

Priority Scheduling for Participatory Delay Tolerant Networks

Afra J. Mashhadi

Dept. of Computer Science

University College London

Gower Street, London WC1E 6BT, UK

A.JahanbakhshMashhadi@cs.ucl.ac.uk

Licia Capra

Dept. of Computer Science

University College London

Gower Street, London WC1E 6BT, UK

L.Capra@cs.ucl.ac.uk

Abstract—Thanks to advances in the computing capabilities and added functionalities of modern mobile devices, creating and consuming digital media on the move has never been so easy and popular. There exist cases where the people producing such media content, and those interested in receiving it, tend to be living in the same geographical area. Delay Tolerant Networking (DTN) protocols have been investigated as an effective means to distribute content in these dynamically changing settings. The main challenge addressed by researchers so far has been the maximisation of delivery probability, while also minimising the overall network overhead (e.g., number of replicas in the system, messages’ path length); because of the participatory nature of these networks, recent efforts have looked not simply to keep the overall network load low, but also to spread it equally across devices. Common to all these approaches is the treatment of messages as if they were all worth the same to recipients; as such, messages are forwarded hop-by-hop from source to destination in a sort of *first-encountered/first-forwarded* fashion. However, because of resource limitations on mobile devices (namely, battery), it is often the case that not all messages can be delivered. In this paper, we propose a new approach for priority-scheduling in participatory DTNs, whereby messages are being forwarded based on a combination of the likelihood of future encounters (physical layer) and the value that recipients attach to such messages (e.g., based on who produced the message). We implemented this priority scheduling on top of an existing DTN protocol, and evaluated the gain it entails in both end-user satisfaction and delivery, over a variety of real mobility traces (physical layer) and message values’ distributions (application layer).

Keywords—content distribution, prioritisation, human delay-tolerant networks

I. INTRODUCTION

Modern mobile phones (e.g., iPhone, Android-powered devices, Blackberry) have been rapidly gaining widespread popularity: functionalities such as high-resolution cameras, MP3 players, and GPS receivers have become a commodity; furthermore, multiple network interfaces of increasing bandwidth (e.g., 3G, Wi-fi, Bluetooth 2) are being offered, thus facilitating the creation and/or consumption of user-generated content (e.g., pictures, videos) on the go. There exist many urban scenarios where such content is appealing to a local community of users only: consider, for example, promo videos of events happening in town (e.g., festivals, street theaters, literary nights), music recordings of new

bands performing in the neighbourhood; media fliers about activities organised by university clubs and societies (e.g., sport races, debates, parties), and so on. In metropolitan cities like London, an estimated average of 5,000 social events take place every day; however, only (less than) half of these are being listed on popular websites (thus accessible via 3G networks from users’ mobile phones). The remainders are being advertised by word-of-mouth, posters affixed in given areas, hand-distributed fliers, and the like. In these cases, where the people producing content, and those willing to receive it, tend to be living in the same geographical area, content distribution can more effectively happen by means of Delay Tolerant Networking (DTN) protocols. These protocols exploit the freely available local networking capabilities of mobile devices (e.g., Wi-fi Direct [1]) to opportunistically distribute content during periods of colocation.

Research in this area has been very active, and a variety of protocols have been proposed (e.g., [2], [3], [4], [5], [6], [7], [8]) that aim to maximise message delivery, while minimising overall network overhead (e.g., number of replicas in the system, message path length). To achieve this goal, most of these approaches leverage upon the observation that human movement is predictable to a certain extent, and thus message carriers can be carefully selected so to favour routes with high delivery probability, whilst avoiding those most likely to fail. Reducing network load is indeed a fundamental challenge, as DTNs are participatory networks made of battery-constrained devices, whose lifetime is not expected to be significantly improved in the coming years (i.e., Moore’s Law will apply to the miniaturisation of battery size, rather than increasing its lifetime [9]). Recent works have been specifically tackling energy consumption issues, with the aim of either reducing *overall* network overhead (e.g., by further limiting the number of replica messages in the system [10], [11]), or with the aim of distributing the load more *fairly* across all participating nodes (e.g., [12], [13], [14]).

Common to all approaches proposed so far is the treatment of messages as if they were all worth the same to end users: the decision of what message to forward next, in the hop-by-hop path from source to destination, is entirely driven by the

next physical encounter, in a sort of *first-encountered/first-forwarded* basis. In other words, the content distribution network puts the same effort in trying to deliver every single message being produced. However, experience with the Web 2.0 demonstrates that users are very keen *prosumers* (producer-consumer) of media content, and the rate of production/consumption is set to increase if users can do so ubiquitously via their mobile phones. Physical resources (e.g., battery) will thus not be sufficient to distribute all content being generated. A question arises as to whether this forwarding mechanism is indeed appropriate in such scenarios: we expect end-users to attach different values (or priorities) to different messages, depending on, for example, who produced them in their social network. If resource limitations prevent us from delivering all messages being produced, a new challenge arises as to how to prioritise message forwarding, so to bring maximum satisfaction to the end users within a participatory DTN.

In this paper, we address the above challenge by means of a new approach for priority scheduling for participatory DTNs. After reviewing the state-of-the-art in DTN research (Section II), we present our approach (Section III), whereby messages are being forwarded based on a combination of the likelihood of future encounters (physical layer) *and* the value that recipients attach to such messages (application layer). To assess the gain in end-user satisfaction that priority scheduling brings, we have implemented it on top of a state-of-the-art DTN routing protocol, and evaluated it across various combinations of real human mobility traces (physical layer) and human social networks (application layer). We report the results of this evaluation in Section IV, before presenting our conclusion and future directions of research (Section V).

II. RELATED WORK

The parallel and steep growth of the prosumer figure on one side, and market penetration of modern mobile devices on the other, has fostered research in the area of DTN protocols, as means of effectively (i.e., high delivery) and efficiently (i.e., low overhead) disseminate content across geographically-bound and spontaneously-formed human networks. The first generation of DTN protocols assumed human movement to mostly follow the random waypoint mobility model [15]; to achieve high delivery probability, these protocols were replicating messages in the network, and relying on incidental deliveries caused by opportunistic encounters [16]. To reduce the amount of traffic they generated (and consequent network overhead), probabilistic routing schemes were developed, which used various forms of controlled flooding to strike a balance between delivery and overhead [4]. As real and large-scale traces of human mobility started to be collected, scientists demonstrated that human movement is actually not random, and that it can be predicted to a large extent [17], [18]. A second generation

of DTN protocols has then been proposed, that reasons upon human mobility patterns in a quest to better trade delivery with efficiency [2], [19], [20], [8].

All the above protocols rely on the spontaneous participation of end-users in the content dissemination network. In order to promote participation, load must be kept low and equally distributed, to avoid depleting the scarce energy available on mobile devices. The last generation of DTN protocols has thus removed the assumption of infinite available resources and unquestioned user participation: for example, RAPID [12] treats routing as a resource-allocation problem, in order to trade delay-related metrics (e.g., average delay) with consumed resources; FairRoute [13] reasons on social interaction strength between individuals to limit the number of messages that an intermediary will accept, thus avoiding some nodes becoming overloaded; CoHabit [14] explicitly reasons in terms of locally available resources and estimates of load at intermediaries, to guarantee an equal participation from each node in the network. These protocols acknowledge the fact that a limit must be placed on the amount of resources (be that bandwidth, battery, storage) that nodes are willing to share, consequently reducing the number of messages that intermediaries can forward at any point of time.

At present, DTN protocols treat messages equally, so that forwarding among intermediaries is entirely driven by physical encounters. However, end-users are not equally interested in all messages; because of the limits imposed by available resources, a new challenge arises as to *what messages* a carrier should forward, so that the overall end-user satisfaction is maximised. We propose an answer to this challenge next.

III. PRIORITY SCHEDULING

In this section, we introduce our approach to prioritisation in participatory DTNs. We first define a general model for measuring messages' priority in the network, based on users' interests (Section III-A). Based on this model, we illustrate the challenges that DTN routing protocols face (Section III-B), before dwelling into the details of the solution we propose (Section III-C).

A. Model

In order to design a content-distribution protocol that prioritises messages' delivery based on their value to end-users, we first need to quantify what this 'value' is. Two different approaches can be followed: users may define either *whom* (people-centric) or in *what* (content-centric) they are interested in. The former is typical of online social networks like Facebook or Twitter, where users explicitly define, in what we call a 'user profile', whom they are interested in receiving content from. The latter is typical of folksonomic-based websites like del.icio.us or CiteULike, where users define the topics they are interested in by means

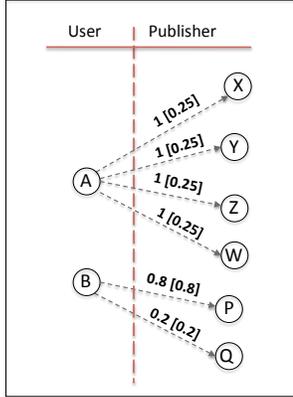


Figure 1: Node A's Interest Network

of freely chosen words (often called tags), regardless of whom published messages on these topics. In this work, we are not interested in what model the applications running on a user mobile phone will adopt. All we need is a way of quantifying how much a user A is interested in receiving a message produced by X .

In people-centric approaches, we would require a *weighted* user social network, whereby A 's profile not only states she wants to receive messages from X , but also how much she wants them $w_{A,X} \in (0, 1]$. These weights can either be explicitly defined by users (like in Rumble.com), or implicitly derived by looking, for example, at the frequency of interaction between users (e.g., in Twitter, the frequency of @username directed messages). Any message m produced by X would then be valued by A as $v_{A,X} \equiv w_{A,X}$. Similarly, in content-centric approaches, we would require A to state her topics of interest t_i , as well as how much she is interested in them $w_{A,t_i} \in (0, 1]$, either explicitly or implicitly (e.g., by monitoring her tags' usage on the folksonomic website). In this case, when X publishes a message m , the value that A attaches to it could be computed as the average weight of those tags t_i that X attached to m and that appear in A 's profile. A more detailed discussion of folksonomy-based content dissemination can be found in [21]. To ease presentation, in the remainder of this paper we focus on the people-centric scenario. An example of a weighted social network is provided in Figure 1, with A being equally (and maximally) interested in users X , Y , Z and W ($w = 1$), and B being more interested in P ($w_{B,P} = 0.8$) than in Q ($w_{B,Q} = 0.2$).

Note that users who have many social connections and who are equally and maximally interested in all of them (e.g., like A in the above example), may risk driving, either unwillingly or selfishly, the whole content distribution network to work for them, to the expense of nodes like B , who may have less social connections and/or of different values. In our model, the value $v_{A,X}$ of a message is thus not simply the (explicit or implicit) weight $w_{A,X}$ in the

social graph, but such weight divided by $\sum_i w_{A,i}$ (that is, the sum of all weighted edges departing from A in the social graph). With reference to Figure 1, messages produced by X , Y , Z and W would thus have a value for A of $v_{A,X} = w_{A,X} / \sum_i w_{A,i} = 1 / (1 + 1 + 1 + 1) = 0.25$, while messages produced by P would have a value for B of $v_{B,P} = 0.8 / (0.8 + 0.2) = 0.8$. This processing aims to guarantee a fair share of network resources are used in support of every single participants; protecting against malicious and adversarial behaviours is outside the scope of this work. We assume that, at the time of publication, the publisher has enough knowledge of the geographically-relevant social network (who is interested in whom and how much) so to calculate these values. This is common to many of the second and third generation DTN protocols previously reviewed (Section II), where information about the social network is exchanged during periods of co-location.

B. Challenges

To better appreciate the challenge we tackle, let us consider the following scenario, grounded on the previous model and example (Figure 1). An intermediary node C is carrying two messages in its buffer (which we also refer to as queue): the first m_1 is a message produced by P and destined to B , with value $v_{B,P} = 0.8$; the second m_2 is a message produced by X and destined to A , with value $v_{A,P} = 0.25$. Let us also assume that C resources are running out, so that it can only forward one more message in the current time period Δt (e.g., within the next day). What message should C forward?

Two alternative approaches could be followed: on one hand, we could let the *physical network* drive the forwarding step entirely. For example, if A is encountered first (or another intermediary node who is along the physical route from X to A), then message m_2 would be forwarded, at the expense of node B (and its higher-valued message m_1). In other words, first-encountered/first-forwarded protocols may cause messages of little value to use up the scarce resources available, at the expense of highly-valued messages; note that this is the approach used by state-of-the-art DTN protocols [8], [22], [2], [14]. On the other hand, we could let the *application layer* drive the forwarding step (i.e., highest-value/first-forwarded). In this case, node C would reserve its remaining forward allowance to m_1 ; however, node B (or another intermediary node who is along the physical route from P to B) may not be encountered for another couple of days, during which C could be re-charged and thus its forwarding allowance increased. Not forwarding m_2 when the opportunity raises may thus result in unnecessarily missed deliveries.

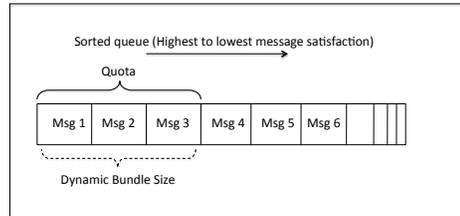
Nodes participating in a DTN must thus be able to allocate the scarce resources available for forwarding messages of high value, whilst also not compromising delivery due to missed opportunities. To do so, we present next a priority

scheduling approach, which reasons upon nodes’ mobility patterns (physical layer) *and* messages’ values (application layer) to achieve high end-user satisfaction without cutting back on delivery.

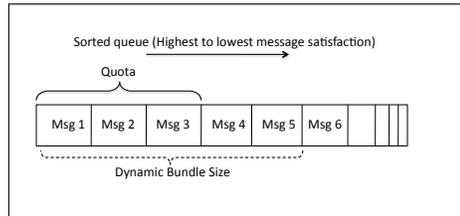
C. Approach

Our approach to priority scheduling in DTNs can be summarised as follow: each node participating in the content distribution network stores messages yet to be delivered in a queue, which is kept *sorted* by decreasing message value. The next message to be forwarded can be *any* belonging to the head of the queue; such head does not simply refer to the first *quota* messages, where *quota* represents the number of messages that the node can forward within a given time period Δt . Rather, it refers to all messages within a *dynamic bundle*, whose size varies depending on the probability of encountering the recipients (or next-hop carriers) of the stored messages. In other words, the application layer provides information (i.e., message values) to *sort* messages in order of priority, thus guaranteeing faster processing for high priority ones; the physical layer provides information (i.e., probability of physical encounters) to dynamically *adapt* the number of messages that are currently scheduled for forwarding, in an attempt to minimise the risk of wasting resources because of missed opportunities. We assume this prioritisation scheme to be deployed on top of an existing human-based DTN routing protocol (e.g., [8], [14], [2]), whereby past node encounters are logged and processed to compute these probabilities. This approach draws inspiration from the TCP flow control mechanism: while TCP uses a *sliding window* to adjust the transmission rate of packets, based on the observed drop rate, we use a *dynamic bundle* to adjust the scheduling of messages, based on the observed encounter regularity.

Before formalising the approach, we illustrate, with an example, the behaviour of the dynamic bundle. First, let us consider a scenario where nodes are connected at all times; let us also assume that, in any given time period Δt , the maximum number of messages a node can forward is $quota = 3$. In this case, because of stable network connection, the probability of delivering any message to its intended recipient is always 1; messages can thus be scheduled to be sent in the very same order they appear in the sorted queue, and the dynamic bundle would exactly refer to the top 3 messages, as shown in Figure 2a ($bundle\ size = quota$). Let us now consider the case of a human DTN, where connection between nodes is opportunistic, yet predictable [17], [18]. In this case, setting the bundle size to equal the node’s quota could be a waste: with reference to Figure 2b, the node would not attempt to deliver *Msg 4* before *Msg 1*, even if an encounter occurred that would enable that. While this guarantees that there will be enough spare resources to deliver higher-valued message *Msg 1* when the relevant encounter occurs, it may also result in an unnecessary missed



(a) Fully Connected Network



(b) Human Network

Figure 2: Dynamic Bundle

opportunity, for example, if the node’s next encounter with the recipient of *Msg 1* is unlikely to occur within this Δt .

In our approach, we thus define the behaviour of the dynamic bundle as follow. Whenever a node inserts a message m in its own queue (sorted by message value), it computes the probability of encountering the message next-hop recipient within Δt ; this probability $Prob_m$ is estimated based on the encounter regularity as monitored by the underlying DTN routing protocol, and used to dynamically adjust the bundle size as:

$$bundle.size = \begin{cases} quota, & \text{if } queue.size() \leq quota \\ n & \text{otherwise} \end{cases} \quad (1)$$

with

$$n = \left\{ \#m_i \mid \sum_{i=1}^{queue.size()} Prob_{m_i} \leq quota \right\}$$

In other words, if the queue currently stores fewer messages than the the quota allows, all of them can be scheduled for forwarding. Otherwise, the bundle size is set to be equal to the maximum number of messages (n) for which the sum of the probability of them being forwarded within Δt does not exceed the *quota*. Let us look back at Figure 2b, and assume the encounter probabilities for the messages in the queue to be: $Prob_{Msg\ 1} = 0.2$, $Prob_{Msg\ 2} = 0.6$, $Prob_{Msg\ 3} = 0.5$, $Prob_{Msg\ 4} = 0.8$, $Prob_{Msg\ 5} = 0.7$, $Prob_{Msg\ 6} = 0.9$. In this case, setting $bundle.size = quota = 3$ is likely to result in missed opportunities, especially because the message at the top of the queue has very little chance of being delivered within this Δt , at the expense of messages *Msg 4* and *Msg 5*, whose value is lower (they are further down in

the queue) but whose delivery probability is very high. Our approach would thus set the bundle size to include the top $n = 5$ messages, whose aggregated delivery probability does not exceed *quota* (i.e., $0.2 + 0.6 + 0.5 + 0.8 + 0.7 \leq 3$). This does not mean that more messages will be forwarded than what *quota* allows; rather, it means that the *quota* messages to be forwarded in the current Δt can be *any* of the n at the head of the queue, with $n \geq \textit{quota}$.

No matter what prediction technique is used by the underlying DTN routing protocol to estimate encounter probability, human mobility carries an inevitable degree of uncertainty with it. In case of high prediction error, the sizing of the dynamic bundle defined by formula 1 could be either too cautious (when actual encounters happen less frequently than predicted, causing missed opportunities), or too aggressive (when actual encounters happen sooner than expected, causing more important messages not to be forwarded because of resources being drained on less important ones). To cater for this uncertainty, the size n of the bundle could be set to increase so to include all messages whose aggregated delivery probability is less or equal to $\alpha * \textit{quota}$, with α being closer to 1 when predictions are highly accurate, $\alpha \in (0, 1)$ when the probabilities tend to underestimate the frequency of actual encounters (so that the forwarding bundle must be reduced), and $\alpha > 1$ when the probabilities tend to overestimate the encounter frequency (so that the forwarding bundle must be increased to avoid missed opportunities). A self-monitoring component could be added to the underlying routing protocol, to dynamically assess the accuracy of the prediction scheme, and thus adjust α accordingly; we leave this self-monitoring/self-adaptive behaviour open for future research, and we set $\alpha = 1$ in our evaluation section. To cater for the inevitable uncertainty characterising human mobility, we adopt a simpler heuristic instead, which does not require changes/additions to the underlying routing protocol: each node maintains information about what proportion of messages it has already sent in the last Δt with respect to the set *quota*: if such proportion is below a given threshold (what we call *loaded boundary*), we let the bundle size n grow as per formula 1; once the percentage of sent messages reaches the loaded boundary (i.e., when getting close to the *quota* for the current time period), we set back the bundle size to *quota*, thus favouring a more cautious behaviour over an aggressive one. In the next section, we study the impact that different values of the loaded boundary have on the satisfaction and delivery achieved.

IV. EVALUATION

In order to evaluate our priority scheduling approach, we had to implement it on top of an existing human DTN routing protocol. Among the protocols reviewed in Section II, we chose CoHabit [14] for the following two reasons: first, it selects message carriers, from source to

destination, based on the observed regularity of encounters; such regularity measure can be directly used by our approach when dynamically setting bundle size, to easily estimate encounter probabilities. Second, it is one of the very few approaches to drop the assumption of infinite resources, and to provide a load-balancing scheme that limits the number of messages that can be forwarded within a time period, by explicitly reasoning in terms of available resources (battery in particular). We use this limit to set the *quota* parameter that our prioritisation scheme relies on. Note that, in this work, we are not interested in assessing the performance of the underlying routing protocol; rather, we aim to quantify the gains that adding priority scheduling afford. Evaluation has been conducted by means of simulation: we thus first describe our simulation settings in terms of metrics, datasets and parameters (Section IV-A), before reporting the obtained results (Section IV-B).

A. Simulation Settings

Metrics. The goal of our prioritisation scheme is to guarantee enough resources are available to forward more important/valued messages, without compromising overall delivery, due to missed opportunities. In our experiments, we have thus measured: *satisfaction gain*, that is, the difference between the average value v of all messages delivered using priority scheduling on top of CoHabit, and using CoHabit (that is, a first-encounter/first-forwarded approach) alone; and *delivery gain*, that is, the ratio of the number of messages delivered with and without priority scheduling on top of CoHabit. Both metrics have been computed based on network-wide measurements.

Datasets and Parameters. In order to evaluate our work, we required two types of datasets: one providing human mobility traces (to simulate encounters), and one providing users' social networks (to determine who is interested in receiving content from whom and to what extent). We discuss the datasets we have selected, and how we have overlaid them, next.

- *Mobility Traces* - we have experimented with two mobility datasets of different topological properties: the *Reality Mining* dataset [23], and a *vehicular dataset* of cabs in San Francisco [24]. The former contains colocation information from 96 subjects at the MIT campus, over the course of the 2004-2005 academic year, to whom Bluetooth-enabled Nokia 6600 phones were given; colocation information was collected via frequent (5 minute) Bluetooth device discoveries. In our experiments, we extracted five months of colocation data, from September to February; we used the first five weeks of these traces as the training period required by the underlying protocol to learn regularities of encounters; the remaining period was then the actual test period during which nodes created and shared content. The MIT dataset has been widely used to evaluate

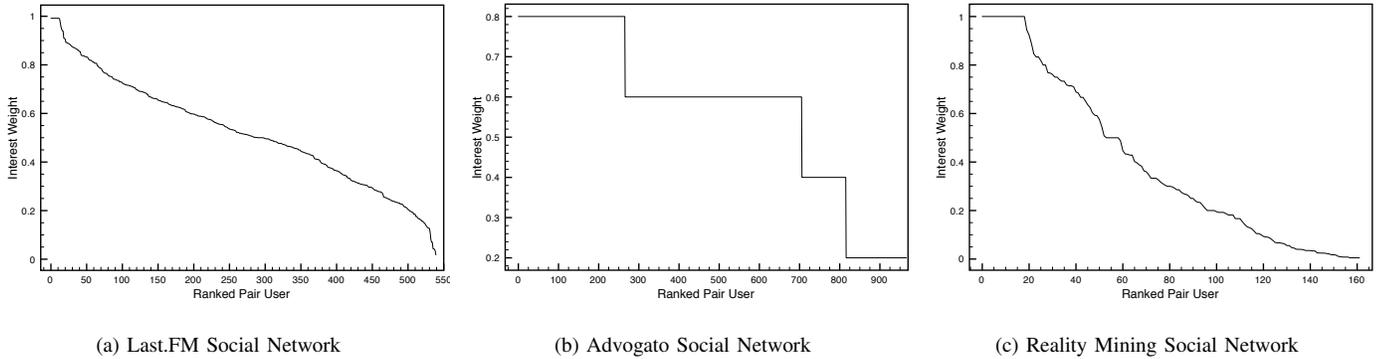


Figure 3: Weight Distribution of the Selected Social Networks

DTN protocols [25]; while being representative of some DTN settings (i.e., university campus), the sparsity of its traces and their very high inter-contact time is not representative of DTN *urban* settings, where nodes are much more frequently connected (as in a MANET), but with short contact time (as in a DTN) [26]. To assess priority scheduling also in these scenarios, our second mobility dataset contains GPS traces recorded by 500 cabs, logged every 10 seconds, over a period of 21 days, in the San Francisco Bay Area. We sampled a subset of 100 users from these traces, making sure the original topological properties of the traces, such as the inter-contact time distribution, were preserved.

- *Social Network* - to model who is interested in receiving content from whom, and to what extent, we have experimented with three distinct social networks, namely Last.FM, Advogato, and Reality Mining. *Last.fm* (<http://www.last.fm>) is a Web 2.0 music social networking website, where users explicitly declare who their social connections are. To sample this dataset, we first gathered 10,000 Last.fm users with a breadth-first search using the Audioscrobbler Web Service (<http://www.audioscrobbler.net/>). We then sampled a (connected) sub-graph of 100 users; their in-degree distribution highlighted the long-tailed degree distribution typical of human social networks. Note that social links have no weight in Last.fm; to obtain a *weighted* social graph, for each explicitly-declared link between users X and Y , we computed the weight $w_{X,Y} \in [0, 1]$ as the cosine similarity between the vectors representing these users' top-50 most listened artists (that is, the more similar their musical tastes are, the higher the weight of the connection). *Advogato* (<http://www.advogato.org/>) is a community discussion board for free software developers; social links between developers are self-reported, and their weight can take one of the following discrete values: observer ($w = 0.2$), apprentice ($w = 0.4$),

journeyer ($w = 0.6$), and master ($w = 0.8$). For our experiments, we extracted a sample of 100 users, once again making sure to preserve the degree distribution of links amongst users. Finally, the last social network was implicitly extracted from the *Reality Mining* dataset; apart from colocation information, this dataset logged voice calls and text messages exchanged between the participants in the study. Similarly to [27], we have used this information to extract an implicit social network whereby a link from user A to user B exists if A sent a text message or made a phone call to B ; we have then associated a weight to each such link based on the normalised number of calls/texts user A had initiated (e.g., if A called B five times, C twice and D three times, then $w_{A,B} = 5/10 = 0.5$, $w_{A,C} = 2/10 = 0.2$, and $w_{A,D} = 3/10 = 0.3$). To highlight the different properties of these datasets, Figure 3 plots the ordered distribution of link weights between each users' pair that exists in the social graph: as shown by the span of the x axis, Reality Mining (Figure 3c) is by far the least connected network, with approximately 160 edges, as opposed to around 600 for Last.fm (Figure 3a) and 900 for Advogato (Figure 3b), for networks of equal size (≈ 100 users). The distribution of interest weights is shown on the y axis: as expected, there exists a small subset of highly valued connections, and a much larger number of lesser valued ones; we expect our priority scheme to take advantage of these differences to forward the messages from the most valued connections first, thus increasing overall network satisfaction.

For completeness, we evaluated our work across all combinations of these social networks and mobility traces. Overlaying between the two has been done multiple times at random, with the exception of the Reality Mining dataset, where there exists a direct (non random) mapping of users between movement and social graph.

The experiments unfold in the very same way as discussed in CoHabit [14]: after an initial training period, during which

colocations are logged to learn movement regularity, and information about a user’s social network is disseminated, each node in the network starts publishing messages for its intended recipients (i.e., connections in the social graph). The rate of publication is set according to users’ activity in the Digg (<http://www.digg.com>) content bookmarking website; the value of a message is dictated by the weight in the graph of the social network being used. *quota* is set to be 50 messages over $\Delta t = 5$ days; all other protocol-specific parameters are set as in [14]. Each simulation has been repeated 5 times, and average results are presented.

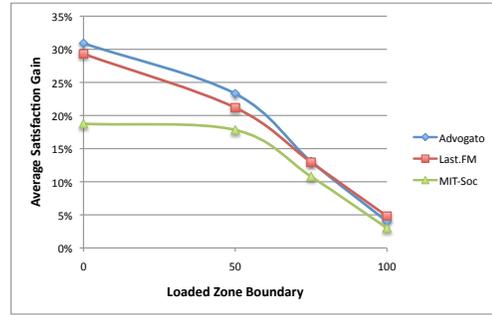
B. Results

We now report the results obtained, first in terms of the satisfaction gain and then in terms of delivery gain, when our priority scheme is deployed on top on an existing DTN routing protocol (namely, CoHabit). More precisely, we highlight the effectiveness of our prioritisation scheme in bringing higher overall network satisfaction, while not compromising network delivery.

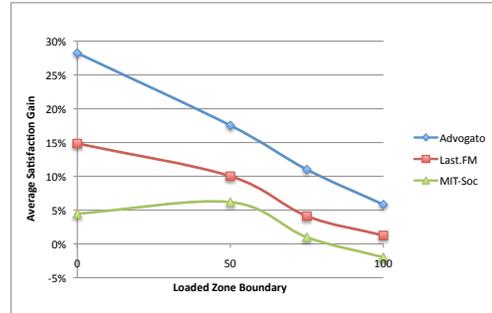
Satisfaction Gain. Figure 4 depicts the gain in overall network satisfaction, while varying the loaded zone boundary parameter: a value of 100% means that such boundary (and the restrictions it entails on bundle size) is not used, and the bundle size is set to grow as per formula 1. Conversely, a value of 0% means the boundary is always set, and the bundle size is restricted to be equal to *quota* at all times; an intermediary value of 50% means that the bundle size is set to grow as per formula 1 up until 50% of *quota* messages have been forwarded in the last Δt ; after that, a more cautious behaviour is started, whereby the bundle size is fixed to *quota*. These three values have been chosen for the following reasons: a loaded boundary of 0 lets us observe the impact of reasoning on message value (application layer), whilst avoiding the susceptibility to encounter prediction error (physical layer) and its consequences on bundle size. We expect such setting to produce the highest satisfaction, though risking to lower delivery due to missed opportunities. At the opposite extreme, a loaded boundary of 100 almost nullifies the use of priority scheduling, with encounters now driving the order of forwarding. Finally, a boundary of 50 represents a trade-off between the two, letting us observe the interplay between message values and network topology of the underlying mobility traces.

Figure 4a presents results when using the Cabs mobility traces (and the three different social networks overlaid on top), while Figure 4b presents results for the Reality Mining mobility traces (again with the three social networks overlaid on top). The following three observations can be made:

- *Impact of Loaded Boundary* - as expected, the lower the value, the higher the satisfaction gain; this is because only *quota* messages can be delivered at any Δt , and while CoHabit chooses to deliver based on a



(a) Cabs



(b) Reality Mining

Figure 4: Satisfaction Gain

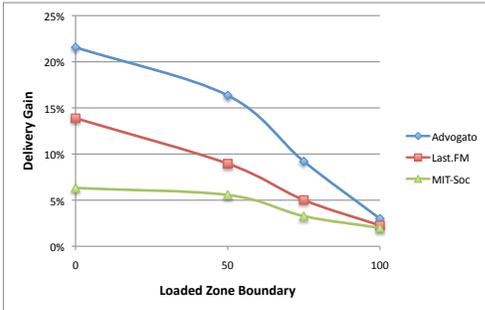
first-encountered/first-forwarded manner, our scheduler prioritises delivery of the *bundle size = quota* most valued messages. With a boundary of 50, the gain is still very high (e.g., approximately 20% across all social networks for Cabs traces). However, for boundary of 100, the gain over CoHabit tends towards zero: this is because *bundle size* \gg *quota*, so that encounters, and not message values, are now driving the forwarding.

- *Impact of Mobility Traces* - for each value of loaded boundary and for each social network, the satisfaction gain is much higher for Cabs than for Reality Mining mobility traces. This is because the former has much more frequent encounters (lower inter-contact time): higher encounter probability means smaller bundle sizes and higher delivery of the most valued messages, thus pushing network satisfaction up.
- *Impact of Social Network* - finally, for each loaded boundary value and mobility traces, we observe that satisfaction gain is highest for the Advogato social network, followed by Last.fm and finally Reality Mining. The reason for this can be best explained by looking at Figure 3: Advogato is the most connected social network, with approximately 25% of its social links having maximum weight. This means the priority scheduler can *choose* to first forward high valued messages in the queue, with neat gain over CoHabit. For Reality Mining, instead, connectivity in the social

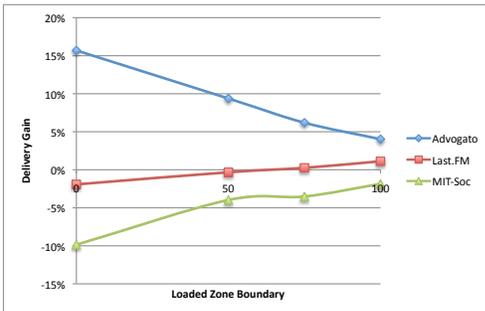
graph is much lower, and of lower values too; this means there is often little the scheduler can do in terms of prioritisation, with consequently lower gain over CoHabit.

Satisfaction gain has been computed as average value of *delivered* messages. We now turn our attention to study the impact of priority scheduling on delivery rate.

Delivery Gain. Figure 5 depicts the gain in overall network delivery, while varying the loaded zone boundary parameter. In particular, Figure 5a presents results when using the Cabs mobility traces (and the three different social networks overlayed on top), while Figure 5b presents results for the Reality Mining mobility traces (again with the three social networks overlayed on top). The following observation can be made: delivery improves across all values of loaded boundary and social networks when deploying the Cab traces. A gain is also observed (though smaller) for the Advogato social network on top of Reality Mining mobility, while no gain nor loss is observed for Last.fm. Across *all* these settings, when looking at both satisfaction and delivery, we can thus conclude that priority scheduling brings an unquestionable benefit, with very low values of loaded boundary being best for Cabs-like traces (small inter-contact time), and medium values for Reality Mining-like traces (high inter-contact time). The single setting where priority scheduling actually causes a loss in delivery is when using Reality Mining mobility traces and social network: a



(a) Cabs



(b) Reality Mining

Figure 5: Delivery Gain

combination of high inter-contact time (i.e., few encounters, thus few forwarding), sparse social network, and cautious forwarding behaviour (boundary of 0) produces a 10% loss in delivery, with only a 5% gain in satisfaction (Figure 4b). Even a more aggressive behaviour (boundary of 100) does not really pay off, with both satisfaction and delivery being on a par with CoHabit.

We can thus conclude that, when opportunities for delivery are particular scarce, *and* the number of high valued message small, a fully opportunistic approach (first-encountered/first-delivered) is better suited than a priority one; on the higher hand, in scenarios that give scope to prioritisation (with more opportunities for delivery and/or higher differentiation in message values), our prioritisation approach brings neat benefits, both in terms of satisfaction and delivery.

As one final insight into the performed experiments, we have measured the delivery time of those messages that both CoHabit and CoHabit with priority scheduling bring to destination. We have then computed the average gain (conversely, delay) in delivery time that priority scheduling brings for high value (conversely, low value) messages. As expected (Figure 6), high value messages ($v \in [0.8 - 0.9]$ on the x axis) are delivered faster (positive value on the y axis), to the expense of low value messages ($v \in [0.1 - 0.35]$ on the x axis), which are now delivered slower (negative value on the y axis). Note that the gain (top-left corner) and loss (bottom-right corner) are of the same order of magnitude (approximately 13' - 15').

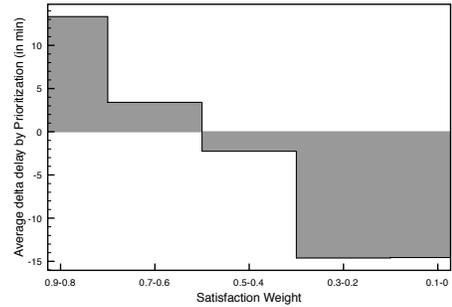


Figure 6: Delay Distribution

V. CONCLUSION AND FUTURE WORK

In this paper, we have presented a priority scheduling approach for participatory DTNs, whereby messages are being forwarded based on a combination of the likelihood of future encounters (physical layer) and the value that recipients attach to such messages (e.g., based on who produced the message). We have implemented this priority scheduling on top of an existing DTN protocol (namely, CoHabit), and evaluated the gain it entails in both end-user satisfaction and delivery, over a variety of real mobility traces and message values distributions.

There are two open directions of research we intend to pursue: in the short-term, we aim to develop a self-monitoring technique that assesses the predictability of the mobility traces, thus dynamically adapting the sizing of the bundle based on live feedback from the physical network. In the longer-term, we intend to investigate content dissemination protocols that are robust against malicious behaviours.

REFERENCES

- [1] Wi-Fi Alliance, “Wi-Fi CERTIFIED Wi-Fi Direct,” *Industry Press Releases*, October 2010.
- [2] P. Costa, C. Mascolo, and M. Musolesi, “Socially-aware Routing for Publish-Subscribe in Delay-tolerant Mobile Ad Hoc Networks,” *IEEE Journal On Selected Areas In Communications (JSAC)*, vol. 26, no. 5, pp. 748 – 760, June 2008.
- [3] E. Yoneki, P. Hui, S. Chan, and J. Crowcroft, “A Socio-Aware Overlay for Publish/Subscribe Communication in Delay Tolerant Networks,” in *Proc. of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems*. 2007, pp. 225–234.
- [4] A. Lindgren, A. Doria, and O. Schelén, “Probabilistic Routing in Intermittently Connected Networks,” *Service Assurance with Partial and Intermittent Resources*, pp. 239–254, 2004.
- [5] J. Leguay, A. Lindgren, J. Scott, T. Friedman, and J. Crowcroft, “Opportunistic Content Distribution in an Urban Setting,” in *Proc. of CHANTS*. 2006, pp. 205–212.
- [6] P. Hui, J. Crowcroft, and E. Yoneki, “Bubble Rap: Social-based Forwarding in Delay Tolerant Networks,” in *Proc. of 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. 2008, pp. 241–250.
- [7] Q. Yuan, I. Cardei, and J. Wu, “Predict and Relay: an Efficient Routing in Disruption-Tolerant Networks” in *Proc. of the 10th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. 2009, pp. 95–104.
- [8] A. J. Mashhadi, S. B. Mokhtar, and L. Capra, “Habit: Leveraging Human Mobility and Social Network for Efficient Content Dissemination in Delay Tolerant Networks,” in *IEEE WOWMOM*, June 2009, pp. 1–6.
- [9] J. A. Paradiso and T. Starner, “Energy Scavenging for Mobile and Wireless Electronics,” *IEEE Pervasive Computing*, vol. 4, no. 1, pp. 18–27, 2005.
- [10] W. Wang, V. Srinivasan, and M. Motani, “Adaptive Contact Probing Mechanisms for Delay Tolerant Applications,” in *Proc. of the 13th Annual ACM International Conference on Mobile Computing and Networking*. 2007, pp. 230–241.
- [11] H. Jun, M. Ammar, M. Corner, and E. Zegura, “Hierarchical Power Management in Disruption Tolerant Networks with Traffic-aware Optimization”, in *Proc. of CHANTS*. 2006, pp. 245–252.
- [12] A. Balasubramanian, B. Levine, and A. Venkataramani, “DTN Routing as a Resource Allocation Problem,” in *Proc. of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 2007, vol. 37, pp. 373–384.
- [13] J. Pujol, A. Toledo, and P. Rodriguez, “Fair Routing in Delay Tolerant Networks,” in *Proc. of IEEE INFOCOM*, April 2009, pp. 837–845.
- [14] A. J. Mashhadi, S. B. Mokhtar, and L. Capra, “Fair Content Dissemination in Participatory DTNs,” UCL Technical Report RN/10/07. 2010.
- [15] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, “A Performance Comparison of Multi-hop Wireless Ad Hoc Network Routing Protocols,” in *Proceedings MobiCom*, 1998, pp. 85–97.
- [16] A. Vahdat and D. Becker, “Epidemic Routing for Partially-Connected Ad Hoc Networks,” Technical Report, 2000.
- [17] I. Rhee, M. Shin, S. Hong, K. Lee, and S. Chong, “On the Levy-Walk Nature of Human Mobility,” in *Proc. of 27th IEEE INFOCOM*. 2008, pp. 924–932.
- [18] C. Song, Z. Qu, N. Blumm, and A. Barabasi, “Limits of Predictability in Human Mobility,” *Science*, vol. 327, no. 5968, p. 1018, 2010.
- [19] A. Mtibaa, A. Chaintreau, J. LeBrun, E. Oliver, A. Pietilainen, and C. Diot, “Are you moved by your social network application?” in *Proc. of the 1st Workshop on Online Social Networks*. ACM, 2008, pp. 67–72.
- [20] E. M. Daly and M. Haahr, “Social Network Analysis for Routing in Disconnected Delay-Tolerant MANETs,” in *Proc. of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. 2007, pp. 32–40.
- [21] G. Lo Giusto, A. J. Mashhadi, and L. Capra, “Folksonomy-based Reasoning for Content Dissemination in Mobile Settings,” in *Proc. of CHANTS*, 2010, pp. 39–46.
- [22] M. Musolesi and C. Mascolo, “CAR: Context-aware Adaptive Routing for Delay-Tolerant Mobile Networks,” *IEEE Transactions on Mobile Computing*, vol. 8, no. 2, pp. 246–260, 2008.
- [23] N. Eagle and A. S. Pentland, “Reality Mining: Sensing Complex Social Systems,” *Personal Ubiquitous Computing*, vol. 10, no. 4, pp. 255–268, 2006.
- [24] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, “A Parsimonious Model of Mobile Partitioned Networks with Clustering,” in *Proc. of COMSNETS*. 2009.
- [25] Z. Zhang, “Routing in Intermittently Connected Mobile Ad Hoc Networks and Delay Tolerant Networks: Overview and Challenges,” *IEEE Communications Surveys & Tutorials*, vol. 8, no. 1, pp. 24–37, 2007.
- [26] J. Whitbeck, and V. Conan, “HYMAD: Hybrid DTN-MANET Routing for Dense and Highly Dynamic Wireless Networks,” in *IEEE Workshop on Autonomic and Opportunistic Communications*, 2009.
- [27] A. Lindgren, C. Diot, and J. Scott, “Impact of Communication Infrastructure on Forwarding in Pocket Switched Networks,” in *Proc. of CHANTS*. ACM, 2006, pp. 261–268.