Context Sensing for Autonomic Forwarding in Opportunistic Networks

Elena Pagani
Computer Science Dept., Università degli Studi di Milano, Italy
CNR-IIT, National Research Council, Italy
Email: pagani@dico.unimi.it

Gian Paolo Rossi
Computer Science Department
Università degli Studi di Milano, Milano, Italy
Email: rossi@dico.unimi.it

***Abstract*—Rank-based policies represent a promising approach for designing message forwarding algorithms that meet the needs of opportunistic networks. In fact, they combine low computation and communication costs with good performance in terms of both latency and delivery rates. Nonetheless, they highly depend on the mobility scenario relevant to the user, and a forwarding policy with good performances in heterogeneous settings has yet to be designed. In this paper, we propose to provide each mobile device with novel *autonomic* observation and reasoning components according to the following objectives: enable the device (i) to achieve awareness about the behavior of the mobility scenario it is moving in, and (ii) to identify the role played by the device within the set of other moving devices. These components are combined into a self-configuring forwarding algorithm that uses them to locally install both the utility function and the relevant settings suitable for the sensed configuration.**

Through of extensive simulations, this paper shows that by properly discriminating between roles it is possible to derive a self-configuring forwarding mechanism that constantly performs well in different mobility settings.

I. Introduction

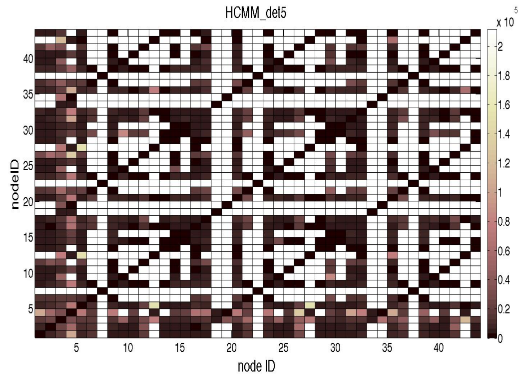
The design of message forwarding algorithms that meet the needs of opportunistic networks (ONs) [18] has won great attention in recent years. A number of proposals have been presented in the literature, which we can divide roughly into three classes: (i) epidemic and gossip algorithms (e.g. [21]); (ii) community-based algorithms (e.g. [11], [5]), and (iii) rank-based algorithms (e.g. [6], [7], [14]). The first class contains algorithms where message forwarding is performed blindly. This helps to minimize the delivery latency and to maximize the delivery likelihood, which we pay for with the generation of a high number of message duplicates and consequent waste of bandwidth. The second class lies in the observation that users tend to form communities and to visit common places of interest. The algorithms in this class may need community detection algorithms, which are still expensive and difficult to compute in a distributed way. Algorithms in the third class analyze the contact dynamics in order to identify the nodes that may act as good relays for message forwarding towards a given destination.

Rank-based algorithms are quite interesting because they combine low computation and communication costs with good performance in terms of both latency and delivery rates. Nonetheless, they highly depend on the mobility scenario relevant to the user. As a consequence of the well-known fact that in daily life people customarily move from location to location according to varying mobility behavior, we advocate a forwarding mechanism which is either independent of mobility or aware of the context of mobility it is operating in so as to select the proper parameter setting. It has been shown, e.g. [17], [7], that each available proposal has been designed for a specific mobility scenario and therefore none of them is able to perform adequately with heterogeneous mobility settings. To overcome the above drawback, in this paper we propose to provide each mobile device with novel *autonomic* observation and reasoning components according to the following objectives: enable a device (i) to achieve awareness about the behavior of the mobility scenario it is moving in, and (ii) to identify the role played by the device within the set of other moving devices (or nodes, in the sequel). Our knowledge about mobility context and local role is then used to choose the rank-based policy most appropriate for that scenario. In our proposed approach, every node locally observes its contact dynamics by recording number and timing of encounters; then it uses them to infer its own role within the group. We distinguish between just two possible roles: sedentary or traveler. This binary information is exchanged at each contact, so enabling the inference of the global mobility behavior. These components are combined into a self-configuring forwarding algorithm that uses them to locally install both the utility function and the relevant settings suitable for the sensed configuration. In [17], we have observed that the best performing path to a destination is obtained by letting emerge the mobility attitudes of nodes that are required in the particular mobility scenario. For instance, if most nodes are sedentary, i.e. confined to their respective communities, the proper mechanism is the one letting traveler nodes (moving freely from community to community) emerge; in fact, they are suitable relays for reaching the community to which the destination belongs. By contrast, if all the nodes have homogeneous attitudes, those with a stronger inclination to encounter the message destination should be preferred. This paper shows that by properly discriminating between roles it is possible to derive a self-configuring forwarding mechanism that constantly performs well in different mobility settings.

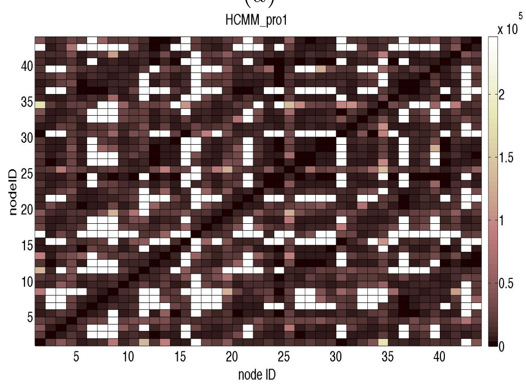
II. Background work

In [17], we analyzed five rank-based forwarding algorithms for Opportunistic Networks in five different mobility scenarios.

978-1-4673-1430-5/12/$31.00



(a)



(b)

Fig. 1. Map of ICTs for every pair of nodes, for (a) a closed community setting, and (b) a mixed community setting.

The algorithms were chosen in order to test the appropriateness of different mechanisms and thereby ensure good performance of the forwarding process. The considered mobility environments were selected to represent different real-life mobility settings; their main differences concerned the type of communities and the presence of travelers. Communities are *closed* when their members tend to operate autonomously without mixing with members of other communities. Under this condition, intra-community communications are mainly supported by a few traveler nodes, i.e. nodes used to contact members of different communities. We can see these characteristics for instance in the maps of the inter-contact time (ICT) for every pair of nodes. With closed communities (fig.1(a)), several nodes never experience an encounter during a window of time of 156 hours (13 working days, from 8 a.m. to 8 p.m.). By contrast, the nodes with identifiers 1 to 5 encounter almost every other node. If this behavior is shown by all nodes, then communities have less defined boundaries (they are *mixed*), nodes are less sedentary and are more likely to visit other communities (fig.1(b)).

The results in [17] show that a forwarding policy with good performances in different mobility scenarios has yet to be designed. In general, destination independency is unable to characterize the relays appropriate for reaching a certain destination. An algorithm performs well when it is able to discriminate between nodes confined to a community different from that of a message destination, nodes confined to the same community as the message destination, and travelers able to establish a link between the two sets. The best results are achieved by Fresh [6] for scenarios with *closed* communities, and by Greedy [7] for scenarios with *mixed* communities. In Fresh, the utility of a node n for a destination d is the time elapsed from its last encounter with d: the lower this time, the greater the utility of n as a relay for messages addressed to d. In Greedy, the utility of n is the number of its encounters with d: the higher the counter, the more useful the node. Hence, Fresh has no history – as it recalls only the last encounter – while Greedy holds the whole history of past encounters. In closed communities, a node n runs continually into nodes belonging to its own community and updates their last encounter times. By contrast, nodes belonging to other communities and travelers are likely to have a much greater associated last encounter time because their encounters with n are sporadic. Thus, good discrimination is achieved by Fresh among different sets.

In mixed communities, the patterns of encounter are very similar for all pairs of nodes. So are the distributions of the last encounter times. The habit (or lack of it) of encountering a node is better revealed if the whole history is considered, that is, if Greedy counters are used. As an example, in fig.2, the *ecdf* is shown of the utility values computed by either (a) a sedentary or (b) a traveler, using the Fresh algorithm in a closed community model. In the former case, there is a set of nodes that either have never been encountered (last encounter time is 0), or have experienced very sporadic encounters and in any case probably far less recently than the nodes in the same community. The nodes in the same community, by contrast, have been encountered very recently. In the latter case, intermediate values are possible, and steps account for visits of the traveler to different communities in the past. With mixed communities, fig.2(b) is the profile common to all nodes. But with Greedy (fig.2(c) and (d)), a sedentary node has zero or a very low counter for all nodes outside its community, and a high counter for the nodes in the same community. A traveler, rather, shows more heterogeneous counters, depending on the different frequency of visits to distinct communities.

Sedentary nodes show a characteristic common to both Fresh and Greedy utilities: there is a large step in the plot, with a flat behavior, unlike what we see with travelers. This characteristic is exploited by our algorithm to differentiate between the two different node roles.

Usually, users are unaware of the characteristics of the environment they currently find themselves in. On the basis of the above results, in this paper we propose a novel rank-based autonomic forwarding algorithm. The algorithm merges Fresh and Greedy utility functions and, depending on the environment, chooses either algorithm. The underlying idea is to learn about the environment through the analysis of the encounter pattern, and then use the ranking which (*in that environment*) better discriminates between the two different different node roles. This allows us to more precisely identify

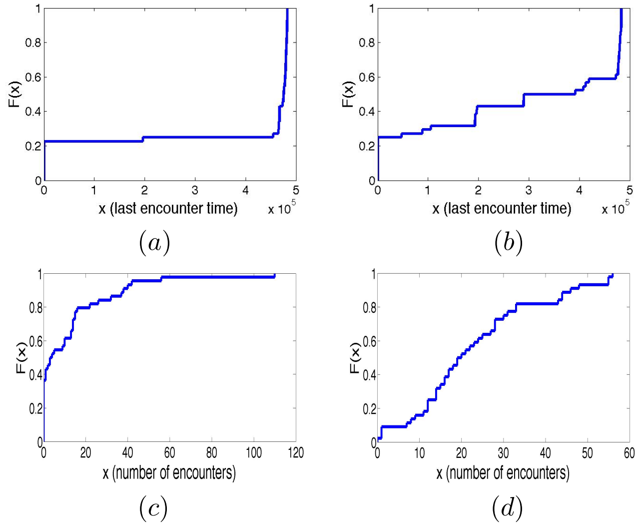


Fig. 2. Empirical cumulative distribution function of: Fresh utilities in a closed community setting, for (a) a sedentary, and (b) a traveler; and of Greedy utilities in a mixed community setting, for (c) a sedentary and (d) a traveler.

effective relays that transport messages to their destinations.

III. Autonomic Algorithm

The autonomic algorithm consists of three steps that each node concurrently runs. The steps aim at (1) ascertaining whether a node is either a sedentary or a traveler, through the observation of its encounter dynamics; (2) determining the characteristics of the environment through the exchange of roles with the encountered nodes; and (3) using what is learned in the other steps so as to adopt the appropriate policy for message forwarding. The three steps are described in the following subsections.

A. Sensing the Node Role

The first step is performed periodically. With a period of size M, a node analyzes its utility distribution and – according to the results reported in the previous section – it determines whether its profile fits the sedentary or the traveler picture.

Algorithm 1 `whoAmI` Procedure

INIT: `my_role` $\leftarrow \perp$; `S/T bit` $\leftarrow 0$; `last_sampling` $\leftarrow 0$;
when (current_time - `last_sampling`) $\geq$ M **do**
  compute *ecdf* of Greedy utilities;
  $\Delta_{\mathcal{U}} \leftarrow min\{\forall k, 1 \leq k < S : \Delta(ecdf)$ between subintervals k and $k+1\}$;
  if $\Delta_{\mathcal{U}} < \Delta_{th}$ **then**
    `my_role` $\leftarrow$ *sedentary*; `S/T bit` $\leftarrow 0$;
  else
    `my_role` $\leftarrow$ *traveler*; `S/T bit` $\leftarrow 1$;
  end if
  `last_sampling` $\leftarrow$ current_time;
end do

Let $\#values$ be the number of distinct utility values a node observes; initially, $\#values = 0$. The mechanism adopted in this paper is as follows: a node estimates the *ecdf* over its utility values. It partitions the interval that the utility values span into S subintervals, with $S = min\{\#values, K\}$ for some parameter K. The slope between two adjacent subintervals is estimated as the difference between the *ecdf* values in those intervals. Let $\Delta_{\mathcal{U}}$ be the minimum of the non-null computed slopes. If the $\Delta_{\mathcal{U}}$ of the node is lower than a threshold Δ_{th}, i.e. it is in the most flat interval of the *ecdf* over the utility values, the node has a low slope, then it is a sedentary node. If the slope is always high, then the node is a traveler. The procedure is summarized in Algorithm 1, where we consider Greedy utilities since they have proven to yield more accurate discrimination, as shown in section IV-A. The value of the S/T bit records the node role, and is used to learn the environment characteristics as described in section III-B. In the Algorithm, the S/T bit is initially set to sedentary. In fact, when a node boots, it detects its neighbors and assumes they belong to its own community. The *ecdf* curve forms over time and evolves towards its real physiognomy as the encounters increase in number. Hence, travelers spend more time than sedentary nodes in identifying their status. In section IV-A we provide an analysis of the role learning phase.

The S/T bit aims at approximating community detection. Yet what we care about here is not a precise determination of community membership. Rather, we are interested in determining whether a certain node has no habit of encountering a destination (it is a sedentary node in a different community) or if it is a traveler that might run into either the destination or a node belonging to the destination community. To some extent, travelers take the role of *popular* nodes exploited by other forwarding algorithms (e.g. [11]). Hence, although this stage is the most computationally expensive, it is far less expensive than community detection algorithms. Moreover, it is performed only periodically. Exactly how often can be decided in terms of the desired trade-off between accuracy and cost.

B. Sensing the Environment

Nodes are assumed to periodically broadcast beacons to notify their presence to one-hop neighbors. Beacons also contain the S/T bit to indicate whether the node considers itself a sedentary or a traveler. Each node n collects – and updates upon each contact – the S/T bits of the encountered nodes. Should node n encounter a majority of sedentary nodes, then n infers that it is moving in a closed community. Otherwise, n will infer that it lives in a mixed community. The procedure is described by Algorithm 2.

It is worthwhile to notice that the procedure does not require the estimation of the global number of nodes in the system. On the other hand, nodes having few or rare encounters might either incorrectly classify the environment, or take too long to come up with an accurate classification. As an alternative, two encountering nodes could exchange the whole set of received S/T bits, and each node could merge the received set with its

Algorithm 2 `whereAmI` Procedure

INIT: `environment` $\leftarrow \perp$; `roles` $\leftarrow \emptyset$;
when contact with node n **do**
  send (my `S/T bit`) to n;
  receive `S/T bit` from n;
  add/update n's `S/T bit` in `roles`;
  if majority of entries in `roles` is 0 **then**
    `environment` $\leftarrow$ *closed*;
  else
    `environment` $\leftarrow$ *mixed*;
  end if
end do

own. However, this strategy might be expensive, as sets might prove excessively large in urban environments, for instance. We have decided to adopt the described low-cost policy. In section IV-B we present the length and accuracy measure of this learning phase.

C. Message Forwarding

At every contact, the two encountering nodes update both Fresh and Greedy utilities for each other and then exchange the same. Each node n analyzes every message m in local buffer:

Algorithm 3 `Rank&Forward` Procedure

INIT: `last` $\leftarrow$ [ ]; `counter` $\leftarrow$ [ ]; buffer $\leftarrow \emptyset$;
when contact with node n **do**
  `last`[n] $\leftarrow$ current_time; //Fresh utility
  `counter`[n] $\leftarrow$ `counter`[n] +1; //Greedy utility
  send (`last`, `counter`) to n;
  receive (`last`$_n$, `counter`$_n$) from n;

  for all messages m in my buffer **do**
    //(let $dest_m$ be m's destination)
    if ($n = dest_m$) **or** (`environment`=*closed* **and** (current_time - `last`[$dest_m$]) $>$ (current_time - `last`$_n$[$dest_m$])) **or** (`environment`=*mixed* **and** `counter`[$dest_m$] $<$ `counter`$_n$[$dest_m$]) **then**
      send m to n;
    end if
  end for
  receive messages from n and insert in buffer;
end do

if m is addressed to the encountered node, it is immediately delivered to its destination. Otherwise, n evaluates whether it is in either a closed or a mixed community environment. In the former case – according to Fresh – every message for destinations that the other node has encountered more recently is forwarded to the other node. In the latter case – according to Greedy – the messages are forwarded if addressed to destinations that the other node has encountered a higher number of times. The procedure is described by Algorithm 3.

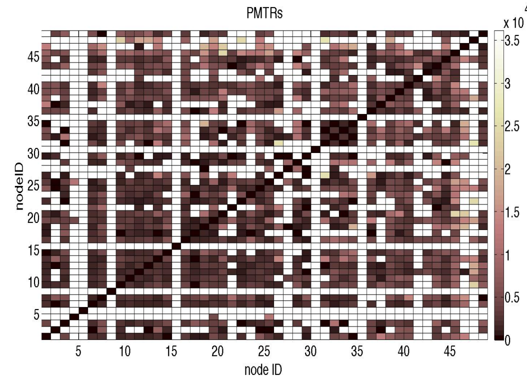


Fig. 3. Map of ICTs for every pair of nodes, for PMTRs.

IV. Performance Evaluation

In our simulations, the sample period is $M = 30$ minutes. The number of subintervals considered for role estimation is $K = 10$. We considered two mobility patterns produced with the HCMM synthetic model [1], where 44 nodes move in a 1000×1000 m. area with speed in [0.5, 1.5] m/s for 156 hours. The transmission range is 10 m. As an initial interaction matrix, we used weights derived from the number of contacts between pairs of nodes in a real trace, namely, the PMTR trace [8]. To the highest number of contacts we assigned weight 0.9. We adopted the weight associated to half of the average number of contacts as a threshold to derive the connection matrix. No reconfiguration is performed and the remaining probability is set to 0.8. In the scenario named HCMM_det5, the next cell is chosen deterministically and the rewiring probability is 0.1; we adopted 5 travelers. In the HCMM_pro1 scenario, the probabilistic criterion is adopted, with rewiring probability of 0.3 and 1 traveler. The two HCMM scenarios yield, respectively, closed and mixed community environments (fig.1). This helped us to evaluate operations of the approach under well-defined mobility conditions.

Moreover, we analyzed the behavior of the autonomic algorithm under real mobility conditions using for this purpose the PMTR trace. The latter concerns 44 people on a campus where each person is equipped with a wireless device called a PMTR (Pocket Mobility Trace Recorder) having 10 m. radio range [8]. In the real dataset, we eliminated nights and weekends, thus producing a dataset covering 13 working days, from 8:00 a.m. to 8:00 p.m. (for a total of 156 h.). The PMTR scenario is a difficult environment: users stay in their offices for lengthy periods, the environment is sparse, and contacts are rare [19]; long latencies and low delivery rates are obtained also when adopting an epidemic diffusion [17]. All pairs of nodes show a quite homogeneous behavior. For the sake of comparison, in fig.3 the ICT map for PMTR is reported.

With the considered data sets and $M = 30$ minutes, the node role is recomputed after 38 encounters in PMTR, 147 encounters in HCMM_pro1, and 477 encounters in HCMM_det5, on average. Although the last data might suggest a need for a more frequent sampling to achieve better accuracy, it

TABLE I
$\Delta_{\mathcal{U}}$ VALUES FOR BOTH FRESH AND GREEDY.

	Fresh		Greedy	
	$\Delta_{\mathcal{U}}$(trav)	$\Delta_{\mathcal{U}}$(sed)	$\Delta_{\mathcal{U}}$(trav)	$\Delta_{\mathcal{U}}$(sed)
HCMM_pro1	0.065	0.071	0.057	0.050
HCMM_det5	0.058	0.047	0.066	0.037
PMTR	0.069	0.045	0.040	0.020

is worthwhile to notice that in HCMM_det5 the encounters occur more often because nodes in the same community keep running into one another. This does not lead to a change of the *ecdf*. Thus, low rate sampling seems suitable for all the considered scenarios.

We must make another observation. All the nodes in our simulations boot at the same time and are initially unaware of the environment, according to the provided algorithms. This results in a learning phase lasting several hours. Indeed, in real settings, the learning should speed up because when a node wakes up it might be surrounded by nodes whose learning phases have ended. And this can promptly provide the new node with an appropriate initial state. However, it is worthwhile to observe that the amount of time a node spends to learn both role and environment does not represent a critical concern. In fact, in daily life people routinely move from location to location while engaging in their social activities, and the mobility within a single location will very likely be the same over time. Upon the first role and environment detection, the relevant setting and node's awareness could be locally recorded with associated geo-position data. When the device moves to that position in the future the local setting can be directly applied by skipping learning phase.

A. Evaluation of role sensing

In order to finetune the mechanism used to learn the node role, we analyzed the behavior of both Fresh and Greedy in the three scenarios in terms of their ability to discriminate sedentary nodes and travelers. In Table I we report the minimum slope of the *ecdf* of the utilities. We analyzed the profiles of the Greedy utilities (as in fig.2) and noticed that they correctly detect a majority of travelers in HCMM_pro1 and a majority of sedentary nodes in HCMM_det5. By contrast, Fresh utilities provide incorrect indications in HCMM_pro1 – in line with the arguments in section II – and all nodes have a very homogeneous behavior w.r.t. PMTRs. Moreover, in HCMM_pro1 Fresh utilities have greater slope for nodes with a sedentary type profile than for nodes with a traveler type one (Table I). Thus, we decided to adopt Greedy for role learning. Through experiments, we determined that a common threshold $\Delta_{th} = 0.046$ allows us to achieve the best performance with all scenarios.
In Table II we report the accuracy in determining node roles for the considered mobility models. For both ranking policies, we report the number of travelers and sedentary nodes as obtained by the *ecdf*'s appearance. Moreover, we report the estimated number of travelers and sedentary nodes by using $\Delta_{th} = 0.046$. We can observe that Fresh *ecdf*'s

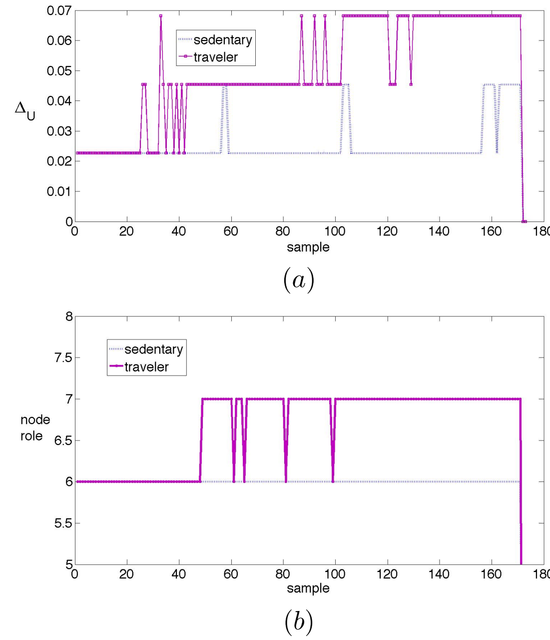


Fig. 4. (a) $\Delta_{\mathcal{U}}$ and (b) node role over time, according to Greedy utilities in HCMM_det5.

apparently show a majority of nodes with a sedentary behavior in HCMM_pro1. This is incorrect as well as the estimated role. In fig.4, the behavior of $\Delta_{\mathcal{U}}$ and the node's estimated role are shown over time, for both sedentary and traveler nodes in HCMM_det5. Similar behavior is observed with HCMM_pro1. For the sake of convenience, in fig.4(b), we assumed the node's role to be 6 when the node estimates it is a sedentary one, 7 otherwise. The node's role is computed according to Algorithm 1. We can observe that – as mentioned in section III-A – travelers need more time than sedentary nodes to appropriately estimate their role. This is because the *ecdf* curve changes slowly from a sedentary profile to a traveler one and in the meantime its slope increases. We observed that, for every node, role estimation in a new geo-positions stabilizes after an average of 20-25 hours.

B. Evaluation of environment sensing

In fig.5, the estimation of the environment nature is shown over time, for all scenarios. In all figures, the plots of all nodes are reported. The estimation is performed according to Algorithm 2. As before, we assumed that environment estimation is 6 for closed communities, 7 otherwise. In both HCMM_pro1 and PMTR, the majority of nodes converges towards a correct estimation of the environment nature, although in the former case it takes a long time (nearly 30 hours on average). Yet, in HCMM_pro1, oscillations are possible also at the end of the simulated time. In HCMM_det5, a minority of nodes wrongly tends incorrectly to interpret the surrounding environment as a mixed community scenario. More precisely, through the exchange of the S/T bit, in HCMM_det5 there are 29 nodes interpreting the scenario as a closed community and 15 as a mixed community. In HCMM_pro1, 31 nodes interpret the

TABLE II
ACCURACY IN DETERMINING NODE ROLES.

	Fresh				Greedy			
	real		estimated		real		estimated	
	traveler	sedentary	traveler	sedentary	traveler	sedentary	traveler	sedentary
HCMM_pro1	17	27	20	24	34	10	23	21
HCMM_det5	9	35	16	28	13	31	15	29
PMTR	31	13	35	6	34	10	26	18

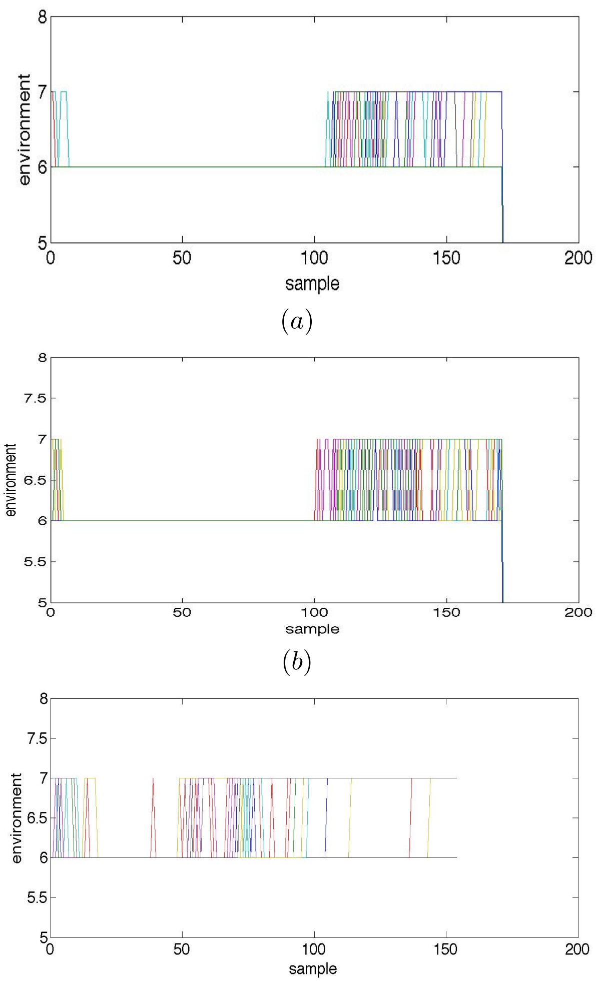


Fig. 5. Environment estimation in (a) HCMM_det5, (b)HCMM_pro1, and (c) PMTR.

scenario as a mixed community and only 13 nodes as a closed community. Finally, in PMTR 32 nodes converge toward a mixed environment while 12 assume a closed community environment. In terms of message forwarding, this means that, along the message delivery path, some relay nodes will forward packets by using Fresh utilities and some by using Greedy utilities. Clearly, fast convergence towards the appropriate interpretation (complete with general agreement on the same) is desirable in order to avoid possible path oscillation and to ensure good performance of the algorithm. Anyway, the performance achieved with this policy is very good and promising, as we show below.

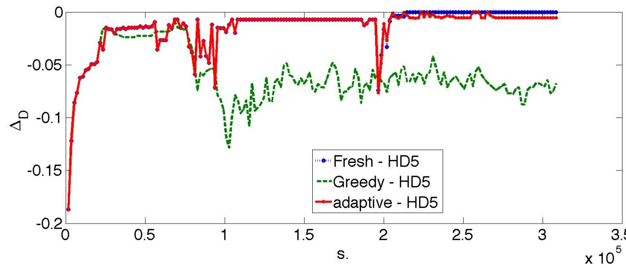
C. *Performance of the autonomic algorithm*

We measured the autonomic algorithm's performance in comparison to those of the original Fresh and Greedy algorithms, in the considered environments. Every M minutes each node generates a message for every other node; messages are forwarded according to the system view and to the utilities of the nodes at generation time. Consequently, dynamic adaptation of the utilities while messages are forwarded is not reproduced and so the shown results are likely worse than in a real environment where forwarding and utility computation are performed concurrently. Results are averaged over all source-destination pairs. Mean performance over all the simulation time is reported in Table III, where Δ_E is the variation in percentage of the number of hops needed to reach a destination with respect to the optimal (Dijkstra) path, Δ_L is the variation in percentage of the latency with respect to the optimal path, and Δ_D is the variation in percentage of the number of reached destinations with respect to the optimal path. We can see from the Table that the autonomic algorithm uses more hops, probably due to inconsistencies in the determination of the environment nature which led the nodes to use different rankings along the paths. Yet, with both synthetic models we achieve a better coverage, with latency comparable or better than the original algorithms. With PMTR a slight decrease in coverage is observed – which might explain the decrease in both latency and number of hops – but performance remains comparable with those of the original algorithms.

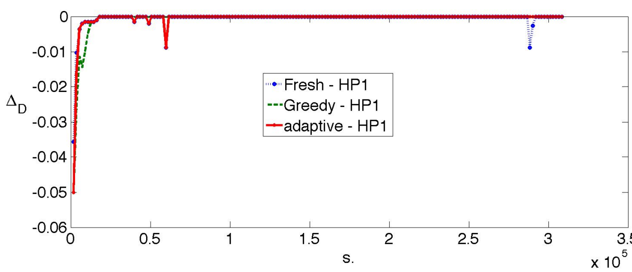
In fig.6, the coverage – expressed as the percentage of destinations lost on average with respect to optimal routing – is reported over time for the two synthetic models. In spite of inconsistencies in determining the scenario nature, *no loop* forms along the paths. Indeed, for both scenarios a greater coverage than with the original algorithms is achieved, thus showing that messages are correctly delivered to their destinations. The behavior of latency (fig.7) offers proof of what we pointed out earlier about the learning times of travelers. In the case of HCMM_det5 right from the start almost all nodes are convinced of being sedentary. They disseminate this view to other nodes, thus forcing to estimate closed communities. As a consequence, most nodes adopt the Fresh ranking to select relays, and latency always converges toward that of the best algorithm. For HCMM_pro1, travelers spend more time trying to understand their true nature and to communicate it to other nodes. Therefore, the latency initially approximates that of the Fresh algorithm – probably because most nodes tend to use Fresh in line with their initial view of

TABLE III
AVERAGE PERFORMANCE OF THE ALGORITHMS

	Fresh			Greedy			autonomic		
	Δ_E	Δ_L	Δ_D	Δ_E	Δ_L	Δ_D	Δ_E	Δ_L	Δ_D
HCMM_pro1	0.17	15.5	-0.001	0.13	8.6	-0.0007	0.53	13.21	0
HCMM_det5	0.09	8.11	-0.009	0.06	9.5	-0.06	0.36	7.7	-0.005
PMTR	0.20	50.2	-0.27	0.16	53.3	-0.26	0.18	48.61	-0.29

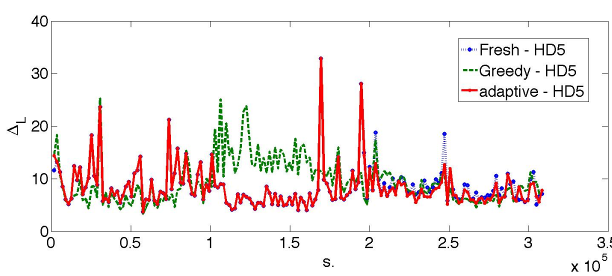


(a)

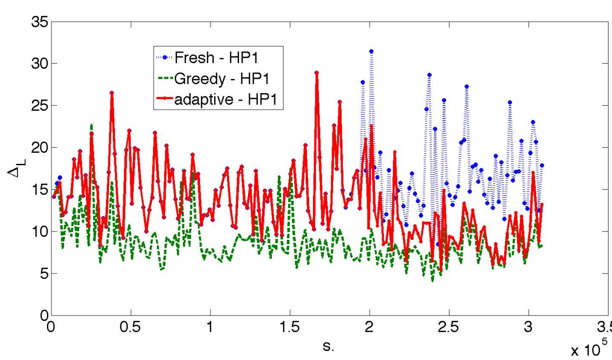


(b)

Fig. 6. Coverage over time for (a) HCMM_det5 and (b) HCMM_pro1.

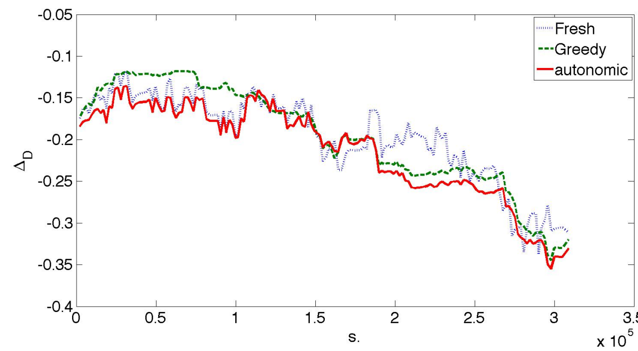


(a)

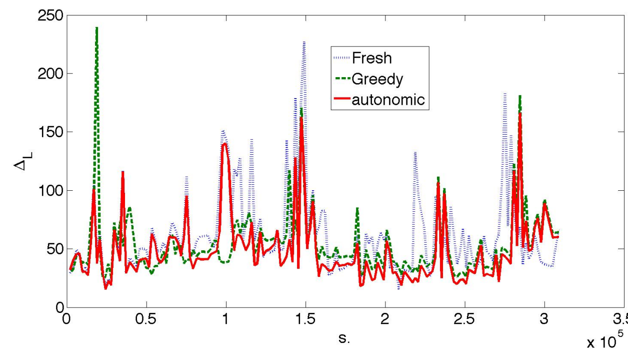


(b)

Fig. 7. Message delivery latency over time for (a) HCMM_det5 and (b) HCMM_pro1.



(a)



(b)

Fig. 8. (a) Coverage, and (b) message delivery latency over time for PMTR.

the environment – and later decreases toward that of the best algorithm.

Finally, we measured the performance of the autonomic algorithm with the real traces. In this case, Fresh and Greedy have very similar performances. The autonomic algorithm tends toward the behavior of Fresh, thus achieving acceptable coverage (fig.8(a)) and the best latency (fig.8(b)).

V. CONCLUSIONS

In this paper, we propose a novel *autonomic* forwarding algorithm for ONs, where each node autonomously discovers the characteristics of its basic everyday environment – in terms of closeness of the communities and presence of travelers – and decides in accordance what is the best rank-based forwarding policy to adopt to maximize performance. The results obtained through simulations show that the approach is promising: with high probability, nodes correctly learn their role and the nature of the scenario in slightly more than a day. This information is used afterwards to appropriately rank candidate message relays, so obtaining a very good performance in terms of both destination coverage and latency. As a consequence, we plan to perform further measures with real traces in order to further validate the approach and gain better insight into the dimensioning of the parameter Δ_{th}. We also intend to investigate alternative policies for the determination of node roles, so as to achieve a quicker and more accurate characterization of the current scenario and also a better consistency among the views of different nodes. Designing these policies in a successful way may lead in the future to extending the proposed algorithm. The ultimate aim of research along these lines is to allow the nodes to *dynamically adapt* to changing scenarios while the users go about their daily business, moving from location to location.

ACKNOWLEDGMENTS

The authors thank Chiara Boldrini for supplying the HCMM code. The funding for this research came partially from the Italian Ministry of Education, University and Research under the PRIN PEOPLENET (2009BZM837) Project and partially from the European Commission under the FP7 SCAMPI (258414) Project.

REFERENCES

[1] C. Boldrini, M. Conti, A. Passarella. *Users Mobility Models for Opportunistic Networks: the Role of Physical Locations*. In Proc. IEEE WRECOM, 2007, pp.1-6.

[2] V. Borrel, F. Legendre, M. De Amorim, S. Fdida. *SIMPS: Using Sociology for Personal Mobility*. IEEE/ACM Transactions on Networking, 17(3), 2009, pp. 831-842.

[3] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, J. Scott. *Impact of Human Mobility on the Design of Opportunistic Forwarding Algorithms*. Proc. IEEE INFOCOM 2006, pp. 1-13.

[4] CRAWDAD- A Community Resource for Archiving Wireless Data At Dartmouth. http://crawdad.cs.dartmouth.edu/

[5] E. Daly, M. Haahr. *Social Network Analysis for Routing in Disconnected Delay-Tolerant MANETs*. In Proc. MobiHoc 2007, pp. 32-40.

[6] H. Dubois-Ferriere, M. Grossglauser, M. Vetterli. *Age matters: efficient route discovery in mobile ad hoc networks using encounter ages*. Proc. ACM MobiHoc 2003, pp. 257-266.

[7] V. Erramilli, A. Chaintreau, M. Crovella, C. Diot. *Diversity of forwarding paths in pocket switched networks*. Proc. ACM/SIGCOMM IMC (Oct.2007), pp. 161-174.

[8] S. Gaito, E. Pagani, G.P. Rossi. *Strangers help friends to communicate in opportunistic networks*. Computer Network 55(2), Elsevier, February 2011, 374-385.

[9] W. Hsu, T. Spyropoulos, K. Psounis, A. Helmy. *Modeling Time-Variant User Mobility in Wireless Mobile Networks*. In Proc. IEEE INFOCOM 2007, pp.758-766.

[10] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, C. Diot. *Pocket switched networks and the consequences of human mobility in conference environments*. Proc. SIGCOMM 2005 Workshop on Delay Tolerant Networking.

[11] P. Hui, J. Crowcroft, E. Yoneki. *Bubble rap: social-based forwarding in delay tolerant networks*. Proc. 9th ACM Int'l Symp. on Mobile Ad Hoc Networking and Computing, May 2008.

[12] T. Karagiannis, J. Le Boudec, M. Vojnoviċ. *Power law and exponential decay of inter contact times between mobile devices*. Proc. 13th ACM MobiCom, Sept. 2007.

[13] K. Lee, S. Hong, S. J. Kim, I. Rhee, S. Chong. *SLAW: A Mobility Model for Human Walks*. In Proc. IEEE INFOCOM 2009.

[14] A. Lindgren, A. Doria, O. Schelnen. *Probabilistic Routing in Intermittently Connected Networks*. In Proc. 1st International Workshop on Service Assurance with Partial and Intermittent Resources, 2004, pp. 239-254.

[15] A. Mei, J. Stefa. *SWIM: a Simple Model to Generate Small Mobile Worlds*. In Proc. IEEE INFOCOM 2009.

[16] M. Musolesi, C. Mascolo. *Designing mobility models based on social network theory*. In ACM SIGMOBILE Mobile Computing and Communications Review 11(3), ACM Press (2007).

[17] E. Pagani, G.P. Rossi. *Utility-based Forwarding: a Comparison in Different Mobility Scenarios*. In Proc. Third International ACM Workshop on Mobile Opportunistic Networks (MobiOpp), 2012.

[18] L. Pelusi, A. Passarella, M. Conti. *Opportunistic Networking: data forwarding in disconnected mobile ad hoc networks*. IEEE Communications Magazine 44(11), November 2006, pp. 134-141.

[19] T. Phe-Neau, M. Dias de Amorim, V. Conan. *Vicinity-based DTN Characterization*. In Proc. Third International ACM Workshop on Mobile Opportunistic Networks (MobiOpp), 2012.

[20] J. Su, A. Chin, A. Popivanova, A. Goel, E. de Lara. *User mobility for opportunistic ad hoc networking*. Proc. IEEE WMCSA 2004.

[21] A. Vahdat, D. Becker. *Epidemic Routing for Partially Connected Ad Hoc Networks*. Technical Report CS-200006, Duke University, April 2000.