Autonomic Cognitive-based Data Dissemination in Opportunistic Networks

Lorenzo Valerio*, Marco Conti*, Elena Pagani† and Andrea Passarella*
**IIT-CNR, Pisa, Italy*
Email: {marco.conti,andrea.passarella,lorenzo.valerio}@iit.cnr.it
†*Computer Science Dept., Università degli Studi di Milano & IIT-CNR, Italy*
Email: pagani@di.unimi.it

***Abstract*—Opportunistic Networks (OppNets) offer a very volatile and dynamic networking environment. Several applications proposed for OppNets - such as social networking, emergency management, pervasive and urban sensing - involve the problem of sharing content amongst interested users. Despite the fact that nodes have limited resources, existing solutions for content sharing require that the nodes maintain and exchange large amount of status information, but this limits the system scalability. In order to cope with this problem, in this paper we present and evaluate a solution based on cognitive heuristics. Cognitive heuristics are *functional* models of the mental processes, studied in the cognitive psychology field. They describe the behavior of the brain when decisions have to be taken quickly, in spite of incomplete information. In our solution, nodes maintain an aggregated information built up from observations of the encountered nodes. The aggregate status and a probabilistic decision process is the basis on which nodes apply cognitive heuristics to decide how to disseminate content items upon meeting with each other. These two features allow the proposed solution to drastically limit the state kept by each node, and to dynamically adapt to both the dynamics of item diffusion and the dynamically changing node interests. The performance of our solution is evaluated through simulation and compared with other solutions in the literature.**

***Keywords*-opportunistic networks; content diffusion; cognitive heuristics;**

I. Introduction

Cognitive psychology studies the way the human brain works and reacts to external stimuli. Several studies show that the brain often perceives the observed events as binary sequences occurring over time [18] and use them to make decisions according to a *frequency-based* reasoning. In particular, *cognitive heuristics* are functional models of the mental processes [12], [13] on which the humans rely 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. Heuristics can thus be seen as methods used by the brain to quickly find a solution to a problem, when the exhaustive search of the optimal solution is impractical or infeasible. Cognitive heuristics have been applied in various fields, such as financial decision making [8], forecasting purchases [10], results of sport events [5], outcomes of political elections [9]. Usually, the solution supplied by heuristics well approximates the optimum.

The capability of heuristics to work in a fast and frugal way makes them an interesting approach to be adopted in OppNets.Opportunistic networks 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. Nevertheless, opportunistic networks support multi hop communication by temporarily storing messages at intermediate nodes, until the network reconfigures and better relays (with respect to the final destinations) become available. Due to the scarcity of resources, the impossibility of building a global system knowledge, and the possibly short time at disposal of the nodes – when a contact occurs – to exchange information and carry on data dissemination, using cognitive heuristics in such an environment looks in principle a sensible approach. 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 real world. In this sense nodes play their role of proxy.

Among the various cognitive heuristics, in this paper we consider in particular the recognition heuristic [12], [13]. In a single sentence, 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 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 recognized university name as more endowed.

In this work, we exploit the recognition heuristic for data dissemination in opportunistic networks. We assume a scenario characterized by the presence of contents – data items – organized in specific topics – 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 which 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 is already carried by (most of) the interested nodes. The work presented in [6] is a preliminary attempt at investigating this approach (see Section II for more details). The main focus of [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 particular, in [6] the recognition decisions were taken based on punctual information about the single data items, and by defining fixed parameters, that had to be tuned depending on the specific networking environment. These features can result in significant scalability problems, and when wrongly tuned in poor adaptation to dynamic environments. Clearly, they represent roadblocks for applying recognition heuristics in concrete cases. In this paper, while we share the overall idea of [6], we drastically modify the actual data dissemination algorithms to remove these roadblocks, by still keeping the benefit brought by the use of recognition heuristics. Specifically, we only 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 drastically reduce (with respect to [6]) state maintained by nodes to implement the data dissemination policies. Our results show that this reduction comes without scarifying 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 permits to avoid using fixed thresholds, and makes the system adaptive to dynamical conditions. More precisely, in this paper we show how the proposed algorithm efficiently reacts to dynamic scenarios where at a certain time nodes may change their interests about channels, or when completely new channels/items are injected in the running system.

II. Related Work

A. Content Distribution in OppNets

In the literature, some works appeared that consider the problem of content diffusion in mixed fixed/mobile networks. In [11], a hybrid infrastructure is considered where *throwboxes* – i.e. devices with both wired and wireless interface – communicate with one another and with the wireless nodes. Nodes upload held items when in the communication range with a throwbox, and possibly download items that satisfy local interests. A similar hybrid infrastructure is considered in [15]. In both proposals, caches are maintained in the nodes belonging to the wired infrastructure with usual cache replacement algorithms.

Several works deal with the problem of content distribution in pure OppNets. In the PodNet Project [14], a framework is considered similar to that of this work. 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. By contrast, in ContentPlace [1], nodes aim at filling their caches in order to maximize 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 communities of the local user, and the *availability*, i.e. the number of users in the communities already owning the item.

Some works consider a publish/subscribe framework. According to this, in [16] 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 [7], 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.
An extensive survey about content diffusion in OppNets can be found in [3].

B. Recognition heuristic in opportunistic networks

In [6], a preliminary version of the approach presented in this work is proposed. For the sake of self-containment, we summarize here its characteristics. The caching 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 respective 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. 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 diffusion 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 to which nodes are subject to (we provide a quantitative analysis of this point in Section IV). These characteristics harm the actual suitability of this approach for its successful application in real world scenarios.

III. Problem Statement and System Assumptions

We consider a system composed by N nodes. Nodes can subscribe to one or more *channels* of interests. We assume that there are K channels available. Every node can generate content *items*. Each item i is labeled 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. Yet, new channels may be created dynamically, nodes can subscribe to them, and items for them may start to appear.

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 **maximize coverage**, i.e., maximize 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 jeopardize the coverage.

IV. Probabilistic Recognition

In [6], punctual information for each item and channel are maintained in order to recognize their popularity. This leads to a non-negligible amount of memory used that limits the usability of this approach in real scenarios. In order to improve the previous approach and make it suitable for large scale scenarios, we have to reduce the amount of information a node maintains about its environment by minimizing the loss of accuracy in terms of acquired knowledge. In [6], the recognition thresholds for items and channels (θ_I and θ_C, respectively) have a different impact in terms of diffusion performance. The former plays a more important role because it regulates the replication level at which a data item is deemed as recognized and not disseminated further. Moreover, it is reasonable to think that the number of items in the system largely exceeds the number of channels. This means that, in terms of scalability, it is critical to reduce the overhead related to keep detailed information about items diffusion (while keeping detailed information about channels popularity is far less a concern). Hence, in this work we focus our efforts on the problem of minimizing the state information maintained about item diffusion, while leaving unchanged the recognition procedure for the channels. We reduce the state maintained at nodes by compressing the knowledge about items diffusion into an *aggregate* measure that lets 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 favor of other less diffused items. Let us focus on a generic node, and let S^{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 finally denote with $S^{ch}_{new} \subseteq S^{ch}$ those items that are definitely new w.r.t. the node experience, i.e. items that a node has never seen before. We define the measure of novelty a node observes upon the encounter e as:

$$N(t) = \frac{|S^{ch}_{new}|}{|S^{ch}|} \tag{1}$$

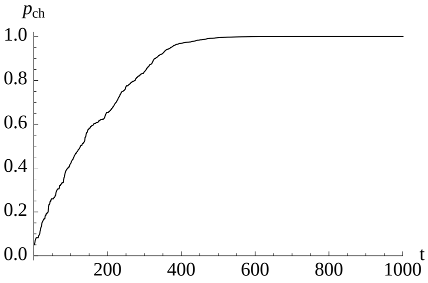


Figure 1. Increasing trend of p_{ch} during the system evolution

Informally,the idea behind probabilistic recognition is as follows. The more times a node receives almost the same kind of information, the stronger the belief that there is nothing more to know for that channel. Thus we are interested in the complement of (1):

$$1 - N(t) = 1 - \frac{|S^{ch}_{new}|}{|S^{ch}|} \tag{2}$$

Equation (2) measures the amount of novelty in the information received from an encountered node w.r.t. a given channel, that we use as an instantaneous indicator of the diffusion of the items in ch. Note that, as explained in detail in Section IV-B, S_{ch} and S^{new}_{ch} can be computed by keeping the state information maintained at nodes constant, irrespective of the number of data items in the system. Let $p_{ch}(t)$ be the estimated degree of diffusion of the items in ch, at the time t. We aggregate the instantaneous information collected during encounters with nodes in a unique index, as follows (assuming that t is a discrete variable incremented at each encounter):

$$p_{ch}(t) = \alpha * p_{ch}(t-1) + (1-\alpha) * (1 - N(t)) \tag{3}$$

where $0 \leq \alpha \leq 1$ regulates the balancing between the past experience and new information. Figure 1 shows the typical trend of p_{ch} we have observed in our simulation (details on the simulation settings are provided in Section V). 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. The index defined by (3) is used to determine when items of channel *ch* are recognized, as described in detail in the following sections.

A. Preliminaries on the Stochastic Mechanism

In order to autonomically recognize the items diffusion, nodes exploit the diffusion probability defined in (3). More precisely, for every known channel ch a node deems the corresponding items as *diffused* or *not diffused*, according to a Bernoulli trial with parameter $p_{ch}(t)$:

$$B(p_{ch}(t)) = \begin{cases} 1 & \Rightarrow \quad \text{Items are diffused} \\ 0 & \Rightarrow \quad \text{Items are not diffused} \end{cases} \tag{4}$$

In this way, as long as a node does not receive any new information about a channel ch, the corresponding value of p_{ch} (one for each channel and different for each node) gets increasingly close to 1, straightening 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.

B. Resulting Algorithm

In this section, we present how the described approach can be practically implemented in order to fuse the recognition heuristic with the probabilistic approach and exploit it 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 [6], and is reported also here for the 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 meetings.
- Channel Hash (CH): a Bloom filter, used to remember recognized 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 for redistribution

Step 1. For every contact between two nodes, each of them increments the counters associated to the other node's subscribed channels until a given threshold θ_C is reached, after that the channel is marked as recognized. If the number of entries in CC exceeds the maximum capacity, then the oldest entry is dropped. In this case, if it was marked as *recognized*, 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 BF), and channels that have been replaced. Once concluded the recognition phase for channel popularities, the second step begins.

Step 2. We will now refer to Algorithm 1. Upon a meeting, 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 same channel (line 9). By querying a Bloom Filter (IH) that contains 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 equation (3). 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, in a slight under estimation of the number of new items, and thus in stopping the diffusion process too early. Or simulation results show that this has, in practice, no impact 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 recognized or not, according to a Bernoulli trial with probability p_{ch} (lines 20–25). 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 [17]. 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 hereafter show that using fixed size Bloom filters have 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^2B algorithm to select the less spread items to be stored for redistribution. Differently from the previous version in [6], MT^2B does not fill the OC by selecting directly between the less spread items but by selecting between those items that belong to the less diffused channels. If the current OC capacity would not be enough to store all the items that could possibly be selected for further dissemination, the MT^2B sorts the items by their p_{ch} value and fills up the OC with the first n items according to its capacity.

Thanks to this approach nodes have to maintain less state information than the one maintained in [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 novel approach, every node has to keep only the state information about channels, thus the memory requirement has an order of magnitude of $O(K)$ because it grows linearly with the number of channels . By contrast in [6] every node has to maintain state information for both channels and items, which means that the order of magnitude in terms memory is $O(K * I)$. The improvement is very significant, as in real scenarios $I >> K$.

V. Performance evaluation

Hereafter, we evaluate the performance of the *Probabilistic Recognition* through a series of experiments by which we show that the proposed solution autonomically converges to or outperforms the results of the best fine-tuned configuration of the algorithm proposed in [6] with a significant reduction of resource consumption.

A. Simulated Environment

Nodes mobility is simulated according to HCMM [2], a mobility model that integrates temporal, social and spatial notions in order to obtain an accurate representation of real user movements. Nodes move in a 6×6 grid corresponding to a $1000 \times 1000m$ square, and are grouped in very compact 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 *travelers*, that 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 between communities. In our scenarios we have as many channels of interest as groups. For each group, all the channels are present with different popularity degrees and assigned to the nodes according to a Zipf distribution [4] with parameter

Algorithm 1 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 $p_{ch}$ be the diffusion probability of the items that belongs to the channel $ch$
Let $B(p_{ch})$ 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
    $p_{ch} \leftarrow \alpha * p_{ch} + (1 - \alpha) * (1 - \frac{I_{ch}}{C_{ch}})$
    if $B(p_{ch}) = 1$ then
        Mark items of $ch$ as *diffused*
    else
        Mark items of $ch$ as *not diffused*
    end if
end for
```

1. Moreover, for each community there is a different most popular channel. This makes the scenario uniform as far as channel popularity is concerned, as 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 assigned to nodes according to a uniform random distribution. The detailed scenario configurations can be found in Table I.

Paramenter	Value
Node speed	Uniform in $[1, 1.86m/s]$
Transmission range	$20m$
Simulation Area	$1000 \times 1000m$
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	$25000s$

Table I
DETAILED SCENARIO CONFIGURATION

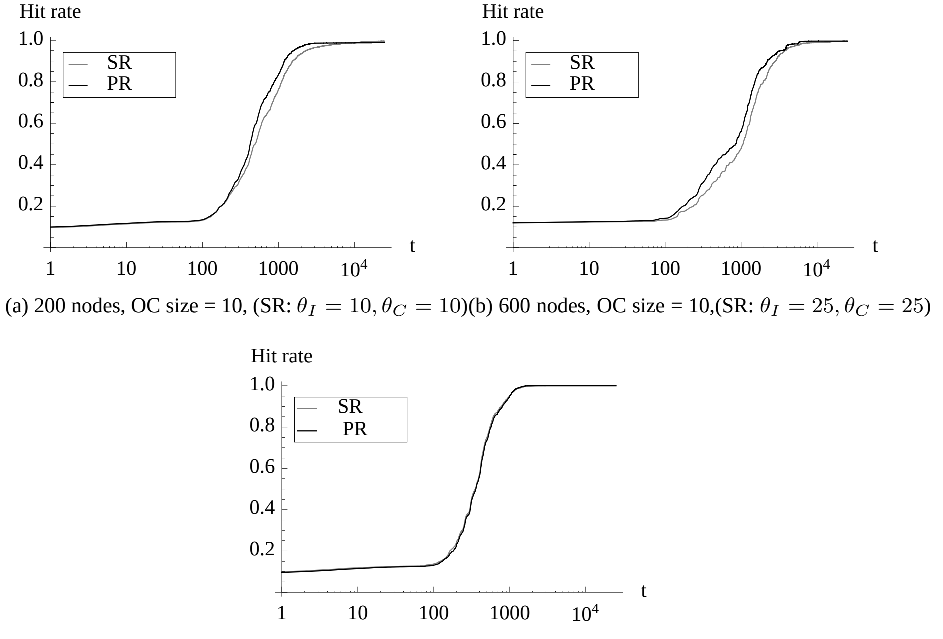
B. Simulation Results

For the sake of simplicity, from now on the acronyms *PR* and *SR* will refer to the Probabilistic Recognition case and Static Recognition, i.e. the algorithm presented in [6], respectively. All the results presented in this paper are mean values obtained on 10 runs where the initial configuration of items and channels were randomly reinitialized. 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 actually 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.

Let us recall that to regulate the dissemination process SR relies on static recognition thresholds, thus in order to have a fair comparison, we fine tuned SR parameters for every scenario. With PR, the nodes exploit the local information they receive from the surrounding environment to build up their own representation about the diffusion process that they use to decide which items are more profitable for redistribution. This kind of awareness has a great impact when the OC size is small. Indeed Figure 2a 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 same behavior holds for a more crowded network also: Figure 2b 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, the two third of nodes are completely unaware about the contents actually present in the scenario. However, also in this case the autonomic approach is able to quickly adapt to the situation reaching complete coverage faster that SR. By contrast, the two approaches become equivalent when the OC size is sufficiently large (OC size 50) to make the item selection a less critical task, as shown in Figure 2c. To have a quantitative understanding about convergence velocity we measure the converge times of the two approaches, which are in Table II. As we can see the probabilistic approach outperforms SR without relying on any parameters' fine tuning. Compared to SR the probabilistic approach is less demanding in terms of resource consumption. Figures 3a,3b,3c

Experiment	PR	SR
Net. Size 200, OC size 10	$2100s$	$3800s$
Net. Size 600, OC size 10	$4400s$	$5800s$
Net. Size 200, OC size 50	$1200s$	$1200s$

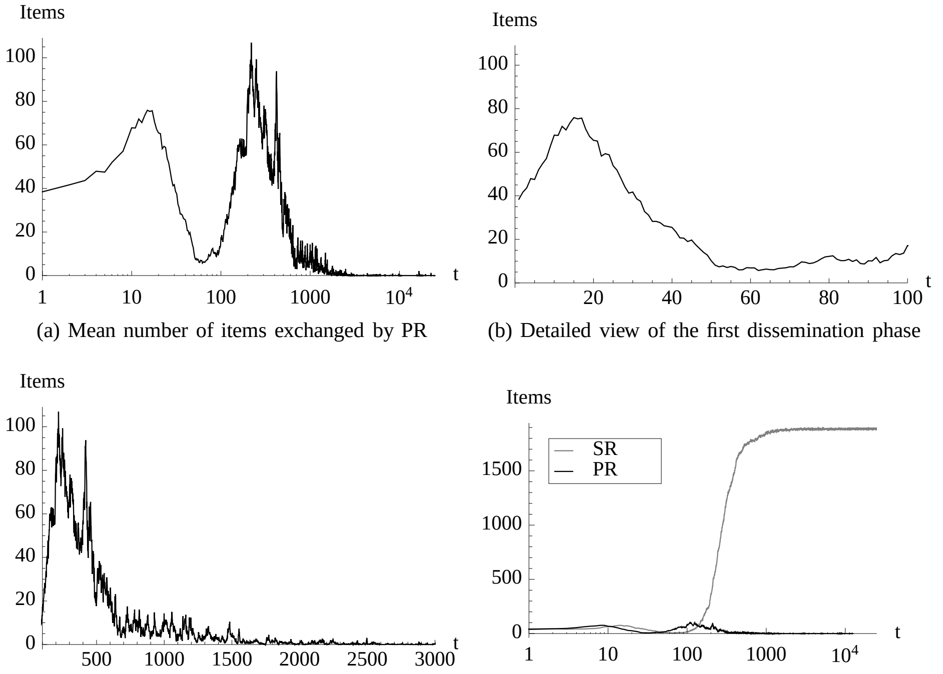
Table II
CONVERGENCE TIME FOR A COVERAGE $\geq 99\%$



(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: $\theta_I = 10, \theta_C = 10$)

Figure 2. Hitrate trends of PR (black curve) and SR (gray curve) with different network size 200 (a)-(c) and 600 (b).



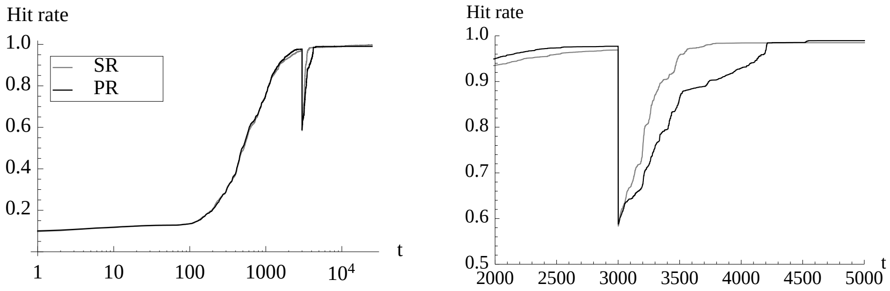
(a) Mean number of items exchanged by PR

(b) Detailed view of the first dissemination phase

(c) Detailed view of the second dissemination phase (d) Comparison between SR and PR network overhead

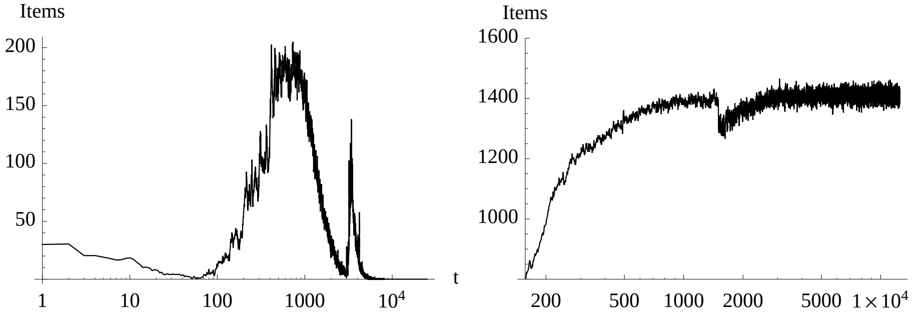
Figure 3. Mean number of items exchanged on a network of size 200

give, at different scales, an insight of 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 (Figure 3b) refers to the dissemination process inside groups before the arrival of the travellers in the community. After the 65-th second of simulated time, the dissemination process restarts due to the presence of travelers inside the community as depicted by the second phase of the process in Figure 3c. Interestingly, after some time both phases show a decrease in the number



(a) (b)

Figure 4. Hit rate trend of PR (black curve) and SR (gray curve) after a channel injection at $3000s$



(a) PR (b) SR

Figure 5. Mean number of items exchanged in a network of 200 nodes. Channel injection at $3000s$.

of exchanged items that 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, in order to maximize the convergence velocity in SR, the data exchange never stops even when all the items are deemed as recognized (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 Figure 3d.

Now we want to 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. A randomly assigned popularity is assigned to the new channel, a random set of nodes (of equal cardinality for each group) is chosen to change their current subscription in favor of the new channel injected. Due to this change, these nodes must clean their SC just after having run the MT^2B algorithm to load in OC possible useful items. At this point the usual probabilistic recognition approach starts to be applied also to the new channel. From Figure 4a we can notice that, in a scenario of 200 nodes, after the channel injection at $3000s$ both PR and SR react to the new stimulus, though with different intensity. SR seems to be more more responsive, but let us remember that it has been fine tuned to obtain this result. By contrast PR autonomically responds to the channel injection restoring the hit rate trend just after $1000s$. This proves that PR well approximate the behaviour of SR that, due to its fine tuning, represents an the upper bound for this scenario. Figure 4b shows in more detail this behaviour. Moreover, in Figure 5 we can see what happens to the network load when the dissemination process restarts due to the injection of a new channel both for PR and SR.

Finally, as anticipated before, we present the results of a sensitiveness analysis to evaluate the robustness of our approach in presence of an even less reliable diffusion information about channels. Thus we devised 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 III where we reported both the maximum coverage obtained and the corresponding convergence time. These experiment 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).

B.F. size reduction	**100%**	**80%**	**60%**	**40%**
Hit rate	$\geq 99\%$	97%	98%	98%
Conv. time	$2100s$	$4000s$	$10400s$	$16300s$

Table III
SENSITIVITY ANALYSIS WITH REDUCED BLOOM FILTER SIZE.

VI. Conclusion

This paper exploits the very recent idea of using functional models of the human brain's cognitive processes to drive data dissemination in opportunistic networks. Initial work in this area [6] has exploited the recognition heuristic (a very well established model in the cognitive psychology field) to design an algorithm whereby nodes, upon contacts, recognise (i.e., quickly determine) what data items available on the encountered node they should fetch to help their dissemination. In [6] the main focus was on demonstrating the general viability of this idea, but the proposed algorithm suffers from significant scalability problems, and must be fine tuned to obtain optimal results. In this paper we solve the above problems, by proposing for the first time a solution suitable for concrete implementation in opportunistic networks. Firstly, in this paper the decisions taken by nodes are based on aggregate information about data items, and do not require that they keep state information for each and every single data item available in the network. Using aggregate information drastically reduces the state maintained by nodes, 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 available in the environment, while the approach in [6] the state maintained by each node grows linearly with the number of data items. Importantly, such an improvement in scalability is not paid with a significant reduction of the performance of the data dissemination process, as nodes are still able to receive what they are interested into within a similar amount of time. Second, in the proposed algorithm nodes use a probabilistic approach to determine the relevance of data items and the usefulness of further replicating them. This provides two key advantages. On the one hand, the proposed algorithm does not need a priori tuning of its parameters to match the characteristics of the environment where it operates, but it is able to dynamically learn the correct behaviour and adapt it where the environment changes (e.g., new types of data are injected). Second, even in static conditions it exploits the probabilistic characteristics to ”change” - once in a while - the behaviour learnt by monitoring the environment conditions, and it is thus able to explore new, and possibly better, configurations.

Acknowledgment

This work is funded partially by the EC under the FET-AWARENESS RECOGNITION Project, grant 257756, FIRE EINS (FP7-288021) and partially by the Italian Ministry of Education, University and Research under the PRIN PEOPLENET (2009BZM837) Project.

References

[1] C. Boldrini, M. Conti, A. Passarella, *Design and performance evaluation of contentplace, a social-aware data dissemination system for opportunistic networks*, Comput. Netw. 54, 589-604, 2010.

[2] C. Boldrini and A. Passarella. *Hcmm: Modelling spatial and temporal properties of human mobility driven by users social relationships*, Comput. Commun. 33, 1056-1074. 2010.

[3] C. Boldrini, A. Passarella, *Data Dissemination in Opportunistic Networks*, Ch. 12 of Mobile Ad hoc networking: the cutting edge directions, Eds. S. Basagni, M. Conti, S. Giordano, I. Stojmenovic, Wiley, 2012.

[4] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, *Web Caching and Zipf-like Distributions: Evidence and Implications*, Proc. IEEE INFOCOM, 1999.

[5] D.G. Goldstein and G. Gigerenzer. *Fast and frugal forecasting.* Int. Journal of Forecasting 25, 760-772. 2009.

[6] M. Conti, M. Mordacchini, and A. Passarella, *Data Dissemination in Opportunistic Networks using Cognitive Heuristics*, Proc. IEEE WoWMoM Workshop on Autonomic and Opportunistic Computing (AOC), 2011.

[7] P. Costa, C. Mascolo, M. Musolesi, G.P. Picco, *Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks*, IEEE Journal on Selected Areas in Communications, vol.26, no.5, 748-760, June 2008.

[8] J. Marewski, W. Gaissmaier, and G. Gigerenze. *Good judgments do not require complex cognition.* Cognitive Process 11, 103-121. 2010.

[9] J. N. Marewski, G. Gaissmaier, L. J. Schooler,D. G. Goldstein, and G Gigerenzer. *From recognition to decisions: Extending and testing recognition-based models for multialternative inference.* Psychonomic Bulletin & Review 17, 3, 287-309. 2010.

[10] M. Monti, L. Martignon, G. Gigerenzer, and N. Berg. *The impact of simplicity on financial decision-making.* In Proc. of CogSci 2009, July 29 - August 1 2009, Amsterdam, the Netherlands. The Cognitive Science Society, Inc., 1846-1851. 2009.

[11] F. De Pellegrini, I. Carreras, D. Miorandi, I. Chlamtac, C. Moiso, *R-P2P: a data centric DTN middleware with interconnected throwboxes*, Proc. Autonomics 2008, 1-10, 2008.

[12] G. Gigerenzer , D.G. Goldstein, *Models of ecological rationality: The recognition heuristic*, Psychological Review, 109(1):75-90, 2002.

[13] D.G. Goldstein, G. Gigerenzer, *Reasoning the fast and frugal way: Models of bounded rationality*, Psychological Review, 103(4):650-669, 1996.

[14] V. Lenders, M. May, G. Karlsson, C. Wacha, *Wireless ad hoc podcasting*, SIGMOBILE Mob. Comput. Commun. Rev. 12, 65-67, 2008.

[15] J. Whitbeck, M. Amorim, Y. Lopez, J. Leguay, V. Conan, *Relieving the wireless infrastructure: When opportunistic networks meet guaranteed delays*, Proc.IEEE WoWMoM 2011, 1-10.

[16] E. Yoneki, P. Hui, S. Chan, J. Crowcroft, *A socio-aware overlay for publish/subscribe communication in delay tolerant networks*, Proc. MSWiM, 225-234, 2007.

[17] P.S. Almeida, C. Baquero, N. Preguiça, D. Hutchison, *Scalable Bloom Filters*, Information Processing Letterns, vol. 101, no. 6, 255-261, 2007.

[18] A.T. Oskarsson, L. Van Boven, G. McClelland, R. Hastie, *What's Next? Judging Sequences of Binary Events*, Psychological Bulletin, Vol.135, pp. 262285, 2008 .

[19] S. Serwe, and C. Frings. *Who will win wimbledon? the recognition heuristic in predicting sports events.* J. Behav. Dec. Making 19, 4, 321-332. 2006