that Tit-For-Tat (TFT) should be directly employed in order to *(a)* isolate free-riders and *(b)* create incentives for nodes to share their resources. This latter point is of key importance as TFT gives content of non-direct interest monetary value. If a node B has content that A is interested in, but A does not have something to give back, A now has the incentive to fetch something for B (perhaps from a remote node that

B never encounters). B now retrieves content that would otherwise be inaccessible to it (due to its mobility pattern), and A retrieves content that is easy accessible but that it couldn't *afford* before. While TFT is well known both in P2P [9] and opportunistic networks [8] communities, it does not answer itself *how mobile devices should optimally (re-)act in the presence of TFT* towards maximizing their revenues. MobiTrade answers this question by introducing a content utility framework that aims to *maximize the expected future exchange value of the content inventory stored by each node.* Intuitively, the value of a piece of content to a node A should depend on *(i)* how many nodes are interested in it, *(ii)* how often does A see these nodes, *(iii)* how much content, interesting to A, do these nodes have, *(iv)* how *well-behaved* are these nodes. MobiTrade uses a simply, robust utility function that implicitly captures all these features, without explicitly measuring each one, and that turns each node into a *merchant* fetching the content that has the highest chance to be sold (and exchanged for content of interest) to its good *clients.* Summarizing, the major contributions of this paper are:

1. We formulate the optimal content sharing problem in DTNs from the perspective of non-altruistic nodes while relying on a tit-for-tat mechanism to isolate free-riders.

2. We propose MobiTrade, a utility-based solution to this problem that predicts the (exchange) value of each piece of content and provides a customized resource allocation strategy for each node, *matched* to its own interests and mobility pattern.

To our best knowledge, this is the first content sharing system for DTNs that can both deal with rational and selfish nodes while at the same time achieving good global outcomes *without explicit hard constraints on the topology and dependency of nodes or on their social behavior.* A work we are aware of in a somewhat related direction is the unpublished work [21]. However, the authors there deal with a content sharing system based on swarms, and thus explicitly resembling existing P2P systems

The rest of this paper is organized as follow. Section 2 describes the MobiTrade architecture. Then, we provide a detailed simulation analysis in Section 3 based on both synthetic and real mobility traces and we compare MobiTrade to different content dissemination policies. Finally, we summarize our conclusions and discuss future work in Section 4.

# 2. MOBITRADE

In this section, we start by presenting the main data records and the generic communication protocol used in MobiTrade. We then look deeper into the channel utility framework used by each MobiTrade node to derive a trading strategy that will maximize its reward, and to manage its resources accordingly. We have implemented our MobiTrade architecture for the Android platform. More details about this architecture and its implementation can be found in [10] and [3], respectively.

## 2.1 MobiTrade Data Records

In a content sharing architecture, users need first to express their interests for different contents. To this end, we borrow the concept of *channels*, introduced in [17], because of its generality and popularity [7, 15]. Specifically, the MobiTrade architecture relies on two data records: content and channel (Fig. 1).

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 or by subscribing to an existing channel [17]. A desirable content is identified based on a match between the channel keywords and the content description. A lot more can be said about this channel structure (e.g. hierarchies, merging and splitting of channels, semantic content matching, etc.). We choose here to use a simple channel structure and focus on the algorithmic part of the system. Each channel record contains a *utility* entry. This is a key quantity for MobiTrade, allowing our system to optimize various important functions. Finally, a *content record*, in addition to its description, contains fields to deal with expired content and security (see [10] for details).

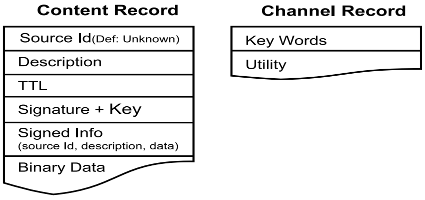

**Figure 1: MobiTrade data records and channel storage**

## 2.2 MobiTrade Protocol

In addition to declaring interest in channels, each node may choose to *carry* a set of channels, i.e. store and share content for these channels. In other words, after consuming a content, a node may choose to keep it in a *shared* part of its storage and make it available to other nodes (similar to *seeding* in P2P systems). Then, each time a new meeting opportunity arises with another mobile device, each device starts by sending its list of (subscribed) channels to the other device. Based on it, each device identifies the contents in its own buffer that match its peer's interests.

Let node A meet node B, and let $X_{A\to B}$ denote the shared contents of A matching B's interests, and $X_{B\to A}$ the contents of B matching A's interests. *If both nodes are collaborative*, A forwards $X_{A\to B}$ and receives $X_{B\to A}$ (this assumption is made in most related DTN content sharing schemes [16, 17, 7, 15, 14, 6]). However, A has no way of ensuring that B will do its part. In fact, it has *no way of affecting B's strategy.* B can decide to forward nothing back (e.g. to save power, or because of malice/selfishness). To protect against such *free-riders*, a Tit-Fot-Tat (TFT) mechanism can be implemented, in which A gives back one content (or X bytes) for every content (or X bytes) it receives from B. If we denote with $R(i)$ the *reward* (amount of content retrieved) for node $i$ by this transaction, the following outcomes are possible (from the perspective of A):

1. $R_A = X_{B\to A}$, $R_B = X_{A\to B}$ (TFT off | A and B collaborative)

2. $R_A = 0$, $R_B = X_{A\to B}$ (TFT off | A collaborative, B selfish)

3. $R_A = R_B = 0$ (TFT on | A collaborative, B selfish)

4. $R_A = R_B = \min\{X_{B\to A}, X_{A\to B}\}$ (TFT on | A and B collaborative)

Outcomes (2) and (3) show the well-known effect of TFT in isolating free-riders. However, outcomes (1) and (4) have some deeper implications. First, if both nodes are collaborative, there is a potential penalty for each transaction if TFT is *on*, equal to

$$|X_{B\to A} - X_{A\to B}| \text{ (potential loss due to TFT).}$$

This is the amount of additional data that could have been retrieved (by one of the peers) during this transaction. The question raised then is whether nodes would have an incentive to turn on TFT (e.g. if they assume that most peers are collaborative). We answered this question in [10].

Second, TFT, in addition to dealing with free-riders, has the important effect of giving *exchange* or *monetary value* to each content stored. In other words, *TFT couples the* *strategies of participating peers, and allows nodes to affect their peers' policies through their own actions.* Assume both nodes are collaborative and $X_{A\to B} > X_{B\to A}$. Then with TFT on, B cannot get all the content of interest in A's shared buffer, because it does not have enough to give back (i.e. enough money to *buy* all this content). A new option is now presented to B: it can try to collect, from around the network, additional content matching A's interests (perhaps with a small additional penalty on its resource usage), in order to increase $X_{B\to A}$[1]. Two positive outcomes come as a result of such a decision: *(i)* B can *afford* more of A's shared content of interest; *(ii)* A now receives additional content of interest *fetched* over one more or multiple hops from nodes that A may not see frequently or ever.

Summarizing, with the addition of TFT, nodes now have incentives to increase the exchange value of their inventory, so as to increase the amount of interesting content they can buy from encountered nodes. In other words, the need for nodes to store and relay *foreign* channels to improve global performance (observed in [16, 15]) and nodes' *selfish* interests are now (better) aligned in a type of *market* system established by TFT. In this market system, the following optimization question arises *from the perspective of each node*: *which channels and how much of each should it carry in its buffer, so as to maximize its future reward* (amount of interesting content retrieved in subsequent contacts)? The following section provides answer to this question.

## 2.3 Optimal Buffer Allocation

Let $R_i^{(n)}$ be random variables measuring the total reward node A receives upon contact $n$ (with some random node) for (selling) content of channel $i$. Clearly, $\sum_i R_i^{(n)} = R^{(n)}$, the total amount of useful content A receives during this $n^{\text{th}}$ contact. Then, if node A is at contact $k-1$, it would like to maximize the following quantity:

$$\sum_{n=k}^{\infty}\sum_i R_i^{(n)} = \sum_i \sum_{n=k}^{\infty} R_i^{(n)}. \quad (1)$$

Let now $X_i^{(n)}$ be identically distributed random variables with average $\bar{X}_i$, measuring the amount of content actually *requested* by the encountered node for channel $i$. Assuming a limited buffer space of size B and no deterministic knowledge of future demands $X_i$, node A can allocate a fraction $\alpha_i B$ ($0 \le \alpha_i \le 1$) to carry content for channel $i$, in order to satisfy the *predicted* demand (and reap the reward from selling this much). Then

$$R_i^{(n)} = \min\{\alpha_i B, X_i^{(n)}\} \quad (2)$$

are identically distributed random variables.

Clearly, the higher $\alpha_i$ the smaller the chance that the actual demand will exceed the amount of content available, and thus the smaller the *opportunity cost*. At the same time, less space is left in the inventory for carrying content that could satisfy demand for other channels. By the law of large numbers, $\sum_{n=k}^{\infty} R_i^{(n)} \to E[R_i]$. This implies that each node can simply focus on maximizing the expected reward upon the next contact. Furthermore,

$$E[R_i] = E[\min\{\alpha_i B, X_i\}] = \int_0^{\alpha_i B} P(X_i > x)dx. \quad (3)$$

A node is then faced with the following optimization problem:

$$\max_{\alpha_1,\alpha_2,\dots,\alpha_{|CH|}} \sum_i \int_0^{\alpha_i B} P(X_i > x)dx, \quad (4)$$

$$\sum_i \alpha_i \le 1, \quad \alpha_i \ge 0, \forall i. \quad (5)$$

The Lagrangian for this optimization problem is

$$\mathcal{L} = \sum_i E[R_i] + \lambda(\sum_i \alpha_i - 1) + \sum_i \gamma_i \alpha_i. \quad (6)$$

and the KKT optimality conditions are

$$\frac{\partial \mathcal{L}}{\partial \alpha_i} = P(X_i > \alpha_i B) + \lambda + \gamma_i = 0, \quad (7)$$

$$\sum_i \alpha_i - 1 = 0, \quad \gamma_i \alpha_i = 0, \forall i, \quad (8)$$

where the last two are the complementary slackness conditions. This is a system of equations that can be solved for $\alpha_i$. By observing Eq.(8) we can conclude that a node either allocates no buffer space for channel $i$ ($\alpha_i = 0$), or if some buffer space is allocated ($\alpha_i > 0$, $\gamma_i = 0$), then

$$P(X_i > \alpha_i^* B) = -\lambda \quad (9)$$

that is, the optimal allocation $\alpha_i^* B$ is equal to the $-\lambda^{\text{th}}$ quantile of $X_i$ ($\lambda$ can be obtained from Eq.(7) and Eq.(8)).

**Unknown Request Distribution :** The above result requires the distribution of requests for channel $i$, $P(X_i \le x)$, to be known. If the distribution is not known or cannot be obtained easily, except up to a first and/or second moment, we can assume that $X_i$ follows a Gaussian distribution $\mathcal{N}(\bar{X}_i, \sigma_i)$ [2]. Then, normalizing

$$P(X_i > \alpha_i B) = P(\frac{X_i - \bar{X}}{\sigma_i} > \frac{\alpha_i B - \bar{X}}{\sigma_i}) = -\lambda$$

$$\Rightarrow \frac{\alpha_i B - \bar{X}}{\sigma_i} = f(-\lambda) \Rightarrow \alpha_i B = f(-\lambda)\sigma_i + \bar{X}_i,$$

where $f(-\lambda)$ is given from the standard Gaussian.

We can solve $\sum_i \alpha_i = 1$ for $f(-\lambda)$ and replace above to get the optimal allocation

$$\alpha_i^* B = \bar{X}_i + \sigma_i[\frac{B - \sum_j \bar{X}_j}{\sum_j \sigma_j}]. \quad (10)$$

Eq.(10) has some very interesting implications:

- ($\sum_j \bar{X}_j \le B$) If the buffer is large enough to satisfy the expected demand, then the optimal policy is to *gamble* the remaining buffer space proportionally to the variance for this channel.
- ($\sum_j \bar{X}_j > B$) However, if the buffer space cannot even fit the expected demand, the optimal policy is to be conservative and not give much space on *risky* (high variance) channels.

Finally, if we further assume that the standard deviation for the random request is proportional to the mean, that is, $\sigma_i = c\bar{X}_i$ or $\frac{\sigma_i}{\bar{X}_i} = c$, for some constant $c$, (i.e. the *relative* uncertainty for each channel is the same), then

$$\alpha_i^* B = \frac{\bar{X}_i}{\sum_j \bar{X}_j} B. \quad (11)$$

In other words, the optimal strategy for each node is to (try) to allocate its buffer space proportionally to the *expected* demand per channel $\bar{X}_i$.

## 2.4 Channel Utility in Practice

The previous section calculates the optimal buffer allocation, given knowledge about the channel demand distribution $P(X_i = x)$, and its mean $\bar{X}_i$[3]. Given this mean, a node can

[1] We note here that this is the case if the *satisfaction* of getting X extra bytes is higher than the cost (battery, bandwidth) expended to collect X bytes to be used in exchange. In most cases, this is a reasonable assumption.

[2] In fact, Gaussian above implies a probability for negative demand. We assume that $\sigma_i$ is such to ensure this probability is negligible and choose to keep this calculation approximate, in the interest of space.

[3] While second and higher moments of the demand $X_i$ could be estimated and considered when dropping or scheduling content (as in Eq.(9) and (10)), due to space limitations, we focus on the simple case of Eq.(11).

derive appropriate utilities per channel $i$, that will be used to drop content (if the buffer is full) and schedule content (if contact duration is limited). However, in practice, a node cannot measure $X_i$ directly. Instead, it can only measure the *actual* amount sold for channel $i$ during contact $n$, namely the reward $R_i^{(n)}$, given by Eq.(2). We therefore propose the following estimator for $\alpha_i^*$, the optimal buffer quota for channel $i$:

$$\hat{\alpha}_i = \frac{\frac{1}{n}\sum_n R_i^{(n)}}{\sum_j \frac{1}{n_j}\sum_{n_j} R_j^{(n_j)}}. \tag{12}$$

If $R_i^{(n)}$ is a stationary and ergodic process (i.e. if $X_i^{(n)}$ is stationary and ergodic), then

$$\hat{\alpha}_i \underset{n\to\infty}{\to} \frac{E[R_i]}{\sum_j E[R_j]}. \tag{13}$$

Indeed, we can calculate $E[R_i]$ as follows

$$\begin{aligned}
\text{(from Eq.(2),(11))} \quad & R_i = X_i + I(X_i > \tfrac{\bar{X}_i B}{\sum_j \bar{X}_j})(\tfrac{\bar{X}_i B}{\sum_j \bar{X}_j} - X_i) \\
\Rightarrow \quad & E[R_i] = \bar{X}_i + P(X_i > \alpha_i^* B)(\tfrac{\bar{X}_i B}{\sum_j \bar{X}_j} - \bar{X}_i) \\
\text{(from Eq.(9))} \quad \Rightarrow \quad & E[R_i] = \bar{X}_i(1 + \lambda - \lambda \tfrac{B}{\sum_j \bar{X}_j}).
\end{aligned}$$

In other words, the expected reward is proportional to the expected demand, and $\hat{\alpha}_i$ is an asymptotically unbiased estimator of the optimal quota, i.e.

$$\hat{\alpha}_i \underset{n\to\infty}{\to} \frac{\bar{X}_i}{\sum_j X_j} = \alpha_i^*. \tag{14}$$

and a node, can keep track of past requests in order to find the optimal buffer quota for this channel.

In practice, in order to absorb spikes in demand, as well as to keep track with long-term trend changes in the per channel demand, we choose to use an Exponential Weighted Moving Average (low pass) filter for averaging. Specifically, for a channel $CH$, the current estimate of the channel utility $\hat{R}_{CH}$ (i.e. the expected reward for $CH$) is updated as:

$$\hat{R}_{CH}^{(n+1)} = \omega \hat{R}_{CH}^{(n)} + (1-\omega) I(CH) CL_{CH}^{(n+1)}, \tag{15}$$

where $\omega$ is the weight associated to the low pass filter. $I(CH)$ is a binary variable that expresses whether the encountered node $B$ is interested or not in $CH$ (e.g. a channel that node A is not currying yet, but B would like A to bring some content for $CH$ next time - see also next section). This variable captures the popularity of a given channel over all MobiTrade devices met by $A$. $CL_{CH}^{(n+1)}$ captures the volume of contents that *could be sold* to device $B$ in the future. This is equal to the actual reward in this round $R_{CH}^{(n+1)}$ plus a *speculation* component used for *(a)* bootstrapping and *(b)* converging to the actual demand, as shown next.

**Collaboration and Bootstrapping :** If a channel is requested for the first time at the $(n+1)^{th}$ meeting, its $R_{CH}^{(n+1)}$ 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. 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 calculation of $CL_{CH}$ below is inspired from TCP slow start, and attempts to best satisfy the above two (conflicting) goals:

$$CL_{CH}^{(n)} = \begin{cases} Max(\alpha, 2.R_{CH}^{(n)}) & \text{if } R_{CH}^{(n)} < \beta, \\ R_{CH}^{(n)} + \alpha & \text{otherwise.} \end{cases}$$

If the (last) measured reward $R_{CH}^{(n)}$ is less than some threshold $\beta$, the predicted future reward (i.e. the future utility of the channel) is doubled, to accelerate the collaboration process at its beginning; after $\beta$ (we take an optimistic approach and choose it equal to the maximum utility value over all channels of $A$), 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 $\alpha$ to $R_{CH}^{(n)}$.

The same factor $\alpha$ also serves to keep selfish nodes in control. Node A will give at most $\alpha$ to node B for channel $CH$, before asking for something in exchange (when TFT is implemented). If $B$ either does not have content to exchange or chooses not to reciprocate, then the weight of its request in Eq.(15) will stay minimal ($\leq \alpha$); a selfish/malicious user is then obliged to collaborate in order to increase the utilities of her channels and thus the portion of content storage these are given. Otherwise, her request is essentially ignored *not affecting the optimal buffer allocation.* 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 or reputation systems.

**Buffer Management and Scheduling Algorithm :** Based on the future reward estimates for channel $i$, $\hat{R}_i$, maintained as shown above, each node can define a buffer quota $B_i$ for each channel. According to Eq.(13) and (14) this is

$$B(i) = \frac{\hat{R}_i}{\sum_j \hat{R}_j} B. \tag{16}$$

Then if the amount of storage channel $i$ is currently occupying is $S(i)$, 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 $W + \sum_i S(i) < B$, then store the content.
- if $S(i) + W > B(i)$ and $W + \sum_i S(i) > B$, 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 the buffer 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 which causes unfairness. Instead, MobiTrade applies the *Weighted Fair Queuing* policy to prevent starvation of channels and ensures that contents are forwarded proportionally to the utility value of the channel they match. More details could be found in [10].

# 3. PERFORMANCE EVALUATION

*Protocols:* We have implemented MobiTrade 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 only corresponds to the forwarding process. The channel utility maintenance is kept *on* in all scenarios. We have also implemented two different versions of the *PodNet* scheme as a baseline for comparison, as described, to our best understanding in [17] and [16]: *(i)* non-collaborative Podcasting, where users just carry and share their own channels [17] (**Podcasting**); *(ii)* collaborative 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 [16] (**Podcasting + Uniform**). This latter strategy was shown to perform best in [16], compared to other heuristics taking into account channel popularity.

*Mobility Models:* To evaluate the different protocols, we use two mobility scenarios, a synthetic mobility model (HCMM) [5] and a real mobility trace (KAIST) [2]. More details about these scenarios can be found in [10]. We integrated both mobility models in NS3. Both case studies consist of simulations that last 24 hours where devices use the 802.11$b$ 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 size. 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). Finally, we consider that each user generates contents periodically that match one of the channels that were requested by users from other groups[4].

## 3.1 Collaborative scenarios

We first evaluate MobiTrade, assuming all nodes are collaborative, using the following four scenarios (Table 1) (there are 50 channels in total): $\mathbf{SC_1}$ implements a *homogeneous traffic* pattern, i.e. each channel has the same size and each user joins the same number of channels. In $\mathbf{SC_2}$, users choose a *random number of channels* to join, but channels still have the same size. In $\mathbf{SC_3}$, users ask for the same number of channels but these have *random sizes*. Finally, $\mathbf{SC_4}$ 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.

**Table 1: Collaborative simulation scenarios**

| Scenario: | $SC_1$ | $SC_2$ | $SC_3$ | $SC_4$ |
|---|---|---|---|---|
| Nbr. of Users: | 50 | 50 | 50 | 40 + 10 transient |
| Requested CH(s)/User | 2 | random [1, 20] | 2 | 2 |
| Size of CH(s) (contents) | 20 | 20 | random [1, 20] | 20 |

**Scenario $\mathbf{SC_1}$:** Fig. 2 compares the performance of MobiTrade with and without TFT and the two versions of Podcasting. The figure of merit is the average delivery rate (DR) [5], defined as the *amount of content received for channels a node requested* divided by the *total amount of content generated for these channels*. This is averaged over all nodes. There are three main observations to be made in Fig. 2. *First*, collecting and sharing foreign channels (**MobiTrade** and **Podcasting + Uniform**) improves performance compared to only storing own channels. This confirms the findings of [16]. *Second*, the uniform sharing policy [16] is clearly not optimal (as suggested also in [15]), and is significantly outperformed by MobiTrade's framework (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 DR by about 6%, compared to the case without TFT. Using game theoretic terms, we show in our our technical report [10] that *rational users will choose to pay this price to secure themselves from selfish users.* Finally, Table 2, summarizes the respective results for the KAIST trace. The results for the KAIST trace (row 1) corroborate the above findings.

**Table 2: Avg. DR (Real KAIST trace, collaborative scenario, content storage size = 110 contents).**

| Policy: | MobiTrade + TFT | MobiTrade - TFT | Podcasting | Podcasting Uniform |
|---|---|---|---|---|
| $\mathbf{SC_1}$ | 0.83 | 0.89 | 0.6 | 0.72 |
| $\mathbf{SC_2}$ | 0.78 | 0.86 | 0.75 | 0.69 |
| $\mathbf{SC_3}$ | 0.79 | 0.88 | 0.68 | 0.74 |

**Scenarios $\mathbf{SC_2}$ and $\mathbf{SC_3}$:** These two scenarios consider the effect of heterogeneity with respect to channel demand ($\mathbf{SC_2}$) and channel size ($\mathbf{SC_3}$). The goal is to examine whether *asymmetry* of demand or supply of content could give rise to *deadlocks* due to the *inherent symmetry* of the TFT mechanism. Figures 3 and 4 show the respective DR 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 $\mathbf{SC_1}$. Results for the KAIST trace are again in agreement (rows 2 and 3 of Table 2)). We conclude that, even in the presence of asymmetric traffic, MobiTrade performs up to almost 2× better even without selfish nodes. 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.

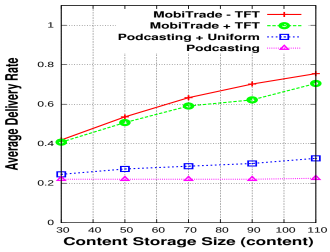

Figure 2: Scenario $\mathbf{SC_1}$.

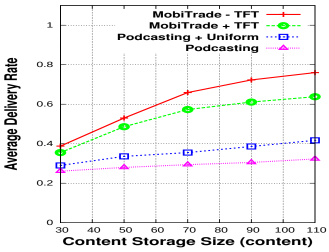

Figure 3: Scenario $\mathbf{SC_2}$.

**Scenarios $\mathbf{SC_4}$:** The objective of this scenario is to study the impact of *node churn* and the ability of MobiTrade to efficiently bootstrap new users. 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. 5 plots the average DR among the 10 new users and the 40 existing ones as a function of time. It is evident there, that the new users are not blocked. Instead, once they join the channels, they are able to collaborate and quickly scale up their performance.

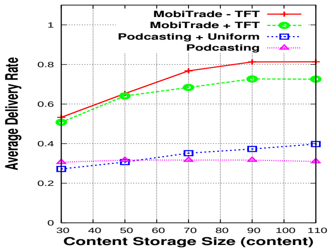

Figure 4: Scenario $\mathbf{SC_3}$.

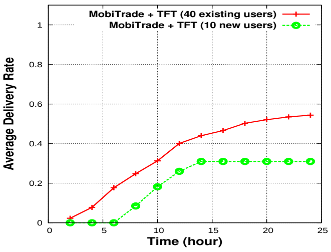

Figure 5: Scenario $\mathbf{SC_4}$

## 3.2 Scenarios with selfish users (SU)

We now turn our attention to scenarios where few or more nodes (selfish) 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 3: In $\mathbf{SS_1}$, we consider 10 selfish users (SU) among 50 that ask for different channels than those requested by the remaining collaborative users (CU). In $\mathbf{SS_2}$, we consider the same number of selfish users which ask randomly for channels already requested by collaborative users.

**Table 3: Simulation scenarios including selfish users.**

| 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 $\mathbf{SS_1}$:** Fig. 6 depicts the average DR (for different user strategies, CU and SU) with and without the TFT

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

[5] More results about the Delivery Delay performance study could be found in our technical report [10]

mechanism enabled. At high congestion (storage of 50 contents), enabling the TFT mechanism increases the average DR among collaborative users by 15% (16% using the KAIST trace, Table 4) 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 5, the Podcasting scheme cannot control selfish nodes, as expected, and as their numbers increase, the latter end up outperforming collaborative ones.

**Table 4: Avg. DR (real KAIST trace, scenario including SU, 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 |

**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. 7 plots the average DR 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 4), while the one of selfish users drops severely, by up to 2.1× for a storage of 110 contents[6]. This result consolidates our findings in Section 2.3 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.

**Table 5: Avg. DR (HCCM mobility, CU/SU ask for different CH.(s), content storage size = 110 contents).**

| 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 |
| Pod. + Uniform(CU): | 0.46 | 0.4 | 0.37 | 0.34 |
| Pod. + Uniform(SU): | 0.29 | 0.33 | 0.36 | 0.39 |

## 4. CONCLUSIONS AND FUTURE WORK

In this work, we investigated the content dissemination problem over DTN 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. In future work, we intend to consider more complex content structures and their effect on our system.

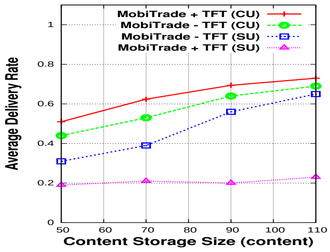

**Figure 6: Scenario SS$_1$, impact on CU and SU.**

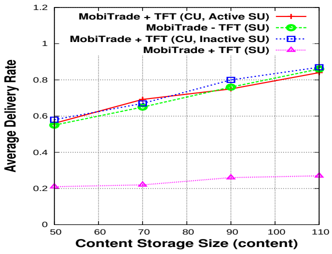

**Figure 7: Scenario SS$_2$, impact on CU.**

## 5. REFERENCES

[1] AT&T: Improving 3G network. http://gigaom.com/2008/06/08/3g-network-iphone/.

[2] KAIST mobility traces. http://research.csc.ncsu.edu/netsrv/?q=node/4.

[3] Mobitrade Android Application. http://planete.inria.fr/MobiTrade.

[4] The NS-3 network simulator. http://www.nsnam.org.

[5] C. Boldrini, M. Conti, and A. Passarella. Users mobility models for opportunistic networks: the role of physical locations. In *Proc. of IEEE WRECOM*, 2007.

[6] C. Boldrini, M. Conti, and A. Passarella. Modelling data dissemination in opportunistic networks. In *Proceedings of the third ACM workshop on Challenged networks*, CHANTS, San Francisco, California, USA, 2008.

[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] L. Buttyan, M. Felegyhazi, and I. Vajda. Barter-based cooperation in delay-tolerant personal wireless networks. In *Proc. of IEEE AOC*, 2007.

[9] A. K. et al. Bithoc: A content sharing application for wireless ad hoc networks. In *Proc. of IEEE PerCom*, 2009.

[10] A. K. et al. MobiTrade: Interest driven content dissemination architecture for Disruption Tolerant Networks. Technical Report, 2011.

[11] V. J. et al. Networking named content. In *Proc. of ACM CoNEXT*, 2009.

[12] M. Felegyhazi, J. P. Hubaux, and L. Buttyan. Nash equilibria of packet forwarding strategies in wireless ad hoc networks. *Transactions on Mobile Computing*, 2006.

[13] 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.

[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*, 2010.

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

[17] M. May, C. Wacha, V. Lenders, and G. Karlsson. Wireless opportunistic podcasting: Implementation and design tradeoffs. In *Proc. of ACM CHANTS*, 2007.

[18] M. Motani, V. Srinivasan, and P. S. Nuggehalli. Peoplenet: engineering a wireless virtual social network. In *Proc. of ACM MobiCom*, 2005.

[19] 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.

[20] N. A. Thompson, R. Crepaldi, and R. Kravets. Locus: A location-based data overlay for disruption-tolerant networks. In *Proc. of ACM CHANTS*, 2010.

[21] X. Zhou, S. Ioannidis, and L. MassouliÃl'. On the stability and optimality of universal swarms. Technical Report, 2011.

[6] We 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 2.3 and the fact that we chose the minimum unit of transmission $\alpha$ to be one content, for simplicity. Increasing the amount of content in the network or reducing the value of $\alpha$, further isolates selfish nodes.