Pervasive and Mobile Computing 4 (2008) 633–657

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

Fast track article

Exploiting users' social relations to forward data in opportunistic networks: The HiBOp solution☆

Chiara Boldrini, Marco Conti, Andrea Passarella *

IIT-CNR, Pisa, Italy

ARTICLE INFO

Article history:
Received 3 August 2007
Received in revised form 22 December 2007
Accepted 10 April 2008
Available online 22 April 2008

Keywords:
Opportunistic networks
Social networking
Routing
Pervasive networks

ABSTRACT

Opportunistic networks, in which nodes opportunistically exploit any pair-wise contact to identify next hops towards the destination, are one of the most interesting technologies to support the pervasive networking vision. Opportunistic networks allow content sharing between mobile users without requiring any pre-existing Internet infrastructure, and tolerate partitions, long disconnections, and topology instability in general. In this paper we propose a context-aware framework for routing and forwarding in opportunistic networks. The framework is general, and able to host various flavors of context-aware routing. In this work we also present a particular protocol, HiBOp, which, by exploiting the framework, learns and represents through context information, the users' behavior and their social relations, and uses this knowledge to drive the forwarding process. The comparison of HiBOp with reference to alternative solutions shows that a context-aware approach based on users' social relations turns out to be a very efficient solution for forwarding in opportunistic networks. We show performance improvements over the reference solutions both in terms of resource utilization and in terms of user perceived QoS.

1. Introduction

Improvements in the miniaturization technology enable the inclusion of an extremely rich set of features in mobile devices. Devices such as standard smart phones and PDAs come with at least a camera, fairly rich text editors, calendars, several wireless technologies (2.5/3 G, Bluetooth, IrDA, WiFi), etc. Users are thus equipped with powerful mobile devices that enable them to generate multimedia content *anytime and anywhere*. This trend goes hand in hand with content generation and sharing models that are becoming increasingly popular as the Internet penetration increases. In the today Internet, content is increasingly generated by the users themselves, and shared in virtual spaces such as Flickr, YouTube, etc. It is not visionary to say that these two trends would have big potential if joined together. The main limitation for this to become reality is an efficient wireless networking support for mobile users. Despite the research efforts in the pervasive networking field, the only mass-market wireless networking solution that actually made an impact on real users is the WLAN technology. Indeed, besides very interesting intellectual results, more than a decade of research on generic MANETs (the standard paradigm to provide wireless networking support in pervasive environments) did not provide significant developments outside the research community [8,9].

We believe that *opportunistic networks* [26], a pragmatic evolution of the generic MANET paradigm, may contribute to fill this gap, and naturally enable data sharing among communities of mobile users, without requiring, as in today's Internet, any pre-existing infrastructure, such as a WLAN or a cellular network. The types of networks that can be formed by mobile

☆ This work was partially funded by the European Commission under the HAGGLE (027918) and SOCIALNETS (217141) FET Projects.
* Corresponding author.
E-mail addresses: c.boldrini@iit.cnr.it (C. Boldrini), m.conti@iit.cnr.it (M. Conti), a.passarella@iit.cnr.it (A. Passarella).

1574-1192/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.pmcj.2008.04.003

devices are extremely dynamic in nature. Due to users' movements and wireless links instability, the topology is always changing, and partitions, possibly lasting for a long time, are extremely likely. MANET routing protocols see these features as exceptions to be managed, and aim at providing an Internet-like, stable view of the network, on which higher-level protocols (from the transport to the application) can be developed. Opportunistic networks do not look at those features as problems. Rather, partitions are seen as a natural state of the network, and node mobility is exploited to bridge disconnected "islands" of users. Typically, paths between communicating nodes are not completely built before starting the forwarding process (as in MANET protocols), but are computed dynamically, on-the-fly, while the messages are proceeding towards the destination(s). This allows opportunistic networks to naturally support end-to-end communications in challenged networks, such as pervasive networks made of users' devices only.

It is clear that, to accomplish this, legacy MANET routing protocols should be drastically redesigned [10,15,16,28]. Currently, envisioning routing and forwarding protocols [1] for opportunistic networks is one of the most compelling issues [26]. As discussed in detail in Section 3, we believe that the legacy-Internet routing approach based on topological information only, which has been inherited by generic MANETs, is not adequate for opportunistic networks. We also highlight the inefficiencies and questionable applicability of routing schemes designed explicitly for opportunistic networks, such as Epidemic Routing [34] and PROPHET [19]. In this paper we advocate *context-aware routing based on users' social behavior*as a far more efficient solution. In general, context information can complement partial and unstable topological information to provide efficient routing schemes in opportunistic networks. Context is usually quite a loose concept in computer engineering, and can be actually used to represent a number of users' properties (e.g., their physical location, the status of their devices, the status of the physical environment around them, etc). In this paper we mostly see the context as a collection of information that describes the reality in which the user lives, and the history of social relations among users. For example, the context can be defined by personal information about the users (e.g., name), about their residence (e.g., address), about their work (e.g., institution), about their hobbies (e.g., address of the sport facilities they go to). The routing protocol can exploit a user to forward messages destined for people sharing similar context properties (e.g., living in the same place, or in a place nearby).

The use of context is somewhat embedded in previous routing proposals in the literature. For example, PROPHET exploits the frequency of contacts between users. MobySpace [18] and MV [7] exploit information about users' mobility patterns and places users visit. This data can be seen as context information. The first contribution of this paper is to provide a far more comprehensive approach to context-aware routing through a *general framework* (described in Sections 4 and 5) that can easily host various flavors of context-aware routing. This framework allows nodes to automatically *learn* the context they are currently immersed in, and *remember* context information they became aware of in the past. Then, it allows nodes to exploit this knowledge base to identify next hops for carried messages. The second contribution of this paper is instantiating the framework, and proposing a context-aware protocol (*HiBOp, history-based routing protocol for opportunistic networks*) that exploits context to represent users' behavior and social relations. HiBOp nodes learn and remember information about users' personal information, behavior, and acquaintances (i.e., social relations), and exploit it to drive the forwarding process.

Finally, in Sections 7–9 we discuss an extensive set of results documenting the advantages of HiBOp over reference protocols (specifically, Epidemic Routing and PROPHET). To highlight the advantages of context-awareness alone, we consider ideal scenarios in which the wireless technology provides unlimited bandwidth, and there is no constraint on the memory available at nodes. Then, we also look at more realistic scenarios, and investigate the impact of bandwidth, memory limitations and variable network load on the protocols' performance. Finally we briefly discuss how to exploit HiBOp to support group communication, which is a very promising communication mode for opportunistic networking scenarios. Specifically, we show the HiBOp performance with anycast traffic. In all the cases we consider, exploiting context information reduces significantly the resource consumption in terms of memory and bandwidth (and thus, indirectly, energy too). This may come at the cost of a limited increase of the message delay and message loss rate, especially in scenarios without tight resource constraints. However, the extent of this increase does not hinder the deployment of sensible applications for opportunistic networks. Furthermore, we show that, besides being always more efficient in terms of resource usage, HiBOp is increasingly efficient in terms of message loss rate and delay as limitations on the available resources become tighter and tighter, up to becoming the best protocol even from this standpoint. Finally, we discuss and show that HiBOp is already able (without design modifications) to support group communication modes such as anycast. This is a big advantage over other protocols exploiting some form of context information (such as PROPHET) that are inherently designed for unicast applications. The comparison between Epidemic and HiBOp shows that HiBOp is far more efficient also with group communication applications.

Context-aware routing for opportunistic networks is a pretty recent field of research, but has attracted a lot of interest from researchers all around the world. Given the novelty of the topic, our work cannot address all possible aspects and some points still remain open issues. Among them, sensitiveness, scalability and privacy are main concerns. The problem of privacy is discussed in Section 6, where we try to give some ideas on possible solutions based on the current state of the art, while not pretending to be exhaustive. The problem of scalability is deeply analysed from the point of view of the load on the network in Section 8.4 and some considerations on scalability with respect to the management of context information are

[1] The distinction between routing and forwarding becomes quite fuzzy in opportunistic networks. Therefore, we use these terms interchangeably in the paper.

also given. A third aspect which would deserve more attention is the sensitiveness of the HiBOp protocol to configuration and environment parameters and to imprecision in context information. Some initial results have been presented in [3]. A more extensive sensitiveness analysis is one the main subjects for future work.

2. Related Work

Since routing is one of the most compelling issues in opportunistic networks, several research groups are working on this topic. For the sake of space, in this section we only mention Epidemic Routing [34] (throughout referred to as Epidemic), PROPHET [19], and CAR [23], which are representative of three fundamental approaches to routing in opportunistic networks. The reader can find a comprehensive survey on routing protocols for opportunistic networks in [26].

Epidemic is representative of the simplest type of routing protocols. Routing is based on pair-wise contacts between nodes, during which nodes exchange a *summary vector* containing the list of messages stored at each node. Based on received summary vectors, each node requests those messages it has not yet available in its buffer. Messages are delivered to the destination when the destination meets a node carrying the messages addressed to it. Epidemic is representative for routing protocols that essentially flood (in a controlled way) the network to route messages (other examples of flood control are provided by Spray & Wait [30,31] and Randomized Flooding [33]). HiBOp aims at drastically reducing the cost of such flooding by exploiting context information. From a different standpoint, one could note that one of the routes that Epidemic uses to deliver a message is optimal, in the sense that it is the quickest one to deliver the message. Identifying this route in advance clearly requires an oracle. HiBOp exploits context information to try to identify this particular route, thus approximating the ideal routing algorithm.

Probabilistic ROuting Protocol using History of Encounters and Transitivity (PROPHET) is an evolution of Epidemic that introduces the concept of delivery predictability. The delivery predictability is the probability for a node to encounter a certain destination. The PROPHET forwarding algorithm is similar to Epidemic except that, during a contact, messages are requested only if the receiving node has greater delivery predictability for the destination. PROPHET is representative for a class of routing protocols that exploit *some* context information to limit the Epidemic Routing flood (other examples are MV [7], MaxProp [6] MobySpace [18], Spray & Focus [32], Island Hopping [27], DTC [21], Last Encounter Routing [12]). HiBOp is able to manage and exploit far richer context information with respect to these solutions. Specifically, it provides a general solution for gathering and managing any type of context information. Furthermore, it shows how to use context information to model users' social behavior, and the performance advantages of this type of context information.

Context-Aware Routing (CAR) aims at fully exploiting context information, as HiBOp does. CAR assumes an underlying MANET routing protocol that connects together nodes in the same MANET cloud. To reach nodes outside the cloud, a sender looks for the node in its current cloud having the highest probability of delivering the message successfully to the destination. CAR provides a well-stated framework to compute this probability based on context information. HiBOp differs from CAR in a number of ways. Firstly, nodes in CAR compute delivery probabilities proactively, and disseminate them in their ad hoc cloud. Therefore, context is exploited to evaluate probabilities just for those destinations each node is aware of. HiBOp is more general, as it does not necessarily require an underlying MANET routing protocol, and is able to exploit context also for those destinations that nodes do not know. Furthermore, the definition and management of context information is not addressed in CAR, while it is a core part of HiBOp. Indeed, CAR is more focused on defining algorithms to combine context information (which is assumed available in some way) to compute delivery probabilities. Finally, and most importantly, CAR does not capture, in the context definition, any information about the users' social behavior, which we demonstrate being a particularly valuable piece of information to achieve an efficient routing scheme in opportunistic networks. Therefore, a direct comparison between HiBOp and CAR in our scenario is not particularly interesting, while it is more interesting comparing HiBOp with Epidemic Routing and PROPHET. Blending together features of HiBOp and CAR is an interesting subject of future work.

This work is an extended version of our previous paper [4]. In this work we describe more extensively the social networking concepts inherited by HiBOp, and how they are implemented in our proposal to achieve a context-aware forwarding framework based on users' social relations. Furthermore, we significantly extend the performance evaluation part, by analyzing the HiBOp behavior under additional resource limitations (with respect to those considered in [4]). Finally, while the work in [4] only considers unicast traffic, in this paper we provide performance results relative to group communication modes as well (specifically focusing on anycast), and we consider why these communication modes are particularly suitable for opportunistic networking applications.

HiBOp has been evaluated in [3] from a different standpoint with respect to the analysis presented in this work. Specifically, in [3] we have analyzed the sensitiveness of HiBOp (in comparison with Epidemic) to the parameters that define the users' social behavior, while in this paper the evaluation is carried out with respect to system parameters such as the available bandwidth and buffer. One of the most interesting findings in [3] is the fact that the more the nodes are social (i.e. the more they mix across the network), the more context based solutions are effective.

3. A context-aware approach to routing based on users' social relations

In opportunistic networks legacy MANET routing and forwarding is not adequate. This is essentially due to the fact that MANET routing protocols try to achieve a consistent and stable view of the path between a sender and a receiver before

starting the forwarding process. The problem with this approach is that in mobile pervasive networks the topology is – in general – so dynamic and rapidly changing, that a stable path might not exist at any time, or be too short lived for the routing protocol to find and exploit it. Therefore, routing based on topological information only is not adequate to the opportunistic networking paradigm. Routing (and forwarding) protocols should exploit a richer set of information than simple topological structures to identify next hops that are "probably good enough" to bring messages closer to the destination(s).

HiBOp is based on these remarks, and exploit context information related to users' behavior and social relations to identify on-the-fly "good enough" next hops. Indeed, HiBOp is inspired by the body of work on social networking and small-world theories (e.g., [17,36]), stemming from the seminal 1967 experiment by Stanley Milgram [20]. In the experiment, Milgram sent packages to a set of people, by only exploiting acquaintance between next hops. Specifically, the packages included a letter stating: "*If you do not know the target person on a personal basis, do not try to contact him directly. Instead, mail this folder to a personal acquaintance who is more likely than you to know the target person*". Milgrams' experiment ended with only 25% of missives reaching the final destinations (the willingness to collaborate of intermediate senders should be taken into account here) but the results of the analysis of the path followed by each letter (the famous theory of six degrees of separation) has originated a huge stream of research.

HiBOp is designed to exploit these ideas in opportunistic networking scenarios. Nodes running HiBOp infer acquaintance between users through context information about the users themselves (e.g., the users working place), and the users' behavior (e.g., how frequently the user meets people living in a particular street). The similarity between the contexts of any two users is exploited as a measure of their acquaintance. Messages are forwarded through users having increasing acquaintance with the destination. In general, for each message more than one copy is injected into the network, each copy following a different route. Some copies may get lost, just as in the Milgram experiment, but some will arrive at the destination with high probability.

To implement these concepts, HiBOp includes two main building blocks. On the one hand, HiBOp nodes run algorithms to *build, update and manage context information* gathered during the node's lifetime. This allows nodes to build a knowledge base on which users' acquaintances can be estimated. On the other hand, HiBOp includes algorithms to *compute estimates* of the acquaintance between users. This allows any two nodes that get in touch to select which is the most suitable node out of them to bring messages closer to the destination(s). The following two sections describe in detail the ideas and algorithms implementing these two building blocks.

4. Context creation and management

The main idea behind HiBOp context management algorithms is to mimic the way in which people get acquainted with each other. This can be represented through two complementary processes. The first process consists in people introducing and exchanging information about themselves, upon meeting and spending time together. The second one consists in remembering information about other people met in the past, re-enforcing information about people frequently met, and aggregating similar information shared by different persons met in the past. The first process is the basis to create new acquaintances, and concerns managing information available at the *present time*. The second one is the basis for ranking acquaintances, and remembering, in a structured way, information about people met in the past. Thus, it concerns the *history*and the *legacy* of information gathered over time.

HiBOp inherits these concepts, and defines the *present context* and the *historical context* of each node. We can anticipate that both the present and the historical contexts are required to realize the HiBOp forwarding scheme. Intuitively, the present context is a snapshot of the local environment the user is immersed in. Based on this snapshot, a node could be seen as a good forwarder because, for example, one of its neighbours lives in the same street of the destination. More in general, HiBOp exploits the current context to evaluate the *instantaneous* fitness of a node to be a forwarder. Taking forwarding decisions based only on instantaneous information would be very limiting, as the present context does not represent users' behaviours and past experiences. For example, a user can be deemed a good forwarder if every morning she passes by the destination's house on her way to work. The idea of exploiting the repetitiveness in the user behaviour comes from the analysis of real mobility traces ([15] among others), which highlights the fact that human beings tend to be people of habit. Based on this consideration, it is reasonable to assume that events that frequently happened in the past are likely to repeat in the near future. The role of the historical context is precisely to emphasize this kind of events (more details on this point are provided in Section 4.1).

Before providing all the details about the HiBOp context management algorithms, let us briefly introduce how context information is represented in HiBOp. The present context of a user is made up of two main components. The first component is information about the user itself, while the second component is information about the current physical neighbors of the node. Information about the user is stored in the Identity Table (IT), an example of which is shown in Table 1. Identity Tables consist of a set of couples $\langle attribute, value \rangle$. The name of attributes is defined by HiBOp, while the values of the attributes are filled in by the users. As will be explained in Section 4.1, the Identity Table is an extensible structure, which allows users to control the stored information (i.e., HiBOp mechanisms work with any choice of information stored in the Table). At each node, information about the current physical neighbors is represented as the set of neighbors' Identity Tables. To gather them, any two nodes getting in touch with each other exchange their Identity Tables (with the optimizations described in Section 4.1).

Table 1
Identity table example

Personal information	
Name	Donald
Surname	Duck
Email	d.duck@iit.cnr.it
Phone	340- 343439847837
NID	PLNPPRXX04XX4Y
Residence	
Street	Feather street, 13
City	Pisa
Work	
Street	Moruzzi street, 1
City	Pisa
Organization	CNR
Hobbies & fun	
Address	Sport street, 10
City	Pisa
Association	SportDuck
System information	
MAC-Bluetooth	01:23:45:67:89:AB
MAC-802.11	09:00:07:A9:B2:EB
IP-Address	168.0.3.14

Table 2
History table structure

Attribute	Class	P_C	H	R
Pisa	City	...	...	...

The historical context is represented through the History table, whose structure is shown in Table 2. At a high level, the History table records attributes seen during the past in the Identity Table of encountered nodes. The example row reported in Table 2 tells that the node has seen the attribute "Pisa" (of class "City"). As explained in detail in Section 4.1, the rest of the information stored in each row of the History table allows HiBOp to estimate the probability of encountering that attribute in the near future. It is worth noting that, thanks to this attribute-centric representation (instead of a node-centric one), HiBOp remembers far more than the mere identity of encountered nodes (as, for example, PROPHET does). All attributes of encountered users let some legacy in the HiBOp history. This is actually a big advantage, because it allows HiBOp to exploit similarities between encountered users and the destination. For example, a node can be deemed a good forwarder because it is very likely to encounter some (unspecified) other user that lives in the same street of the destination.

4.1. Context-management algorithms

Let us firstly focus on Identity Tables. In general, ITs can contain an extensible set of data, including personal information, such as name and surname, behavioral information, such as job place and hobbies, system information, such as network addresses of node's network interfaces, etc. In general, is up to the user to decide what to expose in the node's IT. HiBOp works with any kind of information stored in Identity Tables (i.e., there is no limitation on what can be stored in ITs). The only requirement is that, for any two nodes, the set of information (possibly) stored in one node's ITs be a subset of the information (possibly) stored in the other node's IT. This set is defined by the names of the possible attributes of the IT (left-hand side column of Table 1). Users can only set the value of an attribute, not the attribute itself (e.g. Pisa, but not City). The available set of attributes is provided to the user through the corresponding version of the HiBOp protocol. Subsequent versions of HiBOp must ensure backward compatibility with respect to the set of attributes. We assume that ITs uniquely identify nodes in the network. In particular, the Node IDentity (NID) field is a hash of the IT, and is used to uniquely name a node in the network. Clearly, privacy and security issues are main concerns. A preliminary discussion is presented in Section 6.

Nodes learn the environment around them by exchanging ITs during Neighbor Discovery phases, which nodes perform periodically and asynchronously from each other. The *Current Context* (CC) of a node is defined by the ITs of its current neighbors. Specifically, the time interval between two Neighbor Discovery phases is called Signaling Interval. At the end of every Signaling Interval, each node should send either its IT or its NID. If during the last Signaling Interval it received only ITs or NIDs of nodes that are in its Current Context already, then it simply refreshes its presence by broadcasting its NID. Otherwise, if it received ITs or NIDs for nodes that are not in its Current Context, it broadcasts its complete IT. In this way,

Table 3
Repository table structure

Aggr	Class	Carriers	Cont count	Het count	Red count
Pisa	City	A,B,C	2	1	2

complete ITs are exchanged only among nodes that came in contact during the last Signaling Interval, while stable contacts among neighbors (i.e., contacts lasting for more Signaling Intervals) are refreshed by NIDs. An IT is removed from the CC table when the related node is not in contact anymore. In order to tolerate transitory disconnections or transmission errors, an IT is removed from the CC table after a given number of consecutive Signaling Intervals (after a Death Interval) during which neither ITs nor NIDs are received for that node.

The second building block of HiBOp context representation is the History (Table 2). Here we propose a solution based on a single History table. Recall that the aim of the History table is to emphasize the preferential behavior of users. However the user behavior strongly depends on the current "role" of the user. For example, a user can be a "worker" from Monday to Friday (8:00 am – 8:00 pm) and a "family member" the rest of the time. The people he preferentially meets and the places he preferentially visits change based on his role. The more the historical context information is representative of the current role of the node, the more HiBOp is accurate. A way of achieving this is to use a different History table for each "role" the user owning the device plays in his everyday life and select the History table based on the current role of the node. However, this is mainly an optimization and, for sake of simplicity, in the rest of the paper we will concentrate on a single-role/single-History case.

The History table stores the values the node has seen in ITs of neighbors met in the past. For example, if a node receives an IT with a row ⟨City, Pisa⟩, then there will be a row in the History table whose Attribute field is "Pisa". The Class field is the corresponding *name* of the attribute in the Identity Table ("City" in the example). The reason why we store classes will be clear later on. Three counters are bound to each attribute, i.e., the Continuity Probability (P_c), the Heterogeneity (H), and the Redundancy (R). P_c represents the probability of encountering a node that carries that value in its IT. The H field contains the average number of distinct encountered nodes, which stored that attribute. This field is a sort of fault tolerance index, because high heterogeneity means that there are several distinct chances of encountering that attribute on distinct nodes. The R field contains the average number of occurrences of the attribute *within the same* IT. The redundancy information is valuable, because if a node stores the same attribute several times in its IT, then its link towards that attribute is very high.

The History table is built as the legacy of the evolution of the Current Context. To dynamically update its content, an intermediate data structure is used, called Repository table (whose structure is shown in Table 3). This table has an entry for each attribute the node has recently seen. The evolution of the Repository table can be characterized through two time intervals: the Signaling Interval marks the update time of the Repository, while every Flushing Interval the content of the Repository is merged into the History table. At the end of every Signaling Interval, HiBOp scans its Current Context, and adds a new row in the Repository table for attributes in the Current Context that have not yet a corresponding row in the Repository table. All the other fields for such new rows are set to 0. Both for new and old rows, the values of the related counters are then updated. In more detail, for each attribute with a corresponding row in the Repository field, HiBOp executes the following steps:

- the Continuity Counter is incremented. Therefore, the Continuity Counter stores how many times that attribute has been seen in the Current Context during a Flushing Interval;
- if the node whose IT stores that attribute is not listed in the Carriers list, the Heterogeneity counter is incremented, and the NID of the node is added to the Carriers list. In addition, the Redundancy counter is incremented by the number of times that attribute appears in the IT. Therefore, the Heterogeneity counter stores on how many different neighbors that attribute has been seen (during the current Flushing Interval), while the Redundancy counter stores the total number of entries in the Current Context that contain that particular attribute (during the current Flushing Interval).

Once every Repository Flushing Interval (which is an integer number of Signaling Intervals), HiBOp uses the data in the Repository table to update the History table. For each value of attribute in the Repository table we compute the corresponding Continuity Probability, Heterogeneity, and Redundancy as explained below. Next, we combine these results with the corresponding values in the row associated with that attribute in the History Table as shown in Eqs. (1)–(3). Specifically, a sample of Continuity Probability is computed as

$$p_c^{(\mathrm{rep})} = \frac{Cont\ Count}{M},$$

where M is the number of Signaling Intervals in a Repository Flushing Interval. The sample of the Continuity Probability is thus computed as the probability of having seen that attribute during the previous Flushing Interval (recall that *ContCount* is the number of Signaling Intervals during which that attribute has been seen in the Current Context). The Continuity Probability in the History table is then updated as follows:

$$P_c \leftarrow \delta \cdot P_c + (1-\delta)\, p_c^{(\mathrm{rep})}, \tag{1}$$

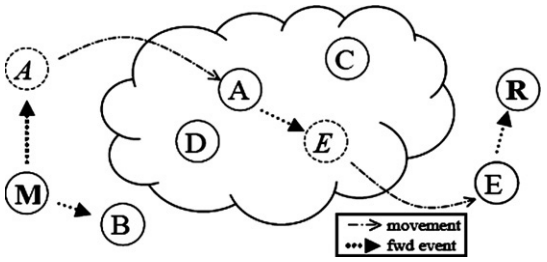


Fig. 1. HiBOp forwarding process.

where δ is a standard smoothed average parameter ($0 \leq \delta \leq 1$). In a similar fashion, the Heterogeneity and the Redundancy are updated as follows:

$$H \leftarrow \delta \cdot H + (1 - \delta) \cdot \mathit{Het\ Count} \tag{2}$$

$$R \leftarrow \delta \cdot R + (1 - \delta) \cdot \frac{\mathit{Red\ Count}}{\mathit{Het\ Count}}. \tag{3}$$

The computation of the Heterogeneity is self-explanatory from Eq. (2). As far as the Redundancy Eq. (3), we compute a redundancy sample as $\frac{\mathit{Red\ Count}}{\mathit{Het\ Count}}$, and then apply again a standard smoothed average. Dividing *RedCount* by *HetCount* means computing the "average redundancy" of the attribute during the Flushing Interval, i.e, the average number of times that attribute has been seen in a *single* Identity Table during the Flushing Interval.

5. Using context information for forwarding operations

Also in the forwarding process HiBOp mimics how people would forward packages by using acquaintance information (as in the Milgram experiment), and forwards messages via nodes with increasing probability of bringing them closer to the destination. This policy is not new in the literature about forwarding in opportunistic networks. The novelty of HiBOp is to *exploit users' social behavior (modeled with context information)* to evaluate these probabilities. One of the salient features is the fact that HiBOp does not evaluate delivery probabilities based only on the history of direct contacts between nodes (as, for example PROPHET does). Instead, HiBOp evaluates probabilities based on the similarity with the destination's context information. This allows HiBOp to exploit "hidden" shortcuts enabled by social relations and users' behavior. For example, HiBOp can discover (and select as forwarders) node that happen to go very often through the street where the destination lives, even though that node and the destination never met neither directly, nor through a sequence of intermediate contacts.

To implement these concepts, a HiBOp sender embeds into a message more information about the destination than a simple network address. The sender should include any subset of the destination's Identity Table it is aware of (ideally, the whole Identity Table). Note that including this possibly large information set in messages' header is not a big overhead in opportunistic networks. Indeed, the typical size of messages (also known as bundles) is significantly higher than the size of IP packets (e.g., bundles can easily contain a whole file) [25]. Delivery probabilities are evaluated based on the *match* between information about the destination, stored in the message header, and the context information stored at each encountered node. A high match means high similarity between the node's and the destination's context. Actually, delivery probabilities can be seen as a measure of this similarity.

Besides this, it should be noted how HiBOp controls message replication, which is a major advantage over state-of-the-art solutions. Specifically, only the sender of a message is allowed to create multiple copies of the message. Other nodes that carry a message compute the delivery probabilities of encountered nodes, and do *not* keep copies of forwarded messages. This allows HiBOp to control and drastically reduce message flooding.

The HiBOp forwarding process can be thus decomposed in three phases (see Fig. 1):

- *Emission*: the sender injects the message in the network, replicating it for the sake of reliability.
- *Forwarding*: exploiting nodes' mobility and contacts, each copy of the message proceeds in the network towards the destination.
- *Delivery*: when a node carrying the message finds the destination the process stops.

The third phase of the process is trivial and is not discussed further. The rest of this section is thus devoted to the Emission (Section 5.1) and Forwarding (Section 5.2) phases.

5.1. Emission phase

In opportunistic networks it is clearly impractical to manage reliability via ARQ mechanisms like in the legacy Internet (or in MANET too). Techniques such as message replication or network coding look more suitable. HiBOp addresses reliability by replicating messages at the sender only. HiBOp assumes that the application notifies a reliability requirement in terms

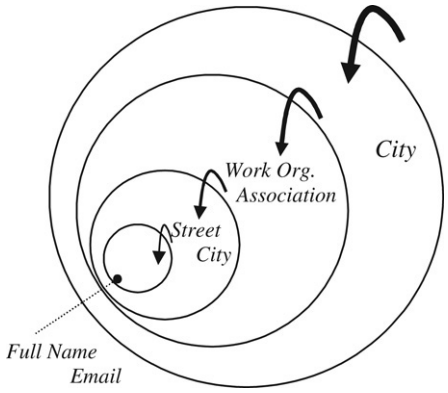


Fig. 2. Precision of attribute classes.

of maximum tolerable message loss, $p_l^{\max}$. Following the mechanisms described in Section 5.2, a sender node gets from its neighbors the probabilities of successfully delivering the message to the final destination. Let us denote them as $p_{\text{succ}}^{(i)}$, where i denotes the i-th neighbor, ordered by decreasing delivery probability, and let us denote the delivery probability of the sender as $p_{\text{succ}}^{(0)}$. Assuming that these probabilities are independent, the number of neighbors (k) to which the message is forwarded by the original sender is evaluated as follows:

$$k = \min\left\{ j | \prod_{i=0}^{j} (1 - p_{\text{succ}}^{(i)}) \leq p_l^{\max} \right\}.$$

Basically, the sender forwards the message to the minimum number of neighbors such that the joint loss probability is below the maximum threshold specified by the application. If not enough neighbors are currently available, the message is forwarded to the available neighbors, is queued at the sender, and new neighbors are used as soon as they become available. Note that, to avoid flooding in case of too low delivery probabilities, neighbors are used as forwarders just if their delivery probability is above a threshold (set to 0.001 in our experiments). The sensitiveness of HiBOp performance to the $p_l^{\max}$ parameter is investigated in [4].

5.2. Forwarding phase

5.2.1. Weighting attributes

The match between contexts should be *weighted* based on the class of the matching attribute. Class weights should represent the *precision* of that class in identifying the destination. For example, a match on the destination's name gives far *more precise* information than a match on the residence city of the destination.

Fig. 2 visually represents the qualitative ranking of class precision we have defined for HiBOp (bigger circles represent lower precision). The definition of the weights we have used reflects this qualitative ranking.

Several functions can be used to assign weights to classes based on the ranking above. In our case, named w_0 the weight of the least significant class, we have computed other weights as $w_{i+1} = w_i + r_i \cdot \beta$, where β is defined as the weight increase parameter, and r_i is the maximum redundancy of the i-th class. The main idea is that (i) weights should be monotonically increasing, and (ii) the relative difference between classes should increase if the less significant one allows for a higher redundancy, because higher redundancy usually means lower significance.

5.2.2. Forwarding based on delivery probability

A node wishing to forward a message broadcasts the destination's information along with its own delivery probability. Nodes that receive such broadcast evaluate their delivery probability and send it back to the inquiring node (with a unicast transmission) if it is higher than the inquiring node's one.

As already said, the delivery probability is a measure of the similarity between the context of the destination and the context of the forwarder candidate. The more these two contexts are similar, the more the forwarder is likely to make the message approach the destination. The similarity of two contexts, i.e. the delivery probability, is given by the number of matches between their attributes. At each node, the delivery probability is computed from three components, related to (i) the node's Identity Table, (ii) the node's Current Context, and (iii) the node's History. In the following we describe how HiBOp currently exploits context information to compute these values. Investigating alternative policies is an interesting subject of future work. Note that HiBOp is actually a framework for context-aware routing in opportunistic networks. The concepts described so far still hold true for any particular implementation of the mechanisms required to realize these concepts.

As far as the Identity Table, the node finds those attributes in the destination information that matches with attributes in its IT. The delivery probability from the IT is then evaluated as the ratio between the sum of the weights of matching attributes, and the sum of the weights of attributes specified in the destination information, i.e.:

$$P_{IT} = \frac{\sum_{j \in \{\text{match}\}} w_j}{\sum_{j \in \{\text{dst\_info}\}} w_j}. \tag{4}$$

As far as the Current Context, recall that it is made up of ITs of current neighbors. For each such IT the node evaluates P_{IT}, and the delivery probability related to the Current Context is the maximum over these probabilities (i.e. we consider the best service a node in the CC can provide):

$$P_{CC} = \max_{j \in CC} P_{IT}^{(j)}. \tag{5}$$

Evaluating the contribution of the History to the delivery probability requires more steps. In fact, in this case we have not only to consider the presence of a given value of attribute, but also the strength with which this attribute has been seen in the past (and therefore is likely to be encountered again in near future). Recall that each attribute in the History table comes with three indices, i.e., the Continuity Probability (P_c), the Heterogeneity (H), and the Redundancy (R). First of all, HiBOp selects the attributes that match with destination information. For such attributes, the R and P_c indices are combined as follows:

$$P_{op}^{(j)} = P_c^{(j)} \cdot \frac{R^{(j)}}{r},$$

where r is the maximum possible redundancy for the class of the j-th matching attribute. Essentially, P_c is scaled according to the potential redundancy that the attribute could achieve. The rational here is that the likelihood of meeting again this attribute in the near future increases when the attribute is repeated more than once in the ITs of encountered nodes. For example, if a user lives and works in Pisa she is a better forwarder for a destination with "Pisa" among its attributes than a user who only lives in Pisa. Similarly to the contribution related to the IT, the contribution related to the History is evaluated based on the weighted average of the $P_{op}^{(j)}$ values, computed as:

$$P'_H = \frac{\sum_{j \in \{\text{match}\}} P_{op}^{(j)} \cdot w_j}{\sum_{j \in \{\text{dst\_info}\}} w_j}.$$

The delivery probability related to the History is evaluated by modifying P'_H according to the H indices of matching attributes. This H index is a measure of the reliability of the P_c value. In fact, P_c reveals only the presence of an attribute over time but is not able to distinguish between the case of an attribute advertised by a single node and an attribute advertised by more than one node. Therefore, P_c being equal, a forwarder with a higher Heterogeneity value is to be preferred, because it offers a better fault tolerance. Including this information allows HiBOp to be much more robust with respect to problems caused by single nodes behaving differently from their consolidated habits. Specifically, HiBOp increases P'_H of a factor $\Delta_{\max}$ at most, scaling this factor according to the average heterogeneity of matching attributes ($\bar{h}$):

$$P_H = \min\left\{1, P'_H + \Delta_{\max} \cdot \left[1 - e^{-(\bar{h}-1)}\right]\right\}. \tag{6}$$

Note that, since $\Delta_{\max}$ is scaled according to an exponential law, the same average heterogeneity increase results in a higher P_H increase for small values of average heterogeneity (e.g., increasing $\bar{h}$ from 1 to 2 has a greater effect than increasing $\bar{h}$ from 10 to 11).

The delivery probability is finally computed by combining Eqs. (4)–(6), as follows:

$$P = \alpha \cdot P_H + (1 - \alpha) \cdot \max\{\eta \cdot P_{CC}, P_{IT}\}. \tag{7}$$

Eq. (7) is made up of two components, weighted with a factor α ($0 \leq \alpha \leq 1$). The first component is P_H, which describes the *legacy* of the node's past history. The second component describes the *current* status of the node's environment. The α factor is used to give more weight to past history or to the current environment. The node's current environment is jointly described by P_{IT} and P_{CC}, which are therefore combined together in Eq. (7). The η factor ($\eta < 1$) scales down P_{CC} with respect to P_{IT}, because P_{CC} is related to a neighbor, while P_{IT} is related to the local node. Let assume two potential next hops A and B, and let assume that the delivery probability of node A (based on its Current Context) is equal to the delivery probability of B, based on its Identity Table (i.e., $P_{CC}^{(A)} = P_{IT}^{(B)}$ holds true). Let us also assume that the other components of the delivery probabilities are 0 (i.e., $P_H^{(A)} = P_H^{(B)} = P_{IT}^{(A)} = P_{CC}^{(B)} = 0$ holds true) Based on Eq. (7), $P^{(A)} = \eta \cdot P_{CC}^{(A)}$ and $P^{(B)} = P_{IT}^{(B)} > P^{(A)}$ hold true. Therefore, the η parameter makes node B preferable as a next hop. This is correct, because forwarding through A surely requires a further hop to give the message to the A's neighbors that generated $P_{CC}^{(A)}$.

6. Privacy and security considerations

Security issues in general, and privacy issues in particular, are clearly main concerns for HiBOp (and for any context-aware routing) scheme, since context might possibly include sensible information about the user. Opportunistic networking is a relatively new area, and privacy and security aspects are still among the less explored research topics. Due to its importance to the HiBOp framework, in this section we mainly focus on privacy issues. Other topics related to the security field in opportunistic networking include cooperation enforcement, encryption, and robustness against DoS attacks to routing operations. A few results related to these topics are available in the literature. For example, the work in [24] proposes, in the framework of the Haggle project [13], a cooperation enforcement scheme tailored to opportunistic networks based on reward mechanisms. This scheme copes with all security problems of typical reward-based mechanisms, including protection against poisoning attacks, cheating actions, and unfairness. The work in [5] analyzes the effect of a wide range of security attacks to routing operations in opportunistic networks built upon dropping packets, flooding, falsifying routing tables, and counterfeiting message acknowledgements. The main finding is that *multi*-path routing is very robust *by design*, i.e., even without any form of authentication. The main reason is because all of these attacks result in nodes' unavailability or path disruptions, which are already considered as characteristic features of the opportunistic network by multi-path routing protocols. Since HiBOp is a multi-path routing protocol, it is expected to be robust against these attacks too. Finally, the Haggle project is also looking at encryption and vulnerability problems of routing protocols similar to Epidemic Routing, and routing schemes exploiting network coding. More details about this body of work can be found in [13].

Before describing how the HiBOp specification can be complemented with privacy support, we would like to highlight why the system evaluated in Section 7 does not include any privacy mechanisms. From a general design standpoint, we do acknowledge that privacy support is fundamental for HiBOp. However, as clearly explained in Section 7, in our experiments we use a set of context attributes that are usually not perceived as very private by users. Specifically, we consider context information, such as the working place and the working address, that people usually publicize in their public Web sites. Furthermore, recall that in HiBOp any user can decide which context information should be exposed in their Identity Table. Therefore, it is not unrealistic to assume that users will be willing to run HiBOp in the configuration considered in our experiments, even without any privacy support. Thus, not including privacy mechanisms does not impact on the validity of the results presented hereafter.

6.1. Coping with privacy, confidentiality, integrity

To the best of our knowledge, no fully-fledged solutions exist to ensure privacy, confidentiality and integrity in opportunistic networks. However, proposals addressing the problem of *key management* could be exploited to tackle these issues. Indeed, as noted in a recent survey [35] on key management for mobile ad hoc networks, since these security aspects are usually addressed via some form of encrypted communication, key management is one of the key blocks to enforce security aspects in mobile ad hoc networks. In the rest of this section, after recalling the key management proposals for opportunistic networks, we specifically focus on privacy, since it is the most compelling issue related to context-aware routing. We discuss possible directions to design privacy solutions on top of available key management solutions.

6.1.1. Key management in opportunistic networks

With respect to general mobile networks, [35] identifies two main approaches to key management, i.e., *authority based* and *fully self-organized*. Authority-based solutions rely on trusted authorities in charge of distributing and managing keys. These authorities can be online or offline, and can be distributed in the network so as to avoid the problem of a having a single point of failure. Ordinary nodes have to receive keys from the authorities before joining the network. Fully self-organizing solutions do not require any trusted authority. Nodes generate keys and authenticate with each other through fully distributed mechanisms. Examples of both authority-based and self-organized solutions tailored to legacy MANETs are proposed, e.g., in [14].

While research on key management for MANETs has received a lot of attention (see, e.g., the extensive reference list in [35]), just a few papers elaborate on how to tailor this body of work to opportunistic and delay-tolerant networks. To the best of our knowledge, only two authority-based solutions [1,29] have been proposed so far. Both are based on Identity-based Cryptography (IBC) [2,11], because IBC provides features particularly suitable for disconnected environments. Specifically, solutions based on IBC just require a trusted Private Key Generator (PKG) for the distribution of *private* keys, while the public key of a node can be computed by knowing the node's identifier only. Ordinary nodes only have to contact PKGs once to obtain their own certified private key. The work in [1] shows that, thanks to these features, IBC solutions (i) reduce the number and frequency of interactions with authorities with respect to standard solutions based on trusted key servers; (ii) allow communications between ordinary nodes and the trusted authority (to retrieve keys) to be asynchronous with respect to communications between ordinary nodes (to exchange data). Therefore, IBC solutions are particularly suitable for disconnected networking environments.

6.1.2. Directions to support privacy in context-aware routing

A first approach to enforce privacy in context-aware forwarding can exploit the concept of *community*, which naturally lends itself to opportunistic networking environments. One of the most interesting applications of opportunistic networks is indeed to support participatory content generation and sharing within communities of users (e.g., according to the concepts of Web 2.0 and User-Generated Content), through an infrastructure-less network made of user devices only, which does not require full connectivity as in the MANET paradigm. In general, a user can be member of different communities, and the same (physical) opportunistic network can support multiple communities at the same time.

If an authority-based key management system is in place, the following simple scheme guarantees privacy support to HiBOp (or to any other context-aware protocol). Specifically, it guarantees that context information about any node is exposed to members of the node's same community only. Note that allowing even unknown users of a known community to know selected information is usually not perceived as a privacy threat (see, e.g., social networking sites).

The scheme works as follows. The trusted authority of the key management scheme is also in charge of registering users to communities, and enforcing public policies to allow users to be part of the community. It also defines keys (either symmetric or asymmetric) reserved for communications between members of the same community. Upon registration, users can decide the set of context information they wish to expose to the other community members, and receive the keys for communicating in the community. Then, the HiBOp protocol at each node exchanges context information only with other members of the node's community, and thus stores context information related to the node's community only. In the forwarding phase, just nodes of the destination community are selected as candidate forwarders.[2]

Clearly, the drawback of this scheme is to have just a limited number of nodes available as candidate forwarders. To overcome this limitation, it is possible to slightly modify HiBOp operations to include in the forwarding process also users that are not member of the destination's community without violating the aforementioned privacy guarantee. Let us firstly focus on the context-management mechanisms described in Section 4. In this regard, the privacy requirement is to ensure that information contained in each node's Identity Tables is exposed only to members of the node's community. Without loss of generality, let's assume that members of the same community exchange Identity Tables by encrypting attributes and values with a key known only to members of the community.[3] Upon meeting a non-member user (say, NM), a member (say M) sends its encrypted Identity Table without violation of its privacy. To store context information and participate in the forwarding process, NM can use a different History Table for each "encountered" community. Since the encryption key used by M is the same across all members of M's community, each couple ⟨attribute, value⟩ shared by any two members of M's community will result in the same ciphered text. Therefore, it is straightforward to see that also NM is able to compute the statistics related to M's community required by the context-management mechanisms (specifically, the indices of the History Table in Table 3). Notably, NM can compute these statistics based on ciphered text only. Therefore, context information related to M's community will not be exposed outside the community itself.

Similar remarks hold true also with respect to the HiBOp forwarding mechanisms (Section 5). In this case, the security requirement is not to disclose context information related to a member destination node (M) to non-member nodes (NM) used as intermediate forwarders. To be considered as a candidate forwarder, NM should evaluate the match between its context and the context of M. Recall that the match is computed with respect to three components, i.e., (i) NM's Identity Table; (ii) the Identity Tables of NM's current neighbors, and iii) the NM's History Table (related to M's community). Clearly, since M's context information will be encrypted, NM will not be able to compute component (i). Furthermore, it will be able to compute component (ii) only with respect to its current neighbors that are members of M's community. Finally, it will be able to compute component (iii) by using the History Table related to M's community. As a complementary remark, note that using NM as a forwarder will not reduce the information about M available to other members of M's community possibly encountered by NM at a later time. NM will carry the encrypted context information about M, which can then be decrypted by other members of M's community without loss of information.

It is easy to see that HiBOp modifications to support privacy along a community-based approach may lead to underestimate delivery probabilities computed without any privacy requirement. Therefore, these modifications allow HiBOp to guarantee privacy with some reduction in forwarding performance. While we do not quantify this reduction in the following, it should be noted that – in general – best forwarders are expected to be members of the destination's community, because members of the same community are more likely to meet. Therefore, we can expect just minor performance reductions when such privacy schemes are adopted.

Such privacy schemes might, in general, result in a higher networking overhead with respect to the case when no privacy support is included. Specifically, when users that are members of different communities should broadcast their encrypted Identity Table once for each community they belong too. The great reduction of traffic overhead of HiBOp with respect to Epidemic and PROPHET (see results in Section 7) suggests that this overhead can be actually accommodated without hindering HiBOp effectiveness. Furthermore, an optimized solution can be designed if communities are hierarchical.

[2] This scheme implies that senders should register to the destination's community to be able to communicate, which is a reasonable requirement.

[3] Note that, as described in Sections 4 and 5, HiBOp needs only to match attributes and values of the destination with attributes and values stored in the candidate forwarder's internal structures. Matches can be computed either on clear text or on ciphered text, without affecting HiBOp operations. Therefore, nodes will not be forced to decrypt context information neither to update context-management structures, nor to perform forwarding operations. Such a solution is therefore viable also from a computational standpoint.

Specifically, let us assume that N communities are defined ($C_1, \ldots, C_N$). Let us also assume that to each community C_i corresponds a set of information in the Identity Tables (IT_i) such that $IT_j \supseteq IT_i$ for each $j > i$. In other words, community C_1 defines a basic set of information (IT_1) that, without loss of generality, could be thought as shared without any privacy constraints across the whole network. Community C_2 defines a slightly more restricted set of information, that include IT_1, and that can be shared across a more restricted set of users, and so on. In this case, if techniques such as Cipher Block Chaining are used to encrypt/decrypt the Identity Tables, encrypted Identity Tables can be broadcasted only once as in the HiBOp operations without privacy support. The only additional requirement is that, if two users share a community C_j, they must also share all communities C_i for $i < j$.

In the framework of the Haggle project, alternative approaches to ensure privacy with respect to the community-based solutions are being considered. A preliminary scheme (described in [13]) requires only that nodes share a common hash function. Nodes exchange and store *hashed* context attributes and values instead of clear text. Matches can be computed on the hashed values without impacting on the protocol's effectiveness. While this is a simple and straightforward solution, it does not guarantee perfect privacy. First, even non-malicious users can become aware of private context information about other users, if they happen to have that particular information in their Identity Table. Furthermore, since all nodes share the same hash function, malicious users can build brute-force attacks to infer context information of other users. Although the possible number of attribute values can be fairly large (e.g. the cities in the world), it is anyway quite limited. Building a brute-force attack is not considered as prohibitive. Based also on this remark, such a simple solution is seen just as an interim solution within the Haggle project. The ultimate goal will be to provide a privacy mechanism robust to brute-force attacks, that, at the same time, allows nodes to compute *complete* matches by exploiting *all* available context information (as described in Sections 4 and 5).

7. Simulation methodology and setup

7.1. Evaluation plan

In order to evaluate the performance of HiBOp we use a custom simulator we have developed. The simulator implements HiBOp, Epidemic, and PROPHET, and allows us to compare their performance in terms of delay, buffer occupation, message loss and amount of traffic generated in the network (the detailed definitions of the performance figures are provided at the end of this section).

We start with a configuration with unlimited resources, both in terms of bandwidth and in terms of buffer available at each node (results are presented in Section 8.1). In this configuration we assume ideal wireless links with infinite bandwidth and negligible transmission delay. This is clearly unrealistic, but allows us to isolate the effect of context awareness from networking effects such as congestion, transmission errors, etc., and from the effect of message drops induced by buffer overflows. Therefore, these results highlight the impact of using context information only.

Since HiBOp exploits context to reduce messages' spread in the network, we can anticipate that neglecting bandwidth and buffer limitations favors Epidemic and PROPHET. As far as bandwidth limitations, we neglect additional delay and message loss related to wireless technologies characteristics (e.g., transmission errors, congestion, etc), which significantly increase delays and message drop rates under high traffic load [32]. All protocols are expected to perform worse as bandwidth limitations are considered. However, since Epidemic and PROPHET generate more traffic than HiBOp (as is clearly shown by our results), the performance worsening of Epidemic and PROPHET is expected to be much more severe than that of HiBOp. Similarly, by not considering buffer limitations, we grant more resources to Epidemic and PROPHET than to HiBOp (thus favoring their performance), as Epidemic and PROPHET spread messages more aggressively than HiBOp does. We can thus expect that Epidemic and PROPHET performance will suffer much more than HiBOp by limitations on the buffer size.

To complement this set of results, we then quantitatively investigate the effect of both bandwidth and buffer limitations on the protocols' performance. Limiting the bandwidth available at each node is clearly an approximation of the real effect of wireless links' features. However, it is a simple (yet significant) way of investigating the effects of wireless technologies limitations (that anyway result in bandwidth limitations) without sticking to any particular technology.

Finally, in the last set of experiments we extend the analysis to group communication modes, and discuss why they could be widely adopted in pervasive mobile networking environments.

7.2. Simulation setup

Our simulation scenario is a square of size 1250 × 1250 m, divided in a 5 × 5 grid. The number of nodes is set to 40, and the transmission range to 70 m. Nodes move according to the traces generated by the Community based Mobility Model [22]. This model is quite different from traditional random models and mimics real human movement patterns. Every node belongs to a social community. Nodes that are in the same social community are called friends, while nodes in different communities are non-friends. Links between nodes represent social relations among nodes, and links' weights represent the strength of the social relations. Each node has links towards its friends. In general, it may also have links towards non-friends, according to the rewiring probability parameter. Links towards non friends represent relations across different communities. Nodes' movements are determined by the attraction towards cells. Each cell exerts an attraction on

Table 4
CMM parameters

Number of nodes	40
Simulation area	1250 × 1250 m
Cells in the grid	5 × 5
Node speed	$U \in [2–9]$ m/s
Number of groups	8
Reconfiguration interval	9000 s
Travelers speed	5 m/s
Number of travelers	8

Table 5
HiBOp parameters

Signaling interval	5 s
$p_i^{\max}$	0.05
Repository flushing interval	1800 s
δ	0.5
η	0.95
$\Delta_{\max}$	1
α	0.5
Death interval	10 s
Default buffer size	50 messages
Default number of senders	20
Default message size	50,000 B

any given node, measured as the sum of the weights of links between the node, and other nodes roaming in the cell. Each node moves (with a uniformly distributed speed) towards the cell to which it is most attracted. Therefore, it is more likely that a node will be in contact with nodes of its community, because they spend more time together. CMM also includes the notion of travelers that do not always move in the cell where they have more friends. From time to time, they move to the second most attractive cell (i.e., to the cell in which they have the second highest number of friends), and then get back to the most attractive cell afterwards. Once in a while a reconfiguration occurs, during which all groups change cell. During a reconfiguration nodes of different groups have chances to meet. To reduce the number of mobility parameters on which the performance of routing protocols depends, we use the following CMM configuration (please refer to Table 4 for the complete list of the relevant parameters). Nodes are divided in 8 groups (each made up of 5 nodes). Nodes have only social relations with their friends, i.e., inside their community. We consider one traveler per group, and a reconfiguration interval 9000 s long (i.e., all nodes of all group move at once towards the same cell once every 9000 s). Unless during reconfigurations, nodes are stuck to move within their social community, and only travelers are used to enable message exchanges between different communities. In this configuration nodes do not mix very much. This has proven not to be a favorable condition for HiBOp, because HiBOp becomes more and more efficient as the nodes mix together, and context information is able to circulate easily in the network [3]. Note that CMM in general, and our CMM configuration in particular, allows us to model typical mobile users scenarios, such as a users moving in a University campus in which communities are made of people attending the same classes, or a working environment in which communities are working departments. Note that we have replicated experiments with different sets of parameters (e.g., with a higher number of nodes). The behavior of the protocols, and the comments we provide in the following, still hold true in these cases.

The set of default HiBOp parameters is show in Table 5. In our simulations, every 5 s, each node sends its NID/IT and discovers new neighbours. This value is a compromise between the responsiveness of the discovering algorithm to changes in the neighbourhood and the need of not overburdening the nodes. Data collected during the signalling intervals are inserted in the Repository, whose entries, in turn, are merged with the History table every 1800 s with a weight equal to 0.5 (i.e. Repository entries and History entries have the same importance). From the forwarding point of view, each message is replicated by its source to obtain a 95% reliability. When computing delivery probabilities, the probability extracted from the History table is considered as important as that of the current environment ($\alpha = 0.5$), while the delivery probability of CC is scaled with a factor 0.95 with respect to the local delivery probability, to take into account the additional hop required to reach a node of the CC.

The context used by HiBOp is the name of the user, and the information about its working place (name, city, address). As mentioned in Section 6, people typically do not perceive this information as particularly sensitive with respect to privacy. To make the context coherent with CMM, similar attributes are given to nodes belonging to the same group. Simulation results show that HiBOp is already able to outperform Epidemic and PROPHET also with such a limited context information.

We consider a messaging application, with twenty senders uniformly distributed among groups. The set of senders is chosen uniformly at random at the beginning of the experiment. The interval between the generation of two consecutive messages at the same sender is modeled according to an exponential distribution, with average 300 s (unless for results in Section 8.4, where the average value is scaled). In unicast experiments, message destination is a friend node with 50% probability, and a non-friend node with 50% probability. Among the friends and non-friends, the destination is chosen uniformly at random. In the anycast experiments the destination is always a non friend. The group of the destination node is

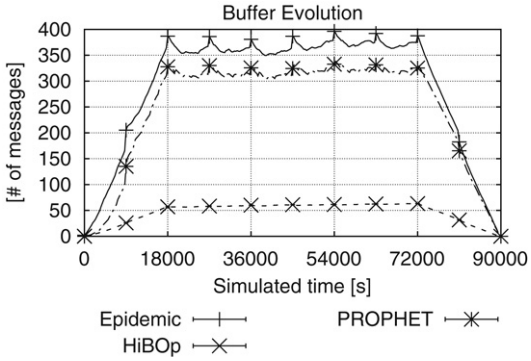


Fig. 3. Average buffer occupation over time.

Table 6
Overall average buffer occupation [5]

Epidemic	PROPHET	HiBOp
366.56 ± 4.9	318.07 ± 5.0	71.54 ± 0.97

chosen uniformly at random. Finally, messages expire after 5 h (18,000 s), which is reasonable for delay-tolerant applications. The size of messages is set to 50 KB.[4] This choice is consistent with related works on this topic [25].

Each simulation run lasts for 90,000 s. We consider the system to be in steady state when the following conditions are both true: the History table has been filled at least once and the average number of messages in buffers does not change over time. In our setup, the second condition is more restricitve and leads us to eliminate the first 18,000 s and last 36,000 s when computing the performance indices.

7.3. Performance indices

We consider two sets of performance figures. The first one accounts for resource utilization, while the second one for user perceived QoS. As far as resource utilization indices we consider the evolution of *average buffer occupation*, and the *bandwidth overhead*. We define the evolution of *average buffer occupation* as the average buffer occupation (in number of messages) over time. Specifically, we periodically sample the buffer occupation of each node, and compute the average occupation at each sample instant over all nodes (we also show that this is a good estimator for the average buffer occupation of nodes in the stationary regime). The *bandwidth overhead* is defined as the ratio between the total number of bytes exchanged over the network during a simulation run, and the total number of bytes generated by senders. The bandwidth overhead thus represents the average cost of injecting a new byte into the network. Note that the total number of bytes exchanged also includes the protocols' overhead. In the case of HiBOp, it thus accounts for the mechanisms required to manage context information and identify suitable next hops. This is a cost paid by HiBOp (with respect to Epidemic and PROPHET), to reduce messages' spread. The traffic utilization index allows us to quantify the payoff of this design choice.

To quantify the users' perceived QoS we consider the *message loss rate* and the *average delay*. Though intuitive, the precise definition of these indices depends on the communication modes. In the unicast experiments, the message loss rate is the ratio between the number of lost messages, and the number of messages generated by senders. In the anycast experiments, the message loss rate is the ratio between the number of messages lost by *all* members of the intended anycast group, and the number of messages generated by senders (i.e., to mark a message as lost, *no member* of the intended anycast group must receive it). The definitions of the *average delay* follow a similar line of reasoning. First of all, a delay equal to the messages timeout value is considered for lost messages. In the unicast case, the average delay is the sample mean of delays measured during a simulation run. In the anycast experiments, the average delay is the sample mean of the delays experienced by the *first node* in the intended group that receives each message.

Each simulation run provides a sample of the performance figures defined above. Unless otherwise stated, we hereafter present confidence intervals (with 90% confidence levels) and average values of the performance figures, achieved by replicating each simulation configuration 20 times with independent seeds.

[4] In delay-tolerant networks the size of messages is generally unlimited and much larger than typical IP payload. In fact, self-contained messages (e.g a whole file) tend to reduce end-to-end interactions between nodes, which is one of the goals of these networks [25].

[5] For sake of readability, these values are shown separately from buffer occupation evolution.

Table 7
Bandwidth overhead

Epidemic	PROPHET	HiBOp
35.59 ± 0.49	33.89 ± 0.40	11.90 ± 0.28

Table 8
User perceived QoS

	Epidemic	PROPHET	HiBOp
Loss rate (%)	0	0	1.25 ± 0.35
Delay (10^3 s)	0.94 ± 0.065	1.10 ± 0.069	2.43 ± 0.15

8. Routing performance under varying resource constraints

In this section we discuss the performance of the routing protocols under investigation as a function of the available resources in terms of bandwidth and buffer space. We first focus on the case with unlimited resources (Section 8.1) and then consider buffer and bandwidth limitations (Sections 8.2 and 8.3, respectively). The communication mode for this set of experiments is unicast.

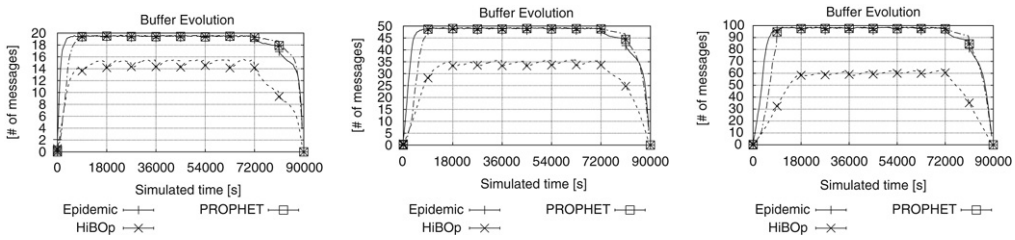
8.1. Results under unlimited resources

This is clearly the best possible configuration for Epidemic and PROPHET. Let us focus on the evolution of buffer occupation over time averaged over all nodes (plotted in Fig. 3), and on the average buffer occupation at any node (Table 6). Since we are not considering any maximum buffer size, this index highlights the maximum buffer demand of the various protocols. We stress the fact that Epidemic and PROPHET require about one order of magnitude more space than HiBOp. Note that the average buffer occupation shown in Table 6 is very close to the buffer occupation averaged over all nodes during the steady state (plotted in Fig. 3). Therefore, in the following we use the latter index as an estimator of the former one. Even though memory is cheap nowadays, messages in opportunistic networks are usually far larger than messages in IP networks, and whole files can be accommodated in a single message [10,25]. For example, in our configuration, for an average message size of 1 MB, nodes running Epidemic should reserve about 400 MB just for routing purposes!

In case of HiBOp, in addition to the space occupied by messages we must take into account the space required for storing context information. At any given time, the memory required by context information is the sum of the IT's size, the size of current neighbors' ITs, and the size of the Repository and History tables. Based on the definitions of these data structures provided in Section 4, and by assuming (i) to represent attribute values with 16 bytes, and (ii) to use standard integer and double data types for the numerical values as appropriate, the average buffer occupation of context information in our simulations is 6340.67 ± 102.99 bytes. This value is the same for all scenarios considered in this paper because it depends only on the underlying mobility settings, which are the same in all our configurations. What clearly emerges from this evaluation is that the space required to store context information is approximately 10% of the space required to store a single message. Therefore, the overhead of context-awareness on buffer occupation is minor with respect to the overall message load.

HiBOp reduces resource consumption with respect to Epidemic and PROPHET also in terms of bandwidth overhead, as shown by Table 7. The traffic generated by HiBOp is about one half of the traffic generated by Epidemic and PROPHET. Recall that the total number of bytes generated includes not only the application-level messages to be forwarded, but also the whole routing and forwarding traffic generated by the protocols. Therefore, this index also accounts for the effect of exchanging Identity Tables and using long message headers in HiBOp, which is an additional overhead with respect to Epidemic and PROPHET.

The high reduction in terms of resource consumption achieved by HiBOp is not paid with a significant reduction in terms of user QoS (i.e., average delay and message loss), as shown in Table 8. HiBOp achieves the highest message loss rate and delay, as expected in an unlimited-resource configuration. However, the HiBOp loss rate is just 1.25%, and the HiBOp average delay is $2.7\times$ and $2.3\times$ the average delay experienced by Epidemic and PROPHET, respectively. Even though there is a clear delay increase, HiBOp overall performance in terms of user QoS remains acceptable even in this extremely favorable scenario for Epidemic and PROPHET. The applications for which opportunistic networks are sensible are delay tolerant in nature. For those applications resource consumption is much more critical than delay, because lower resource consumption allows the user to use mobile devices more efficiently (e.g., longer, with lower costs, etc). Therefore, the amount of additional delay brought by using HiBOp does not hinder the adoption of such a routing protocol, as the higher delays shown in Table 8 are more than compensated (from a user standpoint) by improved resource efficiency.



(a) Buffer = 20. (b) Buffer = 50. (c) Buffer = 100.

Fig. 4. Average buffer occupation.

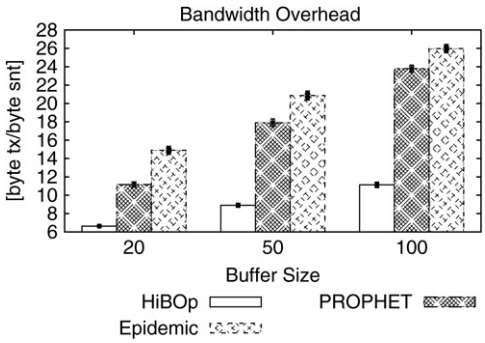


Fig. 5. Bandwidth overhead.

8.2. Results under buffer limitations

We consider three buffer sizes, (20, 50, and 100 messages), representative of low, medium, and high buffer size respectively. We still consider unlimited bandwidth available at each node. We consider joint limitations on bandwidth and buffer size in Section 8.3.

Fig. 4 shows the evolution of average buffer occupation over time for the three protocols. Note that, for ease of reading, the legend is only about HiBOp curves. Epidemic and PROPHET curves all overlap at the maximum buffer size. The plots in Fig. 4 highlight that, as expected, both Epidemic and PROPHET saturate the buffers. Specifically, after an initial startup phase, and before the final cool-down phase (the last 18,000 s in which no new message is generated), buffers are almost always 100% full. Since the figure plots the average buffer occupation over *all* nodes, this means that *all* buffers in the network are saturated. HiBOp is much less greedy in using buffer resources. The fact that the average buffer occupation is much less than the maximum buffer size, means that the probability of HiBOp saturating buffers is very low. As it is shown in the next sections, the number of messages delivered by HiBOp is even higher than the number of messages delivered by Epidemic and PROPHET. Therefore, the comparison based on the average buffer occupation is even a bit unfair to HiBOp.

Fig. 5 shows the resource consumption in terms of bandwidth overhead. HiBOp significantly reduces the traffic overhead with respect to Epidemic and PROPHET. Specifically, the reduction is in the range [41%, 48%] with respect to Epidemic, and [32%, 37%] with respect to PROPHET. Note that the traffic overhead clearly increases with the maximum buffer size, because – on average – each node stores an increasing number of messages to be forwarded when the buffer size increases. Therefore, the traffic generated either to forward these messages, or to exchange information about delivery probabilities, increases too.

Figs. 4 and 5 confirm that HiBOp significantly reduces resource consumption in terms of buffer occupation and traffic overhead and does not saturate the buffer space. From a complementary standpoint, this also shows that HiBOp's context-aware features act as an effective *congestion control* system for opportunistic network. This is a key point, as currently adopted routing protocols tend to be very greedy in resource usage, thus resulting in high resource congestion.

Fig. 6 shows the message loss of the three protocols for varying buffer sizes. As expected, the message loss drops as the buffer size increases, because messages can live longer in nodes' buffers. It is interesting to note that HiBOp message loss is lower (unless for large buffer sizes) than Epidemic and PROPHET message loss. This is not trivial, because HiBOp drastically reduces message replication (as shown by the lower buffer occupancy), i.e., it explores fewer paths towards the destinations. Potentially, this could results in *greater* message loss, if the wrong paths are chosen. Since the HiBOp message loss is lower or equivalent to the Epidemic and PROPHET message loss, this tells that (i) HiBOp allows messages to live longer in the buffers thanks to low replication, thus granting more time for them to be delivered, and (ii) HiBOp is almost always able to choose correct paths based on its context-aware rules. It is interesting to look at Figs. 5 and 6 at once. As the maximum

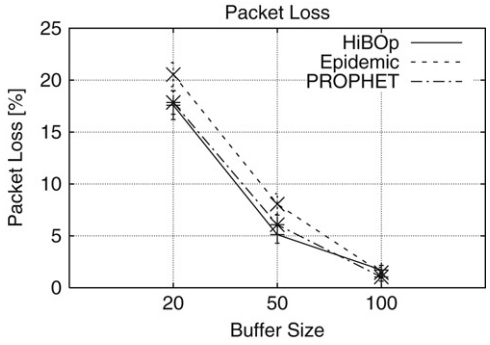


Fig. 6. Message loss rate.

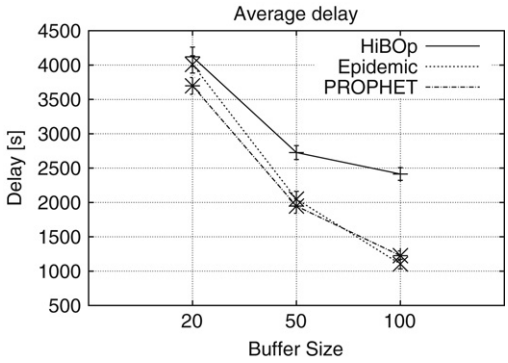


Fig. 7. Average message delay.

buffer size decreases, messages are dropped earlier and live shorter in nodes' buffers. Therefore, the lower the buffer size, the higher the message loss rate (Fig. 6). On the other hand, since messages live shorter in nodes' buffers, they have fewer chances of being exchanged over the network. Or, seen from a complementary standpoint, the amount of traffic that can be exchanged between nodes decreases with the buffer size. Therefore, the lower the buffer size, the lower the bandwidth overhead (Fig. 5).

The performance in terms of message delay is analyzed in Fig. 7. Recall that, to make delay distributions related to different protocols comparable, we added samples equal to the maximum message lifetime (18,000 s) for lost messages. First of all, it should be noted that delay values shown in Fig. 7 (in the order of tens of minutes) are typical of opportunistic networks, even though they might look fairly high. In more detail, delay is the only figure for which HiBOp performs generally worse than Epidemic and PROPHET. However, HiBOp additional delay is significantly reduced as the buffer size decreases, thanks to the lower message loss rate it achieves. These delay figures are acceptable for HiBOp. The average delay increase is tolerable, especially by recalling that protocols like Epidemic are not very likely to be widely adopted, due to their excessive overhead.

8.3. Results under bandwidth limitations

In this section we consider the effect of bandwidth limitations on the protocols performance. Firstly we focus on an unlimited buffer scenario, then we analyze the joint effect of both buffer and bandwidth limitations. As far as the bandwidth limits, we consider representative values for the IEEE 802.11 b technology. Specifically, the highest limit we consider is 5 Mbps, representative of a non-congested real network. We then scale the limit down to 1Mbps, 500 kbps and 250 kbps. For easy of comparison, in the following plots we also show the values achieved with unlimited bandwidth.

Figs. 8 and 9 show the performance in terms of resource consumption when buffers are unlimited. Note that Fig. 8 plots the HiBOp and Epidemic curves only, as PROPHET and Epidemic curves basically overlap, making the plots much less readable. Overall, HiBOp confirms to be much more efficient than Epidemic and PROPHET also when bandwidth limitations are considered. The general effect of bandwidth limitations is to reduce both the buffer occupation and the bandwidth overhead. However, the behavior of the three protocols is quite different. Let us focus on Fig. 9 (similar remarks holds also as far as buffer occupation). If a protocol saturates the available bandwidth, tighter limits on the bandwidth result in lower bandwidth overhead. Indeed, as the available bandwidth decreases, the amount of traffic that can be exchanged during contacts decreases as well (because of saturation), and therefore the bandwidth overhead (defined as the ratio between the number of bytes exchanged over the network and the number of bytes generated by senders) decreases too. In our

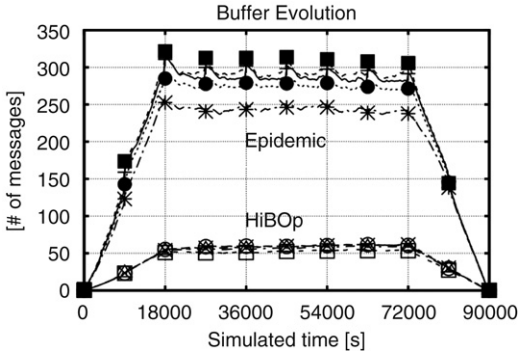


Fig. 8. Buffer occupation (infinite buffers).

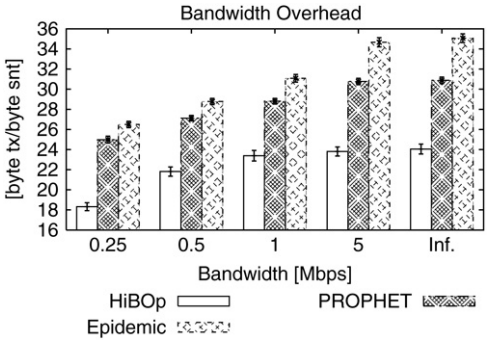


Fig. 9. Bandwidth overhead (infinite buffers).

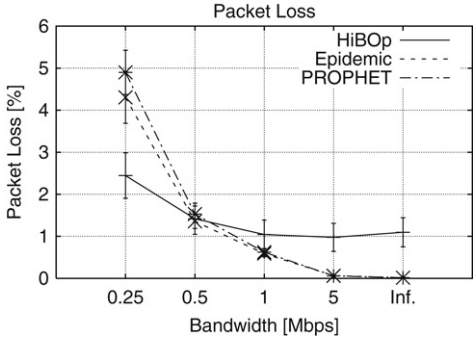


Fig. 10. Message loss rate (infinite buffers).

experiments it is thus expected that, after some value of the bandwidth limit, the bandwidth overhead of all protocols drops. The limit for which the overhead starts dropping is close to the bandwidth demand of the protocol. Fig. 9 clearly shows that this critical limit is highest for Epidemic, lower for PROPHET, and lowest (far lower than Epidemic and PROPHET) for HiBOp. This tells that Epidemic and PROPHET not only easily (with respect to HiBOp) saturate resources in terms of buffers (as shown in the previous sections), but also in terms of bandwidth.

Figs. 10 and 11 show the performance in terms of user QoS. These results are very interesting, and confirm our initial intuition about the increased advantage of HiBOp when real wireless networks constraints are considered. As a side effect of their higher resource usage, Epidemic and PROPHET experience an almost exponentially increasing loss rate as the bandwidth shrinks, which also results in an almost exponential increase of the average delay. On the contrary, HiBOp starts experiencing higher loss rate (and significantly increased delay) just for very tight limits of the bandwidth. This means that not only HiBOp is far more efficient than Epidemic and PROPHET in terms of resource consumption, but, as the network becomes more resource constrained, HiBOp becomes more efficient also in terms of user perceived QoS.

The final set of plots we consider show the joint effect of bandwidth and buffer limitations. For the sake of space, results presented hereafter consider the intermediate case of a buffer size equal to 50 messages. Figs. 12 and 13 show the performance in terms of resource consumption. As far as the average buffer occupation, note that all curves related to the same protocol overlap, and Epidemic and PROPHET curves overlap too. Specifically, note that both Epidemic and PROPHET

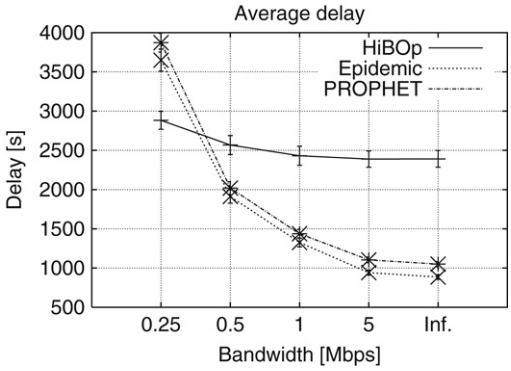


Fig. 11. Average message delay (infinite buffers).

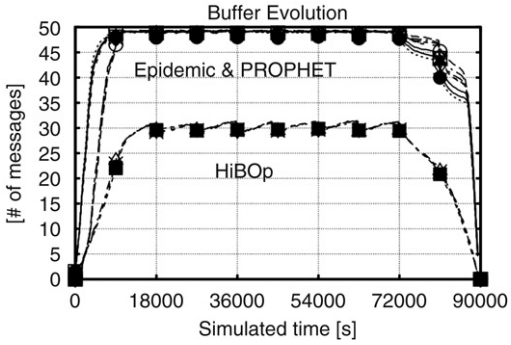


Fig. 12. Buffer occupation (limited buffers).

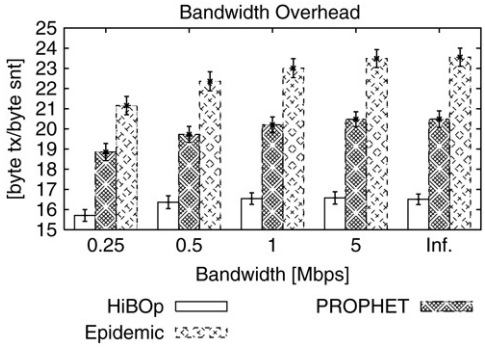


Fig. 13. Bandwidth overhead (limited buffers).

constantly saturate the nodes' buffers, as curves are stuck at the maximum buffer size. By comparing Figs. 10 and 12, it is clear that in the latter case the limitation on the buffer size dominates the limitation on the available bandwidth, as the Epidemic and PROPHET curves are independent of the bandwidth limit. HiBOp confirms to be much less aggressive, and never saturates the buffers. The fact that HiBOp curves overlap (at a value below the maximum buffer size) means that the bandwidth required by HiBOp to work with the experienced buffer occupation is always lower than the limits we have considered. As far as the bandwidth overhead (Fig. 13) and user QoS (Figs. 14 and 15), the protocols behavior is qualitatively similar to that described for the unlimited buffer case.

8.4. Scalability in terms of network load

In this section we provide an analysis of the effect of a varying load on the performance of routing protocols, which is one aspect of the more complex problem of scalability in opportunistic networks. We consider 20 senders, each transmitting with an average interarrival time between packets of 150, 300 and 600 s.

In this first set of results we will consider the case of unlimited bandwidth and buffers. Results show the same behaviour highlighted in Section 8.1: the excellent QoS performance of Epidemic and Prophet is paid with a huge resource consumption,

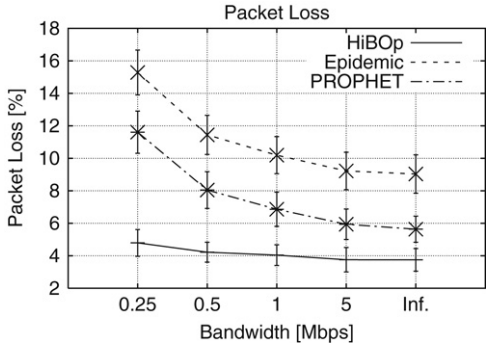


Fig. 14. Message loss rate (limited buffers).

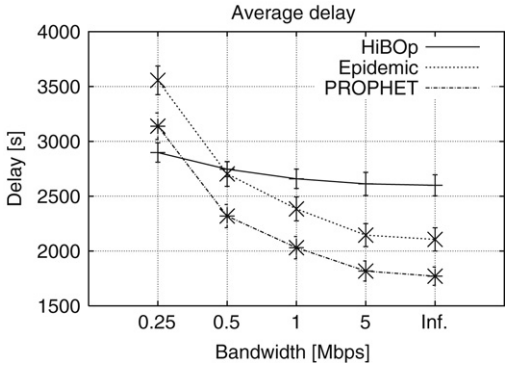


Fig. 15. Average delay (limited buffers).

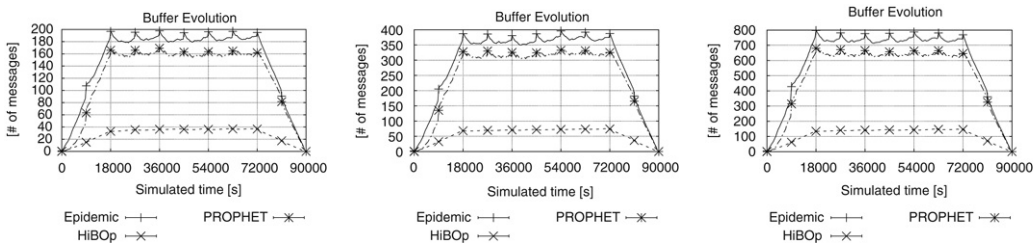
Table 9
User perceived QoS (values doesn't change with the load)

	Epidemic	PROPHET	HiBOp
Loss rate (%)	0,01	0.01	1.25
Delay (10^3 s)	0.94 ± 0.031	1.1 ± 0.034	2.4 ± 0.07

both from the memory and the bandwidth point of view. In fact, the buffer occupation (Fig. 16) of Epidemic is about four times the buffer occupation of HiBOp, while values for Prophet are three times those of HiBOp. The bandwidth overhead (Fig. 17) remains constant with the different loads, Recall that the bandwidth overhead is the ratio between the traffic produced in the network and the traffic generated by senders. Therefore, with unlimited buffers, increasing the network load does not impact on the overhead. Anyway, from the resource consumption point of view, HiBOp is much less greedy than the other protocols with any load. The performance in term of QoS (Table 9) does not change with the load, because, with infinite buffers, message forwarding depends only on the underlying mobility patterns: messages are never dropped for buffer overflow, they can only be deleted if they are out of time (destination not reached in time). Being able of exploiting infinite resources, Epidemic and Prophet have a negligible packet loss, while HiBOp shows some losses (∼1.2%) due to a low probability of selecting wrong next hops. Clearly, these results are strongly dependent on the infinite availability of resources. So, let us now consider a more realistic case, with a limited maximum buffer size set to 50 messages (the intermediate value of the analysis in Section 8.2).

When we limit the resources available to protocols, the performance of flooding based protocols significantly worsen. From the QoS standpoint, we see (Fig. 20) that the delays for the three protocols become closer and closer to each other as the load increases, with the inversion between HiBOp and Epidemic at a very high load (interarrival time equal to 150 s). This inversion is determined by the increased packet loss for Epidemic and Prophet at a high load (Fig. 21). In fact, dissemination-based protocols tend to flood the network with messages, but, with limited buffer space, this results in a fast saturation of the buffer space available on nodes (Fig. 18). Buffers being full, lots of messages are dropped and the QoS performance degrades quickly. The traffic per byte injected in the network (Fig. 19) decreases with the load, because a higher load implies higher packet loss and therefore fewer messages surviving in the network with respect to the amount of messages generated.

From this analysis we can conclude that HiBOp is, in all configurations, far more efficient than other protocols in managing resources and the network saturation is delayed significantly. However, there will be a cut-off in the size of the buffer with respect to the load beyond which all protocols will work in saturation and the performance will degrade permanently.



(a) interarrival = 600 s. (b) interarrival = 300 s. (c) interarrival = 150 s.

Fig. 16. Buffer occupation over time.

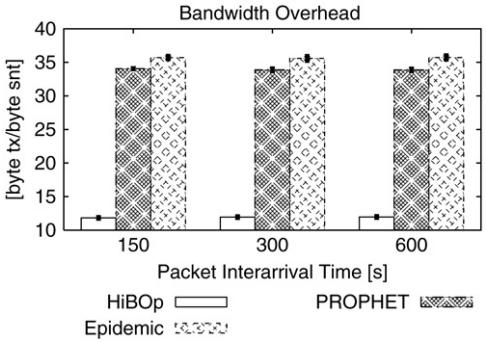


Fig. 17. Bandwidth overhead.

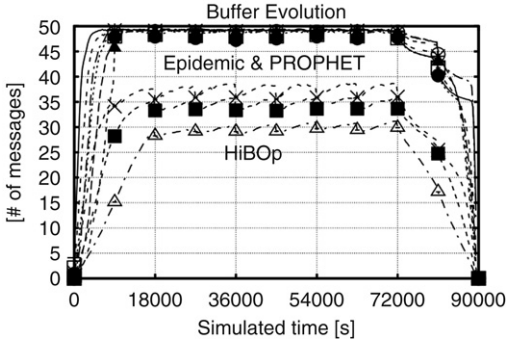


Fig. 18. Buffer occupation over time.

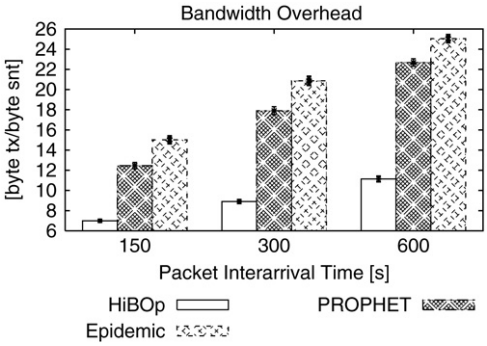


Fig. 19. Bandwidth overhead.

In this section we have evaluated the scalability of Epidemic, Prophet and HiBOp in terms of the load generated by the senders. Scalability should also be evaluated in terms of number of nodes. We do not provide this analysis here, as an extensive scalability and sensitiveness analyses are not targets of this paper. In terms of scalability with the number

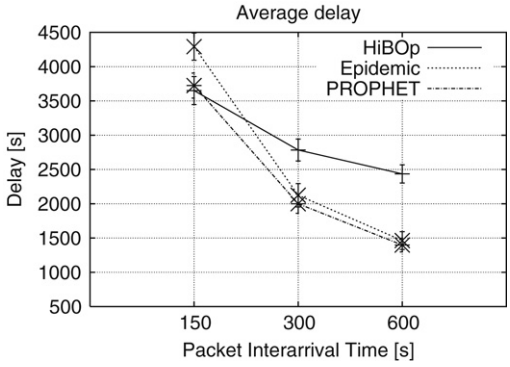


Fig. 20. Average message delay.

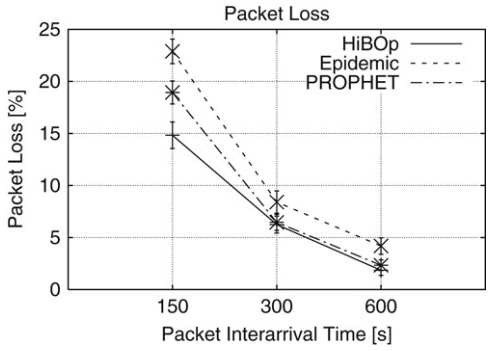


Fig. 21. Message loss rate.

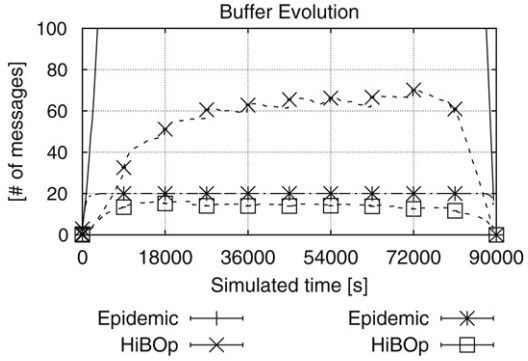


Fig. 22. Buffer occupation (anycast).

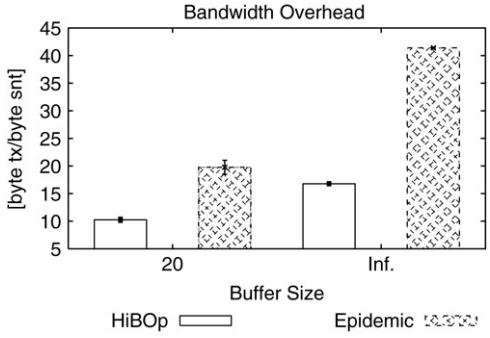


Fig. 23. Bandwidth overhead (anycast).

Table 10
Message loss (anycast)

	Epidemic	HiBOp
Buff = 20	46.34 ± 3.21	26.88 ± 3.76
Buff = inf	0.08 ± 0.16	4.00 ± 0.94

Table 11
Average delay (anycast, 10^3 s)

	Epidemic	HiBOp
Buff = 20	4.82 ± 0.34	3.96 ± 0.37
Buff = inf	1.36 ± 0.12	2.85 ± 0.16

of nodes, we can expect that, increasing the size of the network, the performance of dissemination based protocols will worsen, because these protocols tend to use all available resources and to saturate the network. In addition, increasing the number of nodes in the network will also increase the average path length. This will result in messages going through an increasing number of hops and therefore in more forwarding messages (needed for getting the delivery probabilities in Prophet and HiBOp). However, for HiBOp, because each message is replicated just a small number of times, we expect that the forwarding overhead will increase less than in Epidemic and in PROPHET. Finally, varying the number of nodes will not necessarily impact on the amount of memory required to store context information. This indeed depends on the density of the nodes, rather than on the number of nodes in the network (and, furthermore, we have shown that the amount of memory space required by context information is very small).

9. Supporting group communication in HiBOp

The analysis carried out in the previous sections assumes unicast traffic only. However, in opportunistic networks other types of communication modes (such as multicast and anycast) could actually be adopted much more widely than in the legacy Internet. As mentioned in the Introduction, opportunistic networks are suitable to support sharing of data generated by users sharing inside users communities. In such a scenario, data sources might naturally address data to *groups* of users (instead of single users) interested in the generated data. Similarly, users in opportunistic networks might ignore which other user is providing a required service or is storing a required data. These are example of group communication services, which could be key components of future mobile pervasive networks. In this scenario, multicasting and anycasting are clearly fundamental communication modes.

From a design standpoint HiBOp already includes all the required features to support context-aware multicasting and anycasting. Indeed, senders can simply identify groups (both in multicast and in anycast modes) through sets of information that identify the destination group (i.e., shared by all members of the group). Straightforwardly, senders can use any common set of information in the destinations' Identity Table (e.g., specifying the working address and company name only to address destinations inside a company).

To show the HiBOp potential from this standpoint, we show results comparing HiBOp and Epidemic in an anycast scenario. We do not consider PROPHET as adapting it to group communication would require radical modifications to the protocol (this is indeed another advantage of HiBOp over it). As HiBOp, also Epidemic can be used to support anycast traffic as well without major modifications. As far as the performance indices, recall that we slightly modify the definitions of the delay and message loss rate, as explained in Section 7.3. Also recall that messages are always addressed to non-friend nodes, whose group is chosen uniformly at random for each message (also the node in the destination group is chosen uniformly at random).

The results we show in this section are derived by considering buffer limitations only, as results presented in the previous sections show that varying the buffer limits usually produces higher variability on the protocols' performance. Specifically, we compare the two extreme cases considered in this paper, i.e., unlimited buffers, and buffer size equal to 20 messages. In terms of resource consumption the results show that HiBOp is much more efficient than Epidemic also with anycast traffic (Fig. 22). As expected, Epidemic saturates the buffers while HiBOp does not. When no buffer limitations are considered the HiBOp buffer occupation is one order of magnitude below (Epidemic values are around 400, offscale). Similar remarks with respect to what already observed for unicast traffic holds true as far as the bandwidth overhead (Fig. 23). It is interesting to note that also in the anycast case HiBOp becomes more efficient than Epidemic also in terms of users' QoS as the resources are limited. Specifically, it achieves a significantly lower message loss rate (Table 10), and is also more efficient in terms of average delay (Table 11).

Overall, these preliminary results are a strong indication of the suitability of HiBOp to support both unicast and group communications without requiring any particular customization. This is a great advantage over other reference solutions.

10. Conclusions and future work

In this work we have proposed a context-based routing framework for opportunistic networks, and a specific routing protocol (HiBOp) exploiting context to model users' social behavior and base forwarding decision on this knowledge. The main idea of the framework is to use context information to complement, in the routing and forwarding tasks, the unreliable topological information about the state of the network, typical of opportunistic networks. The framework allows nodes to learn the context they are working in, to remember context information seen in the past, to enforce context information frequently encountered, and to identify suitable next hops to forward data by comparing context information of the candidate next hops and of the destination.

Another contribution of this paper is the fact that HiBOp model users' social behavior (i.e., users' information, users' behavioral patterns, users' acquaintances) through context information, and exploits the framework to build a social-inspired routing protocol for opportunistic networks. In HiBOp nodes are able to automatically learn how users behave and relate with each other, and exploit this knowledge to optimize data forwarding.

We have evaluated the HiBOp performance across a range of parameters' values, in comparison with Epidemic Routing and PROPHET. We have also considered both standard unicast communication models, and group communication scenarios. We have shown that HiBOp is able to drastically reduce the resource consumption, in terms of network traffic and nodes' buffer occupation. This is paid, in some configurations, with a message loss and delay increase, that is however tolerable for typical applications of opportunistic networks. Interestingly, when network resources become constrained, HiBOp becomes more efficient than the reference protocols as far as message loss and delay.

These results clearly indicate that context-aware forwarding based on users' social behavior is a very promising approach for opportunistic networks. Very interestingly, the HiBOp performance in terms of resource consumption show that context-aware routing also embeds features that allow this approach to automatically control congestion in opportunistic networks. With respect to the HiBOp protocols itself, there are several interesting directions of future work we plan to address. One of them is how to achieve fine-grain control on end-to-end reliability and to evaluate the sensitiveness of HiBOp to configuration parameters (e.g. relative weights of the context tables) and to variation in the predictability of user behavior. Another direction is finding analytical bounds for the performance of context-based forwarding. Finally, we also plan to investigate how to exploit HiBOp to proactively disseminate data in opportunistic networks. This is actually one of the most compelling issues to improve data availability in such challenged networking environments.

Regarding the large scale applicability of opportunistic routing protocols, there are still some open issues that should be addressed in future work. Mainly, privacy and security aspects (Section 6), which have never been addressed systematically in the opportunistic framework, should be taken into account and integrated with existing routing protocols. Moreover a complete analysis of the scalability of routing protocols for opportunistic networks should be developed, starting from the initial results under variable load presented in Section 8.4.

References

[1] N. Asokan, K. Kostiainen, P. Ginzboorg, J. Ott, C. Luo, Applicability of identity-based cryptography for disruption-tolerant networking, in: Proc. of the First ACM/SIGMOBILE Workshop on Mobile Opportunistic Networking, MobiOpp 2007, San Juan, Puerto Rico, 11 June 2007.
[2] D. Boneh, M. Franklin, Identity-based encryption from the Weil pairing, in: Proc. of the Conference on Advances in Cryptology, CRYPTO 2001, in: LNCS, vol. 2139, Springer-Verlag, August 2001, pp. 213–229.
[3] C. Boldrini, M. Conti, A. Passarella, Impact of social mobility on routing protocols for opportunistic networks, in: Proc. of the IEEE WoWMoM International Workshop on Autonomic and Opportunistic Communications, AOC 2007, Helsinki, Finland, June 18, 2007.
[4] C. Boldrini, M. Conti, I. Iacopini, A. Passarella, HiBOp: A History Based Routing Protocol for Opportunistic Networks, in: Proc. of the IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, WoWMoM 2007, Helsinki, Finland, June 18–21, 2007.
[5] J. Burgess, G.D. Bissias, M. Corner, B.N. Levine, Surviving attacks on disruption-tolerant networks without authentication, in: Proc. of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2007, Montréal, Québec, Canada, 9–14 September 2007.
[6] J. Burgess, B. Gallagher, D. Jensen, B.N. Levine, MaxProp: Routing for vehicle-based disruption-tolerant networks, in: Proc. of the 25th IEEE International Conference on Computer Communications, INFOCOM 2006, Barcelona, Spain, April 2006.
[7] B. Burns, O. Brock, B.N. Levine, MV Routing and capacity building in disruption tolerant networks, in: Proc. of IEEE INFOCOM 2005, Miami, FL, March, 2005.
[8] M. Conti, S. Giordano, Multihop ad hoc networking: The theory, IEEE Communications Magazine 45 (4) (2007).
[9] M. Conti, S. Giordano, Multihop ad hoc networking: The reality, IEEE Communications Magazine 45 (4) (2007).
[10] K. Fall, A delay-tolerant network architecture for challenged internets, in: Proc. of ACM SIGCOMM, 2003.
[11] C. Gentry, A. Silverberg, Hierarchical ID-based cryptography, in: Proc. of the 8th Intl. Conference on the Theory and Application of Cryptology and Information Security, December 2002.
[12] M. Grossglauser, M. Vetterli, Locating mobile nodes with EASE: Learning efficient routes from encounter histories alone, IEEE/ACM Trans. on Networking 14 (3) (2006).
[13] Haggle Project, Deliverable D4.1, Preliminary design of trust and security mechanisms. Available on-line at: http://www.haggleproject.org/images/d/d3/D4_1.eurecom.pdf, July 2007.
[14] S. Capkun, J.-P. Hubaux, L. Buttyán, Mobility helps peer-to-peer security, IEEE Transactions on Mobile Computing 5 (1) (2006) 43–51.
[15] N. Eagle, A. Pentland, Reality mining: Sensing complex social systems, Personal and Ubiquitous Computing 10 (4) (2006) 255–268.
[16] S. Jain, K. Fall, R. Patra, Routing in a delay-tolerant network, in: Proc. of ACM SIGCOMM, 2004.
[17] J. Kleinberg, The small-world phenomenon: An algorithmic perspective, in: Proc. 32nd ACM Symposium on Theory of Computing, 2000.
[18] J. Leguay, T. Friedman, V. Conan, Evaluating mobility pattern space routing for DTNs, in: Proc. of IEEE INFOCOM, 2006.
[19] A. Lindgren, A. Doria, O. Schelèn, Probabilistic routing in intermittently connected networks, ACM Mobile Computing and Communications Review 7 (3) (2003).
[20] S. Milgram, The small world problem, Psychology Today (1967) 60–67.
[21] A.L. Murphy, X. Chen, Enabling disconnected communication, in: Proc. of the Workshop on Software Engineering and Mobility, co-located with ICSE, Toronto, Canada, May, 2001.

[22] M. Musolesi, C. Mascolo, A community based mobility model for ad hoc network research, in: Proceedings ACM/SIGMOBILE REALMAN, 2006.
[23] M. Musolesi, S. Hailes, C. Mascolo, Adaptive routing for intermittently connected mobile ad hoc networks, in: Proc. of IEEE WoWMoM, 2005.
[24] M. Önen, A. Shikfa, R. Molva, Optimistic fair exchange for secure forwarding, in: Proc. of the First Workshop on the Security and Privacy of Emerging Ubiquitous Communication Systems, 10 August 2007, Philadelphia, PA, USA.
[25] J. Ott, Application protocol design considerations for a mobile internet, in: Proceedings of First ACM/IEEE international Workshop on Mobility in the Evolving internet Architecture, San Francisco, California, December 01–01, 2006, MobiArch '06, ACM, New York, NY, pp. 75–80.
[26] L. Pelusi, A. Passarella, M. Conti, Opportunistic networking: Data forwarding in disconnected mobile ad hoc networks, IEEE Communications Magazine 44 (11) (2006).
[27] N. Sarafijanovic-Djukic, M. Pidrkowski, M. Grossglauser, Island hopping: Efficient mobility-assisted forwarding in partitioned networks, in: Proc. of the 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, SECON 2006, Reston, VA, 28 Sept. 2006.
[28] J. Scott, P. Hui, J. Crowcroft, C. Diot, Haggle: A networking architecture designed around mobile users, in: Proc. of IFIP WONS, 2006.
[29] A. Seth, S. Keshav, Practical security for disconnected nodes, in: Proc. of the First IEEE ICNP Workshop on Secure Network Protocols, NPSec2005, Boston, Massachusetts, 6 November 2005.
[30] T. Spyropoulos, K. Psounis, C. Raghavendra, Efficient routing in intermittently connected mobile networks: The multi-copy case, ACM/IEEE journal of Transactions on Networking 16 (1) (2008) 77–90.
[31] T. Spyropoulos, K. Psounis, C.S. Raghavendra, Spray and wait: An efficient routing scheme for intermittently connected mobile networks, in: Proc. of the 2005 ACM SIGCOMM Workshop on Delay-Tolerant Networking, WDTN '05, Philadelphia, Pennsylvania, USA, August 26–26, 2005.
[32] T. Spyropoulos, K. Psounis, C.S. Raghavendra, Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility, in: Proc. of the Fifth IEEE international Conference on Pervasive Computing and Communications Workshops, March 19–23, 2007.
[33] Y. Tseng, S. Ni, Y. Chen, J. Sheu, The broadcast storm problem in a mobile ad hoc network, Wireless. Networks 8 (2–3) (2002) 153–167.
[34] A. Vahdat, D. Becker, Epidemic routing for partially connected ad hoc networks, Tech. Rep. CS-2000-06, CS Dept., Duke University, 2000.
[35] J. Van der Merwe, D. Dawoud, S. McDonald, A survey on peer-to-peer key management for mobile ad hoc networks, ACM Computing Surveys 39 (1) (2007).
[36] D.J. Watts, Small Worlds The Dynamics of Networks between Order and Randomness, in: Princeton Studies on Complexity, Princeton University Press, 1999.

Chiara Boldrini is a Ph.D. student at the Institute for Informatics and Telematics of the National Research Council (CNR), Italy. She holds a M.Sc. (2006) degree in Computer Engineering from the University of Pisa. She is working on socially-aware protocols and models for routing, data dissemination, and mobility in opportunistic networks.

Marco Conti is a research director at IIT, an institute of the Italian National Research Council (CNR). He co-authored the book "Metropolitan Area Networks" (1997) and is co-editor of the books "Mobile Ad Hoc Networking" (2004) and "Mobile Ad Hoc Networks: From Theory to Reality" (2007). He published in journals and conference proceedings more than 200 research papers related to design, modeling, and performance evaluation of computer-network architectures and protocols. He served as general chair of IEEE MASS 2007 and ACM REALMAN 2006, and as general co-chair of IEEE WoWMoM 2006 and of ACM MobiOpp 2007. He has been TPC chair of IEEE PerCom 2006, and of the IFIP-TC6 Conferences "Networking2002" and "PWC2003". He served as TPC co-chair of ACM WoWMoM 2002, WiOpt '04, IEEE WoWMoM 2005, and ACM MobiHoc 2006. He is Associate Editor-in-Chief of Pervasive and Mobile Computing Journal, and he is on the editorial board of: IEEE Transactions on Mobile Computing, Ad Hoc Networks journal and Wireless Ad Hoc and Sensor Networks: An International Journal.

Andrea Passarella is a Researcher at the IIT Institute of the National Research Council (CNR), Italy. Before joining IIT, he was a Research Associate at the Computer Laboratory of the University of Cambridge, UK. He received the Ph.D. and MS Degrees in Computer Engineering, both from the University of Pisa, Italy, in 2005 and 2001, respectively. His current research is mostly on opportunistic and delay-tolerant networking. More in general, he works on mobile ad hoc networks, specifically on p2p systems, multicasting, transport protocols, and energy-efficient protocols. He was TPC Co-Chair of ACM MobiOpp 2007, Vice-Chair for IEEE REALMAN 2005, ACM REALMAN 2006, and IEEE MDC 2006. He was Demo Co-Chair for IEEE MASS 2007, and is serving as Demo Chair for IEEE PerCom 2009. He served and is currently serving in the TPC of several international conferences, including IEEE PerCom, IEEE WoWMoM, IEEE MASS, and workshops. He is an Associate Technical Editor for IEEE Communications Magazine, and in the Editorial Board of the Inderscience Int. J. Autonomous and Adaptive Communications Systems.