A Mobility Model for Opportunistic Routing Protocols Validation

Matias Richart, Jorge Visca, Javier Baliosian, and Eduardo Grampín
School of Engineering, University of the Republic, Montevideo, Uruguay.
Email: {mrichart, jvisca, javierba, grampin}@fing.edu.uy

***Abstract*—Opportunistic networks –as opposed to infrastructure networks– are built on-the-fly by mobile, intermittently connected ad-hoc nodes, allowing to run delay tolerant applications. The idea is always to *store* a message, to *carry* it for some time, and to *forward* it when a suitable mobile node happens to be in range, all this with the hope that, after some of these *store-and-forward* steps, the message will eventually arrive to its destination. In these conditions, the throughput and delay induced by the routing protocol depends on the network characteristics (number of nodes, movement patterns), data flow characteristics (load, data patterns), and how the parameters of the routing protocol adapt to such a network environment. In particular, the performance of the opportunistic networks depends on some mobility and connectivity characteristics of the involved nodes, such as number of connected components and diversity of encounters between nodes, and the opportunistic routing protocols must be tested and tuned to adapt to those characteristics. However, it is impossible to control such mobility and connectivity parameters in a real setup and complicated to control them in a simulated setup. This paper presents a new mobility model, designed to help with the evaluation of opportunistic routing protocols. The main feature of this model is to allow the configuration of its average number of connected components and the diversity of encounters between nodes. We show how we control those characteristics and we present an example of how the model is used in the evaluation of RON, a content-based opportunistic routing protocol.**

I. INTRODUCTION

While it is true that communication networks have grown in coverage, bandwidth and reliability, reaching unexpected domains, places and public, it is also true that there are still some application domains that operate on disconnected networks by nature (e.g., sensor networks), and regions of the world with little or no communications infrastructure at all. For these situations the research community has been working on a class of network protocols that are tolerant to delays or disruptions storing and forwarding messages when and how it is possible. Variations of this kind of networks have been called Delay Tolerant Networks, Disruption Tolerant Networks (DTN) or Opportunistic Networks (ON), we will use this last one in the believe that is which better describes the case. The basic concept of opportunistic networking is that, in the absence of a fixed connectivity infrastructure, some data of interest is transferred between mobile devices using the "connection opportunity" that arise whenever mobile devices happens to come into the range of other devices because of the mobility of the users.

ONs –as opposed to infrastructure networks– are built on-the-fly by mobile, intermittently connected ad-hoc nodes, allowing to run delay tolerant applications. Many ON routing algorithms have been proposed (see [1] for a comprehensive review) but the common idea is always to *store* the message, to *carry* it for some time, and to *forward* it when a suitable mobile node happens to be in range with the hope that, after some of these *store-and-forward* steps, the message will, eventually, arrive to its destination. In these conditions, the throughput and delay induced by the routing protocol depends on the network characteristics (number of nodes, movement patterns), data flow characteristics (load, data patterns) and how the parameter of the routing protocol adapt to such a network environment.

The work presented in this paper has been developed in the context of the *Domestic Environment Monitoring with Opportunistic Sensor networks* (DEMOS) [2] project. For this project, we have developed RON, an opportunistic protocol that takes advantage of the OLPC-like[1] programs to transport the data produced by a low-cost platform for environmental sensors, such as air-quality sensors, that are deployed at the living premises of children in environmentally vulnerable neighborhoods as well as at their schools, parks, etc.

The environmental data collected by those sensors are transmitted, using opportunistic networking techniques, to the children's laptops as they pass-by during their daily life. Later, using the same techniques, the data is transmitted into a data-collection point at the school and from there to an environment monitoring station.

RON had to be tested and its configuration parameters tuned (bellow we introduce some details about this), unfortunately, the performance of RON and its optimal configuration, as well as for any other opportunistic protocol, depend on some mobility characteristics of the network nodes involved. Some of those mobility characteristics are, for example, the number of connected components of the network connectivity graph and the diversity of encounters between nodes.

However, fixing any of those parameters in a particular value to test RON's performance is impossible for real deployments with the needed scale and very hard in simulated environments using the previous mobility models.

This paper presents a novel, mosaic-based, synthetic mo-

---

[1] OLPC stands for One Laptop Per Child

978-1-4577-1792-5/11/$26.00

```
SUBSCRIBE
subscription_id=sid123
subscriptor_id=collector1
FILTER
mib=temperature
value > 35
END
```

(a) A Subscription.

```
NOTIFICATION
notification_id=notif123
source=sensor_node1
message_type=trap
watcher_id=watch_temp
mib=temperature
value=36.5
END
```

(b) A Notification.

Fig. 1. The Messages of the Notification Bus

bility model designed to evaluate RON and any other opportunistic routing protocol. The model is based in the well known Random Direction mobility model but modifies it in order to have control on the number of connected components of the connectivity graph and on the diversity of node encounters along the simulated period.

The structure of the paper is as follows. A high-level description of the RON protocol is presented in Section II. Later, in Section III, we present our mobility model. In Section IV we study the model and how to control its the connectivity metrics. Finally, in Section V, the paper presents a small example of how the model was used to evaluate RON through simulations.

II. A CONTENT-BASED OPPORTUNISTIC ROUTING PROTOCOL

As said before, the DEMOS architecture includes an opportunistic network between the sensor devices and the data collection points. In an opportunistic network, there is no guarantee of existence of a path between any pair of nodes in a given moment. For a message to reach its target, it may be necessary that some node keeps it in his own memory until it can deliver it to some other node.

There are several methods and algorithms for Opportunistic Routing, but for the purpose of supporting DEMOS operations we have developed RON, a new protocol. The reasons for creating a new protocol can be read in [2].

RON is a publish-subscribe protocol with only two types of messages: subscriptions and notifications. Both types are plain-text multi-line strings (see Figure 1). A notification is a message used to transmit information in the network, and consists of a list of key/value pairs. A subscription expresses the interest of a node on receiving certain notifications. Any node running RON can produce a subscription, then, RON broadcasts the subscription in an ad-hoc manner to all the nodes in the vicinity and they register and forward those subscriptions causing a controlled flow that potentially ends with all the nodes having registered the subscription. However, the time for which a subscription is stored in a node depends on how often it is received. For this purpose each subscription has a quality attached. The quality attribute attempts to predict how good the node is to reach said subscriptor. Qualities are maintained trough exchanging subscription lists with neighbors: when a given subscription is seen in the network, the corresponding quality is raised. As time passes without meeting it, or meeting it with inferior qualities, the subscription ages and the quality gets reduced.

To issue a notification, a node broadcasts a message containing a list of key/value pairs to the nodes in the vicinity. When a node receive a notification, it tries to match the notification's content with the subscriptions that it has registered. If the notification matches any subscription, the node stores the message. Once in the vicinity of new nodes, the process repeats until the notification arrives to all the nodes interested on it. As the storage space in the nodes is limited, at some point a node must decide which messages should keep carrying and which should drop. This is done prioritizing notifications that satisfy subscriptions with higher qualities.

Technically, RON is a gossiping algorithm. Each node maintains a table of already-seen destinations (subscriptions in this case), and a quality associated to each of those. The quality of a destination on a node represents how good is the node to reach that destination. Nodes periodically exchange their destinations lists with their associated qualities (this list is known as a "view"), and update their own qualities in the process. There are several parameters that control the behavior of the protocol. While usually those values are fixed at configuration time, we use a rule engine to adjust those parameters at runtime.

```
 1  N : Array[#N] of Notification
 2  V : Array[#V] of {Subscription , quality}
 3
 4  each t seconds
 5    broadcast(V)
 6
 7  when received_view(V')
 8    V.merge(V')
 9    out_n = matching(N,V')
10    broadcast(out_n)
11
12  when received_notifications(N')
13    N.merge(N')
```

Listing 1. Pseudocode for RON Opportunistic Network Algorithm

The pseudocode of RON protocol is shown in Listing 1, and, as its implementation, it is divided in two processes: one active for the periodic processing and subscriptions issuing, and another passive for the processing of incoming messages. The data structures are two arrays, one for notifications (N, line 1) and another for subscription-quality pairs (V, line 2).

Nodes periodically trigger broadcasts of its View array every τ seconds (line 4), where τ is a configuration parameter. When a View message is received by a node (line 7), two tasks are performed. First the incoming View is merged into the local View: new subscriptions are stored, and qualities associated to subscriptions present with higher qualities in the incoming View are raised (reinforced), or reduced (aged) otherwise. As the space available for storing subscriptions is limited by the size of the View array, some subscriptions are possibly dropped. For this the ones with lowest qualities are selected. The second task is to broadcast all notifications in the local buffer that satisfy some of the subscriptions in the received view (line 10). This is the step where the notifications are replicated in the network. RON takes the "broadcast advantage", and the notifications aren't sent explicitly to the View emitter, but broadcast for any node in range to receive. Thus nodes can use data overheard in the medium, reducing the number of transmissions. A given notification can be transmitted multiple times, each time an interested node is met, which provides for the robustness of the protocol. To limit the total number of notification broadcasts, a maximum number of transmissions parameter is set. Also, nodes abstain from transmitting a notification seen in the air in a configurable window of time, to avoid a broadcast storm where several nodes in range respond a single View message at the same time.

Once a node receives a notification, triggered by a View broadcast either from the same node or another, it proceeds to insert the received notification in the local notification array (line 12). As long as there is free space in this buffer all notifications are stored. Once the buffer is full, decision must be taken on which notifications to carry and which to drop. For this, RON calculates an accumulated quality for notifications: the sum of the qualities of all the subscriptions that the notification satisfies. The notifications with lowest accumulated quality are dropped first. Different policies are possible, for example taking in account the time the notification has been stored or how many times it has been transmitted.

From the previous description results clear that the connectivity metrics have an important impact on the performance of the protocol. For example, intuitively, when the diversity of encounter is high the notifications will spread to more nodes in the network causing less losses and decreasing the delay between the transmission and the reception. Something similar happens with the number of connected components, if this number is low means that more nodes are connected and there is a more direct route from the source to the destination.

The next section describes the model developed to study the performance of RON by this metrics.

III. A MOBILITY MODEL FOR THE STUDY OF OPPORTUNISTIC NETWORKS

A. Model Overview

The model developed consists of a square of land divided in smaller, possibly overlapping squares. The number of squares in which the land is divided and the overlapping ratio between them are parameters of the model. An specific number of nodes is assigned to move inside the boundaries of the small squares. The initial square of land is defined as the simulation scenario and its size is static, and varying the number of parts and the overlapping ratio we can vary the connectivity metrics of the network. To determine the number of nodes assigned to a specific square we keep the same density of nodes on each square. Once the nodes are assigned to a specific square their movement is restricted to that square. Inside the square the movement followed by the nodes is defined by the Random Direction Mobility Model (RDMM).

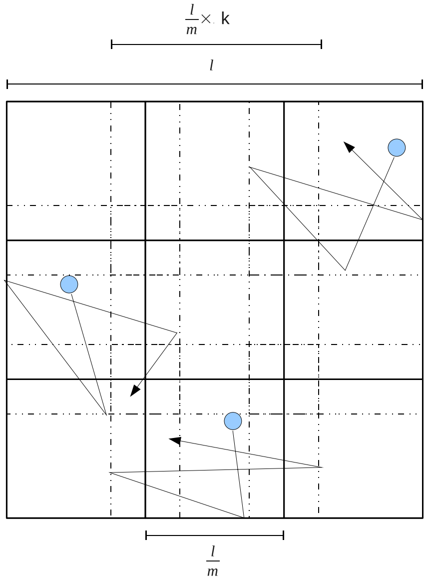

Fig. 2. Model Example

In RDMM nodes move in straight lines with constant speed until a specific distance is traveled or a boundary is reached, then a new direction and speed is randomly chosen. Adding a pause between walks is also possible in the RDMM. Our model uses the implementation of RDMM provided by the NS3 Network Simulator [3].

B. Connectivity Metrics

The connectivity metrics mentioned previously are defined bellow.

1) Node degree: The *node degree* (or diversity of encounters) measures the diversity of encounters as the number of different nodes that a node see all along the studied period, this means that we count the number of different encounters during the total simulation time. This value measures the node's capacity to disseminate a message.

2) Connected Component: To define the number of connected components, we first define the connectivity graph. Defining a connectivity graph in ON's where the nodes are mobile and not always in range is not immediate. Following the ideas in [4] and [5], we define the connectivity graph $G(V, E)$, where the set of vertexes V are the network nodes and E is defined as

$$E = \{(i, j) \mid \exists t \in [t_0, t_0 + T] \text{ and } d_t(i, j) < r\} \qquad (1)$$

where r is the transmission range of the network nodes and $d_t(i,j)$ the distance between nodes i and j at the time t. In other words is the graph where each node of the network is a vertex and if two nodes are in range in some instant there is an edge between their corresponding vertexes.

The number of connected components of the network is the number of components of the graph G. In other works the connected components are also named as clusters of nodes.

This number could be used to measure how disconnected is a network. This impacts on the behavior of the routing algorithm. For example, as mentioned in [6], in a disconnected network with C connected components and $C \ll N$ two nodes may communicate directly or using other nodes as instantaneous forwarding relays, but in a sparse network (C close to N) nodes mostly communicate one-hop at a time.

C. Design Goal

The presented model is designed to direct the movements of a set of simulated wireless network mobile nodes so that the simulated network has a given average node-degree and a average number of connected components.

D. Model Parameters

The configuration and dynamic parts of the model are controlled by the following parameters (see Figure 2): l defines the side length of the main land square in meters, m^2 is the number of squares in which the land is divided, k is the overlapping ratio of those squares, N is the approximate number of mobile nodes (we will describe below why it is not the exact number), and r sets the signal range of the node in meters. The speed of a node is a uniform random variable chosen every time that its direction changes in a range of speeds between v_{min} and v_{max} meters per second.

As seen in Figure 2, in general, the side length of the small squares is $\frac{l}{m} \times k$. However, with that size, keeping all the overlapping regions of the same width and the simulated land unchanged, the squares on the land edges would have sections hanging outside the simulated land. To solve this problem, we decided to cut off those hanging sections and, therefore the coordinates of the small square (i,j) (actually a rectangle now) are calculated as follows:

$$(minX, minY) = (p*i-d, p*j-d) \tag{2a}$$
$$(maxX, maxY) = (p*(i+1)+d),(p*(j+1)+d) \tag{2b}$$

where

$$p = l/m \tag{3a}$$
$$d = (p/2)*(k-1) \tag{3b}$$

p represents in how many parts the edge of the scenario must be partitioned and d is the length that the square must be enlarged or reduced to the sides. So k determines the size of the square where $k = 1$ means the square of area $\frac{l^2}{m^2}$ and increasing the value of k increases the area of the square maintaining the center of it fixed.

As can be seen in previous equations the parameters m and k determine the number and size of the squares and so, how much they overlaps. This will impact on where the nodes can move and how they see each other.

E. Implementation Details

The model is developed to execute on a simulation and consists not only in how the nodes move in the scenario but also in all the network configuration. It is implemented for executing in the NS3 Network Simulator which allows us to create quite realistic WiFi devices with propagation and delay models. We use a Constant Speed Loss Model for the delay and for the propagation a Friis Loss Model implementation[7]. So to approximate the wanted signal range we had to manage the signal power of the WiFi device.

The implementation consists in a script which from the parameters N, k and m creates the simulation script needed to run a simulation on the NS3 Simulator. From this parameters the script divide the scenario and assign the nodes to an specific square.

The parameter N is said to be an approximation of the number of nodes because the actual number of nodes is defined as:

$$\begin{aligned} nNodes = \quad & \sum_{i=1}^{m-1}\sum_{j=1}^{m-1} \left\lceil \frac{N*area(i,j)}{totalArea()} \right\rceil * f(i,j) \\ & + \left\lfloor \frac{N*area(i,j)}{totalArea()} \right\rfloor * f(i,j) \end{aligned} \tag{4}$$

where

$$area(i,j) = \text{area of the (i,j) square} \tag{5a}$$
$$totalArea() = \sum_{i=1}^{m-1}\sum_{j=1}^{m-1} area(i,j) \tag{5b}$$
$$f(i,j) = \sum_{i=1}^{m-1}\sum_{j=1}^{m-1} 1 \pmod 2 \tag{5c}$$

This way we obtain a good approximation to the number of nodes N maintaining the density of nodes on each square.

IV. MODEL STUDY

A. Goal

To study the model we consider the two metrics mentioned in previous sections: the average number of connected components and the average node-degree. The objective is, by simulations, to determine the relationship between the model parameters m and k and the metrics mentioned before. To accomplish this we performed several experiments to see how this metrics vary in function of m and k and then try to find an equation that experimentally relates this parameters. The goal is that later this equation could be used to select an specific instance of the model (values of m and k) for a expected number of connected components or node degree.

Param	Value
l	5000 m
N	100
v_{min}	1 m/s
v_{max}	2 m/s
r	200 m

TABLE I
SIMULATION CONFIGURATION PARAMETERS

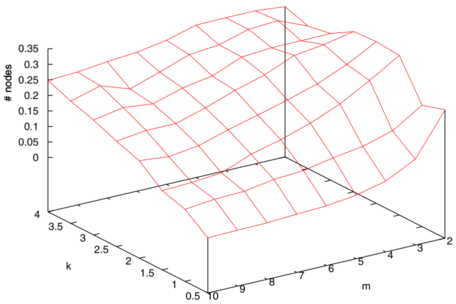

Fig. 3. Normalized Average Node-degree

B. Simulation Setup

The simulation configuration parameters are shown in Table I. These parameters were chosen trying to emulate some real environment where the protocol could be used, for example a small neighborhood where children move with their computers. The speed is the average speed of a person waking and the range is a common range of mobile computers.

After obtaining all the scenario configurations some applications are installed on each node so as to obtain the needed data. Every mobile node broadcast its ID number every second, listen the emissions of other nodes and register the listened IDs. Then, every ten seconds this logs are used to count the number of connected components on that period of time. Using the logs of each node, the connectivity graph is created for every time period and then the number of connected components is obtained from the graph.

This same log is also used to count the node degree as the quantity of different nodes a node see during the simulation period. Then, the average node-degree of the model is calculated. The simulation runs for a period of 10,000 seconds so we obtain 1,000 readings of the simulation state.

C. Average Node-degree

Figure 3 shows how the average node-degree vary in function of different values of m and k. This values are normalized by the number of nodes in the simulation. The average node-degree varies significantly in function of the model parameters with k in the range $[0.5, 4]$.

So as to study in depth this relationships we took a fixed value of m and let vary k and do the same on the other way, fix k and let free m. In Figure 4, for the node degree, we fix $m = 8$ and vary k in the interval $[0.5, 4]$ with a step of 0.5. In the figure it is also represented the standard deviation from the average and a linear approximation to the function obtained. The linear approximation gives us an experimental formula of how the average node-degree varies in function of k with $m = 8$:

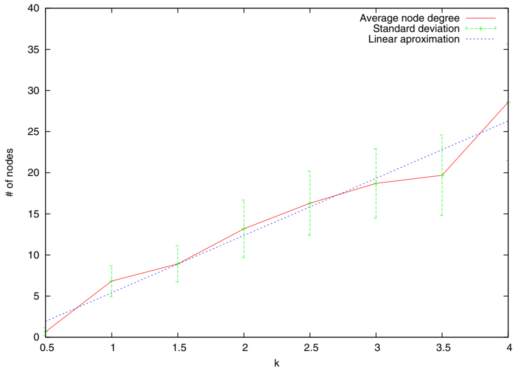

Fig. 4. Average Node-degree, $m = 8$

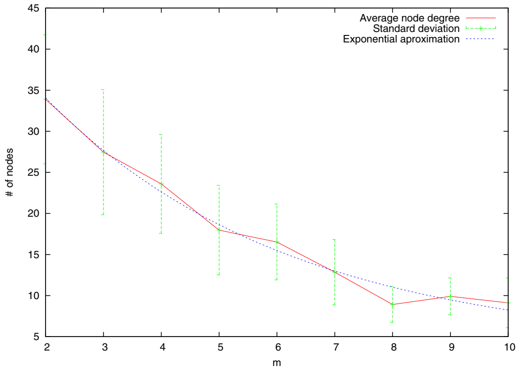

Fig. 5. Average Node-degree, $k = 1.5$

$$nd_{m=8}(k) = 6.95 * k - 1.54 \tag{6}$$

The same was done for $k = 1.5$ and m in $[2, 10]$, the result is shown in Figure 5. In this case we obtain an exponential function to approximate the relationship between the average node-degree and m which formula is:

$$nd_{k=1.5}(m) = 48.75 * e^{-0.23*m} + 3.62 \tag{7}$$

In Figure 4 it can also be seen that the standard deviation grows as the value of k increases while in the case of Figure 5 the standard deviation is higher in lower values of m. An explanation of this behavior is shown in figures 6 and 7 where an histogram of the node degree is drawn. This histograms show the number of nodes with a specific node degree. In both figures there are cases of wide graphs (high standard deviation) and others more tight but with one, two or three differentiated peaks. This is a result of the definition of the model which implicitly determines different zones of node degree on the scenario depending on m and k. An example is shown in Figure 8 where with $m = 3$ and $k = 1$ three different zones

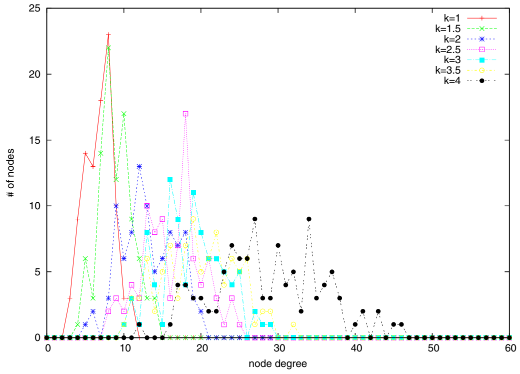

Fig. 6. Node Degree Histogram, $m = 8$

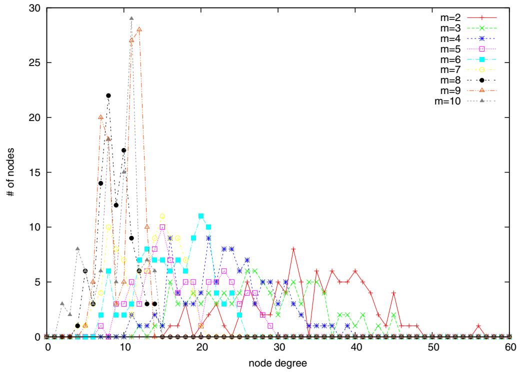

Fig. 7. Node Degree Histogram, $k = 1.5$

are defined depending on the number of adjacent squares of one square. This impacts on the node degree since a node inside a square totally surrounded by other squares will have a higher node degree than those which only have two adjacent squares. The impact of this differentiated zones are more important when the value of m is high and the value of k is low, fundamentally because the squares are not overlapped and there are many squares on every zone.

D. Average Connected Components

Figure 9 shows the variation of the average number of connected components in function of different values of m and k. This values are normalized by the number of nodes in the simulation. The number of connected components only vary on values of k less than one where the different zones defined do not overlap and even more they are not adjacent.

For the average connected components we do the same specific study of fixed values of m and k. For $m = 2$ and for $k = 0.5$ we obtain figures 10 and 11 respectively and the following functions:

$$cc_{m=2}(k) = 60.87 * log(3.24 * k) + 0.77 \tag{8}$$

$$cc_{k=0.5}(m) = 31.38 * e^{0.12*m} - 9.94 \tag{9}$$

In this case the results are not as interesting as for the average node-degree because so as to see a variation in the number of connected components the value k needs to be less than 1. This means that the squares in which the scenario is divided are separated one of each other, clustering the nodes.

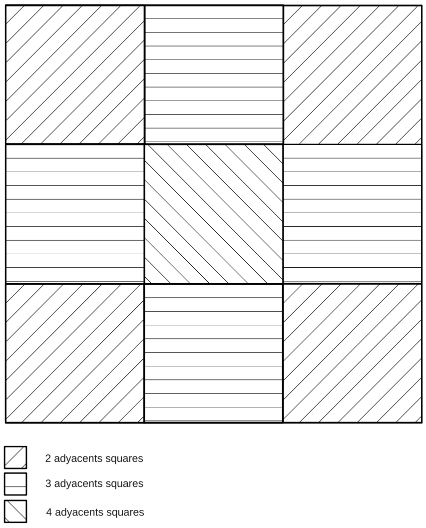

Fig. 8. Model Zones

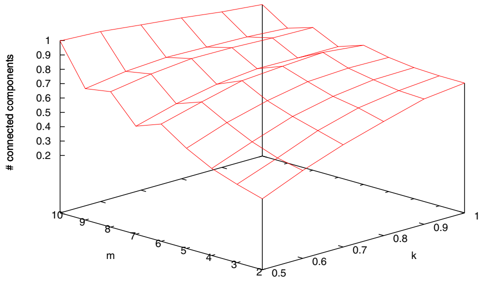

Fig. 9. Normalized Average Connected Components

E. Conclusions of the Study

From this detailed study we can conclude that the developed model meets the goal of finding a way of controlling the connected components and node degree metrics. The equations found show this fact although they are an approximation for an specific setup of the model. For a different setup (number of nodes, speed, size) this relations will vary. We can also mention that the approximation for the average node-degree is more accurate for large values of m and small values of k (see figures 4 and 5).

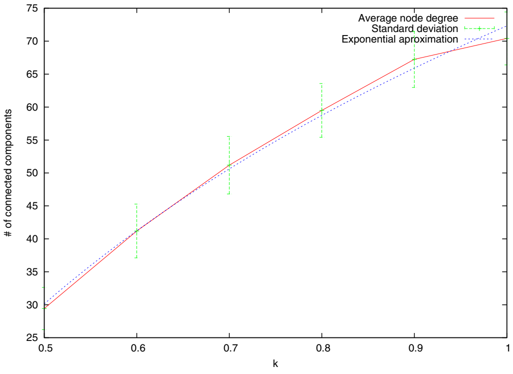

Fig. 10. Average Connected Components, $m = 2$

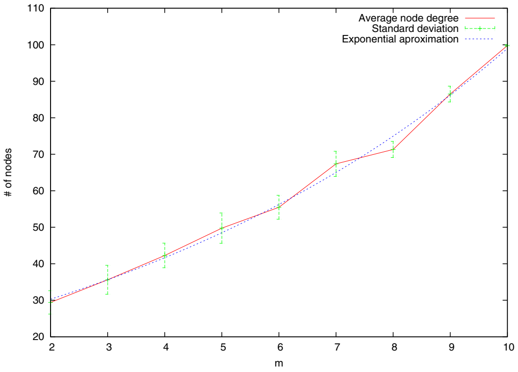

Fig. 11. Average Connected Components, $k = 0.5$

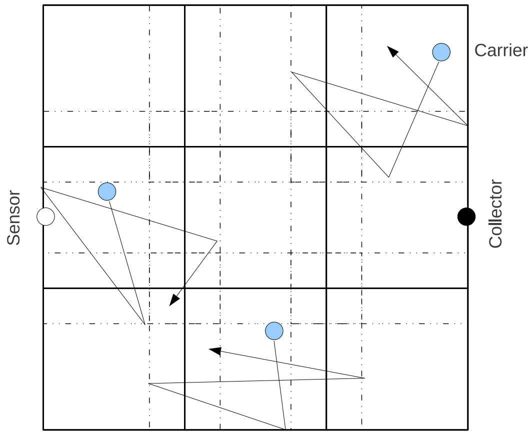

Fig. 12. Scenario for the Protocol Evaluation

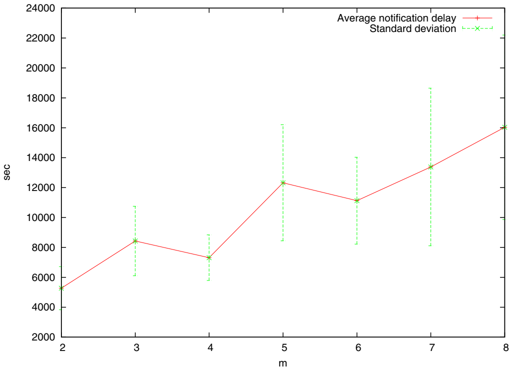

Fig. 13. Average Delay of Notifications

V. Applying the Model

In this section we experimentally evaluate the RON protocol using our mobility model. The objective of these experiments is to evaluate how the transmission delay vary in different scenarios with different numbers of node degree. For this we will apply Equation 7 to transform the parameter m to the average node-degree.

In RON the transmission delay is defined as the time needed for a notification from the sensor to reach the collector. For the experiment we will consider the average delay from all the notifications that arrived.

The experiment presented below was performed on a modified version of the NS3 Simulator. The modifications were centered on the capacity to execute applications such as the Lua virtual machine, on top of the simulated node without any relevant code modification. In this manner, the exact same Lua code developed to run on DEMOS devices is the code tested through simulations/emulations. The modified NS3 code was integrated into the main NS3 code as a contribution [8].

A. Evaluation Setup

1) Network: The setup common to all the experiments is a network of one sensor node, a variable number of carriers and one collector (see Figure 12).

2) Simulation Parameters: The parameters of the simulation are the same of the model evaluation (see Table I). A subscription is issued by the collector at time 0 and a matching notification is issued by the sensor every 500s. Each simulation run lasts 100,000s of simulated time.

B. Study on the Message Delay

In this study, we show the relationship between the transmission delay of a message and the average node-degree of the network.

Seven different simulations were executed with different values of m and the notification delay registered (see Figure 13). Then, Figure 14 shows the notification delay in function of the average node-degree. This graph was obtained by applying Equation 7 to the results of Figure 13.

In Figure 14 we can see that the node degree in the network impacts in the transmission delay in the expected manner: as the node degree increases, the transmission delay decreases.

VI. Related Work

The most basic and common mobility models can be found on this surveys [9] and [10]. In all of the models presented on these surveys in order to modify the metrics we want to fix, the parameters that are available to tune are only the node density, the speed of nodes and the transmission range. Some models also have some specific parameters as pause times or nodes per group. To our knowledge there is no model

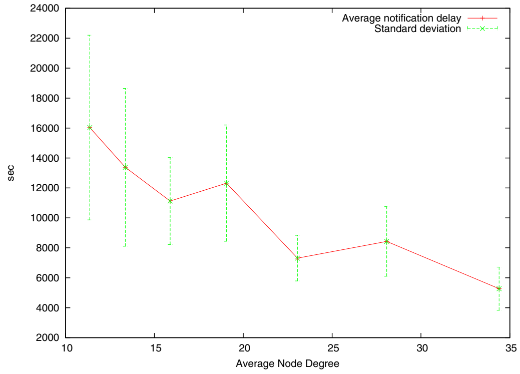

Fig. 14. Average Delay of Notifications

capable of maintaining the parameters mentioned above steady (fundamentally the node density, speed and range) and at the same time allow to modify some metrics like the node encounters or degree.

Some works show the dependencies mentioned above and also study how connectivity can influence the protocol's performance. In [11], equations for the meeting time between nodes are presented for the Random Direction and Random Waypoint mobility models were dependencies between node density, speed and range and meeting time are deducted. Then, in this same article, these equations are used to determine bounds for the delay with any routing protocol. The work in [12] analyze the accordion phenomenon using metrics very similar to us. They study the impact of the average node-degree and connected components (not the same definition we use) on the average delay. The authors conclude that, in the data collected on their study, the oscillation on the connectivity affects the dissemination of data, and moreover, the delay of messages is higher when the node density is low or the network is disconnected. Finally, a new routing scheme is proposed to mitigate the accordion phenomenon effect. Some conclusions from this work show the same behavior that we found in our experiments. For example it is found that in an epidemic routing algorithm (RON behaves similarly with one notification and one subscription), the expected delay decreases as the node degree increases. In [4] authors study the connectivity in function of the node density and the radio transmission range. They study the problem analytically and found equations to achieve networks where each node has at least N neighbors and where if $k-1$ nodes fail the network is still connected (k-connected network) and show the importance of the graph connectivity characteristics in evaluating DTN's. They introduce the idea of modeling an ad-hoc network with random geometric graph which we use in our work.

VII. CONCLUSION AND FUTURE WORK

In this paper we present a new mobility model, designed to help with the evaluation of opportunistic routing protocols. The main feature of this model is the configuration of its average number of connected components and the average node degree without modifying the node density, the transmission range or the speed. We show how we control those characteristics and we present an example of how the model is used in the evaluation of RON, a content-based opportunistic routing protocol.

The model presented in this paper needs more evaluation, and an interesting work that remains to be done is its validation in real-world deployments, however, we believe that the results shown in this paper encourage that kind of future work.

The DEMOS system is in its developing stage, therefore, the model should be extensively applied to evaluate and test the protocol. The routing protocol presented has many aspects to be improved and we expect to find them using the model on different simulation setups. Further, another work to be done is using the model to evaluate a different opportunistic routing protocol in order to determine if the model is general enough.

ACKNOWLEDGMENTS

This work was partially funded by the Latin American and Caribbean Collaborative ICT Research Federation (LACCIR), the Comisión Sectorial de Investigación Científica and the Agencia Nacional de Innovación e Investigación of Uruguay.

REFERENCES

[1] A. Balasubramanian, B. Levine, and A. Venkataramani, "Dtn routing as a resource allocation problem," New York, NY, USA, pp. 373–384, 2007. [Online]. Available: http://dx.doi.org/10.1145/1282427.1282422

[2] J. Baliosian, J. Visca, M. Richart, G. Apollonia, L. Vidal, M. Giachino, and E. Grampín, "Self-managed content-based routing for opportunistic networks," in *IFIP/IEEE International Symposium on Integrated Network Management, 2011. IM '11.*, 1-5 2011.

[3] "ns3 network simulator." [Online]. Available: http://www.nsnam.org/

[4] C. Bettstetter, "On the minimum node degree and connectivity of a wireless multihop network," in *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, ser. MobiHoc '02. New York, NY, USA: ACM, 2002, pp. 80–91. [Online]. Available: http://doi.acm.org/10.1145/513800.513811

[5] F. De Pellegrini, D. Miorandi, I. Carreras, and I. Chlamtac, "A graph-based model for disconnected ad hoc networks," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, may 2007, pp. 373 –381.

[6] M. Piórkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, "On clustering phenomenon in mobile partitioned networks," in *Proceeding of the 1st ACM SIGMOBILE workshop on Mobility models*, ser. MobilityModels '08. New York, NY, USA: ACM, 2008, pp. 1–8. [Online]. Available: http://doi.acm.org/10.1145/1374688.1374690

[7] "FriisPropagationLossModel Class Reference." [Online]. Available: http://www.nsnam.org/doxygen-release/classns3_1_1_friis_propagation_loss_model.html

[8] "ns-3-dce." [Online]. Available: http://code.nsnam.org/mathieu/ns-3-dce/rev/504423977908

[9] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, 2002. [Online]. Available: http://dx.doi.org/10.1002/wcm.72

[10] F. Bai and A. Helmy, "Chapter 1 a survey of mobility models in wireless adhoc networks."

[11] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Performance analysis of mobility-assisted routing," in *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, ser. MobiHoc '06. New York, NY, USA: ACM, 2006, pp. 49–60. [Online]. Available: http://doi.acm.org/10.1145/1132905.1132912

[12] P.-U. Tournoux, J. Leguay, F. Benbadis, V. Conan, M. de Amorim, and J. Whitbeck, "The accordion phenomenon: Analysis, characterization, and impact on dtn routing," in *INFOCOM 2009, IEEE*, april 2009, pp. 1116 –1124.