

# Opportunistic social dissemination of micro-blogs

S.M. Allen, M.J. Chorley, G.B. Colombo, and R.M. Whitaker

*School of Computer Science and Informatics, Cardiff University, Cardiff*  
*{stuart.m.allen, m.j.chorley, g.colombo, r.m.whitaker}@cs.cardiff.ac.uk*

---

## Abstract

Through web sites such as Twitter, micro-blogging has shown a remarkable growth, demonstrating the human desire to share and consume information and knowledge. At the same time, the capabilities of mobile devices such as smart phones has considerably increased, opening up new ways to communicate and share content. In particular it is becoming feasible that mobile devices can directly share content such as micro-blogs without Internet infrastructure. This offers advantages in terms of scalability, and for micro-blogs in particular, it offers the potential to provide content relevant to the end user without explicit subscriptions.

To facilitate this, we propose a totally decentralised push-based scheme for intelligent micro-blogging from mobile devices based on opportunistic networking. This is achieved through mobile devices building interest profiles relevant to communities induced by frequent social interactions. These interest profiles allow the devices to prioritise forwarding the micro-blog payloads that maximise the utility received by others. Detailed simulation studies determine the parameters that affect system performance and demonstrate that the proposed scheme outperforms basic dissemination strategies in terms of the relevance of the received information.

---

## 1. Introduction

Micro-blogging web sites such as Twitter have shown tremendous growth in the short time since their introduction. They allow users to publish short status messages, and to follow (or subscribe) to the updates of other users. Micro-blog posts require less time and effort to write than “traditional” blog posts, yet still allow wide distribution among social networks when compared to email or instant messaging. This is highlighted in an exploratory study of Twitter usage [1], which notes that this brevity further allows the reader to effectively filter large numbers of messages. This work also notes that the broadcast nature of Twitter with voluntary readership reduces the cognitive threshold for the writer to decide to share and the burden of readers to process all updates. The structure of the networks induced by micro-bloggers and their followers [2] makes them an ideal mechanism for rapid dissemination of information amongst ad-hoc social communities. However, this subscription process can make it harder for individuals to discover relevant information external to their social group. The shift to the mobile domain opens up new opportunities for sharing micro-blog posts: that is directly between local devices rather than necessarily using the internet. This reveals the opportunity for users to send and receive micro-blogs while taking into account a common local content that users are participating in (e.g the time of a festival or a journey on public transport).

The overall objective of this work is the proposal of a new totally decentralised push based scheme for intelligent micro-blogging from mobile devices based on opportunistic networking. Here

messages are stored on devices themselves and utilise the interactions between users to forward and filter information about users' preferences. This acquired information is then used to optimise the dissemination of the most valuable micro-blog posts through the network. The protocol is deliberately kept lightweight, restricting nodes to pushing content only. During interactions nodes do not advertise content availability or request specific information, removing the need for any hand-shaking operations or content discovery periods, minimising the protocol overhead. This scheme would particularly suit the dissemination of data during an event that is highly localised spatially and/or temporally. For instance, attendees at a conference could receive relevant information and discussion automatically as they move around the venue, or visitors to a theme park could receive updates about the state of queues or other localised content as they move through the park. Mobile scenarios are necessarily restricted to a limited area, for larger distances an internet based (or hybrid) micro-blogging service would be more appropriate.

In these scenarios much of the information posted via mobile devices is likely to be of only local interest for a short time period. As such, decentralised approaches are a natural evolution of work that has already commenced in this area. For example, [3] describes a prototype system that uses distributed servers to avoid problems that arise from a single service provider. Furthermore, 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.

We consider a pervasive micro-blogging scenario, where users carry smart-phone type devices capable of creating, storing, displaying and forwarding low payload *utterances* (or messages), such as short text messages of less than 200 characters. In place of a centralised cellular network we explore a scenario where messages are to be forwarded opportunistically [4] when devices are within direct range of each other. The protocol is designed for wireless short range technologies, such as Bluetooth and ZigBee, allowing coverage of ranges from 10 to 100m [5, 6]. Messages can either be written directly by the user or generated automatically to reflect their current status (see for example 'the house that tweets' [7]). In this scenario each user has a number of *interests*, and gains utility through reading *recent* messages from other users relevant to their own interests and location. In this work we assume each message has a well-defined topic or hash tag in Twitter terminology (however the protocol will extend to fuzzier classifications).

We will focus exclusively on pure 'push' protocols in which the information senders are not assumed to have any knowledge of the receiving ends current content in order to forward their updates. It is clear that any protocol based on pushing or controlled flooding leads to an increase in the unnecessary dissemination of content.

Therefore, to address these issues, there are two important features of the scenario that affect the evaluation of our push based approach. Firstly, we do not aim to deliver *all* relevant content to *all* interested users. Rather, we aim to ensure that utility is maximised when a user chooses to read the available content on their device. Secondly, we aim to minimise the dissemination of irrelevant information, since pushing irrelevant utterances to devices will force the user to manually filter their content, thus reducing the utility they receive. We conduct our evaluation using appropriate quality metrics adapted from information retrieval (such as *precision* and *recall*) that specifically address the points above.

A popular use of Twitter may be for outside observers to follow the progress of an event from a remote location. However, as a mobile dissemination scenario we are not focused on the global or large-scale dissemination of information, and so issues relating to the transmission of data to

outside observers are not addressed.

The rest the document is organised as follows: Section 2 describes the related work, then Section 3 outlines the data dissemination protocol we examined in this paper. Section 4 describes the experimental scenario while Section 5 presents and discusses the results obtained. Finally, Section 6 draws conclusions on the protocol performance.

## 2. Related Work

Scalability is something that needs careful consideration in social systems, and our decentralised approach allows this to be directly tackled on a local basis. *On-line social networks (OSN)* are known to present major scalability problems, for example Internet micro-blogging services such as Twitter cannot clearly deal with large numbers of ‘friends’ (people followed) in order to optimise information dissemination [8]. Hence, in the mobile context, it is necessary to control social group sizes, and this can be achieved for example by criteria that take into account the frequency of contacts among mobile devices leading to a notion of familiarity as in [9], as well as their commonality of interests (i.e. similarity [10]). Note that these concepts of familiarity and similarity are also related to behavioural science terminology with the necessary balancing of ‘family’ (kinship) and ‘friendship’ links in the formation of social groups [11] (this also refers to the different strengths of relationships in human social networks).

Currently, a limited number of models combining the dissemination of information through opportunistic social networks can be found in the literature. As with our work these are based on the idea that virtual communities such OSNs do not exploit sharing opportunities with people with whom we have daily physical contact. An existing example of social approaches to exploiting opportunistic networks is Mobi-clique [12], a mobile social networking software for smart phones supporting existing OSNs. Mobisoc [13] 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, and nodes are *pull-oriented* to retrieve event notifications from the platform instead of *push-oriented*, which is naturally suited to opportunistic networks. Related platforms for exchanging and managing information and resources using mobile devices have been proposed in [14] and [15].

Other systems in this area are publish-subscribe in which users receive ‘events’ about specific channels to which they subscribe. However, these are only rarely applied to mobile environments, see [16] which also introduces the idea of ‘logical mobility’ (besides ‘physical mobility’) that is the types of subscriptions as well as needs of users may change when they change location. These systems implement data delivery services (using either fixed or mobile broker nodes, see [17]), in which either publishers or subscribers are a-priori aware of the information to be sent or received. This is true also for the so-called ‘content-based’ systems that ‘filter’ events by keywords instead of the mere subscriptions to a limited number of channels [18, 19] (see also content-based delivery [20]).

Many other protocols for dissemination of content opportunistically through ad-hoc networks exist, although these other systems typically include some form of “pull” concept in their architecture, where nodes advertise the content they have available, or the content they would like to receive. One example of such an opportunistic data dissemination system is Prophet [21] which uses a probabilistic routing protocol, allowing nodes to request and receive messages to and from other nodes that are more likely to be able to transmit to the intended destination or recipient. This system uses a similar idea of familiarity to that taken in this paper, in that nodes that are seen more often are more useful for delivery of content than nodes which are seen less often. Another

proposed system for transmitting data through ad-hoc mobile networks is Named Data Networking (NDN) or Content-Centric Networking (CCN) [22, 23]. This system proposes replacing the IP layer of the network stack with the concept of named data, so that content is routed not based on where it needs to go, but on what it is. However, in both these examples nodes must advertise either the content they have available or the content they require, or both. Other dissemination protocols available in the literature all assume handshaking mechanisms to exchange information about their current status before proceeding with the actual exchange of resources. For example, ContentPlace assumes full knowledge of the items stored in the devices caches during pairwise connections [24] whereas epidemic based protocols, such as [25, 26], uses current content information in order to prevent the dissemination of duplicated items.

Our proposed system is an entirely push based protocol, where handshaking and content discovery type actions are minimised. Comparing our work with a protocol where knowledge of the other node is used to inform selection of items to push (such as those mentioned above) would obviously be unfair, as these protocols would necessarily perform better at targeting specific content at other users. However, this increase in performance comes at the cost of higher effort required to select the content to push and increased interaction between nodes at each connection. In a mobile scenario this increased effort and interaction may translate directly to energy usage and should therefore aim to be minimised. Our approach is also significantly different from data delivery since we primarily seek to maximise the satisfaction of users' content available at a particular time and space.

### 3. Data Dissemination Protocol

This section describes the proposed pushed-based protocol for opportunistic 'micro-blogging' dissemination within mobile agents. These represent human individuals in a network of people physically writing and reading this type of information as well as more generic entities such as mobile devices that produce and consume micro-blogs in an automatic fashion and we describe these mobile agents as network 'nodes'.

Agents self-organise via neighbourhoods which are detected and created in our model in order to exchange information about their interest preferences. Subsequently this is used for the dissemination scenario in which agents directly push each other valuable information in the form of micro-blogs in order to maximise their degree of satisfaction. Here we aim to maximise automatic update, selection and storage of the information received without any a-priori registration or subscription, and in such a way that the micro-blogging service is actually useful and feasible to be used: the most valuable micro-blogs are going to be read while irrelevant ones are discarded.

We will discuss in the following sections how utterances are defined and classified; how the utility associated with an utterance is defined; the definition of interest profiles associated to each user; and the criteria behind the formation of neighbourhoods by which means nodes share information about their interest preferences.

#### 3.1. Interest profiles

We assume that each agent carries a set of tags representing its interest preference. These tags are chosen from a clearly defined global set of  $M$  distinct interests. Tags are very useful meta-data and they have been applied to OSNs for suggestion and recommendation systems [27], as well as used for partitioning networks in social communities [28]. For ease of definition, we assume a well defined global set of tags. In fact, each node does not need to know about the existence of any tag

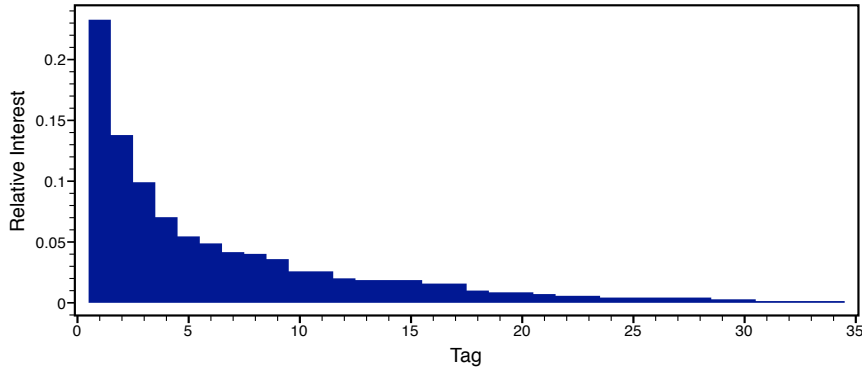


Figure 1: Example Interest Profile, One Node

outside its own set of interests and those of its ‘social neighbours’. There is no reason why new tags cannot be introduced to the system as nodes exchange content, but for simplicity the set of tags is kept static in this work.

Each mobile device is represented by a node  $x$  in a set  $V$  with a predefined *interest profile* denoted by  $\mathbf{I}_x$ , consisting of a vector  $(I_{x1}, I_{x2}, \dots, I_{xM})$ . Each element  $I_{xi} \in [0, 1]$  denotes the relative interest of node  $x$  in the  $i^{\text{th}}$  tag. This represents the probability that a node  $x$  produces (writes) an utterance related to the  $i^{\text{th}}$  tag as well as its desire of acquiring information about (reading) the topic at a certain time. An example of this interest value distribution is illustrated in figure. 1. Note that  $I_{x1} = 0$  implies  $x$  has no interest in tag  $i$ . The values are normalised such that

$$\sum_{i=1}^M I_{xi} = 1$$

### 3.2. Definition of Utterances

Utterances are defined as low-payload data (for example short text messages) of a small enough size that they can be opportunistically passed with high probability. Every node has a cache in which it stores utterances that are received, to be read and assessed for utility at a given point. The cache is organised in a queue system so that the earliest received utterance is removed when a newer one is received or created. Each node  $x$  creates new utterances over time (defined in our experiments according to a certain probability distribution). Each utterance has a single tag  $i$  selected from the *interest profile*  $\mathbf{I}_x$  on a weighted probabilistic basis (assuming therefore that nodes only create utterances of interest to themselves. Note however that this scenario could be generalised so that a node stored two interest profiles, one considering the tweets the node would like to receive and one considering the utterances it will write, which could then be combined (using a weighted sum for example) to give a single interest profile concerning both reading and writing). Each utterance has a single tag as the tags describing content must be carried within the payload of the micro-blog. Increasing the number of tags for an utterance therefore reduces the space available for useful content. Once created, the utterance is stored and made available for pushing. Other variables characterising each utterance are the *time*  $u_t$  when it was generated and the *location*  $u_l$  where it was written. While it is not necessary for each node to be a producer of information for the protocol to function, for simplicity we do not examine the differences between producers, consumers and prosumers in this paper.

### 3.3. Utility metric

When an utterance is received and read by an agent the degree of satisfaction attained by the reader is quantified by the definition of a *utility* metric. This is calculated as a combination of partial utilities due to interest (matching with one of the tags composing the individual interest profile of the receiving agent); time (utility expires after a certain time); and location (utility is the more valuable the closer the receiving agent is to the location where it was generated).

Note that the importance of the locality concept within peer to peer mechanisms for resource sharing has been widely recognised in the literature, for example in the definition of caching resources of the nodes in proximity, as in [29, 30].

### 3.4. Formation of the community graph

Our protocol is based around the formation of a community graph,  $G$ , which can be defined to relate node and to prioritise content across the network. Each node has then defined its own *neighbourhood*  $N_x$  formed by those nodes linked to him in  $G$ . Links are formed according to two criteria: *familiarity* and *similarity*, creating groups of nodes that are either familiar to each other, similar to each other, or both.

**Familiarity** is the evidence of repeated physical proximity between nodes, which can be used to define a link. Based on past evidence these links represent opportunities to forward content to other nodes. In this work this is calculated as the number of historical pairwise contacts that nodes have over a certain amount of time.

**Similarity** is the degree to which nodes have common interests and can also be used to define a link between them. This indicates that they potentially have some content of value to each other.

Both of these concepts are widely used in the literature in *community detection* algorithms, see for example [9, 17] that base the formation of social groups on the duration of inter-contacts between mobile entities and [31, 32] that use similar connection patterns for the detection of groups of components in complex or very large networks.

*Familiarity* links are in our experiments simply placed by averaging the number of contacts obtained in a number of runs of long periods of time (simulations up to 24 hours duration), thus including in a specific node's neighbourhood those agents with whom he has physically connected with a frequency above a given threshold. This choice of defining neighbourhoods a-priori before the actual simulation reduces the number of random variables that could influence the system performance. However, a dynamic community formation could be performed in parallel with this protocol.

We define *similarity*,  $S_{i,j}$ , between a pair of nodes  $i$  and  $j$ :

$$S_{i,j} = \sum_{m=1}^M I_{im} * I_{jm}$$

Users  $i,j$  have  $S_{i,j} = 1$  if they each have exactly the same distribution of interests (tags and probabilities), and that  $S_{i,j} = 0$  when they have no non-zero weighted tags in common. This similarity value is calculated a-priori before a simulation.

For each node neighbourhoods are then formed by enlarging or restricting the set of familiar nodes  $\{F\}$  respectively by combining or adding to the set of similar nodes  $\{S\}$ , forming either the intersection or the union of the two sets. Each node has its own neighbourhood of similar/familiar nodes, and may belong to any number of other neighbourhoods, which may all be distinct from each other.

### 3.5. Definition of push protocols

Each agent  $x$  calculates and stores an individual *push vector*  $\mathbf{P}_y^x$  for each member  $y$  of its neighbourhood  $N_x$ . In addition we define the *community profile* for node  $x$  as the sum of the push vectors extended to all members of its neighbourhood:

$$\mathbf{P}^x = \sum_{y \in N_x} \mathbf{P}_y^x$$

Let  $d(x, y)$  denote the minimum distance between  $x$  and  $y$  in the community graph  $G$ . Let  $D_i(x) = \{y \in V : d(x, y) = i\}$  be the set of nodes at distance  $i$  from node  $x$ . Note that  $D_0 = \{x\}$  and  $D_1 = N_x$  and that within this general definition any individual node is included in only one such set. While  $x$  may not have frequent opportunities to push relevant content directly to nodes other than their direct neighbours, it is possible that neighbouring nodes can forward such information. To explore this we define the push vector to include the interests of nodes at a distance  $k$  from  $x$  in the community graph:

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

Note that this does not actually require all nodes to know the complete (global) enumeration of all interests but only those included in their own neighbourhoods. We consider the values  $k = 0, 1$  and  $2$ , namely where the push vector takes into account only the interest of the neighbouring node (giving (1) - one hop ); the neighbouring node and its neighbours (2) (two hops) or it goes three hops away (3).

$$\mathbf{P}_y^x = \alpha_0 \mathbf{I}_y \tag{1}$$

$$\mathbf{P}_y^x = \alpha_0 \mathbf{I}_y + \alpha_1 \sum_{\tau \in N_y - \{x\}} \mathbf{I}_\tau \tag{2}$$

$$\mathbf{P}_y^x = \alpha_0 \mathbf{I}_y + \alpha_1 \sum_{\tau \in N_y - \{x\}} \mathbf{I}_\tau + \alpha_2 \sum_{\tau \in Y - \{x\}} \mathbf{I}_\tau \tag{3}$$

where  $Y = \cup_{\sigma \in N_y} N_\sigma - \{y\}$  and  $\alpha_0, \alpha_1$  and  $\alpha_2$  are weights between zero and one.

This seemingly simple approach allows us to keep the push protocol relatively lightweight, increasing the ease of implementation.

## 4. Experimental scenarios

This section outlines the scenario used in the experimental tests. In particular we describe: the definition of the interest profiles of each node; the mobility models used to describe the movement of nodes; which metrics will be used to evaluate the system performance; and, finally, the (default) setting of the parameters characterising the model.

### 4.1. Profile of Interests for users

Each node  $x$  holds an interest profile  $\mathbf{I}_x$  which is a vector defined on the global set of tags, as defined in Section 3.1. To more realistically model the definition of the interest profile for each

node we synthesise data by crawling the Delicious<sup>1</sup> website, a collaborative tagging system that allows users to bookmark web resources annotated with tags. The aggregation of tags assigned by users forms a ‘folksonomy’ (a user generated classification scheme). Basing node profiles on an existing OSN allows us to preserve the structure of that scheme among users; this could not easily be represented by a random definition of such interest profiles.

Delicious allows users to *follow* other individuals to subscribe to their bookmarks. We crawl Delicious as follows. Starting from a single Delicious account, a breadth-first exploration of the graph formed by these links allows us to determine a *network* of users that we may expect to have higher similarity in their interests. To determine the interest profiles of users in such a network, let  $T_X$  be the aggregated set of tags used by a collection  $X = \{x_1, \dots, x_n\}$  of the Delicious users. Let  $F_{xt}$  denote the number of bookmarks tagged with  $t$  by user  $x$ , and define the relative weight of tag  $F_{xt}$  among all tags as:

$$f_{xt} = \frac{F_{xt}}{\sum_{t' \in T_X} F_{xt'}}$$

To avoid the long tail of infrequently used tags, we restrict our attention to a subset  $T_X(m) = \{t_1, \dots, t_m\}$  consisting of the  $m$  most popular tags used by  $X$ , where the popularity of tag  $t$  is given by:  $\sum_{x \in X} f_{xt}$ . The interest profile of node  $x$  is then defined by:

$$I_{xi} = \frac{F_{xt_i}}{\sum_{t \in T_X(m)} F_{xt}}$$

The experiments in this paper construct a sample data-set of tags which are determined by considering four sub-networks each of 30 Delicious users rooted at distinct individuals, each sub-network restricted to a maximum of 99 tags.

#### 4.2. Mobility models

In this work we have focused on mobility models that aim to consider a social dimension for the mobility of the nodes (to different extents); these mobility models determine the opportunities for nodes to share data during a simulation. Assuming that nodes are organised into communities of similar individuals based on interests, we have then implemented two models that are both variations of the well-known ‘random waypoint model’, but that present different restrictions to nodes movement inside and outside the geographical areas originally assigned to each community.

The first model is the *Modified Random WayPoint (MRWP)*, based on a simple extension of the basic ‘random waypoint model’, that takes into account the hypothesis that groups of similar individuals (e.g. friends) may tend to share similar movement patterns. Mobile nodes are partitioned into communities that have their mobility restricted to a specific area of the geographical region considered. Within each of the areas nodes move according to the original random waypoint (for a detailed description of this model see [33]).

A second model considered is the *Home-cell Community based Mobility Model (HCMM)* proposed in [34]. This model is based on a pre-partitioning of the network into communities each initially assigned to distinct specific area (cells) of the whole geographical region. Nodes tend to move inside their assigned area (home cell) following a simple random waypoint but some of them can also move outside these areas towards different ones belonging to different communities (destination cell). This happens with a probability (attraction) based on the number of connections

---

<sup>1</sup>www.delicious.com



(links) that each node has with the nodes inside and outside its own community. HCMM is a model derived from the original *Community based Mobility Model* (CMM) proposed by Musolesi et al. in [35], with the variation that when a node moves outside its home cell it has a further probability of returning back to it straight away during the next movement. Finally, every ‘reconfiguration period’ each community of nodes is assigned a new home cell and move towards it as the next destination cell.

HCMM represents a more realistic mobility model, characterised by nodes returning to their ‘home’ after a period of movement, while the MRWP model offers a less realistic, but still useful model showing the interactions between nodes with a smaller range of movement and fewer interaction opportunities.

### 4.3. Utility

An utterance  $u$  provides  $x$  with utility depending on the time, current location of the node, the parameters defining the utterance (its tag, time and location of generation), its position in the cache, and the interest profile of the receiver. We assume that spatial and temporal validity are important concepts when considering the usefulness of micro-blogs within a mobile scenario; beyond some distance and time from their origin they become less useful and so provide less utility.

We define  $t(u, t)$  as a time function that decreases linearly with time  $t$  from a value of one when  $t = u_t$  to zero when  $t$  reaches an expiration time  $t = t_{exp}$ .  $s(u, d)$  is a spatial function that decreases linearly with the distance  $d$  between the current location of the receiver and the location  $u_l$  (expressed in cartesian coordinates) where  $u$  was generated; it assumes value one when  $d = 0$  and zero when  $d$  reaches a maximal threshold distance  $d = d_{max}$ .

An utterance produces utility not at the time the agent has received it but when it is read by the user. The corresponding utility is then assessed with the introduction of a *reading time* interval when the latest utterances received (and currently kept in storage in the device’s cache) finally produce an utility for the receiver. We assume that this, as well as the *writing time* interval when an agent generates a new utterance, follows a Poisson probability distribution (using different average values for reading and writing). These could be equal for all of the networks nodes as well as vary for each individual agent. Note that, in any case each node follows its own independent Poisson process and so having the same average reading/writing time for each node does not imply that all nodes are acting simultaneously.

Moreover, the utility produced depends on what position (‘page’)  $u_p$  in the cache an utterance has when it is read (see figure 2): the closer to the bottom of the cache, the greater overhead in accessing the utterance, thus the utility received is reduced. This is represented by another function  $r(p)$  that decreases with the position  $p$  that an utterance can have in the cache. We assume here the cache is displayed to the user in  $p_{max}$  different pages (see Figure 2) and  $r(p)$  is a discrete function which varies linearly from value one



Figure 2: Utterances Organised in Pages - One Page on Mobile Device

corresponding to the first page ( $u_p = 0$ ) to zero corresponding to the last available page in the cache ( $u_p = p_{max}$ ).

The utility finally scored when node  $x$  reads an utterance tagged by tag  $i$  is then defined as

$$U(t, d, u, x, p) = I_{xi} * t(u, t) * s(u, d) * r(p)$$

#### 4.4. Selection and storage

When two nodes come into contact the criteria for selection of which utterance should be pushed to the connected agent depends on whether or not the connected node is a neighbour. Namely, in an interaction between node  $x$  and node  $y$ :

- if  $y$  belongs to the neighbourhood of  $x$  then we assume that node  $x$  selects an utterance from its cache using a roulette wheel selection based on  $x$ 's estimate of the utility  $y$  will gain, based on the push vector  $\mathbf{P}_y^x$  and weighted by the utility that each stored utterance would produce according to the probability values  $P_{yi}^x$  as defined in  $\mathbf{P}_y^x$ . Namely, the weight for an utterance relative to tag  $i$  is then:

$$P_{yi}^x * t(u, t) * s(u, d)$$

- if  $y$  does not belong to the neighbourhood of  $x$  then we consider three alternative strategies:
  - *Push Community to Stranger*: select a utterance from its cache using a roulette wheel selection based on the ‘community profile’  $\mathbf{P}^x$  and weighted by the utility that each stored utterance would produce according to the probability values  $P_i^x$  as defined in  $\mathbf{P}^x$ .
  - *Push Random to Stranger*: randomly select and push an utterance from its cache.
  - *No Push to Stranger*: do not push any utterance and close the connection.

As has already been mentioned, the protocols tested in this work do not include any type of ‘pull’ or remote content selection mechanisms, so comparison against other opportunistic data dissemination systems can not be meaningfully undertaken. To assess the performance of these different protocols, we instead compare with a *Push Random* protocol. 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. This protocol represents a baseline “flooding” protocol to compare the performance of the other protocols against. In this case every utterance received is pushed onto the cache for possible re-pushing later.

An agent  $x$  stores in its cache only those utterances whose tags are included in its community profile  $\mathbf{P}^x$ . Other received utterances that do not satisfy this requirement are disregarded. It may happen that a node receives a duplicate of an utterance for which it has already computed utility. Such duplicated utterances will not be stored in the node’s cache to control the number of duplicates evident in the network nor will any utility be derived from reading them.

#### 4.5. Definition of metrics for evaluation

We consider the following metrics that capture different aspects of system performance: the global cumulative utility; global precision and global recall.

**Global cumulative utility.** The cumulative sum of the utilities received by every node from reading utterances, at each time step. This metric measures the global ‘degree of satisfaction’ of the system over all the valuable utterance received by all network nodes.

**Global precision.** This is the ratio of the number of valuable utterances received by all nodes (those producing a positive utility) to the total number of utterances received by all nodes. This metric (as well as the following) is an adaptation of that introduced in [36] concerning information retrieval.

**Global recall.** This is the ratio of the number of valuable utterances received by all nodes in the network to the total number of potentially valuable utterances generated in the network (i.e. those that could have been successfully received by any of the nodes to produce a positive utility). Note that this considers as potentially valuable any utterance matching any of the tags in the interest profile of any node, assuming that the utterance is instantaneously received in the same location as it was generated. This metric does not therefore take into account degradation in utility due to time or locality.

#### 4.6. Default parameter settings

The simulation scenario considers a population of  $N=120$  mobile nodes placed into a geographical region of  $500 \times 2500 m$ . The region is further divided into four distinct adjacent areas (cells) each of size  $500 \times 625 m$  to which the nodes belonging to each of the four sub-networks composing the default input data set are initially assigned (see Section 4.1). Note that we relate the initial location of the nodes with the similarity of their interest profiles. Unless stated otherwise we consider for each group of experiments the following definition of the parameters and variables characterising the system.

The default duration of the experiments is of 300,000 time-steps of 0.1 second each, equivalent to 8.33 hours. For each of the experiments we have conducted five runs with different random seeds for each of the proposed push protocols.

HCMM is assumed to be the default mobility model used in our experiments. Each of the nodes is given values of speed selected randomly within the interval (0,3) km/h to represent an average walking speed in a dense environment and has a transmission range of 50 meters and a time to connect of 0.5 sec. Note that the transmission range of 50m can be considered as intermediate, with respect of the current state of art for these technologies and also indicated as desirable in some circumstances (for example industrial wireless data networks, see [37]). Since utterances are resources of very limited size we consider a payload for transmission equal to one single time-step (0.1 sec). This difference between the time to establish a connection and the time to transmit data represents the fact that establishing a connection may take more time than transmitting data for low-payload utterances. In each ‘push’, one utterance is transmitted.

We assume nodes write utterances according to a Poisson process having a mean inter-arrival time of 10 minutes whereas they read their cache and receive utility from stored utterances following a separate Poisson process with an inter-arrival time of 0.1 sec (so a reading time interval corresponding, on average, to one time step). This reading time allows comparison with previous experiments using a similar model [33], and is consistent with for example, a user being notified of and reading an utterance as soon as it is received.

We have defined a maximum cache size of 100 utterances for each node. As the protocol is focused on the spatially and temporally relevant data there is little need to keep larger numbers of

items in cache, since the content quickly becomes irrelevant either spatially, temporally or both. The number of pages displayed in the cache is equal to  $p_{max} = 10$ . We assume that the expiration time of an utterance  $t_{exp} = 2$  hours and the default value for the maximal threshold distance is  $d_{max} = 500m$  as these values represent reasonable defaults for the type of scenarios under consideration.

For simplicity we have considered unitary values for the weights of the three sample cases for the definition of the push protocols (see Section 4.1) namely:  $\alpha_0 = 1$  for case (1) (one hop away in the definition of the push protocols  $P_y^x$ );  $\alpha_0 = 1, \alpha_1 = 1$  for case (2) (two hops away); and  $\alpha_0 = 1, \alpha_1 = 1, \alpha_2 = 1$  for case (3) (three hops away).

## 5. Experimental Results

We have carried out multiple experiments varying the parameters within the system in order to understand the system performance and behaviour. Firstly, the protocol used for pushing utterances between nodes is examined in Section 5.1, followed by the parameters defining the neighbourhoods of nodes in Section 5.2. We then examine the spatial component of the system, varying the threshold distance for communication in Section 5.3. The average times at which utterances are read and written by nodes are then tested in Section 5.4, following which we examine the effect of the interest profiles for each sub-network in Section 5.5. Finally, we examine how the protocol performs if the mobility model is changed in Section 5.6. For all experiments, we present the results from the 2 hops push profile (case 2) unless otherwise stated.

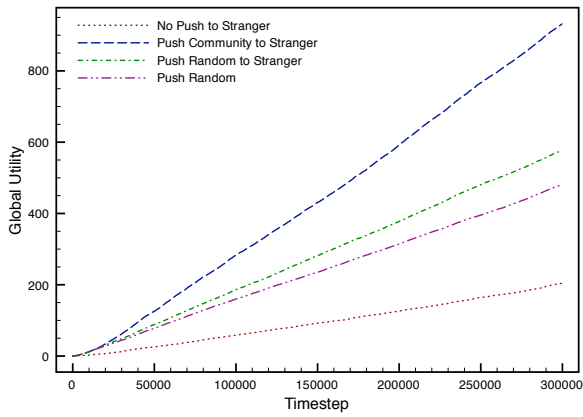
### 5.1. Push Protocol

In the first group of experiments we compare the different criteria for selecting and pushing utterances when a pair of nodes meets and connects: *Push Community to Stranger*, *Push Random to Stranger* and *No Push to Stranger*. For comparison we compare with the *Push Random* scenario. Results are shown in figure 3 for the different metrics defined in Section 4.5.

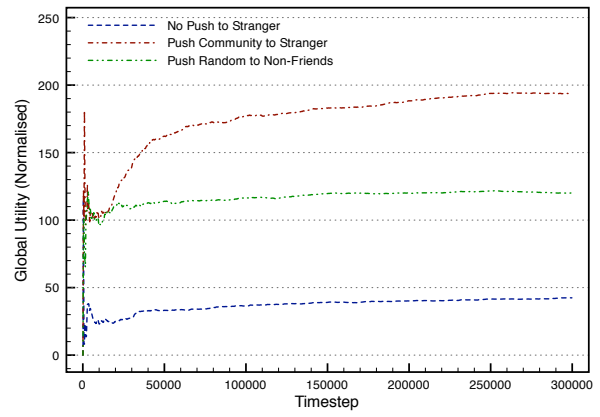
The basic random flooding protocol shows the worst outcomes for all the metrics considered and is considered a *lower benchmark* for the overall system performance. This can be used to effectively ‘normalise’ the results for the global cumulative utility metrics in order to compare the relative performance of the *intelligent push protocols*. The normalisation is obtained by dividing the global utility value at each time step by that produced by the random benchmark at the same time step (see Figure 3(b)).

To indicate the absolute performance, we consider an hypothetical *upper benchmark* for the cumulative global utility calculated as if any of the generated utterances were instantaneously delivered in that specific location to any of the nodes that could receive (in theory) utility from it (i.e. for which the utterance’s tags matches the profile of interests of the nodes). Figure 4 shows this *hypothetical instantaneous delivery* benchmark. This measure is unrealistic since nodes are limited by their physical mobility, thus the total utility of the system can only produce a fraction of this theoretical benchmark. However, it gives some context to the improvements of the push protocol.

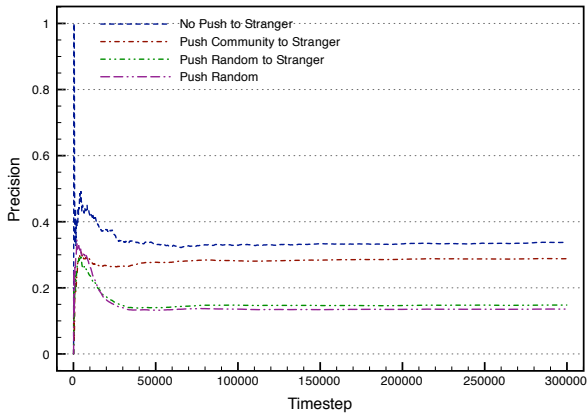
The protocol that only pushes utterances to nodes within the neighbourhood produces the best precision value since it considerably reduces the number of items flooded through the network. However, for the same reason it is worse in terms of global utility and recall. The protocol that pushes to strangers according to the community profile introduced in Section 4.1 shows the best general performance and will be therefore used in the rest of the experimental part of this paper.



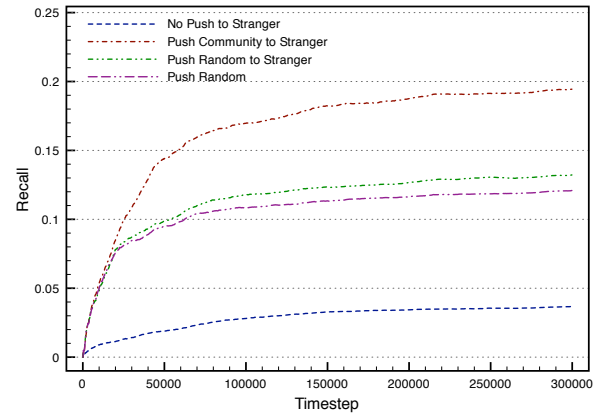
(a) Global Cumulative Utility



(b) Global Cumulative Utility, normalised



(c) Global Precision



(d) Global Recall

Figure 3: Push Protocol Comparison

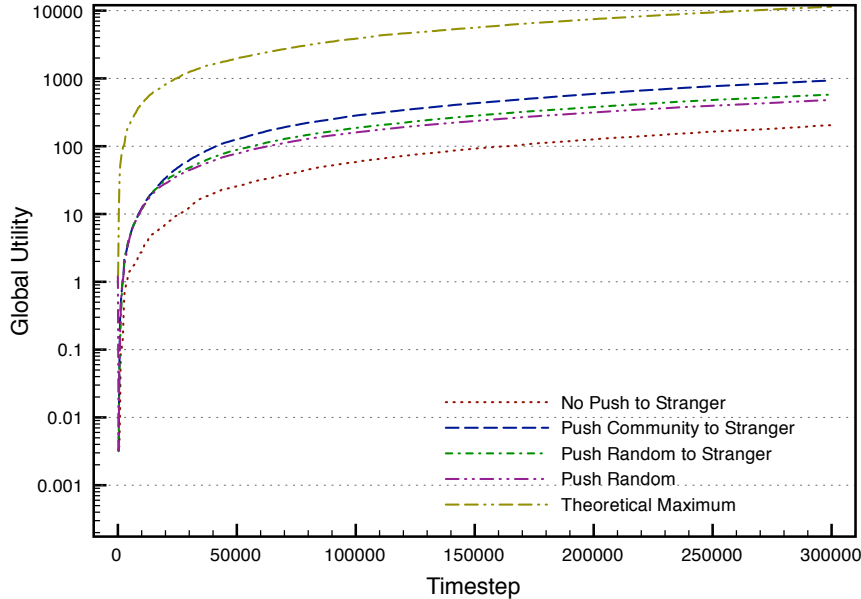


Figure 4: Global Cumulative Utility - Push protocols compared to random and upper hypothetical benchmark, Logarithmic Scale

### 5.2. Effect of neighbourhood size and structure

This group of experiments investigates how the system performance may be affected by different criteria on the formation of neighbourhoods. Table 1 shows the relation obtained in preliminary tests between familiarity (number of contacts) and average size of the neighbourhoods over all network nodes.

We run our simulations with different neighbourhoods constructed with different values of familiarity and similarity thresholds. Tables 2 and 3 show the results for the three sample push protocols, 1 Hop, 2 Hops and 3 Hops. The global utility metric has been normalised with the benchmark obtained applying the random push protocol to the same data set (so the normalisation is, in this case, the same for all single experiments shown in the Tables).

The figures in the table show how groups of particularly large (familiarity 250, more than 250 interactions per test runs) or small (familiarity 2000) size worsen the performance for all of the metrics considered, producing the best overall performance when the neighbourhood is formed by the union of the set of nodes with familiarity and similarity thresholds of 500 and 0.9. Note that dealing with very large neighbourhoods also has the drawback of being rather unrealistic, since it is not actually possible to record and store information about every node in the network.

The 2 Hop case (case 2) appears to produce the best (or comparable) outcomes in terms of recall and global utility although its superiority is marginal and contained within a 10% improvement. 1 Hop (case 1) only considers the profile of interest of the receiver in the definition of the push profiles produces, and so tends to maximise the precision metric (since the receiving node is only being pushed utterances that are directly valuable for itself). However, this is only obtained when the size of a neighbourhood is significant whereas for smaller groups (see fam. 1000 and above) the protocols considering more ‘hops away’ appear to be superior with this metric.

Based on these experiments with the size and structure of the neighbourhood, the default

Table 1: Relation between familiarity threshold and average neighbourhood size

Familiarity	Average Community Size
250	110.6
500	52.3
1000	15.1
2000	4.7

Table 2: Global Cumulative Utility Normalized to Random protocol

Fam $\cup$ $\cap$ Sim.	1 Hop	2 Hops	3 Hops
2000 $\cup$ 0.9	154.93	169.23	98.19
1000 $\cup$ 0.9	183.06	193.70	185.89
500 $\cup$ 0.9	<i>195.41</i>	<b>205.40</b>	<i>201.66</i>
250 $\cup$ 0.9	176.15	190.51	185.89
2000 $\cup$ 0.5	171.74	171.99	139.95
1000 $\cup$ 0.5	192.21	197.50	197.50
500 $\cup$ 0.5	183.83	190.59	195.48
250 $\cup$ 0.5	173.40	188.71	194.78
1000 $\cap$ 0.5	150.51	146.36	51.54
500 $\cap$ 0.5	153.86	155.37	76.15
250 $\cap$ 0.5	156.55	161.66	77.73

neighbourhood for the remaining experiments in the paper will be comprised of the set of nodes with familiarity threshold of 500 and a similarity threshold of 0.9.

### 5.3. Effect of the spatial parameter

This group of experiments investigates the sensitivity of the system to varying the maximal threshold distance  $d_{max}$  in the definition of the spatial component of the utility (see Section 4.3). We have there mentioned how this component and the concept of disseminating useful micro-blogs locally between mobile nodes that may connect within a specific local area can be of considerable importance in the definition of the system.

Tables 4, 5 and 6 consider a range of values for  $d_{max}$  expressed in meters (keeping all other parameters as default as given in Section 4.6), where an infinite value of this parameter corresponds to a scenario in which the spatial component is not considered in the utility definition. From these results we can see that the performance of the system worsens as the the local area of validity of utterances received is reduced. This is a consequence of the fact that this reduces the number of the overall utterances that can produce a positive utility since no utility will be gained for those that were generated outside the current local area. Similar results would obviously be obtained looking at the absolute value of the cumulative utility metric shown in Table 4 (for  $d_{max} \rightarrow 0$  the global utility will clearly tend to zero too, since no utility could be produced for  $d_{max} = 0$ ).

However, if we look at the normalised value for the global utility we obtain an opposite behaviour, see Figure 5. Therefore reducing the local area of validity of the received utterances

Table 3: Global Precision and Global Recall

	1 Hop	2 Hops	3 Hops
Familiarity and Similarity Values	Precision		
	Recall		
2000 $\cup$ 0.9	0.0417	0.0517	0.0563
	0.1676	0.1690	0.1634
1000 $\cup$ 0.9	0.2117	0.2880	0.2858
	<i>0.1924</i>	0.1944	0.1939
500 $\cup$ 0.9	<b>0.4297</b>	<i>0.3094</i>	<i>0.3124</i>
	0.1862	<b>0.1968</b>	<b>0.1968</b>
250 $\cup$ 0.9	0.4145	0.3066	0.3082
	0.1740	0.1915	0.1916
2000 $\cup$ 0.5	0.0515	0.0683	0.0885
	0.1806	0.1735	0.1878
1000 $\cup$ 0.5	0.2348	0.2955	0.2956
	0.1886	0.1946	0.1946
500 $\cup$ 0.5	0.3463	0.3067	0.3083
	0.1801	0.1914	0.1916
250 $\cup$ 0.5	0.4098	0.3061	0.3084
	0.1712	0.1903	0.1918
1000 $\cap$ 0.5	0.0340	0.0330	0.0301
	0.1653	0.1583	0.1308
500 $\cap$ 0.5	0.0381	0.0398	0.0412
	0.1635	0.1632	0.1471
250 $\cap$ 0.5	0.0408	0.0425	0.0440
	0.1631	0.1670	0.1503



Table 4: Global Cumulative Utility

$d_{max}$ (m)	1 Hop	2 Hops	3 Hops
250	437.19	502.96	517.17
500	886.96	932.30	915.35
1000	1140.25	1268.42	1292.28
$\infty$	<i>2088.62</i>	<b>2172.07</b>	<i>2090.56</i>

Table 5: Global Cumulative Utility Normalised to Random protocol

$d_{max}$ (m)	1 Hop	2 Hops	3 Hops
250	<i>214.21</i>	<i>246.07</i>	<b>253.43</b>
500	195.41	205.40	201.66
1000	130.58	145.24	147.99
$\infty$	126.93	132.03	127.05

actually increases the system performance in terms of global utility when compared to the random push benchmark. This confirms the importance of the concept of ‘locality’ in resource exchange environments.

#### 5.4. Effect of the Read/Write time

This section investigates how the system performance may be affected by the use of different values of the parameters defining the interval period in which agents write new utterances and read those currently stored in the cache (and potentially receive a positive utility from them). The parameters regulating this are the mean inter-arrival time for the two different Poisson processes corresponding to average writing and reading time intervals.

We have considered different values of the read and write times,  $rw = Read/Write$ , where the read time denotes the average time that utterances are read by a node, and the write time the average time between utterances being written. Tables 7 and 8 show the utility, precision and recall values obtained for each  $rw$  combination. We can observe that as expected the default values in which utterances are on average produced every 10 mins and read every time-step (so simulating a scenario where on average they are read as soon as received, e.g. pop up notifications) produces the best (or comparable) outcomes for all the metrics considered (see also figure 6).

Results in terms of precision and recall show in general a decrease in their values when both average reading and writing intervals are increased. This is a consequence of the fact that increasing the writing time obviously reduces the number of total utterances available in the system and so the number of ‘valuable’ utterances available for a node to receive is also reduced. Moreover, increasing the reading time makes it possible that some of the utterances fall out of the cache before they can be read (or move towards the bottom part of it where they will score lower utility), thus reducing the number of those that can produce a positive utility. A shorter reading time would allow the node to read these utterances before they drop out of the cache or fall too low, while the longer reading time may only allow the reading of utterances received later. This decline is observed also in terms of normalised global cumulative utility, although the differences appear

Table 6: Global Precision and Recall

	1 Hop	2 Hops	3 Hops
$d_{max}$ (m)	Precision		
	Recall		
250	0.4527 0.1227	0.2874 0.1347	0.2685 0.1331
500	0.4297 0.1862	0.3094 0.1968	0.3124 0.1968
1000	0.4786 0.2228	0.3087 0.2271	0.3212 0.2336
$\infty$	<b>0.4793</b> <i>0.2231</i>	<i>0.3131</i> <b>0.2469</b>	<i>0.3215</i> <i>0.2338</i>

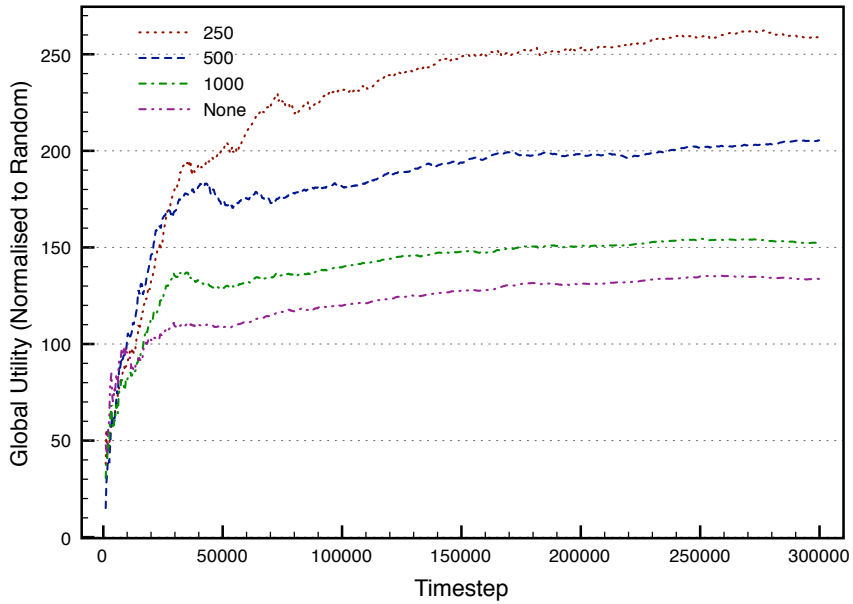


Figure 5: Global Cumulative Utility Normalised - Effect of the spatial parameter,  $d_{max}$

Table 7: Global Cumulative Utility Normalized to Random protocol

$rw$	1 Hop	2 Hops	3 Hops
0.1/10	<i>195.41</i>	<b>205.40</b>	<i>201.66</i>
5/20	178.51	186.84	185.15
15/20	175.51	146.35	170.21
15/30	165.54	172.83	163.22
30/30	161.89	161.19	154.64
30/60	139.61	157.71	158.55

here less important (see Figure 6(a)). Note that the performance of the three different intelligent push protocols remains generally the same for different values of  $rw$ , with the 2 Hops and 3 Hops cases in general marginally better (or comparable) for the global utility and recall metrics, while the 1 Hop case produces the best results in terms of precision.

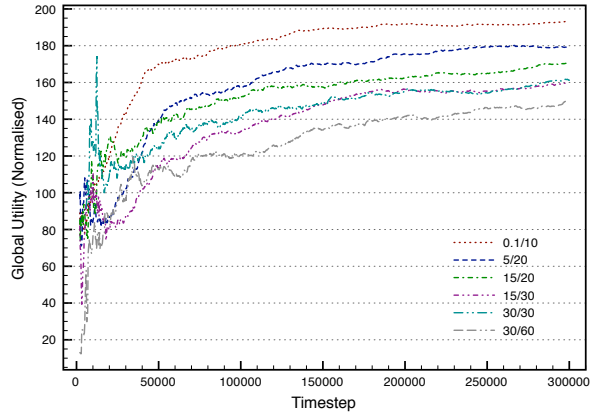
If we look more in detail to the distinct contribution of reading time intervals we can conclude that increasing the reading time (while the writing time remains fixed) always reduces the performance with all the metrics considered. However, if we increase the writing interval (while the reading one remains fixed) we can observe that while the global utility metrics still declines precision seems to remain unaffected (see the comparison between 30/30 and 30/60 or the one between 15/20 and 15/30), thus indicating that the reading time is an important part of the performance of the system, and the recall metric also appears to improve its performance. These results could be explained by observing that reducing the writing frequency reduces the number of total utterances flooding the network. This eventually helps the performance in terms of precision and (especially) recall since these metrics divide their values by a similar quantity (total number of utterances received and total number of potentially valuable respectively, see Section 4.5).

Finally, figure 7 shows the average utility obtained at a time step divided by the number of times a node has accessed its cache in order to read new utterances and obtain utility. As the read time is increased, the utility per read also increases. However, increasing the writing time (while keeping the reading time static) reduces the utility received per read, as fewer utterances are created between reading periods, so fewer new utterances are read, reducing utility. Overall it is clear to see that reading every 0.1 seconds (essentially reading an utterance as soon as it is received) results in a very low score per read, and the use of more realistic read and write times delivers much better utility per read.

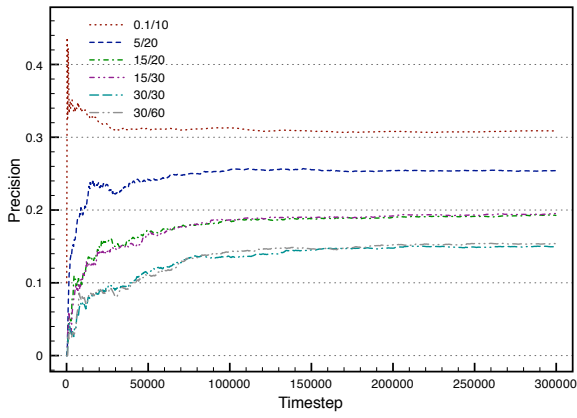
In conclusion, although larger reading and writing times reflects a much more realistic scenario, the performance of the system worsens when these intervals are increased. However, frequent accessing of the cache dramatically reduces the utility received by nodes each time they read updates.

### 5.5. Dependence on the input interest profiles

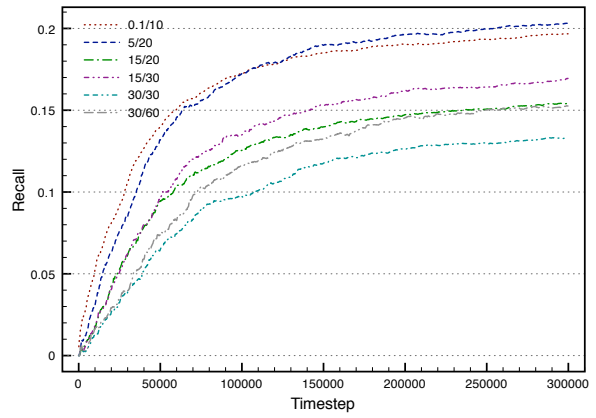
These experiments investigate the effect of using different interest profiles as the input data set. Starting from the default data set derived from Delicious (see Section 4.1) and finding four sub-networks all sharing tags from a global set of 99 distinct tags, we have artificially generated other data-sets presenting different ‘interest overlaps’. Specifically, we have generated one data set in which each of the sub-networks use a completely independent set of tags (so that the global



(a) Global Cumulative Utility (Normalised)



(b) Global Precision



(c) Global Recall

Figure 6: Effect of the Read/Write Ratio

Table 8: Global Precision and Recall

	1 Hop	2 Hops	3 Hops
$rw$	Precision		
	Recall		
0.1/10	<b>0.4297</b>	<i>0.3094</i>	<i>0.3124</i>
	0.1862	0.1968	0.1968
5/20	0.3300	0.2535	0.2536
	<i>0.1918</i>	<b>0.2034</b>	<b>0.1992</b>
15/20	0.25125	0.1926	0.1969
	0.1461	0.1538	0.1549
15/30	0.2428	0.1914	0.1937
	0.1716	0.1712	0.1692
30/30	0.1884	0.1496	0.1561
	0.1226	0.1332	0.1306
30/60	0.1744	0.1514	0.1594
	0.1397	0.1531	0.1554

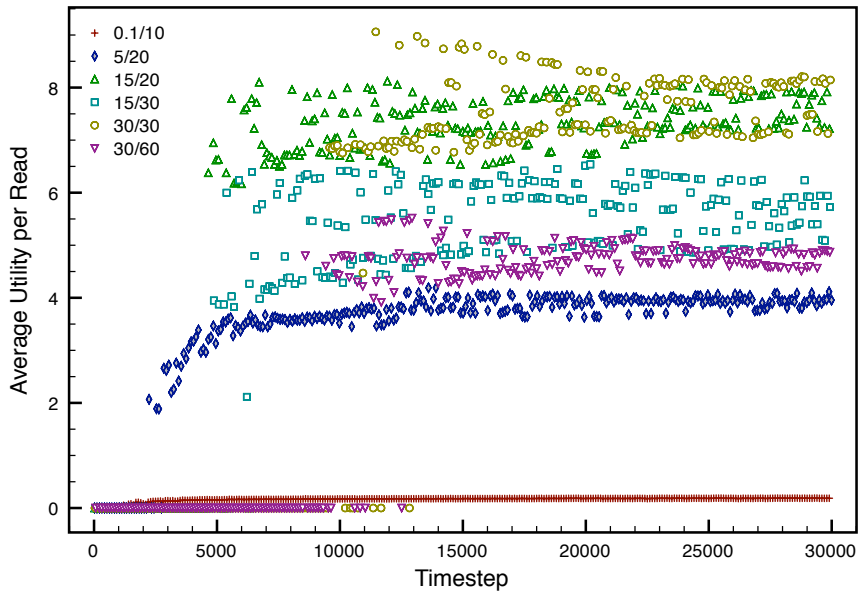


Figure 7: Average Utility per Read- Effect of the Read/Write ratio

Table 9: Effect of the overlapping of distinct Sub-Networks

Number of distinct tags	99	157	196	228	253	312
Global Cumulative Utility	<b>932.30</b>	644.36	477.94	353.11	284.48	151.86
Global Precision	<b>0.3094</b>	0.2215	0.1867	0.1462	0.1216	0.0793
Global Recall	<b>0.1968</b>	0.1895	0.1851	0.1739	0.1638	0.1137

number of tags for the system raises to 312) with no overlapping or sharing of tags between nodes belonging to any pair of sub-networks. In addition we have created intermediate cases, generated by recreating the data set one interest at a time, giving a certain probability of a tag being shared between pairs of distinct sub-networks. As the level of overlap between the sub-networks decreases, the number of distinct interest tags within the network increases.

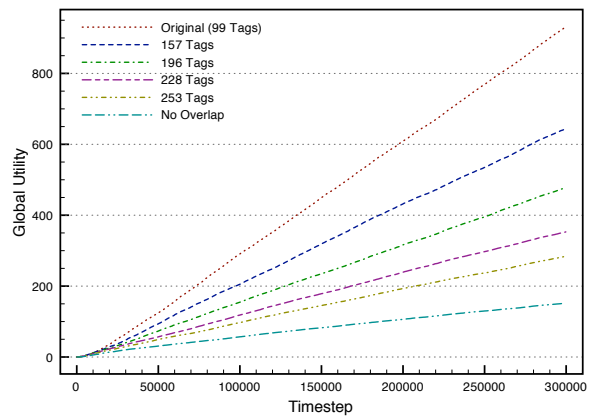
Results in table 9 and figure 8 show that there is a clear decrease in performance in terms of precision and recall with a reduction of the overlapping degree, in particular the precision values drops dramatically (for simplicity we here present only the results obtained with the 2 Hops protocol). A degradation of the performance is observed also in terms of absolute values of the cumulative global utility (see figure 8(a)) that reduces significantly with a decrease in the overlapping of interests between different subnetworks.

However, these results could have been expected by noting that nodes belonging to distinct sub-networks have much less chances to exchange valuable content when they meet when if there is little or no overlapping of the tags composing their interests. In fact, it is true that the mobility model used tends to prioritise movement (and then the number of contacts, the storage and sharing of valuable utterances) within neighbourhoods (which are supposed to reflect the original home sub-networks) but it also happens that nodes often migrate from their home community to a different one, thus losing the initial relation between location of the nodes and similarity of their interest profiles.

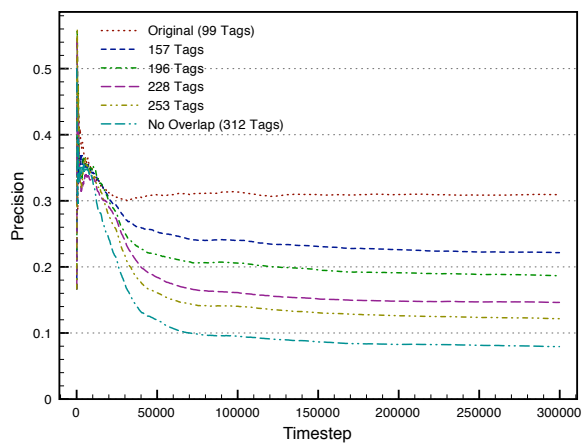
### 5.6. Dependence on the mobility model

We finally examine the protocol performance when different mobility models are used during simulation runs, comparing HCMM to the modified random way point (MRWP) model (see Section 4.2). We here examine the case where all communities share a common set of tags and the case where each community has a completely distinct set of non-overlapping tags (all results refer to the 2 Hops protocol).

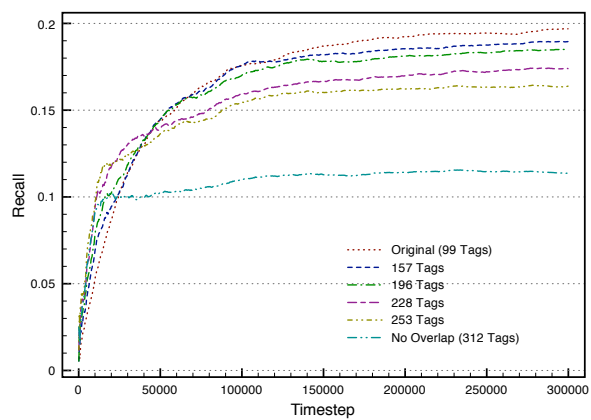
Figure 9 shows that for the case where communities share a common set of interests the performance in terms of utility for both HCMM and MRWP is very similar. However, for the case where there is no overlap of interest tags between communities the performance is very different, with HCMM resulting in a much poorer performance than MRWP. This difference in behaviours was, however, predictable as HCMM encourages many more contacts between different communities than the MRWP model, thus losing the initial relation between location of the nodes and the similarity of their interest profiles. Nodes following HCMM meet a larger number of nodes dissimilar to them to large degrees, thus receiving less utility. The MRWP model restricts nodes to interacting within their own community or at most with nodes of one other community, therefore the majority of other nodes they interact with have a similar interest profile and, consequently, the overall utility is improved.



(a) Global Cumulative Utility



(b) Global Precision



(c) Global Recall

Figure 8: Effect of the Overlapping Networks

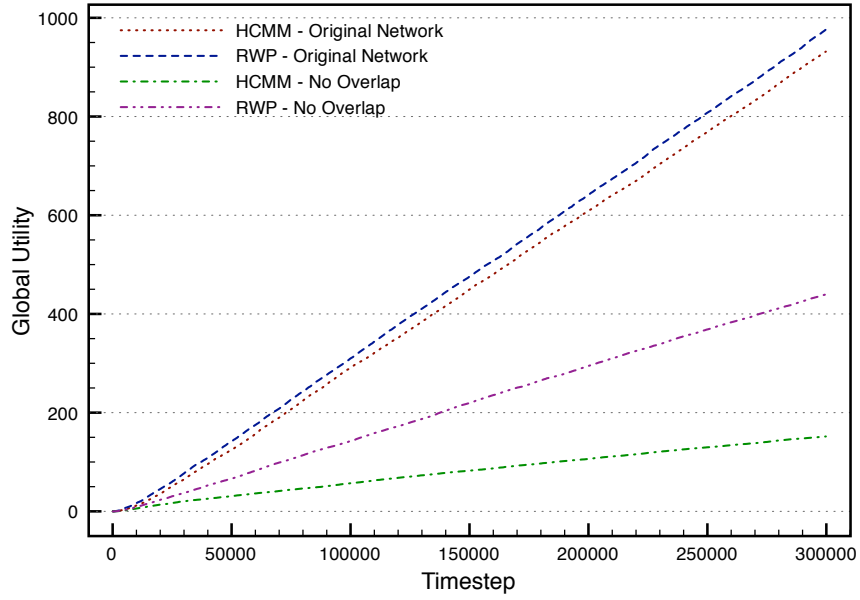
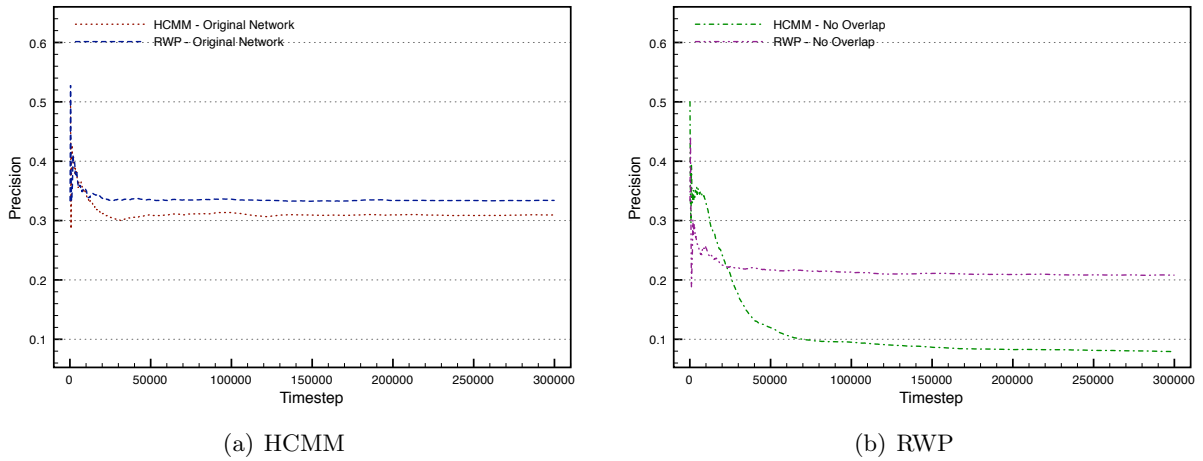


Figure 9: Global Cumulative Utility - Effect of the Mobility Model



(a) HCMM

(b) RWP

Figure 10: Global Precision - Effect of the Mobility Model



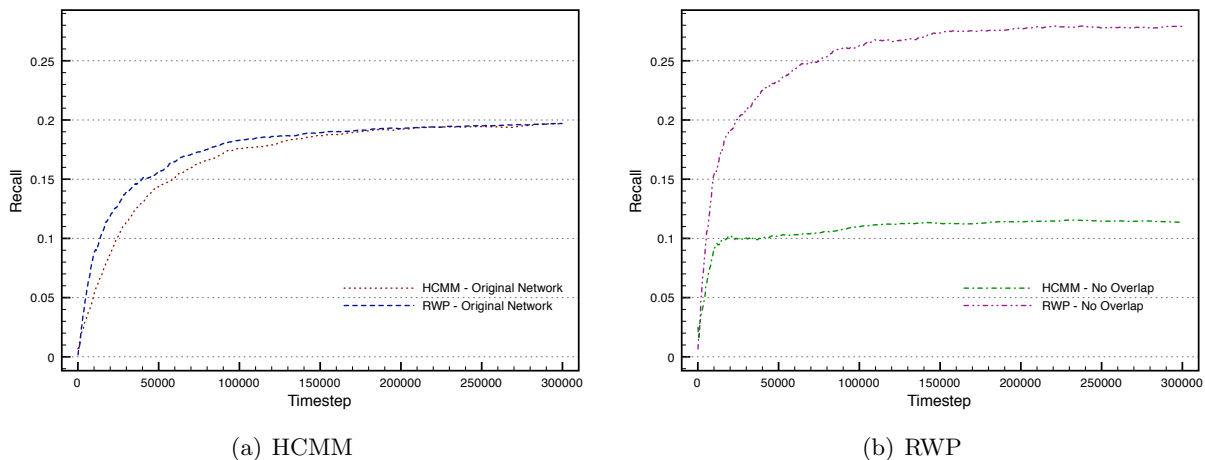


Figure 11: Global Recall - Effect of the Mobility Model

Precision (figure 10) and recall (figure 11) are similarly affected with a comparable behaviour for both mobility models in the case where communities share a common set of tags; and with MRWP outperforming HCMM in the case where each community has a distinct set of tags. Note that when examining the recall metric, it can be seen that the MRWP model results in a significantly better performance with the non-overlapping interest profiles than with the common interest profile (whereas precision drops in this case with both mobilities). This can be explained by the fact that, with this specific data-set the restriction of the MRWP model results in a higher recall, as far more of the tweets received by each node are likely to be produced within its local area.

## 6. Conclusions & Future Work

We have presented a protocol for decentralised data dissemination of low payload resources (utterances or micro-blogs) through a network of mobile agents that participate in opportunistic pairwise interactions. Agents are organised based on similarity (of interests) and familiarity (number of contacts), allowing each of them to receive the most relevant information at each interaction while minimising the (often vast) amount of non-useful data/messages received.

Precision of the utterances exchanged between agents is maximised when they only select and push items to nodes within the same neighbourhood, as the number of items carried through the network is reduced. However, best overall performance considering global utility, precision and recall is achieved when agents push data to nodes outside of their neighbourhood based on the community profile.

Neighbourhoods are defined by the similarity of interests and familiarity of nodes. Higher values of familiarity or similarity (increasing the number of nodes in the neighbourhood) require the storage of large amounts of information and represent an unrealistic scenario. Best overall performance is obtained when neighbourhoods are comprised of nodes that have a level of familiarity or similarity above a given threshold.

Reducing the spatial component of the protocol seems to have a negative effect on the global utility of the system, but when compared to the behaviour of the random benchmark actually represents the best situation. Hence a reduced distance threshold represents a more realistic scenario, corresponding to short range communication and local interactions.

Overall performance is maximised when data items are written and read with short frequencies. Reducing the read/write time to a more realistic timescale (e.g. reading every 30 minutes, writing every hour) reduces the global performance. However, the performance of each of the proposed push protocols (intelligent pushing) remains comparable (no matter what the reading and writing time) and much superior to a sample random scenario in which no specific procedure is followed to flood information over the network.

As interest profiles for each subnetwork composing the input data become less overlapped, the overall system performance worsens, with reductions in the values of global utility, recall and precision. Best performance is achieved when all nodes share a set of common interests, while the worst performance is seen when each sub-network has a completely distinct set.

The protocol produces similar results for two separate mobility models when data with similar profiles of interests are used. When completely distinct interest profiles are used (no-overlapping sub-networks), the protocol delivers better performance in global utility, precision and recall when a restricted random waypoint mobility model is used than when HCMM is used, as the number of nodes with which a node may interact is restricted mostly to those with which a common set of interests is shared.

This model has been shown to work in a theoretical simulation setting, future work could move forward in two phases. The simulation work can be furthered by adapting the mobility model to use real world trace based data, while a further phase of work would be to move forward and implement the micro-blogging scenario in a real world application. Issues such as utterances containing multiple interest tags could also be studied. A further direction would be to study the semantic analysis of utterances.

## Acknowledgements

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

- [1] 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.
- [2] 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.
- [3] 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.
- [4] L. Pelusi, A. Passarella, M. Conti, Opportunistic networking: Data forwarding in disconnected mobile ad hoc networks, Communications Magazine, IEEE 44 (11) (2006) 134–141.
- [5] in Shyan Lee, Y.-W. Su, C.-C. Sh, A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi, in: Proceedings of the IEEE Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference, IEEE, 2007.
- [6] L. Ruiz-Garcia, L. Lunadei, P. Barreiro, I. Robla, A review of wireless sensor technologies and applications in agriculture and food industry: State of the art and current trends, Sensors 9 (6) (2009) 4728–4750.
- [7] A. Stanford-Clark, The house that twitters, telegraph.co.ukdoi:http://www.telegraph.co.uk/science/science-news/6156291/The-house-that-Twitters.html.
- [8] 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.
- [9] 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.
- [10] E. Jaho, I. Stravarakakis, Joint interest- and locality- aware content dissemination in social networks, in: Proceedings of WONS'09, 2009.

- [11] 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.
- [12] 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.
- [13] A. Gupta, A. Kalra, D. Boston, C. Borcea, Mobisoc: a middleware for mobile social computing applications, *Mob. Netw. Appl.* 14 (1) (2009) 35–52.
- [14] S. Schulz, K. Herrmann, R. Kalcklösch, T. Schwotzer, Takeup: Trust-based agent-mediated knowledge exchange for ubiquitous peer networks, in: *AMKM*, 2003, pp. 89–106.
- [15] T. Schwotzer, K. Geihs, Shark a system for management, synchronization and exchange of knowledge in mobile user groups, in: *Journal of Universal Computer Science*, 2002, pp. 149–156.
- [16] L. Fiege, A. Zeidler, F. C. Grtner, S. B. Handurukande, Dealing with uncertainty in mobile publish/subscribe middleware, in: *In 1st International Workshop on Middleware for Pervasive and Ad-Hoc Computing*, 2003, pp. 60–67.
- [17] 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, pp. 225–234.
- [18] J. Wang, J. Cao, J. Li, Supporting mobile clients in publish/subscribe systems, in: *ICDCSW '05: Proceedings of the First International Workshop on Mobility in Peer-to-Peer Systems*, IEEE Computer Society, Washington, DC, USA, 2005, pp. 792–798.
- [19] J. Wang, B. Jin, J. Li, An ontology-based publish/subscribe system, in: *Middleware '04: Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware*, Springer-Verlag New York, Inc., New York, NY, USA, 2004, pp. 232–253.
- [20] 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.
- [21] A. Lindgren, A. Doria, O. Schelén, Probabilistic routing in intermittently connected networks, *ACM SIGMOBILE Mobile Computing and Communications Review* 7 (3) (2003) 19. doi:10.1145/961268.961272.
- [22] M. Meisel, V. Pappas, L. Zhang, Ad hoc networking via named data, in: *Proceedings of the fifth ACM international workshop on Mobility in the evolving internet architecture*, ACM, 2010, pp. 3–8.
- [23] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, R. Braynard, Networking Named Content, in: *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, ACM, 2009, pp. 1–12.
- [24] 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.
- [25] A. Datta, S. Quarteroni, K. Aberer, Autonomous gossiping: A self-organizing epidemic algorithm for selective dissemination in wireless mobile ad-hoc networks, in: *Proceedings of ICSNW 2004*, World, 2004, pp. 126–143.
- [26] T. Spyropoulos, K. Psounis, C. S. Raghavendra, Spray and wait: an efficient routing scheme for intermittently connected mobile networks, in: *WDTN '05: Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, ACM Press, New York, NY, USA, 2005, pp. 252–259.
- [27] S. Golder, B. A. Huberman, The structure of collaborative tagging systems, *Journal of Information Science* 32 (2).
- [28] D. Ramage, P. Heymann, C. D. Manning, H. Garcia-Molina, Clustering the tagged web, in: *Proceedings of WSDM '09*, ACM, New York, NY, USA, 2009, pp. 54–63. doi:http://doi.acm.org/10.1145/1498759.1498809.
- [29] M. K. Denko, J. Tian, Cooperative caching with adaptive prefetching in mobile ad hoc networks, in: *WIMOB '06: Proceedings of the 2006 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, IEEE Computer Society, Washington, DC, USA, 2006, pp. 38–44.
- [30] H. Shen, M. S. Joseph, M. Kumar, S. K. Das, Precinct: A scheme for cooperative caching in mobile peer-to-peer systems, in: *IPDPS '05: Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05) - Papers*, IEEE Computer Society, Washington, DC, USA, 2005, p. 57.1.
- [31] J. Wang, C.-H. Lai, Detecting groups of similar components in complex networks, *New Journal of Physics* 10.
- [32] I. X. Y. Leung, P. Hui, P. Liò, J. Crowcroft, Towards real-time community detection in large networks, *Physical Review E* 79 (6).
- [33] 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.
- [34] 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.
- [35] M. Musolesi, C. Mascolo, A community based mobility model for ad hoc network research, in: *REALMAN '06: Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, ACM, New York, NY, USA, 2006, pp. 31–38.
  - [36] W. B. Frakes, Introduction to information storage and retrieval systems, in: *Information Retrieval: Data Structures & Algorithms*, 1992, pp. 1–12.
  - [37] N. Baker, Zigbee and bluetooth - strengths and weaknesses for industrial applications, *Computing and Control Engineering* (16) (2005) 20–25.