Why Neighbourhood Matters: Interests-Driven Opportunistic Data Diffusion Schemes

Iacopo Carreras, Francesco De Pellegrini, Daniele Miorandi,
David Tacconi and Imrich Chlamtac
CREATE-NET, v. alla Cascata 56, 38100 – Povo, Trento (Italy)
name.surname@create-net.org

ABSTRACT

Opportunistic communications have been extensively studied in the last few years, with special emphasis on the optimal trade-offs between the resources needed to operate such systems and the performance experienced by the end-users. We focus in this work on an application scenario in which users exploit opportunistic communications to exchange and diffuse data of potential interest. We assume a data-centric architecture, in which users would like to receive messages whose content matches a set of interests they expressed. We propose a data diffusion scheme based on the combination of two features: (i) interest–driven one–hop data exchange (upon a transmission opportunity, a node requests data matching a set of interests) (ii) neighbourhood–based data conveyance (each node maintains a list of the most interesting contents for its nearest neighbours and looks actively for such kind of data). An algorithmic solution is proposed, together with a protocol implementing it. The performance of the proposed scheme is evaluated by simulating the data diffusion process, exploiting contact traces and interests lists gathered through real–life experiments.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]:

General Terms

Design, Algorithms, Experimentation

Keywords

Mobile computing, opportunistic communications, user interests, data-centric architecture, experimental measurements

1. INTRODUCTION

Opportunistic communications systems have recently gained attention from the research community [11, 13]. Such interest has been fostered by the increasing adoption, in various electronic devices (e.g., mobile phones, gaming consoles, MP3 players etc.) of some form of proximity wireless communications, such as Bluetooth and/or WiFi. This has the potential to pave the way to a future in which users can produce, access, share and consume digital resources (content, services) without the support of a fixed infrastructure [7]. Clearly, such systems are not meant to represent a replacement to infrastructure–based solutions, but, rather, to complement them, by building "computing clouds" able to work at the edges of existing infrastructure.

Opportunistic communication systems are known to require —among the other things— a given level of cooperation in order to perform efficiently. This is related mostly to the need of having nodes relaying packets not destined to themselves. This is clearly essential in address-based architectures, where opportunistic networking systems are leveraged to support end–to–end communications. This is the case of many delay-tolerant network deployments, which are aimed at supporting IP packets relaying. Notwithstanding, there is a variety of application scenarios for which a different approach, based on a data–centric architecture, is more appropriate. This is especially true in scenarios where opportunistic communications are used to diffuse information that might be potentially of interest for many, possibly unknown, users. Data–centric architectures may alleviate the problems related to the need of cooperation, but do not solve the problem. In such systems, cooperation plays a twofold role. First, the willingness of a node to share its data with other nodes. Second, the willingness of a node to gather and carry data which do not match its set of interests. Concerning the first point, we assume that nodes adopt a fully cooperative behavior, i.e., they are always available to share their data. In this work, we focus on the second aspect, and devise algorithmic solutions that are able to leverage certain features of the contact patterns among users in order to reach a good trade-off between user-perceived performance and resource consumption.

Our approach is based on a set of experimental measurements carried out during early 2008. The experiments we performed spanned four weeks, and consisted of 21 users (research and administrative staff at our institution) carrying around a mobile device, running a custom software tracing number and duration of contacts (registered through the Bluetooth interface). This was complemented by a set of questionnaires aimed at understanding the potential interests of users in given categories of data (including, e.g., music, news, cinema etc.). The data gathered were used

CHANTS'08, September 15, 2008, San Francisco, California, USA.

to build a Java-based trace-driven simulator, which reproduces the diffusion process of data of various types (in terms of content) over the system (accounting for real parameters like duration of contacts, amount of data transferred during a contact, discovery time etc.).

In particular, we were interested in understanding whether the social network the user is part of (exemplified by a set of "nearest neighbours") can be leveraged to increase the system performance. In such respect, we introduced a class of opportunistic data diffusion algorithms, which we call k–neighbours–based relaying. The idea is the following one. Each node selects a set of k "nearest neighbours", based on the contact pattern (number of contacts within a given timeframe and their duration). It then builds a list of interests by merging its own ones with those of its nearest neighbours (properly weighted and/or ordered). Upon a meeting, it communicates such list to the other node, asking for data matching such interests. In this way, it aims at receiving both the content it is interested in and the content its nearest neighbours are looking for. As nearest neighbours are nodes with which frequent/long transmission opportunities occur, the net effect is to build "fast" paths (mainly two–hop ones) between the content sources and the users interested in it. Performance is measured in terms of two parameters. The first one is represented by the utility perceived by the user, measured as the fraction of interesting content available in the system the user actually receives. The second one is a measure of the overall system efficiency, expressed as the average ratio between the number of nodes which receive a message and the number of nodes potentially interested in that content. Simulation results are presented, showing that k–neighbours relaying schemes are able to offer an interesting trade–off between utility and efficiency across a variety of traffic loads.

Related work

In data–centric architectures for opportunistic communications, the reference forwarding scheme is represented by epidemic routing [14, 6]. This is rooted in the ease of implementation and robustness with respect to network conditions typical of such schemes. Epidemic routing share many principles with controlled flooding [4], which has been extensively described through fluid approximations and infection spreading (see for example [15]). The control of forwarding has been addressed in the ad-hoc networks literature, e.g in [9] and [3]. In [9], the authors describe an epidemic forwarding protocol based on the *susceptible-infected-removed* (SIR) model [15]. Authors of [9] show that it is possible to increase the message delivery probability by tuning the parameters of the underlying SIR model.

Notice that in the case of end-to-end opportunistic networks, the target is to deliver messages with high probability to the intended destination: all message copies stored at nodes other than the destination represent overhead. Conversely, in the case of data centric forwarding, the trade-off involves a notion of *utility*, because not all message copies are redundant, as they can be relevant for a variety of users. In [2], this trade-off has been explored by applying a *publish-subscribe* paradigm, where each user takes into account other users "subscriptions" for making appropriate data caching decisions. However, due to the overhead related to the regular update of users interests, such approach is not viable for large-scale and extremely dynamic systems.

As such systems depend heavily on the user behaviour (in terms of contact patterns as well as set of interests), various groups have recently started investigating the impact of social aspects on system design and performance [5, 10, 12, 8]. In particular, the structure of the social network the user is in was proved to significantly influence the performance of the system. Starting from this observation, social interactions have been accounted for in order to design opportunistic forwarding and data diffusion schemes. In [5], the community structure behind the social interactions has been studied in order to improve forwarding algorithms. The authors showed that there exists a limited set of nodes, called *hubs*, which play a central role in the diffusion of information. Being aware of the community structure, in [5] the authors showed that an extremely efficient trade-off between resources and performance can be achieved.
In [8], the impact of different social-based forwarding schemes were evaluated in the case of a DTN routing protocol. Similar to our work, real world mobility patterns were obtained from Bluetooth proximity measures. The authors of [8] showed that incorporating a friend/strangers classification in the forwarding policies can be beneficial in different application scenarios.

In this paper, the novelty is that we combine the user preferences (type of content the user is interested in) with the social structure the user is in. Our evaluation is based on trace-driven simulation, performed using traces obtained from real–world experiments

The paper is organized as follows. Section 2 describes the main features of the opportunistic communication system used in the tests. Section 3 describes the general application simulated, whereas in Section 4 we discuss the performance figures of interest for the system at hand and introduce an algorithmic solution. Section 5 reports on the numerical results. The last section is devoted to concluding remarks.

2. EXPERIMENTAL SETTING AND METHODOLOGY

Given the central role played by users in the considered application scenarios, we believe it is important to reproduce a realistic setting. This allows also to incorporate in the performance evaluation the constraints coming from (i) current technology, in terms of, e.g., amount of data transferred upon a contact opportunity, discovery times etc (ii) preferences expressed by users. We then run an experimental measurements campaign involving 21 users. This experiments, although limited in size, allow us to explore both the "social" and "technological" aspects of the application scenario under evaluation.

2.1 Contact Tracing

Similarly to the experiments conducted in [12, 1], we have monitored people's encounters by tracing their proximity for a 4 weeks period. During the experiment, 21 workers — playing different roles within our organization and working on different floors of the same building — were asked to carry a mobile phone running a java application, and relying on Bluetooth connectivity for exchanging data. The application periodically triggers (every 60 seconds) a Bluetooth node discovery. Whenever another device is detected, its Bluetooth address, together with the current timestamp is saved in the permanent storage of the device for a later

processing. Nodes are using the phone's internal clock for determining the timestamp. Since SIM cards are inserted in the phones, it is possible to synchronize such clock to the GSM network. This ensures a sufficient level of precision, especially when compared to the granularity of the peer discovery process.
A fixed device (a Bluetooth–enabled laptop) is used for gathering data from users passing by and transmitting it to a remote repository. The final outcome of the experiment is a unique trace storing a series of contacts. Each contact is characterized by a timestamp, by the ID of the nodes involved and by the duration of the contact. In Tab. 1 we reported a summary of the experimental traces collected.

Participants	21
Experiment Duration	4 weeks
Registered Contacts	14100
Discovery Period	60 s

Table 1: Summary of the experimental settings.

Tab. 2 summarizes the main features in terms of contact duration and inter–meeting times (expressed in terms of mean, standard deviation, minimum and maximum value). It can be easily observed that most of the meetings are relatively short in duration, lasting no more than 100 sec., although few contacts persist for a very long time. This is due to users not carrying the mobile phone with them, and leaving it on their desk.

Metric	Mean (s)	Std. (s)	Min (s)	Max (s)
Inter–meeting	24156	101588	90	1812085
Contacts Duration	473	1047	45	55717

Table 2: Intermeeting time and contacts duration characteristics (expressed in seconds): mean, standard deviation, minimum and maximum.

2.2 User interests

In opportunistic content distribution and sharing applications, users' interests play a very important role. We decided therefore to couple our study on the movement and contact patterns of people with a set of questionnaires aimed at capturing users' preferences and their potential interests in digital contents. In particular, we asked each user involved in the experiment to fill a questionnaire, consisting of 10 questions addressing their preferences in terms of content categories (i.e., music, cinema and news) of the information they could access and distribute over such system. This provides us a feedback on the categories of common interests. A brief summary of the questionnaires results is reported in Tab. 3.

As it can be seen, a large level of heterogeneity is present in terms of users' preferences. The natural question is then wheter this knowledge can be leveraged in order to design smart algorithms, able to increase the system performance.

3. SYSTEM MODEL AND PROBLEM FORMULATION

In this work, we consider an opportunistic content sharing and distribution system, where users download (from an unspecified destination that could be the Internet or some dedicated base stations) daily content such as, e.g., music, news, on their mobile phones. Such information can be then exchanged between users when they happen to be in proximity. When an encounter occurs, depending on the specific data diffusion scheme that is applied, users can exchange all the data available on their phone, or keep only the one that matches their interests and forward the other to other users that might be interested, or they can only exchange the data they are interested in.
Each user has a list of well–defined interests, which is specified through a suitable application interface. Data is exogenously injected into the system by some fixed devices. Each data unit (from now on called also, with a slight abuse of terminology, a message or an item) is marked with some metadata describing its content. The metadata associated to a message and the user's interests are expressed using a common language (as an example, standard ontology languages could be used).

For the sake of simplicity, we assume in the following that each message is marked with one single tag, describing its content (in principle, a message could belong to various categories). The set of possible categories is assumed to be finite and is denoted by Ψ. Each message has also a time–to–live field, expressed in seconds, which determines the persistence of the message in the system.

Summarizing, each item c injected into the network is characterized by:

1. the item's category; we use the notation $\psi(c) \in \Psi$ addressing the category of item c;
2. the item's *content generation time*, which represents the generation time of the specific content item;
3. the item's *time to live* (TTL), which determines the temporal validity of such item (in seconds), starting from its content generation time.

Each user is characterized by:

- a set of interests $\varphi_i \subseteq \Psi$, specifying the content user i is interested in;
- an interests vector, which is a list of pairs $<$ *category*, *weight* $>$ where the *category* represents a specific interest and the *weight* is a numerical value in the range $(0, 1]$ expressing the level of interest. All categories represented in φ_i are present in the interest vector with unitary weight.

Data is exchanged upon meetings following a pull–based technique: when node i meets node j, it sends a query message containing its interests vector. Node j then checks the availability of such type of content in its internal buffer and transmits the items matching the received query, in decreasing order of weight.

Two performance metrics are used. First, we define a network-wide *utility* figure aimed at capturing the perceived quality of service by the users. This accounts for the ability of the mechanism to deliver the content present in the system to the users interested in it. Of course, we also need to account for the resources spent in order to deliver such contents: in order to do so, we introduce another performance figure, which we refer to as *efficiency*, aimed at describing the ability of the proposed mechanism to limit the usage of

Cinema							
Thriller	Science fiction	Dramatic	Romantic	Comedy	Horror	Documentary	Italian
19%	14%	14%	10%	15%	9%	11%	7%
Music							
Etnic	Rock	Pop	Disco	Rap/ Hip-Hop		Jazz	Classic
13%	18%	16%	12%	10%		15%	14%
News							
Meteo	Politics	Chronicle		Economy	Sport		Culture
11%	21%	18%		16%	14%		15%
28.95%	36.85%	28.95%		45%		21%	34%

Table 3: Users preferences expressed for the cinema, music and news content categories.

resources. For the application scenario under consideration, indeed, the optimal operating condition is to deliver each content item *only* to those users interested in it. This ensures that any content injected into the network will reach the intended (= interested) destinations, and that no redundant messages are generated, thus minimizing the resources that are used for running the opportunistic service. We refer to this as the *ideal* case with obvious meaning. For each content category ψ we define set S_ψ as the set of nodes which are interested in category ψ.
Now, when item c is injected in the network, over time it will reach a set I_c of the network nodes. At any instant in time we denote by S_c the set of infected nodes being interested in the content category $\psi(c)$: $S_c = S_{\psi(c)} \cap I_c$.
Given a set of M messages, we define the following figures:

- *Network Infection Ratio* (NIR): measures the average fraction of infected nodes per message, namely

$$NIR = \frac{1}{M}\sum_{i=1}^{M}\frac{|I_{c_i}|}{N}, \qquad (1)$$

where $|\cdot|$ identifies the cardinality of a set and N denotes the number of users in the system.

- *Utility* (U): measures the average number of *interested* nodes infected by a message, i.e.,

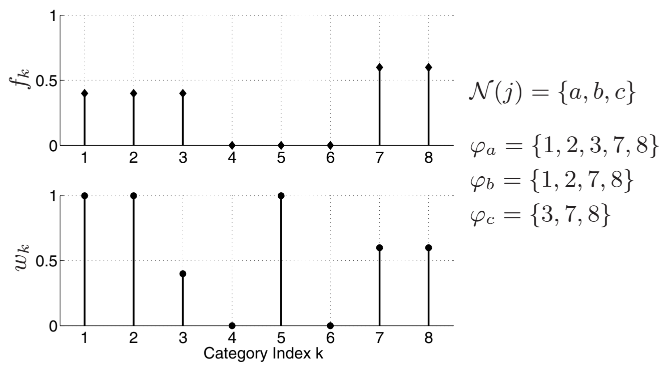
$$U = \frac{1}{M}\sum_{i=1}^{M}\frac{|S_{c_i}|}{|S_{\psi(c_i)}|} \qquad (2)$$

- *Efficiency* (E): measures the tradeoff between message delivery and resources consumption. It is defined as the average ratio between the fraction of interested nodes infected and the fraction of overall nodes infected. In symbols:

$$E = \frac{1}{M}\sum_{i=1}^{M}\frac{|S_{c_i}|}{|S_\psi(c_i)|}\frac{N}{|I_{c_i}|} \qquad (3)$$

In order to characterize this metrics, we start noticing that $|S_{c_i}| \leq |S_{\psi(c_i)}|\ \forall c_i$ so that $0 \leq U \leq 1$. Furthermore, as $\frac{|I_{c_i}|}{N} \leq 1\ \forall i$, $U \leq E$. Equality holds only if, before the TTL timer expires, any message c_i gets received by all nodes.
In the ideal case $|S_{c_i}| = |S_{\psi(c_i)}|$, so that $U = 1$. We also notice that, $S_{c_i} = I_{c_i}$, so that from (3) the maximum efficiency is

$$E_{\max} = \frac{N}{M}\sum_{i=1}^{M}\frac{1}{|S_\psi(c_i)|} \qquad (4)$$



Sorted list of categories sent to i: $O = (1, 2, 8, 7, 3)$

Figure 1: Example of construction of the category weights at node j. For example, the second item stored belongs to category 1; list O represents the list of categories of items received at i.

From the definition of the above metrics, it is clear that in order to improve the efficiency, nodes should forward data c_i only to nodes in $S_{\psi(c_i)}$. Nevertheless, this severely limits $|I_{c_i}|$, especially when data has a finite lifetime (as this hinders the possibility of exploiting relaying), potentially harming the utility figure.

4. ALGORITHMS AND PROTOCOLS

In this section we propose an algorithmic approach aimed at trading off utility for redundancy with the aim to increase the efficiency. The key intuition is rather simple: we leverage local exchange of data among nodes which have stronger encounter patterns. We call them *nearest neighbours*. We also account for the categories that are mostly represented in such set of nodes and try to forward first data that correspond to more popular categories. This technique has two appealing features from a distributed implementation standpoint: (i) it is receiver based, since the filtering is operated at the receiver side; (ii) only local information on each node encounter pattern is needed.

4.1 Algorithmic description

Our proposed scheme is based on two distinct procedures that run in parallel: the first one is used to build the interest vector; the second procedure leverages a receiver–driven diffusion scheme for data exchange.
Nearest neighbours discovery: this procedure builds an es-

timate of the set of nearest neighbours $\mathcal{N}(i)$ of a node i. Any node in the network traces the nodes met during a given time window and the length of the respective contact. Nodes are then sorted according to the sum of the contact durations, and the first k of them are considered as nearest neighbours. The parameter k is used to control the utility/efficiency tradeoff. For $k = 0$ we limit the usage of resources, but relaying is not exploited. Larger values of k lead to the use of larger amount of resources, but potentially increase the utility figure as well. It is worth noticing that we detect nearest neighbours on the basis of cumulative contacts duration, rather than on contacts frequency. This is motivated by the fact that, for the scenario under consideration where nodes may be expected to exchange large amount of contents (e.g., images, multimedia files, etc.), the contacts duration better reflects the possibility to exchange data between any pair of nodes.

As a second step, this procedure evaluates the relative frequency of the *categories* within $\mathcal{N}(i)$, $\varphi_{\mathcal{N}(i)} = \cup_{i\in\mathcal{N}(i)} \varphi_i$, building the histogram of such categories with respect, i.e., measuring the relative frequency f_ψ of interest in ψ within $\mathcal{N}(i)$:

$$f_\psi = \frac{|S_\psi \cap \mathcal{N}(i)|}{|\varphi_{\mathcal{N}(i)}|}, \quad \psi \in \varphi_{\mathcal{N}} \tag{5}$$

Forwarding based on interests vector: the second procedure is used to drive opportunistic forwarding as follows. Once two nodes meet, after a customary discovery phase, nodes exchange their interest vector. The interest vector contains a set of categories with an associated weight. The weights are determined at node i as follows. For category ψ, we have:

$$w_\psi = \begin{cases} 1 & \text{if} \quad \psi \in \varphi_i; \\ f_\psi & \text{if} \quad \psi \in \varphi_{\mathcal{N}(i)} \setminus \varphi_i; \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

An example of the construction of the category weights is depicted in Fig. 1, where weights are calculated based on the set of nearest neighbours $\varphi_{\mathcal{N}} = \{1, 2, 3, 7, 8\}$ relative to nodes $\mathcal{N}(j) = \{a, b, c\}$.

After receiving the interest vector, the node starts sending the matching items present in its internal buffer. This is done by following (deterministically) the weight associated to each category. Items whose category is ranked higher are transmitted first. If multiple items stored belong to the same category, the order of transmission among them is sorted according to a LIFO principle. Of course, an item c such that $f_{\psi(c)} = 1$ will be transmitted first. In the example depicted in Fig. 2, node j starts transmitting two items belonging to category e. When all such items have been transmitted, node i will start forwarding sequentially items c such that $0 < f_{\psi(c)} < 1$ (in this case, message 3, which belongs to category a). The process lasts until either all data of interest are transmitted, or the contact ends. We notice that, due to the finite duration of a contact, it may happen that some matching items (with lower weight) will not be transmitted.

4.2 Protocol Implementation

The aforementioned scheme has been implemented in a java–based middleware. All transactions reported in Fig. 2 have to be understood at the application level. Batches of messages are passed to the Bluetooth L2CAP protocol. Piggybacking is employed to maximize the system performance.

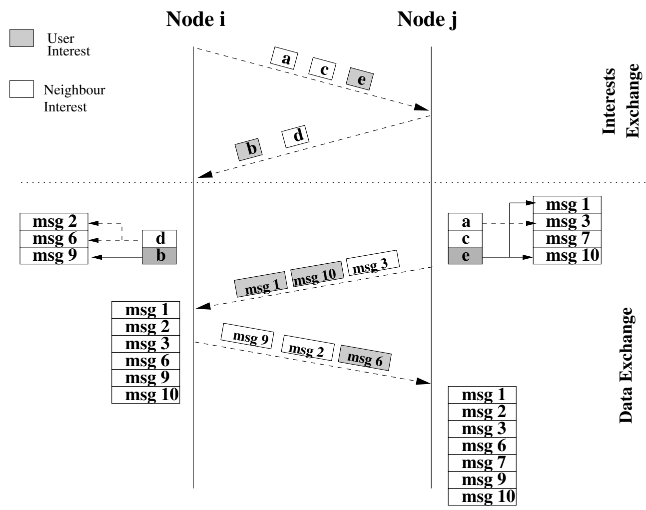


Figure 2: Graphical representation of the data flows in the proposed scheme.

At each node, data is maintained in a buffer, where new data is added on the head. Upon meeting a node, a routine is triggered to eliminate from the buffer the messages whose TTL has expired.

Upon meeting a node, a message containing the node's own interests is built and transmitted to the neighbouring node, using historical data on meeting frequency and duration (`neighContactTime[]` in Alg. 1). Upon receiving an interest vector, the set of data to be transmitted is created by sorting the data in the buffer as follows. First, the categories with the highest weight in the interests vector are selected. The set of available data is scanned; if a positive match is found, the corresponding item is sent to the L2CAP. Once the buffer has been scanned, the set of categories corresponding to the second highest weight are considered. The procedure is repeated iteratively until either all entries in the interests vector have been considered or the contact has ended (Alg. 2).

Algorithm 1 Interests Generation and Transmission

```
Require: neighContactTime[ ]
neighbours ← updateNeighbours(neighContactTime[ ]);
interestsVector ← [ownInterests,
                   evaluateInterests(neighbours)];
sendInterests(interestsVector);
```

Algorithm 2 Messages Transmission

```
Require: dataBuffer[ ]
queryVector ← receiveInterests( );
contVector ← sort(dataBuffer,queryVector);
for i = 0 to contentVector.size do
   send(contVector[ i ]);
end for
```

5. EXPERIMENTAL EVALUATION

In order to evaluate the performance of the proposed algorithm and protocols, we have developed a simulator that

reproduces the opportunistic diffusion of contents, exploiting the contact trace and users preferences collected during the first phase of this work and described in Sec. 2.
The simulator, at startup phase, loads the users, together with their interests, and then processes one content at a time, as measured during the field experimentation. At regular rate, new content items are injected into the network. Each content item is characterized by (i) a specific category, which is determined starting from the interests of the user receiving that item, (ii) a Time To Live (TTL), which determines the time validity of such item, starting from the instant when such item is injected into the network (iii) a specific format, which specifies the size (expressed in bits) of the item being exchanged and, therefore, the time that is needed for exchanging it given the bluetooth data rate. With this respects, we have run separate measurements for evaluating the bluetooth performance that is possible to experience through the $JSR82$ java APIs for bluetooth. Measurements show that is possible to obtain up to 600 Kb/s for sufficiently long data transfers, and an average of 25 s service discovery time. Once a content is expired, it is not diffused any longer by the nodes. At each contact, nodes exchange messages according to the protocol described in Sec. 4.2, and applying the interests-driven forwarding mechanism.
The performance of the system is evaluated according to the metrics introduced in Sec. 3, *Network Infection Ratio* (NIR), *Utility* and *Efficiency*, and averaging over 20 runs. Results are reported in terms of 95% confidence intervals. In order to randomize simulations and obtain sufficiently tight confidence intervals of the results, each run starts at a random time instant, uniformly distributed over the entire duration of the contact trace. The trace is then iteratively reproduced until all messages injected in the network have expired.

Data Format	Audio ($5Mb$ of size and $90s$ of transfer time)
Data TTL	48 h
Number of Subcategories of Interest	6 and 2 (out of 21)
Diffusion Mechanisms	flooding, k–neighbours
Message Rate	1–50 msgs/h

Table 4: Summary of the experimental settings: audio contents, with a $48h$ TTL are emulated applying the flooding and k–nearest neighbours–based diffusion mechanisms.

As a first experiment, we have assumed audio contents, with $5Mb$ size and $90s$ of transfer time, to be injected into the network with a $TTL = 48h$. All users are assumed to be interested in all the three considered categories (music, news, cinema). For each category, each user is assumed to be interested in two sub-categories (e.g., $music \rightarrow \{rock, jazz\}$). Totally, each user is interested in 6 types of content among the possible 21 subcategories (see Tab. 3). Users are assigned to categories according to the outcomes of the questionnaires, as reported in Tab. 3. We have evaluated the case of flooding and k–nearest–neighbours based diffusion for various network loads (measured in messages/hr).

For this first experiment, the NIR results are depicted in Figs. 3. As intuitively clear, the highest NIR is obtained by the flooding mechanism. In this case, no specific policy is applied to the content diffusion process, and nodes exchange

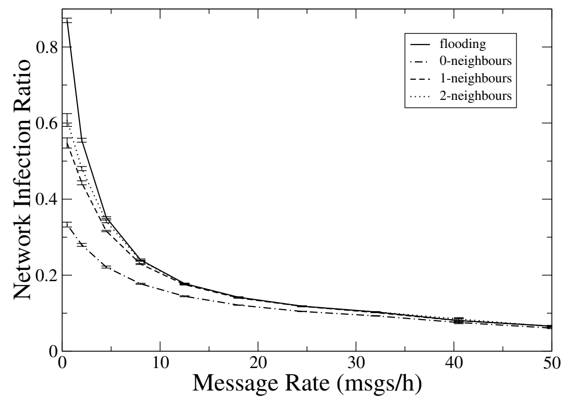


Figure 3: *Network Infection Ratio* in the case of flooding and k–nearest–neighbours–based diffusion. Contents are injected at a variable rate, and with an expiration time equals to $TTL = 48h$. Each user is interested in 6 subcategories (out of 21).

all the data stored in their buffers, being constrained only by the limited contact duration. The NIR of the interest–based diffusion mechanism depends on the size of the neighbourhood. The case of 0–neighbours corresponds to a node behaving in a selfish manner, storing and exchanging contents that are relevant to its own interests only. For larger values of the neighbourhood size, nodes start to work in a cooperative manner, exchanging, with lower priority, messages destined not only to themselves, but also to their nearest neighbours. This is reflected in a higher value of the NIR. Given the limited size of the experimental trace, further increases of the neighbourhood size do not vary significantly the performance. For all cases, the NIR decreases as the message rate increase. This is due to the limited contact duration (in this work we are not assuming any constraint on the nodes buffers).

Fig. 4 presents the Utility results for the same settings as before. As expected, flooding performs best in the presence of very low loads, but its performance degrades quickly, and k–nearest–neighbours scheme takes over. In this case, exchanging messages according to the users interests provides a better Utility, since this metric takes into account how many interested users a message with a specific category was able to reach. Clearly, increasing the size of the neighbourhood corresponds a less selfish behavior and provides a better utility.

In Fig. 5, the resulting system efficiency is plotted for the same settings. The efficiency reflects how well the system behaves in delivering contents to interested users only. In this respect, the flooding is outperformed by the nearest neighbours–based scheme, independently from the size of the neighbourhood. This is due to the completely blind diffusion mechanism, which propagates messages to any node, independently from its interests. The nearest neighbours–based diffusion mechanism efficiency changes depending on the size of the neighbourhood and of the message rate. For low network loads, a limited neighbourhood size leads to a better efficiency. Under such conditions, the benefit offered in terms of utility is not sufficient to pay for the increase us-

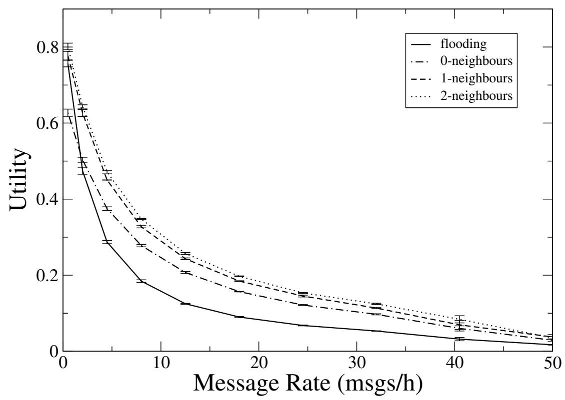


Figure 4: *Utility* in the case of flooding and k–nearest–neighbours–based diffusion. Contents are injected at a variable rate, and with an expiration time equals to $TTL = 48h$.

age of resources. However, the beneficial impact of nearest neighbours become evident for message rates greater than 10 msgs/h. In this case, diffusing messages also for nearest neighbours allows nodes to maximally exploit any contact opportunity, and the larger the size of the neighbourhood, the greater the gain (at least up to $k = 2$).

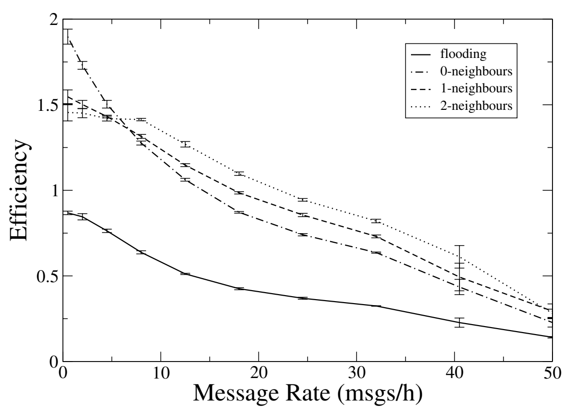


Figure 5: *Efficiency* in the case of flooding and k–nearest–neighbours–based diffusion. Contents are injected at a variable rate, and with an expiration time equals to $TTL = 48h$.

As a second experiment, we have assumed nodes to be interested in 1 category of content only: either music, cinema or news. This corresponds to increasing the "diversity" of users interests, and therefore to reducing the likelihood that any 2 nodes meeting will share a common interest. As before, each node is interested in 2 sub-categories. In this case, each node is interested in 2 out of 21 possible types of content. The assignment of (sub-)categories to nodes is done again by exploiting the data gathered through the questionnaires.
Fig. 6 presents the NIR results for this second case. As before, the flooding is able to maximally diffuse a message over the global network, outperforming k–nearest neighbours–based diffusion schemes.

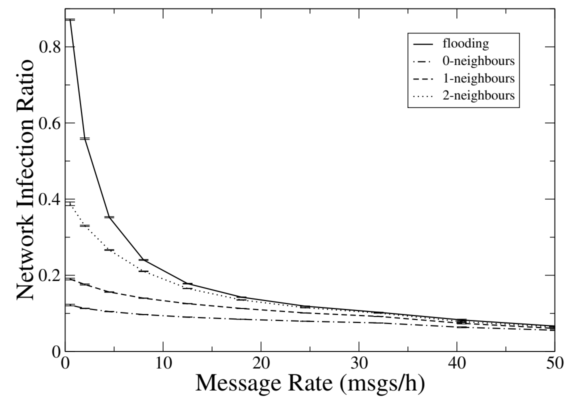


Figure 6: *Network Infection Ratio* in the case of flooding and k–nearest–neighbours–based diffusion. Contents are injected at a variable rate, and with an expiration time equals to $TTL = 48h$. Each user is interested in 2 subcategories (out of 21).

Fig. 7 presents the utility for this second setting. Also in this case, the interests-based diffusion with 2 neighbours performs best, but in this case the gain is much higher. This is due to the limited number of encounters with nodes sharing similar interests. In this case, a purely selfish behavior (neighbourhood size equals to 0) severely limits the utility of the diffusion mechanisms and, for low values of the message rate, is even outperformed by the flooding. Differently, even a minimal cooperative behavior leads to a much more effective diffusion of data.

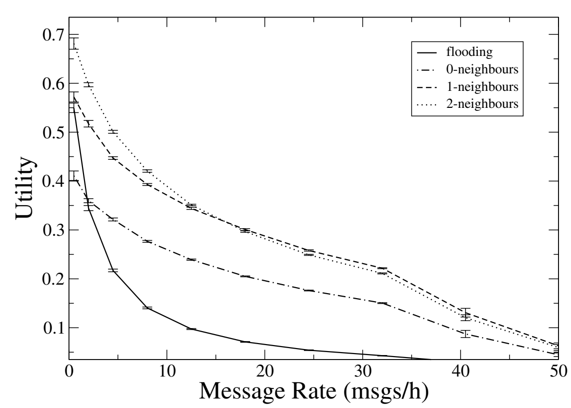


Figure 7: *Utility* in the case of flooding and k–nearest–neighbours–based diffusion. Contents are injected at a variable rate, and with an expiration time equals to $TTL = 48h$.

Finally, Fig. 8 presents the efficiency of the system for this second setting. For this second setting, the 1–nearest–neighbour diffusion mechanisms performs best. This is due to the optimal trade-off between the level of cooperation and the message redundancy that is introduced into the network. Indeed, especially for high values of message rate, the 1–nearest–neighbour interests-based diffusion scheme is able to achieve a very high utility (Fig. 7), but introducing a limited number of redundant messages (Fig. 6), if compared, e.g., with the 0–nearest–neighbour mechanism.

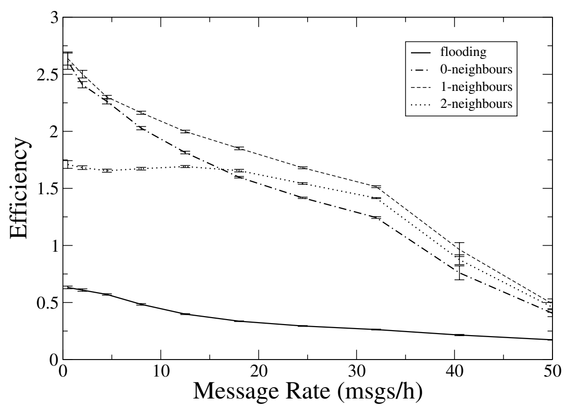


Figure 8: *Efficiency* in the case of flooding and k–nearest–neighbours–based diffusion. Contents are injected at a variable rate, and with an expiration time equals to $TTL = 48h$.

6. CONCLUSIONS

In this work, we have presented a class of opportunistic data diffusion mechanisms, tailored to a data–centric architecture, that exploit the concept of "neighbourhood" for offering a robust trade-off between user–perceived performance and usage of resources. The basic idea is to let nodes gather content of potential interests for their local neighbourhood (defined, roughly speaking, as the set of nodes with which they can best communicate), trading off resources (storage and communication) for performance (number of interested nodes reached). Algorithmic solutions have been proposed and implemented. Performance results were presented, exploiting the outcomes of an experimental measurement campaign carried out involving real users.

The work at hand sheds some light on the possible advantages that designers can take by leveraging the "social" aspects of opportunistic communication systems and we do believe that the proposed approach is general enough to be extended to broader application scenarios. Indeed, the two main assumption behind the proposed solution are (i) the presence of persistence neighbourhood, which holds for any social setting (ii) the possibility for users to express preferences on a limited set of categories.

Future work will be devoted to extend apply the proposed algorithmic solution to other real-world traces, where it possible to observe the social dimension, and to experiment the proposed scheme in a real–world setting, with real users and real contents.

7. ACKNOWLEDGMENTS

This work has been supported by the European Commission within the framework of the BIONETS project IST-FET-SAC-FP6-027748, `www.bionets.eu`.

8. REFERENCES

[1] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on the design of opportunistic forwarding algorithms. In *Proc. of INFOCOM*, Barcelona, Spain, April 23–29, 2006.

[2] P. Costa, C. Mascolo, M. Musolesi, and G. P. Picco. Socially-aware Routing for Publish-Subscribe in Delay-tolerant Mobile Ad Hoc Networks. *IEEE JSAC*, 26(5), June 2008.

[3] A. E. Fawal, J.-Y. L. Boudec, and K. Salamatian. Performance analysis of self limiting epidemic forwarding. Technical Report LCA-REPORT-2006-127, EPFL, 2006.

[4] K. Harras, K. Almeroth, and E. Belding-Royer. Delay tolerant mobile networks (DTMNs): Controlled flooding schemes in sparse mobile networks. In *Proc. of IFIP Netwoking*, Waterloo, Canada, May, 2005.

[5] P. Hui and J. Crowcroft. Bubble rap: Forwarding in small world dtns in ever decreasing circles. Technical Report UCAM-CL-TR-684, Univ. of Cambridge, Computer Laboratory, May 2007.

[6] A. Khelil, C. Becker, J. Tian, and K. Rothermel. An epidemic model for information diffusion in MANETs. In *Proc. of ACM MSWiM*, pages 54–60, Atlanta, Georgia, Sept. 28, 2002.

[7] V. Lenders, M. May, G. Karlsson, and C. Wacha. Wireless ad hoc podcasting. *ACM/Sigmobile Mobile Computing and Communications Review*, Feb. 2008.

[8] A. G. Miklas, K. K. Gollu, S. Saroiu, K. P. Gummadi, and E. de Lara. Exploiting social interactions in mobile systems. In *Proc. of UBICOMP*, Innsbruck, Austria, Sept 2007.

[9] M. Musolesi and C. Mascolo. Controlled Epidemic-style Dissemination Middleware for Mobile Ad Hoc Networks. In *Proc. of ACM Mobiquitous*, July 2006.

[10] M. Musolesi and C. Mascolo. Designing mobility models based on social network theory. *ACM SIGMOBILE Mobile Computing and Communication Review*, 11(3), July 2007.

[11] L. Pelusi, A. Passarella, and M. Conti. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *IEEE Comm. Mag.*, 44(11), Nov. 2006.

[12] Reality Mining. Machine perception and learning of complex social systems. http://reality.media.mit.edu/.

[13] J. Su, J. Scott, P. Hui, J. Crowcroft, E. de Lara, C. Diot, A. Goel, M. Lim, and E. Upton. Haggle: Seamless networking for mobile applications. In *Proc. of Ubicomp*, pages 391–408, Sept. 2007.

[14] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-2000-06, Duke University, 2000.

[15] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance modeling of epidemic routing. *Comput. Netw.*, 51(10):2867–2891, 2007.