CTG: A Connectivity Trace Generator for Testing the Performance of Opportunistic Mobile Systems

Roberta Calegari, Mirco Musolesi, Franco Raimondi, and Cecilia Mascolo
Department of Computer Science
University College London
London, United Kingdom

ABSTRACT

The testing of the performance of opportunistic communication protocols and applications is usually done through simulation as i) deployments are expensive and should be left to the final stage of the development process, and ii) the number of varying parameters in these systems is so high that it would be very hard to conduct thorough testing of all the functionality within a single deployment. Therefore, protocols and applications are often plugged into mobility simulators to test their performance; however, until recently, most of the testing has been conducted with random mobility models which do not mirror reality. Furthermore, despite disconnections playing a very prominent role in the performance of any opportunistic mobile system, most models do not really account for it. A different approach to testing is the use of real traces of movement collected in specific domains as test cases. These cases, however, do not allow for flexible performance testing, as they are specific for a given scenario with fixed connectivity properties.

In this paper we propose the Connectivity Trace Generator (CTG), a tool for the automatic generation of connectivity traces, which takes as input real mobility traces and is able to output a set of traces with similar connectivity properties, which can be used as test cases. This allows developers to investigate the impact of the variation of connectivity patterns, number of hosts, and other parameters on the protocol or application under investigation.

We use a real case study (the Dartmouth campus connectivity traces) to show how CTG allows protocol developers to play with some connectivity and density parameters so to best conduct performance testing of different aspects of protocols and applications.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Store and forward network, Wireless communication; C.4 [**Performance of Systems**]: Modeling techniques

General Terms

Design, Verification

ESEC/FSE'07, September 3–7, 2007, Cavtat near Dubrovnik, Croatia.

Keywords

Delay tolerant networking, opportunistic systems, connectivity model, protocol testing.

1. INTRODUCTION

Opportunistic mobile systems [6] are decentralized distributed systems which account for the ability of the hosts to be disconnected from each others for some periods of time. Hosts act as carriers for the information, which is relayed from a source to a destination through a number of forwarding and carrying steps, following the connectivity patterns of the hosts. Applications of opportunistic mobile systems are growing in number and range from more social network related ones (e.g., news/content distribution among people travelling on public transport, or between vehicles) to more domain specific ones such as wildlife monitoring, remote villages connectivity, military scenes and rescue operations. Decentralization can be motivated by many factors including lack of total coverage (in remote or extensive areas) or by the ability to offer a free service by avoiding a centralized infrastructure. The testing of these systems is usually performed through simulations, given the considerable number of entities involved, moving in possibly large geographical areas.

Despite their soaring number, the performance of opportunistic, and, in general, mobile systems is still tested with very rudimentary techniques with little attention to mobility aspects. In particular, the core protocols at the basis of the communication among the hosts are often tested using non realistic models, including random mobility models, such as the very popular Random Way Point model [9]. However, the performance analysis of protocols using random mobility models may provide inaccurate information and give wrong insights into the real applicability of protocols and applications. For this reason, recent years have seen the proliferation of mobility traces collected in realistic mobile applications including students patterns in campuses [7], people attending conferences [8] and cities and streets circulation [15]. Repositories have also been created to collect all these measurements (e.g., CRAWDAD Project [10] at Dartmouth College), which can be used as test data. However, no matter how many traces can be collected, this will always look like a small amount, in many cases insufficient, with respect to the variation needed for a thorough performance testing of these systems (for example with respect of the number of nodes/density).

A number of pioneering works [1, 2, 7, 19] have studied traces in order to gain insight about the real mobility patterns. A key study in this area is the work on connectivity patterns presented by Chaintreau et alii in [5] which illustrates the fundamental insight that contacts duration and inter-contacts time between individuals are distributed according to power-law distributions[1] and that these patterns may be used to develop more efficient opportunistic protocols.

In this paper we present a novel Connectivity Trace Generator (CTG) tool for testing opportunistic mobile systems. The tool is based on a model of connectivity presented in the workshop paper [4]. We validate our tool through a case study using the Dartmouth traces [11] as input to CTG, generating multiple traces but for different number of hosts and demonstrating that they have similar connectivity patterns (see Section 4). Our work differs from previous approaches in that probability distributions describing the patterns of colocation of mobile users are exploited for the first time as *direct inputs* of a tool. The distribution of the average number of people that an individual meets during a certain period of time (e.g., a day) is also an input of the tool. All these distributions can be extracted by measurement of connectivity on real traces.

By using the proposed model and the CTG tool we show how researchers are able to conduct performance testing and gain a better understanding of opportunistic communication protocols and applications. Effectively, this is done by generating synthetic connectivity traces through the *variation* of a number of parameters, while maintaining the same distributions observed in the input traces. Synthetic traces can then be used as test cases for performance testing of some existing opportunistic protocols. Notice that with a fixed set of traces, as the one used in input, this analysis would not be possible; our approach allows the detection of protocol behaviours which could not be discovered by the use of real traces only.

We present an example of this process using the Dartmouth traces and by testing the performances of the pure epidemic protocol proposed by Vahdat and Becker [20], the CAR protocol [13] and of a random choice based protocol (see Section 6).

To summarise, the contributions of this work are the following:

- we design and implement the Connectivity Trace Generator (CTG) tool which, from real traces, generates synthetic realistic traces;
- as a case study, we start from real traces and we verify that the synthetic traces generated from these using CTG have the same patterns of connectivity;
- we demonstrate how the synthetic traces can be used as test cases in conducting performance testing of some existing opportunistic protocols.

2. THE APPROACH AT A GLANCE

The key steps of our approach are depicted in Figure 1.

- **Derivation of Connectivity Distributions from Real Traces.** The input of CTG is a set of real traces. These are processed by a trace analyser to generate the parameters required by the tool. Additionally, a range of variations for the parameters is provided in input. As a concrete case study, we used the log session traces of the campus WLAN of Dartmouth College [11], to obtain empirical distributions for residence time, colocation and degree distribution of the nodes. The traces were collected by researchers at Dartmouth College from April 2001 to June 2004. The network is composed of 450 access points over an area of about 200 acres. The total number of users logged in these traces is 13889. These traces can be employed in conjunction with the model presented in Section 3. This model aims at representing the properties of the *colocation* of two users as a function of the probability for a user of being in a specific place for a given time. We will refer to this duration as *residence time* in the reminder of this paper.

  The analysis of the traces is presented in Section 5 and is performed using a number of scripts of CTG.

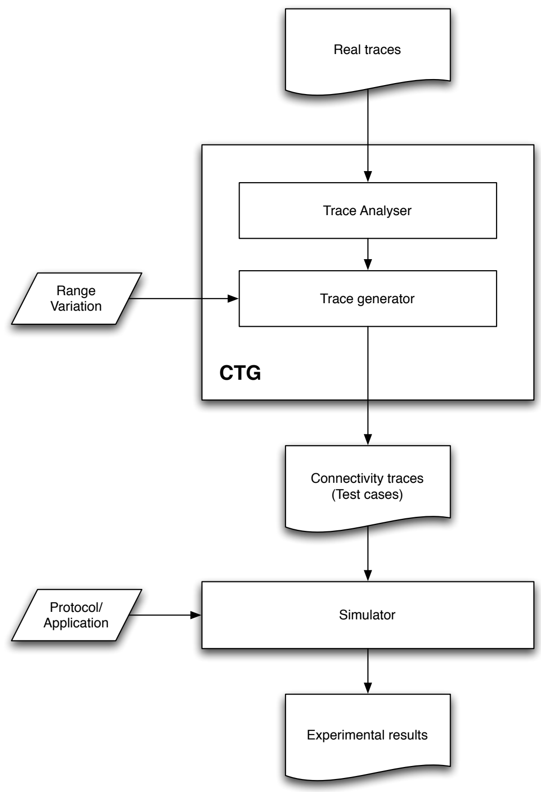

Figure 1: Connectivity Trace Generator.

- **Trace Generator.** Based on the connectivity model, the trace generator component of CTG allows for the generation of synthetic traces. The input parameters of this component are the relevant parameters of the connectivity model, namely: number of nodes, the contacts duration (i.e., the time interval in which two devices are in radio range) and inter-contact time (i.e., the time interval between two contacts), and node degree (i.e., number of neighbours) distributions.

  The process of generation is based on the selection of the desired number of hosts and on the construction of a connectivity graph of all the potential contacts of each host. In other words, we map each host to a node of the graph and we link a pair of nodes with an edge if the two hosts have a potential of becoming in contact. The connectivity graph is then used to unfold a number of connection links between users for each time instant. In other words, we use the connectivity graph

[1] Power-law distributions are characterised by the following form:

$$P(x) = x^{-k}$$

with $k \geq 0$.
A power-law distribution is also called scale-free since it remains unchanged to within a multiplicative factor under a re-scaling of the independent variable x [16].

as a basis for a *time-varying graph of instant connectivity* for each instant t. In these time-varying graphs (one for each time instant), each link is either *active* if the two hosts are colocated, or is not present if the two are not.

This process is completely automated and implemented by the trace generator component that produces traces containing the events of connections and disconnections for each pair of nodes of the simulation scenario and the time of each event. These traces can be used as test cases for the testing of opportunistic mobile systems (see Section 6). A detailed description of the implementation is presented in Section 4.

- **Performance testing of Opportunistic Protocols.** In order to show how the synthetic traces can be used as test cases for the performance testing of opportunistic communication protocols, we considered the following protocols: Flooding, Epidemic routing [20], CAR [13] and a Random-choice based carrier selection protocol. Our evaluation concentrates on message delivery, overhead, and delay as metrics by varying the contacts patterns. We implemented the protocols using the OMNeT++ discrete-event simulator [21] The results of this evaluation are discussed in Section 6.

3. THE CONNECTIVITY MODEL

In this section we review a connectivity model to be used in conjunction with our tool. A more detailed description can be found in [4]. At the core of the model is the methodology for computing the probability distribution of colocation (i.e., connectivity) times between two users, starting from a minimal data set (see below). The definition of the model is based on some simplifying assumptions:

- users' behaviours are *independent*; this means that we assume the behaviour of a user does not depend on other users' behaviours.
- users' behaviours are *uniform*: all users have the same behaviour.

As shown in [4], these assumptions are sufficient to capture the real connectivity patterns. A refined model could take into account users' degree of correlation in order to model non-uniform or non-independent users' behaviour.

We denote with X and Y two random variables for the duration of the sessions of two generic users a and b, respectively. The probability that a user a will remain in a given location for a time t (i.e., the residence time) is given, under our assumptions, by a probability density function $p_X(t)$; all users' behaviour is described by the same distribution $p_X(t)$. $p_X(t)$ is interpreted as the probability that the residence time will last t seconds.

In addition to the distribution $p_X(t)$, we assume that a probability density function $p_R(t)$ is available, representing the probability that the temporal distance between the beginning of two sessions of two colocated users is t (see Figure 2: t represents the "delay" of one session with respect to another).

Our aim is to compute a probability density function $p_C(t)$, representing the probability that the colocation (i.e., contact) between any two users a and b lasts t. Without loss of generality, we assume that a's session starts before b's session (the other case is symmetrical). If a and b are colocated (i.e., in contact), then only two cases can occur (see Figure 2):

1) b starts with a delay R and terminates after a (i.e., the two sessions overlap),

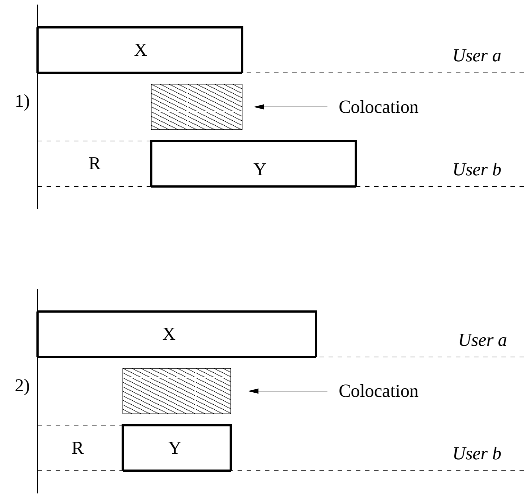

Figure 2: Connectivity cases.

2) b starts with a delay R and terminates before a (i.e., b's session is contained in a's session).

The probability of occurrence of case 1) is given by the probability that Y is more than $X - R$, which we write as $p(Y > X - R)$. Analogously, the probability of occurrence of case 2) is given by the probability that Y is less than $X-R$, written as $p(Y \leq X-R)$. Overall, case 1) and 2) contribute to $p_C(t)$ as follows:

$$p_C(t) = p(Y > X - R)p_{X-R}(t) + p(Y \leq X - R)p_Y(t) \quad (1)$$

where $p_{X-R}(t)$ represents the probability that $X - R$ lasts t. As mentioned above, under our assumptions users are characterised by the same behaviour, therefore, for all t, we have $p_X(t) = p_Y(t)$, and X and R are two independent random variables; thus, we can write

$$p_{X-R}(t) = \int_0^{+\infty} p_X(t+r)p_R(r)dr \quad (2)$$

Intuitively, Equation (2) states that $X - R$ lasts t if X lasts $t + r$ and R lasts r, integrated over all possible delays r from 0 to $+\infty$.

We evaluate now the term $p(Y > X - R) = p(X - R < Y)$. Notice that this is a number and represents the weight of p_{X-R} in Equation (1). For a fixed y, we have

$$p(X - R < y) = \int_0^{y} p_{X-R}(k)dk \quad (3)$$

and therefore

$$p(X - R < Y) = \int_0^{+\infty} \left(\int_0^{y} p_{X-R}(k)dk \right) dy \quad (4)$$

Taking into account Equations (1) and (2), we can rewrite $p_C(t)$ in terms of the known functions $p_X(t)$ and $p_R(t)$, as follows:

$$p_C(t) = \chi \int_0^{+\infty} p_X(t+r)p_R(r)dr + (1-\chi)p_X(t) \quad (5)$$

where $\chi = p(X - R < Y)$ is defined by Equation (4).

To summarise: Equation (5) allows the computation of the probability distribution for the colocation (i.e., contact) duration of users in a place as a function of their residence time in that location and the arrival delay. It has been shown in [4] that the value computed using this equation for p_c corresponds to the measured one.

In CTG, Equation (5) can be used if the distribution of colocation is not available from the scenario under investigation.

4. CONNECTIVITY TRACE GENERATOR

In this section we present the Connectivity Trace Generator tool (CTG), which is able to produce traces characterised by given connectivity properties. The tool is founded on the model presented in Section 3 and it takes the distributions of residence time, inter-contact time, and degree distribution as input parameters.

The traces generated by our CTG tool abstract away from spatial movements and concentrate on connectivity and inter-contact times, which are key to the testing of opportunistic protocols.

We now illustrate the steps of the algorithms implemented by the Connectivity Trace Generator. The basic idea is to allow the generation of traces of arbitrary time length. A case study is presented in Section 5 using the Dartmouth traces.

4.1 Generating the Potential Contacts Graph

The first step in CTG is the generation of the *Potential Contacts Graph*. The inputs of the tool are the number of nodes (N) on which the traces of connectivity need to be generated, the distribution of the contact times ($p_C(t)$) distribution ($p_{pc}(t)$), the time duration of the traces, and the distribution of the time elapsed between two colocations of the same pair of users. This is called the *inter-contact time* and its distribution is denoted by $p_{IC}(t)$. For the sake of this section we assume that these input parameters are chosen by an expert tester. In the next section we will show how these can be derived from the existing traces.

Each of the N hosts is mapped to a vertex of the potential contact graph. An edge between two vertexes exists if a potential contact is possible during the total desired simulation time. This means that, in the case of our example, an edge between two vertexes A and B exists if and only if the individuals have a chance of being colocated at least once during the period of the traces duration.

We have implemented a procedure to generate a graph given its number of vertexes and their degree distribution, within a certain approximation. Intuitively, the procedure non-deterministically tries to build a graph with the desired properties, and iteratively refines the solution up to the desired approximation level, varying from 2.5% to 4% in our examples.

4.2 Generating the Instant Snapshot Contact Graphs

Once the potential contact graph is generated, the actual connectivity traces can be produced as a sequence of *instant snapshot contact graphs*, one for each instant of time. A connectivity graph for time t represents the network of connected vertexes at time t. The instant snapshot contact graphs are generated as follows: the first connectivity graph is generated from the potential contact graph assuming that each potential edge is non active (i.e., no vertexes are connected and the edge is in an “off” state). Then, the distribution of inter-contact times $p_{IC}(t)$ is used to assign a duration to the “off” time of each edge. When this time has elapsed, the distribution of connectivity time $p_C(t)$ is used to assign a duration to the “on” time of each edge. Thus, for each edge there is a sequence of durations off/on, distributed following $p_{IC}(t)$ and $p_C(t)$.

A sequence of instant snapshot contact graphs is obtained by looking at each edge. Finally, the instant snapshot contact graphs are appropriately parsed to generate connectivity traces. We have implemented a parser for the Omnet++ [21] event simulator, but parsers for other kind of simulator, such as NS2, can easily be implemented.

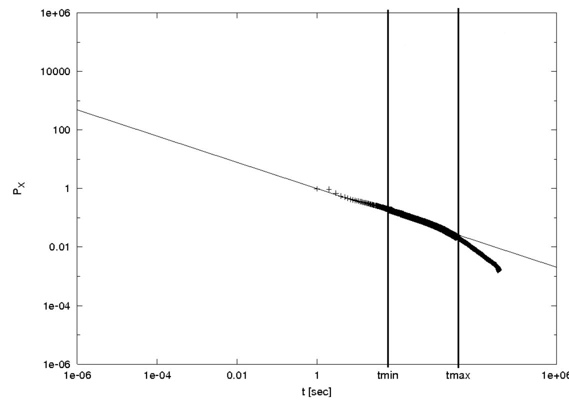

Figure 3: Distribution of residence time in Academic Building 22, log-log scale (all users over 4 years).

The generated connectivity traces represent the test cases. The CTG tool performs all the generation steps automatically based on the input of the distributions $p_{pc}(t)$, $p_C(t)$, and $p_{IC}(t)$.

5. EXTRACTING THE INPUT PARAMETERS FROM REAL TRACES

In this section we show how to extract input parameters for our tool from real traces such as the ones collected in the CRAWDAD repository [10]. Real traces can be collected as part of a first deployment of the mobile system or through other kinds of investigations, and despite giving a general perspective over the context in which the system is to run, they are often quite limited in scope. CTG allows to use this initial and approximate input to generate further test cases.

To show how this can be done, we have used traces from [11] to derive $p_X(t)$, $p_{IC}(t)$, $p_R(t)$ and $p_C(t)$. Section 5.2 describes how the maximum number of contacts of the original traces are scaled by the tool to produce test cases with different sizes, and Section 5.3 compares the traces generated by CTG with the real traces.

5.1 Case study: parameters from the Dartmouth traces

We consider a selection of traces from [11], from 01/04/2001 until 30/06/2004. These traces record connections and disconnections of users at a number of access points in the Dartmouth campus; in particular, the data available include MAC addresses, locations of access, and timestamps. In the traces analysed we found 13889 different users and 178 different locations.

As an example, Figure 3 reports the cumulative distribution of residence times for all users in Academic Building 22 in a log-log scale. For any given duration t, the value on the y axis gives the probability that the session of a user lasts t or more seconds.

As previously observed in a number of works (see for instance [5]), the distribution of the residence time at a given access point follows a power law in a range of values[2], denoted by $[t_{min}, t_{max}]$ in our paper (points between the vertical bars $x = t_{min}$ and $x = t_{max}$ in Figure 3). In the traces analysed we found $t_{min} = 60sec$ and $t_{max} = 13397sec$. Figure 3 also reports the interpolated curve to

[2]Notice that if a probability density function is a power law of the form $f(x) = x^{-\alpha}$, then its cumulative distribution is a power law with coefficient $-(\alpha - 1)$.

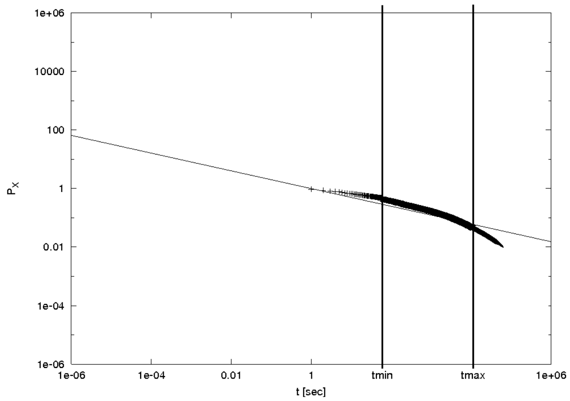

Figure 4: Distribution of residence time in Residence Building 20, log-log scale (all users over 4 years).

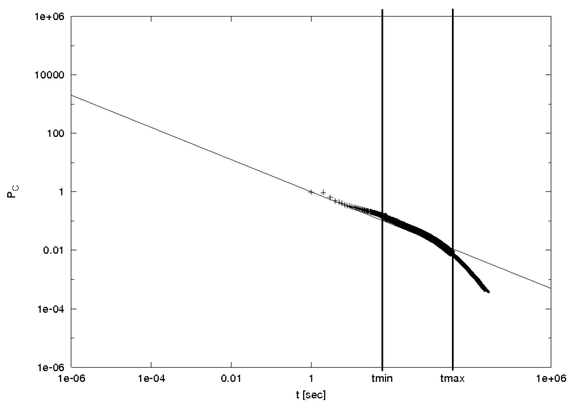

Figure 5: Distribution of colocation time in Academic Building 22, log-log scale (all users over 4 years).

obtain the coefficient for the cumulative distribution of $p_X(t)$ (denoted by $P_X(t)$), from which the actual coefficient k_X of $p_X(t)$ can be computed (see the the straight line in Figure 3). Another example location is provided in Figure 4, where the same behaviour is observed.

For the calculation of $p_{IC}(t)$, the inter-contact time probability, we proceeded similarly to $p_X(t)$ and, as expected [5], we found a power law distribution with coefficient k_{IC} in the range $[t_{min}, t_{max}]$. Analogously, we evaluated the distribution $p_R(t)$ and we found a power law distribution with coefficient k_R. The interpolated coefficients k_X, k_{IC}, and k_R for two locations and their average value for the whole campus, over four years, are reported in Table 1. Due to space limitations, we do not include graphs and coefficients for all locations; these are available from the authors upon request.

We then computed the actual distribution of colocation time for two generic users. The distribution $p_C(t)$ of the duration of colocation obtained form the traces follows a power law. As above, we interpolate the distribution to obtain the coefficient of the power law. Two example locations are reported in Figures 5 and 6.

The coefficients for the two locations and the average coefficient over all locations are reported in Table 2.

Since the results presented above are averaged over four years, we repeated the process of computing $p_C(t)$ but for a time window of 8 hours only (from 9am to 5pm), averaged over a period of one month in the middle of an academic term (from 19/04/2004 to 19/04/2004). We performed these measures to rule out the possibility that, at a smaller time scale, the behaviour of the distribution of colocation could be different.

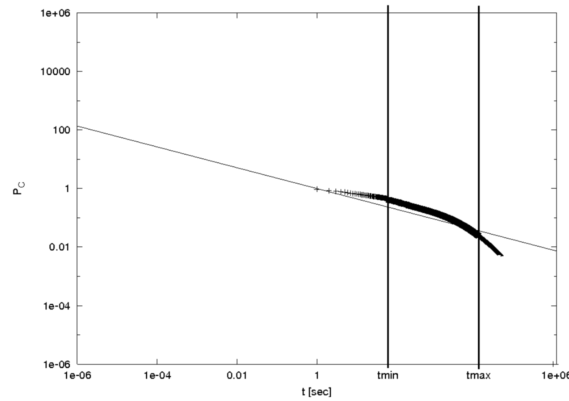

Figure 6: Distribution of colocation time in Residential Building 20, log-log scale (all users over 4 years).

Location	$-k_X$	$-k_{IC}$	$-k_R$
Academic Building 22	-1.448	-1.745	-1.062
Residential Building 20	-1.303	-1.909	-1.047
All campus (averaged)	-1.281	-1.553	-1.064

Table 1: Value of the coefficients k_X, k_{IC}, and k_R at two locations and average values for the whole campus.

Location	$-k_C$
Academic Building 22	-1.551
Residential Building 20	-1.356
All campus (averaged)	-1.327

Table 2: Value of the coefficients for colocation at two random location and average values for the whole campus, over four years.

Location	$-k_C$	$-k_{IC}$
Academic Building 18	-1.430	-1.420
Residential Building 20	-1.209	-1.207
All campus (averaged)	-1.268	-1.280

Table 3: Value of the coefficients for colocation at two random location and average values for the whole campus, 9am - 5pm from 19/04/2004 to 19/05/2004.

Additionally, we interpolated the distribution of inter-contact time for the same time windows, and, again, we obtained a power law. The interpolated values for colocation and inter-contact coefficients are reported in Table 3.

5.2 Degree distribution

Let us assume we want to generate traces for 8 hours and for 200 hosts mirroring the behaviour of Dartmouth traces in the period 19 April to 19 May 2004, a period without holidays, considering only the hours 9am to 5pm.

In the real traces for the period considered, we found 1892 users and a maximum number of potential contacts for an individual equal to 42 (i.e., the individual with the maximum number of potential contacts had 42 contacts). The distribution of the number

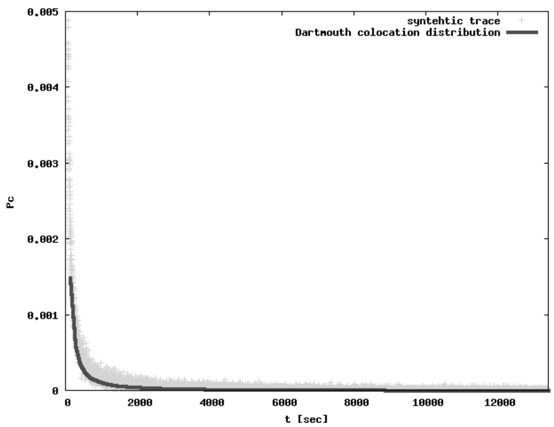

Figure 7: Comparison between synthetic trace and real power-law distribution using Dartmouth coefficient: contact time.

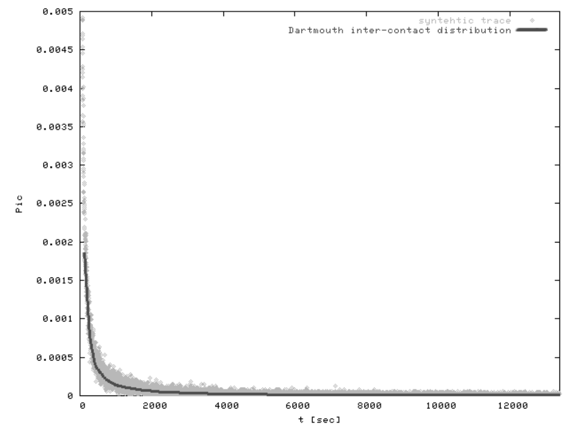

Figure 8: Comparison between synthetic trace and real power-law distribution using Dartmouth coefficient: inter-contact time.

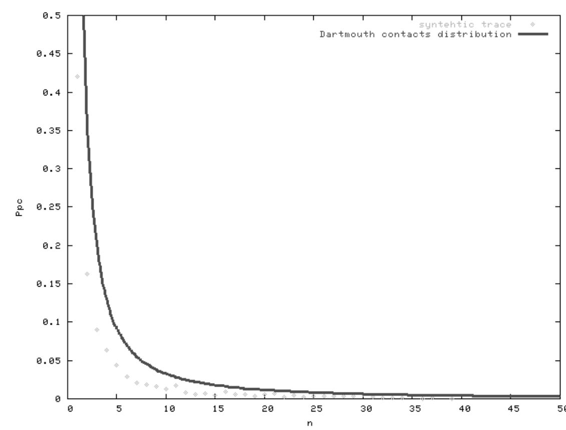

Figure 9: Comparison between synthetic trace and real power-law distribution using Dartmouth coefficient: degree distribution.

of contacts is *the distribution of the degrees of the vertexes* in the Potential Contact Graph. In order to calculate this in an appropriate manner, we have interpolated the distribution of the degrees for the traces, and obtained a power law in the range up to 42 individuals. We denote this distribution by $p_{pc}(n) = n^{-k_{pc}}$: $p_{pc}(n)$ gives the probability that a node has degree n. The measured value for k_{pc} is -1.484. In order to be able to generate traces with similar patterns, which may be used in the evaluation of opportunistic protocols, we need to obtain a distribution for the 200 users, instead of 1892. This is done through geometric similarity over the frequency graph of the degrees of vertexes (see Appendix). Intuitively, the maximum number of contacts scales up or down proportionally to the square root of the ratio of the total number of vertexes in the graph. Thus, the maximum number of contacts for a single user when 200 users are present is computed to be 13, and we take the same coefficient for the power law distribution. This enable us to generate a sequence of random degrees following the desired distribution.

5.3 Validation of Synthetic Traces vs Real Traces

We now show that the generated traces have the same characteristics as the ones in input, by comparing contact distribution, inter-contact time distribution, and node degree distribution.

Figure 7 shows how the cumulation function for the colocation values of the generated traces (lighter marks) mirrors the contact time distribution of the original traces in input (darker line).

Figures 8 and 9 compare the generated values (lighter marks) with the original input values for inter-contact time and degree distribution.

While the graphs shown here have been generated for the sake of comparison with the original traces, we remark that CTG can be used to generate different traces with a lower number of users or with different contact patterns. These traces are used in the next section to test various opportunistic protocols.

6. PROTOCOL PERFORMANCE TESTING

When the synthetic traces are obtained using CTG, they can be used for testing mobile systems. With respect to the use of real traces for testing, the synthetic traces allow more variability: the number of nodes, connection degree of the nodes, simulation time as well as contact and inter-contact distribution can be varied to conduct performance testing of the system. In this section, we present the performance of opportunistic protocols evaluated using our synthetic traces as test cases. We implemented four routing protocols using the OMNeT++ discrete-event simulator [21], namely flooding plus three opportunistic ones: epidemic routing [20], the Context-aware Adaptive Routing (CAR) protocol [13] and a protocol based on the random selection of the message carriers for asynchronous delivery based on store-and-forward. We show how the synthetic traces generated allow flexible performance testing.

6.1 Performance Evaluation using CGT

We use three performance indicators (delivery ratio, average delay, overhead) for the evaluation of the protocols and we vary the distribution of the inter-contacts time, of contacts duration and structure of the potential contacts network to demonstrate the performance testing which could be conducted.

Choice of the Parameters As described in Section 4.1, we have generated a set of traces for 200 nodes and eight hour of simulation time, which mirror the behaviour of traces of 1892 users in the period 9 April to 19 May 2004 in a eight hour time window, from 9am to 5pm. We studied the protocol performance by varying one parameter at a time in the synthetic traces. During the simulation we sent 1000 messages in 8 simulated hours. The messages are sent from randomly chosen senders to randomly selected recipients. We use a buffer size equal to 1000 (i.e., infinite) for the store-and-forward protocols. The figures of this section show a 95% confidence interval obtained using multiple runs.

The following is a brief description the key characteristics of the four protocols.

Flooding The study of the performance of the flooding protocol provides an upper bound of the delivery ratio that is possible to obtain using synchronous protocols, i.e., protocols that are based on the existence of a path between the sender and the receiver of a message when the message is sent. It does not exploit store-and-forward mechanisms.

Epidemic protocol Similarly, the upper bound in terms of delivery ratio for store and forward approaches can be obtained by using the epidemic routing protocol. In particular, for our experiments, we used the pure epidemic routing protocol proposed by Vahdat and Becker in [20]. According to this protocol, when two hosts become neighbours (i.e., they are connected), they determine which messages each possesses that the other does not, using summary vectors that index the list of messages stored at each node; they then exchange and store them in their buffer. This mechanism leads to an eventual delivery of messages, if a transitive path between the sender and the recipient exists over the considered period of time.

Context-aware Adaptive Routing (CAR) The Context-aware Adaptive Routing (CAR) is a unicast opportunistic protocol based on the intelligent selection of message carriers enabling a store-and-forward delivery to the recipients. The key ideas of the protocol are the following. Each host calculates its delivery probabilities. This process is based on the *prediction* of the behaviour of the nodes in terms of patterns of colocation and relative mobility based on Kalman filter based time series forecasting. The calculated delivery probabilities are periodically sent to the other hosts in the connected cloud as part of the update of routing information using a modified version of the DSDV protocol [9]. Each host maintains a logical forwarding table of tuples describing the next logical hop, and its associated delivery probability, for all known destinations. Each host uses local prediction of delivery probabilities between updates of information. If the message carrier, while moving, meets a host with a higher delivery probability, the message is transferred to the host with the higher delivery probability. The details of the design of this protocol can be found in [13]. In our simulation, every node is (re-)evaluating the probability of the nodes in the cloud every 60 seconds.

Random Choice of Message Carriers Finally, we also consider a randomised version of CAR, where the message carriers are selected randomly among all the neighbours currently reachable in the connected cloud by means of DSDV. The algorithm is implemented as follows. Every 60 seconds, the node selects a random entry in the routing table (also considering the entry about the node itself). Then the message is sent to the selected node where it is stored (or is maintained in the buffer of the host) for a subsequent retransmission.

6.2 Performance testing results

Impact of Inter-contacts Time Distribution We studied the impact of inter-contacts time distribution by varying the exponent of the power law. The performance in terms of delivery ratio, average overhead and number of messages are shown in Figure 10, 11, 12, respectively. The power law coefficient of the synthetic traces following the Dartmouth traces patterns is -1.28; we generated traces varying the coefficient in the range $[-1.75, -0.75]$.

gA vertical line corresponding to the value of the coefficient for these traces is drawn in every graph presented in this section. Notice that by using only the set of Dartmouth measurements we

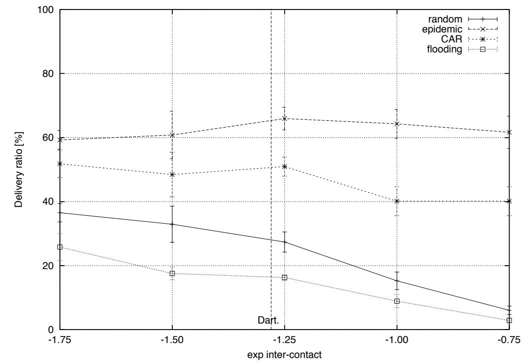

Figure 10: Impact of inter-contacts time distribution: delivery ratio vs inter-contacts time exponent.

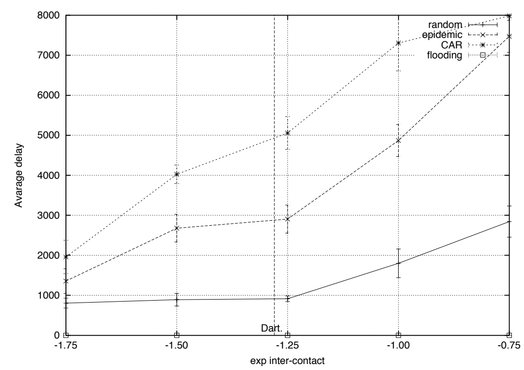

Figure 11: Impact of inter-contacts time distribution: average delay vs inter-contacts time exponent.

would have only a point in this graph instead of a curve. The curve allows to see trends in the protocol behaviour which would go undetected otherwise.

A higher coefficient leads to a larger number of long disconnections in average. The results with the specific values extracted by the traces using the Dartmouth coefficient can be found in Table 4. As expected, the delivery ratio of flooding decreases as the coefficient increases, since, when the flooding takes place, every nodes has a lower number of neighbours (i.e., the probability that two nodes are colocated is lower).

Instead, the performance of the epidemic protocol in terms of delivery ratio is not affected by the variation of the distribution of the inter-contacts durations: the chance of *infecting* the neighbours is more or less constant in this range of variation. In fact, since the maximum inter-contact time that we are considering is 4 hours (corresponding to a t_{max} of 14,000sec), two pairs of nodes have a chance of getting in reach at least two times during the simulated period (8 hours); at the same time, it is important to note that this does not guarantee that the message dissemination will reach all the (transitively) connected network. In any case, from the analysis of the results, the probability that a path exists between a randomly selected pair of nodes is considerably less than 1. Considering the

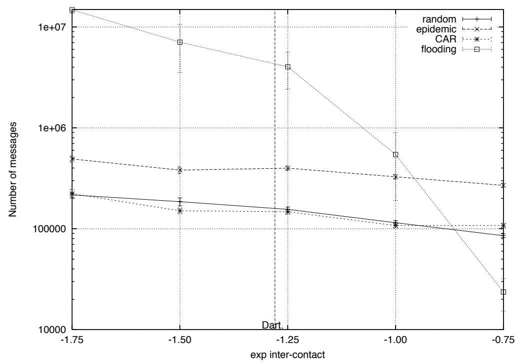

Figure 12: Impact of inter-contacts time distribution: number of messages vs inter-contacts time exponent.

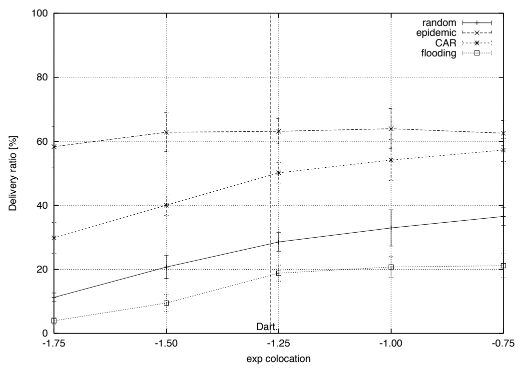

Figure 13: Impact of colocation distribution: delivery ratio vs colocation exponent.

performance of the epidemic protocol, its value can be estimated to be close to 0.6 (in average). We also observe that the performance of CAR deteriorates as k increases. This is due to the fact that the accuracy of the prediction algorithm decreases as the disconnection intervals increase. The random choice protocol suffers of a lower chance of being in reach of the recipients as the inter-contacts time increases. Naturally, as shown in Figure 11, the average delivery delay increases, with increasing inter-contact time, for all the protocols.

Impact of the Colocation Distribution Similarly, in order to study the impact of colocation distribution, we varied the coefficient of this distribution. The performance in terms of delivery ratio, average overhead and number of messages are showed in Figure 13, 14, 15, respectively. The power law coefficient extracted from the set of the Dartmouth trace taken into consideration was -1.268; in this case, too, we varied the coefficient in the range $[-1.75, -0.75]$. A higher coefficient leads to a larger number of long colocation periods between pair of nodes. In a sense, the results are symmetric to the ones obtained varying the inter-contacts time. We only observe that, once again, the results of the epidemic protocol are not affected by the variation of this distribution.

Impact of the Structure of the Potential Contacts Network The

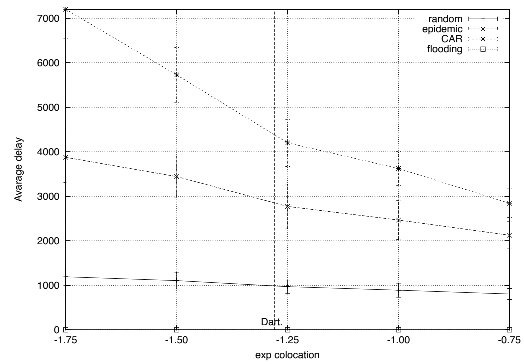

Figure 14: Impact of colocation distribution: average delay vs colocation exponent.

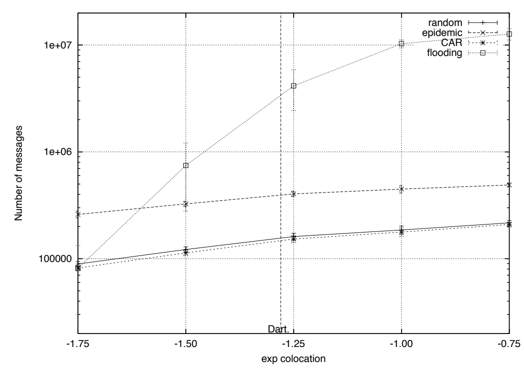

Figure 15: Impact of colocation time distribution: number of messages vs colocation exponent.

network structure of the Dartmouth traces is characterised by a coefficient equal to -1.485, and the generated traces have a maximum number of potential contacts equal to 13. We studied the impact of the number of neighbours, by varying the maximum number of edges of the potential contacts graph $n_{c_{max}}$. The impact of the variation of the maximum number of neighbours of the potential contacts graphs is shown in Figures 16, 17 and 18 in terms of delivery ratio, average delay and overhead. As the maximum number of contacts in the power-law distribution increases, the performance of the protocols of flooding, CAR and random choice improve, since the number of potential contacts is larger. CAR benefits from a wider choice of potential carriers and a greater number of transfer opportunities. The performance of the epidemic protocol also improves as the maximum number of contacts increases, leading to a better network connectivity (from a transitive point of view, over time).

7. RELATED WORK AND DISCUSSION

There is a growing interest in approaches for testing mobile systems and applications, see for instance [18]. Most of these approaches, however, concentrate on testing aspects related to context awareness (see, for example, [22]). Instead, what is proposed in this

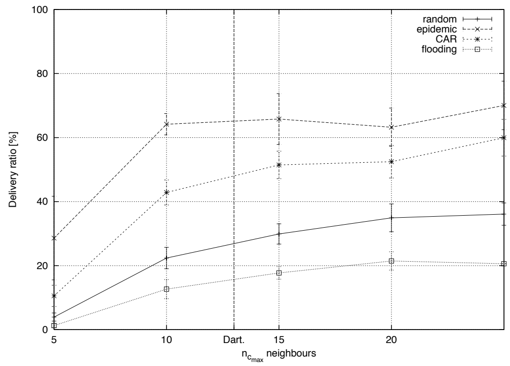

Figure 16: Impact of the structure of the potential contacts network (maximum number of neighbours): delivery ratio vs maximum number of neighbours.

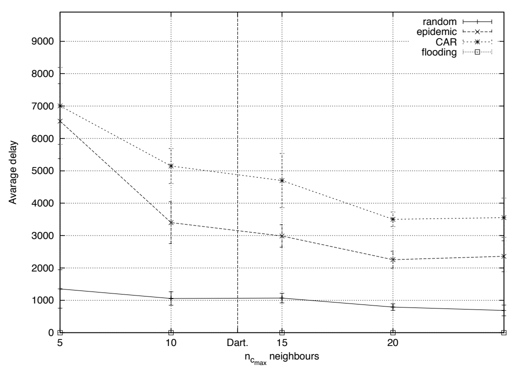

Figure 17: Impact of the structure of the potential contacts network (maximum number of neighbours): average delay vs maximum number of neighbours.

paper relates to how to provide automatic generation of connectivity test cases in order to test the performance of communication protocols and applications in opportunistic mobile systems. Our approach allows flexible performance testing of new protocols and applications. Indeed, when a system is being prototyped, some logging and measuring data could be collected through a small scale trial on the usage patterns. The connectivity traces could then be analysed and, using our methodology, a simulation on a larger scale could be carried out on the communication protocol under investigation, using larger (i.e., with a higher number of hosts) synthetic traces.

We see this as a first step towards a more comprehensive solution for the verification of mobile systems; in this sense, our work lacks a metric for coverage criteria of the generated test cases. An investigation along these lines for a similar problem has been presented in in [17]: we leave the issue of evaluating coverage conditions open for future work.

In parallel, the mobile systems community has offered its own solution to testing communication protocol through the generation of mobility models for simulators: some of these are purely random, such as [3, 9], while more recent ones tend to base their logic on some real concepts, such as social aspects [14] and specific movement scenarios (e.g., downtown traffic [12]). With respect to the state of the art in mobility model building, for the first time in this work the results of the analysis of available traces have been used to derive a connectivity model for simulation studies. Our approach allows for experiments considering different connectivity patterns that can be derived from real traces, or simply be provided empirically. This is not possible with the other existing models, including the recent ones based on probability of transitions, since the contacts patterns distribution are not input parameters. Our methodology can also be used to study extreme cases, such as scenarios with very sparse ad hoc networks, characterised by very long disconnection intervals.

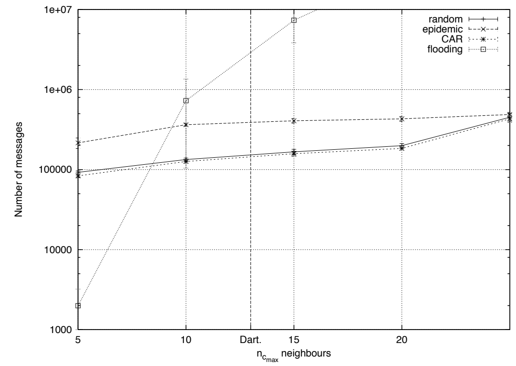

Figure 18: Impact of the structure of the potential contacts network (maximum number of neighbours): number of messages vs maximum number of neighbours.

	Flooding	Epidemic	CAR	Random
Delivery	18.76%	62.7%	49.95 %	28.52 %
Delay [s]	2636.20	2636.30	4192.15	954.77
Overhead	3276463	397153	150516	158658

Table 4: Performance of four protocols considering the scenario extracted by the traces of Dartmouth College campus, 9am - 5pm from 19/04/2004 to 19/05/2004.

8. CONCLUSIONS

In this paper we have presented CTG, a tool for the generation of synthetic connectivity traces which can be used for testing the performance of communication protocols and applications at the heart of opportunistic mobile systems. The tool can be used in conjunction with a connectivity model, which builds on probability distributions for residence time of individuals ($p_X(t)$), the distribution of time intervals between connections ($p_{IC}(t)$), and a distributions of delays in overlapping connections ($p_R(t)$). These distributions can be easily obtained from real traces, and we have presented an example of this in Section 5, where human mobility traces from the Dartmouth College have been analysed.

To emphasize the applicability of our generator, we have used the synthetic traces to test some opportunistic protocols. In particular, we presented how different connectivity patterns affect the performance of the protocols in terms of delivery ratio, total number of messages, and average delay.

We have in mind a number of future applications for this work, starting from more tests on the applicability of the methodology to other types of traces. For instance, using our approach, one could analyse connectivity patterns and the implied performance of opportunistic protocols in different scenarios, such as a train station with very short connections between individuals, or a library with typically longer connections.

Acknowledgments: We would like to thank Enrico Denti, Wolfgang Emmerich, Dimitrios Moustakas and Michele Sama for their valuable comments on an earlier draft of this paper. We would like to acknowledge the support of EPSRC through projects CREAM and UBIVAL and the European Union through projects PLASTIC and RUNES.

9. REFERENCES

[1] A. Balachandran, G. M. Voelker, P. Bahl, and P. V. Rangan. Characterizing user behavior and network performance in a public wireless lan. In *Proceedings of SIGMETRICS '02*, pages 195–205, New York, NY, USA, 2002. ACM Press.

[2] M. Balazinska and P. Castro. Characterizing Mobility and Network Usage in a Corporate Wireless Local-Area Network. In *1st International Conference on Mobile Systems, Applications, and Services (MobiSys)*, San Francisco, CA, May 2003.

[3] J.-Y. L. Boudec and M. Vojnovic. Perfect simulation and stationarity of a class of mobility models. In *Proceedings of INFOCOM'05*, pages 2743–2754, March 2005.

[4] R. Calegari, M. Musolesi, F. Raimondi, and C. Mascolo. A human connectivity model for opportunistic mobile systems. 2007. UCL CS Research Note RN/07/08. Submitted for publication.

[5] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on the design of opportunistic forwarding algorithms. In *Proceedings of INFOCOM'06*, Barcelona, Spain, April 2006.

[6] K. Fall. A delay-tolerant network architecture for challenged internets. In *Proceedings of SIGCOMM'03*, August 2003.

[7] T. Henderson, D. Kotz, and I. Abyzov. The changing usage of a mature campus-wide wireless network. In *Proceedings of MOBICOM'04*, pages 187–201, New York, NY, USA, 2004. ACM Press.

[8] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Pocket switched networks and human mobility in conference environments. In *WDTN '05: Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 244–251, New York, NY, USA, 2005. ACM Press.

[9] D. Johnson and D. Maltz. Dynamic source routing in ad hoc wireless networks. In T. Imelinsky and H. Korth, editors, *Mobile Computing*, volume 353, pages 153–181. Kluwer Academic Publishers, 1996.

[10] D. Kotz and T. Henderson. CRAWDAD: A Community Resource for Archiving Wireless Data at Dartmouth. *IEEE Pervasive Computing*, 4(4):12–14, October-December 2005.

[11] D. Kotz, T. Henderson, and I. Abyzov. CRAWDAD trace dartmouth/campus/movement/01_04 (v. 2005-03-08). Downloaded from http://crawdad.cs.dartmouth.edu/, March 2005.

[12] K. Maeda, K. Sato, K. Konishi, A. Yamasaki, A. Uchiyama, H. Yamaguchi, K. Yasumotoy, and T. Higashino. Getting urban pedestrian flow from simple observation: Realistic mobility generation in wireless network simulation. In *Proceedings of MSWiM'05*, pages 151–158, September 2005.

[13] M. Musolesi, S. Hailes, and C. Mascolo. Adaptive routing for intermittently connected mobile ad hoc networks. In *Proceedings of the 6th International Symposium on a World of Wireless, Mobile, and Multimedia Networks (WoWMoM 2005). Taormina, Italy.* IEEE press, June 2005.

[14] M. Musolesi and C. Mascolo. A Community based Mobility Model for Ad Hoc Network Research. In *Proceedings of the 2nd ACM/SIGMOBILE International Workshop on Multi-hop Ad Hoc Networks: from theory to reality (REALMAN'06).* ACM Press, May 2006.

[15] V. Naumov, R. Baumann, and T. Gross. An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces. In *Proceedings of MobiHoc'06*, pages 108–119, New York, NY, USA, 2006. ACM Press.

[16] M. E. J. Newman. Power laws, Pareto distributions and Zipf's law. *Contemporary Physics*, 46:323, 2005.

[17] M. J. Rutherford, A. Carzaniga, and A. L. Wolf. Simulation-based test adequacy criteria for distributed systems. In *Proceedings of the 14th ACM SIGSOFT international symposium on Foundations of software engineering (FSE-14)*, pages 231–241, New York, NY, USA, 2006. ACM Press.

[18] T. Sohn, W. G. Griswold, J. Scott, A. LaMarca, Y. Chawathe, I. E. Smith, and M. Y. Chen. Experiences with Place Lab: an open source toolkit for location-aware computing. In *Proceedings of ICSE'06*, pages 462–471. ACM Press, 2006.

[19] D. Tang and M. Baker. Analysis of a local-area wireless network. In *Proceedings of MOBICOM'00*, pages 1–10, New York, NY, USA, 2000. ACM Press.

[20] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-2000-06, Department of Computer Science, Duke University, 2000.

[21] A. Vargas. The OMNeT++ discrete event simulation system. In *Proceedings of the European Simulation Multiconference (ESM'2001)*, Prague, June 2001.

[22] Z. Wang, S. Elbaum, and D. Rosenblum. Automated generation of context-aware tests. In *Proceedings of ICSE'07.* ACM Press, 2007.

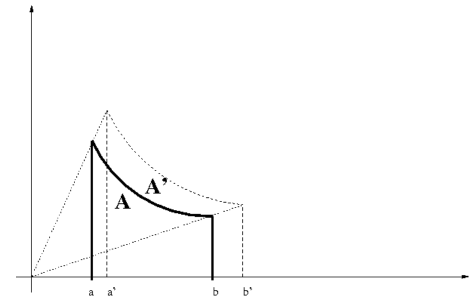

Figure 19: Scaling up (or down) a graph.

APPENDIX

In this appendix we present our methodology for scaling up or down a given distribution $f(x)$ of the degrees of the vertexes in a graph by means of an example (for the sake of this appendix, $f(x)$ could be any distribution limited between a and b).

Let $G = (V, E)$ be a graph with N vertexes, and let the degree of the vertexes be distributed in accordance with a distribution $f(x)$ between a minimum degree value of a and a maximum degree b: as an example, see the bold line in Figure 19 (notice that we are approximating a discrete distribution over natural numbers with a continuous distribution). For any number x between a and b, $f(x)$ is the number of vertexes whose degree is x. Notice that the number of vertexes N is equal to the area below the curve (denoted by **A**).

In order to generate a graph with M vertexes with a topological structure similar to G, we project the frequency $f(x)$ so that the new area A' under the frequency is equal to M.

Using simple equivalences, we can evaluate an approximate value for a' and b' as:

$$a' \approx \sqrt{\tfrac{A'}{A}}a \qquad b' \approx \sqrt{\tfrac{A'}{A}}b$$

(where $A = N$ and $A' = M$). This results approximates the distribution with a trapezium. The exact solution would require the explicit knowledge of $f(x)$ but, for the kind of distributions considered in this paper, the correction would be minimal.