Simulating the statistics of the first meetings using dynamic "open environments"

Lorenzo Bracciale, Pierpaolo Loreti, Giuseppe Bianchi
Department of Electronic Engineering
University of Rome "Tor Vergata" Rome, Italy
{lorenzo.bracciale,pierpaolo.loreti, giuseppe.bianchi}@uniroma2.it

***Abstract*—Algorithms and protocols for opportunistic, delay tolerant and wireless ad-hoc networks are designed and validated by simulating the people interactions induced by the nodes mobility. There are cases in which we are interested in simulating just the first interaction between a pair of nodes, for instance to assess the performance of a discovery or epidemic routing protocol. In this cases nodes rapidly extinguish their utility hence it is not convenient to simulate these scenarios using a fixed amount of nodes. Thus we present a novel simulation methodology that introduces the "open environment" where nodes enter, can interact through meeting with other nodes and then exit, keeping the focus on the environment rather than on what happen before and after the nodes stay in the interesting area. The proposed approach uses the statistical distribution extracted from the real traces to reproduce directly the human interaction pattern without going through the traditional random way point approach. Meetings are simulated by a time-varying graph that holds the state of the interactions in the environment, while adapting to the statistics of single node to its history. We show that even in a simple scenario, the epidemic infection, Markov memory-less models have been fairly far from the interaction scenarios that the method reproduces.**

I. Introduction

Human mobility simulations play an important role for a wide range of application scenarios in the fields of ad-hoc and opportunistic networking where mobile nodes can exchange data only when they are near enough to allow a wireless communication. In fact, the standard way to validate or assess the performance of systems, protocols and algorithms relies on simulating the interactions of the people devices by a "contact pattern" that describes when the communication between two nodes is possible. Contact patterns are usually derived from a mobility patterns that can be created in two ways: i) by synthetic mobility models (such as the popular random waypoint and its variant [1] [2] [3]) or ii) using real mobility traces [4].

However there are some scenarios when we are interested in simulating *just the first interaction between a pair of nodes*. Suppose for instance that we are interested in the performance assessment of a an epidemic routing protocol (such as [5]) where an information carried by an "infected" mobile node is spread to all the nodes it encounters that in turn become infected too and participate to the data dissemination process. Clearly, in this case, simulation methods that use a fixed number of nodes can not lead to an interesting stationary situation because after a given time the 100% of the nodes become infected. Indeed, as the nodes encountering process goes on, nodes rapidly extinguish their utility for the simulation purposes and they tend to be useless. [1] However if new nodes continuously enter and exit from the simulated area, they could balance this loss of utility introduced by the protocol operations.

For this reason, in this work we present a novel simulation methodology of contact patterns that introduces the concept of "open environments": a place where nodes enter, can interact with other nodes and exit. By this interaction, nodes could change the behavior of other nodes before exiting the system. Even if nodes permanence in the environment could be very short in time, and their interaction with other nodes could be very marginal, they contribute in changing the overall system information that could eventually converges to a non trivial "environment steady state" (as will be presented and discussed in V). In this way, the open environments can easily simulate the transitory and steady state of dynamic place such as a subway station or a city square, where we are more interested in *focusing on the state of the environment rather than on what happen before and after the nodes stay in the interesting area*.

Moreover in our approach we use the statistical distribution extracted from the real traces to infer directly the human interaction patterns without simulating nodes movement. Contacts are simulated by a time-varying graph that keeps track of the previously encountered nodes. Based on that, an analytical procedure is devised to make new encounters happen in a way that each node in the system experiments new contacts with a temporal distribution derived by a stationary Poisson process. In this way we can tune the rate of the encounters *as seen by each node*. With this methodology able to simulate a wide range of collaborative protocols and algorithms based on the popular assumption that each node meets other nodes with a given rate and that the inter-meeting time is exponentially distributed [6] [7] [8], [9], [10].

The main contributions of this works are: i) provide a novel analytical framework to reproduce new nodes meeting pattern according to statistic derived by mobility traces ii) describe the simulation operations iii) provide insights on the interaction procedures in the open environment through a simple epidemic routing scenario.

This work is organized as follow: in section II we compare our methodology with the state of art, in section III we introduce the concept of the open environment and the mathematical frameworks behind the simulator that is describe

[1] A similar case arose when we want to simulate a discovering protocol on a opportunistic network: after a given amount of time all the nodes in the environment will be discovered.

in section IV. In section V we simulate the epidemic routing scenarios using the proposed method and compare it with a standard approach. Finally conclusions are drawn.

II. Related works

A very popular way to validate distributed algorithms and protocols that runs on mobile devices against real cases is to use real human mobility traces that have been collected in several experiments such as from the HAGGLE project [11] or from MIT reality mining [12] and others [4].

However the performance results obtained in this way are tightly bound to the particular environment where the traces are logged and the relative acquisition techniques. Moreover, the great part of these traces exhibits a very coarse granularity: for example in [12] traces are obtained using a Bluetooth scan every 5 minutes, and in [13] every 120s. This is a consequence of the particular measurement technique adopted that is typically based on periodic Bluetooth scans performed on a set of hand-held devices. For this reason these traces could not be suited when we are interested with very short contact durations (e.g. ¡ 20s). This problem has been tackled in [14] and overcome by using the IEEE 802.15.4 to records the encounters.

Another viable solution is to use synthetic mobility model in which the most popular is the Random Waypoint model that works as follow: users are randomly created and moves inside a given area (usually a square area or a torus for avoiding border effects). Each user repeats these operation cyclically: i) decides a random destination ii) moves toward this point with a random velocity iii) reached the destination, waits of a certain amount of time. If two terminals get close enough, depending to the simulated radio technology, a meeting can occur. Such kind of models is simple to implement and offers a good control on the simulation parameters since it is devised completely by the area, the speed statistics and waiting times. However there is a debate on the ability of this methodology in reproducing real human pattern [15].

Then there have been some efforts in adapting RWP to reproduce the pattern derived from real mobility traces such as [16], [1].

To asses the impact of connectivity pattern, the focus can be moved from the mobility model to the connectivity model, that focuses on the evolution of the emergent connectivity graph that is changing over time as nodes move rather than the geographical information over time [1].

The direct usage of statistics has been proposed in [17] and in [18]: both works start from the generation of a fixed number of nodes and then, applying two different methodologies, they try to reproduce the connectivity patterns mimic the real traces statistics. [17] applies these statistics to a time-varying graph that models possible interactions among terminals in a specific hot spots. On the contrary in [18] Tan et al. first generates the number of meetings that each node experiences during the simulation and after they generates these meetings according to a target distribution. In any cases both methodologies can be considered as classical and can not be used for reproducing the statistics of the process of first interactions since they use a closed environment where a fixed amount of nodes stands for all the duration of the simulation.

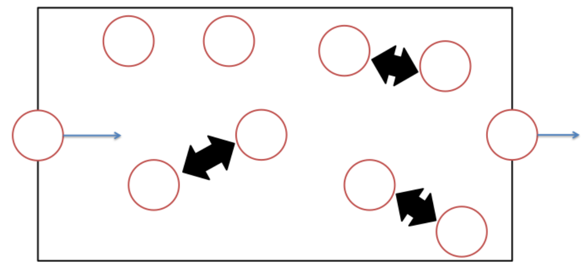

Fig. 1. The simulated open environment, where nodes can enter and exit. Inside the environment nodes interact each other when a meeting occurs.

III. Dynamic open environments

A. *Why the environment has to be open?*

The *open environment*, depicted in figure 1, is a conceptual place where nodes enter **can** interact through meetings with other nodes in the environment and then exit. By interactions, nodes could change the states (hence the behavior) of the other nodes in the environment that in turn could interact with other nodes before exiting the system. Even if the permanence of the nodes in the system is very short, and their interaction with other nodes very marginal, they contribute to the change of the overall information contained in the environment that could eventually converges to a steady state even if nodes does not have enough time to converges to this state by their own.

To clarify, let consider for instance the case of a distributed algorithm that runs on users smartphones and suppose that we are interested of assessing performance of a discovery protocol for the case of a subway scenario. Even if users enter and exit from a subway train rapidly, they can interact each other to share for instance the statistical characteristic of the environment such as the distribution of the inter-meeting times or the average contact duration. This information will persist in the environment because we can suppose that, with high probability, there are always users that are waiting for the trains. Hence, when new users arrive, they can acquire this information and collectively upgrade it by active measurements.

Thus the purpose of this simulation methodology is to provide the tools to analyze the overall state of an environment created by the interactions of the nodes that flow rapidly through it.

To define the *open environment*, we need to characterize the flow of the nodes and the interactions between them.

B. *The node flow*

The flow of nodes in the environment is specified by the distribution of inter-arrivals time and by the distribution of the time of nodes permanence in the system. To easy characterize the arrival process, we considered a Poisson process: the inter-arrivals time of new nodes in the environment are exponentially distributed with rate λ_e. We assume that the nodes remain in the environment for a certain time that can be constant or exponentially distributed with mean $1/\mu_e$.

Thus, at the steady state we have on average $N_e = \lambda_e/\mu_e$ nodes in the environment (the popular Little result). We point out that the number of node in the environment is a key performance parameter in many scenarios.

C. Node's interactions

Reproducing the inter-meeting time process is more tricky since we need to simulate that each node in the environment meets *new* nodes with an exponentially distributed inter meeting time with parameter λ_u.

This is a very popular analytical assumption [6] [7].

We point out that literature studies such as [8], [9], [10], proved that inter-contact time distribution between two pairs of node can be assumed as exponential. In particular [8] have analyzed several popular real world traces showing that 85% of the pair distributions fit an exponential law according to χ^2 test.

Moreover both [9], [10], have demonstrated that is not in contrast with the well known heavy tailed distribution (with or without the exponential cut-off) of the *aggregated* inter meeting time. Simulating such hypothesis is not straightforward if we use the RWP because it does not offer a direct control on the encounters as seen by each node.

However, a simple generation of inter-meeting times as exponential random variable with a constant mean will not result in a correct behavior because: i) each encounter involves *two* nodes ii) a node should meet only *new* nodes (i.e. nodes that are not met before with that node) given that we are not interested in multiple encounters of the same pair of nodes but just in unique contacts.

To address this problem, we keep track of the encounters in the simulation by a time-varying graph where nodes are the nodes in the environment and edges, with cardinality L, are the "possible meetings", i.e. the meetings between two nodes that did not met before. The information contained in this graph and the markovian propriety of the encounter process allows designing the following two step procedure to pinpoint when the next meeting occurs and which nodes will be involved.

First we generate a set of exponentially distributed random variables, one for each link in the graph. Each of these variables has a parameter λ_i calculated as explained below. Then we select the minimum of those realizations that specifies the time of the next meeting and the nodes involved (the ones incident with the associated link).

Given that the minimum of a set of exponentially distributed random variables with parameter $\lambda_1, ..., \lambda_k$ is another exponential random variable with parameter $\lambda^* = \lambda_1 + ... + \lambda_k$, we impose that the sum of λ_i for all the edges incident to a generic node must be λ_u, for all the nodes in the environments. In other words, we extract one exponential random value per link in a way that each node sees an exponential RV with parameter λ_u even if some of its connections have been previously removed. To calculate all the λ_i we need to solve the following system of equations:

$$\mathbf{\Lambda_n} = \mathbf{A}\mathbf{\Lambda_l}$$

where:

- $\mathbf{\Lambda_n}$ is a vector of dimension N (nodes in the environment) with all the elements equal to λ_u [2]
- $\mathbf{\Lambda_l}$ is a vector of the links in the graph, with dimension $N(N-1)/2 \geq L$ and where the i-th element λ_i represents the parameter of the RV associated to the i-th link.
- $\mathbf{A}$ is a matrix with dimensions $N \times L$ and defined as:

$$a_{i,j} = \begin{cases} 1, & \text{if the node j-th is an endpoint for the link i-th} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Solving the system of equation by matrix inversion, we obtain the values for the vector $\mathbf{\Lambda_l}$ and then we generate a set of exponential random variables whose parameters are taken from $\mathbf{\Lambda_l}$. By taking the minimum values, we define the time when the next meeting will occurs and the nodes involved in that encounter. This procedure must be repeated i) when a node enters the environment because the possible meeting list must be updated; ii) when a node exits from the environment because all the possible meeting with that node must be removed; iii) when a meeting occurs because the selected link is removed.

Using this solution we succeed in simulating a stationary stochastic environment where each node meets other nodes with an exponential inter meeting time with mean $1/\lambda_u$. The encounters duration last D seconds where D is a RV distributed according to a generic cdf $F_D(t)$. For example, the distribution of duration can be a uniform, exponential or Pareto.

IV. Simulating an Open Environment

A. The simulator architecture

The simulator is implemented in python and depends on the mathematical library Numpy that is used for the inversion of the matrix. The simulator is event based and is organized into the following classes:

- The *Simulator* represents the main class that provides a configuration interface and is used for control the execution.
- The *Calendar* handles the simulation scheduling and provide methods to manage an ordered list of events
- The *Event* is a scheduled action that runs at a particular time; examples of events are the arrival and departure of nodes in the environment, meetings among nodes, measurements.
- The *Open Environment* provides the operations to manage the encounters among nodes and integrates all the statistical models.
- The *Node* is the entity that enters into the environment and interacts with other nodes and consequently changes its own state. The Node can also include the protocols that have to be simulated when a contact occur.

The following events are modeled in the simulator: the start of simulation, a node entering the environment, a node exiting the environment, the start of interactions, the start of a meeting,

[2] The proposed solution can also works for the case where each node see a different λ_u, simulating nodes that moves faster and ones that moves slower

the end of a meeting and the end the simulation. Moreover the simulator includes some special measuring events that are used to periodically monitor relevant parameters.

B. The simulator operations

The main goal of the simulator is to reproduce the environment as described above, and to test how the proposed algorithm performs. In particular, the simulator:

- Reproduce the process of users that enter and exit the environment
- Reproduce the process of users that "physically" meet each other inside the environment

Arrival and departure events from the open environment of all nodes are generated at the beginning of the simulation and added to the calendar. The generation of the meeting starts after a certain time T_0 defined by the simulator configuration: the first meeting event is generated at time $T_1 = T_0 + \Delta_1$ using the procedure illustrated in Section III for the calculation of Δ_1 and repeated for finding the subsequent Δ when the possible meeting graph is updated because of that meeting. We point out that each Δ is a exponential RV hence the memoryless propriety allows an event to be cancelled. In fact the procedure is repeated also when graph changes because nodes enter and exit to/from the open environment.

If the matrix $\mathbf{A}$ is singular we generate the next meeting randomly extracting a link with rate $\lambda_u * N_e/2$. This fallback procedure should happen very rarely (or never happen at all) if the simulator configuration parameters are properly set (e.g. $\lambda_e > \lambda_u$ otherwise graph edges are pruned too fast and the associated the matrix $\mathbf{A}$ become singular).

Each time there is a meeting event the meeting duration is calculated generating a RV according to the specified distribution. If a departure event occurs for a node that has an on going meeting, the node is removed from the graph to avoid its selection in new meetings but a new departure time is computed for that node just after the end of the meeting in order to preserve the statistical distribution of the contact duration.

C. Insights

In Figure 2, each node tries to estimate the mean inter-contact rate $\widehat{\lambda_u}$ for new contacts and we plot the mean absolute error $|\lambda_u - \widehat{\lambda_u}|$. As we can see, the more the time nodes stay in the environment (whose mean is $1/\mu_e$), the smaller is the error that could be also significantly different from the "real" value if the nodes do not have enough samples to reduce the variance of the estimated parameter. This is a challenging and interesting scenario that simulates the case of a fast environment against which we could validate several distributed algorithms whose goal could be to make nodes cooperate for a global estimation given that they can not estimate environment parameters individually. Given that we simulate just the new contacts (i.e. encounters with nodes without repetitions), we must remove a pair of nodes from the possible contacts graph. In figure 3 we show the number of possible contacts among node on the total number of pair available $N(N-1)/2$. As we can see the average available

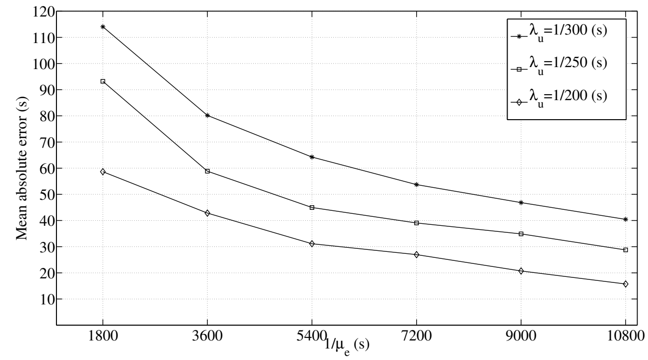

Fig. 2. Mean absolute error of the mean λ_u as seen by nodes, varying new contact rate λ_u and the mean time inside the environment $1/\mu_e$

encounters selectable by the simulator decrease as the λ_u grows, forcing each node to select a biased list of nodes. This is needed to preserve the stochastic proprieties on the nodes that must encounters only new nodes with rate λ_u.

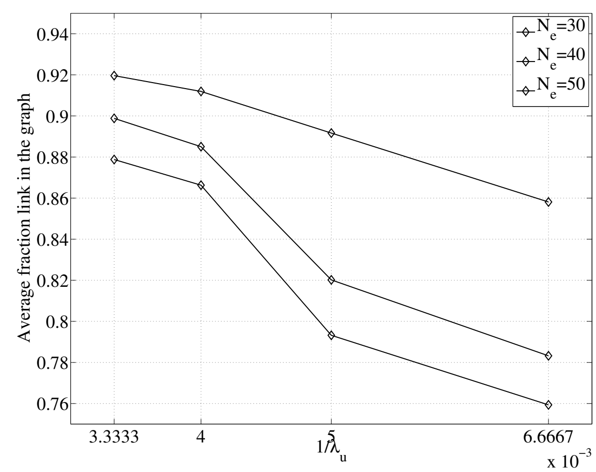

Fig. 3. Number of possible contacts between nodes, on the total number of nodes pairs in the network.

V. Application scenario

As said, the proposed simulator could be effectively used in several scenarios where we are interested in studying the effects on some distributed algorithm in a open environment. In this section we present a very simple scenario where we use the simulator for evaluating the persistence of an information that is spread with simple epidemic routing algorithm similar to [5].

The algorithm works as follow:

- Each node can be in one of two states: *Infected* and *Not Infected*
- When an infected node encounters a not infected node, the last one becomes infected with probability α
- Initially, just one node is infected

The simulated environment mimics an high dynamic environment (such as the subway station or a crowded street) with the following parameters:

- Nodes enter in the environment with rate λ_e
- Nodes stay in the environment with average $1/\mu_e$
- Each node encounters *new* nodes with rate λ_u
- All the random variables are distributed according to an exponential distribution

Figure 4 shows the ratio of infected nodes during the 70000 seconds of simulation time, varying $\alpha = [1, 0.3, 0.1]$. The parameters of the simulation are $\lambda_e = 1/45s^{-1}$, $\mu_e = 1/1800s^{-1}$, $\lambda_u = 1/60s^{-1}$ while the encounter duration does not affect the result of the simulation.

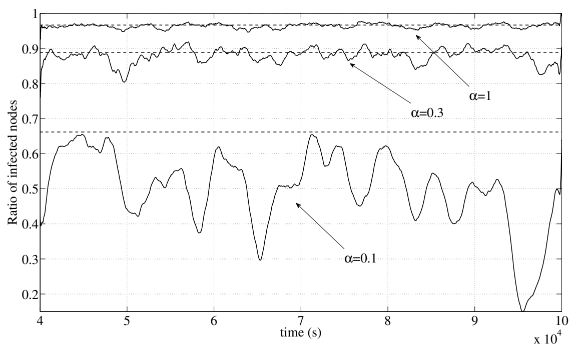

Fig. 4. Ratio of the infected nodes varying the probability of infection $\alpha = 1, 03, 01$

Even in this simple case, the presented results present a significant difference from the case when each node encounters *any other* nodes with a given rate λ.
This latter case is far easy to simulate because it suffices to periodically extract a pair of nodes every time T distributed as an exponential random variable with rate $2\lambda/N$ where N is the number of node in the system at a given time. Indeed, if every node should experiment a new encounter with rate λ, every possible encounter (that are $N(N-1)/2$) can be represented as an exponential random variable X_i with rate $\lambda/(N-1)$. Then, using the memoryless propriety of the exponential distribution, we can write a discrete event simulator where we periodically generate a RV with rate equals to $min(X_1 \cdots X_{N(N-1)/2})$ that corresponds, for the propriety of exponential random variable, to the rate $2\lambda/N$. However this case can be easily investigated also without any simulator, and directly by means of an analytical approach. For this reason and to pinpoint the difference between these two cases we models it with a two-dimentional Markov chain depicted in figure 5.

Each state of the chain is described by the tuple (n, m) where n are the node in the environment and m are the number of infected nodes. A transition occurs when:

- new node enter in the environment. This happens event with rate λ_e.
- nodes exit from the environment. Each node stay in the environment for a time distributed as a RV with mean $1/\mu_e$

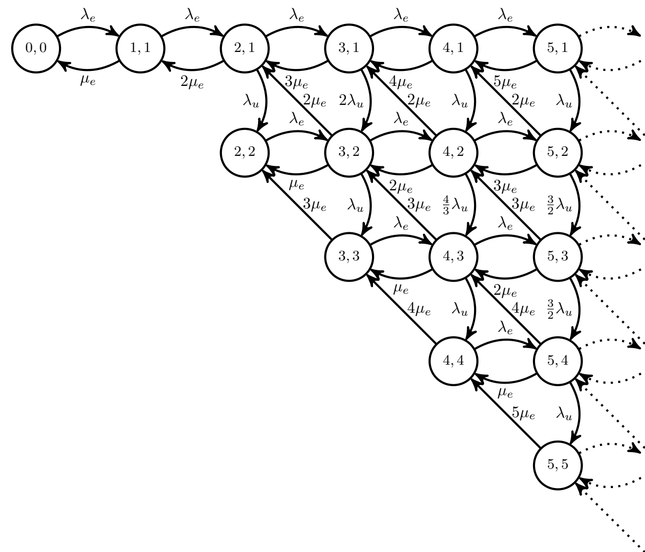

Fig. 5. Bi-dimentional Markov chain representing the infection process in the simulated open environment.

- new node has been infected. These are the vertical transitions that are related to the encountering rate where the relation is $\alpha\lambda_u\frac{m(n-m)}{n-1}$ and is obtained by the above-mentioned ratio of encounter selection $2\lambda_u/n$ times the probability that the encounter involves an infected and a non infected node that is $\frac{2m(n-m)}{n(n-1)}$.

We deliberately impose that if there is only one node in the environment, it must be an infected node. We did this to avoid the presence of an absorbing state and for studying the steady state of the process. Figure 6 shows the difference in terms of infected nodes between this approach and the simulated one. Despite in both cases a node experiment encounters with rate λ_u, as we can see the discrepancy of the results are far from being marginal for an average network with 40 nodes.
Hence, encountering just new nodes with a fixed statistic is more difficult to achieve either analytically because of the memory of the already encountered nodes, and via simulations, as explained in section IV, however it is worth for all the case in which we must reproduce such analytical assumptions.

VI. Conclusions

In this work we presented a methodology for simulating contacts among nodes that satisfies this general constraint: each node must encounter a new nodes every time T that is a random variable distributed according to an exponential distribution with mean $1/\lambda_u$. For doing so, we had to simulate an "open environment" that is an environment where nodes enter and exit while insides the environment nodes meet each other so that the given statistic assumption is preserved.

We provided the mathematical procedure for contact selections as well as the details on the simulator implementation and architecture. With the proposed methodology we can simulate fast mobility in small or large environments and study the evolution of the environment characteristic that can converge

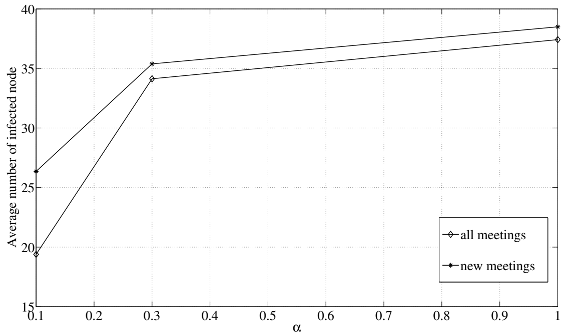

Fig. 6. Average number of infected nodes varying the probability of infection α. Figure shows the difference of considering λ_u as the ratio of all the meetings between one node and any other node, or just between one nodes an other unmet nodes.

to a steady state also if nodes do not because they can stay in the environment just for a small amount of time. This scenario together with the well-defined stochastic proprieties of nodes encounters pave the way to assess performance of distributed algorithms such as distributed tolerant network routings and discovery algorithms.

References

[1] M. Musolesi and C. Mascolo, "Mobility models for systems evaluation," in *Middleware for Network Eccentric and Mobile Applications*, B. Garbinato, H. Miranda, and L. Rodrigues, Eds. Springer Berlin Heidelberg, 2009, pp. 43–62. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-89707-1_3

[2] L. Bracciale, D. Battaglino, A. Detti, G. Bianchi, N. Blefari-Melazzi, A. Bragagnini, and M. Turolla, "Delay performance of a publish subscribe system deployed over a memory-constrained, delay tolerant network," in *Ad Hoc Networking Workshop (Med-Hoc-Net), 2011 The 10th IFIP Annual Mediterranean*, June 2011, pp. 25–32.

[3] A. Detti, C. Loreti, and P. Loreti, "Effectiveness of overlay multicasting in mobile ad-hoc network," in *Communications, 2004 IEEE International Conference on*, vol. 7, June 2004, pp. 3891–3895 Vol.7.

[4] D. Kotz and T. Henderson, "Crawdad: A community resource for archiving wireless data at dartmouth," *Pervasive Computing, IEEE*, vol. 4, no. 4, pp. 12–14, Oct 2005.

[5] E. Yoneki, P. Hui, and J. Crowcroft, "Wireless epidemic spread in dynamic human networks," in *Bio-Inspired Computing and Communication*, ser. Lecture Notes in Computer Science, P. Li, E. Yoneki, J. Crowcroft, and D. Verma, Eds. Springer Berlin Heidelberg, 2008, vol. 5151, pp. 116–132. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-92191-2_11

[6] W. Gao and Q. Li, "Wakeup scheduling for energy-efficient communication in opportunistic mobile networks," in *Proceedings of the 32th IEEE Conference on Computer Communications*, ser. INFOCOM, 2013.

[7] L. Bracciale, P. Loreti, and G. Bianchi, "Human time-scale duty cycle for opportunistic wifi based mobile networks," in *Digital Communications - Green ICT (TIWDC), 2013 24th Tyrrhenian International Workshop on*, Sept 2013, pp. 1–6.

[8] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: a social network perspective," in *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, ser. MobiHoc '09. New York, NY, USA: ACM, 2009, pp. 299–308. [Online]. Available: http://doi.acm.org/10.1145/1530748.1530790

[9] A. Passarella and M. Conti, "Characterising aggregate inter-contact times in heterogeneous opportunistic networks," in *Proceedings of the 10th international IFIP TC 6 conference on Networking - Volume Part II*, ser. NETWORKING'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 301–313. [Online]. Available: http://dl.acm.org/citation.cfm?id=2008826.2008855

[10] V. Conan, J. Leguay, and T. Friedman, "Characterizing pairwise inter-contact patterns in delay tolerant networks," in *Proceedings of the 1st International Conference on Autonomic Computing and Communication Systems*, ser. Autonomics '07. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007, pp. 19:1–19:9. [Online]. Available: http://dl.acm.org/citation.cfm?id=1365562.1365588

[11] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD data set cambridge/haggle (v. 2009-05-29)," Downloaded from http://crawdad.cs.dartmouth.edu/cambridge/haggle, May 2009.

[12] N. Eagle and A. (Sandy) Pentland, "Reality mining: sensing complex social systems," *Personal Ubiquitous Comput.*, vol. 10, no. 4, pp. 255–268, Mar. 2006. [Online]. Available: http://dx.doi.org/10.1007/s00779-005-0046-3

[13] P. H. Augustin Chaintreau, "Pocket Switched Networks: Real-world mobility and its consequences for opportunistic forwarding," 2006 Computer Laboratory, University of Cambridge, Tech. Rep., Feb. 2005.

[14] Y. Wang, B. Krishnamachari, and T. Valente, "Findings from an empirical study of fine-grained human social contacts," in *Wireless On-Demand Network Systems and Services, 2009. WONS 2009. Sixth International Conference on*, 2009, pp. 153–160.

[15] J. Yoon, M. Liu, and B. Noble, "Random waypoint considered harmful," in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*, vol. 2. IEEE, 2003, pp. 1312–1321. [Online]. Available: http://dx.doi.org/10.1109/infcom.2003.1208967

[16] A. Jardosh, E. M. Belding-Royer, K. C. Almeroth, and S. Suri, "Towards realistic mobility models for mobile ad hoc networks," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '03. New York, NY, USA: ACM, 2003, pp. 217–229. [Online]. Available: http://doi.acm.org/10.1145/938985.939008

[17] R. Calegari, M. Musolesi, F. Raimondi, and C. Mascolo, "Ctg: A connectivity trace generator for testing the performance of opportunistic mobile systems," in *Proceedings of the the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering*, ser. ESEC-FSE '07. New York, NY, USA: ACM, 2007, pp. 415–424. [Online]. Available: http://doi.acm.org/10.1145/1287624.1287684

[18] F. Tan, Y. Borghol, and S. Ardon, "Emo: A statistical encounter-based mobility model for simulating delay tolerant networks," in *World of Wireless, Mobile and Multimedia Networks, 2008. WoWMoM 2008. 2008 International Symposium on a*, June 2008, pp. 1–8.