Exploiting user interest similarity and social links for micro-blog forwarding in mobile opportunistic networks

S.M. Allen[a], M.J. Chorley[a], G.B. Colombo[a], E. Jaho[b], M. Karaliopoulos[b], I. Stavrakakis[b], R.M. Whitaker[a]

[a]*School of Computer Science & Informatics,Cardiff University,Cardiff, UK*
{*stuart.m.allen, m.j.chorley, g.colombo, r.m.whitaker*}*@cs.cardiff.ac.uk*
[b]*Department of Informatics and Telecommunications, National & Kapodistrian University of Athens, Athens, Greece*
{*ejaho,mkaralio,ioannis*}*@di.uoa.gr*

Abstract

Micro-blogging services have recently been experiencing increasing success among Web users. Different to traditional online social applications, micro-blogs are lightweight, require small cognitive effort and help share real-time information about personal activities and interests. In this article we explore scalable pushing protocols that are particularly suited to the delivery of this type of service in a mobile pervasive environment. Here, micro-blog updates are generated and carried by mobile (smart-phone type) devices and are exchanged through opportunistic encounters. We enhance primitive push mechanisms using social information concerning the interests of network nodes as well as the frequency of encounters with them. This information is collected and shared dynamically, as nodes initially encounter each other and exchange their preferences, and directs the forwarding of micro-blog updates across the network. Also incorporated is the spatiotemporal scope of the updates, which is only partially considered in current Internet services.

We introduce several new protocol variants that differentiate the forwarding strategy towards interest-similar and frequently encountered nodes, as well as the amount of updates forwarded upon each encounter. In all cases, the proposed scheme outperforms the basic flooding dissemination mechanism in delivering high numbers of micro-blog updates to the nodes interested in them. Our extensive evaluation highlights how use can be made of different amounts of social information to trade performance with complexity and computational effort. However, hard performance bounds appear to be set by the level of coincidence between interest-similar node communities and meeting groups emerging due to the mobility patterns of the nodes.

Keywords: micro-blogging, social networks, pervasive, mobile, content dissemination.

1. Introduction

Online micro-blogging services have become very popular in recent years. The basic idea behind them is to allow users to post short messages and automatically receive updates from other specific users who they decide to 'follow'. Although the 'follower' relationship is reminiscent of traditional online social networks

(OSNs), it is also substantially dissimilar from a typical online friendship in that links are essentially uni-directional and may not be reciprocated. Also, users do not necessarily receive any kind of information about their followers and their interests, while followers can be blocked from receiving updates if so desired.

Thanks to user-originated features such as 'retweeting' (the forwarding of a received tweet, allowing its spread far beyond the set of followers of its original source), micro-blogging has become an effective tool for information diffusion, similar to news media services [1]. Additional user-originated features include a form of tagging (hash-tag) that allows categorisation of updates by topic. Micro-blogging has found applicability in emergency scenarios (updates during riots in Kenya, Egypt, Iran and Libya), and facilitated information dissemination at institutional level or in critical situations (help during large-scale emergencies, live updates to track traffic delays), thus serving as a powerful instrument of cooperation [2, 3]. It has also been recommended as an effective alternative for reducing overload in working environments [4].

Capabilities such as targeting a specific user in a post (reply), or sending direct messages to users suggest a (weak) definition of 'friendship' between users that have participated in a given number of these more direct interactions. Alternative interpretations consider 'friends' those users for whom the follower relation is reciprocated [1]. According to an exploratory study of Twitter usage [5], it is the compulsory brevity of the updates that further allows the reader to effectively filter large numbers of messages. This feature reduces the cognitive threshold for the writer to decide to share and the burden for the reader to process all updates. Because of their particular requirements in terms of message size and their purpose to inform, warn, share and offer opinions, micro-blogs have been often compared to the concept of 'utterance' in linguistics [6, 7, 3, 8].

Although micro-blogging has essentially been thought of as an online service, the particular structure of the induced followers' networks [9] makes them an ideal mechanism for the rapid dissemination of information amongst ad-hoc social communities. The application of these services to the mobile domain suggests opportunities for sharing micro-blog posts directly among local devices that can send and receive content while taking into account the local or temporal context.

In this work, we explore scalable decentralised push-based protocols for micro-blogging using mobile devices in pervasive opportunistic environments [10]. According to our scenario, low payload micro-blogs (utterances) are generated by the devices (nodes), stored directly in their memory, and opportunistically exchanged upon their pairwise interactions. Our work extends the basic flooding concepts introduced in [8] by exploring in detail the role of similarity of interest between users. As a consequence of the peculiar nature of micro-blogging services, where forwarding an update does not imply any direct knowledge of the current status of followers, a user forwarding an utterance will have no knowledge about their follower's stored micro-blogs or preference for particular content. A push-based strategy such as this must therefore follow some form of (pure or controlled) flooding strategy. However, to enhance these strategies' performance by attempting to reduce the number of irrelevant utterances delivered to users, we allow them to store limited

social context about a subset of users and use it to direct the forwarding of micro-blogs to encountered nodes.

The contribution we make is to explore different ways to use friendship, interest similarity and familiarity between nodes to better suit various possible real-life scenarios. This is important because it allows protocols to be adaptive to social content. In the baseline scenario, friendship links form between individual nodes having similar interests. Different interest similarity thresholds can modulate the selectiveness of nodes in forming such relationships. Information about interest profiles is dynamically shared within the resulting social groups of friends and used to prioritise the dissemination of content among them. Note that basing the social group exclusively on similar interests aims to reduce the computational effort for storing, processing, and selecting updates not closely related to a node's own interests. At a second level, nodes may keep account of nodes they encounter frequently (*familiar* nodes). The relationships with these nodes, which do not necessarily share similar interests, can be seen as a further (weaker) form of friendship that can be exploited to disseminate content on behalf of other 'familiar' individuals.

We do not aim to deliver all relevant content to all nodes that may wish to receive it. Instead, we aim to deliver content that is interesting to a node given some spatial or temporal context. We do not guarantee that users will always receive all the interesting content that may be in the system for them, but aim to ensure that the content they do receive is of the most interest. We aim therefore to reduce the dissemination of irrelevant content, thus removing the need for users to filter the content they receive. In order to quantitatively assess the system's ability to meet these aims we use two metrics describing information retrieval quality, precision and recall.

Although the protocol could be extended to fuzzier classifications, we assume in this work that each utterance is characterised by a well-defined topic (tag). Nodes gain positive utility when receiving an update that matches one of their individual interests (tags). The obtained utility also accounts for the spatial and temporal validity of micro-blogs. The local aspect, in particular, even if it is not originally considered in on-line services, is particularly suited to mobile pervasive scenarios.

We assess different strategies for selection and pushing of utterances and find that the most successful strategies take into account not only the individual interest profiles of friend nodes but also use a 'community profile' to push items to nodes that are not friends, but may be familiar due to repeated interactions. A performance tradeoff must be made between using all non-friend nodes and using only those that are familiar to us. Further tests reveal that considering the spatial and temporal validity of content has an important impact on the system performance, and that pushing more than one utterance per encounter may deliver better performance in terms of recall, but worsen the performance in terms of precision. Finally we examine the effect of basing the friend set of a node on both familiarity and similarity rather than just similarity.

The remainder of the document is organised as follows. Section 2 describes the main components of the proposed micro-blog dissemination protocol, including the social information maintained by nodes and

the criteria for forwarding micro-blogs to encountered nodes. Section 3 presents the methodology and experimentation scenarios we have devised for evaluating the protocol and revealing its main tradeoffs. Our experimental evaluation is structured into five sets of experiments, whose results are reported in Section 4. Related research and the differentiation elements of our work are presented in Section 6, while Section 7 summarizes our conclusions about the protocol performance and proposes future work items.

2. Micro-blog Dissemination Protocol

This section introduces our protocol for opportunistic dissemination of 'micro-blogs'. The protocol is executed by mobile agents that interact opportunistically in the physical space. As the agents come within range of one another, they establish connections and exchange micro-blogs. Although our initial motivation has been to develop a protocol for adapting and testing current online micro-blogging services in a mobile opportunistic environment, the protocol can involve generic mobile device nodes (*i.e.*, not necessarily human users) that can interact automatically using wireless technologies.

Our protocol is *push*-based. Micro-blogs (also described as utterances) are forwarded to encountered agents on the expectation that they carry some value for them, rather than because they are explicitly requested by them as a *pull* model would dictate. Our objective has been to come up with a lightweight protocol that circumvents the overheads of information discovery/advertisement operations, which are not well suited to opportunistic environments. At the same time, the protocol draws on information about the users' interests to control and direct the amount of floating information in the network. Nodes exchange information about their preferences upon first contact and so iteratively build *interest profiles* for the other nodes in the network. These profiles help them form implicit 'groups of friends', wherein utterances are forwarded with higher care than among non-friend nodes. Only friend's interest profiles are stored in the long term, increasing the efficiency and scalability of the system. Ideally, the generated utterances are only forwarded to nodes that are interested and actually 'consume' them, making minimum use of the storage and battery resources of nodes.

As we are considering a decentralised push based system, we assume nodes use only short range communication to transfer information. No assumption is made on the availability of a cellular data network, making the protocol ideal for use in areas of low or no signal, such as on underground transport systems. As such it is also suitable for use when no centralised data connection or sufficient infrastructure is available, such as at a festival or a large sporting event, or when roaming in a foreign country where data services may be costly. Moreover, it is suitable for use with any type of device, not just those mobile phones with a data connection to a cellular network. Most tweets in such a system are likely to have local interest, so using a decentralised system allows only local resources to be consumed in the consumption and forwarding of data. Using the cellular data network would require fundamental changes to the architecture of the system, and

may also have associated battery costs for the mobile devices.

The following paragraphs present the main protocol ingredients: the interest profiles capturing the nodes' interests in different topics and the method to form groups of friends out of these interest profiles; the utterances, the elementary data unit that is forwarded upon pairwise node encounters, and their specification; and, finally, the criteria that determine which utterance should be pushed to which node.

2.1. Interest profiles and sets of friends

We assume that the interests of the network nodes can be coded into a finite well-defined global set $\mathcal{M}$ of distinct topics (tags). Each mobile node x is interested in a subset of these topics that form its tag set[1]. The interests of node x are captured in his/her *interest profile* and are formulated by an $M = |\mathcal{M}|$-dimensional vector $\mathbf{I}^x \in [0,1]^M$, where the relative interest of a node in a particular tag or topic may take any value between 0 and 1 (inclusive). Non-zero entries in $\mathbf{I}^x$ correspond to the tag set of node x, whereas zero entries denote lack of interest in the corresponding tag. Practically, the entry I_i^x reflects the rate at which a node x generates micro-blogs related to the i^{th} tag and/or its desire to receive information about the respective topic. The interest profile vector is normalized for each node with $\sum_{i=1}^{M} I_i^x = 1$.

We associate two sets of nodes with each node x; both are dynamically built as nodes opportunistically interact with each other. Each node therefore builds a distinct community of nodes with which it can interact, with the level of friendship determining the manner and level of interaction between the nodes.

set of familiar nodes Familiar nodes are those nodes that come into contact more frequently with x. We measure the contact frequency through the number of pairwise encounters: two nodes are noted as familiar if this number exceeds a certain threshold thr_F within a given time-window. In addition, for a node x we include some of the nodes that, despite meeting less frequently with x, are familiar with other nodes y that are themselves familiar with x. The complete procedure for determining those nodes is described in [12]. Note that we base the definition of the familiarity exclusively on the number of contacts and not on the duration, since the latter parameter is not influential in our simulation (connections are open and close within a single time step after nodes have forwarded (pushed) the selected items to the connected nodes. For each node in the network the group of nodes familiar to it constitutes its *familiarity set*.

set of friends Friend nodes are those that have 'similar' interests with node x. The interest similarity between two nodes can be measured through some measure of the similarity of their distribution of interests. To this end, we have adopted the 'proportional similarity(PS)' metric, sometimes referred to

[1]The use of a defined set can be seen as the natural extension to fuzzy classification of interests (see an adaptation of the concept of fuzzy set [11]) in which each tag may have a probability value of being associated to other tags that are similar in meaning but substantially different semantically.

as the 'Czekanowski index'. With the PS metric, the interest similarity $PS(x, y)$ between two nodes x and y, with interest distributions $\mathbf{I^x}$ and $\mathbf{I^y}$, equals [13]:

$$PS(x, y) = 1 - \frac{1}{2} \sum_{i=1}^{M} |I_i^x - I_i^y| \,. \tag{1}$$

The PS metric produces values in the interval [0,1] and is shown in [13] to satisfy all 11 criteria suggested as suitable for a measure of distributional similarity. A PS value of zero corresponds to two completely dissimilar nodes (nodes whose corresponding interest profiles have no tags in common with positive frequencies), whereas a value of one denotes the maximum degree of similarity (two nodes having identical interest profiles composed of the same tag set with exactly coincident non-null frequency values).

In this work the forwarding of data draws heavily on the similarity of nodes' interests. This reflects the concept that the ultimate goal for a node is the reading and disseminating of updates closely related to its own interests. Therefore, each node x in the network determines its *set of friends*, N_x as follows. When a node y encounters a node x it is added to its 'set of friends' N_x if $PS(x, y)$ exceeds some threshold thr_S.

$$N_x \doteq \{y | PS(x, y) > thr_S\}. \tag{2}$$

This set is dynamically built as nodes interact, *i.e.*, node pairs compute their pairwise PS values upon their first encounter. Out of the individual friend sets, we can then form the undirected *community graph* $G_c = (V, E_f)$, where V is the set of network nodes and E_f the set of friendship links between all node pairs.

2.2. Utterance definition

Utterances are defined as low-payload data (*e.g.*, short text messages) that are produced and stored by each network node. Contrary to online micro-blogging services, utterances are not pushed to a node's friends immediately after their generation but rather upon subsequent opportunistic contacts. Nodes generate a new utterance u over time in accordance with their interest profiles and annotate them with the respective tag t_u. Each node generates utterances according to a Poisson(λ) process with an average rate λ; altering the value of λ will change the average time between the generation of utterances, this provides a suitable estimate for the generation of messages within a system [14].

2.2.1. Utterance time and spatial validity

The utility of an utterance and the interest that other nodes have in receiving and reading it, is expected to decrease with its lifetime. Also, some of the utterances may have only local significance and rapidly lose

their importance when delivered outside the geographical area in which they were generated. To capture the potential spatiotemporal scope of utterances, we annotate them with both temporal and spatial scope attributes, as follows:

time expiry attribute, t_{exp} An utterance u remains useful up to time t^u_{exp} after its generation time, t^u_0. If $t^u_{exp} = \infty$, the utility of the utterance remains intact permanently.

local reach attribute, l_{max} An utterance has no value if received outside a given geographical area that includes its original generation place l^u_0 (its 'home' area).

Upon their generation, utterances are stamped with the quadruple $\{t_0, t_{exp}, l_0, l_{max}\}$. Only utterances that may provide some utility to the receiving node are selected for pushing during node encounters. In general, the utterance 'validity' could be any monotonically decreasing function $f(t - t^u_0, l - l^u_0)$, where t and $l = (l_x, l_y)$ are the current time and spatial node coordinates. Note that whereas the time validity of an utterance is lost once its lifetime exceeds t_{exp}, its spatial validity can be regained if the node that carries it moves back to the utterance's home area.

2.2.2. Utterance utility for receiving nodes

Only a certain proportion of the circulating utterances bear actual value for the nodes. The main goal of our proposed protocol is to maximise the average 'degree of satisfaction' for individual nodes during the dissemination of updates in the network. Formally, we introduce a utility function $U(u, t, l)$, which quantifies the satisfaction a user gets upon receiving an utterance u of tag t_u at time t and location l. The utility function captures two components. First, the utterance's relevance to the node's interests is considered. The better an utterance matches the interest profile of the receiving node, the more valuable it is. As discussed in Section 2.1, the interest of a node x in utterance u assigned to tag t_u is the $(t_u)^{th}$ entry of its interest profile vector $\mathbf{I^x}$. Secondly, the utterance validity is considered, as described in Section 2.2.1.

The overall utility gained by a node upon receiving the utterance u at time t and space l is given by:

$$U(u, t, l) = I^x_u * f(t - t^u_0, l - l^u_0). \tag{3}$$

In our experimentation in Section 4, we consider functions, $1 - u(t - t^u_0 - t^u_{exp})$ for the time validity and $1 - u(l - l^u_0 - l^u_{max})$ for the spatial validity components, respectively. Here $u(\cdot)$ is the step function and should not be confused with an utterance u. Thus the overall utility is:

$$U(u, t, l) = I^x_u * \Big(1 - u(t - t^u_0 - t^u_{exp})\Big)\Big(1 - u(l - l^u_0 - l^u_{max})\Big). \tag{4}$$

2.3. Definition of the push protocol

With respect to actual forwarding, each node x classifies all other network nodes into two categories based on its and other nodes' interest profiles: those belonging to its set of friends N_x and those that are not included in N_x. However, nodes do not need to exchange their own interest profiles, as defined in Section 2.1, but rather *push profiles* $\mathbf{P}^{xy}$, maintained by each of the nodes x for each member y of its 'set of friends' N_x.

2.3.1. Push profiles vs. interest profiles

Push profiles are $M-$dimensional vectors, whose elements P_i^{xy} determine the probability with which node x forwards utterances of tag i to the friend node y. When compared to the interest profiles, push profiles allow the forwarding decision to account for information beyond the interests of the encountered node. Nodes are then able to forward information that is not only relevant to the node they have encountered (y), but that may also be relevant to the nodes that y may subsequently meet. For example there may be cases where a node y in N_x is not itself very interested in utterances of certain tags, but some of its own friends are. We can then let the forwarding decision towards nodes in N_x take into account the interests of nodes at various distances k from x in the community graph G_c.

Formally, let $d(x, y)$ denote the minimum distance (hopcount) between x and y on G_c and $D_i(x) = \{y \in V : d(x, y) = i\}$ be the set of nodes at distance i from node x. Note that $D_0 = \{x\}$, $D_1 = N_x$, and, within this general definition, any individual node is included in only one of these node sets that are centered on node x. The *k-order push profile* of node $y \in N_x$ is given by:

$$\mathbf{P}^{xy}(k) = \sum_{i=0}^{k} \alpha_i \sum_{\tau \in D_i(y) - \{x\}} I_\tau$$

where $\{\alpha_i\}$ are constants determining the weight that the interests of the $i - hop$ neighbors have on the forwarding decision.

The parameter k introduces a tradeoff between the amount of information that a node should collect about the preference of interests of other nodes in the network and the effectiveness of its forwarding decisions. A value of $k = 0$ corresponds to a more shortsighted push profile that takes into account only the interests of the receiving node y; $k = 1$ includes in the push profile the interests of node y and its neighbours (two hops away overall); and $k = 2$ considers the profiles of all nodes up to three hops away from the selecting node on G_c.

Note that for $k > 0$ node y needs to communicate to x only its aggregate push vectors without the need to fully disclose the individual interest profiles of its own friends. The push profiles for each value of k are therefore composed such that:

$$\mathbf{P}^{xy}(0) \quad = \quad \alpha_0 \mathbf{I}_y \tag{5}$$

$$\mathbf{P}^{xy}(1) = \alpha_0 \mathbf{I}_y + \alpha_1 \sum_{\tau \in N_y - \{x\}} \mathbf{I}_\tau \quad (6)$$

$$\mathbf{P}^{xy}(2) = \alpha_0 \mathbf{I}_y + \alpha_1 \sum_{\tau \in N_y - \{x\}} \mathbf{I}_\tau + \alpha_2 \sum_{\tau \in Y - \{x\}} \mathbf{I}_\tau \quad (7)$$

where $Y = \cup_{\sigma \in N_y} N_\sigma - \{y\}$ and α_0, α_1 and α_2 are weights between zero and one.

In addition, each node computes its *community profile* as the sum of the push vectors over all members of its 'set of friends':

$$\mathbf{P}^x = \sum_{y \in N_x} \mathbf{P}^{xy} \quad (8)$$

The community profile vector is then normalized for each node with $\sum_{i=1}^{M} P_i^x = 1$.

Preliminary experiments have shown little difference when adjusting settings of the weights α_i. The values of k that optimise the performance seem to depend on the input data set used, but actual performance differences observed are not significant. In general, the more similar nodes are, the fewer hops are necessary to build a push profile covering all interests. In addition, values of $k > 1$ do not produce any performance improvement (this may be related to the partial applicability of the transitivity property to the concept of similarity). Therefore in the experiments presented in this paper we considered profiles formed only by a node and its direct neighbours in G (*i.e.*, $k = 1$ corresponding to weight values $\alpha_0 = \alpha_1 = 1$ and $\alpha_2 = 0$).

2.3.2. Selecting the utterance for forwarding

The push profiles defined in equations (5)-(7) are then used to select which utterances to forward during encounters with peers that belong to the same community (friends) and peers that do not (strangers).

Ideally a node would store in its cache only micro-blogs that is itself interested in (those he wants to forward and make other people read) and those that can be of some interest for its friends (similar to itself to some extent). However, since utterances are very low payload data, we assume that buffer space is not a concern and focus exclusively on the assessment of different utterance forwarding strategies. This allows us to focus exclusively on the selection process in our proposed protocol (updates not corresponding to a node's own interest or to those of his friends will not be considered for selection).

It is important to recall here that our exclusive focus is on *push* mechanisms that better emulate current on-line micro-blogging services. Hence, we exclude more 'expensive' *pull* mechanisms involving exchange of information of the nodes at *each* encounter (such as handshake mechanisms after the connection of peers and exchange of current cache content). This means that nodes do not have knowledge on what the pairing node is currently carrying in its cache to use when taking decisions about what items to push to it; these decisions are rather based on earlier acquired knowledge about nodes in the same social community.

Inevitably, the dissemination of utterances results in some sort of (controlled) flooding process and the question is how to make this most informed and efficient. Our protocol relies on knowledge of interest similarity across network nodes to achieve this. Two forwarding *modes* can be broadly defined, as opposed

to the interest-agnostic mode (*push random*), where nodes randomly select utterances from their caches to forward to the encountered peer without accounting for their interests.

First, nodes forming friendship links may use the individual push profiles for selecting utterances to forward during pairwise encounters. Specifically, when node x meets a social friend y it probabilistically selects a given number of utterances according to the stored push protocol $\mathbf{P^{xy}}$, with the further constraint that invalid utterances are not considered in the selection process. We refer to this forwarding mode as *Push according to friends interests*.

Secondly, when a node x does not store directly the push profile of the currently encountered node, it can use an aggregate of the profiles of all his social friends for forwarding (*Push according to community profile*). Similarly to the previous mode, this selection strategy selects utterances from the whole cache probabilistically according to its community profile defined in equation (8); again, invalid utterances, either time- or location-wise, are eliminated.

3. Experimentation methodology and set-up

In this section, we summarise our approach to assessing the performance of our push protocol. We describe the scenarios we chose for the interest profiles of nodes and their mobility patterns in the physical space; the push protocol variants for forwarding to friend and stranger nodes; and the metrics that summaris e our protocol performance dynamics.

3.1. Mobility model

Mobility models determine the frequency and duration of encounters between different nodes, thus dictating the micro-blog dissemination opportunities. In this work, we have chosen a 'social' mobility model, the Home-cell Community based Mobility Model (HCMM) [15], which can take into account underlying social structure among the network nodes. HCMM structures the physical space into disjoint *cells* and organizes the network nodes into groups (communities) with common spatial context: all nodes of a given community are randomly assigned to one of these cells, the community's *home cell*. The nodes' movement patterns *across* cells are determined by the social attraction forces exerted upon them by other nodes and follow a fixed pattern. As long as nodes move within their home cell, they select to stay therein or move towards another destination cell with a probability proportional to the social links they maintain with nodes that are already at or move towards that cell. Each visit to a destination cell lasts variable time and is succeeded by a return to the home cell with a certain probability. *Within* cells nodes move following a simple *random waypoint* model. A detailed description of the model can be found in [15].

HCMM lets the mobility patterns of nodes correlate with each other and reflect both their social relationships and their preferences to move towards certain locations. In our work, we use the model to

	No. Familiar Nodes		
thr_F	Min	Max	Avg.
50	84	108	95.73
150	37	68	51.19
200	22	63	41.27
350	7	42	20.05
500	3	32	11.82

Table 1: Statistics for the per-node familiarity node sets; the last three columns list the number of those sharing the same home cell

generate highly variable frequencies of pairwise node encounters and, manipulating the familiarity threshold parameter thr_F, control the cardinalities of nodes' familiarity sets $|F|$ (see Section 2). It is then possible to vary the level of similarity in the interests of nodes sharing the same home cell (see 2.1) and, thus, generate different scenarios for the overlapping of nodes' similarity and familiarity sets.

Table 1 shows statistics about the sizes of nodes' familiarity sets for different values of thr_F, when we run simulations with 120 nodes distributed in groups of 30 across four cells. It also reports how many of them share the same home cell, reflecting the joint impact of social links and location preferences on the resulting pairwise encounters. As it can be seen, nodes may become more selective as to which nodes they will consider familiar by increasing thr_F. The size of the familiarity set has a direct impact on what a node should do and how much effort it needs to devote in forwarding (*e.g.*, Protocol B in 3.3). In our experiments in this paper we have used $thr_F = 150$ for a time window of 50000 iteration. This is consistent with the setting used in [8], which conducted a tuning and a study on the sensitivity and optimisation of the familiarity threshold, and we set the parameter that was forming a similar number of familiar nodes on average per node (about 50 nodes)

Note that, differently from [8], the set of familiar nodes is here of minor impact since are not used directly to define the social communities (as we focus exclusively on interest similarity for their formation)

3.2. Interest profiles of nodes.

The nodes' interests are simulated in one of two ways:

- a) they are synthetically generated by manipulating the Zipf(s) distribution, where s is the skewness parameter of the distribution[2]. Through synthetic interest profiles we can vary *controllably* the average

[2]The Zipf distribution has been shown to be a good model for the popularity of web objects [16] and is remarkably flexible in capturing a wide range of distributions, from the uniform ($s = 0$) to highly skewed ones with power-law characteristics ($s >> 0$).

similarity of interests among nodes with the same home cell (*in-similarity*) and with different home cells (*out-similarity*). In and out-similarity can be defined analitically as following. Given n_i a generic node belonging to community c_i the *in-similarity* of c_i is defined as $\frac{\sum_{ij} sim(n_i,n_j)}{n^2} \forall n_i, n_j \in c_i \, and \, n = |c_i|$ and the *out-similarity* of c_i as $\frac{\sum_{ij} sim(n_i,n_j)}{n*m} \forall n_i \in c_i, n_j \in c_j : c_i \neq c_j \, and \, n = |c_i|, m = \sum_{j \neq i} |c_j|$ where the function $sim(a, b)$ calculates the similarity of nodes n_a and n_b.

- b) they are extracted out of real data of an online social networking application. This way, we can generate a more realistic structure for interests' similarity, which cannot be easily synthesised through probability distributions.

3.2.1. Synthetic interest profiles.

In our experiments we have considered a population of 120 nodes, distributed in four groups of 30 nodes. Each group of nodes is placed in a separate cell. The interests of each node are distributed over m tags, out of a global set of $|M|$ tags ($m < |M|$) (see Section 3.2.2). Each node's relative interests over these tags are derived from a Zipf(s) distribution, where the value of s is chosen randomly in $[0, 2]$. We examine four cases, varying the way the nodes'interests are distributed over the corresponding tags. We consider the case with $m = 30$ and $M = 120$.

Case 1 Tags are selected randomly and the ranking of tags is random for each node.

Case 2 Interest communities start to appear. Within each home cell, all tags are common but the preference ranking is different for each node. The sets of common tags are disjoint between different home cells.

Case 3 The interest community structure gets stronger. As with *Case 2*, only now all nodes in the same home cell have the same preference ranking for tags.

Case 4 This is a further variant of test *Case 2*, but with high similarity of interests across nodes in different home cells, *i.e.*, the same 30 tags attract the interests of all 120 nodes.

3.2.2. Interest profiles extracted from real data.

We have collected real data by crawling the Delicious[3] website, a collaborative tagging application that allows users to bookmark web resources annotated with tags. Delicious users *follow* other individuals through subscribing to their bookmarks. The aggregation of users' tag selections forms a 'folksonomy', a user-generated classification scheme.

We have crawled Delicious as follows. Starting from a single Delicious account (root user) – chosen randomly among those placing recent bookmarks on the website – we have conducted a breadth-first exploration of the graph formed by these links. This search has traversed a *network* of users, who are expected

[3] www.delicious.com

to have similar interests. Let $X = \{x_1, \ldots, x_{30}\}$ be the collection of the 30 Delicious users returned by the search and TG_X be the aggregated set of tags used by them.

To avoid the long tail of infrequently used tags, we prune the resulting sets by restricting our attention to a subset consisting of the M most popular tags used by X, $TG_X(M) = \{tg_1, \ldots, tg_M\}$. If B_i^x denotes the number of bookmarks tagged with i by user x, the interest rates of node x for tag $i \in TG_X(M)$ equal:

$$I_i^x = \frac{B_i^x}{\sum_{j \in TG_X(M)} B_j^x}.$$

The value of M is set to 120, as in *Cases* 1 to 4.

Repeating this process for each of the 30 nodes in X yields the interest profiles of the nodes we place within a single cell of the mobility model presented in Section 3.1. The same procedure is followed to derive the interest profiles of nodes in the remaining cells, by choosing a different root user randomly each time. The setting that occurred from processing the Delicious data in this way is termed as *Case* 5.

Table 2 reports the *in-similarity* and *out-similarity* values for all 5 cases. The index used to compute pairwise similarity is the proportional similarity index introduced in Section 2.1. We also report the value of the *modularity* metric. Modularity quantifies the quality of a division of a network into communities [17], high values implying dense connections (high similarity) between nodes within communities and sparse connections (low similarity) between nodes in different communities. A value of modularity close to zero indicates that links are dispersed randomly throughout all the network, and there is no clear community structure.

Case 1 reports low in- and out-similarity values. This is due to the fact that the interest profiles of nodes are chosen randomly, and thus unrelated to any initial location assignment of nodes to home cells. *Case* 2 and 3 increase the sharing of interests inside the network communities (in-similarity) since nodes assigned to the same home cell use exactly the same 30 tags; since the sets of common tags are disjoint for each home cell, the out-similarity representing the degree of tag overlapping between nodes belonging to different home cells is zero. The degree of similarity inside cells is higher for *Case* 3, where the nodes also share the same order in the ranking of their preferences for each tag. On the other hand, *Case* 4 has the same in-similarity value with *Case* 2 but it also presents high out-similarity values between the different cells. Finally, the real data setting in *Case* 5 reports values of in- and out-similarity between those in *Case* 1 and 4. Regarding the modularity values we note that *Cases* 2 and 3 present the highest values. This is anticipated since there are no links between cells, and thus there is a natural distinction into communities. Such distinction does not exist in *Case* 1 and 4, and the modularity values approach zero. This also holds for *Case* 5, which attests that there is no clear community structure for the Delicious network.

Finally, Table 3 reports statistics for the cardinality of the similarity node set (*i.e.*, friends' node set) for the five interest profiles. It can be seen that a high similarity threshold restricts the set of friends quite severely in some cases.

Table 2: In-similarity, out-similarity and interest community structure modularity for the five interest profile cases

data set	in-similarity	out-similarity	modularity
Case1	0.14137	0.11119	0.0365
Case2	0.43417	0	0.7480
Case3	0.72543	0	0.7499
Case4	0.43417	0.41009	0.0241
Case5	0.25353	0.18447	0.0672

	Number of Friends					
	Similarity 0.5			Similarity 0.2		
Data Set	Min	Max	Average	Min	Max	Average
Case1	0	1	0.08	0	37	13.9
Case2	0	21	8.61	12	29	25.68
Case3	14	29	24.06	27	29	28.9
Case4	0	71	35.48	62	119	105.13
Case5	0	22	4.76	0	89	53.88

Table 3: Friends Node Set Statistics (Similarity 0.5)

3.3. Forwarding protocol variants

We run experiments with three push protocol alternatives. Each one determines the utterances to push to friend and stranger nodes by combining differently the forwarding modes described in Section 2.3.2.

Protocol A - Each of the nodes x pushes utterances to friends y according to the friend's interest profile P^{xy}, pushes to strangers according to its community profile P^x.

Protocol B - Each of the nodes x pushes to friends according to the friend's interest profile P^{xy}, pushes to all familiar strangers according to its community profile P^x, pushes random utterances to all non-familiar strangers.

Protocol C - Each of the nodes x pushes to friends according to the friend's interest profile P^{xy}, pushes random items to all strangers.

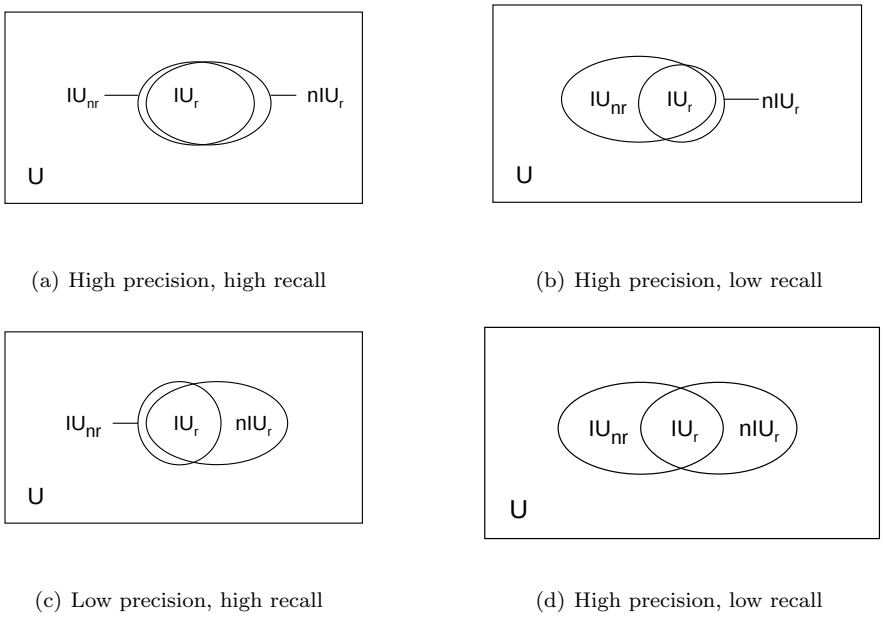
Push Random - Each of the nodes x pushes random items to all encountered nodes.

Push random serves as the basic low-effort flooding benchmark. It represents the least computationally intensive selection process, as utterances are not checked for validity or how well they match against interest profiles before being pushed. On the other hand, protocols A, B and C exploit interest similarity information and both push utterances matching the interest profiles of encountered friends (and/or their 1-hop neighbours'). However, protocol A is more 'socially-selfish' in that it forwards to *all* strangers only utterances that are of interest to its own friend set. For *every* stranger node, the protocol must invest the extra computational effort of locating an utterance that fits the community profile, checking its validity both spatially and temporally and then pushing that to the non-friend node. Protocol B however discriminates between stranger nodes that are met often ("familiar strangers") and those that are seen irregularly and few times ('non-familiar strangers"). For familiar strangers, the computational effort is the same with protocol A, whereas for non-familiar strangers utterances are selected and forwarded randomly from the cache, without spending time and computational effort on checking the utterance validity or matching it with an interest profile. Finally, protocol C is, from a complexity point of view, an intermediate protocol between protocol B and the push-random variant, which pushes only randomly selected content to all stranger nodes.

3.4. Performance evaluation metrics

The performance of the protocol is determined by its capacity to disseminate interesting information across the network without wasting network resources. The dissemination capacity of the protocol is assessed through two metrics with origins in the field of information retrieval. If U is the total set of utterances produced in the network, IU^x the produced utterances of interest to node x, and IU_r^x(resp. nIU_r^x) the interesting (resp. non-interesting) utterances it receives, we can define:

Global precision, P_G. The ratio of the number of valuable utterances received by all nodes (those producing a positive utility) $\sum_{x \in V} IU_r^x$ to the total number of utterances received by all nodes $\sum_{x \in V} (IU_r^x + nIU_r^x)$.



(a) High precision, high recall

(b) High precision, low recall

(c) Low precision, high recall

(d) High precision, low recall

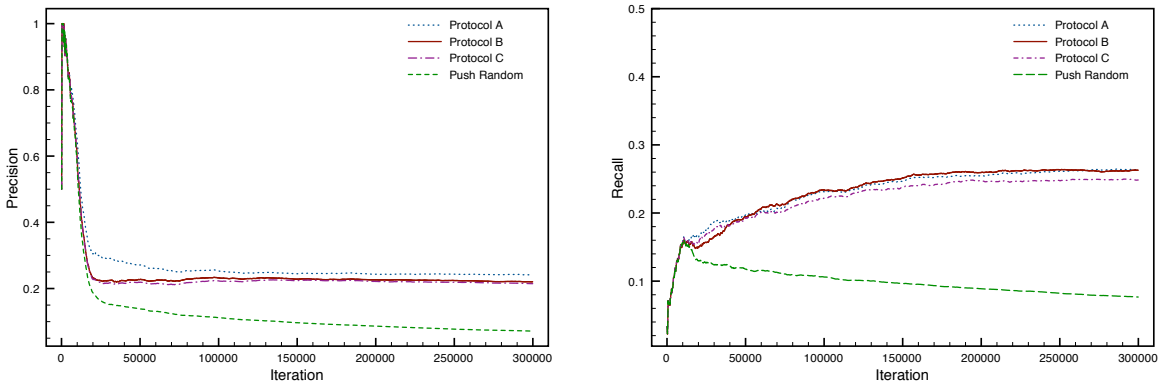
Figure 1: Possible scenarios for the sets of (non-)interesting utterances a node does (not) receive and related precision/recall values.

Global recall, R_G. The ratio of the number of valuable utterances received by all nodes in the network $\sum_{x \in V} IU_r^x$ to the total number of potentially valuable utterances generated in the network (those that could have been successfully received by any of the nodes to produce a positive utility) $\sum_{x \in V} IU^x$.

An efficient push protocol must present good scores for both precision *and* recall metrics (Figure 1(a)). Otherwise, nodes may feature high precision values but still receive only a small number of utterances that fall in their interests (low recall, as in Figure 1(b)); or, achieve high recall values but only at the expense of also receiving an unnecessarily high number of uninteresting utterances (low precision, as in (Figure 1(c))).

4. Experimentation Results

This section presents the results from our extensive experimentation with the micro-blogging protocol. In all simulation runs, the HCMM default parameterisation in [15] is used. The physical space is divided into four cells of size 500m $\times$ 500m. Nodes are divided into four groups with interests distributed as discussed in Section 3.2.2 and each group is assigned to a different HCMM 'home' cell. In all simulation runs, unless explicitly stated otherwise, each node generates utterances according to a Poisson(λ) process with an average rate λ of one utterance every 10 minutes and pushes one utterance on each encounter with another node. One utterance transmission per encounter is considered initially for simplicity, later in section 4.3 we consider pushing more than one utterance per encounter.



(a) Precision

(b) Recall

Figure 2: Precision and Recall, Interest profile dataset 2, Similarity threshold thr_S=0.2

The default utterance spatial validity[4] is $l_{max} = 500$m and their temporal validity t_{exp} is 2 hours. The duration of all experiments reported in the rest of the paper is $300,000$ iterations, with each iteration step corresponding to 0.1 second. Average precision and recall values over the network population are calculated and plotted for each simulation iteration.

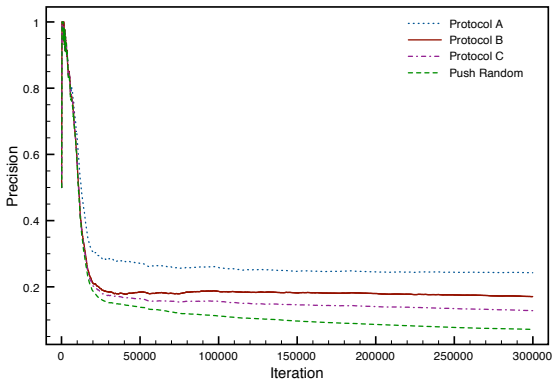
For all input data sets, two values have been considered for the similarity threshold, thr_S, when constructing the set of friends for each node: 0.2 and 0.5. The thr_S parameter modulates the friends' set cardinality; higher values result in fewer friends and less overhead in the protocol operation. The familiarity threshold thr_F is set to 150 encounters.

4.1. Comparison of dissemination protocols

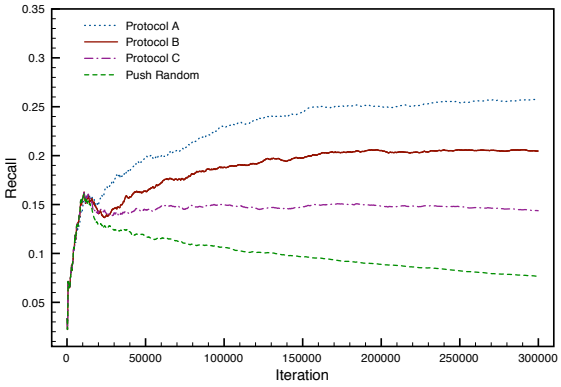
This section compares the four dissemination strategies defined in Section 3.3 under two scenarios for the nodes' interest profiles: one drawing on synthetic interest profiles with high in-similarity and zero out-similarity (*Case 2*); and one drawing on real data with more balanced distribution of in-similarity and out-similarity values (*Case 5*). Figures 2 to 5 report the comparison results for all four protocols.

Looking at the general trend of the precision and recall curves, both of them converge to constant values after relatively short transitive phases. Precision values experience overshoots as the protocol starts getting into operation in the network, as nodes acquire interest profiles and push utterances inline with the interests of the receiving nodes. Likewise, recall values start from zero since it takes some time till nodes start receiving potentially interesting utterances at their caches and grow more smoothly with time towards fixed values. Both metrics appear to reach steady values after a number of iterations that is scarcely dependent on the interest profile scenario, when the fraction of 'floating' utterances that are any time valid in the

[4]For the sake of simplicity, we consider squares of length $a = 2l_{max}$ centered on the utterance generation location

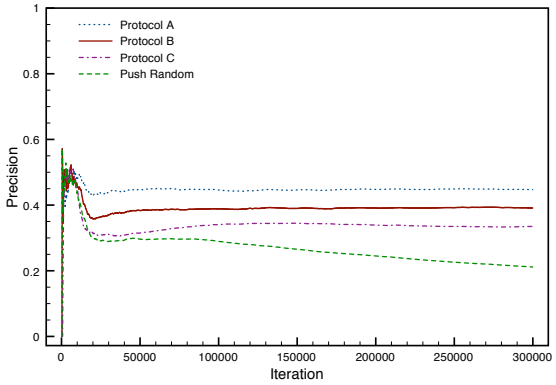


(a) Precision

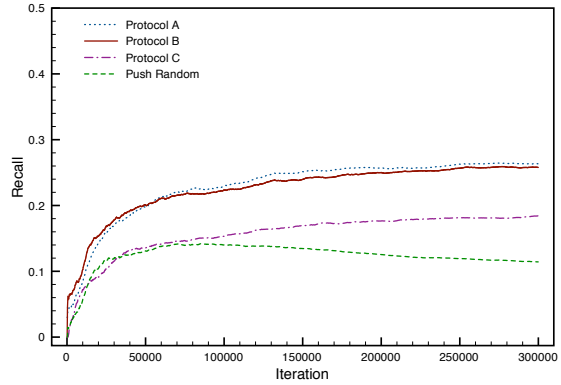


(b) Recall

Figure 3: Precision and Recall, Interest profile dataset 2, Similarity threshold thr_S=0.5

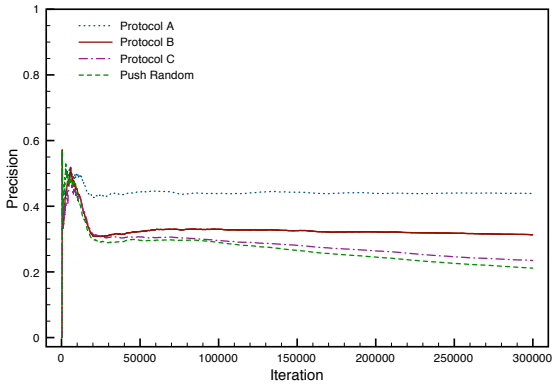


(a) Precision

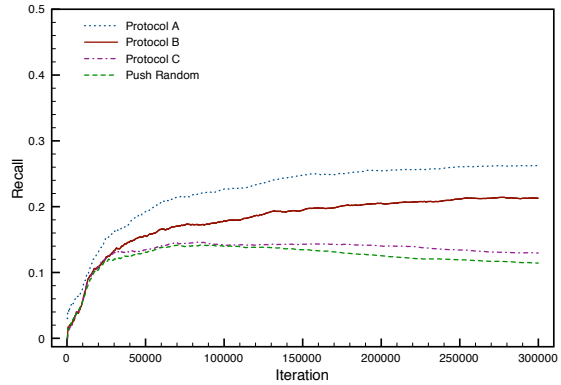


(b) Recall

Figure 4: Precision and Recall, Interest profile dataset 5, Similarity threshold thr_S=0.5



(a) Precision



(b) Recall

Figure 5: Precision and Recall, Interest profile dataset 5, Similarity threshold thr_S=0.5

network stabilises. However, the achieved convergence values and the relative protocol behaviour clearly depend on the interest profiles (a complete comparison of the performance for all different data sets is shown in Section 4.4).

Beyond these common trends, the following remarks can be made when comparing the plots in figures 2-5:

- The precision performance of protocols A, B, and C is considerably better in Case 5 than in Case 2. This is largely due to the milder differences of interests among nodes with different home cells in Case 5. Because of the higher interest similarity between all nodes, utterances matching the average interests of the friend nodes have higher chances to appeal to 'stranger' nodes, *i.e.*, nodes with interests that are not similar enough to classify them as friends. On the contrary, under Case 2 nodes only rarely receive an utterance of interest upon encounters with stranger nodes because there is zero out-similarity, *i.e.*, their interests lie in completely different tags. When the sets of friend nodes are adequately large (lower similarity tresholds), the three protocol demonstrate the same performance, as shown in Figures 2(a) and 2(b).

- Although for higher thr_S the set of friends, *i.e.*, similar, nodes becomes smaller, the performance of protocols A and B is not affected (see Fig. 2 *vs.* Fig. 3 and Fig. 4 *vs.* Fig. 5) since nodes can still push relevant enough utterances to the set of nodes dS that fall outside of their similarity set due to the thanks to the use of the push-community mode towards all stranger nodes. On the contrary, the performance of the others protocol deteriorate substantially.

- The push-random protocol always exhibits the worst behavior setting a lower performance bound. Its performance changes across the two datasets but does not depend on the cardinality of the similarity node set.

In summary, the results suggest that the achieved push protocol performance is not always intuitive but rather jointly determined by the size of similarity node sets, the way the interests are distributed within and across nodes sharing the same home cell, and the exploitation of familiar nodes. Note that, for a given node, its familiar nodes are those that end up doing the most of the actual forwarding work for it. While it may be a waste of effort to push something of more use to a non-familiar stranger, it is not a waste to do so for a familiar stranger in the expectation that he will make the same effort in return. Protocols A and B can use familiar nodes as 'bridges' in order to push utterances of interest to other nodes. Familiarity links can then emerge as a different, although lower, level of friendship than those based on similarity of interests. Such practices have their direct analog in real life; for example, with work colleagues that may not necessarily share interests with us, but with whom we may have an inclination to cooperate. Furthermore, there may be situations in which none of the individuals we have the opportunity to come into contact with bears any

similarity of interests with us. Therefore, we naturally address the most familiar ones among them (rather than pure strangers) in order to disseminate resources representing our interests (with the aim of reaching other network individuals having more similar interests to ourselves).

In particular, we can see how the performance of Protocol C degrades dramatically in some cases. This reinforces the idea that simply pushing random items to stranger nodes is ineffective, and that some knowledge of community interest must be used for those stranger nodes (as carried out by protocols A & B) in order to improve the protocol performance. The 'bridging' action performed by these stranger nodes between groups of familiar nodes is therefore an important contribution to the effectiveness of the model.

In part this bridging and focus on community interest profiles results in a content replication strategy within the network that emphasises our own interests. However, content replication is not a focus of this study, so this is not evaluated against other strategies.

Note that our protocol implicitly assumes that all network nodes adhere in a cooperative way to our proposed protocols and all share the final goals of not only receiving the most useful updates but also disseminating those that can be of highest utility for other network nodes (including those we are not necessarily in contact with or even aware of their existence). Extending the definition of the protocol by considering the existence of malicious or uncooperative behaviours would require the direct introduction of trust and cooperation links among nodes. This goes way beyond the scope of this paper and can be considered as a future enhancement of the push protocol.

Overall, Protocol B represents a trade-off between global precision and recall and the effort required by nodes to carry out pushes. Therefore, we retain it for the rest of the experiments presented in this section.

4.2. Effect of the time and spatial components

This subsection discusses the impact of the time and spatial scope of utterances on the protocol performance. We have run experiments with various values of the time validity parameter t_{exp}, ranging from 0.5 hrs to infinite time (no utterance expiry), and two values of the spatial validity parameter, l_{max}=250m and 2000m, *i.e.*, spatial validity is restricted to the size a single cell and to the whole physical space, respectively. Tables 4 to 7 compare the global precision and recall values obtained at the end of simulations for each combination of spatial and temporal validity, for both Case 2 and Case 5 input data sets and similarity threshold equal to 0.5.

As can clearly be seen, the precision and recall values increase as the spatial and temporal scope of utterances grows. As utterances remain valid longer, they are forwarded in more encounters and reach more nodes that would like to consume them. These results confirm how the scope of the micro-blogs currently available in the network (i.e. their spatial and temporal relevance) should be taken into account in the definition of the model.

Table 4: Global Precision, Interest profile dataset 2, Similarity threshold thr_S=0.5

		t_{exp}		
		0.5h	2.0h	Inf
l_{max}	250	0.18678	0.19225	0.20650
	500	0.19293	0.21177	0.22646
	2000	0.18940	0.21971	0.22997

Table 5: Global Recall, Interest profile dataset 2, Similarity threshold thr_S=0.5

		t_{exp}		
		0.5h	2.0h	Inf
l_{max}	250	0.10328	0.17529	0.18694
	500	0.16004	0.24339	0.23625
	2000	0.19482	0.29117	0.26918

Table 6: Global Precision, Interest profile dataset 5, Similarity threshold thr_S=0.5

		t_{exp}		
		0.5h	2.0h	Inf
l_{max}	250	0.11631	0.19328	0.20330
	500	0.18629	0.30182	0.32277
	2000	0.24952	0.40849	0.45104

Table 7: Global Recall, Interest profile dataset 5, Similarity threshold thr_S=0.5

		t_{exp}		
		0.5h	2.0h	Inf
l_{max}	250	0.12814	0.17271	0.18281
	500	0.17037	0.24288	0.25532
	2000	0.21121	0.31381	0.32028

4.3. Number of items pushed per encounter

In all experiments presented so far, the number of forwarded utterances over the full simulation is constant (one push per encounter). In this section, we compare three alternatives with respect to the number of forwarded utterances and the way these are chosen upon each encounter: a) push one item from the cache probabilistically; b) push the most relevant (top-k) items; or, c) push k items from the cache probabilistically.

Nodes select utterances to push probabilistically using a roulette wheel selection based on the relative interest values stored in the push profile (see Section 2.3.2). When pushing to node y, a node x will select an utterance u with tag i (u_i) with probability:

$$p_{u_i} = \frac{P_i^{xy}}{\sum_{j=0}^{n} P_j^{xy}}$$

Utterances with high relative interest values have a higher probability of being selected for forwarding.

When nodes adopt the push top-k strategy, their stored utterances are ordered according to the encountered node's push profile. The cache of a node x when pushing to node y with each position containing an utterance u with some interest tag (i, j, k, ...) is ordered so that:

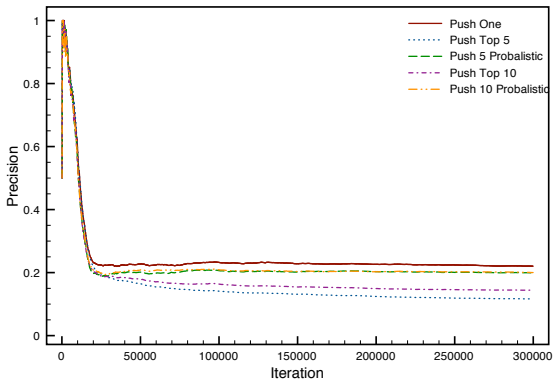
$$\ldots \leq P_{u_i}^{xy} \leq P_{u_j}^{xy} \leq P_{u_k}^{xy} \leq \ldots$$

that is, items below u_i in the cache have lower relative interest values in the push profile, and items above u_i have a higher relative interest value in the push profile. The top k items in the cache are then selected for forwarding. Under this forwarding strategy the same most relevant utterances will always be selected for forwarding, as long as they do not fail in the spatiotemporal check. For this reason, the overall variety in the disseminated utterances is significantly reduced and the subset of most relevant updates ends up being replicated multiple times in the nodes' caches, penalising the protocol performance. In cases, where nodes exercise the push-random strategy, *i.e.*, towards non-familiar strangers, k items are simply chosen at random out of the stored items.

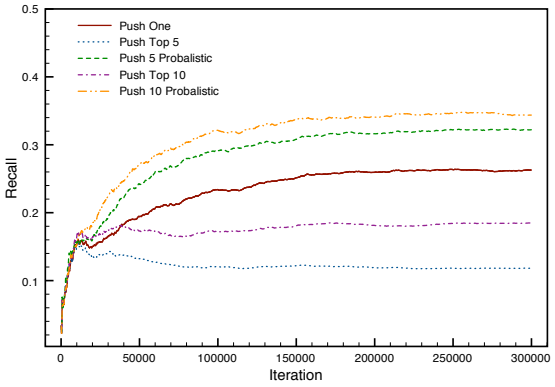
Figures 6 to 9 plot precision and recall values, as obtained in these experiments for cases 2 and 5 and similarity thresholds $thr_S = \{0.2, 0.5\}$.

As explained above, in general, pushing the top k items from the cache delivers worse precision and recall than in all other scenarios. However, the performance gap is narrower when the friend sets are more strictly defined (higher similarity threshold).

On the other hand, the performance of the remaining three forwarding variants, *i.e.*, push one, push 5(10) probabilistic, presents a tradeoff between precision and recall. In all cases the 'push one' scenario delivers better precision than any of the variants forwarding more items per encounter. This means that

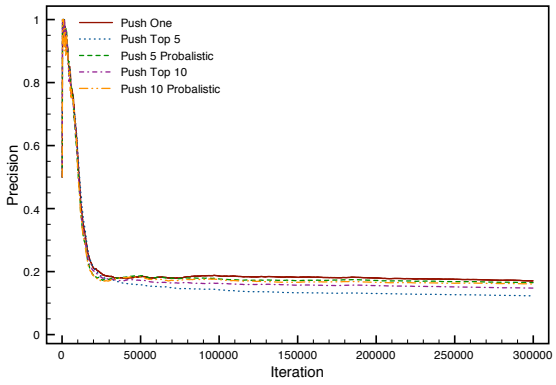


(a) Precision

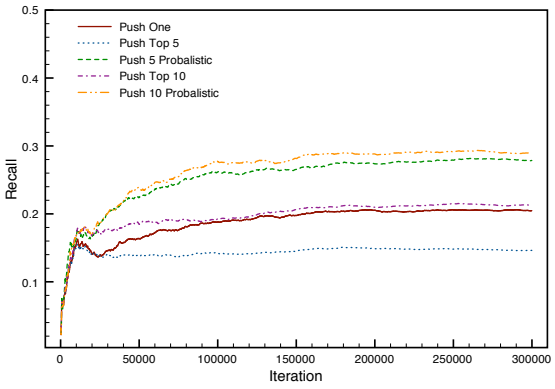


(b) Recall

Figure 6: Precision and Recall, Interest profile dataset 2, Similarity threshold thr_S=0.2

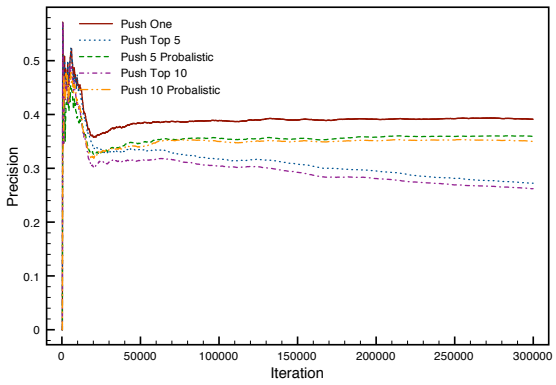


(a) Precision

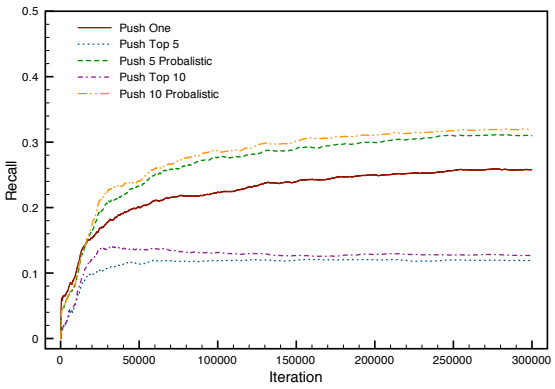


(b) Recall

Figure 7: Precision and Recall, Interest profile dataset 2, Similarity threshold thr_S=0.5

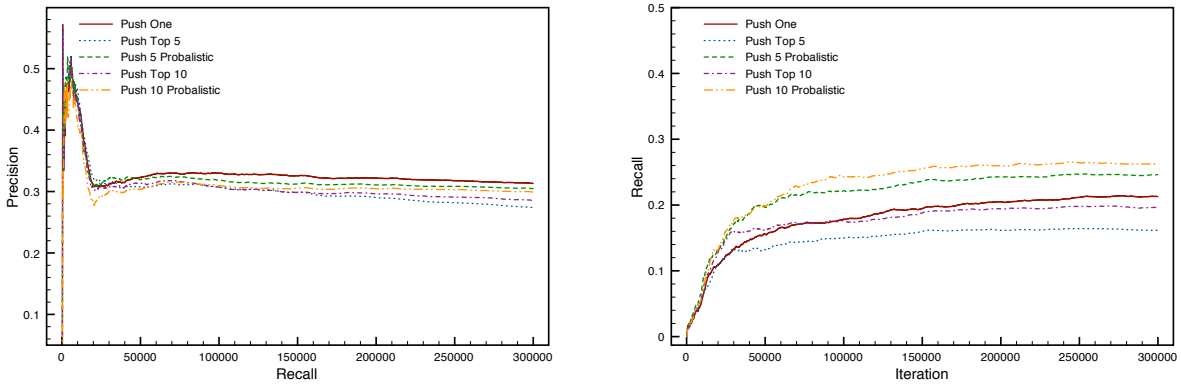


(a) Precision



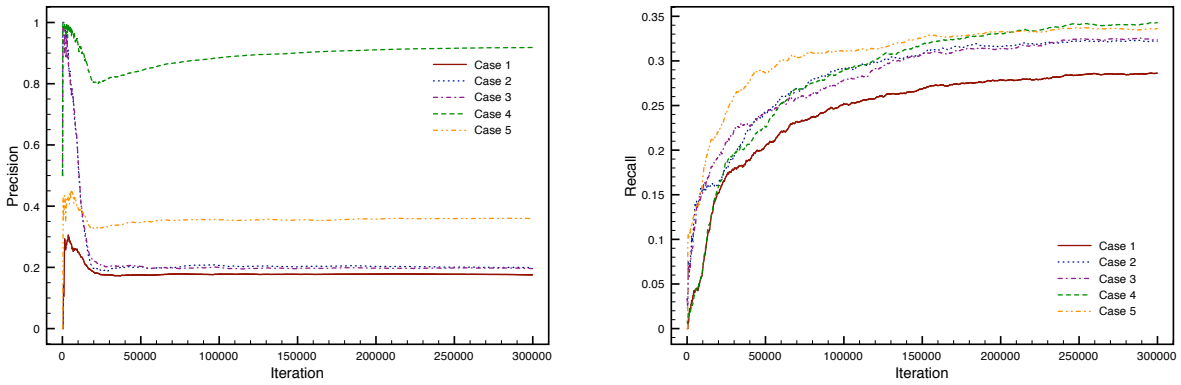
(b) Recall

Figure 8: Precision and Recall, Interest profile dataset 5, Similarity threshold thr_S=0.2



(a) Precision (b) Recall

Figure 9: Precision and Recall, Interest profile dataset 5, Similarity threshold thr_S=0.5

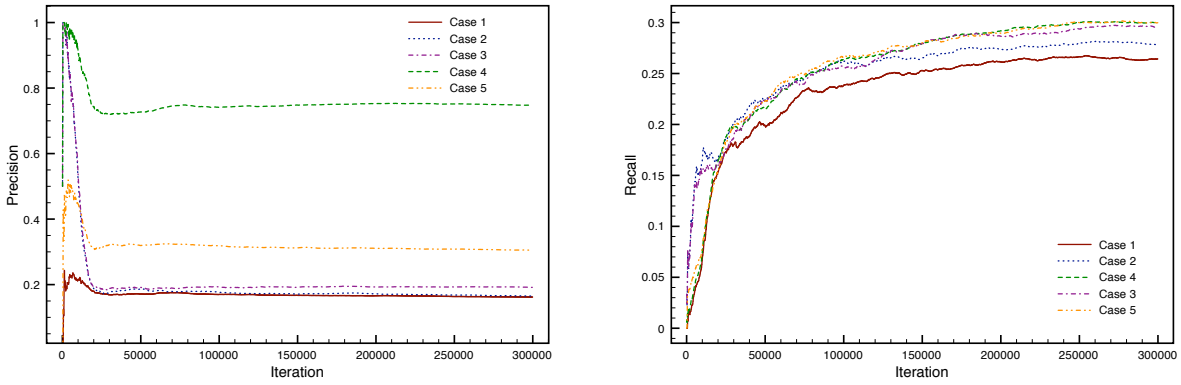


(a) Precision (b) Recall

Figure 10: Precision and Recall, All Data-Sets, Sim 0.2

pushing more items actually introduces 'noise' in the nodes' caches and tends to reduce the ratio of the useful utterances in comparison to the total amount received at a particular time of the run.

Pushing more items inevitably improves the dissemination in terms of global percentage coverage of the available items. The recall delta improvement when increasing the probabilistically forwarded utterances from five to ten is less dramatic than when they are increased from one to five. Apparently, the number of useful utterances a node gains access to would further improve if items swapped all their cache content at each encounter; nevertheless, this would clearly overload the network and sacrifice performance in terms of precision.



(a) Precision

(b) Recall

Figure 11: Precision and Recall, All Data-Sets, Sim 0.5

Table 8: Global Precision - Global Recall

	Similarity 0.5		Similarity 0.2	
Data Set	Precision	Recall	Precision	Recall
Case1	0.15549	0.25193	0.17569	0.28626
Case2	0.16445	0.26408	0.19896	0.32218
Case3	0.18544	0.27642	0.19600	0.32362
Case 5	0.28888	0.28667	0.35342	0.32913
Case 4	0.72911	0.28906	0.91839	0.33823

Table 9: Average similarity per node for the familiarity sets

Data Set	Proportional Similarity
Case1	0.12898
Case2	0.14869
Case3	0.16676
Case 5	0.22196
Case 4	0.41713

4.4. Effect of different datasets

In this section, we conduct experiments with all five interest profile data sets introduced in Section 3.2.2. In all simulation runs, five utterances are pushed during each encounter, chosen probabilistically as described in Section 4.3. The resulting precision and recall values are plotted against simulation time in Figures 10 and 11, whereas their steady-state values are summarised in Table 8. It can be directly seen that all five data sets yield similar recall values, while there is large performance differentiation among them in terms of precision. Note that the actual length of the transition phase for each data set seems to be lightly but positively correlated with the precision scores of the protocol for the respective data set.

As intuitively expected, the artificial *Case 4*, introduced as an extreme scenario for interest similarity across the network (the interests of all network nodes are spread over the same 30 tags), produces the best outcomes overall. On the other extreme, the worst results are presented by *Case 1*; for $thr_S = 0.5$ the nodes' friend sets are almost empty (Table 3). Less intuitively, remarkable performance is achieved by the protocol for *Case 5*, where the nodes' profiles are extracted out of a real-world online social application. Although the dataset features low levels of interest similarity among nodes sharing the same home cell (see 2), it yields better performance than both *Case 2* and *Case 3*, which feature high in-similarity and modularity scores. The protocol benefits more from the better balanced distribution of interests across the network (*Case 5*) rather than the stronger similarity of interests within limited groups of nodes (*Cases 2* and *3*).

An insightful node-level index in this respect, positively correlated with the performance the protocol exhibits for the five interest profile datasets, is the average similarity of its familiarity set F_x. For each node, this index describes how much interest-similar are on average the nodes it encounters more frequently and uses more heavily as forwarding relays. Table 9 reports these indices, as calculated at the end of the simulation for each node (steady-state), and averaged over the whole node population.

4.5. Synthetic traces

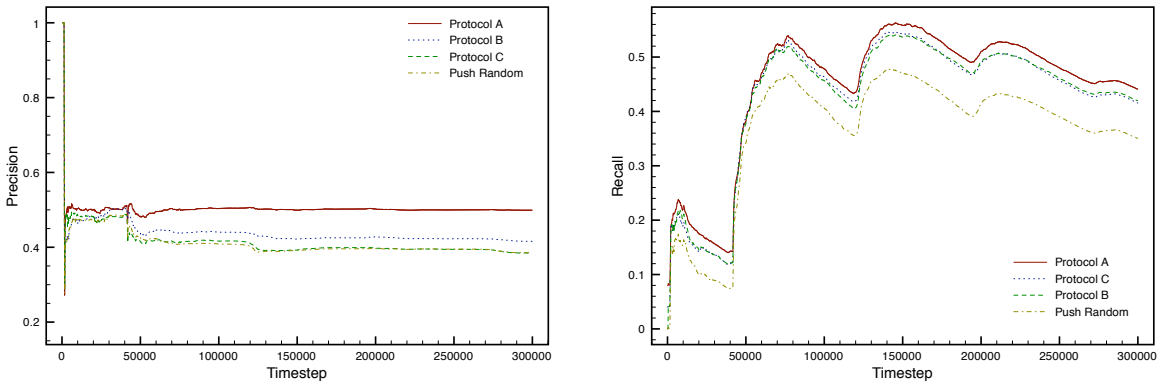
Using the Haggle Infocom 2006 mobility traces shows essentially the same results as when using HCMM. Haggle traces are reduced to 300,000 timesteps to allow comparison with earlier results using HCMM, and are applied to the same test cases considered earlier (see Section 3.2.1) to represent different distributions of interest preferences within the network nodes. All other parameters are the same as used in the rest of the paper. Nodes connect in pairs to other network nodes, then push one or more micro-blogs from their caches and disconnect (we consider as before that this happens within one time step). This allows a node to connect with as many node as are in range as possible, increasing the variety of contacts, although it is worth noticing that with these traces we may not always have more than one node in range at the same time, a thing that was very likely to happen with the mobility considered earlier. If more than one node is in range at the same time the selection of which node to connect to is done randomly. Note that if two nodes stay in range for a long duration they can potentially repeatedly push utterances to each other by repeating the sequence above for a number of times.

Examining the results obtained using the Haggle trace in Figures 12 to 13 show that the shape of the curves is slightly different to the results previously obtained using HCMM, although we can see a similar convergence. This can be due to the fact that here nodes appear to connect to a lower number of peers. This is confirmed by the presence of a smaller size for the familiarity set (31.03 nodes) obtained with he same familiarity threshold $thr_F = 150$ used for all of the experiments discussed earlier in this article.

Moreover, the traces used were originally spread over a number of days and this necessarily involves periods of time (such as night time) when nodes do not come into contact with one another. In our simulation however, the nodes will still produce new utterances (for consistency with the simulation conducted in the rest of the paper), this will clearly decrease the recall value during these periods. This results in the periodic fluctuation in the recall value seen in the results.

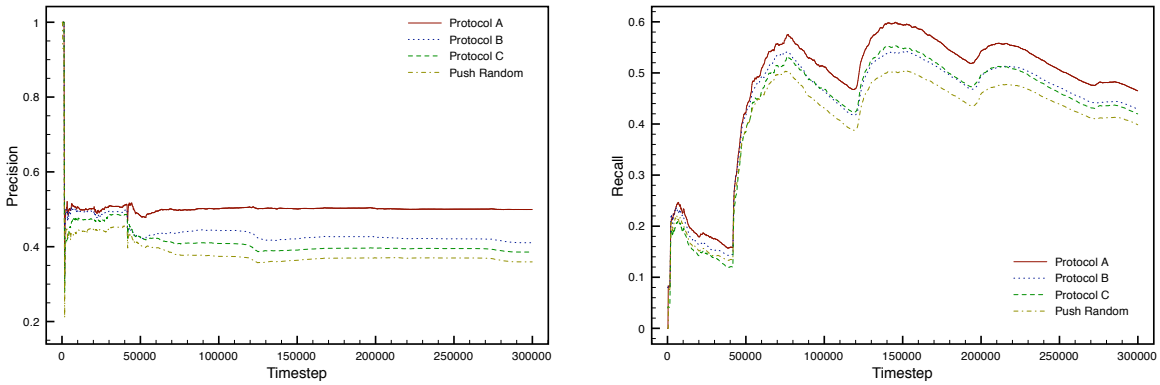
Protocols A,B,C,D perform very similarly to the results obtained with the simulated mobility model (see Figures 12 and 13) with the two protocols A and B that apply our pushing mechanism (based on community profiles) also to nodes outside the social community (either all (A) or only the familiars (B)) resulting in higher performances.

Tables 11 and 12 show the precision and recall values obtained for protocols A and B for the same input data sets used in 4.4 (see Section 3.2.1 for a detailed description). Protocol A is again producing the higher performances thus confirming the results obtained earlier (see Section 4.1). Note that with this specific mobility the application of Protocol B is less immediate. In fact, this protocol exploits, for a given node, the set of its familiar nodes beside all nodes included its social community by forwarding them according the so-called push community protocol based on the 'community profile' (see section 2.3). Therefore the benefits of this protocol are strictly dependent on the ability of such familiar nodes to travel outside their original geographical community to reach other areas and there forward those micro-blogs proper of their



(a) Precision (b) Recall

Figure 12: Precision and Recall, Interest profile dataset 5, Similarity threshold thr_S=0.5



(a) Precision (b) Recall

Figure 13: Precision and Recall, Interest profile dataset 5, Similarity threshold thr_S=0.2

Table 10: Average similarity per node for the familiarity sets

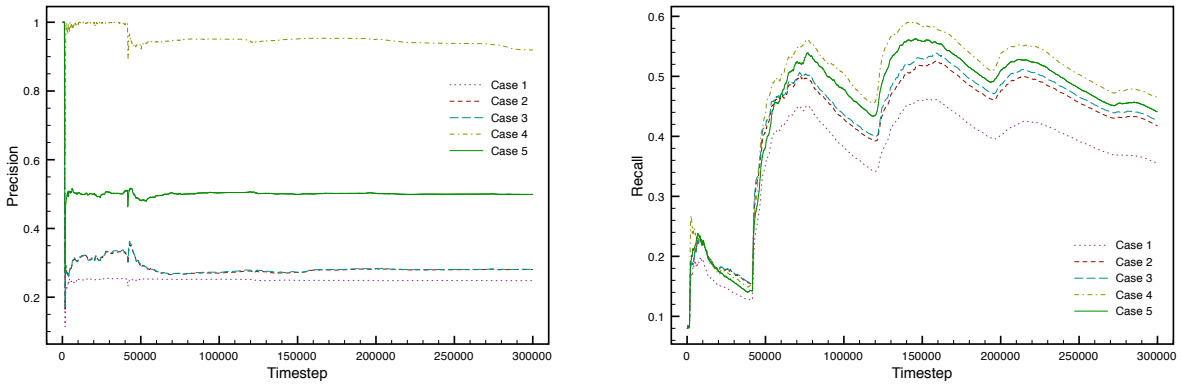
Data Set	Proportional Similarity
Case1	0.159687
Case2	0.200554
Case3	0.232841
Case 5	0.284385
Case 4	0.411996

Table 11: ProtocolB - Global Precision - Global Recall

	Similarity 0.5		Similarity 0.2	
Data Set	Precision	Recall	Precision	Recall
Case1	0.188698	0.338208	0.207787	0.375912
Case2	0.217378	0.408279	0.217671	0.404708
Case3	0.213602	0.392840	0.217734	0.405313
Case 5	0.384623	0.415303	0.410608	0.419664
Case 4	0.901553	0.390607	0.91839	0.39298

Table 12: ProtocolA - Global Precision - Global Recall

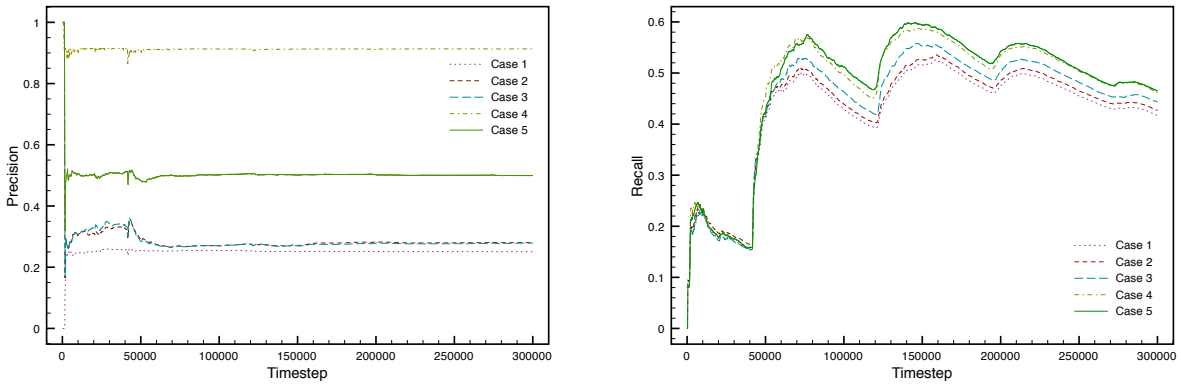
	Similarity 0.5		Similarity 0.2	
Data Set	Precision	Recall	Precision	Recall
Case1	0.24810	0.35520	0.25053	0.35952
Case2	0.28080	0.41756	0.27876	0.42683
Case3	0.28118	0.42572	0.28053	0.44316
Case 5	0.49907	0.44074	0.49942	0.46484
Case 4	0.91937	0.46479	0.91324	0.46156



(a) Precision

(b) Recall

Figure 14: Precision and Recall, All Data-Sets, Sim 0.5



(a) Precision

(b) Recall

Figure 15: Precision and Recall, All Data-Sets, Sim 0.2

original community (acting then as 'bridges'). However, whereas the simulated mobility based on HCMM allowed each node to come potentially into contact with any other node in the network (all moving within a geographically restricted area) with the synthetic traces we cannot expect any particular patterns of movements and this makes the application of protocol B less predictable.

If we compare the performance of the different data sets, the positive correlation with the index representing the *similarity of the familiarity set* (most frequently encountered nodes, see Table 10) can still be observed for both protocols. This emphasises the fact that the protocol performance increases the more similar are the nodes that come into contact. The values of this index are (almost in all cases) higher than those considered previously with HCMM (also a consequence of having smaller familiarity sets), and this has the consequence of producing on average higher values of precision and recall.

Note that the fact that similar nodes come into contact more frequently clearly improves recall, since a node has more opportunity to receive those micro-blogs most valuable for himself that are produced by the nodes sharing more interests (higher similarity value). We can also see that, in general, reducing the similarity threshold (thus increasing the number of social friends to which our actual protocol is applied and so the computational effort since we store individual profiles for any of the nodes in the social community) does not seem to produce very significant improvements in terms of precision (which is the most significant performance measure to evaluate the quality of the updates provided to a given node). This confirms a more effective choice of the protocol when applied to a restricted social community of similar nodes.

A good performance is once again shown by the real case dataset (Case5) that presents homogenous values of similarity throughout the network, thus making at least a percentage of the utterances carried in each node's cache potentially valuable for the rest of the network peers. This is clearly evident in Figures 14 and 15 that show the performances of Protocol A for both of the similarity thresholds considered and

presents a very similar behaviour to the one shown in Section 4.4: the effect of the different interest profiles used as an input is much more significant in terms of precision while only a marginal improvement can be observed for the recall metric.

5. High level analysis and discussion

This section proposes a more high level analysis of the results discussed aiming to identify what are the parameters and the variables that mainly influence the performance of the model. In particular, attention will be given to the response of the model to variations in the input data representing the distribution of preference of interests of the individual nodes and how the use of different mobility models may affect the performance.

5.1. Impact of the protocols on the lower bound

As shown in section 4.1 protocols A and B clearly outperform the other forwarding strategies tested. This is a consequence of the fact that these two protocols not only exploit the social community of a node (by keeping record of the distribution of interests of the individual nodes that are members of it and forwarding micro-blogs accordingly) but also those nodes outside the social community (all nodes for protocol A and only familiar nodes for protocol B). The interest profiles of the nodes outside the social community are here not directly stored and used for direct forwarding but are instead used as 'bridges' to forward micro-blogs representing the interests of the sending node and its community. Hence the effectiveness of such strategies is dependent on the ability of those nodes to explore the network, thus disseminating resources of potentially common interest. Any of these two variations of the protocol clearly outperform the basic protocol that pushes utterance randomly for any encounter. In this random scenario, nodes do not form neighbourhoods with similar or familiar nodes, but rather push an utterance selected at random from the cache to every node that they meet, thus this protocol can represent a baseline flooding to compare the performance of the other protocols against. Note that, to our knowledge, this protocol constitutes the only pure push protocol that could be identified for comparison (free from any form of hand-shaking during pairwise connections, or any other exchange of knowledge about the status of the resources currently carried by an node).

A two way analysis of variance (ANOVA, see [18, 19]) has been conducted on the average values of precision and recall at the end of the run for the two sample cases shown in Figures 2 to 5 (for different values of the similarity threshold) showing that both protocols A and B produce improvements of statistical significance with respect to the values produced by the lower bound represented by the random push protocol. We obtain values of the F-statistics of 14.55 for precision and 39.03 for recall, both greater than the tabled values of 7.70. The complete ANOVA tables are shown in Figures 16 and 17 of Section 8 considering a significance level $\alpha = 0.05$. Hence, we can reject the null hypothesis that the improvements brought by our

protocols A and B with respect to the basic random pushing are not statistically significant. In addition, we can see that the F-statistics calculated on the mean variations for the different cases tested is lower than the tabled values for both precision and recall (4.02 and $2.83 < 6.38$), therefore we can accept the null hypothesis that the different cases tested do not produce effects of statistical significance (with respect of the other source of variation).

5.2. Impact of the spatial and temporal parameter

We have also conducted an ANOVA to evaluate the impact of the temporal and spatial parameters considered in tables 4 to 7 (again assuming for simplicity the absence of any mutual interaction). Results for a significant level $\alpha = 0.05$. are shown in Tables 18 to 21 separately for precision and recall and for each of the sample cases considered 2 and 5. The analysis on the recall values show that both the spatial and temporal parameters have a significant impact on the performance (although the effect of the spatial parameter shows higher values of the calculated F-statistics, thus leading potentially to lower values for the significance level α). For precision, this tendency is increased as we can see that in the first case (5) the time influence the null hypothesis of no impact on the results can be only barely rejected ($19.63 > 19.00$). Furthermore, for the other case considered (Case 2 in Figure 21) we cannot reject the null hypothesis of not producing any statistically significant variation for both temporal and spatial parameter ($F = 6.11, 16.26 < 19.00$).

This behavior, that overall stress the major impact of the spatial component on the potential utility of each generated utterance, can be explained by considering the different nature of these quality metrics and by considering the way they are computed in our simulation. In fact, the values of precision calculated as the ratio between the useful utterance received to the total received can be only slightly affected by variations in the temporal and spatial validity of the utterance (since the proposed protocol takes into account such validity in the selection of the most useful utterance to forward to the currently pairing node). On the contrary, recall could be more heavily affected since a restriction in the time and space validity of micro-blogs will necessarily reduce the percentage of those able to reach destination nodes potentially interested into them within their range of validity. Note that the reasons above together with the difficulties in effectively predicting nodes mobility (and so the actual encounters that are possible for a single node) all have a significantly negative impact on the recall values. As a consequence, the precision metric appears as largely the most important for an overall quality measure of the performance.

5.3. Impact of the Interest profiles in relation to protocols, mobility, and similarity thresholds

By analysing in Sections 4.4 and 4.5 the results obtained for both the mobility patterns considered (simulated mobility with the HCMM model and synthetic traces) we have observed how the response of the system to different dat sets, representing distribution of interests of the network nodes, appears to be significantly different in terms of precision but relatively similar for recall. In particular, it is the effect of

such distributions of interests for the most frequently encountered nodes that suggests a positive correlation with the improvements in the overall system performance (we have also noted how this index representing the similarity of the familiar nodes appear higher for the results obtained with the synthetic traces). We then conclude our discussion by evaluating to what extent the variations in node's interest preferences introduced by the different datasets have an impact in relation to other system variables such as the use of different similarity thresholds, the application of the specific strategy, and the mobility models themselves.

5.3.1. Effect of the similarity thresholds

We have observed how the results for the different input datasets (e.g different distribution of interests for the network nodes) generally increase when lowering the similarity threshold, thus enlarging the social communities of each node and then the number of nodes towards which the friend's interest profiles are recorded and used directly for the selection of the utterances to push. However, we want here to investigate if such differences are statistically significant with respect to those caused by the use of the different datasets and how this impacts the different metrics considered. We have conducted a series of ANOVA's examining the results shown in Table 8 (for the simulation using HCMM) and Table 11 (for the synthetic traces) whose details are shown in Tables 22 to 25. We can observe from the results (Tables 22 and 24) that the improvements on the dependent variable precision are not significant for both mobility models since the F-statistics calculated is lower than the tabled one ($3.81, 7.07 < 7.70$), whereas the impact of the different distributions of interests in the input file is well above the tabled threshold ($F=58.49, 283.78 > 6.38$) and so we can clearly reject the null hypothesis that their impact is not significant on the precision values observed.

Different outcomes are observed on the dependent variable recall (Tables 23 and 25). They confirm the statistical significance of the impact of the different interest distributions showing values of the calculated F-statistics above the tabled ones for both mobility models, although with much closer gaps than with the precision variable ($F=15.64, 8.04 > 6.38$). In addition, the increase in values caused by an enlargement of the social community are significant only for the HCMM case ($39.68 > 7.70$) whereas the opposite happens for the results obtained with the synthetic traces ($F=2.17 < 7.70$, that does not allow the rejection of the null hypothesis of no significant influence). The latter result could be related to the fact that with the synthetic traces the most frequently encountered nodes appear more similar to each other compared with the simulated mobility (see Tables 9 and 10). This has the effect of producing higher performance in recall since similar nodes sharing interests in the same micro-blogs topics come into contact directly, and further enlargements of node communities do not bring about any further significant improvements. This also confirms the fact that the response in terms of recall for using different input profiles of interest is not very important.

In conclusion, enlarging the social community by reducing the value of the similarity threshold does not produce important improvements especially in terms of precision that, as mentioned above, constitutes the

most important metric for a measure of the quality performance of the system. Note that precision is the only parameter that is measurable by each node individually in a distributed way while recall can only be a global metric computed centrally by the system.

5.3.2. Effect of the specific dissemination strategy

Having established the statistical significance of the impact of the different data sets providing different distributions of the interest preferences in the network, we can now investigate this effect in relation to other parameters, as for example the actual choice of the dissemination strategy. We have sen earlier in Section 5.1 how protocols A and B both produce a significant effect in comparison to the basic push-random protocol adopted as lower bound. We have then conducted a further ANOVA on the differences produced by the two protocols over the whole range of input datasets considered (see Tables 26 to 29). We conducted two separate analyses for the dependent variables precision and recall and for both the similarity thresholds (since similarity shows a significant effect on one of the dependent variables). We are considering for simplicity only the results obtained with the synthetic mobility traces (see Tables 11 and 12).

Results show how the null hypothesis stating no significant impact from the use of any of these two protocols can be always accepted. In particular, for precision this analysis produces very low level for the F-statistics values ($0.24, 0.14 < 5.31$, see Tables 26 and 27), whereas for recall the calculated values are closer (although still lower) to the tabled ones ($F=3.90, 4.70 < 5.31$, see Tables 28 and 29). Therefore the variations in the results produced by either protocols A or B cannot be considered of particular importance, thus making the choice of protocol B preferable since requiring far less computational effort as explained in Section 4.1.

5.3.3. Effect of the mobility

Finally, we are interested in investigating the statistical relevance of the results obtained with the different mobility models when computed over the whole range of input datasets representing the interest profiles of the network nodes. We have observed in Section 4.5 how the precision and recall values obtained with the synthetic traces are on average higher than those for the simulated mobility (HCMM), and how this can be related to higher values of the similarity of the familiar set index in the former case. We have conducted separate analysis of variance for the two dependent variables precision and recall and for the two similarity thresholds used (see Tables 30 to 33) to determine how this improvement may be significant in relation to the entire range of input datasets representing different distribution of interest preferences within nodes

Results show a different behaviour for the two metrics considered. For recall there is an influence of the different mobility with respect to the interest data sets ($F=43.31, 57.43 > 5.31$, see Tables 32 and 33). This effect can be explained by considering that with the synthetic traces the most frequently encountered nodes are more similar to each other (in relation to the simulated HCMM model) and this can lead to a

higher recall rate (since nodes come directly into contact with those producing utterances they are interested in). Moreover, the impact on recall is the direct consequence of the fact that this metric does not present significant variations within the different interests data sets (for both mobility models).

However, more importantly, the precision metric is clearly not affected by the change in mobility, which accepts the null hypothesis of no impact of mobility on precision for the different data sets ($F=0.30, 0.03 < 5.31$, see Tables 30 and 31). This is a clear measure of the effectiveness of the selection mechanism of the protocol that is designed to push and forward valuable content for different patterns of movement of the network nodes (and so different similarity level of the encountered nodes).

With precision being the most indicative quality metric for our model, we can then conclude that the impact of the distribution of interest profiles in the network (represented by the different input data sets) is of major impact on the performance with respect of all size of the social communities (similarity threshold); dissemination strategy; and mobility model used.

6. Related Work

Scalability has long been identified to be a challenge for social systems. *On-line social networks (OSN)* present major scalability problems; as an example, Internet micro-blogging services such as Twitter cannot clearly deal with people having many followers [20]. Hence, and even more so in the mobile context, it is mandatory to control the size of social groups. This can be achieved, for example, by imposing grouping criteria that take into account the frequency of contacts among mobile devices (*i.e.*, *familiarity* [12]) or the commonality of users' interests (*i.e.*, interest similarity [21]). Note that the concepts of familiarity and similarity have a direct analog in human social networks that diversely balance 'family' (kinship) and 'friendship' links [22].

Two examples of social approaches to opportunistic networking are Mobi-clique and Mobisoc. Mobi-clique [23] is a mobile social networking software for smart phones supporting existing OSNs. Its testing on the participants of two scientific conferences has shown that the social network of friends could be completed and improved by considering directly the nodes inter-contacts and sharing information such as their on-line profiles. Mobisoc [24] is another middleware platform aiming to monitor, manage, and share the social organisation of physical mobile communities. However, these architectures are centralised rather than distributed, involve retrieval of event notifications, and rely on the sharing of user profiles rather than proper content.

Social information has inspired the design of *routing* protocols in opportunistic environments such as HiBOp [25] and BUBBLE Rap [26]. HiBOp makes use of social context (*e.g.*, places preferentially visited, hobbies) to assess the potential of encountered nodes as relays towards the message destination. BUBBLE Rap uses more formal concepts and metrics from Social Network Analysis, such as Communities and Be-

tweenness Centrality, to inform its forwarding decisions. Both HiBop and BubbleRap use social information to transmit messages to a single destination that *is* interested in it. On the contrary, our protocol disseminates a particular instance of 'messages', *i.e.*, microblogs, to many destinations, which *may be* interested in them.

Therefore, more relevant to our work are *content dissemination* schemes that have inspired from social attributes. A recent example is ContentPlace [27] that exploits social relationships between users to control the placement of data objects and optimise content availability. With ContentPlace nodes are explicitly viewed as members of one or more locality-related communities; for example, the workplace or the residence neighbourhood. Content forwarding is then utility-driven: content objects present different utilities for each node depending on its own preferences and the preferences of the communities it is a member of. Upon encounters, nodes request those objects in the encountered nodes' caches that maximise their aggregating utility; by weighing appropriately the utility function terms, a node can balance differently its local against the other nodes' interests in his choice of replicated objects.

ContentPlace combines *push and pull* mechanisms for content dissemination with 'downloading' nodes choosing which items should they copy from the encountered nodes' caches. Moreover, nodes need to maintain detailed estimates about the popularity and availability of each data object to all other nodes in the network. A purer *push* approach that also accounts for the content preferences of nodes is described in the Haggle search-based network architecture (SNA) in [28]. Thereby content objects carry metadata describing the broader topic(s) they belong to, while network nodes also declare their interests in these topics. Haggle nodes dynamically construct a semi bipartite relation graph capturing the interest of nodes to content objects and the relevance among data objects themselves and, upon encounters with other nodes, use it to decide what should be pushed to the node. The system also recruits typical *delegate* forwarding schemes acknowledging that interest-based forwarding through direct contacts alone cannot always suffice to serve the dissemination objectives.

Our micro-blog sharing protocol shares some ideas with the two content dissemination schemes. First, it relies on the mobility of nodes and their equipment with short range wireless technologies to opportunistically exchange micro-blogs. Secondly, it explicitly accounts for the interests of users in its push decision, as Haggle SNA and, more implicitly, ContentPlace did and relies on metadata (*i.e.*, tags) to describe the thematic scope of data. On the other hand, it differentiates from the two schemes in important directions. Contrary to Haggle SNA, our protocol does not need to retain a detailed account (relation graph) of the matching between all network nodes and data objects. It introduces a hierarchy in the level of state that has to be maintained at each node, with more detailed information (push profiles) maintained for nodes with similar interests, and no information for nodes with little or no interests' overlap. Moreover, the stretched specification of push profile for $k > 0$ (see Section 2) in combination with simple forwarding policies towards interest-dissimilar nodes, circumvents the deadlocks that exclusive use of interest forwarding creates in Haggle SNA. When

compared to ContentPlace, our scheme saves both the pull overhead (exchange of caches' content upon encounter, per-object utility computations) and the detailed estimates that have to be carried out in the background about the popularity and availability of objects in each community and the network as a whole.

This reduced complexity of our push protocol has been largely dictated by the nature of its workload. Micro-blogs are small-payload data, which are are expected to be generated dynamically and in large quantities at extremely diverse locations in the network. Moreover, compared to typical content dissemination, micro-blogs present a casual type of communication with often restricted spatiotemporal scope. This is an aspect only partially considered in current on-line micro-blogging services like Twitter. In [1], it is reported that once a new trendy topic appears in Twitter, *e.g.*, a news item, the majority of related updates (as well as the majority of re-tweets about that topic) are produced within a very short time window. Furthermore much of the information posted via mobile devices is likely to avail spatially bounded scope. As such, decentralised approaches are a natural evolution of work that has already commenced in this area. For example, [29] describes a prototype system that uses distributed servers to avoid problems that arise from a single service provider. In addition, when information has strong local relevance, subscribing to the updates from individual users is of lesser importance. Instead, users should be provided with the local information with the most relevance to their interests, irrespective of the author (spatial information).

Finally, the adoption of a simple pure push approach to the dissemination of micro-blogs emphasises the decoupling between receiving and consuming a micro-blog instance. Nodes do not necessarily have to become aware of the received updates upon the time of their reception; they are rather free to consume them (read their caches) in their own time (speaking of users [1] uses the term 'potential readers'). A node may well find in its own cache valuable information about desired topics, for example in the form of suggestions/recommendations about facts or events, that is earlier unknown to it.

Our work in this paper expands and further formalises in several ways the preliminary studies reported in [6] and [8]: friendship links between node pairs form based on the similarity of their interests rather than the frequency of their encounters; four forwarding modes are specified and combined to compose four different variants of interest-similarity based forwarding; utilities accounting for the spatiotemporal scope of micro-blogs are formalised. Furthermore, the assessment of the micro-blog dissemination protocols is far more systematic and exhaustive: the nodes' mobility patterns are now directed by a social mobility model (rather than the simpler random way point model); the intersection of the similarity and familiarity sets are controlled via introduction of proper synthetic interest profiles; far more protocol parameterisation options (*e.g.*, number of forwarded utterances per encounter) are analytically evaluated. Finally all simulations in the previous studies are performed in a static environment, where social groups and the corresponding interest distributions within them are assumed to be known a-priori before the beginning of the simulation. In these experiments the set of friends and the set of familiar nodes are constructed dynamically during simulations as nodes come into contact with each other.

7. Conclusions

This article proposes a protocol for dissemination of micro-blogs in opportunistic mobile environments. It is a *push* protocol that circumvents the processing overheads of pull-based approaches and is suited to the requirements of its data workload, *i.e.*, large numbers of small payload micro-blogs, often with restricted spatiotemporal scope. In parallel, it is consistent with current practice in online micro-blogging services.

The protocol exercises *interest-based forwarding*. It draws on limited social information about the way the nodes' interests intersect over different thematic areas to make informed forwarding decisions and improve the efficiency of the micro-blog dissemination process. The mobile devices are organised by the protocol into communities of *similar* nodes with common interests. At the same time, the mobility of nodes gives rise to sets of *familiar* nodes that meet with higher frequency with each other. The maintained state and the forwarding behaviour of nodes differ according to the community(ies) the encountered nodes are classified into. Utterances are forwarded to friend (*aka* similar) nodes inline with their interests (push-profile); whereas, for the rest of the nodes, utterances are either randomly selected or account for the average interests of the push-node's friends (push-community).

We have detailed the protocol components and evaluated variants of interest-based forwarding under a wide set of scenarios for the nodes' interests, their selectiveness in establishing friendship links, and the frequency of encounters with nodes that have similar interests. In all cases, the protocol performance is assessed via two metrics derived from the field of information retrieval: precision and recall. In particular, it is precision that can be considered as the principal measure of the performance quality and whose optimization has been directly addressed by the protocol design in terms of selection and storage of micro-blogs. Nodes mobility has been simulated through the application of a computational model reproducing the nature of human contacts as well as the use of real traces made available to the public. The protocol behaviour could be summarised into the following points:

- The most successful strategies combine pushing according to individual profiles and community profiles. We propose two variations of the protocol that both outperform a basic strategy based on the random dissemination of micro-blogs. However, the performance benefits have to be balanced against the computational effort involved in each strategy implementation. For this reason, an intermediate solution that selects utterances for forwarding based on individual profiles for all friend nodes, while it invokes the community push profile for familiar nodes may be preferable.

- When utterances have restricted spatiotemporal scope the recall metric appears as the most affected while the precision metric does not produce significant variations, thus confirming the effectiveness of the protocol design. In addition, the spatial component of the utility appears having greater influence on the performance than the temporal one, thus suggesting that a proper evaluation of micro-blogging dissemination protocols may have to take the spatial attributes of utterances into account.

- Besides additional computation effort, the possibility to push more than one utterance upon each encounter gives rise to a precision-recall trade off: more utterances of interest reach the nodes' caches but only together with larger quantities of irrelevant updates.

- The scheme performance turns out to always benefit from high coincidence between the nodes' similarity and familiarity node sets. An index that is positively correlated with the precision and recall values the protocol achieves is the average (over all network nodes) similarity index calculated for the set of familiar nodes, the ones more heavily involved in forwarding action.

- The considerations above suggest as how it is the distribution of interests of the network nodes that has the major impact. This is confirmed by a statistical analysis of the results showing as the effect of various system parameters (such as the similarity threshold (that actually defines the size of the social communities and so the range of nodes of direct application of the push-protocol); the particular dissemination strategy used; and the specific of mobility adopted) is not significant when compared to the influence that the different distributions of the nodes profile of interests has on the on the system performance. To take this into account a number of distinct input data sets have been generated and used (although all necessarily based on the 'tags' model). These include a number of artificially generated profiles based on a probability distribution that simulates the 'long-tail' of interest preferences of the individual nodes together with one data sets obtained by crawling data from a real on line social network.

As future work on the protocol, we would promote two items. Firstly, the two mobility models that we have considered in this work propose possible way to capture social context in the way nodes move in the physical space, yet still potentially allowing nodes to explore the geographical regions considered in its entirety. Further insights to the performance potential could be given through the assessment of the protocol with other mobilities that can extend the physical region of movement as well as impose potential restrictions on the nodes mobility, for example by forcing similar nodes to move within specifically defined areas. Secondly, the different forwarding modes introduced in Section 3.3 express different levels of cooperation across the network. The push-community mode, for example, is a form of interest-community selfishness and assumes reciprocation in the nodes' behaviour. The vulnerability (resp. resilience) of the protocol to different instances of node misbehaviours is a research item worth exploring.

Acknowledgements

This research has been funded by SOCIALNETS grant 217141, an EC - FP7 Future Emerging Technologies project concerning pervasive adaptation.

References

[1] H. Kwak, C. Lee, H. Park, S. Moon, What is Twitter, a social network or a news media?, in: WWW '10: Proceedings of the 19th international conference on World wide web, ACM, New York, NY, USA, 2010, pp. 591–600.

[2] B. Krishnamurthy, P. Gill, M. Arlitt, A few chirps about twitter, in: Proceedings of the first workshop on Online social networks, WOSP '08, ACM, New York, NY, USA, 2008, pp. 19–24.

[3] C. Honeycutt, S. C. Herring, Beyond microblogging: Conversation and collaboration via twitter, Hawaii International Conference on System Sciences 0 (2009) 1–10.

[4] E. Berlin, Can microblogging platforms help reduce the email glut?, Work Literacy.

[5] D. Zhao, M. B. Rosson, How and why people twitter: the role that micro-blogging plays in informal communication at work, in: GROUP '09: Proceedings of the ACM 2009 international conference on Supporting group work, ACM, New York, NY, USA, 2009, pp. 243–252.

[6] S. Allen, G. Colombo, R. Whitaker, Uttering: Social micro-blogging without the internet, in: Proceedings of ACM/SIGMOBILE MobiOpp 2010 The Second International Workshop on Mobile Opportunistic Networking, 2010, pp. 58–64.

[7] B. J. Jansen, M. Zhang, K. Sobel, A. Chowdury, Twitter power: Tweets as electronic word of mouth (2009).

[8] S. M. Allen, M. J. Chorley, G. B. Colombo, R. M. Whitaker, Opportunistic social dissemination of micro-blogs, Ad Hoc Networks, 2011.

[9] A. Java, X. Song, T. Finin, B. Tseng, Why we twitter: understanding microblogging usage and communities, in: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis, ACM, 2007, pp. 56–65.

[10] L. Pelusi, A. Passarella, M. Conti, Opportunistic networking: Data forwarding in disconnected mobile ad hoc networks, Communications Magazine, IEEE 44 (11) (2006) 134–141.

[11] L. A. Zadeh, Fuzzy sets, Information Control 8 (1965) 338–353.

[12] P. Hui, E. Yoneki, S. Y. Chan, J. Crowcroft, Distributed community detection in delay tolerant networks, in: Proceedings of MobiArch '07, ACM, New York, NY, USA, 2007, pp. 1–8.

[13] J. Vegelius, S. Janson, F. Johansson, Measures of similarity between distributions, Quality and Quantity 20 (4) (1986) 437–441.

[14] R. Malmgren, D. Stouffer, A. Motter, L. Amaral, A poissonian explanation for heavy tails in e-mail communication, Proceedings of the National Academy of Sciences 105 (47) (2008) 18153.

[15] C. Boldrini, A. Passarella, Hcmm: Modelling spatial and temporal properties of human mobility driven by users' social relationships, Computer Communications 33 (9) (2010) 1056 – 1074.

[16] D. N. Serpanos, G. Karakostas, W. H. Wolf, Effective caching of web objects using zipf's law, in: IEEE International Conference on Multimedia and Expo (II), 2000, pp. 727–730.

[17] M. E. J. Newman, Modularity and community structure in networks, Proceedings of the National Academy of Sciences 103 (23) (2006) 8577–8582. doi:10.1073/pnas.0601602103.

[18] O. J. Dunn, V. A. Clark, Applied Statistics : Analysis of Variance and Regression, John Wiley and Sons Australia, Limited, 1974.

[19] S. M. Ross, Introduction to Probability and Statistics for Engineers and Scientists, Second Edition, 2nd Edition, Academic Press.

[20] A. Java, X. Song, T. Finin, B. Tseng, Why we twitter: An analysis of a microblogging community, Advances in Web Mining and Web Usage Analysis (2009) 118–138.

[21] E. Jaho, I. Stravrakakis, Joint interest- and locality- aware content dissemination in social networks, in: Proceedings of WONS'09, 2009.

[22] S. G. B. Roberts, R. I. M. Dunbar, T. V. Pollet, T. Kuppens, Exploring variation in active network size: Constraints and ego characteristics, Social Networks 31 (2) (2009) 138–146.

[23] A.-K. Pietilainen, E. Oliver, J. LeBrun, G. Varghese, C. Diot, Mobiclique: middleware for mobile social networking, in: WOSN '09: Proceedings of the 2nd ACM workshop on Online social networks, ACM, New York, NY, USA, 2009, pp. 49–54.

[24] A. Gupta, A. Kalra, D. Boston, C. Borcea, Mobisoc: a middleware for mobile social computing applications, Mob. Netw. Appl. 14 (1) (2009) 35–52.

[25] C. Boldrini, M. Conti, I. Iacopini, A. Passarella, Hibop: a history based routing protocol for opportunistic networks, in: Proceedings of WoWMoM '07, 2007, pp. 1–12.

[26] E. Yoneki, P. Hui, S. Chan, J. Crowcroft, A socio-aware overlay for publish/subscribe communication in delay tolerant networks, in: MSWiM '07: Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems, ACM, New York, NY, USA, 2007.

[27] C. Boldrini, M. Conti, A. Passarella, Contentplace: social-aware data dissemination in opportunistic networks, in: Proceedings of MSWiM '08, ACM, New York, NY, USA, 2008, pp. 203–210.

[28] F. Bjurefors, P. Gunningberg, E. Nordstrm, C. Rohner, Interest dissemination in a searchable data-centric opportunistic network, in: Proc. IEEE European Wireless Conference, Lucca, Italy, 2010, pp. 889–895.

[29] A. Passant, T. Hastrup, U. Bojars, J. Breslin, Microblogging: A semantic and distributed approach, in: Proceedings of the 4th Workshop on Scripting for the Semantic Web, Citeseer, 2008.

8. Appendix: ANOVA Tables

This section presents a list of the anova tables. For both one and two ways analysis when the calculated values of the F-statistics are greater than the tabled ones we can reject the null hypothesis that the corresponding effect of the specific treatment or source of variation is not of statistical significance.

Source of variation	Sum of squares	DF	Mean square	F-statistics	F-tabled
Cases	0.04787	3	0.01588	4.0272	< 6.3882
Push Protocols	0.05738	1	0.05738	14.5522	> 7.7086
Residual	0.011829	3	0.00394		

Figure 16: Two-way ANOVA on the dependent variable Precision to evaluate the impact of protocols A and B against push-random in relation to the sample cases 2 and 5 with different similarity thresholds 0.2 and 0.5 (no interaction assumed, $\alpha = 0.05$)

Source of variation	Sum of squares	DF	Mean square	F-statistics	F-tabled
Cases	0.00115	3	0.00038	2.8316	< 6.3882
Push Protocols	0.00383	1	0.00383	39.0304	> 7.7086
Residual	0.00040	3	0.00013		

Figure 17: Two-way ANOVA on the dependent variable Recall to evaluate the impact of protocols A and B against push-random in relation to the sample cases 2 and 5 with different similarity thresholds 0.2 and 0.5 (no interaction assumed, $\alpha = 0.05$)

Source of variation	Sum of squares	DF	Mean square	F-statistics	F-tabled
Spatial parameter	0.01417	2	0.20670	93.0980	> 19.0003
Temporal parameter	0.01316	2	0.00708	86.4946	> 19.0003
Residual	0.00030	4	0.00007		

Figure 18: Two-way ANOVA on the dependent variable Recall to evaluate the impact of the spatial and temporal parameters (no interaction assumed, $\alpha = 0.05$, Case 2, similarity threshold of 0.5)

Source of variation	Sum of squares	DF	Mean square	F-statistics	F-tabled
Spatial parameter	0.02180	2	0.25280	41.0227	> 19.0003
Temporal parameter	0.01232	2	0.03085	23.1966	> 19.0003
Residual	0.00106	4	0.00095		

Figure 19: Two-way ANOVA on the dependent variable Recall to evaluate the impact of the spatial and temporal parameters (no interaction assumed, $\alpha = 0.05$, Case 5, similarity threshold of 0.5)

Source of variation	Sum of squares	DF	Mean square	F-statistics	F-tabled
Spatial parameter	0.05923	2	0.02961	33.8067	> 19.0003
Temporal parameter	0.03439	2	0.01719	19.6300	> 19.0003
Residual	0.00350	4	0.00087		

Figure 20: Two-way ANOVA on the dependent variable Precision to evaluate the impact of the spatial and temporal parameters (no interaction assumed, $\alpha = 0.05$, Case 5, similarity threshold of 0.5)

Source of variation	Sum of squares	DF	Mean square	F-statistics	F-tabled
Spatial parameter	0.00055	2	0.20619	6.1198	< 19.0003
Temporal parameter	0.00148	2	0.00027	16.2655	< 19.0003
Residual	0.00018	4	0.00074		

Figure 21: Two-way ANOVA on the dependent variable Precision to evaluate the impact of the spatial and temporal parameters (no interaction assumed, $\alpha = 0.05$, Case 2, similarity threshold of 0.5)

Source of variation	Sum of squares	DF	Mean square	F-statistics	F-tabled
Interest profiles	0.62414	4	0.15603	58.4975	> 6.3882
Similarity threshold	0.01018	1	0.01018	3.817	< 7.7086
Residual	0.01066	4	0.00266		

Figure 22: Two-way ANOVA on the dependent variable Precision to evaluate the impact of the the use of different similarity threshold in relation to the different interest profiles Cases 1 to 5 (no interaction assumed, $\alpha = 0.05$, HCMM model)

Source of variation	Sum of squares	DF	Mean square	F-statistics	F-tabled
Interest profiles	0.00239	4	0.00059	15.6493	> 6.3882
Similarity threshold	0.00534	1	0.00534	39.6822	> 7.7086
Residual	0.00015	4	0.00003		

Figure 23: Two-way ANOVA on the dependent variable Recall to evaluate the impact of the the use of different similarity threshold in relation to the different interest profiles Cases 1 to 5 (no interaction assumed, $\alpha = 0.05$, HCMM model)

Source of variation	Sum of squares	DF	Mean square	F-statistics	F-tabled
Interest profiles	0.73495	4	0.18373	283.7827	> 6.3882
Similarity threshold	0.00045	1	0.00045	7.0711	< 7.7086
Residual	0.00025	4	0.00006		

Figure 24: Two-way ANOVA on the dependent variable Precision to evaluate the impact of the the use of different similarity threshold in relation to the different interest profiles Cases 1 to 5 (no interaction assumed, $\alpha = 0.05$, Synthetic traces)

Source of variation	Sum of squares	DF	Mean square	F-statistics	F-tabled
Interest profiles	0.73495	4	0.18373	8.0419	> 6.3882
Similarity threshold	0.00045	1	0.00045	2.1769	< 7.7086
Residual	0.00025	4	0.00006		

Figure 25: Two-way ANOVA on the dependent variable Recall to evaluate the impact of the the use of different similarity threshold in relation to the different interest profiles Cases 1 to 5 (no interaction assumed, $\alpha = 0.05$, Synthetic traces)

Source of variation	Sum of squares	DF	Mean square	F-statistics	F-tabled
Between protocols	0.00626	1	0.00626	0.14593	< 5.3176
Within protocols	0.34340	8	0.04292		

Figure 26: One-way ANOVA on the dependent variable Precision to evaluate the impact of the use of the different protocols (A and B) against the whole range of interest datasets (no interaction assumed, $\alpha = 0.05$, Synthetic traces, similarity threshold 0.2)

Source of variation	Sum of squares	DF	Mean square	F-statistics	F-tabled
Between protocols	0.01041	1	0.01041	0.2437	< 5.3176
Within protocols	0.34163	8	0.04270		

Figure 27: One-way ANOVA on the dependent variable Precision to evaluate the impact of the use of the different protocols (A and B) against the whole range of interest datasets (no interaction assumed, $\alpha = 0.05$, Synthetic traces, similarity threshold 0.5)

Source of variation	Sum of squares	DF	Mean square	F-statistics	F-tabled
Between protocols	0.00247	1	0.00247	4.7039	< 5.3176
Within protocols	0.00420	8	0.00052		

Figure 28: One-way ANOVA on the dependent variable Recall to evaluate the impact of the use of the different protocols (A and B) against the whole range of interest datasets (no interaction assumed, $\alpha = 0.05$, Synthetic traces, similarity threshold 0.2)

Source of variation	Sum of squares	DF	Mean square	F-statistics	F-tabled
Between protocols	0.00252	1	0.00252	3.9040	< 5.3176
Within protocols	0.00516	8	0.00064		

Figure 29: One-way ANOVA on the dependent variable Recall to evaluate the impact of the use of the different protocols (A and B) against the whole range of interest datasets (no interaction assumed, $\alpha = 0.05$, Synthetic traces, similarity threshold 0.5)

Source of variation	Sum of squares	DF	Mean square	F-statistics	F-tabled
Between mobility	0.00171	1	0.00171	0.0356	< 5.3176
Within mobility	0.38495	8	0.04811		

Figure 30: One-way ANOVA on the dependent variable Precision to evaluate the impact of the different mobility (HCMM and synthetic traces) against the whole range of interest datasets (no interaction assumed, $\alpha = 0.05$, similarity threshold 0.2)

Source of variation	Sum of squares	DF	Mean square	F-statistics	F-tabled
Between mobility	0.01227	1	0.01227	0.3093	< 5.3176
Within mobility	0.31732	8	0.03966		

Figure 31: One-way ANOVA on the dependent variable Precision to evaluate the impact of the different mobility (HCMM and synthetic traces) against the whole range of interest datasets (no interaction assumed, $\alpha = 0.05$, similarity threshold 0.5)

Source of variation	Sum of squares	DF	Mean square	F-statistics	F-tabled
Between mobility	0.01593	1	0.01593	43.3165	> 5.3176
Within mobility	0.00263	8	0.00032		

Figure 32: One-way ANOVA on the dependent variable Recall to evaluate the impact of the different mobility (HCMM and synthetic traces) against the whole range of interest datasets (no interaction assumed, $\alpha = 0.05$, similarity threshold 0.2)

Source of variation	Sum of squares	DF	Mean square	F-statistics	F-tabled
Between mobility	0.03330	1	0.03330	57.4332	> 5.3176
Within mobility	0.00463	8	0.00057		

Figure 33: One-way ANOVA on the dependent variable Recall to evaluate the impact of the different mobility (HCMM and synthetic traces) against the whole range of interest datasets (no interaction assumed, $\alpha = 0.05$, similarity threshold 0.5)