Pervasive and Mobile Computing 16 (2015) 115–135

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

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

Fast track article

Scalable data dissemination in opportunistic networks through cognitive methods☆

Lorenzo Valerio [a,*], Andrea Passarella [a], Marco Conti [a], Elena Pagani [a,b]

[a] IIT-CNR, Pisa, Italy

[b] Computer Science Department, University of Milan, Italy

ARTICLE INFO

Article history:
Available online 28 May 2014

Keywords:
Opportunistic networks
Cognitive heuristics
Content diffusion

ABSTRACT

The Future Internet scenario will be characterised by a very large amount of information circulating in large scale content-centric networks. One primary concern is clearly to replicate and disseminate content efficiently, such that – ideally – it is replicated and spread only in those portions of the network where there are interested users. As centralised data dissemination solutions are unlikely to be feasible due to the sheer amount of content expected to circulate, nodes themselves must locally take data dissemination decisions, taking into account contextual information about users interests. In this paper, we consider a mobile opportunistic networking environment where mobile nodes exploit contacts among each other to replicate and disseminate content without central control. In this environment, we see nodes as proxies of their human users in the cyber world made up by mobile devices. Accordingly, we want nodes to act as much as possible as their users would do if they had to disseminate information among each other. We thus propose a new solution based on cognitive heuristics. Cognitive heuristics are *functional* models of the human mental processes, studied in the cognitive psychology field. They describe the judgement process the brain performs when subject to temporal constraints or partial information. We illustrate how these cognitive processes can be fruitfully implemented into a feasible and working ICT solution, in which decisions about the dissemination process are based on aggregated information built up from observations of the encountered nodes and successively exploited through a stochastic mechanism to decide what content to replicate. These two features allow the proposed solution to drastically limit the state kept by each node, and to dynamically adapt to the dynamics of content diffusion, the dynamically changing node interests and the presence of churning of nodes participation to the data dissemination process. The performance of our solution is evaluated through simulations and compared with reference solutions in the literature.

1. Introduction

Among other features, mobility and content-based approaches are two key characteristics of reference Future Internet scenarios [1]. This means that people equipped with mobile devices will play a central role in the whole information

☆ This work is funded partially by the EC under the FET-AWARENESS RECOGNITION Project, grant 257756, FIRE EINS (FP7-288021).

* Corresponding author. Tel.: +39 50 315 3059.

E-mail addresses: lorenzo.valerio@iit.cnr.it, lorenzo.valerio@gmail.com (L. Valerio), andrea.passarella@iit.cnr.it (A. Passarella), marco.conti@iit.cnr.it (M. Conti), pagani@di.unimi.it (E. Pagani).

http://dx.doi.org/10.1016/j.pmcj.2014.05.005
1574-1192/

environment [2]. Indeed, people's activities – through their mobile devices – will strongly contribute to the process of data and content creation, by generating huge information flows that, together with the already existing complex information environment, needs to be properly managed and diffused. This is likely to push to the limit existing solutions or conventional approaches—for example, it is foreseen that even 4G cellular networks will not be able to cope with the huge data traffic demand of the users in the coming years [3]. Therefore, approaches based on exploiting direct contacts between user devices in the dissemination of content become very interesting. On the other hand, it is reasonable to assume that a very significant part of this content will be very contextualised, i.e. relevant only at specific times and/or geographic areas, often interesting for specific groups of users only. Therefore, exploiting information like human behaviour, mobility patterns, social habits and other similar information, users' mobile devices can contribute to the distribution of content in an efficient way, i.e. without flooding the network with information that is irrelevant for a large fraction of the users.

In this scenario, Opportunistic Networking [4] schemes represent a viable and natural way to efficiently disseminate contents among interested users. Opportunistic networks (OppNets) are self-organising mobile networks where the existence of simultaneous end-to-end paths between nodes is not taken for granted, while disconnections and network partitions are the rule. Opportunistic networks support multi-hop communication by temporarily storing messages at intermediate nodes, until the network reconfigures and better relays (towards the final destinations) become available. However, the management of the huge amount of information circulating in the environment can easily become a problem when resources available at mobile nodes (e.g., battery and memory) are taken into consideration [5,6]. Thus, mobile devices must adopt schemes able to cope with this complex environment, allowing them to discern what information is really important and interesting to be disseminated. This scenario motivates the research of lightweight distributed solutions that allow mobile devices to take autonomous decisions about what information they should disseminate, among the vast amount of data possibly available on the other encountered devices.

Here, we present a data dissemination approach based on a class of cognitive models called *cognitive heuristics*. Briefly, cognitive heuristics are algorithmic descriptions of the mental processes the brain uses to quickly take decisions in conditions of partial or incomplete knowledge. The capability of heuristics to work in a fast and frugal way makes them an interesting approach to be adopted in OppNets. Among the various cognitive heuristics, in this paper we consider in particular the recognition heuristic [7,8]. Briefly, it states that, when confronted between two possible alternatives, the brain selects the one that it "recognises". The behaviour of this heuristic can be explained through the following example: a person asked to indicate which university is more endowed without having any direct information about the real entity of endowments, will make his selection according to other indirect information like how often a university name comes to his attention. The more often he hears a university name the more likely he will indicate the recognised university name as more endowed.

In this work, we exploit the recognition heuristic for data dissemination in opportunistic networks. We assume a scenario characterised by the presence of content – hereafter referred to as data items – organised in specific topics – hereafter referred to as channels of interest – and nodes interested in some of those topics. Moreover, nodes act as both contents generators and data carriers, indeed, contacts between nodes are the only way to disseminate data items in the system. A key problem every node part of a data dissemination system for opportunistic network has to face is dynamically deciding when specific data items must be diffused more or less aggressively. In this paper we exploit the recognition heuristic to address both these aspects: (i) in order to decide whether diffusion has to be boosted for a certain item, nodes in our system recognise what items are of interest for several nodes; (ii) in order to decide whether an item is already sufficiently diffused, nodes in our system are able to recognise that it is already carried by (most of) the interested nodes. It is worth noting that this approach is not "yet another bio-inspired protocol". In our scenario, nodes are actual proxies of their human users in the cyber world. By using the same cognitive processes of their users, nodes behave very similar to how human counterparts would behave if facing the same problem in the physical world. The work presented in [9,6] is a preliminary attempt at investigating this approach (see Section 2 for more details). The main focus of [9,6] was to highlight that using the recognition heuristic is a viable option. In this paper, we turn this idea in the definition of a concrete system for opportunistic networks, by investigating how cognitive heuristics can be applied taking into consideration key restrictions of opportunistic networks, i.e. resource limitations and dynamic conditions.

In this paper (and in our previous work [10]), we exploit aggregate information for driving the behaviour of the recognition heuristic, that is, we investigate how the cognitive heuristics could be applied by starting from aggregate information about the dissemination state of data items only. This can be seen as the application of another cognitive mechanism aimed at maintaining only few essential information about the state of the surrounding environment and permits to limit the memory used by nodes to maintain the state information needed for the data dissemination policies. Moreover, our results show that this reduction comes without sacrificing (w.r.t. state of the art solutions) the performance in terms of delivering data items to interested users. Another key feature is represented by the introduction of a stochastic mechanism that drives the recognition process. This stochastic mechanism makes the system adaptive to dynamical conditions. In cognitive psychology these cognitive heuristics represent an algorithmic alternative w.r.t. another class of cognitive models mostly based on Bayesian probabilities. As we will explain in Section 4, our data dissemination solution, although based on cognitive heuristic, extends the classical heuristic model by introducing some parameters that we estimate exploiting the Bayesian framework. With respect to [10], in this paper we show that the recognition heuristic applied to data dissemination in opportunistic network can be seen as an instance of a Bayesian problem. In addition, we provide a much more extended set of performance results, showing that the proposed algorithm efficiently reacts to dynamic scenarios where at a certain time (i) nodes may change

their interests about channels, or (ii) completely new channels/items are injected in the running system and, finally, (iii) churning nodes are present in the network. Moreover, we supply performance results obtained with real contact traces.

2. Background and related work

2.1. Cognitive models

Cognitive psychology studies the way the human brain works and reacts to external stimuli. Here we consider models addressing two different but complementary points of view: on the one side there are Bayesian cognitive models and on the other side heuristic approaches [11]. Briefly, Bayesian cognitive models support the idea that the decision making process carried out by the human brain in condition of uncertainty or partial information can be well described through the classical Bayesian probabilistic framework. The basic idea is that people behave as *intuitive statisticians* [12], that is, their judgements closely correspond to the classical Bayesian statistical norms that explain how prior beliefs or hypothesis may be updated rationally in light of new information. Specifically, as will be described formally in Section 4, observations of a phenomenon are used to estimate the probabilities of future event occurrences. Whenever a new observation is available, these probabilities are updated (according to the Bayes rule) to incorporate this additional knowledge. Alternative models try to describe the decision making process as an *heuristic process*. Gigerenzer et al. [13] suggest that the information people perceive and use to take decision can be simply expressed as frequencies, i.e. how many events are favourable (according to some criterion) over the total number of events observed. The algorithmic procedures with which the brain elaborates these frequencies so as to come up with a final decision or judgement are called *cognitive heuristics*. Precisely, *cognitive heuristics* are functional models of the mental processes [7,8] on which the human brain relies to *quickly* take appropriate actions also in presence of incomplete knowledge of the situation. They do not aim at reproducing the detailed physiology of the brain's processes (as neural networks), but model their functionality. Therefore, heuristics can be seen as simple algorithmic models of the complex process used by the brain to quickly find a solution to a problem, when the exhaustive search of the optimal solution is impractical or infeasible due to the lack of complete information or temporal constraints. Cognitive heuristics have been applied in various fields, such as financial decision making [14], forecasting purchases [15], results of sport events [16], outcomes of political elections [17]. Usually, the solution supplied by heuristics well approximates the optimum.

Although they could appear almost opposite approaches, the Bayesian and heuristic points of view are compatible and very closely connected. While the probabilistic models and methods specify the nature of the cognitive problem to be solved and the information needed for solving it, heuristic models represent the set of algorithms and tricks through which human cognitive processes operate in order to solve a cognitive problem [18]. Thus, we can say that the heuristic modelling is an operational approximation of the more complex and complete Bayesian description of such cognitive processes. In this work we include both approaches in the definition of the algorithms used by nodes to take data dissemination decisions.

2.2. Content distribution in OppNets

Several solutions for data dissemination in opportunistic networks have been proposed in the literature. A survey can be found in [19]. Some papers consider the problem of content diffusion in mixed fixed/mobile networks. In [20], a hybrid infrastructure is considered where *throwboxes* – i.e. devices with both wired and wireless interfaces – communicate with one another and with the wireless nodes. Nodes upload held items when in communication range with a throwbox, and possibly download items that satisfy local interests. A similar hybrid infrastructure is considered in [21]. In both proposals, caches are maintained in the nodes belonging to the wired infrastructure with usual cache replacement algorithms.

Several papers deal with the problem of content distribution in pure OppNets. The PodNet project [22] considers a scenario similar to the one in this paper. Nodes may subscribe to channels of interest. Upon each encounter, nodes exchange items in order to retrieve those belonging to the subscribed channels. Then, other items may be exchanged and loaded in a *public cache* in order to facilitate their dissemination to interested nodes. The items to be maintained in the public cache are chosen depending on the channel popularity, but blindly to social aspects. With respect to our mechanisms, PodNet uses much simpler policies, such as replicating according to the popularity of the channels.

By contrast, some other works exploit social information associated to nodes in order to disseminate data in the network. In ContentPlace [23–25], nodes aim at filling their caches in order to maximise both the local utility (i.e. the interests of the local user) and the community utility. The latter forces nodes to carry items that the local user is not interested in, but that are of interest for the users belonging to the same social communities of the local user. For the aim of item selection, two opposite indexes are considered: the *access probability*, i.e. the number of users interested in the item and belonging to the same communities as the local user, and the *availability*, i.e. the number of users in the communities already owning the item. In [26] authors propose a social-aware solution to find the optimal placement of a given piece of content in an opportunistic network. The idea is to iteratively migrate contents to nodes that are increasingly "central" to the overall network, i.e. nodes such that the average cost of accessing the content from any other interested node is increasingly lower. To this end they defined a metric that captures the node's social significance to establish paths between nodes. Finally, they used this metric as the basis for creating a small scale network sub-graph over which the small-scale content placement problem is solved sequentially until the optimal or near-optimal location is identified.

Other works consider a publish/subscribe framework. According to this, in [27] some nodes are identified as *brokers*, and are in charge to coordinate item distribution and to convey items to interested nodes. The brokers are the most popular nodes in terms of social ties and encounters with the other nodes. In SocialCast [28], nodes distribute information about the channels they are interested in. Each node uses this information and its pattern of encounters to compute its own utility for each interest. When two nodes $n1$ and $n2$ encounter, an item is sent from $n1$ to $n2$ if $n2$ has greater utility than $n1$ for the item channel. This approach uses routing – more than caching – in order to deliver content to interested nodes. Moreover, it relies on the assumption that nodes belonging to the same social community share the same interests.

Completely different approaches for data dissemination leave the definition of heuristic policies used to make local caching decisions in favour of solutions based on global utility functions to be solved as a global optimisation problem. Nodes single caches are viewed as a big, cumulative caching space. Reich and Chaintreau [29] for example, focus on the problem of finding a global optimal allocation for a set of content items assuming that users are impatient, i.e., users' interest for items monotonically decreases with the time they have to wait before their request is fulfilled. The problem of what items to fetch upon contact is defined as a global optimisation problem in a similar way as in ContentPlace. Differences are in the fact that the resources to be considered are not the single node cache, but the global cache is used instead, and the utility function is not computed from a node individual point of view, but it is defined globally. Although such global optimisation approach can find the optimum, it requires global knowledge of the network and a priori information on how users behave, that in practice is very unlikely to be available in an opportunistic scenario.

We point out that none of the above mentioned approaches exploit models coming from the cognitive sciences as we do. We are exploring a completely novel direction with the aim at investigating the suitability of human brain models to devise concrete and effective ICT solutions able to cope with the typical Opportunistic Networking problems.

2.3. Recognition heuristics in data dissemination for opportunistic networks

In [9,6], a preliminary version of the approach presented in this work is proposed. The mechanism is based on two concurrent algorithms: *Recognition* and *Modified-Take-The-Best* (in the following, for short, MT^{2}B). The former aims at determining what channels and items are popular. A channel is popular when many nodes are subscribed to it. An item is popular when it is held by many nodes. Upon an encounter between two nodes, the nodes exchange the set of channels they are subscribed to, and the list of items they hold. For every channel to which the other node is subscribed, and every item it holds, a counter is incremented. When, a channel/item counter is greater than a threshold θ, then the corresponding channel or item is deemed as popular. Two different thresholds, θ_C and θ_I can be used for channels and items respectively.

MT^{2}B aims at determining what items are useful and should then be kept in the local cache. Specifically, it is assumed that nodes contributed a limited size local cache to the dissemination process, and therefore a selection must be made about what data items to store in this cache, among those that are available on the encountered nodes. The utility of an item grows with the popularity of the channel it belongs to, and decreases as it becomes more diffused. According to the status information maintained by Recognition, MT^{2}B ranks the items owned by an encountered node for decreasing utility. In particular, the following rules are used: (i) items belonging to unpopular channels are considered useless; (ii) already diffused items are considered useless. Then, subject to the local memory availability, a node selects the most useful items and uploads them in its own local cache. In this sense, channel popularity boosts the caching of (currently) unpopular items, while item popularity stops replication in further nodes.

This approach has two main drawbacks. On the one side it relies on fixed thresholds to be tuned according to the environment, the node mobility and their encounter pattern. Moreover, in presence of highly dynamical scenarios where new items are continuously created, this staticity of parameters becomes even more limiting. On the other side, the amount of punctual state information every node has to keep in order to take decisions about the diffusion state of data items can become intractable w.r.t. the memory constraints nodes are subject to (we provide a quantitative analysis of this point in Section 4). These characteristics harm the actual suitability of this approach for its successful application in real world scenarios.

In this paper we go beyond these limitations reducing the state maintained at nodes by compressing the knowledge about items diffusion into an *aggregate* measure that let us identify, in terms of probability, if the items belonging to a given channel of interest are spread enough, so as to stop their diffusion in favour of other less diffused items. Then, we exploit the *aggregate* measure to drive the dissemination process through a stochastic mechanism. The stochastic mechanism permits to remove from our approach the dependency on the fixed threshold used to recognise the data items' degree of diffusion, thus making it scenario-independent. Given that in real world scenarios the number of channel is far less than the number of data items, in terms of scalability, keeping detailed state information about channel popularity is not a critical aspect. Hence, in order to recognise when a channel becomes popular we keep the mechanism presented in [9,6] unchanged.

The work presented in the following sections is an extension of the one we presented in [10]. The main extensions that we add in this paper regards a detailed mathematical description of the model we used to define the aggregate measure of items diffusion and an extensive set of experimental results, including a comparison with another state-of-the-art solution and tests under various, different scenarios.

3. Problem statement and system assumptions

We consider a system composed by N nodes. Nodes can subscribe to one or more *channels* of interest. We assume that there are K channels available. Every node can generate content *items*. Each item i is labelled with the identifier of the channel of interest it belongs to, $i.ch$. A node can generate items also for channels it is not subscribed to. There is no global knowledge of the channel subscriptions, nor of the pattern of encounters among nodes. Nodes have finite memory availability, thus being unable to store an unlimited number of items. Items have an infinite lifetime, i.e. there is not any maximum lifetime after which they become irrelevant to nodes subscribed to their channel. New channels may be created dynamically, nodes can subscribe to them. New items belonging to existing channels may appear dynamically in the system. Finally, we assume also that, due to energy saving policies, nodes may activate their network interfaces for limited time slots only and disappear from the network (and the dissemination process) for the remaining time.

Due to the lack of global knowledge, nodes have to discover the system status, and take decisions about what items to cache accordingly. Caching permits to carry items around the network till encountering nodes interested in them. As the primary goal, for each item i belonging to a channel ch, the diffusion procedure must *maximise coverage*, i.e., maximise the probability that all nodes subscribed to ch will eventually receive i. Taking into account the characteristics of the OppNets, a secondary goal is to also consider energy saving and (more in general) resource consumption, by limiting communication when this does not jeopardise the coverage.

4. Data dissemination based on probabilistic recognition

As it will be further explained in Section 4.2, our dissemination algorithm strongly depends on the estimation of a parameter we will denote as π^{ch}. The parameter π^{ch} is defined as *the probability with which a generic node can find already known data items belonging to the channel of interest ch, during encounters with other peers*. Specifically, a data item is already known if it has been seen already on other previously encountered nodes. It is worth noting that π^{ch} is a local parameter, therefore its value differs among nodes. Hereafter we present how we estimate the π^{ch} value following a Bayesian approach. Precisely, we are facing the problem of the online estimation of the time-varying probability distribution π^{ch} given the information a tagged node receives by other peers during contacts. Note that the model we describe refers to a tagged node n and a specific channel of interest ch.

4.1. Bayesian roots of probabilistic recognition

Let us consider a tagged node, and let S_t^{ch} be the set of items belonging to a certain channel ch, received during an encounter e at time t with another node. Let us denote with $S'^{ch}_t \subseteq S_t^{ch}$ the set of items that are definitely new w.r.t. the node experience, i.e. items that a node has never seen before. Note that, as explained in detail in Section 4.2, S_t^{ch} and S'^{ch}_t can be computed by keeping the state information maintained at nodes constant, irrespective of the number of data items in the system.

The information about which data items in S_t^{ch} are already known (or completely new) w.r.t. the node experience can be encoded through the indicator function (1) into a binary string $\mathbf{b}_t^{ch}$ of length $|S_t^{ch}|$ where: 1s correspond to data items that are already known by the node and 0s to new ones.

$$\mathcal{H}(s) = \begin{cases} 1 & \text{if } s \notin S'^{ch}_t \\ 0 & \text{otherwise} \end{cases}, \quad \forall s \in S_t^{ch}. \tag{1}$$

We consider each element $b_i \in \mathbf{b}_t^{ch}$ as a realisation of a random variable B_i belonging to the random vector $\mathbf{B}_t^{ch}$ of the same size of $\mathbf{b}_t^{ch}$. Moreover, we assume that all the $B_i \in \mathbf{B}_t^{ch}$ are i.i.d. and follow a Bernoulli distribution of parameter π_t^{ch}:

$$B_i \sim \text{Bernoulli}(b; \pi_t^{ch}).$$

Thus, by ideally concatenating all the random vectors $\mathbf{B}_k^{ch}$ with $0 \leq k \leq t$, we obtain a sequence of Bernoulli random variables where at any point in time

$$E[B_{it}] = \pi_t^{ch}.$$

We remark that the value of π_t^{ch} is the number of already known data items and it can change between subsequent observations due to the dynamics of the their spreading in the system.

Let us now consider a new random variable $Y_t^{ch} = f(\mathbf{B}_t^{ch}) = \sum_{\forall B_i \in \mathbf{B}_t^{ch}} B_i$. Y is thus the number of already seen data items at time t, and follows a binomial distribution with parameters $n_t^{ch} = |S_t^{ch}|$ and π_t^{ch}

$$p(y_t : n_t, \pi_t) = \binom{n_t}{y_t} \pi_t^{y_t} (1 - \pi_t)^{n_t - y_t}, \quad y_t = 0, \ldots, n_t \tag{2}$$

where, for simplicity, we omitted the superscript ch. From now on we turn our attention on the resulting process $\{\mathbf{Y}_k\}_{0 \leq k \leq t}$ composed by a sequence of Binomial random variables.

Now we briefly introduce the theoretical framework, extensively presented in [30], through which nodes can estimate the parameter π_t^{ch} based on locally available information only. In order to improve the readability, from now on we will omit the superscript *ch* but let us recall that every node keeps a separate π_t value for each channel of interest.

Let us consider the distribution of π_{t-1} given all the past information expressed through the Bayes formula:

$$p(\pi_{t-1}|Y_1,\ldots,Y_{t-1}) = \frac{p(Y_1,\ldots,Y_{t-1}|\pi_{t-1})p(\pi_{t-1})}{\int_0^1 p(Y_1,\ldots,Y_{t-1}|\pi_{t-1})p(\pi_{t-1})\mathrm{d}\pi_{t-1}} \propto p(Y_1,\ldots,Y_{t-1}|\pi_{t-1})p(\pi_{t-1}) \tag{3}$$

where, according to Bayesian statistics, the first term of (3) is the Likelihood function and the second one is the prior distribution of π_{t-1}. Let us remember that the prior distribution at time $t-1$ is conditioned to all the past observations up to time $t-2$: $p(\pi_{t-1}) = p(\pi_{t-1}|Y_1,\ldots,Y_{t-2})$. That is, the prior belief we are considering at every time step takes into account all the information observed until the last encounter. In the Bayesian framework it is well known that a common prior distribution for π_{t-1} is the Beta distribution with parameters a_{t-1}, b_{t-1} [31]:

$$p(\pi_{t-1} : a_{t-1}, b_{t-1}) = \frac{\Gamma(a_{t-1}+b_{t-1})}{\Gamma(a_{t-1})\Gamma(b_{t-1})}\pi_{t-1}^{a_{t-1}-1}(1-\pi_{t-1})^{b_{t-1}-1}. \tag{4}$$

In this case, Beta's parameters a_{t-1}, b_{t-1} count respectively the number of known and not known data items seen up to time $t-1$. Given that the distribution of items in the network evolves over time, our problem of estimating the current value of π_t passes through the updating of the Beta parameters by means of an Exponential Weighted Moving Average (EWMA) Filter, as described in [30]. Firstly we apply a discount factor $\alpha \in (0, 1]$ to their value at time $t-1$:

$$a_{t|t-1} = \alpha a_{t-1} \tag{5}$$

$$b_{t|t-1} = \alpha b_{t-1} \tag{6}$$

then, when the tth observation becomes available the distribution of π_t is properly updated. In this specific case, the prior and the posterior distributions belong to the same distribution family, i.e. they are *conjugate* distribution, hence the parameter π_t follows again a Beta distribution with parameters:

$$a_t = a_{t|t-1} + y_t \tag{7}$$

$$b_t = b_{t|t-1} + n_t - y_t. \tag{8}$$

Summarising, we are counting, on average through an EWMA Filter, how many known (y_t) and not known ($n_t - y_t$) items are received by a node during contacts. We need a filtered mean other than a normal one because of the non-stationarity of the parameter distribution π that evolves together with the item diffusion process in the system.

Finally, starting from the updated status of the beta parameters we obtain the current distribution of π_t for which we calculate its conditional expectation

$$E[\pi_t|Y_k, k = 1,\ldots,t] = \tilde{\pi}_t = \frac{a_t}{a_t + b_t}. \tag{9}$$

As we can notice, for every time step the value of $\tilde{\pi}_t$ only depends on $\tilde{\pi}_{t-1}$ and the new observation. This suggests that we can approximate the conditional expectation (9) by filtering the sample mean of the binary vectors received during contacts, as described in the following.

Let us consider again the two sets S_t^{ch} and $S_t'^{ch}$ previously defined. We define a measure of the ratio of unknown data items a node observes upon the encounter e in terms of the sample mean on the current observation:

$$N_t = \frac{|S_t'^{ch}|}{|S_t^{ch}|}. \tag{10}$$

Eq. (10) measures the ratio of novel information received from an encountered node w.r.t. a given channel. We use its complement as an instantaneous indicator of the diffusion of the items in *ch*. Let $\hat{\pi}_t^{ch}$ be the estimated degree of diffusion of the items in *ch*, at the time t. Finally, we can estimate $\hat{\pi}_t^{ch}$ using a standard exponential smoothed average technique, as follows:

$$\hat{\pi}_t^{ch} = \alpha\hat{\pi}_{t-1}^{ch} + (1-\alpha)(1-N_t) \tag{11}$$

where $0 \leq \alpha \leq 1$ regulates the balancing between the past experience and new information. As demonstrated in Appendix the estimations in (11) and (9) are equivalent within a scale factor, thus in our algorithm we decided to use the one in (11). Fig. 1 shows the typical trend of both $\hat{\pi}^{ch}$ and $\tilde{\pi}^{ch}$ we have observed in our simulations (details on the simulation settings are provided in Section 5). It shows that as time passes, items become more and more spread, and the probability of observing new items goes to zero bringing the diffusion probability close to 1.

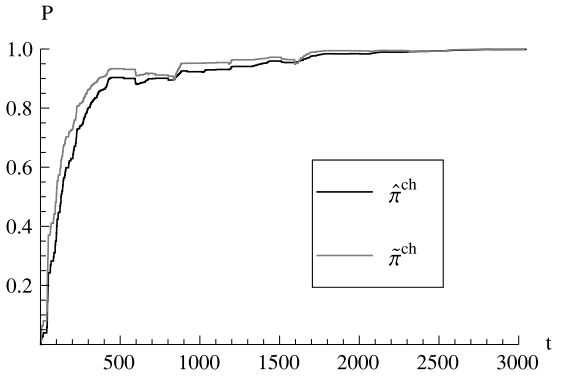


Fig. 1. Increasing trend of the diffusion probabilities π^{ch} and $\hat{\pi}^{ch}$ during the system evolution.

4.2. Data dissemination algorithms

In this section, we present how the model described in Section 4.1 can be practically implemented in order to merge the recognition heuristic with the probabilistic approach and exploit them in an opportunistic networking scenario. Before doing so, let us briefly recall the structure we assume about each node's memory space. This is the same used in [9,6], and is reported here for reader's convenience:

Data caches.

- Local Items cache (LI): contains the items generated by the node itself;
- Subscribed Channel cache (SC): contains the items belonging to the channel the node is subscribed to and obtained by encounters with other nodes;
- Opportunistic Cache (OC): contains the most "useful" items from a collaborative information dissemination point of view. These items are obtained by exchanges with other nodes and belong to channels the node is not subscribed to.

Recognition cache.

- Channel Cache (CC): whenever a node meets another peer subscribed to a given channel, the channel ID is put in this cache, along with a counter.
- Items' Channel Cache (ICC): contains the channel IDs and the aggregate information about the diffusion probability of items.
- Item Hash (IH): a Bloom filter, used to remember which items a node sees along encounters.
- Channel Hash (CH): a Bloom filter, used to remember recognised channels no longer present in CC.

The main logical steps of the data dissemination algorithm based on probabilistic recognition are as follows (upon encountering with another node):

1. recognise which channels are popular;
2. recognise if the items of a channel are spread;
3. fill up the shared memory with the less spread items of popular channels for redistribution.

Step 1. Now we will refer to Algorithm 1, the same proposed in [9] and here reported for reader's convenience. For every contact between two nodes, each of them increments the counters associated to the other node's subscribed channels (line 6) until a given threshold θ_C is reached, after that the channel is marked as recognised (line 8). If the number of entries in CC exceeds the maximum capacity, then the oldest entry is dropped (line 18). In this case, if it was marked as *, recognised*, the channel ID is recorded in a Bloom Filter (CH). In this way, the nodes can distinguish between channels that are not in CC because they have never been seen (in this case they are not in the CH), and channels that have been replaced. Once concluded the recognition phase for channel popularities, the second step begins.

Step 2. As discussed in Section 4.1 we minimise the state information maintained about item diffusion into an aggregate measure. We will now refer to Algorithm 2. Upon an encounter, two nodes exchange the content summary of their caches (LI + SC + OC). Let us consider the set of item IDs received and belonging to a given channel (line 9). By querying the Bloom Filter IH containing the information about all the items received during past encounters, we count how many of them are definitely new (lines 11–14) and update the diffusion probability (line 19) corresponding to that channel according to Eq. (11). It is worth noting that the decision of counting the new items instead of the replicas is driven by the intrinsic characteristics of the Bloom Filter. Due to the probabilistic nature of a Bloom Filter, there is a non-null probability of obtaining a false positive when querying if an item is present in the data structure. By contrast, the negative answer is always true, thus we rely only on definitely negative answers, which may lead, in principle, to a slight under-estimation of the number of new items, and thus to stopping the diffusion process too early. Our simulation results show that this has, in practice, no impact

Algorithm 1 Channel Recognition

```
1: Let ch be an observed channel.
2: Let θ_C be the recognition threshold for channels
3: Let B be the CC max size
4: if CC.contains(ch) then
5:     if ch is not recognised then
6:         R_ch ← R_ch + 1
7:         if R_ch = θ_C then
8:             mark ch as recognised
9:         end if
10:        reset ch.TTL
11:    end if
12: else
13:    if CC.size = B then
14:        select the channel ch′ with oldest TTL
15:        if ch′ is recognised then
16:            CH ← CH ∪ ch′
17:        end if
18:        CC ← CC\ch′
19:    end if
20:    CC ← CC ∪ ch
21:    ch.counter = 1
22:    Set ch.TTL
23: end if
```

on the effectiveness of the dissemination process. Once updated the diffusion probability, we use it to decide whether the data items of that channel are recognised or not according to a Bernoulli trial with probability $\hat{\pi}^{ch}$ (lines 20–25):

$$B(\hat{\pi}^{ch}) = \begin{cases} 1 \Rightarrow \text{Items are considered as diffused} \\ 0 \Rightarrow \text{Items are considered as not diffused.} \end{cases} \tag{12}$$

In this way, as long as a node does not receive any new information about a channel *ch*, the corresponding value of π^{ch} (one for each channel and different for each node) gets increasingly close to 1, strengthening over time the belief that the items of *ch* are diffused. The drawback of using in the recognition process an aggregate measure together with the stochastic approach is that this results in a loss of granularity w.r.t. the information about the single items diffusion. However, the benefit is twofold: (i) the nodes can autonomically adapt to the local scenario, and do not need to rely on a predefined threshold to be tuned, and (ii) the randomness of the decision process permits to sporadically restart the diffusion of almost spread items thus increasing the probability of reaching those few nodes that for some reason are not aligned with the mean condition of the system.

In principle, from a technical point of view the size of the Bloom filter (IH) should be defined a priori based on the number of elements to be stored and the desired false positive probability, being impossible to store extra elements without increasing the false positive probability. In this work, we explore two possibilities. On the one hand, we use a Scalable Bloom filter, a variant of Bloom Filters that can adapt dynamically to the number of elements stored, while assuring a maximum false positive probability [32]. This solution guarantees a fixed false positive rate, at the cost of a modest linear increase of the state size with the number of items. On the other hand, we also consider fixed size Bloom filters, dimensioned as a fraction of the theoretical optimal size (computed with complete information about the number of data items in the system). This guarantees a constant state size, irrespective of the number of data items, at the possible cost of an increase of the false positive rate. Simulation results presented in Section 5.2.7 show that using fixed size Bloom filters has no significant effect on the performance of the data dissemination process.

Step 3. The results of the probabilistic recognition process are then exploited by the MT^{2}B algorithm to select the less spread items – of the recognised channels – to be stored for redistribution. We will now refer to Algorithm 3. Although it only partially differs from the one presented in [9,6], we will completely describe it for the reader's convenience. Upon meeting, nodes exchange summaries of the items they are carrying in their caches. Items belonging to the node's subscribed channel are fetched and stored in the node's SC (lines 2–6). Then, each node selects which of the remaining items owned by the other peer should be fetched in order to be redistributed. Firstly, each node selects those items whose channel is *recognised* (lines 11–14). If the number of these items is greater than the OC maximum capacity, each node selects those items whose channel is *recognised* but currently marked as *not diffused* (lines 16–21). Finally, if the selected items are still too many w.r.t. the OC maximum capacity, the MT^{2}B sorts the items by their ascending $\hat{\pi}^{ch}$ value and fills up the OC with the first *n* items according to its capacity (lines 22–24).

Thanks to this approach nodes have to maintain less state information than that maintained in [9,6]. Let us assume the Bloom filter size as fixed, and let us denote with K the number of channels and I the number of items per channel. In the

Algorithm 2 Probabilistic Recognition

Let M be the set of items received from another node.
Let I_{ch} be the counter for the items in M that belongs to the channel ch and are not present in IH
Let C_{ch} be the counter for the items in M that belongs to the channel ch
Let $\hat{\pi}^{ch}$ be the diffusion probability of the items that belongs to the channel ch
Let $B(\hat{\pi}^{ch})$ be a Bernoulli random number generator
Let $0 \leq \alpha \leq 1$
$I_{ch} \leftarrow 0$
$C_{ch} \leftarrow 0$
for all $i \in M$ **do**
    if ICC.contains($i.ch$) **then**
        if ($\neg$ IH.contains(i)) **then**
            $IH \leftarrow IH \cup i$
            $I_{i.ch} \leftarrow I_{i.ch} + 1$
        end if
        $C_{i.ch} \leftarrow C_{i.ch} + 1$
    end if
end for
for all $ch \in$ ICC **do**
    $\hat{\pi}^{ch} \leftarrow \alpha * \hat{\pi}^{ch} + (1 - \alpha) * (1 - \frac{I_{ch}}{C_{ch}})$
    if $B(\hat{\pi}^{ch}) = 1$ **then**
        Mark items of ch as *diffused*
    else
        Mark items of ch as *not diffused*
    end if
end for

Algorithm 3 Modified *Take-The-Best*

Les M be the set of items received from another node
for all $i \in M$ **do**
    if $i.ch$ = subscribedChannel **then**
        SC $\cup = i$
    end if
end for
Let $M' = M -$ SC
Let B the OC storage capacity limit
Let $I = M' \cup$ OC
Let *recChItems*$= \emptyset$
for all $i \in I$ **do**
    if $i.ch$ is recognised **then** *recChItems* $\cup = i$
    end if
end for
Let *notSpreadItems* $= \emptyset$
if *recChItems*.size$> B$ **then**
    for all $r \in$ *recChItems* **do**
        if $r.ch$ is marked as *not diffused* **then**
            notSpreadItems $\cup = r$
        end if
    end for
    if *notSpreadItems*.size $> B$ **then**
        Rank *notSpreadItems* in ascending order according to their $\hat{\pi}^{ch}$ value
        Select and keep in OC the first B objects of *notSpreadItems*
    else
        OC $\cup =$ *notSpreadItems*
    end if
else
    OC $\cup =$ *recChItems*
end if

Table 1
Detailed scenario configuration.

Parameter	Value
Node speed	Uniform in (1, 1.86 m/s)
Transmission range	20 m
Simulation area	1000 m × 1000 m
Number of cells	6 × 6
Number of nodes	200, 600
Number of channels	8
Number of items	200 (25 per channel)
Number of groups	8
Number of travellers	56 (7 per group)
Simulation time	25 000 s

probabilistic approach, every node has to keep a Bloom Filter, a counter and an items' diffusion probability value for every channel only, thus the memory requirement has an order of magnitude of $O(K)$ because it grows linearly with the number of channels. By contrast in [9,6] every node has to maintain a counter for each channel and a counter for each data item, which means that the order of magnitude in terms of memory is $O(K * I)$. The improvement is very significant, as in real scenarios $I \gg K$.

5. Performance evaluation

In this section, we evaluate the performance of the *Probabilistic Recognition Algorithm* (hereafter denoted as PR) through a set of experiments by which we show that the proposed solution converges to or outperforms the results of the best fine-tuned configuration of the algorithm proposed in [9] with a significant reduction of resource consumption.

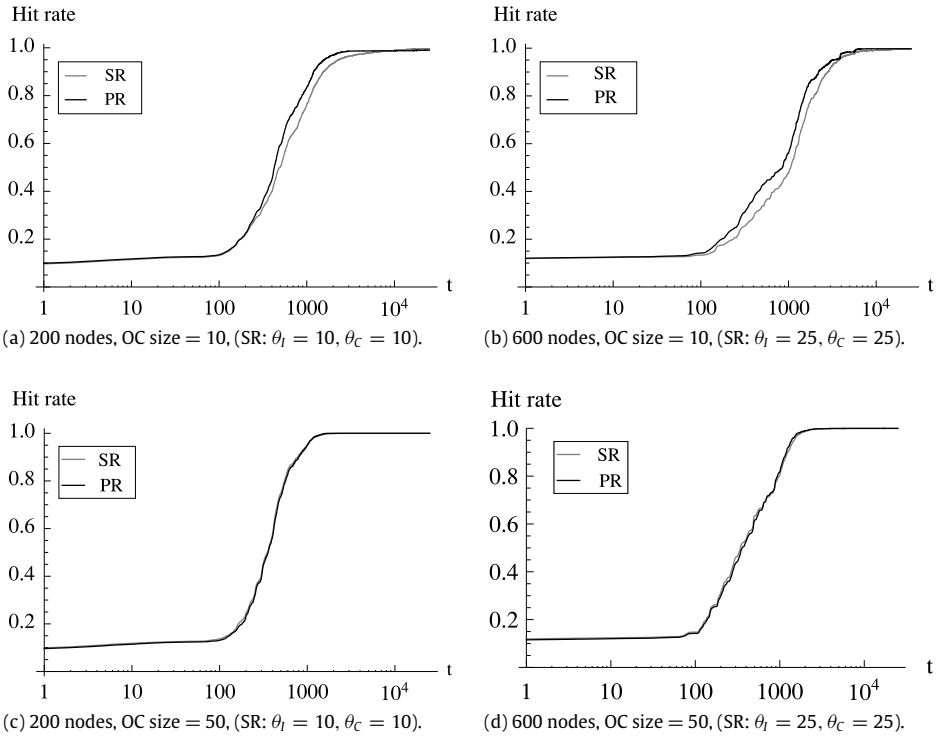
5.1. Experimental setup

In our experiments, nodes mobility is both simulated according to HCMM [33] and drawn from real traces. A detailed description of the real traces will be provided in Section 5.3. HCMM is a mobility model that integrates temporal, social and spatial notions in order to obtain an accurate representation of real user movements. We used it to generate synthetic traces with these parameters: nodes move in a 6 × 6 grid corresponding to a 1000 m^2 square area, and are grouped in very closed communities placed far from each other so as to avoid any "border effect" e.g. involuntary communication between groups. Nodes mobility is limited inside the groups they belong to, except for few of them called *travellers*, which are allowed to visit other groups. With this configuration we want to simulate different social communities where usually people stay, apart for few of them that due to their social relationships can meet people from different social communities. In this context, the only way to exchange data is through nodes mobility, and travellers play an important role because they are the unique bridge among communities.

In our scenarios there are as many channels of interest as groups and every node subscribes to one channel. For each group, all the channels are present with different popularity degrees and are assigned to the nodes according to a Zipf distribution [34] with parameter $\alpha = 1$. Specifically, at the beginning of the simulation, each node selects the channel to be subscribed to. The probability to select channel i is $P_i = \frac{1}{i^\alpha}$ where i denotes the ith most popular channel in the node's community. For symmetry, popularity of the channels is rotated, such that (i.e., considering all groups together) all channels have the same average number of subscribers. Moreover, for each community there is a different most popular channel. This makes the scenario uniform as far as channel popularity is concerned, since the same number of nodes is subscribed to each channel, while the popularity of channels within individual groups is skewed according to a conventional model (Zipf law). Every channel has the same number of items which are initially univocally assigned to nodes according to a uniform random distribution. The detailed scenario configurations are supplied in Table 1. Note that in this scenario the hypothesis that for any tagged node the probability of seeing each data item of a given channel on other encountered nodes with the same probability does not hold true (at least at the beginning of the simulation). However, simulation results show that the effectiveness of the PR algorithm does not suffer from this simplifying assumption.

5.2. Simulation results

We compare PR with the policy proposed in [9,6] that we indicate as Static Recognition (SR). In [6] SR was compared with non-heuristic based data dissemination policies. Namely, in comparison with ContentPlace [25] – one of the reference solutions in the literature – SR shows its effectiveness in terms of hit rate and its better efficiency in terms of network overhead. Thus, we compare PR with SR only. SR represents an "optimal" distributed policy, since it maintains the whole knowledge about observed items. In the simulations, we tune the SR parameters to their optimal values based on the specific scenario considered. This is done by exploiting the sensitiveness analysis presented in [6]. Note that, in practice, it would not be possible to do such fine tuning, and the optimal parameters for SR would need to be estimated. Therefore, we compare



(a) 200 nodes, OC size = 10, (SR: $\theta_I = 10, \theta_C = 10$).
(b) 600 nodes, OC size = 10, (SR: $\theta_I = 25, \theta_C = 25$).
(c) 200 nodes, OC size = 50, (SR: $\theta_I = 10, \theta_C = 10$).
(d) 600 nodes, OC size = 50, (SR: $\theta_I = 25, \theta_C = 25$).

Fig. 2. Hit rate curves of PR (black curve) and SR (grey curve) with different network size 200 (a)–(c) and 600 (b)–(d).

PR against the optimal ideal performance of SR. We performed several measures with PR and different values of θ_C in the range [1, 20] in the considered scenarios. The obtained results show negligible differences—not reported here for the sake of space. Hence, we set the θ_C parameter to 10 and kept unchanged for all experiments.

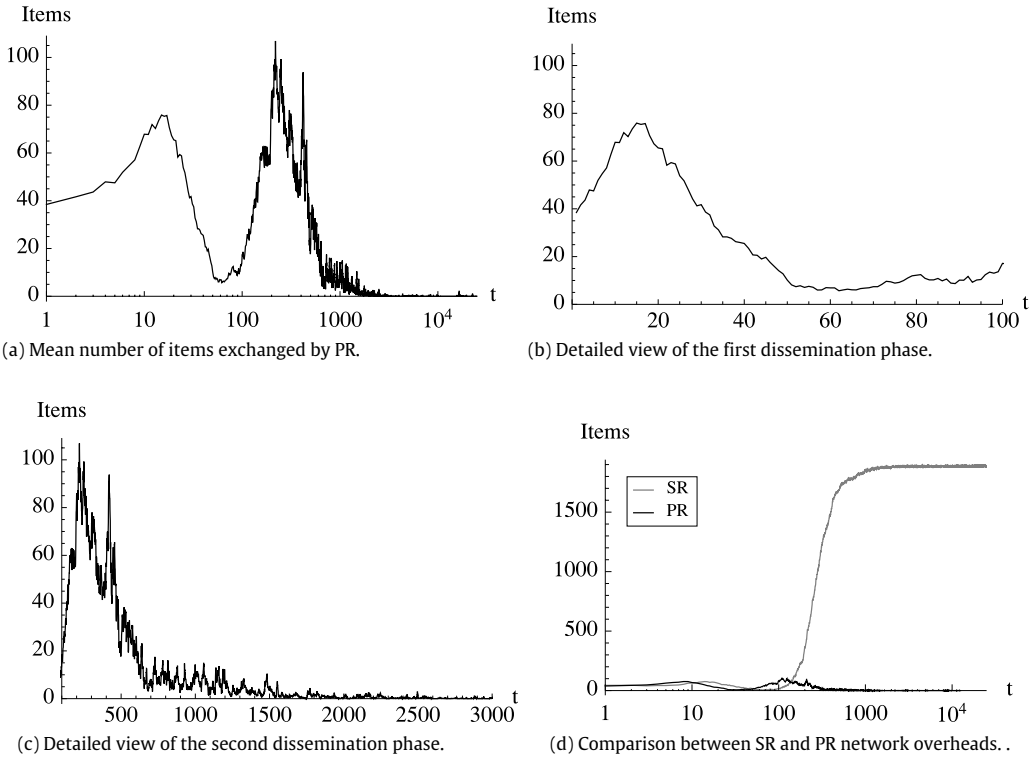
Simulations aim at analysing to what extent PR is able to approximate the performance of SR, in spite of maintaining reduced information and having to autonomically adapt to different environments. We evaluate the performance of both approaches in terms of *hit rate, convergence time* and *network overhead*. The hit rate at a given time is defined as the mean value over nodes of the ratio between the number of items currently present in the SC of each node w.r.t. the total number of data items of the channel to which the node is subscribed. Convergence time is defined as the time instant when the hit rate exceeds 99%. The instantaneous network overhead is measured as the mean number of items exchanged at a given time instant. All the results presented in this paper are averaged on 10 runs where the initial configuration of items and channels were randomly initialised. Confidence intervals at a level of 95% have been computed for hit-rate curves, but not reported in figures for the sake of readability, as they are very close to the hit-rate curves.

5.2.1. Opportunistic cache size sensitivity analysis

The capability of inferring system state from partial knowledge reveals particularly useful when memory for the OC is limited. Indeed Fig. 2(a) shows that in a network composed by 200 nodes with an OC size of 10 items, PR reaches a hit rate greater than 99% more quickly than SR; the gain in terms of convergence time is $\geq$24% (see Table 2). The same behaviour holds for a more dense network as well: Fig. 2(b) highlights the distribution ability of PR in a scenario configured with a network of 600 nodes, an OC size of 10 items, and a number of items significantly smaller than the network size (200). In this configuration, at the beginning of the simulation, just one third of nodes are aware about the contents actually present in the scenario. However, also in this case PR is able to quickly adapt to the situation reaching complete coverage faster than SR. By contrast, the two approaches become almost equivalent when the OC size is sufficiently large (OC size 50) so as to make the item selection a less critical task, as shown in Fig. 2(c) and (d). In order to have a quantitative understanding about convergence velocity, we measure the converge times of the two approaches, shown in Table 2. As we can see, PR outperforms SR without relying on any parameters' fine tuning.

5.2.2. Network overhead

Compared to SR the probabilistic approach is less demanding in terms of resource consumption. Fig. 3(a)–(c) show, at different scales, the mean number of items exchanged by nodes during the simulation on a network of 200 nodes. As we can see, there are two separated phases in content distribution, the first one (Fig. 3(b)) refers to the dissemination process inside



(a) Mean number of items exchanged by PR.
(b) Detailed view of the first dissemination phase.
(c) Detailed view of the second dissemination phase.
(d) Comparison between SR and PR network overheads. .

Fig. 3. Mean number of items exchanged on a network of size 200.

Table 2
Convergence time for a coverage ≥99%.

Experiment	PR (s)	SR (s)	Gain PR vs. SR
Net. size 200, OC size 10	2100	3800	44%
Net. size 600, OC size 10	4400	5800	24%
Net. size 200, OC size 50	1200	1200	–
Net. size 600, OC size 50	1920	2000	4%

groups before the arrival of travellers in the communities. After the 65-th second of simulated time, the dissemination process restarts due to the arrival of travellers inside the communities as depicted by the second phase of the process in Fig. 3(c). Interestingly, after some time both phases show a decrease in the number of exchanged items – which is an indicator of the convergence of the diffusion – and, even more important, it demonstrates that PR does not waste resources to retransmit useless contents. By contrast, as can be derived by the analysis of the algorithm presented in [6], in order to maximise the convergence velocity in SR, the data exchange never stops even when all the items are deemed as recognised (in that case, according to SR nodes exchange data items selected according to a uniform sampling process). Thus, it becomes clear the advantage coming from the probabilistic approach when compared to the network load induced by SR, as shown in Fig. 3(d).

In order to further investigate the differences between PR and SR, we made more comparable the two approaches by limiting the network overhead of SR. To this end, we modified the latter by simply removing the constraint according to which every node must keep its OC always full. This modification directly affects the network overhead due to the fact that only unrecognised items become eligible to enter in OC, allowing it to be in some occasion only partially filled. For simplicity, from now on we will denote the modified version of SR with the acronym SR*. In order to obtain a direct and fair comparison, we run an experiment with the same configuration setup of the former one: 200 nodes, $\theta_I = 10$, $\theta_C = 10$, OC size = 10. As we can notice from Fig. 4(a), by limiting message exchanges, SR* shows a strong decrease in performance: the convergence time goes from 3800 s to ∞, i.e. the hit rate never reaches 99%. Even defining the convergence time at a lower hit rate, it is clear from Fig. 4(a) that PR always outperforms SR*. Moreover, as depicted in Fig. 4(b) the number of items exchanged along time by SR*, although lower than the number of exchanges triggered by SR, is still significantly higher than that of PR and, moreover, message exchange never stops. This behaviour is caused by the replacement policy of the information present in the Recognition Item Cache in SR*, i.e. the cache used by SR in order to *recognise* data items. More precisely, when only few items remain to be disseminated but their number is greater than the size of the Recognition Item Cache the information contained in it is refreshed too frequently and the items cannot reach the *recognised* status, thus leading to the unlimited message exchange that is evident in Fig. 4(b).

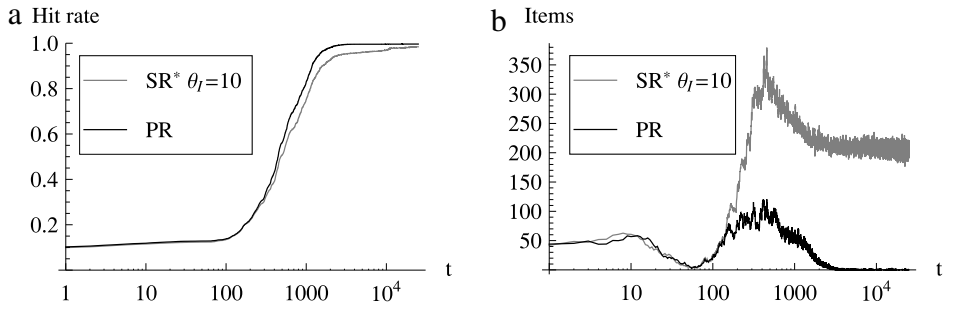


Fig. 4. Comparison between PR and SR* in terms of hit rate (a) and network overhead (b).

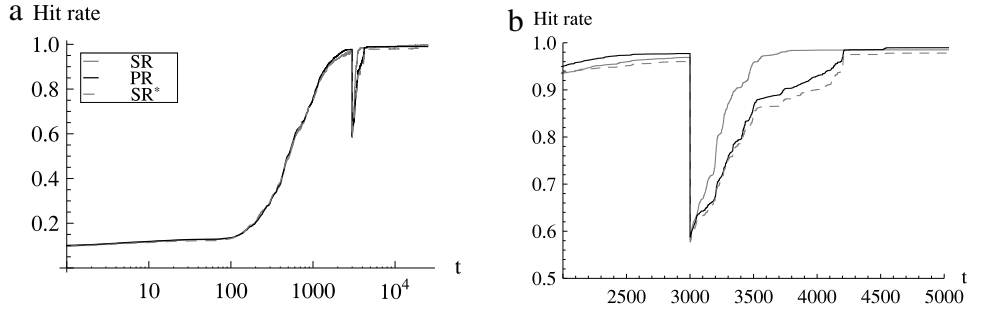


Fig. 5. Hit rate curve of PR (black curve), SR (grey curve) and SR* (dashed grey curve) after a channel injection at 3000 s.

5.2.3. Dynamic scenario: channel injection

In this section we study how PR behaves in a more challenging scenario. At a certain time during the simulation, a set of new items belonging to a new channel are injected in the environment. The number of nodes changing their subscription in favour of the new channel is randomly extracted, and the nodes are randomly chosen with a uniform distribution across groups. Due to this change, these nodes clear their SC just after having run the MT^{2}B algorithm to load in OC possible useful items. At this point the usual Probabilistic Recognition algorithm starts to be applied also to the new channel.

In Fig. 5(a) we can notice that, in a scenario of 200 nodes, after the channel injection at 3000 s both PR, SR and SR* react to the new stimulus, though with different intensity. SR seems to be more responsive, but let us remember that it has been fine tuned to obtain this result and, even more important, it never stops the dissemination process. Indeed, observing the behaviour of SR*, we notice that it is less demanding in terms of networks overhead, but it does not reach convergence. By contrast PR autonomically responds to the channel injection restoring the hit rate trend just after 1000 s. This indicates that PR well approximates the behaviour of SR that, due to its fine tuning, is an upper bound for this scenario. Fig. 5(b) shows in more detail this behaviour. Moreover, in Fig. 6 we can see what happens to the network load when the dissemination process restarts due to the injection of a new channel for both PR, SR and SR*. Again, note the huge difference in overheads between the three approaches.

5.2.4. Dynamic scenario: items injection

In this section, we analyse the behaviour of PR when increasing the scenario's dynamicity. Differently from the previous experiment, at a certain time during the simulation a large number of new data items belonging to an already existing channel are injected in the system. Newly generated items are randomly assigned to nodes belonging to the same randomly chosen community. In this way, we are trying to put our approach in an unfavourable condition, according to which all new items must traverse the entire system in order to reach all the interested nodes. Moreover, this kind of experiment is a way to simulate what can happen in real scenarios where users are allowed to generate new contents belonging to already existing topics. Here nodes are not requested to modify their memory status (as for the previous experiment) but they are expected to simply react to the injection. In Fig. 7 we see how PR and SR react to the injection of 1500 new data items belonging to the channel 0 in a network of size 200. Apparently, it seems that both PR and SR have almost the same performance but if we consider Table 3, which reports the time instants – averaged over all channels – when PR and SR reach a specified hit rate level, we can notice that PR proves to be more effective than the best SR's parametrisation ($\theta_I = 25, \theta_C = 10$) for this scenario. Moreover, looking at the last row in the table we observe how a good parametrisation for a given scenario ($\theta_I = 10, \theta_C = 10$ in the experiment of Section 5.2.3) can be a wrong parametrisation for another scenario. This highlights the advantage we obtain from PR which self-tunes to the optimal operating condition. In order to have a deeper understanding on the behaviour of both the approaches in presence of a massive injection of new data items, let us consider Fig. 8 where we show the difference between the hit rate curves for the channel subjected to the new items injection and one of the other channels not affected by it.

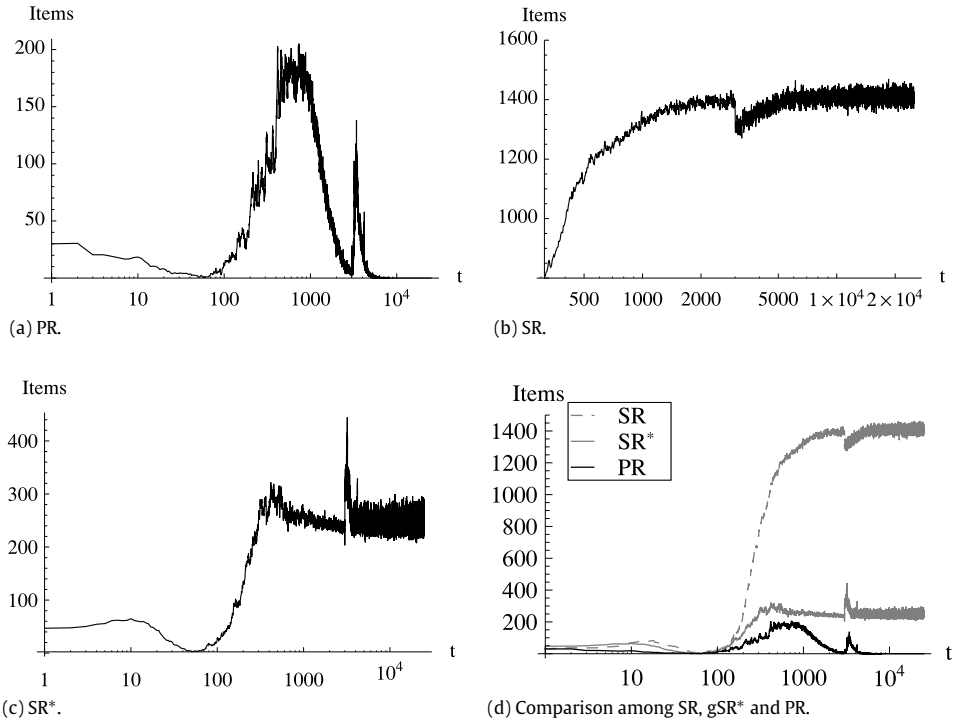


Fig. 6. Mean number of items exchanged in a network of 200 nodes. Channel injection at 3000 s.

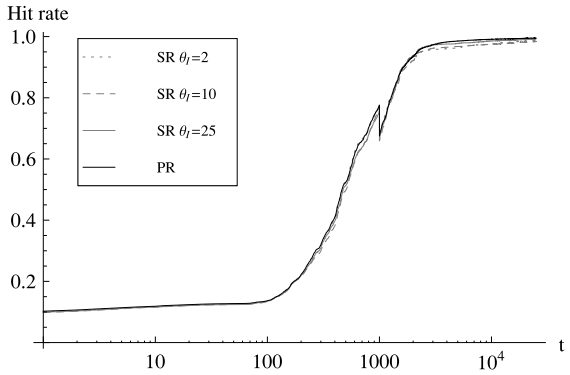


Fig. 7. Hit rate curves of PR and SR with different parametrisation. Injection of 1500 items at 1000 s.

Table 3
Time instants in which PR or SR reach a specified hit rate level.

Hit rate	PR	SR $\theta_I = 2$	SR $\theta_I = 10$	SR $\theta_I = 25$
50%	462	467	503	474
60%	581	575	611	604
70%	794	801	837	836
80%	1267	1260	1276	1299
90%	1616	1590	1705	1590
93%	1893	1945	2073	1827
96%	2287	2940	2664	2235
97%	2725	5504	4972	2586
98%	3312	11000	13449	5046
99%	6163	∞	∞	12943

As we can see in Fig. 8(b), hit rate curves are quite similar to the one presented in Fig. 2. Instead, in Fig. 8(a) we see how PR and SR react to the injection of 1500 items. Here SR is apparently more responsive than PR, but this higher responsiveness is justified by the policy adopted by SR according to which it never stops the message exchange in order to maintain the OC

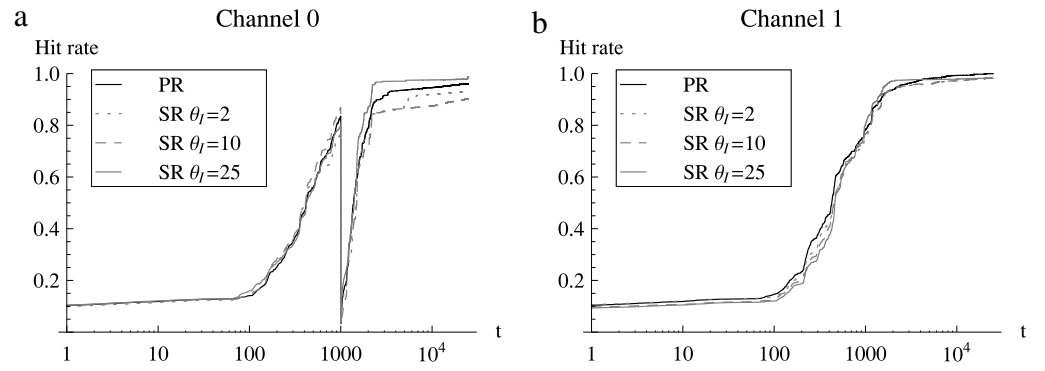


Fig. 8. Hit rate curves of PR and SR of each separate channel of interest. At 1000 s, 1500 items belonging to channel 0 are injected into the system.

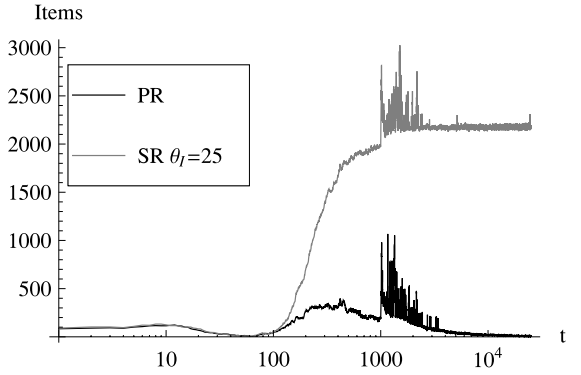


Fig. 9. Network overhead when 1500 new data items of channel 0 are injected at 1000 s.

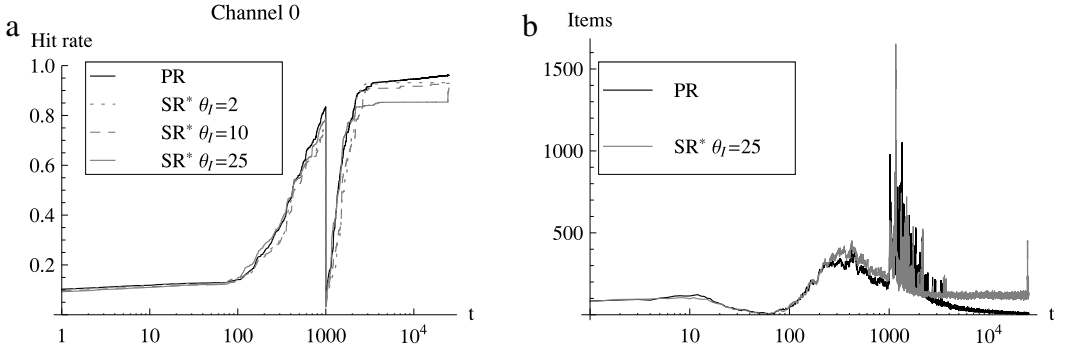


Fig. 10. Hit rate curves of PR and SR* (a) and comparison between the corresponding network overhead (b).

cache always full, as shown in Fig. 9. Conversely, when we compare PR and SR* – which limits the message exchange – we notice that SR* becomes less responsive than PR, as shown in Fig. 10(a). Moreover, as before we notice that SR* never really stops the dissemination (Fig. 10(b)) due to the fact that some items never reach the *recognised* status and are continuously exchanged.

5.2.5. Dynamic scenario: repeated items injection

In order to analyse our solution in even more realistic conditions where new data items belonging to an existing channel appear in the system along time, we devised a second experiment of items injection. In this case, newly generated data items are repeatedly injected in a network of 200 nodes at different time steps. More precisely, in this experiment every 1000 s from time 1000 s to time 6000 s, 200 new data items belonging to channel 0 are injected in a randomly chosen community. Fig. 11(a) shows the comparison between the hit rate curves of PR and SR. As we can see every 1000 s the hit rate level decreases due to the injection of 200 new data items in the system. As shown in Table 4 PR reaches convergence slightly faster than the standard SR in a more efficient way w.r.t. the network overhead, as shown in Fig. 12(a).

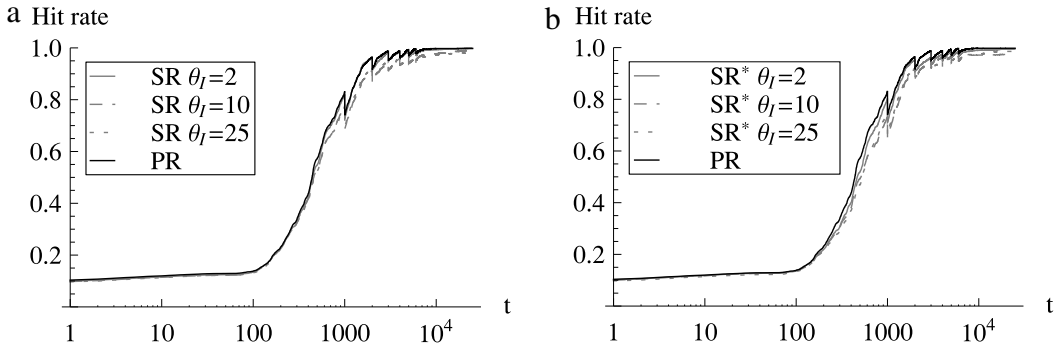


Fig. 11. Hit rate curve for (a) PR and SR, and (b) PR and SR*.

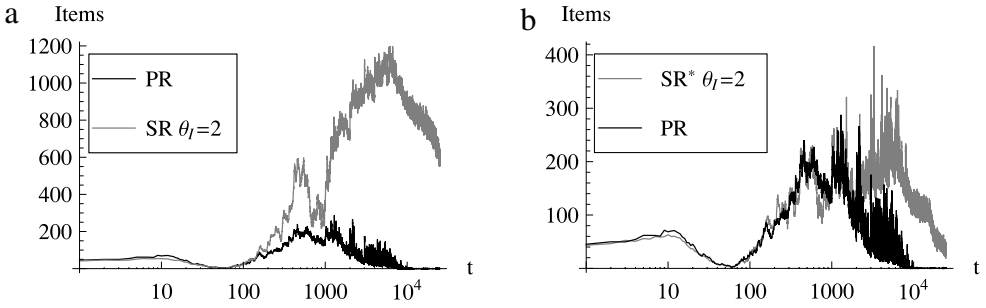


Fig. 12. Network overhead comparison of PR vs. (a) SR and (b) SR* during repeated item injection.

Table 4
Convergence time comparison between PR, SR and SR* during repeated item injection.

PR	SR	SR*	
	6815 s	8316	$\theta_I = 2, \theta_C = 10$
6450 s	19 153 s	∞	$\theta_I = 10, \theta_C = 10$
	∞	∞	$\theta_I = 25, \theta_C = 10$

For the sake of completeness, we performed the same experiment also with SR*. As reported in Fig. 11(b) the hit rate curve behaviour is almost the same as in the previous results, but looking at Table 4 we notice that even though SR* is more efficient in terms of network overhead w.r.t. its standard version, as shown in Fig. 12(b), the time needed to reach convergence is still larger than that of PR.

These experiments strongly highlight the unfeasibility of SR (and SR*) for a real world deployment. As we can notice, its good performance is subject to many factors: the scenario, the OC occupation policy, the rate at which new items appear in the system, the number of new contents to manage and so on. For each of the above mentioned factors we must find a different "best" parametrisation of SR (and SR*) and this strongly limits its real usability. Conversely, PR proves to be more flexible across a range of different scenarios, it always reaches the coverage with limited network overhead and in reasonable time.

5.2.6. Churning nodes

We now explore how PR behaves under other, more challenging conditions where every 1000 s, each node has a probability to become inactive. Although inactive, a node continues to move in the environment but it neither participates to the dissemination process nor collects information it is interested in. Anyway, it does not delete the information and experience collected so far. Once every 1000 s, nodes can become active again and restart the dissemination process. We consider a network of 200 nodes. In the following we show results in which every node, since time 1000 to time 5000, has a deactivation (and re-activation) probability of 0.5 so as, on average, only half of the nodes are active. Fig. 13(a) shows a comparison between the typical hit rate trend obtained in a scenario with (black curve) and without churning nodes (grey curve). After the first probabilistic deactivation at 1000 s we can notice a drastic decrease in the slope of the churning curve. This is quite expected if we think that almost 50% of nodes stops exchanging information. Interestingly, between every two activation/deactivation time steps we can notice two separated phases as shown in Fig. 13(b): a strong increase in the hit rate curve followed by another slowdown just before the next activation/deactivation point. We justify this behaviour by the fact that all the sleeping nodes, after being active, re-join the information dissemination process and start to collect

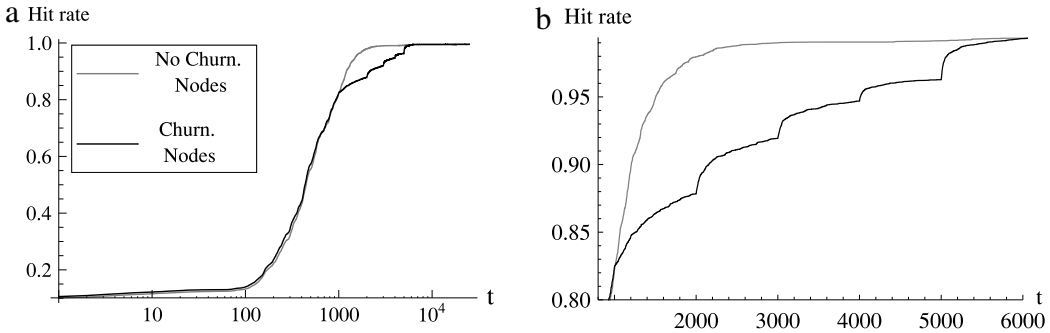


Fig. 13. (a) Hit rate curves of PR with churning nodes (black curve) and without churning nodes (grey curve). (b) Detailed view of the hit rate curves. Activations and deactivations occur every 1000 s from time 1000 to time 5000.

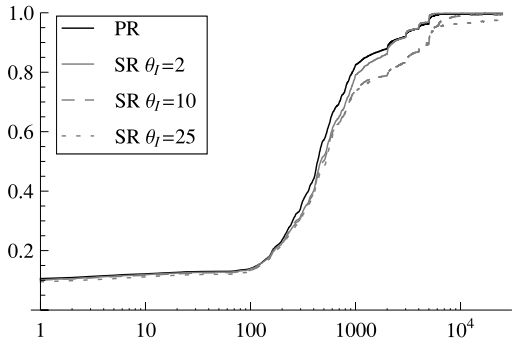


Fig. 14. Hit rate comparison between PR and SR with churning nodes.

Table 5
Convergence times with churning nodes with OC size 10.

Algorithm	Conv. time
PR Churn	5300 s
SR $\theta_I = 2, \theta_C = 10$	5100 s
SR $\theta_I = 10, \theta_C = 10$	8900 s
SR $\theta_I = 25, \theta_C = 10$	∞

Table 6
Sensitivity analysis with reduced Bloom Filter size.

B.F. size reduction	100%	80%	60%	40%
Hit rate	$\geq$99%	97%	98%	98%
Conv. time (s)	2100	4000	10 400	16 300

and redistribute information with other peers. However, at the same time other nodes become inactive, which triggers the "slowdown" phase: when the active nodes have collected and distributed all the temporarily available information, the entire dissemination process reaches another plateau until the next re-activation. After time 5000 all nodes return to be active and the dissemination process reaches convergence very quickly. When compared to a usual scenario where nodes are not allowed to become inactive (see Table 2), we notice that the convergence time is more than doubled, as shown in Table 5, yet our approach still yields quite good performance as shown in Fig. 14.

5.2.7. Bloom filter sensitivity analysis

Finally, as anticipated in Section 4.2, we present the results of a sensitiveness analysis to evaluate the robustness of our approach in the presence of an even less reliable diffusion information about channels. To this end we performed a series of experiments where the IH size was reduced up to 40% of its initial size, that, in normal conditions is set to the number of items present in the scenario (200). Results can be found in Table 6 where we reported both the maximum coverage obtained and the corresponding convergence time.

These experiments show that finely dimensioning the size of IH is not of primary importance. Even when IH is drastically under dimensioned, PR still archives almost 100% hit rate (even though through a slower dissemination process). This shows

Table 7
Real world experimental setup.

Experimental dataset	Rollernet	Infocom06	SigComm09
N. nodes	62	78	76
Trace length (s)	9976	322 149	334 537
Mean *n*. contact per pair per hour	11.67	0.2194	0.3712
Mean contact length (s)	5.65	111.2	4.71
Mean inter-contact time (s)	366.4	8128	5708
N. channels	8	8	8
N. items per channel	100	100	100

that PR basically does not need tuning of parameters and can be used without prior knowledge in a range of situations, as its performance is mostly unaffected by the size of the IH, which is the only parameter that in principle needs to be tuned.

5.3. Real world scenarios

We evaluated the performance of the algorithms with three real traces, namely, Rollernet [35], Infocom06 [36], and SigComm09 [37]. Rollernet and SigComm09 are available in the CRAWDAD repository, while Infocom06 has been supplied by the researchers performing the experiment. As it will emerge from the datasets' description, such real traces, although similar in terms of number of involved nodes, present very different characteristics. Hence, they represent an interesting test bed for our cognitive solution and its competitors.

The *Rollernet* trace involves 62 iMote devices equipped with a Bluetooth interface, and recording contacts among both themselves and other Bluetooth devices in the surrounding environments, with a scanning period of roughly 15 s. The trace was collected distributing the devices to rollerbladers moving across Paris over 3 h. The Rollernet trace comes with symmetrised contacts: a contact between two nodes A and B lasts the time that A sees B or B sees A.

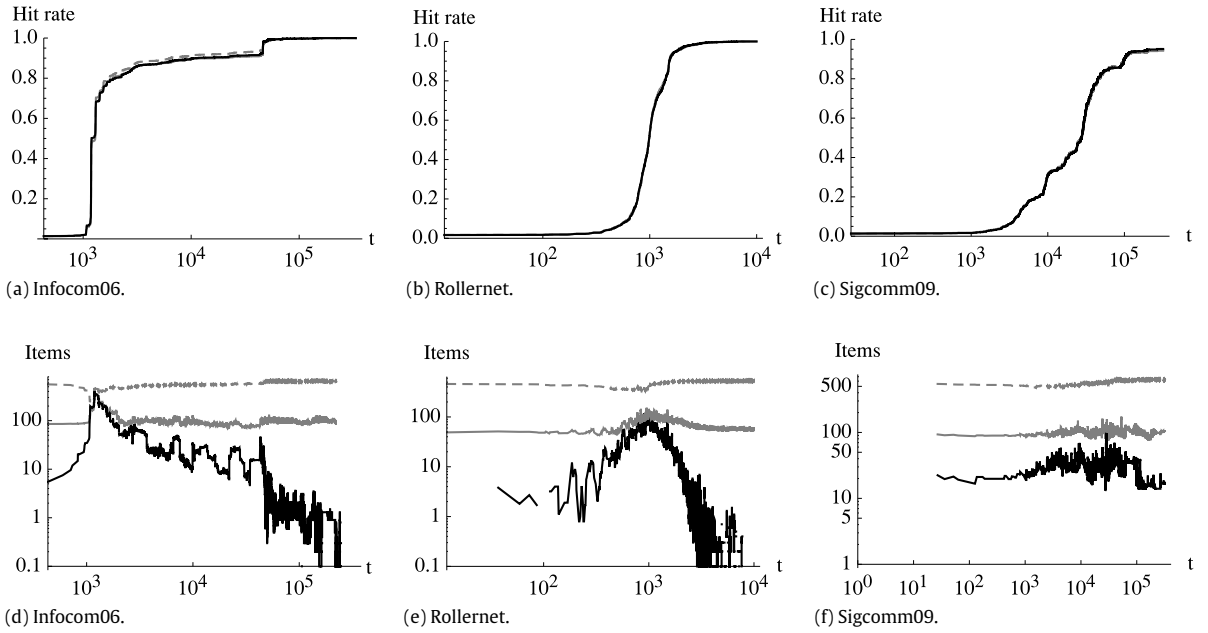
The *Infocom06* trace involves 78 iMotes equipped with Bluetooth interface and distributed amongst the participants to the IEEE INFOCOM 2006 Conference, plus other 20 iMotes placed in fixed sites. The scanning granularity in this case is around 2 min. The authors of [36] decided to take the contact duration so that two consecutive contacts are merged into one if they are separated by no more than an enquiry period (that is, contact loss is likely due to a message loss). For our experiments, we considered only contacts amongst the mobile devices.

The *SigComm09* trace involves 76 participants to the ACM SIGCOMM 2009 Conference. Contacts are detected through the Bluetooth interface of the participants' smartphones with a granularity of about 2 min. In this case, the available traces are unprocessed; we performed a symmetrisation consisting in both (i) taking a contact interval between two nodes A and B equal to the maximum time one of the two sees the other, and (ii) merging multiple subsequent contacts of, let us say, A with B into one if there exists just one overlapped contact of B with A.

In all cases, we dropped the contacts with devices external to the experiment. Table 7 shows the main characteristics of the traces. In spite of involving almost the same number of nodes, the three traces strongly differ. In SigComm09, 13.5% of the pairs of nodes do not observe any contact along the whole trace. As far as the other pairs are concerned, 6.28% of them just had one contact, so we computed the inter-contact time for the remaining pairs. In Rollernet, only 0.82% of the pairs never meet throughout the experiment, and 1.96% had just one contact. It is definitely the most dense trace and describes a very dynamic environment. In Infocom06, 0.65% of the pairs have no contact and 0.93% have just one contact. Both Infocom06 and SigComm09 have a low number of contacts per pair. Yet, contacts in Infocom06 are longer; this is revealed not only by the mean, but also by the maximum intra-contact time, which is of the order of one minute for both Rollernet and SigComm09, but amounts to roughly 45 min for Infocom06. The last two rows of Table 7 show the experimental setup. We measured the three algorithms with 8 channels, and generated 100 items per channel. Nodes choose the channel to subscribe according to a Zipf distribution (with parameter 1). As it is impossible to determine people relationships, we assumed that all nodes belong to the same community.

In order to have a fair comparison between the approaches, we carefully tuned the parameters of SR and SR* for each scenario. Presented results are mean values averaged on 10 runs. Moreover, confidence intervals at level 95% have been computed but not reported in graphs for the sake of readability.

Obtained results confirm what emerged in synthetic environments, that is, PR proves its adaptation abilities to different scenarios reaching the same dissemination performance of the best fine tuned competitors. Indeed, looking at Fig. 15(a–c) and Table 8 we notice that, beyond having almost the same hit rate trend, all the approaches reach the same dissemination level. In fact, with Infocom06 and Rollernet traces the maximum level of diffusion is achieved by all the cognitive approaches, while with the Sigcomm09 trace they reach a diffusion level in the range of 94%. However, we recall that in the Sigcomm09 scenario the 13.5% of pairs of nodes experience only one contact in the whole trace at most. Moreover, convergence times are in line with what presented in synthetic scenarios' results, PR is always faster or equal than the competitor approaches. We precise that, although in the Sigcomm09 scenario none of the approaches reaches the maximum coverage due to the underlying contact patterns, we decided to report convergence times for the sake of comparison. In this case they refer to the time instant when each algorithm reaches the 94% of coverage.



(a) Infocom06. (b) Rollernet. (c) Sigcomm09.

(d) Infocom06. (e) Rollernet. (f) Sigcomm09.

Fig. 15. First row: Hit rate curves for PR (solid black), SR (dashed grey) and SR* (solid grey). Second row: Network overhead comparison between PR (solid black), SR (dashed grey) and SR* (solid grey). Plots in the second row are in Log–Log scale for better readability. Parameter settings $\theta_C = 10$, $\theta_I = 25$, OC size = 10.

Table 8
Hit rates and convergence time of PR,SR and SR* for the three real world scenarios.

	Sigcomm09		Rollernet		Infocom06	
	Hit rate	*Conv. time* (s)	*Hit rate*	*Conv. time* (s)	*Hit rate*	*Conv. time* (s)
PR	94%	131 656	100%	3170	100%	50 670
SR	94%	163 048	100%	3390	100%	50 720
SR*	94%	199 147	100%	3410	100%	51 520

Efficiency properties of our solution are shown in Fig. 15(d–f) where, for each scenario PR proves to be more efficient than SR and SR* in terms of network overhead. Here also, previously obtained results in synthetic scenarios are confirmed. Indeed, PR obtains the same dissemination level of SR and SR* limiting the number of items exchanged and, most important, recognising when the item dissemination should be stopped. We point out that in the Sigcomm09 scenario nodes running PR do not stop the message exchange because the content diffusion process is not yet concluded at the end of the trace. Nevertheless, PR exploits contacts more efficiently than SR and SR*, as reported in Fig. 15(f). In conclusion, presented results on real traces confirm that with PR is possible to autonomously control the dissemination process limiting both the status information stored by mobile nodes and the network overhead without sacrificing neither efficiency nor latency.

6. Conclusion

In this paper, we investigated how the functional models of the human brain can be exploited to drive data dissemination in opportunistic networks. We started from the algorithm presented in [9,6] which for the first time proposes to use the recognition heuristic to develop an ICT solution for data dissemination. This solution has two main weaknesses. First, nodes maintain punctual status information about the diffusion of all data items leading to critical scalability problems, i.e. status information maintained by nodes linearly grows with the number of items in the system. Second, the strong dependence from fixed parameters to be fine tuned beforehand in every different scenario limits the real usability of this solution. In this paper we, solve the above problems by proposing for the first time a solution suitable for concrete implementation in opportunistic networks. Firstly, the use of an aggregate information about diffusion state of data items permits to limit the state maintained by nodes, without affecting the effectiveness of the data dissemination process. This feature makes the system much more scalable, and suitable for adoption in large scale environments. In particular, the state maintained with the algorithm proposed in this paper is constant with respect to the number of data items to be disseminated. Importantly, such an improvement in scalability is not paid with a significant reduction of the performance, as nodes are still able to receive what they are interested in within a similar amount of time. Second, the proposed approach uses a probabilistic approach (again derived from cognitive models through a Bayesian framework) to determine the relevance and usefulness of data items to be replicated. The probabilistic mechanism has the advantage of being completely autonomous and scenario

independent. It does not need a priori tuning of its parameters, which are automatically and continuously updated as long as nodes discover new information about the specific scenario. This feature gives to our solution a great flexibility and lets it easily adapt to very different scenario configurations irrespective of the large number of factors – e.g. cache size, network size, amount and frequency of appearance of new information in the system – affecting the behaviour of more static solutions. Third, despite the fact of relying on compressed state information and probabilistically taken decisions, our solution is able to generate very low network overhead with the remarkable advantage of being more energy efficient.

Appendix. Equivalence between $\tilde{\pi}_t$ and $\hat{\pi}_t$ diffusion indexes

As stated in Section 4, under a reasonable (for our context) assumption further specified, the Eqs. (9) and (11) are equivalent within a scale factor. Hereafter we present the passages to prove this claim expressed through the following relation:

$$\tilde{\pi}_t^{ch} \equiv \hat{\pi}^{ch}. \tag{A.1}$$

Let us begin by expanding the left hand side of (A.1):

$$\tilde{\pi}_t^{ch} = \frac{a_t}{a_t + b_t} \tag{A.2}$$

$$= \frac{\alpha a_{t-1} + y_t}{\alpha a_{t-1} + y_t + \alpha b_{t-1} + n_t - y_t} \tag{A.3}$$

$$= \frac{\alpha a_{t-1} + y_t}{\alpha a_{t-1} + \alpha b_{t-1} + n_t} \tag{A.4}$$

$$= \frac{\alpha a_{t-1} + y_t}{\alpha(a_{t-1} + b_{t-1}) + n_t}. \tag{A.5}$$

From now on we will assume the number of items belonging to the channel *ch* that a node can see during contacts as constant; let us denote it with n. Although it represents a simplification, our simulation results prove that our system is robust w.r.t. this assumption. Thus, Eq. (A.5) becomes

$$\tilde{\pi}_t^{ch} = \frac{\alpha a_{t-1} + y_t}{(\alpha + 1)n}. \tag{A.6}$$

We can rewrite (A.6) as a series of the following form:

$$\tilde{\pi}_t^{ch} = \frac{1}{n(1+\alpha)} \sum_{j=0}^{t} \alpha^{t-j} y_j. \tag{A.7}$$

Let us now consider the expansion of $\hat{\pi}_t$, where we make the same assumption as before according to which a node sees a fixed number n of data items for a channel during each contact:

$$\hat{\pi}_t = \alpha \hat{\pi}_{t-1}^{ch} + (1-\alpha)\frac{y_t}{n} \tag{A.8}$$

$$= \alpha^t (1-\alpha)\frac{y_0}{n} + \alpha^{t-1}(1-\alpha)\frac{y_1}{n} + \cdots + \alpha^0(1-\alpha)\frac{y_t}{n} \tag{A.9}$$

$$= \sum_{j=0}^{t} \alpha^{t-j}(1-\alpha)\frac{y_j}{n}. \tag{A.10}$$

Now, by applying a scale factor to Eq. (A.10) we obtain Eq. (A.7):

$$\frac{1}{(1-\alpha)(1+\alpha)} \sum_{j=0}^{t} \alpha^{t-j}(1-\alpha)\frac{y_j}{n} = \frac{1}{n(1+\alpha)} \sum_{j=0}^{t} \alpha^{t-j} y_t = \tilde{\pi}_t^{ch}. \tag{A.11}$$

Concluding, from Eq. (A.11) we see that the two indexes are equivalent within a scale factor

$$1 - \alpha^2.$$

An empirical example about the convergence of $\hat{\pi}^{ch}$ and $\tilde{\pi}^{ch}$ is reported in Fig. 1.

References

[1] M. Conti, S. Chong, S. Fdida, W. Jia, H. Karl, Y.-D. Lin, P. Mähönen, M. Maier, R. Molva, S. Uhlig, M. Zukerman, Research challenges towards the future Internet, Comput. Commun. 34 (18) (2011) 2115–2134. http://dx.doi.org/10.1016/j.comcom.2011.09.001. URL: http://www.sciencedirect.com/science/article/pii/S0140366411002714.

[2] M. Conti, S.K. Das, C. Bisdikian, M. Kumar, L.M. Ni, A. Passarella, G. Roussos, G. Tröster, G. Tsudik, F. Zambonelli, Looking ahead in pervasive computing: challenges and opportunities in the era of cyber-physical convergence, Pervasive Mob. Comput. 8 (1) (2012) 2–21. http://dx.doi.org/10.1016/j.pmcj.2011.10.001. URL: http://www.sciencedirect.com/science/article/pii/S1574119211001271.

[3] Cisco, Cisco visual networking index: global mobile data traffic forecast update, 2012–2017. http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.pdf (2012).
[4] L. Pelusi, A. Passarella, M. Conti, Opportunistic networking: data forwarding in disconnected mobile ad hoc networks, IEEE Commun. Mag. 44 (11) (2006) 134–141. http://dx.doi.org/10.1109/MCOM.2006.248176.
[5] A. Balasubramanian, B. Levine, A. Venkataramani, DTN routing as a resource allocation problem, SIGCOMM Comput. Commun. Rev. 37 (4) (2007) 373–384. http://dx.doi.org/10.1145/1282427.1282422.
[6] M. Conti, M. Mordacchini, A. Passarella, Design and performance evaluation of data dissemination systems for opportunistic networks based on cognitive heuristics, ACM Trans. Auton. Adapt. Syst. 8 (3) (2013) 12:1–12:32. http://dx.doi.org/10.1145/2518017.2518018.
[7] D. Goldstein, G. Gigerenzer, Models of ecological rationality: the recognition heuristic, Psychol. Rev. 109 (1) (2002) 75–90.
[8] G. Gigerenzer, D. Goldstein, Reasoning the fast and frugal way: models of bounded rationality, Psychol. Rev. 103 (9) (1996) 650–669.
[9] M. Conti, M. Mordacchini, A. Passarella, Data dissemination in opportunistic networks using cognitive heuristics, in: IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM, 2011, pp. 1–6. http://dx.doi.org/10.1109/WoWMoM.2011.5986145.
[10] L. Valerio, M. Conti, E. Pagani, A. Passarella, Autonomic cognitive-based data dissemination in opportunistic networks, in: 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks", IEEE, 2013, pp. 1–9.
[11] T.R. Krynski, J.B. Tenenbaum, The role of causality in judgments under uncertainty, J. Exp. Psychol. Gen. 136 (3) (2007) 430–450.
[12] C. Peterson, L. Beach, Man as an intuitive statistician, Psychol. Bull. 68 (1967) 29–47.
[13] G. Gigerenzer, U. Hoffrage, How to improve Bayesian reasoning without instruction: frequency formats, Psychol. Rev. 102 (1995) 684–704.
[14] J. Marewski, W. Gaissmaier, G. Gigerenzer, Good judgments do not require complex cognition, Cogn. Process. 11 (2) (2010) 103–121. http://dx.doi.org/10.1007/s10339-009-0337-0.
[15] M. Monti, L. Martignon, G. Gigerenzer, N. Berg, The impact of simplicity on financial decision-making, in: Proceedings of CogSci, 2009, pp. 1846–1851.
[16] D. Goldstein, G. Gigerenzer, Fast and frugal forecasting, Int. J. Forecast. 25 (2009) 760–772.
[17] J. Marewski, W. Gaissmaier, L. Schooler, D. Goldstein, G. Gigerenzer, From recognition to decisions: extending and testing recognition-based models for multialternative inference, Psychon. Bull. Rev. 17 (3) (2010) 287–309. http://dx.doi.org/10.3758/PBR.17.3.287.
[18] N. Chater, M. Oaksford, The Probabilistic Mind: Prospects for a Bayesian Cognitive Science No. 1, in: The Probabilistic Mind: Prospects for a Bayesian Cognitive Science, Oxford University Press, 2008.
[19] C. Boldrini, A. Passarella, Data Dissemination in Opportunistic Networks, John Wiley & Sons, Inc., 2013, pp. 453–490. http://dx.doi.org/10.1002/9781118511305.ch12.
[20] F. De Pellegrini, I. Carreras, D. Miorandi, I. Chlamtac, C. Moiso, R-P2P: a data centric DTN middleware with interconnected throwboxes, in: Proceedings of the 2nd International Conference on Autonomic Computing and Communication Systems, Autonomics'08, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, 2008, pp. 2:1–2:10. URL: http://dl.acm.org/citation.cfm?id=1487652.1487654.
[21] J. Whitbeck, M. Amorim, Y. Lopez, J. Leguay, V. Conan, Relieving the wireless infrastructure: when opportunistic networks meet guaranteed delays, in: 2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM, 2011, pp. 1–10. http://dx.doi.org/10.1109/WoWMoM.2011.5986466.
[22] V. Lenders, M. May, G. Karlsson, C. Wacha, Wireless ad hoc podcasting, SIGMOBILE Mob. Comput. Commun. Rev. 12 (1) (2008) 65–67. http://dx.doi.org/10.1145/1374512.1374535.
[23] C. Boldrini, M. Conti, A. Passarella, Context and resource awareness in opportunistic network data dissemination, in: Proceedings of the 2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks, WOWMOM'08, IEEE Computer Society, Washington, DC, USA, 2008, pp. 1–6. http://dx.doi.org/10.1109/WOWMOM.2008.4594890.
[24] C. Boldrini, M. Conti, A. Passarella, Design and performance evaluation of contentplace, a social-aware data dissemination system for opportunistic networks, Comput. Netw. 54 (4) (2010) 589–604. http://dx.doi.org/10.1016/j.comnet.2009.09.001. URL: http://www.sciencedirect.com/science/article/pii/S1389128609002783.
[25] C. Boldrini, M. Conti, A. Passarella, Contentplace: social-aware data dissemination in opportunistic networks, in: Proceedings of the 11th International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM'08, ACM, New York, NY, USA, 2008, pp. 203–210. http://dx.doi.org/10.1145/1454503.1454541.
[26] P. Pantazopoulos, I. Stavrakakis, A. Passarella, M. Conti, Efficient social-aware content placement in opportunistic networks, in: 2010 Seventh International Conference on Wireless On-Demand Network Systems and Services, WONS, 2010, pp. 17–24. http://dx.doi.org/10.1109/WONS.2010.5437139.
[27] E. Yoneki, P. Hui, S. Chan, J. Crowcroft, A socio-aware overlay for publish/subscribe communication in delay tolerant networks, in: Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems, MSWiM'07, ACM, New York, NY, USA, 2007, pp. 225–234. http://dx.doi.org/10.1145/1298126.1298166.
[28] P. Costa, C. Mascolo, M. Musolesi, G. Picco, Socially-aware routing for publish–subscribe in delay-tolerant mobile ad hoc networks, IEEE J. Sel. Areas Commun. 26 (5) (2008) 748–760. http://dx.doi.org/10.1109/JSAC.2008.080602.
[29] J. Reich, A. Chaintreau, The age of impatience: optimal replication schemes for opportunistic networks, in: Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, CoNEXT'09, ACM, New York, NY, USA, 2009, pp. 85–96. http://dx.doi.org/10.1145/1658939.1658950.
[30] A.C. Harvey, C. Fernandes, Time series models for count or qualitative observations, J. Bus. Econom. Statist. 7 (4) (1989) 407–417.
[31] D. Fink, A compendium of conjugate priors, Tech. Rep., Environmental Statistics Group, Dept of Biology, Montana State University, 1997.
[32] P.S. Almeida, C. Baquero, N. Preguiça, D. Hutchison, Scalable bloom filters, Inform. Process. Lett. 101 (6) (2007) 255–261. http://dx.doi.org/10.1016/j.ipl.2006.10.007. URL: http://www.sciencedirect.com/science/article/pii/S0020019006003127.
[33] C. Boldrini, A. Passarella, HCMM: modelling spatial and temporal properties of human mobility driven by users' social relationships, Comput. Commun. 33 (9) (2010) 1056–1074. http://dx.doi.org/10.1016/j.comcom.2010.01.013. URL: http://www.sciencedirect.com/science/article/pii/S0140366410000514.
[34] L. Breslau, P. Cao, L. Fan, G. Phillips, S. Shenker, Web caching and zipf-like distributions: evidence and implications, in: INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, vol. 1, 1999, pp. 126–134 http://dx.doi.org/10.1109/INFCOM.1999.749260.
[35] P. Tournoux, J. Leguay, F. Benbadis, V. Conan, M. Dias de Amorim, J. Whitbeck, The accordion phenomenon: analysis, characterization, and impact on DTN routing, in: INFOCOM 2009, IEEE, 2009, pp. 1116–1124. http://dx.doi.org/10.1109/INFCOM.2009.5062024.
[36] P. Hui, People Are The Network: Experimental Design And Evaluation of Social-based Forwarding Algorithms, UCAM-CL-TR-713 713, Cambridge University, 2008.
[37] A.-K. Pietilainen, E. Oliver, J. LeBrun, G. Varghese, C. Diot, Mobiclique: middleware for mobile social networking, in: Proceedings of the 2Nd ACM Workshop on Online Social Networks, WOSN'09, ACM, New York, NY, USA, 2009, pp. 49–54. http://dx.doi.org/10.1145/1592665.1592678.