Analysis of cognitive heuristics for content dissemination in opportunistic networks

Raffaele Bruno, Marco Conti, Matteo Mordacchini and Andrea Passarella
IIT–CNR, Pisa, Italy
{r.bruno,m.conti,m.mordacchini,a.passarella}@iit.cnr.it

ABSTRACT

When faced with large amounts of data, human brains are able to swiftly react to stimuli and assert relevance of discovered information, even under uncertainty and partial knowledge. These efficient decision-making abilities rely on so-called *cognitive heuristics*, which are rapid, adaptive, lightweight yet very effective schemes used by the brain to solve complex problems. In a content-centric future Internet where users generate and disseminate large amounts of content through opportunistic networking techniques, individual nodes should exhibit those properties to support a scalable content dissemination system. We therefore study whether such cognitive heuristics can also be used in such a networking environment. To this end, in this paper we develop an analytical model that describes a content dissemination mechanism for opportunistic networks based on one such heuristics, known as the recognition heuristic. Our model takes into account the different popularities of content types, and highlights the impact of the shared memory contributed by individual nodes to make the dissemination process more efficient. Furthermore, our model allows us to investigate the performance of the dissemination process for very large number of nodes, which might be very difficult to carry out through a simulation-based study.

Categories and Subject Descriptors

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

General Terms

Algorithms, Design, Performance

Keywords

Opportunistic Networks, Cognitive heuristics, Data Dissemination, Analytical Model

1. INTRODUCTION

In the Future Internet scenario, the active participation of users in the production and diffusion of content, using mobile devices in connection with more traditional CDNs and P2P networks, will create a very large and dynamic information environment [14]. A considerable part of these data will also be very contextualized, i.e. relevant only at specific times and/or geographic areas, and of interest only for specific groups of users. Opportunistic networking techniques will thus become a very important complement to infrastructure-based networks supporting mobile users (such as cellular and WiMAX) in order to efficiently disseminate content to interested users [15]. As mobile devices carried by users will have an active role in the data dissemination process, and considering the large volume of dynamic information that will be produced, devices will be faced with the very challenging task of determining the relevance of discovered content and selecting the most interesting data for the user. In order to be effective, this task should be performed rapidly. Furthermore, devices will have to make their decisions using only a partial knowledge of the environment and of the available content.

One approach to address the aforementioned problems is to embed autonomic decision-making abilities into mobile devices. In this paper, we focus on a new (to the best of our knowledge) direction in the autonomic networking field, and we exploit results coming from the cognitive psychology area, by using models of how the human brain assesses the relevance of information under partial knowledge. In fact, the human brain is able to achieve effective decision-making results in front of partial information, and the presence of noisy data by exploiting cognitive processes known as *cognitive heuristics*. These heuristics are the key psychological tools that allow the brain to come up with a highly effective decision under conditions of uncertainty, limited time, knowledge, and computational capabilities. Moreover, they are adaptive tools that are able to refine their effectiveness through a continuous interaction with the environment. In the past, cognitive sciences were able to derive accurate computational models for those cognitive processes. One of the most simple and effective heuristics is known as the *recognition heuristic*. The role of this heuristic is to assert the relevance of items when choosing among a pair (or a set) of objects. More specifically, it works by assuming that given a pair of objects, if one is recognized (i.e. the brain is able to recall that it already "heard" about that object) and the other is not, the recognized object has a higher value with respect to a given (possibly unknown) evaluation criterion.

The simplicity of this heuristic, its effectiveness in decision making processes, and its adaptivity, make it a suitable candidate for embedding a human cognitive process into autonomic, self-aware ICT devices dealing with a content-centric Internet.

We have designed an algorithm for data dissemination in opportunistic networking environments that is inspired by the recognition heuristic. It allows us to simplify and limit the complexity of the data selection task in an opportunistic network, while maintaing a high effectiveness in the data dissemination process. The full specification of this algorithm and preliminary results based on simulations are shown in [2]. In this paper, we first briefly overview the main features of this algorithm. Then, in Section 5 we describe our analytical model of the content dissemination mechanism. This model takes into account the different popularities of content types, and highlights the impact of the shared memory contributed by individual nodes to make the dissemination process more efficient. In Section 6 we present a comparison between the model predictions and related simulation results, showing that the model is able to capture the transient and steady-state behaviors of the data dissemination process. In particular, the model is able to estimate the level of replication of data items in the network, and thus to predict the probability that users will receive the content they are interested in within a given time from the instant it was generated in the network. Furthermore, our model allows us to explore network scenarios with large numbers of users and data items, which are generally difficult to study using simulation tools. Finally, Section 2, 3, and 7 complete the paper by summarizing related work, describing the main features of the recognition heuristic, and drawing the main conclusions of this work, respectively.

2. RELATED WORK

Data dissemination algorithms have been proposed for diverse families of mobile networks. The techniques proposed in [18]are representative of a body of work focusing on caching strategies for well-connected MANETs. On the contrary, the focus of this paper is on networking environments where content dissemination is more challenging and such policies cannot be applied. The first work that investigated the problem of content dissemination in opportunistic networks was developed in the PodNet Project [9]. Each node in the PodNet system is subscribed to a channel and devote part of its memory space for storing data items belonging to the channel it is subscribed to, and part for supporting a collaborative exchange of information. More precisely, when two nodes meet they exchange their cached data items and use heuristics based on the popularity of data channels to decide which fetched data items to store. A main limitation of PodNet is that it does not exploit any social information about nodes. More advanced approaches exploit information about users social relationships to drive the content dissemination process [19, 3, 1]. Specifically, the work in [19] defines a pub/sub overlay over an opportunistic network, where some nodes act as brokers, dispatching relevant content toward the most interested other peers. These brokers are the most "socially-connected" nodes, i.e., those nodes that are expected to be most available and easily reachable in the network. SocialCast [3] proposes a first attempt to exploit social information in dissemination processes. This is also the goal of the work in [1], where, however, a more refined and complete approach is used, based on social-aware dissemination strategies. Content dissemination is driven by the social structure of the network of users. Nodes evaluates which data items to store taking into account the social utility of the items, i.e. how they are likely of interest to users the node has social relationships with (and which, therefore, are expected to be in touch in the near future).

With respect to these approaches, in this paper we take a completely new direction, by borrowing models of human cognitive processes coming from the cognitive psychology domain. As this approach is still totally unexplored, in this paper we limit the set of contextual information that we use to the very minimum, and, for example, we do not exploit information about users social structures. This allows us to obtain initial exploratory results about the feasibility of this novel approach.

3. THE RECOGNITION HEURISTIC (RH)

Cognitive heuristics can be defined as simple rules used by the brain for facing situations in which people have to act quickly, relying on a partial knowledge of all the problem variables, the evaluation criterion of the different possible choices is not known, and the problem itself may be ill-defined in such a way that traditional logic and probability theory are prevented to find the optimal solution. Then, cognitive heuristics are able to deal with difficult problems by answering simpler problems. Early works on cognitive heuristics put the accent on how these cognitive processes could lead to systematic errors and cognitive biases. In contrast with this vision, Gerd Gigerenzer and most of recent psychological literature focus the research attention on the ability of these "brain tools" to produce very accurate judgements and the environmental conditions under which they are more effcient. Although the general concept of heuristic is similar to that widely used in computer science, heuristics used by human cognitive processes are formalized in a different way by cognitive sciences.

One of the simplest cognitive heuristics, which has attracted a broad attention by cognitive science in the last decade, is the *recognition heuristic* [6, 4]. The recognition heuristic is based on a very simple rule. When evaluating a couple of objects, and one is *recognized* (i.e. the brain is able to recall that it already "heard" about that object) and the other is not, this heuristic infers that the recognized object has a higher value with respect to a given evaluation criterion. The recognition heuristic is effective when the recognition of objects is highly correlated with the ideal evaluation criterion that should be used to select among the possible choices if complete information would be available. In this case, the heuristic is said to be *ecologically rational.* In fact, the correlation is adaptively derived from the environment by exploiting the presence in it of some *mediators* that carry information (coded in variables) used by the heuristic itself to approximate the value of the objects with respect to the criterion.

To better understand how the recognition heuristic works, the foundational study on recognition heuristic [4] uses, as an example, the estimation of the university endowments. To illustrate this example we refer also to Fig. 1, which depicts the general elements involved in the recognition heuristic, and the relationships between them. The evaluation criterion to be used in the example is the value of the endowment. This information is generally not publicly avail-

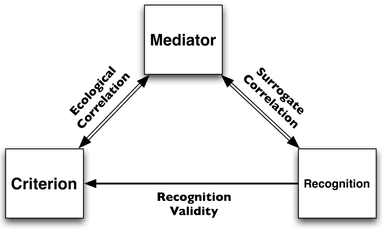

Figure 1: Ecological rationality of the recognition heuristic

able. Nevertheless, it is argued that newspapers could act as mediators, since they periodically publish news related to the most important universities. Thus, the number of times a university appears on the newspapers could be a strong indicator that it has larger endowments than universities that do not or rarely appear on the media. In other words, in this case the mediator variable related to the evaluation criterion is the number of citations. As said before, the correlation between mediators and the evaluation criterion is called *ecological correlation*. When a person has to choose which university has the biggest endowment between a couple of institution names, he uses the recognition heuristic and chooses a recognized name against an unknown one. Clearly, in this case, newspapers influence the recognition, since the more they cite an institution, the more likely that institution name will be remembered and, thus, recognized. Since the brain evaluates options exploiting the citations on newspapers instead of the real, unknown criterion, the relation between the recognition and the mediators is called *surrogate correlation*. From this example it is straightforward to notice that the effectiveness of the recognition heuristic, i.e. the *recognition validity*, is continuously reinforced by the stimuli received from the environment.

The simplicity of the recognition heuristic has made it a powerful tool in support of decision-making processes. As such, it has been successfully used in various fields [11], like financial decision-making processes [12], the forecast of future purchase activities [7], the results of sport events [17], or political election outcomes [10].

4. AN RH-BASED DATA DISSEMINATION SCHEME FOR OPPORTUNISTIC NETWORKS

In this study we consider a network scenario where mobile nodes generate data items and other nodes can be interested in those items. More precisely, users are interested in data channels, i.e. high-level topics to which the data items belong. The goal is to bring all the data items of a given channel to all the nodes that are interested in it. To this end, each node contributes a limited amount of storage space to help the dissemination process, since *contacts* between users are the only way to disseminate data items. Ideally, upon encountering another node, a node should select which data items to fetch and keep in its storage space, such that the total *utility* of its local storage space for the overall dissemination process is maximized. Solving this problem exactly would clearly require global knowledge and central coordination, which is unfeasible in opportunistic networks. Therefore, the algorithm we have originally proposed in [2] (and summarized in this section) exploits the recognition heuristic to efficiently approximate the utility of storing the data items that can be exchanged during a pair-wise contact between two nodes. Based on these approximated utility values, each node stores only the most useful data items, selected among both those already available locally and those available on the encountered node, until it's storage space is full.

With reference to both the description given in Sec. 3 and the elements reported in Fig. 1, we have that the *unknown criterion* to be estimated is then the utility of fetching an item, with respect to the global information dissemination process. Since we said that the only way to obtain information is through contacts between nodes, each peer considers the other nodes as the *environmental mediators* depicted in Fig. 1. Indeed, they are the only mean by which the information needed by the recognition heuristic can be collected.

About the information exchange process, assuming that each individual node is interested in only one channel, nodes exchange only minimal information when meeting: the channel they are interested into and a summary of the data items they currently store. At each encounter, this information is used to evaluate the utility of those data items during future meetings with other nodes. Essentially, a node estimates how many users are interested in the channels of discovered items and how those items are spread in the network. At a very high level, nodes decide which items to store based on two simple assumptions: *i*) the more other nodes request a given channel, the more its data items are relevant, *ii*) the less a data item is replicated in the network, the more useful could be to store a copy of that item. These assumptions bind the observed information carried by the *mediators* with the estimation of the inaccessible evaluation criterion, and, thus, constitute the *surrogate correlation* of the recognition heuristic.

Since in our algorithm there are two types of decisions to be taken to estimate the utility of the data items to fetch (i.e., how relevant a channel is, and how relevant a data item is) we use two distinct recognition heuristics to separately recognize channels and data items. Intuitively, a node recognizes a channel as soon as it becomes “enough popular”, i.e., as soon as that node encounters enough other nodes that are interested in the same channel. Hence, if a channel is recognized, it means that several users are interested in it and, thus, it is worth (for the overall community) to circulate its data items. Furthermore, a node recognizes a data item if it is “spread enough”, i.e., as soon as it is encountered on at least a given number of other nodes. If a data item is not recognized, it means that only a few users have a copy of it in their memory, so it should be replicated more broadly to increase its diffusion. On the basis of these considerations, a node decides to fetch a data item from the encountered node if: *a*) it recognizes the channel the data item belongs to, and *b*) is does not recognize the data item itself. Finally, to recognize a channel (or a data item) each node maintains a counter that count how many times it encountered a node that it is interested in that channel (stores that data item). Then a channel (data item) is recognized if this counter reaches a given recognition threshold. Furthermore, the recognition counters is decremented if the channel (data item) is not seen for a while. The exploitation of such “recognition thresholds” is consistent with the cog-

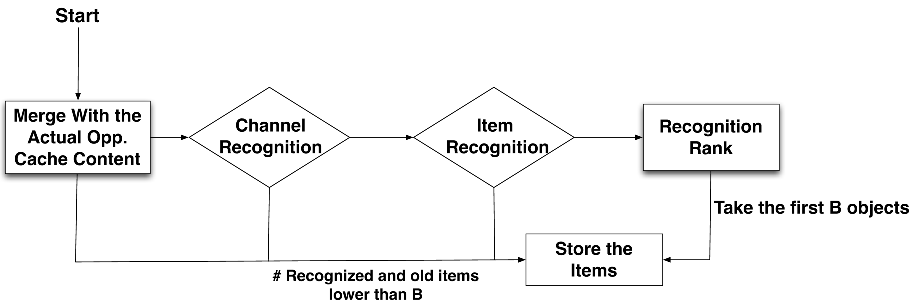

Figure 2: Modified *Take-the-Best* algorithm

nitive psychology research on how the recognition memory works in the brain (e.g. [16]).

As we have assumed that the shared storage space contributed by the nodes is limited, we have to define an algorithm that exploits the recognition level of channels and data items to select which items to keep (among the ones currently already available at the node or available on the encountered node) in case the storage space is insufficient to take all available data items. This algorithm thus works as a replacement algorithm for data items in the nodes storage space. To this end, we use a *modified Take-the-Best* algorithm [5], taken from the cognitive psychology literature. This algorithm ranks two (or more) options using several steps. On each step a heuristic is applied (e.g. the recognition heuristic), until it reaches the first (*best*) step where the heuristic can discriminate among the options. In our case, nodes decide which data item to store using a three-step version of the Take-the-Best algorithm. The first step applies the recognition heuristic to the channels, the second one uses the recognition of data items, while the last step ranks the items according to their recognition value, as explained below. The steps of the algorithm are shown in Fig. 2, where B denotes the size of the memory space dedicated to store data items belonging to channels the node is not interested in. More precisely, the data items available on an encountered node (hereafter called "new" items) are evaluated together with the ones already stored in the node (hereafter called "old" items). To this end, we first apply the recognition heuristic to the channels, i.e. all the items whose channel is not recognized are discarded. If there is enough space to store all the remaining items, nothing else happens. Otherwise, we apply the recognition heuristic to data items, i.e. non-recognized data items are dropped. In case the number of remaining items still exceed the size of the node cache, items are ranked according to their recognition level using the following two criteria. First of all, data items with lower recognition values (i.e., whose counter is lower) are more relevant that the ones with higher recognition levels, because the former ones are considered less widespread and, thus, more useful to fetch. Secondly, among the data items with the same recognition value, new ones are more relevant than old ones. Note that the heuristics applied in the subsequent steps are of increasing complexity. This is also a typical cognitive scheme, which tries to use the lower possible amount of resources to discriminate between possible choices.

A more detailed description of the recognition algorithms and of the modified Take-the-Best algorithm is given in [2].

4.1 Node Caches

To implement the above algorithm, each node makes use of several internal caches. Fig. 3 shows the internal architecture of the caches of a single node. With respect to this figure, we have two classes of caches: data caches and recognition caches. More precisely,

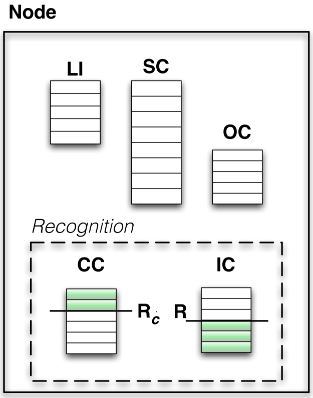

Figure 3: Node Architecture

Data caches:

- LI is the cache containing the *Local Items*, i.e. the items generated by the node itself
- SC is the *Subscribed Channel* cache, i.e. the cache containing the items belonging to the channel the node is interested to and obtained by encounters with other peers. Furthermore, we assume that nodes are able to reserve enough space for the channel they are interested in. Thus, the SC cache is assumed unlimited.
- OC is the *Opportunistic Cache*, i.e. the cache containing the objects obtained through exchanges with other nodes and belonging to channels the node is not interested in. This cache is the part of the node's storage space contributed for the overall efficiency of the data dissemination process, beyond the particular interest of the individual node. Therefore, we assume that it has a limited size. Its content consists of the items the node believes to be the most "useful" for a collaborative information dissemination process. They are selected using the values contained in the *Recognition* caches.

Recognition caches:

- CC is the *Channel Cache*: whenever the node meets another peer subscribed to a given channel, the channel ID is put in this cache, along with a counter. The counter is incremented every time a node subscribed to the same channel is found. Note that the storage space required to maintain this information is very limited compared with the size of data items. Thus, the size of this cache is assumed unlimited.

- IC is the *Item Cache*: similarly to the previous cache, when a new data item is seen in an encountered node, its ID is put in this cache, along with a counter. The counter is incremented every time the object is seen again. We assume that also this cache has no space limits.

Finally, we remind that the CC and IC caches make use of two different *recognition thresholds*, R_c and R, respectively. Items and channels whose counters exceed the value of the threshold are marked as recognized. This classification is used by the OC cache in order to handle its data, using the modified *Take-the-Best* algorithm described before.

5. ANALYTICAL MODEL

This section develops the analytical model describing the temporal evolution of the replication levels for data items belonging to a *tagged cannel c*. More precisely, in our model we use a set of embedded Markov chains to describe the status of the various caches deployed at the mobile nodes of the opportunistic network. Since the status of the caches of a node can change only when it exchanges data items upon encountering another node, the embedding points for our analysis are the time instants at which pairs of nodes meet.

Concerning the number of Markov chains needed in our model and their state space size, we use a distinct Markov chain for each channel to describe the evolution of the channel recognition level in the CC cache of a generic node. Similarly, we use a separate Markov chain for each channel to describe whether a generic data item of that channel is stored in the SC cache. For the IC and OC caches, we consider a generic data item for each channel, and a Markov chain describes (i) the evolution of the recognition level of that data item in IC; and (ii) whether the data item is stored in OC and in which position, respectively. It is important to observe that in this study we consider a homogeneous network scenario, where each node meets the other nodes in the network with equal probability at an encounter event[1]. Therefore, it is sufficient to derive the steady state probabilities for the aforementioned Markov chains by considering a single *tagged node*, since the same average behavior is observed in all the other nodes. In conclusion, our model requires to track the evolution of four Markov chains per channel. As channels (data types) can be reasonably assumed to be much less than data items, the complexity of the model is reasonable.

The goal pursued by our analysis is to derive the fraction of nodes in the network that, at time t, have a copy of a data item a belonging to channel c either in SC or OC, hereafter denoted as $r_t(a)$. In fact, the knowledge of $r_t(a)$ is enough to fully characterize the dissemination process of the data items in the opportunistic network. In the following sections, we will describe how the state probabilities of the Markov chains describing the cache status are used to derive $r_t(a)$ for each channel. Although the formal definition of the state spaces of these Markov chains is reported later, for the sake of presentation clarity in Tab. 1 we list the main notations used in our analysis.

Symbol	*Definition*
N	Number of nodes of the network
M	Number of items in the network
C	Number of channels in the network
$a.c$	Channel to which data item a belongs
$r_t(a)$	Fraction of nodes storing a copy of a
$Pop(c)$	Number of nodes subscribed to channel c
$\psi_a^t[1](\psi_a^t[0])$	Prob. that data item a is (is not) in SC at time t
$\nu_c^t[i]$	Prob. that channel c has a recognition level i in CC at time t
$\upsilon_a^t[i]$	Prob. that data item a has a recognition level i in IC at time t
$\varphi_a^t[i]$	Prob. that data item a is stored in OC at time t with a recognition level i
R_c	Threshold for channel recognition
R	Threshold for data item recognition

Table 1: List of mathematical notations

5.1 Subscribed Channel Cache

Let us consider a generic data item a belonging to the channel c to which the tagged node is subscribed. Then, in the Markov chain describing the status of the SC cache, state 1 is used to indicate that a copy of data item a is stored in SC, while state 0 indicates that SC does not hold a copy of a. Let us assume that at time $t+1$ the tagged node encounters another node, while the previous encounter event occurred at time t. Intuitively, the probability $p_{0,1}$ to move from state 0 to state 1 at time $t+1$, i.e., the probability that a copy of data item a is fetched by the tagged node from the node encountered at time $t+1$, is given by $p_{0,1} = r_t(a)$. On the other hand, the probability $p_{0,0}$ that a data item not present in SC at time t is also not present at time $t+1$ can be computed as $p_{0,0} = 1 - p_{0,1}$. Since we have assumed that the SC cache is large enough to contain all the data items of the channel the tagged node is subscribed to, a data item cannot be dropped by the SC cache. Formally, this implies that $p_{1,1} = 1$. Now, let $\psi_a^t = \{\psi_a^t[0], \psi_a^t[1]\}$ be the state vector of the Markov chain at time t, and let us denote with $P = \{p_{i,j}\}^2$, $i, j = 0, 1$, its transition matrix evaluated at time $t+1$. Then, we can compute the state vector at time $t+1$ as follows

$$\psi_a^{t+1} = \psi_a^t P \; . \tag{1}$$

The initialization distribution of the state vector is $\psi_a^0 = \{1, 0\}$, since the SC is empty at the beginning of the system.

5.2 Channel Recognition

As described in Sec. 4.1, CC stores the number of times each channel has been seen by the tagged node. Thus, the state i for the Markov chain used to describe the status of the CC for the tagged channel c represents its recognition level.

[1] This condition holds for the stationary regime of several popular mobility model, such as the Random Waypoint model [13].

[2] Note that the $p_{i,j}$ values depend on the parameter t, but for the sake of notation simplicity we omit it.

The state 0 represents the condition of channel c not yet observed. It is important to note that the probability for a channel to be recognized (i.e. to reach the recognition threshold) depends only on its popularity $Pop(c)$. Then, the transition probabilities of this time homogenous Markov chain are given by

$$p_{i,j} = \begin{cases} Pop(c)/N & \text{if } j=i+1, i \in [0, R_c-1] \\ \alpha^i(1-Pop(c)/N) & \text{if } j=i-1, i \in [1, R_c] \\ 1-(\frac{Pop(c)}{N}(1-\alpha^i)+\alpha^i) & \text{if } j=i, i \in [1, R_c-1] \\ 1-Pop(c)/N & \text{if } i,j=0 \\ 1-(1-Pop(c)/N)\alpha^i & \text{if } i,j=R_c \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

where $Pop(c)/N$ is the probability of encountering a node subscribed to channel c, and α $(0 < \alpha < 1)$ is a discount parameter. More precisely, if a channel is not observed in the CC cache of the encountered node, then the tagged node decreases its recognition level i by one with a probability that decreases exponentially as i increases. This approach avoids that a tagged node considers relevant a channel that is no more popular in the network. On the other and, the probabilities of remaining in the same recognition state after the encounter event are simply the complements of the total probabilities of moving to other states. Special cases are state $i=0$, where the recognition level cannot decrease, and state $i=R_c$, where the recognition level cannot increase.

Now let $\nu_c^t = \{\nu_c^t[0], \nu_c^t[1], \ldots, \nu_c^t[R_c]\}$ be the state vector at time t and $P = \{p_{i,j}\}$ the transition probability matrix. It holds that $\nu_c^{t+1} = \nu_c^t P$. Since at time t a node has no information on the channels, we have that the initial distribution is $\nu_c^0 = \{1, 0, \ldots, 0\}$.

5.3 Item Recognition

Conceptually, the recognition process of data items is similar to the channel recognition. In fact, IC maintains a list of the observed data items and their recognition levels, defined as the number of times they have been observed in encountered nodes. Thus, the state i of the Markov chain used to describe the status of the IC for the generic data item belonging to channel c represents its recognition level, while state 0 represents the condition of data item a not yet observed. Considering that the probability that a tagged node finds a copy of data item a in the caches of another node encountered at time t+1 depends only on its replication level $r_t(a)$, the transition probability matrix of this Markov chain can be calculated as

$$p_{i,j} = \begin{cases} r_t(a) & \text{if } j = i+1, i \in [0, R-1] \\ \gamma^i(1-r_t(a)) & \text{if } j = i-1, i \in [1, R] \\ 1-[r_t(a)(1-\gamma^i) + \gamma^i] & \text{if } j = i, i \in [1, R-1] \\ 1-r_t(a) & \text{if } i,j = 0 \\ 1-(1-r_t(a))\gamma^i & \text{if } i,j = R \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

where γ $(0<\gamma<1)$ is the discount factor for the item recognition level.

Now let $\upsilon_a^t = \{\upsilon_a^t[0], \upsilon_a^t[1], \ldots, \upsilon_a^t[R_c]\}$ the state vector at time t and $P = \{p_{i,j}\}$ the transition probability matrix. It holds that $\upsilon_a^{t+1} = \upsilon_a^t P$. Since at time t a node has not information on the data items, we have that the initial distribution is $\upsilon_a^0 = \{1, 0, \ldots, 0\}$.

5.4 Opportunistic Cache

Differently from the other caches, OC has a limited size B. As described in Sec. 4.1, only copies of data items belonging to recognized channels can be stored in the OC cache. However, given that OC can contain only a subset of available data items, a replacement policy is needed to drop less useful data items. We recall that our modified *Take-the-Best* replacement policy gives a higher priority to data items with lower recognition levels. In case of ties, the data items found in the caches of the encountered node are preferred to the ones already in the OC cache of the tagged node.

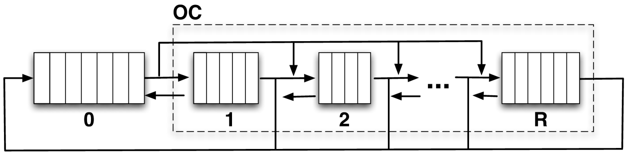

Figure 4: Queuing network modeling the OC cache.

To tackle the problem of modeling the evolution of OC, we represent this cache as a *queuing network*, as shown in Fig. 4. In this queuing network, a sub-queue i stores all data items that are in the OC and have the same recognition level i. The sub-queue 0 is a virtual queue that contains all the data items that are outside of the OC, independently of their recognition levels. Then, the ith element $(i>0)$ in the state vector $\varphi_a^t = \{\varphi_a^t[0], \varphi_a^t[1], \ldots, \varphi_a^t[R]\}$ represents the probability that the generic data item a is in the OC cache with recognition level i at time t. It is also important to note that each individual sub-queue has not a fixed size, but we must ensure that the sum of the numbers of data items stored in these sub-queues (excluding sub-queue 0) is lower or equal to B.

It is intuitive to note that modeling the replacement policy used to manage the OC cache is a hard task. To facilitate the analysis we split the problem into two simpler sub-problems. First of all, we model the reordering of the data items stored in the OC cache due to changes in their recognition levels after an encounter event. For instance a data item that was initially stored in sub-queue i should be moved to sub-queue $i+1$ if its recognition level is increased upon an encounter event. Since this process involves only an internal reordering of stored data items, no data items are dropped. The second step in the analysis takes into account that new data items fetched by the caches of the encountered node may enter the OC of the tagged node at a sub-queue that depends on their recognition level. Due to cache size constraints, some of these data items could not be allowed to enter the OC cache, or some data items already stored in OC could be removed to let new data items to enter the OC cache. In addition, in both steps we consider the possibility that data items in the OC are moved to sub-queue 0 – i.e. are dropped – if their channel is not recognized anymore after the encounter occurring at time $t+1$. In the following we separately describe these two modeling phases.

Step 1. Let us introduce an auxiliary Markov chain, whose state vector $\varphi_a' = \{\varphi_a'[0], \varphi_a'[1], \ldots, \varphi_a'[R]\}$ represents the probability that the generic data item a is in the OC cache with recognition level i *after* the encounter event, but *before* new data items are inserted in the OC. Then, we have that $\varphi_a' = \varphi_a^t P'$, where P' is the transition probability matrix of the auxiliary Markov chain modeling the data item reordering. Since no data items enter OC in this phase, we have that $p'_{0,0} = 1$. However, a data item could be removed from OC if the channel it belongs to loses its recognition, which happens with probability $(1 - \nu_a^{t+1}[R_c])$, or the recognition level of the data items becomes null. In other words

$$p'_{i,0} = 1 - \nu_a^{t+1}[R_c], i \in [2, R] \ , \tag{4}$$

while

$$p'_{1,0} = (1 - \nu_a^{t+1}[R_c]) + \nu_a^{t+1}[R_c]\gamma(1 - r_t(a)) \ . \tag{5}$$

Formula (5) can be explained by noting that a data item of a recognized channel can change its recognition level from one to zero only if it is not in the caches of the encountered node, and the tagged node applies the discount factor γ to the data-item recognition level stored in IC (see formula (3)). Following a similar line of reasoning as in (5) we can compute the probability that a data item already in the OC moves to either backwards or forwards sub-queues. More formally we have that

$$p'_{i,i-1} = \nu_a^{t+1}[R_c]\gamma^i(1 - r_t(a)), i \in [2, R] \ , \tag{6}$$

and

$$p'_{i,i+1} = \nu_a^{t+1}[R_c]r_t(a), i \in [1, R-1] \ . \tag{7}$$

On the other hand, the probability of remaining in the same sub-queue after the encounter event at time $t+1$ is simply given by the complement of the sum of the probabilities of moving backwards or forwards and to leave the OC cache. More formally, it hods that

$$p'_{1,1} = 1 - (p_{1,0} + p_{1,2}) \tag{8}$$

$$p'_{i,i} = 1 - (p_{i,0} + p_{i,i-1} + p_{i,i+1}) \, , i \in [2, R-1] \tag{9}$$

$$p'_{R,R} = 1 - (p_{R,0} + p_{R,R-1}) \tag{10}$$

We now exploit the knowledge of the state vector φ'_a to compute the new average number of data items in each sub-queue i, say B'_i, after the internal reordering, which is simply given by

$$B'_i = \sum_{a=1}^{M} \varphi'_a[i] \ . \tag{11}$$

Step 2: To derive the final status of the OC cache, i.e. $\varphi_a^{t+1}[i]$, in this step we first compute the average number $N_{0,i}$ of new data items that are eligible to enter OC at sub-queue i. Then, we compute the number F_i of available frees slot at each sub-queue i of the OC. The F_i value is the key parameter we need to compute the probability that a new data item is either discarded or cached, and the probability that an old data item stored in OC is removed.

To compute the $N_{0,i}$ quantity we should observe that a new data item not already stored in the OC of the tagged node is eligible for entering the OC cache in the sub queue i if and only if it is stored in the caches of the encountered node, if the channel it belongs to is recognized and its recognition level before the encounter event was $i-1$. Formally, this can be written as follows

$$N_{0,i} = \sum_{a=1}^{M} r_t(a)\nu_a^{t+1}[R_c]\upsilon_a^t[i-1]\varphi_a^t[0] \ . \tag{12}$$

It is important to remind that $N_{0,i}$ expresses the number of new data items that can be potentially copied in the OC cache. However, the actual number of new data items that are copied in OC will depend on the number of free slots. More precisely, let us denote with F_i the *maximum* number of free slots that new data items can occupy in the sub-queue i of the OC cache. It holds that

$$F_i = B - \sum_{j=1}^{i-1} B_j^{t+1} \ , \tag{13}$$

with $F_1 = B$. Indeed, data items (both new or old) with recognition level equal to one have the highest precedence and they can use the entire OC. On the contrary, data items with recognition level equal to i $(i > 1)$ can use only the part of OC not used by data items with lower recognition levels. It is also important to point out that in formula (13) we must use the B_i^{t+1} value because it provide the size of sub-queue i in OC *after* completing the internal reordering, the insertion of new items and the removal of old items. However, it is quite straightforward to observe that B_i^{t+1} is simply given by

$$B_i^{t+1} = \min(N_{0,i} + B'_i, F_i) \ . \tag{14}$$

Formula (14) can be explained by noting that if $N_{0,i} + B'_i > F_i$, then there are enough free slots in OC for all the new items that should enter at level i and for the old items that are already at level i after the reordering. On the other case, some new items will be discarded and/or some old items will be dropped till only F_i slots are occupied[3]. Now, by using formula (13), (14) and the initial condition $F_1 = B$ we can iteratively compute all remaining B_i^{t+1} and F_i values.

Finally, to compute the state vector φ_a^{t+1} we introduce an auxiliary transition probability matrix P such that $\varphi_a^{t+1} = \varphi'_a P$. As observed in formula (12), a new data item is eligible for entering OC with probability $r_t(a)\nu_a^{t+1}[R_c]\upsilon_a^t[i-1]$. However, that new data item will certainly be fetched by the tagged node if $N_{0,i} \leq F_i$, otherwise only a fraction $F_i/N_{0,i}$ will be fetched. Thus, the probability $p_{0,i}$ that a new item effectively enters the OC cache can be expressed as

$$p_{0,i} = \begin{cases} r_t(a)\nu_a^{t+1}[R_c]\upsilon_a^t[i-1] & \text{if } N_{0,i} \leq F_i \\ r_t(a)\nu_a^{t+1}[R_c]\upsilon_a^t[i-1]\frac{F_i}{N_{0,i}} & \text{otherwise ,} \end{cases} \tag{15}$$

On the other hand, an old data item that was already stored in the OC cache should be removed if the new data items have consumed all (or most of the) free slots. More precisely, we have that

$$p_{i,0} = \begin{cases} 0 & \text{if } N_{0,i} + B'_i \leq F_i \\ 1 & \text{if } N_{0,i} > F_i \\ 1 - \frac{F_i - N_{0,i}}{B'_i} & \text{otherwise .} \end{cases} \tag{16}$$

Formula (16) indicates that in case $N_{0,i} + B'_i \leq F_i$ there is no need of replacing data items because the free slots can accommodate both new and old data items. On the other hand if $N_{0,i} \leq F_i$ a fraction $1 - \frac{F_i - N_{0,i}}{B'_i}$ of old data items has to be removed from the OC cache.

5.5 Item Diffusion

[3] Following the same line of reasoning it is easy to compute the number $\hat{N}_{0,i}$ of new items that will effectively enter the OC cache at time $t+1$ as $\min(N_{0,i}, F_i)$.

At the beginning of network operations data items are not replicated, but they are stored in the LI caches of the nodes that have generated them. Since data items are assumed to be uniformly distributed over the network nodes, we have that at the initialization time $r_0(a) = 1/N$. After an encounter event occurring at time t the replication of each data item is given by

$$r_t(a) = r_0(a) + (1 - r_0(a)) \left[\frac{Pop(a.c)}{N} \psi_a^t[1] + \left(1 - \frac{Pop(a.c)}{N}\right) \sum_{i=1}^{R} \varphi_a^t[i] \right] \qquad (17)$$

where either the node has generated the data item with probability $r_0(a)$ or it is subscribed to the channel the data item belongs to and it has a copy of a in its SC cache, or it is not subscribed to the channel the data item belongs to and it has a copy of a in one of the sub-queues of its OC cache.

6. PERFORMANCE EVALUATION

6.1 Model Validation

In this section we validate the model accuracy comparing the analytical results with the results obtained using the simulator environment presented in [2]. More precisely, we consider a network composed of 45 nodes, 3 different channels with 99 data items each (297 data items in total). The channel popularity follows a zipfian distribution with parameter 1. In other words, giving a channel j, the probability of a node to be subscribed to that channel is given by $Pop(j) = N \cdot j^{\beta} / \sum_{i=1}^{M} i^{\beta}$, with $\beta = 1$. Thus, channel 1 is the most popular, while channel 3 the least popular. Finally, nodes are assumed to move according to a Random Waypoint model in a square area of side 1Km. The node speed is uniformly sampled in the range [1,1.86] m/s, which are typical pedestrian speeds [8], and we applied the techniques described in [13] to make sure that the model runs in stationary conditions. To better evaluate the impact of the OC cache on the data dissemination process, in the following simulations we assume that the content of SC is *not shared* during an encounter event. To take into account this assumption in the model, only formula (17) has to be changed to remove the $\psi_a^t[1]$ term. In the following graphs average values and 90% confidence intervals are computed by conducting 100 simulations of each scenario with different random seeds.

In Fig. 6, 8 and 10 we plot the temporal evolution of the *hit ratio*, defined as the fraction of data items of the channel a node is subscribed to which are stored in SC, using $R_c = 5$ and $R = 5$ for all the channels. As the curves indicate, the larger the OC the shorter the convergence time to hit ratios equal to one. For instance, all the channels reach a hit ratio equal to one around 800s when OC size is 3 slots, while this time is reduced to 200s if the size is 50 slots. Finally, the comparison between analytical and simulation results confirm that our model is sufficiently accurate to predict the temporal evolution of the hit ratios, independently of channel popularity or OC sizes. In Fig. 5, 7 and 9 we show the temporal evolution of the data-item replication level $r_t(a)$ for the OC cache, using $R_c = 5$ and $R = 5$ for all the channels. For the sake of clarity, the sub-plots in the bottom rows directly compare the analytical and simulation results for the same channel, with the most popular channel in the rightmost sub-plot and the least popular one in the leftmost sub-plot. Important observations can be derived from the shown results. First of all, the replication levels show clear peaks. In addition, the peak recognition value is reached before for channels with high popularity than for channels with low popularity. This is due to the fact that the most popular channel and its data items get recognized before all the other channels. Thus, the data dissemination process for the most popular channel can start before than for other channels. However, after this peak the replication levels of all the channels decrease. This is a typical behavior of the recognition heuristic and of the replacement policy used for the OC cache. Indeed, the more diffused a data item is (i.e., the more recognized), the less relevant it is for the dissemination process. This implies that less diffused items can increment their replication level more easily than data items of popular channels. This also explains why the hit ratio of the least popular channel is comparable to the hit ratio of the most popular channel. The second interesting observation is that the replication levels converge to steady values that are almost the same for all the channels. Typically, this happens when all the data items have reached the maximum recognition level, i.e. R. In this case, all data items are considered equivalent for the dissemination process and they are equally distributed in the OC caches. Finally, it is important to point out that our model is remarkably accurate in predicting this stationary behavior of the data-item replication levels. Furthermore, our model is able to predict the times at which the replication levels reach their maximum and minimum values.

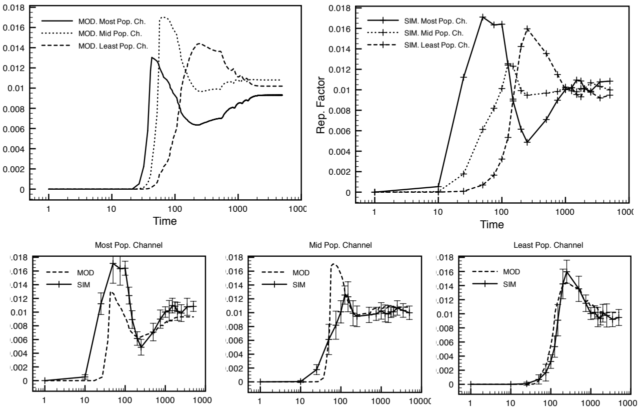

Figure 5: $r_t(a)$ for $R_c = 5$, $R = 5$ and $B = 3$.

6.2 Scalability Analysis

In this section we aim at demonstrating the scalability of our model. More precisely, an accurate simulation of node mobility, communication protocols and cache management policies would be impossible to be executed in feasible time for large-scale opportunistic networks involving thousands of nodes and hundreds of channels. In these cases, we argued that the analysis can help in exploring the behavior of the system even with large numbers of involved peers and objects. To validate this statement, we have solved our model considering a network with $N = 1000$ nodes, $M = 100$

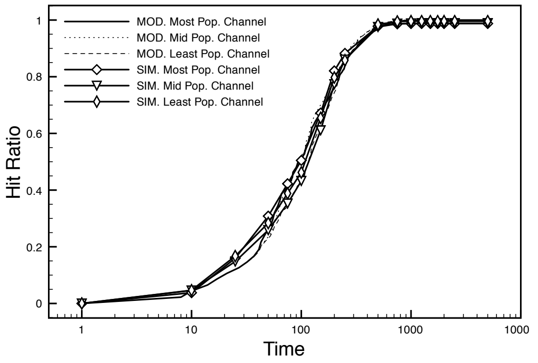

Figure 6: Hit ratios for $R_c = 5$, $R = 5$ and $B = 3$.

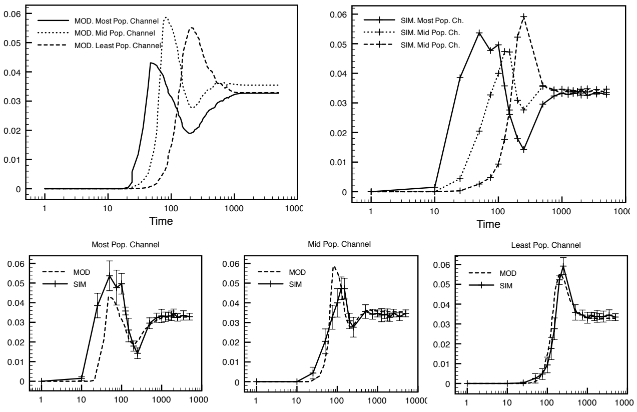

Figure 7: $r_t(a)$ for $R_c = 5$, $R = 5$ and $B = 10$.

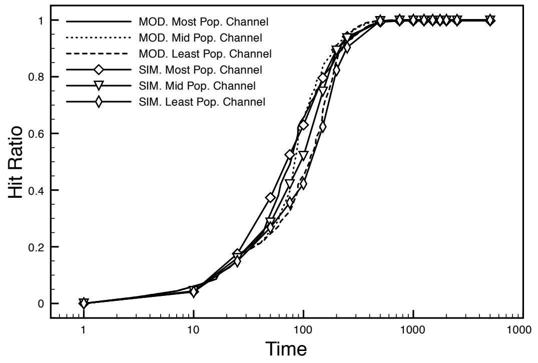

Figure 8: Hit ratios for $R_c = 5$, $R = 5$ and $B = 10$.

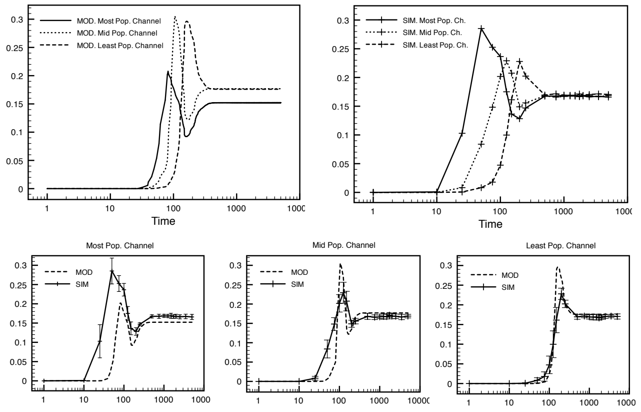

Figure 9: $r_t(a)$ for $R_c = 5$, $R = 5$ and $B = 50$.

channels, and ten objects per-channel (1000 data items in total). This values have been chosen to better represent a "long-tail" scenario, where a relatively large number of small groups of users are subscribed to specific channels. Furthermore, channel popularity still follows a zipfian distribution of parameter 1.

Fig. 11 and 12 show the hit ratios for the most and least popular channels, respectively, and for various sizes of the OC cache. As benchmark scheme we consider an epidemic-based data dissemination protocol, where meeting nodes only share the data items for the channels they are both subscribed to. The first important observation is that the OC cache ensures a substantial performance gain over the epidemic scheme. In particular, for the least popular channel the time to reach a hit ratio equal to one is one order of magnitude shorter when using the OC cache. Nonetheless, it should to be noticed that, for the most popular channel, a greater OC does not necessarily imply a performance improvement over the epidemic scheme. This is due to the fact that, with a larger OC, the data items of the most popular channel quickly reach the maximum recognition threshold R, even before they have beed copied in all the nodes subscribed to their channel. However, when a data item is highly recognized it is less relevant for the data dissemination process, and less diffused items have higher priority during item replacements in OC. The combination of these behaviors may negatively affect the effectiveness of the diffusion process for the high popular channels.

The better understand the impact of channel recognition thresholds on these behaviors, Fig. 13 show the hit ratios of the most and least popular channel for an OC size equal to ten slots and two different channel recognition thresholds. As expected, the larger the channel recognition threshold, the slower the data dissemination process. However, while for the most popular channel this delay seems to affect all hit ratio values at the same manner, in the case of the least popular channel high hit ratio values are more disadvantaged than lower ones. We have also investigated a network scenario where the channel recognition threshold is set to zero for all the channels. In this case, all the channels are considered recognized from $t=0$ and there are no differences between the time instants at which the data dissemination process starts for channels with different popularities. Then, in Fig. 14 we show the hit ratios of the most and least popular channel for two different OC sizes. On the other hand, the size of the OC cache has still a remarkable impact on the convergence times of the hit ratios. In the last set of results, shown in Fig. 15, we use the same parameter setting used for obtaining the results in Fig. 14, but we increase the number of nodes by ten times (10000 nodes in total).We can observe that when increasing the number of nodes the most popular channel does not experience significant improvements in the

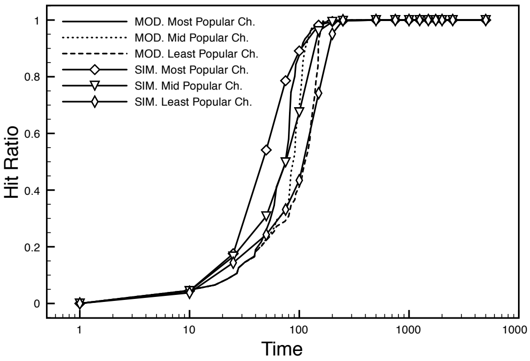

Figure 10: Hit ratios for $R_c = 5$, $R = 5$ and $B = 3$.

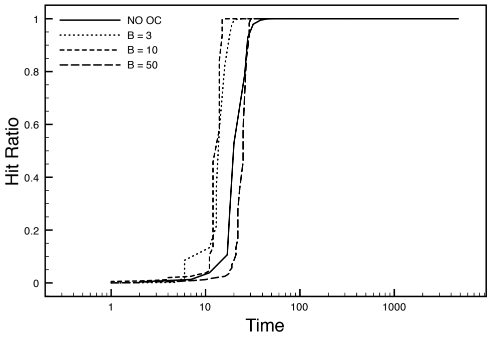

Figure 11: Hit ratio of the most popular channel for varying B values, $R_c = 5$ and $R = 5$.

convergence time of the hit ratios. On the other hand, a quite significant improvement can be observed for the least popular channel because a larger pool of OC caches ensures that more copies of the most rare data items are circulating in the network.

7. CONCLUSIONS

In this paper we have developed an analytical model to describe the performance of a content dissemination mechanism for opportunistic networks that relies on a cognitive scheme, known as the recognition heuristic, to decide whether to store a copy of data items fetched from the caches of encountered nodes. Our model takes into account the different popularities of content types, and highlights the impact of the shared memory contributed by individual nodes to make the dissemination process more efficient. In particular, we have shown that our model is sufficiently accurate to capture the transient and steady-state behaviors of key performance indexes of the data dissemination system, such as data replication levels and hit ratios. Furthermore, the low computational complexity of our model has allowed us to explore network scenarios with large numbers of involved peers and objects, which are problematic to study using only a simulation-based approach. Future work involve the extension of our model to consider heterogeneous environments where there may be nodes with different mobility patterns, or data items may have different priorities. Furthermore, we plan to study how to extend the recognition heuristic to include mediators that take into account social-based information about the users.

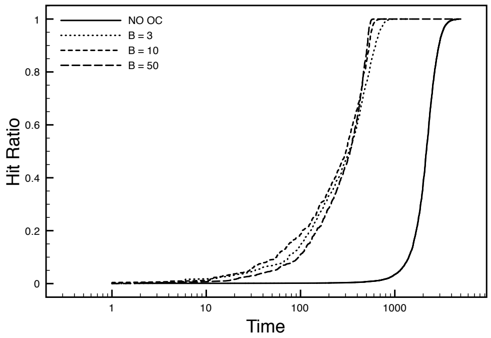

Figure 12: Hit ratio of the least popular channel for varying B values, $R_c = 5$ and $R = 5$.

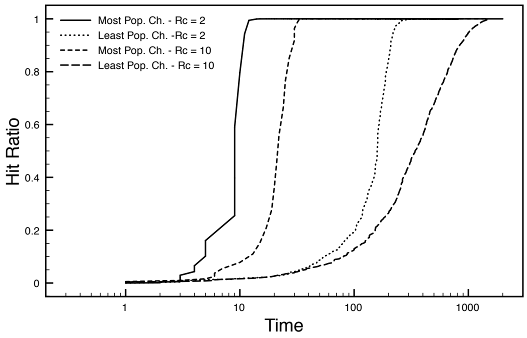

Figure 13: Hit ratios for the most and the least popular channels and varying R_c values, $R = 5$ and $B = 10$.

8. ACKNOWLEDGEMENTS

This work is funded by the EC under the RECOGNITION (FP7-IST 257756) and EINS (FP7-FIRE 288021) projects.

9. REFERENCES

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

[2] Marco Conti, Matteo Mordacchini, and Andrea Passarella. Data dissemination in opportunistic networks using cognitive heuristics. In *Proc. of IEEE WOWMOM 2011*, 2011.

[3] P. Costa, C. Mascolo, M. Musolesi, and G.P. Picco. Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks. *IEEE JSAC*, 26(5):748–760, 2008.

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

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

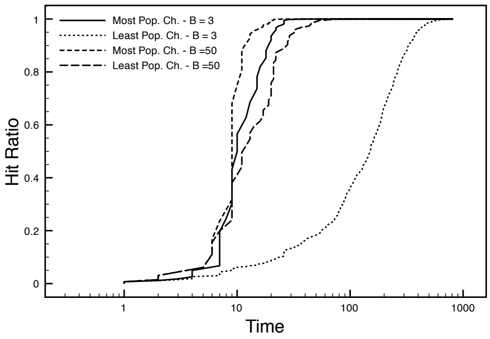

Figure 14: Hit ratios for the most and the least popular channels and varying B values, $R_c = 0$ and $R = 5$.

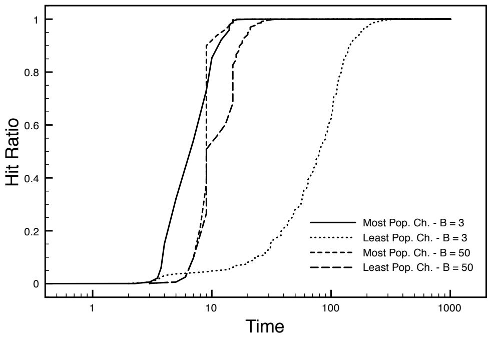

Figure 15: Hit ratios for the most and the least popular channels and varying B values, $R_c = 0$, $R = 5$ and $N = 10000$.

[6] D.G. Goldstein and G. Gigerenzer. *The recognition heuristic: How ignorance makes us smart*, pages 37–58. Oxford University Press, 1999.

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

[8] D. Karamshuk, C. Boldrini, M. Conti, and A. Passarella. Human mobility models for opportunistic networks. *IEEE Comm. Mag.*, 46(12):157–165, Dec. 2011.

[9] V. Lenders, M. May, G. Karlsson, and C. Wacha. Wireless ad hoc podcasting. *ACM MC2R*, 12:65–67, Jan. 2008.

[10] J.N. Marewski, W. Gaissmaier, L.J. Schooler, D. 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.

[11] Julian N. Marewski, Wolfgang Gaissmaier, and Gerd Gigerenzer. Good judgments do not require complex cognition. *Cogn. Process*, 11:103–121, 2010.

[12] M. Monti, L. Martignon, Gigerenzer. G., and N. Berg. The impact of simplicity on financial decision-making. In *Proc. of CogSci 2009*, 2009.

[13] W. Navidi and T. Camp. Stationary distributions for the random waypoint mobility model. *IEEE Trans. Mob. Comput.*, 3(1):99–108, 2004.

[14] A. Passarella. A survey on content-centric technologies for the current Internet: CDN and p2p solutions. *Comput. Comm.*, 35(1):1–32, January 2012.

[15] L. Pelusi, A. Passarella, and M. Conti. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *IEEE Comm. Mag.*, 44(11):134 –141, Nov. 2006.

[16] L.J. Schooler and R. Hertwig. How forgetting aids heuristic inference. *Psychological Review*, 112(3):610, 2005.

[17] Sascha Serwe and Christian Frings. Who will win wimbledon? the recognition heuristic in predicting sports events. *J. Behav. Dec. Making*, 19(4):321–332, 2006.

[18] L. Yin and G. Cao. Supporting cooperative caching in ad hoc networks. *IEEE Trans. Mob. Comput.*, 5(1):77–89, 2006.

[19] E. Yoneki, P. Hui, S.Y. Chan, and J. Crowcroft. A socio-aware overlay for publish/subscribe communication in delay tolerant networks. In *Proc. of ACM MSWIM'07*, pages 225–234, 2007.