Ad Hoc Networks 25 (2015) 314–329

Contents lists available at ScienceDirect

Ad Hoc Networks

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

# Weak social ties improve content delivery in behavior-aware opportunistic networks

Elena Pagani [a,b,*], Lorenzo Valerio [b], Gian Paolo Rossi [a]

[a] Computer Science Department, Università degli Studi di Milano, via Comelico 39, 20135 Milano, Italy
[b] Institute for Informatics and Telematics, National Research Council, via Moruzzi 1, 56124 Pisa, Italy

ARTICLE INFO

*Article history:*
Received 4 March 2014
Received in revised form 21 July 2014
Accepted 23 July 2014
Available online 9 August 2014

*Keywords:*
Opportunistic networks
Content-based addressing
Information-centric networking
Human social interactions

ABSTRACT

In the era of pervasive mobile computing, human encounters can be leveraged to enable new forms of social interactions mediated by the personal devices of individuals. In this framework, emerging needs, such as content dissemination, social discovery and question and answering, advocate the raising of novel communication paradigms where the binding content-recipients is not provided by the sender (in the classical IP addressing style), but directly executed by specific recipients with interest in it.

This paper proposes a novel communication protocol, named InterestCast, or *I*Cast, solving the problem for a wide range of social scenarios and applying to an opportunistic network whose nodes are the personal devices of moving individuals, possibly interacting with fixed road-side devices. The protocol is able to chase users' interests decoupling content tags from locations and social communities. In order to cross community boundaries and reach farthest destinations, *I*Cast adopts mechanisms that properly extract *weak ties*, i.e. encounters between nodes that rarely interact, but that connect different communities. The main advantages the proposal achieves are: it ensures remarkable performance results; it is simple and feasible and it keeps computational and networking costs low; it can preserve users' privacy.



## 1. Introduction

Real-life encounters are the oldest form of human communication where individuals mediate the (verbal) information exchange from source(s) to recipient(s) through either single or multi-hop paths. In the era of pervasive mobile computing, human encounters can be leveraged to provide an intermittently-connected network where the delivery of delay tolerant information is mediated by the personal devices of individuals. The growing interest in human social interactions is hastening the shift from human-mediated to computer-mediated communications. Emerging needs, such as content dissemination, social discovery and question and answering, inspire the creation of novel communication paradigms where the binding between content and recipients is not provided by the sender (in the classical IP-addressing style), but directly executed by specific recipients with an interest in it. According to this paradigm, unaddressed contents can be freely advertised on the network with a tag describing the content type, while human encounters drive the information flow towards potential recipients that extract the content from the stream when its type and local personal interest are matching. This is giving rise to an information-centric anycast communication capable to intercept the user's interests and to preserve, on the one hand, the communication resources and, on the other, the user's privacy. We call it *Behavior-cast*. Such a new

* Corresponding author at: Computer Science Department, Università degli Studi di Milano, via Comelico 39, 20135 Milano, Italy. Tel.: +39 02 50316271.

*E-mail addresses:* pagani@di.unimi.it (E. Pagani), lorenzo.valerio@iit.cnr.it (L. Valerio), rossi@di.unimi.it (G.P. Rossi).

http://dx.doi.org/10.1016/j.adhoc.2014.07.005
1570-8705/

form of interaction requires a suitable programming and networking platform supporting it.

The design of routing algorithms for *Behavior-cast* has to face with a number of challenging issues whose complexity is mainly due to the fact that the set of recipients is unknown, its cardinality is unpredictable and changes dynamically over time as a result of mobility and temporary disconnections, the users' behavior needs to be somehow defined and captured. This problem has become a hot research topic in Opportunistic Networks (ONs) and a few research works have recently faced with it. Yet, they assume that users sharing the same interests either have social relations or visit the same places; we survey them in Section 3. All the aforementioned assumptions greatly simplify the original problem because they inherently confine message delivery within a specific location and/or community. On the other hand, some real experiences show that human interests are not only bound to specific locations or closely assigned to a given community. In [8], it is argued that the most valuable data are obtained leveraging the weak social ties with people belonging to other social communities, rather than the strong ties with people within our same community. Indeed, in the latter case, knowledge amongst socially related people is quite uniform and communication does not bring any novel information. The *undersound* testbed application, developed in the framework of the BioNets Project [2], revealed that people are willing to exploit extemporary encounters with someone having the same interests in order to share content, even if no social ties exist. By contrast, social ties may arise *as a consequence* of a shared interest. In [14], the authors observe that the correlation among all planes is not immediate, while in [25] it is shown that this correlation varies with changing scenarios. These and other results in the literature [17,11] show that *weak ties* are fundamental in order to bridge the gap between different communities and reach out all the target destinations of a certain content.

The **contribution** of this paper is manifold. First, we define the novel *BehaviorCast communication paradigm* and introduce a functional architecture to support it. Second, we present a novel routing algorithm (*InterestCast*, or *I*Cast for short) that solves the problem in an opportunistic network of individuals, possibly interacting with fixed stations [29]. The algorithm comes in two flavors: basic (b*I*Cast) and weighted (w*I*Cast). We adopt utility functions that find weak ties and properly chase the user's interests independently of locations and social communities. While b*I*Cast does so in a stateless and privacy-preserving manner, w*I*Cast provides better effectiveness at the expenses of some state maintenance. The capability of the algorithms of solving the BehaviorCast problem is formally analyzed. Their performances are measured through simulation techniques in realistic scenarios, and compared with other solutions in the literature that cope with the problem in a subset of the conditions discussed above. *I*Cast may also be used to support communication according to the *publish-subscribe* paradigm. Indeed, it provides space, time and synchronization decoupling [9] between content producers and consumers. It realizes a sort of distributed event service where each node may contribute to convey content to interested nodes. According to [9], *I*Cast implements the content-based scheme.

The main advantages of this proposal are: the algorithms succeed both in keeping coverage very high, and in reducing the consumption of network resources in comparison with the other solutions in the literature. This is achieved without observing any significant increase of delivery latency. The algorithm adopts utility functions with low computational cost, that each node autonomously computes by using only local information collected during encounters with other nodes, i.e. the solution is distributed, does not require any global information, and is thus viable in practice. The analysis shows that both algorithms have a cost $O(|I|)$ – with $I$ set of interested destinations – in terms of number of transmissions performed by a node, number of content replicas contemporarily existing in the system, and, for w*I*Cast, memory overhead for state information.

## 2. Problem definition and system assumptions

We consider a mobile network composed of $N$ nodes that communicate through wireless links. A node may be either the personal device of a user, which moves with him/her, or a fixed station, as in the case of a road-side gateway to/from a wired network. Thus, we are considering a hybrid urban network infrastructure [13,18,34]. Throughout this paper, all nodes, both fixed or mobile, have the same capabilities; each node operates as source, recipient and forwarder of messages with specified interests. We assume without loss of generality that only one interest $\mathcal{I}$ is assigned to a node. The purpose of the algorithm is to deliver a message to (approximately) all nodes matching the interest $\mathcal{I}$.[1]

A message (Fig. 1) contains: the source identifier, the target expressed as a `tag` indicating the interest(s) matching the content, a `lifetime` constraining the existence of the content in the system, a `content ID`, a `type flag`, and the content. When a node $n$ receives a content $c$, $n$ records the arrival time of $c$. When $c$ is forwarded to another relay, the `lifetime` is updated by subtracting to the current value the amount of time that $c$ spent in $n$. No global synchronization is required. When the `lifetime` reaches 0, $c$ is dropped from buffers and no more exchanged. The `content ID` uniquely identifies the content. The `type` flag indicates whether the content matches a *local* interest and it is forwarded to the node only for delivery, or it must also be buffered for *relay*ing to other nodes. Addressing is performed on a per-content basis. Content's `tags` and node interests do not have to match exactly. Folksonomic reasoning [20] is used to match interests with content `tags` and when a matching is verified the message is delivered to the local recipient.

The problem we consider in this work is defined as follows:

**Definition 1** (*BehaviorCast Problem*). Let $I$ be a set of recipients with a common interest $\mathcal{I}$. Let U be the set of

[1] Complete coverage of the destinations might be hindered by partitions.

| source | interest tag (target) | lifetime | type | content ID | data |
|---|---|---|---|---|---|

**Fig. 1.** Structure of a content message.

all nodes in the system. An algorithm solving the BehaviorCast Problem for a content $c$ labeled with interest $\mathcal{I}$ must ensure the following properties:

- **Validity:** (i) $c$ must be delivered to a subset $I' \subseteq I$ of recipients, but (ii) no node in $\bar{I} = U - I$ must deliver $c$.
- **Effectiveness:** the service should approximate total coverage, i.e., it should keep the cardinality of $I'$ as close as possible to that of $I$.
- **Efficiency:** the service should minimize both (i) the number of nodes $\in U - I$ used as relays, and (ii) the number of transmissions performed by those nodes.
- **Eventual Termination:** within a finite time from the generation of $c$, no more messages are exchanged containing $c$, and no more memory overhead is paid to store $c$.

The validity property guarantees that recipients without interest about $\mathcal{I}$ are not spammed with unwanted messages, that is, it excludes broadcast (adopted, e.g., in [34]). The effectiveness property has to do with user satisfaction. In order to achieve it, nodes outside $I$ may be used as relays if they increase the probability of reaching the destinations. However, the efficiency property excludes resorting to trivial solutions such as epidemic diffusion to solve the problem. Without the eventual termination property, an algorithm could work indefinitely, thus guaranteeing perhaps that coverage asymptotically tends to 100%, but also implying an infinite overhead. In Section 5, we provide a formal analysis to prove that the proposed algorithms fulfill the properties of the BehaviorCast problem.

To determine the intended recipients of a message, we introduce the *Timed Delivery* Model (TDM). In TDM, the delivery of a message is constrained within the time interval $[t_1,\ t_2]$, where $t_1$ is the time of message generation and $t_2$ is the message expiry time. When a message $m$ with label $\mathcal{I}$ is issued, the intended recipients of $m$ are all nodes in $I$ that are reachable within the time interval $[t_1, t_2]$.[2]

In order to prove that the proposed algorithms fulfill Definition 1, the following assumption on mobility is used:

**Definition 2** (*Encounter assumption*). If two nodes $n_1$ and $n_2$ have shown habit of encounters in the past, hence this habit preserves in the future. This can be expressed formally as follows: let $\lambda_{n_1,n_2}(t)$ be the contact rate at time $t$ with which $n_1$ and $n_2$ encounter. Then $\lambda_{n_1,n_2}(t) \approx \lambda_{n_1,n_2}(t + \Delta t),\ \forall n_1, n_2 \in U,\ \forall t \in \mathbb{N}$.

In Fig. 2, a reference functional architecture is represented; continuous arrows mark communications amongst modules, while dashed arrows indicate data flows; the numbers in the figure indicate the steps explained hereafter. A user may configure the interests for which s/he is willing to receive content in a local table (1). When a user generates a new content, it is stored in the local repository (2), and passed for diffusion via the service primitive `send.Req` to the underlying BehaviorCast module (BCM) which stores it in its local buffer (3). When a node $n$ has a contact with another node, the link layer entity responsible for channel set-up and exchange of Layer 2 beacons notifies the presence of a new neighbor in a Layer 3 table (4). The mechanisms adopted to detect the existence of new neighbors, set up a channel with them and access the channel depend on the specific wireless technology adopted. Nodes that enter in reciprocal radio range exchange Layer 3 beacons carrying state information – such as e.g. the local interest and a summary of the held content – according to the implemented algorithm (5). This information is obtained from the BCM, which has access to the user interests (6). The state information is used to update the local knowledge maintained by $n$ (7) to compute its own utility and the utility of the other node for each message held in the buffer using the utility engine included in the BCM (8). A message in the local buffer is sent to the other node when either the message interest matches against the other node's interest or the other node has higher utility than $n$ for the message interest (9). Received messages are stored in the buffer for further dissemination. Moreover, the BCM module uses its interest engine to match the content interest against the local interest (6): messages that are of local interest are notified to the user via the `delivery.Ind` service primitive and inserted in the local content repository (10). The algorithms solving the BehaviorCast problem implement the BCM.

## 3. Related work

The design of a platform for BehaviorCast is still in its infancy. Solutions adhering to the IP multicast model are not suitable for ONs as they assume a global a priori knowledge of the group membership (e.g., [5,12,19,37]) and possibly try to form a tree-like routing infrastructure (e.g., [36,35]). By contrast, the behavior of a user can be defined considering several aspects, and different behaviors shall be considered for content diffusion. As an example, in [30], the temporal, spatial, and activity profiles of the users are studied. In [23], user behavior is analyzed in terms of visited locations and accessed web domains.

Some preliminary works taking into consideration users' interests for message diffusion, actually do not solve the BehaviorCast problem. MobiClique [31] allows a user to *flood* messages to all members of an interest group (taken from social networks) s/he belongs to. Both PeopleRank [26] and ML-SOR [32] support unicast communications and consider the number of interests shared amongst nodes to derive indicators able to single out promising relays to reach a certain destination. In [1,24], interests are used to diffuse content among similar nodes, with the aim of maximizing the satisfaction of each user in terms of his/her interest in the received content, rather than of maximizing the probability for a node to receive all contents of interest.

[2] Our model is similar to the *Temporal Membership Model* in [37].

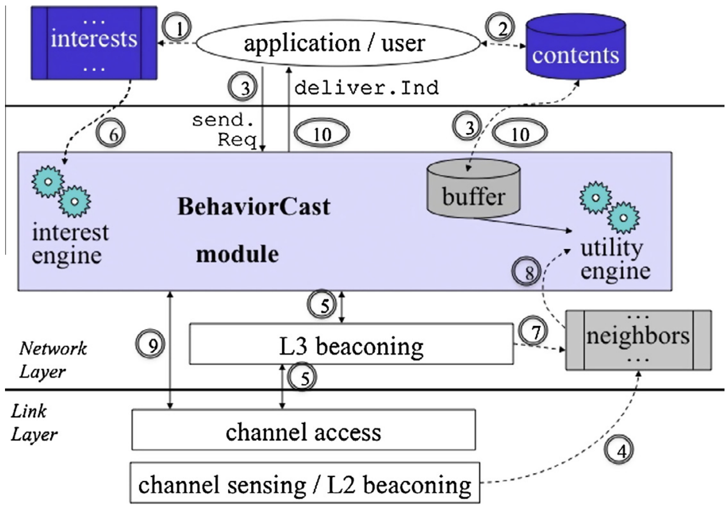


**Fig. 2.** Functional architecture.

To solve problems more similar to BehaviorCast, some preliminary results are obtained by starting from the basic assumption that user's interests and customarily frequented locations are closely related to one another. ProfileCast [15] belongs to this category. A content generated by a node is addressed to ("is of interest for") nodes used to visit the same locations as the source. Another class of approaches (SocialCast [7], ONSIDE [6], SANE [22]) goes a little further by assuming that users with the same interests have the attitude to meet with each other more often than with other users. These are the solutions in the literature more similar to the *I*Cast algorithms proposed in this work and that we use for performance comparison in Section 6. In particular, we consider ProfileCast, and SocialCast as the representative of its class as it is the most stable proposal. They are described in more detail in the following subsections for readers' convenience. Habit [21] considers a different granularity in that nodes indicate a list of *sources* whose content they are interested in. By exchanging this information, each node builds a content dissemination network that is used by a source to compute the whole paths – forming a sort of a tree – to reach the interested users; these paths are included in the message and used as in source-routing.

In Section 4, we describe in details the *I*Cast algorithms, which solve the BehaviorCast problem in ONs reasoning with a content-based granularity. They adopt a fully distributed approach, do not require any a priori knowledge on the system nor make any assumption about either the interests nature or the reciprocal mobility attitudes of nodes with similar interests. This work improves our previous papers [28,27] in several respects, mainly in introducing a novel variant of the basic algorithm and in performing a more accurate algorithm analysis.

### 3.1. ProfileCast

In ProfileCast [15], nodes must a priori agree on a set of reference locations. For a given time slot $t$, each node $n$ records the time ratio spent at each location in that slot, thus generating an *association vector* $AV_n^t$ with as many entries as locations. The association matrix $AM_n$ is built, whose column $i$ is $AV_n^i$, with $i \leqslant$ number of time slots observed so far. Hence, the association matrix represents the behavior of the node along its history. The association matrix is decomposed as $AM = U\,D\,V^T$ by using *Singular Value Decomposition* (SVD). The left singular vectors in $U$ capture – in decreasing order of importance – the node behavior in terms of time spent in the locations. The $i$-th vector $u_i$ has an associated weight, computed from the diagonal matrix $D$ of singular values as $d_{ii}/\sum_{\forall j} d_{jj}$, that describes the significance of the vector.

When two nodes $n_j$ and $n_k$ encounter, they exchange their first $x_j$ and $x_k$ left singular vectors, such that for each node the sum of the weights of the sent vectors in not less than a threshold $\theta_W$. In [15], node similarity $S_{jk}$ is defined as:

$$S_{jk} = \sum_{J=1}^{x_j} \sum_{K=1}^{x_k} weight_{u_J^j}\; weight_{u_K^k} \left| u_J^j \cdot u_K^k \right| \quad (1)$$

If, let us say, $n_j$ owns a message $m$ to be distributed to similar nodes, and $S_{jk} \geqslant \theta_{PC}$ for a certain threshold $\theta_{PC}$, then the two nodes are considered *sufficiently* similar, and $n_j$ forwards $m$ to $n_k$ that is considered an interested node. This procedure is summarized in Fig. 3(a).

ProfileCast has been recently extended (CSI:D [16]) so as to allow dissemination of content when the target profile is independent of the behavior profile expressed as visited locations. The goal is pursued by passing the content to some message holders with very different behavior profiles, in the attempt of randomly taking dissimilar paths in the encounter plane (Fig. 3(b)) so as to span it and to have some probability that the holders run into the destinations similar in the behavior plane. In this work, we choose to study the original ProfileCast, in order to avoid introducing biases due to the policy of holder selection, thus guaranteeing a fairer comparison.

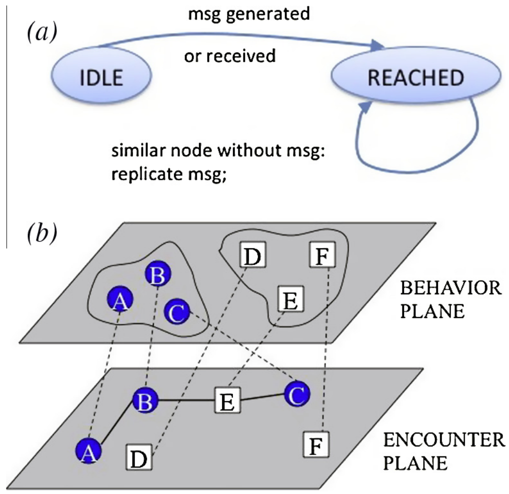


**Fig. 3.** (a) Finite state automaton for a ProfileCast node and (b) difference between behavior similarity and encounter attitudes.

### 3.2. SocialCast

In [7], the SocialCast algorithm is proposed for content diffusion in ONs according to the publish-subscribe paradigm. The solution heavily relies on the assumption that nodes having a common interest also tend to encounter more often than with other nodes. Given a certain interest $\mathcal{I}$, the routing of messages labeled with $\mathcal{I}$ leverages a utility function based on two factors, namely (i) the probability $U_{col}$ of being colocated in the future with nodes having $\mathcal{I}$ as their interest, and (ii) the estimation $U_{cdc}$ of the change degree of connectivity. The former bases on the mentioned assumption to select promising relays to the group of interested nodes. The latter aims at selecting nodes whose neighborhood frequently changes. Those nodes are of interest because they have more options to disseminate information, and to encounter either good relays or interested nodes. In detail, for each interest $\mathcal{I}$, in each node $n$, the two factors are sampled with frequency $\tau$ this way:

$$U_{col_{n,\mathcal{I}}}(t) = \begin{cases} 1 & \text{if } n \text{ was located with a} \\ & \text{subscriber for } \mathcal{I} \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$$

$$U_{cdc_{n,\mathcal{I}}}(t) = \frac{|\mathcal{N}(t-\tau) \cup \mathcal{N}(t)| - |\mathcal{N}(t-\tau) \cap \mathcal{N}(t)|}{|\mathcal{N}(t-\tau) \cup \mathcal{N}(t)|}$$

where $\mathcal{N}(t)$ is the set of neighbors of $n$ at time $t$. Leveraging the *encounter* assumption (Definition 2), the future values of both $U_{col}$ and $U_{cdc}$ are predicted using Kalman filters, obtaining $\widehat{U}_{col}$ and $\widehat{U}_{cdc}$ respectively. In [7], the predicted values are used in the computation of the utility $\mathcal{U}$, to decide message exchange:

$$\mathcal{U}_{n,\mathcal{I}} = w_{col}\ \widehat{U}_{col_{n,\mathcal{I}}} + w_{cdc}\ \widehat{U}_{cdc_{n,\mathcal{I}}} \tag{2}$$

with $w_{col}$ and $w_{cdc}$ weights assessing the relative importance of the two factors.

With period $T$, each node $n$ updates its own utility for each interest, and broadcasts the values to its own one-hop neighbors, together with its own interest and the identifiers of the held content (*interest dissemination* phase). For each neighbor $\bar{n}$ whose message is received, and for each interest $\mathcal{I}$, $n$ evaluates whether $U_{\bar{n},\ \mathcal{I}} > U_{n,\mathcal{I}} + \varepsilon$; in this case, $\bar{n}$ is a better relay than $n$ for the content tagged with $\mathcal{I}$. Then, all data matching $\bar{n}$'s interest, and all data for which $\bar{n}$ is a better relay, are forwarded to $\bar{n}$. As far as the latter data are concerned, they are discarded from $n$'s buffer. Duplicate forwarding is avoided by transferring the data not already owned by $\bar{n}$, according to the list it broadcasted in the interest dissemination phase. Associating either a TTL or a lifetime to data prevents their indefinite existence in the system. Fig. 4 summarizes the procedure.

## 4. InterestCast

In this Section we consider the problem of forwarding a content to destinations belonging to the set of neighbor in the *behavior plane* but not necessarily in the *encounter plane* (see Fig. 3(b)). Nodes A, B, and C are neighbors in the behavior plane as they share a common interest $\mathcal{I}_1$.

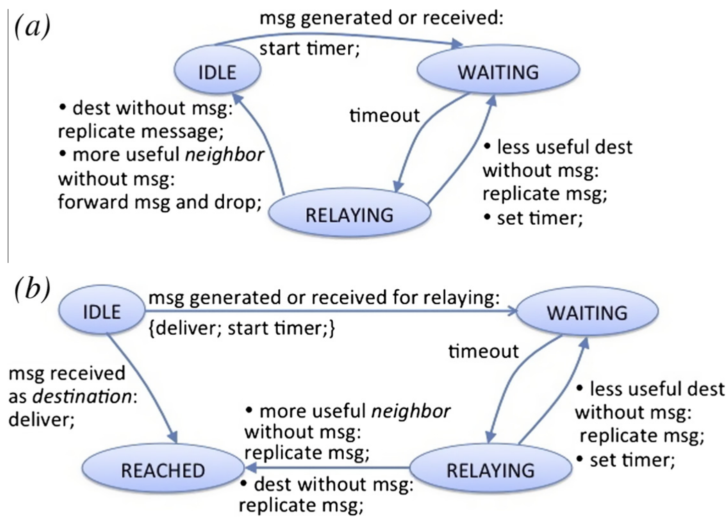


**Fig. 4.** Finite state automata for (a) a SocialCast relay and (b) a SocialCast destination.

Similarly, nodes D, E, and F share a common interest $\mathcal{I}_2$. Yet, neighbor nodes in the behavior plane may be far from one another in the encounter plane, as they are not used to encounter. In order to fulfill the effectiveness property in Definition 1, an algorithm should resort to "bridge" nodes, able to connect nodes that are neighbors in the behavior plane but not in the encounter plane (such as nodes B and C, that shall communicate through E). *I*Cast aims at discovering those bridges and using them for message relaying. As a consequence of having multiple recipients sharing a common interest $\mathcal{I}$, we can expect to have potentially several relays involved, and that their selection should be influenced by their inclination to encounter nodes declaring $\mathcal{I}$ as an interest. We propose two utility functions for basic and weighted *I*Cast (b*I*Cast and w*I*Cast respectively). Both aim at identifying *weak ties* among communities, which allow to span the nodes in the system and maximize the probability of reaching all the destinations. b*I*Cast is stateless and preserves privacy; yet its performance – although very good on average (Section 6), – may suffer local maximums. w*I*Cast guarantees high effectiveness in every situation (Section 5) at the expenses of some state maintenance.

**Algorithm 1.** `Basic` *I*`Cast`

1: INIT: `counter` $\leftarrow$ []; buffer $\leftarrow \emptyset$;
2: **when** contact with node $\bar{n}$ **do**
3: receive ($\mathcal{I}_{\bar{n}}$) from $\bar{n}$;
4: send (my$\mathcal{I}$) to $\bar{n}$;
5: **if** (`counter`[$\mathcal{I}_{\bar{n}}$] already exists for interest $\mathcal{I}_{\bar{n}}$) **then**
6: `counter`[$\mathcal{I}_{\bar{n}}$] $\leftarrow$ `counter`[$\mathcal{I}_{\bar{n}}$] +1;
7: **else**
8: allocate `counter`[$\mathcal{I}_{\bar{n}}$] $\leftarrow$ 1;
9: **end if**
10: my$\mathcal{U} \leftarrow \{\forall$ known $\mathcal{I}$'s, `counter`[$\mathcal{I}$]$\}$;
11: send my$\mathcal{U}$ to $\bar{n}$;
12: receive $\mathcal{U}(\bar{n})$ from $\bar{n}$;
13:
14: **for all** messages $m$ in my buffer **do**
15: //(let $\mathcal{I}_m$ be the interest to which $m$ is addressed)
16: **if** ($\mathcal{I}_m == \mathcal{I}_{\bar{n}}$) **then**
17: send $m$ to $\bar{n}$ and keep copy;
18: **else if** ($\mathcal{U}_{\mathcal{I}_m}(\bar{n}) >$ my$\mathcal{U}_{\mathcal{I}_m}$) **then**
19: send $m$ to $\bar{n}$;
20: **if** (I'm not interested in $\mathcal{I}$) **then**
21: remove copy from buffer;
22: **else**
23: keep copy;//for other destinations *only*
24: **end if**
25: **end if**
26: **end for**
27: receive messages from $\bar{n}$ and put them into buffer;
28: deliver to application the messages tagged with my$\mathcal{I}$;
29: **end do**

For b*I*Cast, let us indicate with $C(t)$ the set of contacts occurred in the system up to time $t$. Let us define an encounter by the tuple $\langle n1, n2, t\rangle$, such that at time $t$ nodes $n1$ and $n2$ enter in mutual communication range.[3] Then, $\forall \bar{t} \geqslant t,\ \langle n1,\ n2,\ t\rangle \in C(\bar{t})$. The utility function for a node $n$ at time $t$ w.r.t. a generic interest $\mathcal{I}$ is:

$$\begin{aligned} \mathcal{U}_{\mathcal{I}}(n,t) &= |\{\forall t' \leqslant t, \forall \bar{n} \in I,\ \ \bar{n} \neq n : \langle n, \bar{n}, t'\rangle \in C(t)\}| \\ &= \sum_{x \in C(t)} 1_I(x) \end{aligned} \tag{3}$$

which counts the number of encounters that node $n$ had with interested nodes, in the past. The function is computed by each node $n$, using just local knowledge, every time it encounters a node whose beacon includes $\mathcal{I}$. Nodes do not need to know beforehand the set of interests of the other nodes; rather, when an unknown interest is discovered from a neighbor's beacon, a new element is allocated for it and initialized to 1.

Algorithm 1 supplies the pseudo-code for b*I*Cast executed by a generic node $n$. The local knowledge construction and the utility function computation are in lines 5–9. A relevant aspect of the algorithm is the message replication mechanism. Whenever a node $n$, with no interest in $\mathcal{I}$, forwards a message $m$ to a node with higher utility (lines 18–19), $n$ delegates the other node to continue forwarding, and hence removes the copy of $m$ from its own buffer (line 21). By contrast, if $n$ forwards the message to a legitimate recipient, then $n$ maintains the message copy (line 17). In fact, its habit of encountering recipients in $I$ might be useful for delivering $m$ to other destinations. Nodes in $I$ always maintain the message (line 23) and they can forward a copy to other recipients (line 16); yet, they forward the message to a more useful relay at most once in order to prevent a proliferation of replicas. We assume that, as in [33], summary vectors are exchanged in Layer 3 beacons – together with local interests – to prevent forwarding of duplicate messages. Fig. 5 summarizes the procedure and allows to compare the three approaches we consider in this work (Figs. 3(a) and 4).

The advertisement of content on the network is worth continuing until the time validity of the tagged message expires. This is the motivation underpinning the `lifetime` we introduced in Section 2. In Section 6, we show through simulations that the algorithm has also an interesting self-stabilization property: once a content $c$ is carried by high-utility relays, no more forwardings take place apart to destinations and the traffic generated for $c$ settles down.

### 4.1. Weighted InterestCast

The performance of b*I*Cast can be improved by removing the limitation of the utility in Eq. (3) due to the fact that it has no memory of encounters. As an example, consider two nodes $p_1$ and $p_2$ such that both have utility – according to b*I*Cast – $u_b = 100$ resulting, in case of $p_1$, from encountering twice 50 different nodes and, in case of $p_2$, from encountering 20 different nodes, 5 times each. Intuitively,

[3] I.e., $t$ is the time at which the contact begins.

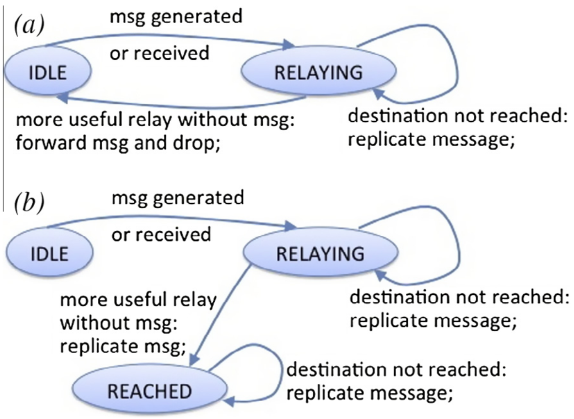


**Fig. 5.** Finite state automata for (a) an *I*Cast relay and (b) an *I*Cast destination.

$p_1$ is more useful to boost coverage. The problem can be overcome by adopting a utility function that awards nodes with a larger variety of encountered nodes in *I*. This is achieved by resorting to Shannon's entropy: each node *n* records a list $\texttt{recipients}_{\mathcal{I}}$ whose size is upper bounded by $|I|$ such that $\texttt{recipients}_{\mathcal{I}}(x)$ is the number of encounters *n* had with $n_x \in I$. According to the described variation, the *weighted* utility $\mathcal{U}^w_{\mathcal{I}}(n,t)$ is computed as:

$$\mathcal{U}^w_{\mathcal{I}}(n,t) = -\sum_{\forall n_x \in I} \frac{\texttt{recipients}_{\mathcal{I}}(x)}{\sum_{\forall k}\texttt{recipients}_{\mathcal{I}}(k)} \cdot \log\left(\frac{\texttt{recipients}_{\mathcal{I}}(x)}{\sum_{\forall k}\texttt{recipients}_{\mathcal{I}}(k)}\right) \tag{4}$$

Algorithm 2 supplies the pseudo-code for w*I*Cast executed by a generic node *n*, with highlighted lines emphasizing the differences with b*I*Cast. Both the forwarding mechanism and the interest discovering mechanism of w*I*Cast are the same as b*I*Cast. Nodes do not need to know a priori the number of devices in the system: when a new node is discovered with interest in $\mathcal{I}$, a new element is allocated in $\texttt{recipients}_{\mathcal{I}}$ and initialized to 1 (line 8).

**Algorithm 2.** `Weighted` *I*`Cast`

1: INIT: buffer $\leftarrow \emptyset$; `recipients` $\leftarrow$ [];
2: **when** contact with node $\bar{n}$ **do**
3: receive $\mathcal{I}_{\bar{n}}$ from $\bar{n}$;
4: send my_$\mathcal{I}$ to $\bar{n}$;
5: **if** ($\texttt{recipients}_{\mathcal{I}_{\bar{n}}}(\bar{n})$ already exists for $\mathcal{I}_{\bar{n}}$ **and** $\bar{n}$) **then**
6: $\texttt{recipients}_{\mathcal{I}_n}(\bar{n}) \leftarrow \texttt{recipients}_{\mathcal{I}_n}(\bar{n}) + 1$;
7: **else**
8: allocate $\texttt{recipients}_{\mathcal{I}_n}(\bar{n}) \leftarrow 1$;
9: **end if**
10: update my_$\mathcal{U}^w$ according to Eq. (4);
11: send (my_$\mathcal{U}^w$) to $\bar{n}$;
12: receive ($\mathcal{U}^w(\bar{n})$) from $\bar{n}$;
13:
14: //... message exchange as in Algorithm 1
15: **end do**

The use of $\mathcal{U}^w_{\mathcal{I}}$ generates a remarkable performance improvement (see, Section 6) that might be payed with the memory space each node has to allocate and whose size grows with the growth of considered interests, nodes, or both. Yet, the memory overhead for a certain interest $\mathcal{I}$ is $O(|I|)$.

## 5. Analysis of *I*Cast

In this section, we analyze how the *I*Cast algorithms solve the BehaviorCast problem according to Definition 1. The *Validity* property is guaranteed by the definition of the `delivery.Ind` primitive provided by the architecture. The *Eventual Termination* property is guaranteed by the `lifetime` management policy. Hence, we analyze just *efficiency* and *effectiveness* of the proposed algorithms. We adopt the same notation as in Definition 1. Let $F^m_t \subseteq \bar{I}$ be the set of non-interested nodes used as relays for a message *m* at a time *t*. Let $M^m_t$ be the set of nodes holding the message *m* at a time *t*. Let $[t^m_0, T_m]$ be the interval in which *m* exists in the system. Finally, let $s^m_u$ be the number of times a generic node $u \in U$ forwards the message *m*. The following property holds for both b*I*Cast and w*I*Cast as they adopt the same forwarding policy.

**Property 1** (*Efficiency*). *(1) The maximum number of nodes* $\in \bar{I}$ *used as relays for a message m is* $O(|I|)$. *(2) The maximum number of transmissions for a message m performed by a generic node is* $O(|I|)$.

**Proof.** According to the algorithms' pseudo-codes, a copy of a message *m* is created just when: (i) a node $u \in U$ replicates *m* to a node $k \in I$; or (ii) a node $u \in I$ encounters $k \in \bar{I}$ with $\mathcal{U}_{\mathcal{I}}(k,t) > \mathcal{U}_{\mathcal{I}}(u,t)$. Therefore, the maximum number of *m*'s replicas corresponds to the situation when *m* is generated by a node $\in \bar{I}$, all nodes $\in I$ receive *m* and each of them forwards the message to a distinct relay $\in \bar{I}$. Hence, the cardinality of $M^m_t$ is bounded by:

$$|M^m_t| \leqslant 2 * |I| + 1, \quad \forall t \in [t^m_0, T_m] \tag{5}$$

And, the number of nodes $\in \bar{I}$ contemporarily used as relays is at most

$$|F^m_t| \leqslant |M^m_t| - |I| \leqslant |I| + 1 \tag{6}$$

In the worst case, the number of transmissions that a node $u \in U$ has to do corresponds to the situation in which *u* contacts all the destinations before encountering another node $\in \bar{I}$ with a better utility. Then, the maximum number of transmissions required to *u* for the message *m* is bounded by

$$s^m_u \leqslant |I| + 1 \tag{7}$$

Thus for a given message, the maximum number of transmissions required to a node *u* is $O(|I|)$. This is very profitable if we consider that in real world scenarios $|I| \ll |\bar{I}|$.

As far as *effectiveness* is concerned: in the following, let us consider a network where nodes are organized in two closed communities, $z_1$ and $z_2$, connected only by one traveller. Interested nodes are present in both communities and they are not travellers. The traveller is the *weak tie* between communities. Moreover, communities are assumed to be physically far from each other, then the time spent by the traveller to move to and from the communities affects its contact rate. Finally, mobility inside communities is assumed to be exponential. Let $s \in U$ be a non traveller node belonging to either $z_1$ or $z_2$. Let $\tau \in \bar{I}$ be a traveller node. Let $\lambda_{s,s'}(t)$ and $\lambda_{\tau,s'}(t)$ be the contact rate at time $t$ with which nodes $s$ and $\tau$ meet a node $s'$, respectively. $\square$

**Assumption 1.** Due to the time needed by a traveller to move from $z1$ to $z2$, it is reasonable to assume the following relation

$$\lambda_{s,s'}(t) > \lambda_{\tau,s'}(t), \quad \forall s, s', \tau \in U \tag{8}$$

According to Assumption 1 from now on the time variable for contact rates will be omitted. Moreover $\bar{\lambda}_s$ ($\bar{\lambda}_\tau$) denotes the mean contact rate for the node $s$ ($\tau$) w.r.t. all the other $s' \in U$.

**Property 2** (*Effectiveness*). *Through the utility function $\mathcal{U}$ the algorithm identifies the most useful relays to be used in order to approximate the total coverage.*

**Proof** (*bICast*). Considering the utility function of Eq. (3), for each time interval $\Delta t = [t_1, t_2]$ , the sequence of increments of $\mathcal{U}$ can be interpreted as a stochastic process $\mathbf{V_u} = \{V_i\}_{i=t+\Delta t}$ where $V_i \sim Binomial(\bar{n}_u, p_u)$, with $\bar{n}_u = \bar{\lambda}_u \Delta t$ and $p_u = \frac{|K \subseteq I \cap U'|}{|U'|}$. Here $U'$ is the set of nodes belonging to the communities visited by $u$ and $K$ is the set of interested nodes $\in U'$. For each contact, nodes $u_1,\ u_2 \in U'$ have the same probability of encountering a node $\in I$:

$$p_{u_1} = p_{u_2} \iff K_{u_1} = K_{u_2}$$

Thus, their utility functions are affected only by the order relation between their contact rates $\bar{\lambda}_{u_1},\ \bar{\lambda}_{u_2}$:

$$\mathcal{U}_\mathcal{I}(u_1, t) > \mathcal{U}_\mathcal{I}(u_2, t) \iff \bar{\lambda}_{u_1} > \bar{\lambda}_{u_2} \Rightarrow \bar{n}_{u_1} p_{u_1} > \bar{n}_{u_2} p_{u_2} \tag{9}$$

In a single community scenario, b*I*Cast is able to select the most useful relays according to the value of their utility function. However, in a scenario with communities connected by travellers, the total coverage is not guaranteed. Indeed, according to (8) the contact rate of a sedentary node $s$ is reasonably greater than that of a traveller $\tau$, due to the time spent by $\tau$ for moving between communities. Hence, on average we have the following order relation

$$\mathcal{U}_\mathcal{I}(s, t) > \mathcal{U}_\mathcal{I}(\tau, t)$$

that has the effect of preventing to relay a message to a weak tie to other communities and to reach far destinations. Therefore, b*I*Cast might not guarantee the Effectiveness property. $\square$

**Proof** (*wICast*). Let us now consider two nodes $s$ and $\tau$ such that $\tau$ is a traveller node that visits both community $z_1$ and $z_2$, while $s$ moves only inside community $z_1$. Let us denote with $U' \subseteq U$ the set of nodes belonging to the communities visited by $\tau$, and with $K \subseteq U'$ the set of interested nodes $\in I$. Then, due to the habits of $s$ and $\tau$ the following relation holds:

$$|D_s(t)| < |D_\tau(t)| \iff |K_s| < |K_\tau| \tag{10}$$

where $D_s(t), D_\tau(t) \subseteq I$ are the set of interested nodes met by the node $s,\ \tau$ up to time t, respectively. Thus, a traveller can meet a greater variety of different nodes $\in I$ w.r.t. a non traveller node. This ability is well captured by the Shannon entropy, used in w*I*Cast as utility function. Let us rewrite:

$$\hat{f}_u(d) = \frac{\texttt{recipients}_\mathcal{I}(d)}{\sum_{\forall c \in C(t)} \texttt{recipients}_\mathcal{I}(c)} = \frac{\sum_{\forall c \in C(t)} \delta_{c,d}}{\sum_{\forall c \in C(t)} 1_I(c)} \tag{11}$$

The function $\hat{f}_u$ is the empirical probability mass function that measures the conditional probability the node $u$ has to meet the destination $d$ given that the encountered peer is a node $\in I$, and $\delta_{c,d}$ is the Kronecker delta function. Every node $u$ in the system stores these empirical probabilities into a stochastic vector $\mathbf{d}_u$ of size $|D_u(t)|$. Given that for each

$$\sum_{i=1}^{|D_u(t)|} d_i = 1, \quad \forall u \in U$$

we have the following implication

$$\mathcal{U}_\mathcal{I}(\tau, t) > \mathcal{U}_\mathcal{I}(s, t) \iff |D_\tau(t)| > |D_s(t)| \tag{12}$$

In this way w*I*Cast avoids to get trapped inside a community. In fact, a traveller which encounters more interested nodes is always preferred w.r.t. a non traveller. We point out that this is far better than simply counting how many destinations a traveller has encountered. That is, the use of entropy as utility value permits to differentiate between travellers with the same number of encountered destinations because it considers also the frequencies with which they use to meet them. In other words we are considering both the quantity and the quality of a traveller. In fact, given the same number of encountered destinations, more homogeneous contact frequencies are preferred. On average, this is profitable in situations when two travellers use to meet the same set of destinations though at different rates. In light of this, the Effectiveness property for w*I*Cast is derived.

As an example, we use an unrealistic yet demonstrative setting, represented by the trace HEXT produced with HCMM [4], whose characteristics are shown in the last line of Table 1. This trace describes an environment where encounter communities are strongly closed (see, social modularity produced by the Louvain algorithm [3]). This trace involves 4 communities of 11 nodes each, connected by 12 travellers having IDs from 1 to 12. Interests are asymmetrically distributed, that is, the 50% of interested nodes belong to the same community while the other communities involve only one destination each. This represent a difficult configuration because the first community actually is an attractor in which messages could remain trapped, preventing their complete dissemination. Fig. 6 shows the utilities for each node; nodes belonging to the same community are shown as dots with the same color. With the b*I*Cast utility, in this peculiar environment travellers have utilities lower than those of sedentary nodes in the same community as the interested nodes (red community). By contrast, with the w*I*Cast utility, travellers

**Table 1**
Main characteristics of the PMTR and the HCMM traces.

| Scenario | # Contacts | Length (s) | ICT (s) | Location mod. | Social mod. |
|---|---|---|---|---|---|
| PMTR | 11829 | 521.98 | 3381 | 0.536 | 0.419 |
| HD5 | 139714 | 11.88 | 1599 | – | 0.488 |
| HEXT | 196148 | 3.29 | 1097 | – | 0.613 |

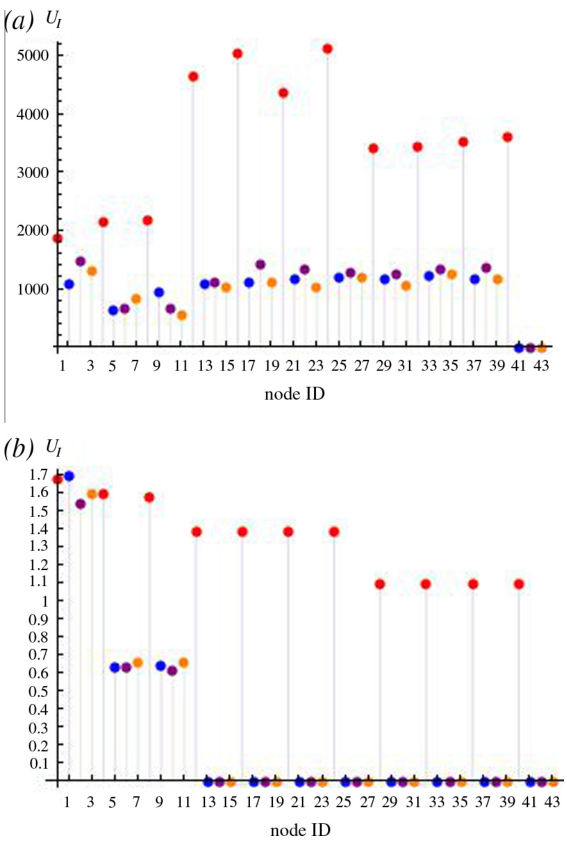


**Fig. 6.** Utility values for (a) b*I*Cast and (b) w*I*Cast in the HEXT trace.

are more useful than the sedentary nodes in other communities. As a consequence, a source in, e.g., the blue community can correctly relay a message for the red community to one of the travellers. □

## 6. Performance evaluation

In this section, *I*Cast is compared with both ProfileCast and SocialCast. We consider different scenarios satisfying the different assumptions of the algorithms and allowing a meaningful and fair comparison. To this aim, contact traces are needed, supplying: (i) mobility behavior of the nodes in terms of visited locations (thus satisfying the ProfileCast assumptions); (ii) real or realistic contacts characterizing social attitudes of people (for SocialCast); (iii) realistic interest distribution across nodes (for *I*Cast). In Section 6.1, we introduce the traces we adopted – satisfying the above requirements – and we analyze their characteristics. In Section 6.2, ProfileCast is compared with a simplified version of *I*Cast where the interest is a certain location and a content is addressed to the nodes accustomed to visit it. In Section 6.3, SocialCast is compared with *I*Cast by assigning the same interest to all nodes belonging to a certain encounter community.

Simulations recreated a variety of conditions. In all cases, without loss of generality, we assumed that only one interest exists in the system. Message generation starts after 12 h, so as to allow the algorithms to initialize their utility values. We assume that every hour, each source generates a message labeled with the interest. Content generation stops before the end of the traces (after 7 days) so as to allow the dissemination of the last generated messages. Nodes have infinite buffers, there is no bandwidth constraint, and messages have a lifetime longer than the time needed to deliver them, thus removing possible biases in the comparison. The performance indexes analyzed are: *coverage* (percentage of recipients in *I* that deliver the message) – which estimates the algorithms *Effectiveness* – mean number of hops to reach a recipient, mean latency to a recipient, number of nodes involved as relays in the forwarding of a message. All indexes are averaged over all recipients in *I* and all sources. The mean number of hops and the number of involved forwarders are an indirect measure of the algorithms *Efficiency*.

### 6.1. Scenarios

We used a real contact trace and a synthetic trace. For the former, we use the PMTR trace [11] (available in the CRAWDAD repository). This is a trace recorded in our campus, with fine granularity in terms of both time (1 s. beaconing frequency) and space (10 m. radio range), and long duration. Since we conducted the experiment, we know what nodes were located in fixed locations, and what are both the social and the interest groups, which allows us to build scenarios satisfying the requirements above. To the best of our knowledge, there are no other real contact traces publicly available, supplying the same characteristics. The synthetic trace (HD5) supplies an artificial (and thus well known and manageable) yet realistic environment. It has been produced with HCMM [4]. Both traces involve 44 nodes, moving over a $1000 \times 1000\ m^2$ area at pedestrian speed (0.5–2 m/s). The traces duration spans 13 working days, at 12 h per day (8:00 AM to 8:00 PM). In Table 1, the characteristics of the two traces are reported. The PMTR trace shows sedentary nodes with few long contacts, and high inter-contact time (ICT). Though, over the observed time window, all nodes run into each other. By contrast, the HD5 trace describes closed communities connected

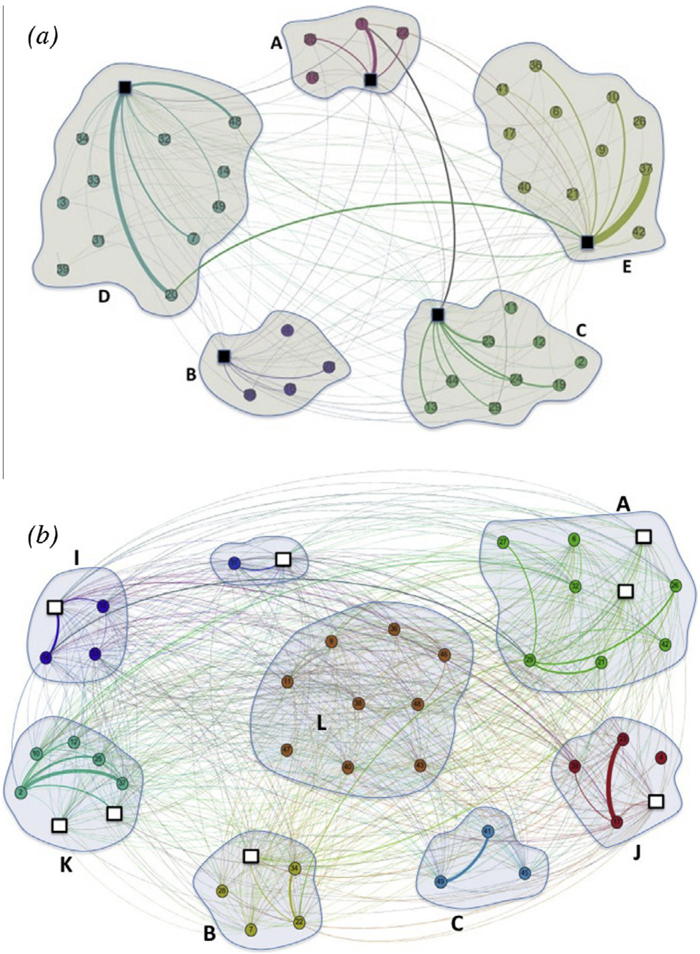


**Fig. 7.** (a) Location-based and (b) encounter-based communities in the PMTR trace.

by 5 traveller nodes. Contacts are more frequent and shorter than in the PMTR trace.

From those traces, we built scenarios with nodes grouped according to their habits in terms of either visited locations, or encounter habits; the former has been possible just with the PMTR trace, where we know the nodes placed in fixed locations. To this purpose, the Louvain algorithm [3] has been used, as it is considered one of the best in the literature [10], it avoids grouping nodes in a giant community, and it achieves greater modularity than other algorithms in the literature. To extract location-centered communities, a weighted graph $G$ is obtained from the contact trace, so that an edge exists between two nodes if one of them is a fixed location, and the other has ever met that location at least once. The edge weight is the mean duration of the contacts between the node and the fixed location. For encounter-based communities, we run the Louvain algorithm on both the PMTR and the HD5 trace, this time using the reciprocal of the ICT as edge weight. The Louvain algorithm is supposed to detect significant communities when the modularity is greater than 0.4. The last two columns of Table 1 report the modularities obtained in all cases, while Fig. 7 shows both PMTR location-based and encounter-based communities. The edge thickness is proportional to the weight. In Fig. 7(a), the fixed nodes are represented as black squares. White squares in Fig. 7(b) represent a social and interest community of students in the same class year; its orthogonality to the encounter groups enforces the evidence that encounters and interests are not necessarily correlated. With the HD5 trace, the Louvain algorithm produces 7 encounter communities of size ranging between 3 and 12 nodes.

### 6.2. Location-based BehaviorCast

In this subsection, we compare *I*Cast with ProfileCast, assuming that the interest is described in terms of customarily visited locations. The five scenarios in Fig. 7(a) are used.

#### 6.2.1. Analysis of the scenarios

We start determining the best parameter settings for ProfileCast in the considered scenarios. For each scenario, we computed the similarity either between nodes belonging to the considered location-based community (*simIN*),

**Table 2**
ProfileCast – similarity between nodes in and out of community.

| $\theta_W = \mathbf{0.9}$ | scenA | scenB | scenC | scenD | scenE |
|---|---|---|---|---|---|
| # nodes | 5 | 5 | 10 | 12 | 12 |
| *simIN* | 0.2123 | 0.0197 | 0.2715 | 0.2537 | 0.2360 |
| *simOUT* | 0.2071 | 0.1211 | 0.1519 | 0.1703 | 0.1618 |
| $\theta_{PC}$ | *0.21* | *0.01* | *0.16* | *0.175* | *0.165* |

or between nodes belonging to that community and nodes outside of it (*simOUT*). The time slot used to build association vectors is 12 h. Results are shown in Table 2, for $\theta_W = 0.9$; mind that $\theta_W$ determines the degree of summarization of the nodes' behavior. The shown values are averaged over all possible pairs of nodes and are computed at the end of the experiment, that is, after 13 association vectors were collected. We also considered $\theta_W = 0.7$ and $\theta_W = 0.8$: in all cases, variations are negligible, in the order of 0.6% on average. We decided to use $\theta_W = 0.9$ in the rest of the paper.

As it can be observed, for all, but one, scenarios *simIN* is greater than *simOUT*, thus correctly individuating communities of nodes with similar mobility behavior. These evaluations directly impact on the setting of the similarity threshold ($\theta_{PC}$) and can have heavy influence on the algorithm performances and efficiency, (see, Section 6.2.2). The setting of the value of $\theta_{PC}$ might depend on the specific location-based community to be addressed, and also on the day we are considering. This makes the choice of the appropriate setting very tricky because nodes have no a priori knowledge of the overall system behavior.

### 6.2.2. Sensitivity of ProfileCast to $\theta_{PC}$

In this section, we analyze how the choice of $\theta_{PC}$ impacts on the performance of ProfileCast. Coherently with the setting proposed in [15] we consider, for the different scenarios, $\theta_{PC} = 0.5$ and we compare the influence of this setting against a new set of values obtained from the evaluation of *simIN* and *simOUT*, as reported in Section 6.2.1. In fact, the proper value for $\theta_{PC}$ should be $simIN > \theta_{PC} > simOUT$, in order to try to maximize coverage for the considered target community, while avoiding bothering uninterested nodes. The incorrect delivery of a message to an uninterested node causes a waste of node's resources in terms of memory, processing, energy and radio channel. The percentage of uninterested nodes in the system receiving the message is an indirect global estimate of such inefficiency. In the following measures, sources are all and only the nodes in the target community.

In Fig. 8, the behavior of the coverage is shown for two different scenarios and different values of $\theta_{PC}$. When the ratio between *simIN* and *simOUT* is greater than 1 (Fig. 8(a)), the coverage is very stable also for values of $\theta_{PC}$ not perfectly fitting with the criteria above, that is, either slightly greater than *simIN* or lower than *simOUT*. Nonetheless, with a higher value (namely $\theta_{PC} = 0.5$), the coverage drastically drops. Scenarios scenC, scenD and scenE behave similarly to scenA. If, by contrast, similarities are in the reverse relationship (Fig. 8(b)), $\theta_{PC}$ must be low enough to guarantee that the interested nodes are reached. The

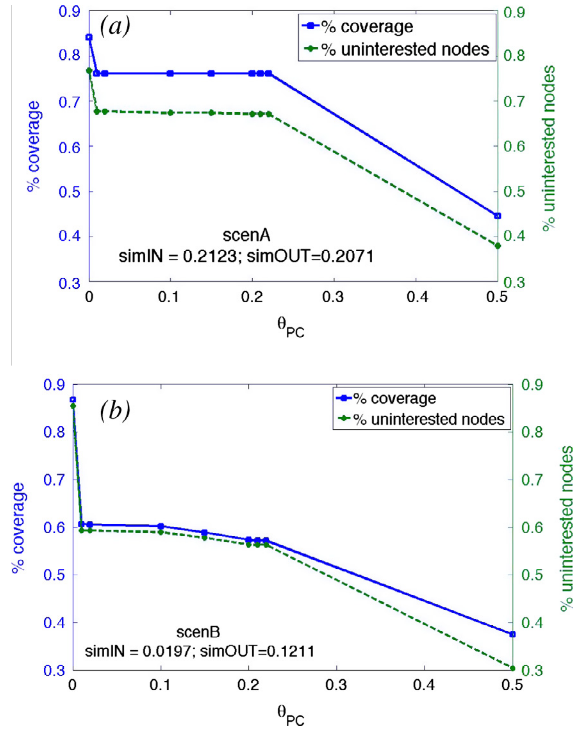


**Fig. 8.** Mean coverage and percentage of uninterested nodes receiving the message, for different values of $\theta_{PC}$ in (a) scenA and (b) scenB.

effects of this choice affect resource saving: the number of nodes not belonging to the target community and receiving the message in spite of this behaves exactly the same as coverage. In scenB, a high coverage (> 80%) is achieved when no threshold is set, thus resorting to a broadcast. In this case, 33 out of the 39 uninterested nodes – the 84% – receive the message.

In general, the value $\theta_{PC} = 0.5$ proposed in [15] turned out to be unsuitable to perform adequately in the PMTR scenarios. A viable policy to adapt its value to unpredictable conditions seems to be hardly identified. This inability might be paid with high waste of communication, energy and processing resource.

### 6.2.3. ProfileCast and ICast with location-based communities

Last line of Table 2 shows the value of $\theta_{PC}$ we adopted for comparison between ProfileCast and *I*Cast. The values are chosen in order to maximize coverage while limiting the useless message exchanges. For all the scenarios with $simIN > simOUT$ this is coincident with a value equal to $simOUT + \varepsilon$ for small $\varepsilon$.

The performances of the three algorithms differ in terms of both coverage (Fig. 9(a)) and resource waste (Table 3). For all scenarios, ProfileCast is unable to reach the same coverage as both b*I*Cast and w*I*Cast, while at the same time it delivers the messages to a ratio of uninterested nodes higher than the relays used by both *I*Cast. In order to correctly interpret the results in Table 3 – averaged over all scenarios – it is worth to point out that the number of hops performed by a message is measured

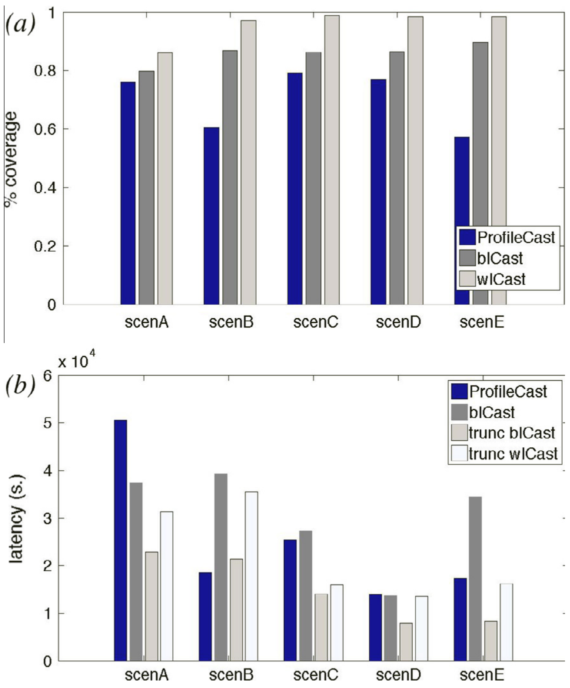


**Fig. 9.** Comparison between ProfileCast and InterestCast in terms of (a) obtained coverage, and (b) latency.

**Table 3**
Location-based communities – resource usage for different algorithms.

| | Uninterested nodes (%) | Gain (%) | Transmissions/ node | Gain (%) | # Hop |
|---|---|---|---|---|---|
| ProfileCast | 62.6 | | 1.003 | | 3.07 |
| b*I*Cast | 47.4 | 24.3 | 1.105 | −10.1 | 2.91 |
| w*I*Cast | 35.1 | 43.8 | 0.764 | 23.9 | 4.73 |

when a copy of it is received at a destination. Yet, content diffusion occurs along a sort of a tree structure springing from the source; the number of nodes not in *I* receiving a message, and the number of transmissions performed by nodes, are counted for all tree branches, even for those that do not lead to a destination. Furthermore, in the case of *I*Cast, uninterested nodes are involved in the forwarding process not because the message is erroneously delivered to them – as in ProfileCast – but because they are useful bridges to reach the target community. w*I*Cast shows the best performance in terms of both coverage and resource saving at nodes: just 35% of the nodes not in *I* are involved in the message forwarding (that is, around 12 nodes), thus achieving a gain of almost 44% with respect to ProfileCast in terms of memory usage at uninterested nodes. As far as energy saving is concerned, while nodes perform around 1 message transmission with ProfileCast, b*I*Cast is a bit more exigent, while w*I*Cast achieves a gain of almost 24% in battery usage. This is due to a more clever choice of the relays, due to the adopted utility function that magnifies the differences amongst candidate forwarders. In this perspective, the mean number of hops can be correctly interpreted: w*I*Cast takes paths straight to the destinations, by just involving the nodes needed to this aim, and succeeds in reaching also destinations "difficult" for other approaches.

In Fig. 9(b), we show the latency to reach the maximum coverage of recipients for the two protocols. The latency obtained by b*I*Cast is apparently higher than that of ProfileCast, but this is due to the fact that it is evaluated on the reached destinations, and b*I*Cast obtains a higher coverage. For a fair comparison, we evaluated the b*I*Cast and w*I*Cast latency by truncating the message diffusion when the same coverage as ProfileCast is reached. We can observe that b*I*Cast takes approximately the same or less time to reach the same number of destinations as ProfileCast. We conjecture that for ProfileCast, the involvement of uninterested nodes – which become on their behalf message forwarders to nodes similar to *themselves* (rather than to the source) – introduces a factor of "deviation" from the path to the targeted recipients. Truncated w*I*Cast spends a time slightly greater than that of truncated b*I*Cast. This is particularly evident in case of scenB: mind that scenB has a reversed ratio between *simIN* and *simOUT*. In spite of being an unfavorable setting for ProfileCast, w*I*Cast reaches almost all destinations; yet, waiting useful relays to them requires more time. In summary, under the considered scenarios, ProfileCast reveals both less effective and less efficient than the two *I*Cast algorithms.

### 6.3. Social-based BehaviorCast

In this section, we compare *I*Cast and SocialCast, when the interested nodes belong to the same encounter

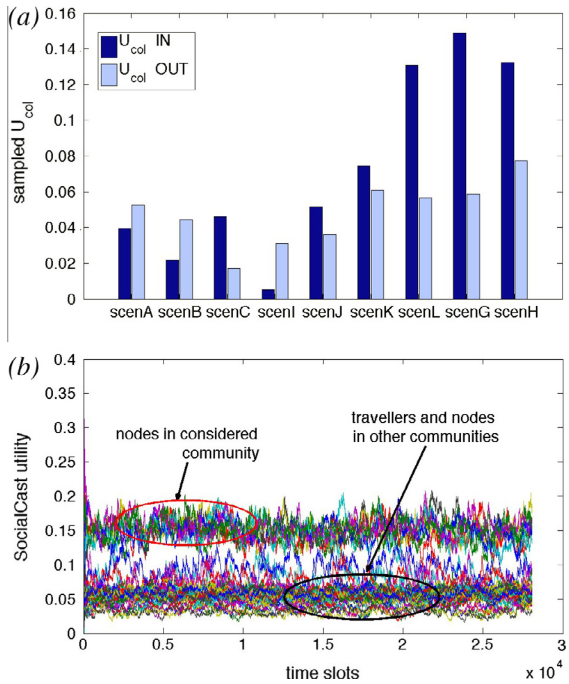


**Fig. 10.** (a) $U_{col}$ values in different scenarios for the PMTR trace and (b) utilities $\mathcal{U}$ for all nodes in SocialCast (Eq. (2)).

**Table 4**
SocialCast – reference scenarios.

| PMTR | | | | HD5 | | | |
|---|---|---|---|---|---|---|---|
| Scenario | Size | $U_{col}$ IN | $U_{col}$ OUT | Scenario | Size | $U_{col}$ IN | $U_{col}$ OUT |
| scenC | 3 | 0.046 | 0.017 | scenA | 3 | 0.111 | 0.002 |
| scenJ | 5 | 0.0517 | 0.0358 | scenC | 5 | 0.027 | 0.008 |
| scenL | 9 | 0.1308 | 0.0567 | scenG | 9 | 0.071 | 0.021 |
| scenG | 8 | 0.1489 | 0.0586 | | | | |
| scenH | 11 | 0.1324 | 0.0774 | | | | |

community. As communities, we use a subset of those emerging from the clusterization described in Section 6.1; they are shown in Fig. 7(b) for PMTR. Besides, in PMTR we considered two scenarios scenG and scenH that are two real communities of users bound by common interests: a group of students in the same class year, and the group of researchers in Computer Networks respectively. Both are distributed over six encounter-based communities; scenG is represented in Fig. 7(b) by white squares. As an initial step, we analyze the sensitivity of SocialCast to the parameter setting, and we determine good values for the parameters of the Kalman filters in the considered scenarios. Throughout this section, some parameters are however fixed. Namely, the period $\tau$ for state sampling is set equal to the period $T$ for interest dissemination, whose value is 20 s. as in [7]. The weights associated to $\widehat{U}_{col}$ and $\widehat{U}_{cdc}$ in Eq. (2) are $w_{col} = 0.75$ and $w_{cdc} = 0.25$ as in [7]. SocialCast allows senders to generate $\gamma$ copies of a content; for the sake of comparison, we set $\gamma = 1$.

### 6.3.1. Analysis of the scenarios

For every scenario, the median value of both $U_{cdc}$ and $U_{col}$ was 0 for all nodes for both PMTR and HD5 traces. A median 0 for $U_{cdc}$ means that – for more than half of the samples – in two consecutive samples a node has no neighbors; a median 0 for $U_{col}$ means that a node has no neighbors in $I$ for more than half of the samples. On the other hand, there could be encounters or changes occurring between two consecutive samples that pass completely disregarded.

In Fig. 10(a), $U_{col}$ is reported for the different scenarios of the PMTR trace: either averaged for all nodes in the community of interested users ($U_{col}$ IN), or for the nodes outside that community ($U_{col}$ OUT). Scenarios scenA, scenB and scenI have a value of $U_{col}$ greater for nodes outside $I$ than for the members of $I$. We decided to discard those scenarios, as they violate the SocialCast assumption. This also occurs with the HD5 trace. The scenarios selected for the simulations, and their characteristics, are summarized in Table 4. In Fig. 10(b), we show the behavior of the SocialCast utility function for all nodes in an HD5 encounter-based community of 12 nodes. Predictions have been obtained by setting the measurement noise covariance $R$ to 1 and the process noise covariance $Q$ to $10^{-5}$ for both Kalman filters. This plot explains the performance of SocialCast discussed in Section 6.3.2. It is evident that utilities are not well differentiated. For a source not belonging to the community where destinations reside, passing the message to travellers is really difficult. Indeed, travellers move around and encounter the destinations far less than the sedentary nodes belonging to the target community, which have $U_{col}$ 10 times greater than the other nodes. And, travellers have a lower $U_{cdc}$, as they may not encounter any node while moving between communities, while sedentary nodes continuously run into other nodes in the same community and their neighborhood changes. In fact, throughout the HD5 trace, travellers have 2213 encounters on average, while sedentary nodes observe 6881 encounters on average. That is, SocialCast fails in characterizing weak ties, as utility values are very similar.[4] Results with the PMTR trace are more fuzzy but provide the same indications.

Fig. 11 motivates our choice for $R$ and $Q$: we compare the $U_{cdc}$ samples and the predictions $\widehat{U}_{cdc}$ for a node with the PMTR trace and two different settings of the covariances. Similar results are achieved also for the HD5 trace, and for $U_{col}$ and $\widehat{U}_{col}$ in all scenarios. A high value of $Q$ and a low value of $R$ produces predictions more noisy and quickly changing according to the samples. Yet, as many sampling are done when nodes have no neighbors, this setting risks to yield many predictions equal to 0. As a consequence, no discrimination would be possible between relays and SocialCast might degrade to a Direct Contact policy. In this work, we thus adopt $R = 1$ and $Q = 0.00001$. Some experiments with the reverse setting show that it dramatically affects performance in terms of low coverage and high latency, because nodes are less able to choose appropriate relays.

We also analyzed the impact of the hysteresis $\varepsilon$ used to decide whether to forward a message (Section 3.2). With an hysteresis of 0.01, coverage decreases of 1–1.5%, but latency considerably increases up to 70% because several forwardings are prevented due to very similar utilities. Therefore, in the sequel we use $\varepsilon = 0$.

### 6.3.2. SocialCast and ICast with encounter-based communities

In the following measures, all nodes in the system act as source of messages to the members of the considered scenario; every source generates one message per hour. In Fig. 12, we compare the performance of SocialCast, b*I*Cast and w*I*Cast in the PMTR scenarios. According to the considerations above, SocialCast is less effective than both flavors of *I*Cast in reaching the destinations as it takes inappropriate paths, particularly when the communities of the source and the destination differ. In fact, sources in the

[4] Mind that both $U_{cdc}$ and $U_{col}$ are in the range $[0,1]$. In particular, $U_{col}$ only provides a qualitative indication of familiarity with target nodes, instead of quantitative, i.e., it does not discriminate between nodes encountering either one, or several destinations. Recall the argumentation of Section 5 concerning w*I*Cast.

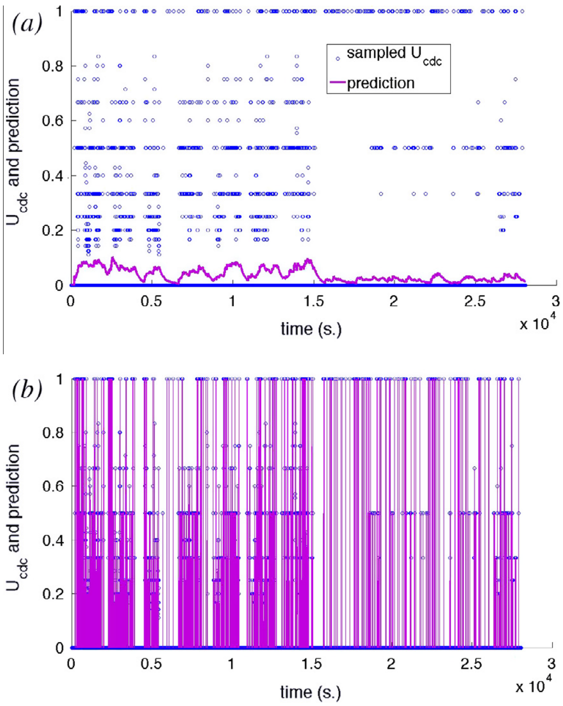


**Fig. 11.** Comparison between $U_{cdc}$ and $\hat{U}_{cdc}$ with (a) $R = 1$ and $Q = 0.00001$, or (b) $R = 0.00001$ and $Q = 1$.

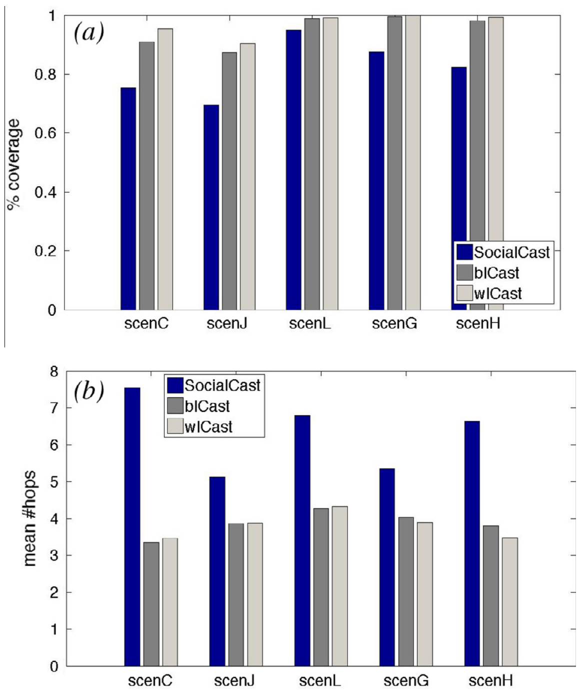


**Fig. 12.** (a) Coverage and (b) mean number of hops to destinations with the PMTR trace.

community of *I* achieve a coverage 15% higher than external sources, on average. This drawback of SocialCast yields longer paths (Fig. 12(b)). In comparison to the truncated versions of both b*I*Cast and w*I*Cast, latency is 10% higher; this is also due to the fact that in most of the interest dissemination phases there is no interested neighbor or no neighbors at all. The number of existing copies of the content is roughly 37% less than b*I*Cast and w*I*Cast. Though, this does not indicate a clever resource usage of SocialCast; rather, it confirms our consideration: content is replicated just when either a destination is reached, or a reached destination is also a relay and it encounters a more useful node (Fig. 4). The low number of copies confirms that SocialCast forwards the message along a series of relays without encountering any destination.

The closed communities of HD5 even better emphasize this behavior (Fig. 13). The SocialCast mean coverage is around 64% of the destinations, while both b*I*Cast and w*I*Cast obtain a coverage 45% higher, that is, 93% of the destinations. Sources within the target community show a coverage 22% higher than external sources. A long time is spent before selecting a suitable next relay and paths are tortuous.

Finally, we validate through simulations the property of *Eventual Termination*. Fig. 14 shows both the number of transmissions and the cumulative number of reached recipients over time for b*I*Cast, with PMTR. The former index is measured as the number of forwarding operations in "slices" of 2000 s. of simulation: every 2000 s. we count the number of message forwarding operations occurred from the end of the previous sample. In Fig. 14, results concern scenL with a source external to the social communities to which the recipients belong. Interestingly, the algorithm shows a sort of autonomic ability to self-stabilize; in fact, the number of transmissions decreases as the number of reached recipients increases. This is motivated by

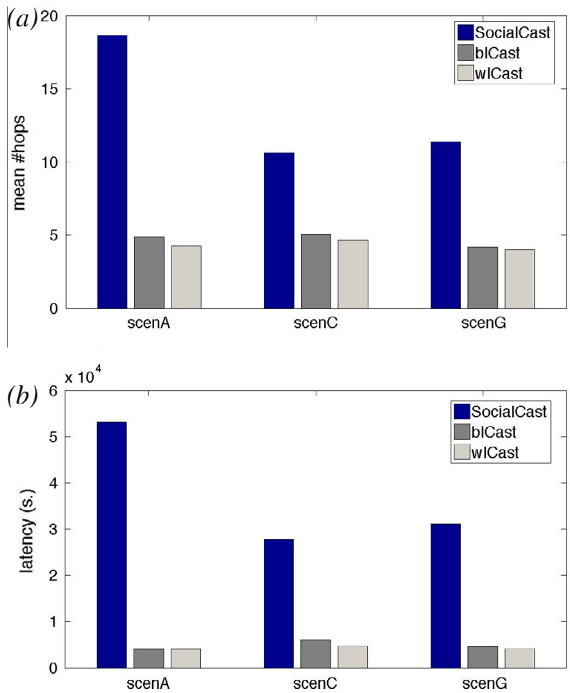


**Fig. 13.** (a) Mean number of hops and (b) latency to destinations with the HD5 trace.

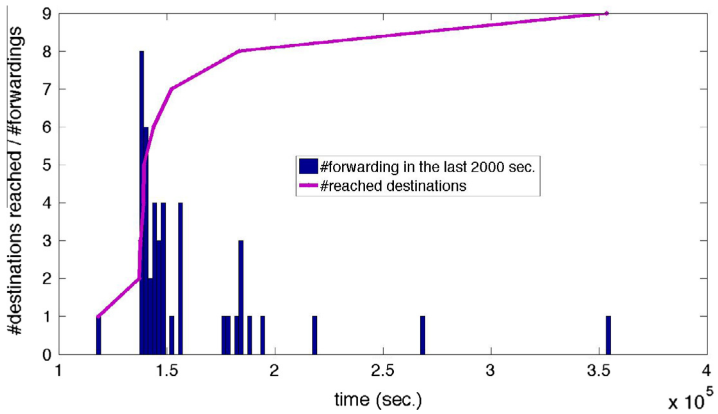


**Fig. 14.** Performance in terms of transmissions and reached recipients, for scenL of PMTR, and an external source.

observing that messages are forwarded from lower to higher utility nodes and are then removed from the buffers of forwarders; this gradually leads to confining message copies only in the buffers of nodes with high utility for $\mathcal{I}$, which have reduced forwarding opportunities. It is worth to notice that communication gradually turns off *before* the message lifetime is reached, that is, it settles down when all interested users are reached.

## 7. Conclusion

In this work, we introduce the *BehaviorCast* communication paradigm, in which the binding content-recipients is not provided by the sender (in the classical IP-addressing style), but directly executed by specific recipients with an *interest* in a tagged content. We propose the *I*Cast algorithms to solve the problem. *I*Cast provides a new form of content-driven addressing and has been explicitly devised to operate over Opportunistic Networks of individuals. *I*Cast uses utility functions expressly designed to find weak ties allowing to connect different communities and chase target destinations wherever they are, independently from locations or social communities, thus providing a very general solution to the problem. We show by means of both analysis and simulations the remarkable performance of the proposed algorithms in comparison with similar approaches presented in the literature. In particular, the experiments show the superior capability of *I*Cast of appropriately capturing human behaviors and of adequately exploiting them for content diffusion.

## Acknowledgments

This work was funded by the EC under the EINS (FP7-FIRE 288021), MOTO (FP7 317959), and EIT ICT Labs MOSES (Business Plan 2014) Projects.

## References

[1] S.M. Allen, M.J. Chorley, G. Colombo, E. Jaho, M. Karaliopoulos, I. Stavrakakis, R.M. Whitaker, Exploiting user interest similarity and social links for micro-blog forwarding in mobile opportunistic networks, Pervas. Mobile Comput. 11 (2014) 106–131.
[2] A. Bassoli, J. Brewer, K. Martin, I. Carreras, D. Tacconi, Undersound and the above ground, in: 5th Int'l Mobile Music Workshop (MMW), 2008.
[3] V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech.: Theory Exp. (10) (2008).
[4] C. Boldrini, M. Conti, A. Passarella, Users mobility models for opportunistic networks: the role of physical locations, in: Proc. IEEEE WRECOM, 2007, pp. 1–6.
[5] M. Chuah, Y. Xi, An encounter-based multicast scheme for disruption tolerant networks, Comput. Commun. 32 (16) (2009) 1742–1756.
[6] R.I. Ciobanu, R.-C. Marin, C. Dobre, V. Cristea, C.X. Mavromoustakis, ONSIDE: socially-aware and interest-based dissemination in opportunistic networks, in: Proc. IEEE NOMS, 2014, pp. 1–6.
[7] P. Costa, C. Mascolo, M. Musolesi, G.P. Picco, Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks, IEEE J. Selected Areas Commun. 26 (5) (2008) 748–760.
[8] D. Easley, J. Kleinberg, Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press, 2010.
[9] P. Eugster, P. Felber, R. Guerraoui, A.-M. Kermarrec, The many faces of publish/subscribe, ACM Comput. Surv. 35 (2) (2003) 114–131.
[10] S. Fortunato, Community detection in graphs, Phys. Rep. 486 (2010) 75–174.
[11] S. Gaito, E. Pagani, G.P. Rossi, Strangers help friends to communicate in opportunistic networks, Comput. Networks J. 55 (2) (2011) 374–385.
[12] W. Gao, Q. Li, B. Zhao, G. Cao, Multicasting in delay tolerant networks: a social network perspective, in: Proc. 10th ACM International Symposium on Mobile Ad hoc Networking and Computing (MobiHoc), 2009, pp. 299–308.
[13] B. Han, P. Hui, V.A. Kumar, M.V. Marathe, G. Pei, A. Srinivasan, Cellular traffic offloading through opportunistic communications: a case study, in: Proc. CHANTS, 2010.
[14] T. Hossmann, T. Spyropoulos, F. Legendre, Know thy neighbor: towards optimal mapping of contacts to social graphs for DTN routing, in: Proc. IEEE INFOCOM, 2010.
[15] W.-J. Hsu, D. Dutta, A. Helmy, Profile-cast: behavior-aware mobile networking, SIGMOBILE Mobile Comput. Commun. Rev. 12 (1) (2008) 52–54.
[16] W.-J. Hsu, D. Dutta, A. Helmy, CSI: a paradigm for behavior-oriented profile-cast services in mobile networks, Ad Hoc Networks 10 (8) (2012) 1586–1602.
[17] S. Ioannidis, A. Chaintreau, On the strength of weak ties in mobile social networks, in: Proc. 2nd ACM EuroSys Workshop on Social Network Systems (SNS), 2009, pp. 19–25.
[18] E. Koutsogiannis, L. Mamatas, I. Psaras, Storage-enabled access points for improved mobile performance: an evaluation study, in: Proc. 9th Int'l Conf. on Wired/Wireless Internet Communications (WWIC), 2011.
[19] U. Lee, S.Y. Oh, K.-W. Lee, M. Gerla, RelayCast: scalable multicast routing in delay tolerant networks, in: Proc. ICNP, 2008.
[20] G. Lo Giusto, A.J. Mashhadi, L. Capra, Folksonomy-based reasoning for content dissemination in mobile settings, in: Proc. ACM CHANTS, 2010, pp. 39–46.

[21] A.J. Mashhadi, S.B. Mokhtar, L. Capra, Habit: leveraging human mobility and social network for efficient content dissemination in Delay Tolerant Networks, in: Proc. IEEE WoWMoM, 2009, pp. 1–6.
[22] A. Mei, G. Morabito, P. Santi, J. Stefa, Social-aware stateless forwarding in pocket switched networks, in: Proc. IEEE INFOCOM, 2011, pp. 251–255.
[23] S. Moghaddam, M. Carvalho, A. Helmy, Domain and location specific modeling of mobile users online interests, in: Proc. IWCMC, 2013, pp. 1436–1441.
[24] A. Moghadam, H. Schulzrinne, Interest-aware content distribution protocol for mobile disruption-tolerant networks, in: Proc. IEEE WoWMoM, 2009, pp. 1–7.
[25] A. Mtibaa, M. May, M. Ammar, On the relevance of social information to opportunistic forwarding, in: Int'l Symp. Modeling, Analysis and Simulation of Computer Systems, 2010, pp. 141–150.
[26] A. Mtibaa, M. May, C. Diot, M. Ammar, PeopleRank: social opportunistic forwarding, in: Proc. IEEE INFOCOM, 2010, pp. 1–5.
[27] E. Pagani, G.P. Rossi, Content dissemination on location-based communities: a comparative analysis, in: Proc. 6th IEEE Mobilware, 2013.
[28] E. Pagani, G.P. Rossi, Interest-driven forwarding for delay-tolerant mobile ad hoc networks, in: Proc. 9th IEEE IWCMC Conference, 2013, pp. 718–723.
[29] T.-M. Pham, S. Fdida, DTN support for news dissemination in an urban area, Comput. Networks J. 56 (2012) 2276–2291.
[30] F. Pianese, X. An, F. Kawsar, H. Ishizuka, Discovering and predicting user routines by differential analysis of social network traces, in: Proc. 14th IEEE WoWMoM, 2013.
[31] A.-K. Pietiläinen, E. Oliver, J. LeBrun, G. Varghese, C. Diot, MobiClique: middleware for mobile social networking, in: Proc. ACM WOSN, 2009, pp. 49–54.
[32] A. Socievole, E. Yoneki, F. De Rango, J. Crowcroft, Opportunistic message routing using multi-layer social networks, in: Proc. 2nd ACM HP-MOSys, 2013, pp. 39–46.
[33] A. Vahdat, D. Becker, Epidemic Routing for Partially Connected Ad Hoc Networks. Technical Report CS-200006, Duke University, April 2000.
[34] J. Whitbeck, M. Dias de Amorim, Y. Lopez, J. Leguay, V. Conan, Relieving the wireless infrastructure: when opportunistic networks meet guaranteed delays, in: Proc. IEEE WOWMOM, 2011, pp. 1–10.
[35] J. Wu, Y. Wang, A non-replication multicasting scheme in delay tolerant networks, in: Proc. 7th IEEE MASS Conference, 2010, pp. 89–98.
[36] P. Yang, M.C. Chuah, Context-aware multicast routing scheme for disruption tolerant networks, Int. J. Ad Hoc Ubiquitous Comput. 4 (5) (2009) 269–281.
[37] W. Zhao, M. Ammar, E. Zegura, Multicasting in delay tolerant networks: semantic models and routing algorithms, in: Proc. ACM SIGCOMM Workshop on Delay-tolerant networking (WDTN), 2005, pp. 268–275.

**Elena Pagani** is Associate Professor at the Computer Science Department of the Università degli Studi di Milano, and Associate Researcher at the Institute for Informatics and Telematics of the National Research Council in Pisa. Her current research interests include network protocols and architectures, wireless technologies, MANETs and delay tolerant networks, and performance evaluation. She served as Web Chair for INFOCOM 2013, and is an Area Editor for the Elsevier Computer Communications Journal.

**Lorenzo Valerio** received a Ph.D. degree with the Department of Mathematics, University of Milano, Milano, Italy. Currently, he is Researcher at the Institute for Informatics and Telematics of the National Research Council, Pisa, Italy. His current research interests include the inference of spatial and temporal processes, the connection between mobility, and computational learning models, and finally mobile technologies such as mobile networks and mobile operating systems.

**Gian Paolo Rossi** is Full Professor of Computer Networks at the Computer Science Department, Universita' di Milano, where he leads the Computer Networks Laboratory. He spent 2 postdoc years at the Joint Research Center of the EC in Ispra. He joined the University of Milano in 1980 as an Assistant Professor. From 2000 to 2006, he chaired the Faculty track in Computer Science. His research interests include architecture and protocol design, performance evaluations, modeling and analysis. He served as TPC member in the most important international conferences. In past years, he had part or coordinated national and international research projects.