Modeling and Simulation of Mobility of Crowds

Sushma Patil, Eitan Altman, Manjesh K Hanawal
Team MAESTRO, INRIA, Sophia Antipolis, FRANCE
{shanawal,eitan.altman, mhanawal}@sophia.inria.fr

Julio Rojas-Mora
Institute of Statistics, Universidad Austral de Chile, Chile
jcredberry@gmail.com

***Abstract*—Mobility models studied in the networking community usually assume independence between the movement of individuals. While this may well model sparse networks, there are many scenarios that might not follow this assumption. In contrast, within other communities, such as road traffic engineering, biology and computer graphics, models of mobility usually take into account the dependence of the mobility pattern of an individual with respect to that of its neighbors. Our goal in this paper is to study how this dependence impacts the performance measure from the networking point of view. In particular, we implement a bio-inspired model for mobility of crowds and, by simulation, we study how mobility influences the performance measures of a distributed network. We perform statistical analysis on the samples obtained through simulations. In particular, we study the distribution of the message delivery time and show that it is light tailed, with exponential tail distribution.**

***Index Terms*—Mobility, flocking, statistical analysis**

I. Introduction

Mobility of nodes in a wireless network have a direct influence on the performance of various protocols used for message delivery; performance metrics like delay in message delivery, energy consumed, probability of successful message transmission, etc., are functions of mobility pattern of nodes. It is desirable to know the performance of a protocol by testing it on real world traces of the node mobility patterns collected through experiments. However, often many a times traces of the desired scenario are not available as the wireless network may not yet be deployed. In such scenario, one relies on the data generated from synthetic models to study network performance. Synthetic models that describe a mobility scenario under consideration, in a realistic fashion, can be complex. The models available in literature tradeoff between ease of implementation/analysis and being more realistic. Depending on which synthetic model best captures node mobility, the choice of the synthetic model is made for evaluation of the performance of a protocol.

Quite often, simple mobility models like Random Waypoint or its variants [6] are assumed as they are relatively easy to implement and analyze [5]. These models assume that movement of each mobile node (MN) is independent of others, and also their movement does not depend on their past locations or speeds. These assumptions are justifiable in a very sparse network where MNs are spread far from each other and hence their movement can be assumed to be uncorrelated. These models, though easy to analyze, lead to unrealistic scenarios, like sharp turns or sudden stops. To avoid such cases models are proposed that correlate a node's mobility on its past movements to varied degrees of generality, but with increasing complexity of tractability. In Gauss-Markov models [7], current location of a node is made to depend on its past location through a controlled parameter. For very thorough surveys on mobility models see [4] and [14].

In many scenarios MNs may not move independent of each other, but in a group, or following a particular reference node in the group. For example, a group of people visiting a museum or touring a city following an instructor. To cater to this kind of dependent mobility, several models are proposed by the network community. In the Reference Point Group Mobility (RPGM) model all of the nodes follow a path traveled by a logical center [8]. The logical center may be made to follow a predefined path or can follow random waypoint mobility. The other MNs move according to some random waypoint in the vicinity of the logical center. Several other useful group mobility models can be derived as variants of RPGM: column group mobility, nomadic group mobility, pursue mobility [4], etc.

We shall be interested in crowd movements where MNs are humans carrying wireless terminals. Here the MNs may not follow any particular MN, nor their movement is independent of each other. In such crowd movement, they avoid colliding with each other and maintain a safe distance from their neighbors. For example, people moving in a market or a busy commercial area. In this paper, we aim to study such crowd mobility where mobility of a MN depends on its neighbors. In biology, such kind of mobility is extensively studied to understand movement of flock of birds, school of fish or heard of animals. Rules governing such mobility are extremely interesting if one wishes to recreate patterns of such flock movements through artificial life. Indeed, scientists in the field of computer graphics synthesize mobility of such crowds to create beautiful patterns [1]. To synthesize such patterns they implement rules that govern how mobility of an entity in the group depends on its neighbors.

Craig Reynolds [9] proposed a systematic way to synthesize aggregate motion of crowds using computer graphics. He referred to each entity in the crowd as 'boid'. The synthesized movement of the boids is based on his finding that the complex auto-organization of the group into complex macroscopic patterns is determined by three simple microscopic rules that each individual in the group follows: (i) Boids try to fly towards the center of mass of neighboring boids. (ii) Boids try to keep a small distance away from other objects (including other boids). (iii) Boids try to match velocity with near boids. If each boid is implemented as an independent actor, which navigates according to its local perception of the dynamic environment obeying the above rules, the interaction of the boids results in the aggregate action that looks very synchronized as if centrally controlled.

The primary aim of [9] is to recreate beautiful patterns observed in nature. Variants of the rules proposed by Reynolds are used to study local interaction and evolutionary patterns in biology [1]. In one of such studies, Ariel Dolan [2] implemented creatures that guard their territory against intruders, and study evolutions of populations. He referred to such life-like creature as 'Floy', which interact with their neighbors as follows: (i) stay close to your fellows but not too close. (ii) if you see an intruder, move towards it and attack. The first rule of Dolan is similar to the first two rules of Reynolds, but they differ in how the Floys track its neighbors: In Dolan's Floy model, each Floy tracks two of its randomly chosen neighbors and tries to be close to them, whereas in Reynolds' model, each boid tries to be in the center of its neighbors that are within a certain radius. Also, in Dolan's model there is no matching of velocities among the Floys.

In Reynolds' boids model the three rules make the mobility of boids highly dependent on each other; the boids stay close to each other, and result in patterns that give a feeling that all the boids are guided centrally. However, relaxed rules of Dolan's Floys model makes Floys to avoid too much togetherness, which lead them to move in an ensemble that spreads over a wide region. Each Floy may stray away from its group and wander somewhat randomly in a given territory around its group members. But it will not go too far away, as soon it will tend to move towards its group due to the first rule. This behavior is similar to that of a crowd movement where the nodes move from one location to other. Obviously, MNs avoid colliding with each other and do not stray far from its neighboring fellows while it moves in a given territory. Thus, we can use rules similar to that in the Floy model to simulate and study crowd movements in ad hoc networks.

Our aim in this paper is to study performance in our crowd mobility model, from the networking point of view. We consider the delay tolerant network scenario. Each MN in the crowd can have a message that another MN in the crowd is interested. The MN which have this message (source MN) can spread the message in the crowd by giving it to any other MN that comes close to it. The message is thus relayed among the crowd and can finally reach the MN that is interested in this message (destination MN). We shall study the time taken for a message sent by a source MN, using the other MNs as relays, to reach the destination MN. In particular, we shall be interested in identifying the tail distribution of the message transfer delay and its statistics, i.e., the probability that the message takes larger than a threshold time to reach destination. Such analysis is of primary importance as a message may become irrelevant if it does not reach the destination within a stipulated time. We also study how the transmission distance influences the message delivery time. This is equivalent to analyzing energy-delay trade off in wireless networks [15], as larger transmission range means more power transmission with lesser delay and vice versa.

The paper is organized as follows: In Section II, we discuss the simulator we built to study crowd mobility based on Craig Reynolds' boid model. In Section III we discuss how we adopt the boid model to study network crowd movement and explain the simulation settings. In Section IV we discuss the statistical results of the message delivery time. In Section V we discuss the effect of transmission range on the message delivery time. Finally, in Section VI we discuss the possible extensions and summary of our work.

II. Crowd Mobility Simulator

In this section we discuss the simulator we built to study crowd mobility based on Reynolds' boid model. We refer to each boid as a node, henceforth.

Below we discuss how each node updates its location and velocity according to Reynolds' three rules discussed in the introduction. In the 2-D version of the simulator each node begins with a fixed velocity v_0 at an arbitrary location in a rectangular region. In applying rule 1 (cohesion), each node looks for other nodes within a radius, say R_1, and finds the position that is the center of mass with respect to the location of these nodes. In applying rule 2 (separation), each nodes looks for other nodes within a radius, say R_2, and finds the position that is the center of mass with respect to the the locations of these nodes. In applying rule 3 (alignment), each node looks for relative velocity of the other nodes within a distance, say R_3, with respect to itself, and finds the average velocity. The new velocity is obtained by combining the values obtained from the three rules with weights γ_1, γ_2, and γ_3, respectively, and adding to the current velocity. For example, let $A_1 = (a_1, b_1)$, $A_2 = (a_2, b_2)$, and $A_3 = (a_3, b_3)$ be the values obtained from rule 1,2, and 3, respectively, for a node which has current velocity $V_n = (Vx_n, Vy_n)$ and is at position $X_n = (x_n, y_n)$. Here the first component in each vector corresponds to the x-axis, and the second to the y-axis. Then, the node moves to the new position $X_{n+1} = (x_{n+1}, y_{n+1})$ with velocity $V_{n+1} = (Vx_{n+1}, Vy_{n+1})$, given by

$$V_{n+1} = V_n + \gamma_1 A_1 - \gamma_2 A_2 + \gamma_3 A_3$$

and

$$X_{n+1} = X_n + \Delta * V_{n+1},$$

where Δ is a constant that governs how fast the nodes update their position.

Note that the weights γ_1, γ_2, and γ_3 govern the relative importance assigned to each rule. By adjusting these parameters one gets varied degree of dependency among the nodes' movement. For example, if γ_1 is large while γ_2 and γ_3 are small, then nodes stay very close to each other, but seem to change directions and come close to each other often. If γ_2 is large while γ_1 and γ_3 are small, then nodes are equally spaced from each other but move more randomly and change direction often. If γ_3 is large while γ_1 and γ_2 are small, then all the nodes align themselves in a particular direction and continue

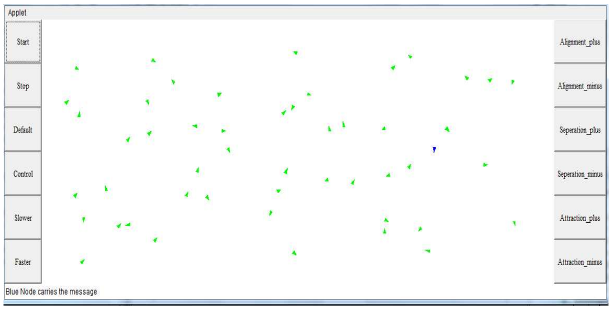

Fig. 1. Snap shot of Crowd Mobility simulator: Blue node has message

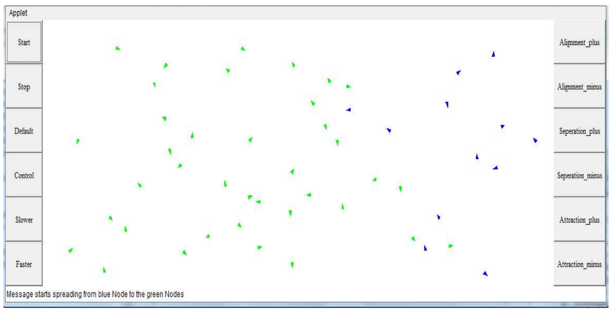

Fig. 2. Snap shot of Crowd Mobility Simulator: Blue nodes have the message and green ones are yet to receive it

to move in the same direction till they hit the boundaries. By varying these weights one gets varied degree of randomness in the movements of the nodes. We put an upper limit on the velocity of each node. If the velocity of a node exceeds this limit, denoted by $v_{\max}$, then its velocity is set to $v_{\max}$.

We implemented Reynolds' three rules in Java with an interface to control the weights. A snapshot of our implementation is shown in figures 1 and 2. In the simulator, the number of nodes (N), and the parameters R_1, R_2, and R_3 can be set. If the values of R_1, R_2, and R_3 are high, i.e., nodes take into account movement of other nodes from a larger area in deciding their new velocity, and the movement becomes more dependent. When the values of these parameters are decreased, the movement tends to become more random. Thus, by changing various parameters in the simulator, the movement can be changed from being completely random to perfectly synchronized.

In building the aforementioned simulator, we used the Dolan's Java templates which he developed to implement his Floy model. We implemented Reynolds' boid with the help of these templates and built in various functionalities to control movements of boids that help us study crowd mobility in ad hoc networks as discussed in the following sections. The simulator and its source code are available online at [17].

III. An adaptation of boids model to networking

In crowd movement, people try to avoid colliding with each other by maintaining a safe distance, while also not straying far away from its neighbors. This scenario can be simulated by appropriately setting the weights γ_1, γ_2, and γ_3 and the parameters R_1, R_2, and R_3 in the simulator. Note that in crowd mobility, velocity matching is not an important criterion. However, avoiding collisions and not straying away far from neighbors are important conditions, with the former having more priority. These criteria correspond to setting $\gamma_3 = 0$, and $\gamma_2 > \gamma_1$ in the simulator.

We adopt the boid model to study crowd movement in wireless ad hoc networks. In particular, we will be interested in scenarios like delay tolerant networks (DTNs) where end-to-end connectivity between source and destination is not available. Everybody in the crowd is assumed to have a transmitter and receiver, and can transmit, receive and relay messages. We henceforth refer to each boid-like entity or each person in the crowd as a "mobile node" (MN). The MNs are battery powered and can not transmit beyond a certain transmission range. Let T denote the transmission range of each MN. They can transmit or receive messages when within distance T from other MNs. If the other MNs already have this message, then they simply ignore the message.

In this preliminary study, we consider epidemic routing scenario in a delay tolerant network [11], i.e., a MN having the message transmits it to all the MNs that are within the transmission range T. We leave other protocols like, two hop routing, direct contact, spray and wait [13] for future study. Also, we assume that the transmissions are error free and instantaneous. Further, all the nodes are identical and participate equally in relaying the message. In the simulator it is possible to set a certain fraction of MNs not to participate in message spreading. However, in this paper we restrict to the case where all the nodes participate in message spreading.

In epidemic routing the copy of the message is available with a large number of MNs and it is likely that it will reach the destination with high probability. But the time required for the message to reach the destination, or the delay in delivery of the message, can be still very high. For example, small MN density, small transmission range, can result in longer time for two MNs to be within transmission distance of each other hence leading to large time delays. In DTNs, the message delivery is of primary interest than the time taken it to be delivered at the destination, but the sooner the message gets delivered the better: in some case the message may lose its relevance if it fails to reach within a stipulated time. Thus, it is of interest to know probability of message taking too long to be delivered at the destination. Or, more generally, to know the tail distribution of the delay.

IV. Experimental setup and statistical analysis

A. Experimental Setup

In this section we discuss simulation setup and data collection for empirical study of the delay distribution. In the simulator we set the parameters as follows: $R_1 = R_3 = 100, R_2 = 25, \gamma_1 = 0.001, \gamma_2 = 1.0, \gamma_3 = 0, v_0 = 2, \Delta = 0.095, T = 25, v_{\max} = 20$. The choice of these parameters is based on visual monitoring, such that the simulated mobility looks close to that of a crowd mobility. The above parameters are set as default values in the simulator. The simulation is run with 50 MNs that start at random locations with MN_1 as the source and MN_n as the destination. In each run, we set a warmup period of 30 seconds before collecting any data. Once this warmup period lapses, we start a timer at the destination and the note the time when it receives the message for the first time. We made 500 runs to collect samples of the random variable D.

In this paper, we present statistical analysis with the parameter set to default values. However, we note that the same observations continues to hold for other set of parameters as well. In [18] we present numerical results for a range of parameters and some preliminary observation on sensitivity analysis. In the following subsections, we explain the statistical analysis for one set of samples collected from the simulator. These samples are available online at [17].

B. Statistical Analysis

In this section we analyze the delay distribution using statistical tools available in MATLAB. We obtain the empirical distribution using the samples generated through the simulator. It is observed that the exponent of the tail distribution is linear, and the slope is obtained using linear regression. Finally, we derive the theoretical distribution that best represents the tail of the delay distribution.

Let $F_D(\cdot)$ denote the cumulative distribution function (CDF) of D. We will characterize the delay distribution by evaluating the empirical distribution (ECDF) of D which we denote as $\hat{F}_D(\cdot)$. We also calculated upper and lower bounds (see Figure 3) that corresponds to 95% confidence intervals of the ECDF [12].

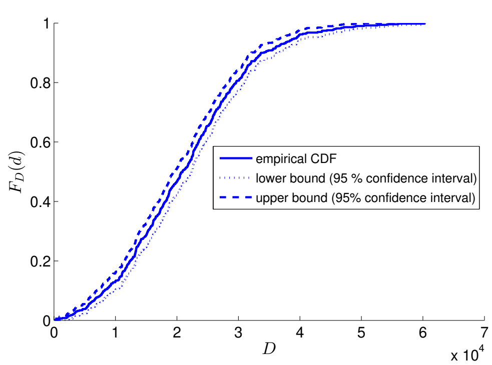

Fig. 3. empirical CDF of delay

We next evaluate the exponent of the tail distribution, i.e., rate of change or slope of:

$$y(d) = -\log(1 - \hat{F}_D(d)).$$

This function is plotted in Figure 4, as the ECDF, with its upper and lower confidence interval bounds. As seen from Figure 4, $y(d)$ is an almost linear function for large values of D.

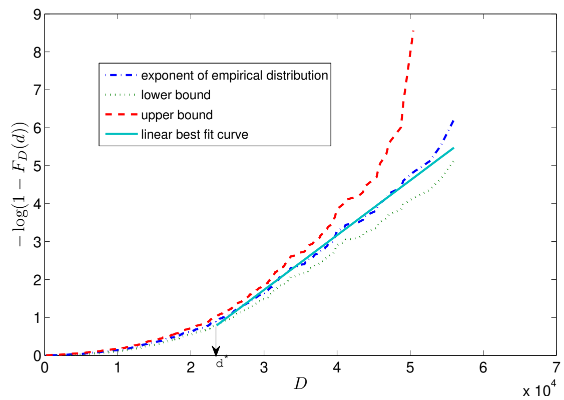

Fig. 4. Exponent of tail distribution

C. Linear regression of $y(d)$

In this subsection we evaluate the slope of $y(d)$ using linear regression. Let $d = (d_1, d_2, \cdots, d_m)$ and $f = (f_1, f_2, \cdots, f_m)$ denote the time sequence and the corresponding values of the cumulative probability. Define $y_i = -\log(1-f_i)$ for each $i = 1, 2, \cdots m$ and denote $y = (y_1, y_2, \cdots, y_m)$. As our interest is in the behavior of the tail distribution, we take into account only the last 40% of sample of the function (d, y) to evaluate the slope. Let M denote the size of this truncated vector (d, y), and re-index its samples from 1 to M. For the samples we have, the sample size after truncation is $M = 201$.

Before we proceed to apply regression analysis on the samples, let us first verify that there is significant linear relationship between d and y using regression the t-test. The *correlation coefficient* for the pair (d, y) is $R = 0.9960$, and the corresponding *test statistics* for the data is:

$$t = r\sqrt{\frac{M-2}{1-R^2}} = 156.7324,$$

whereas the critical value is 1.972 at *significance level* 0.05. Clearly, at significance level 0.05, there is a significant linear relation between d and y as $t > 1.9720$.

We next evaluate the slope and intercept of the linear function that best represents the linear relation between d and y. These values are given, respectively, as follows:

$$m = \frac{M\sum d_i y_i - (\sum d_i)(\sum y_i)}{M\sum d_i^2 - (\sum d_i)^2} = 0.0001443$$

and:

$$b = \frac{\sum y_i - m\sum d_i}{M} = -2.602.$$

The best fit linear function in shown in figure 4 in a continuous line. Note that, in the region of interest, its tail, the linear function lies within the upper and lower confidence interval bounds. Thus the logarithm of the tail of the ECDF can be approximated by the linear line with slope and intercept as computed above, and the confidence interval for this fit

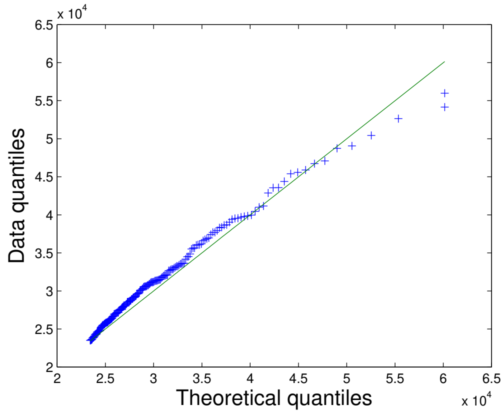

Fig. 5. q-q plot

is 95 %. Also, the adjusted R^2 value as a measure of the *coefficient of determination* for this fit is 0.9919. Hence, with a high confidence value, we can approximate $y(d)$ by a linear function, which in turn implies that tail of the empirical distribution can be approximated by an exponential function, i.e.:

$$\Pr\{D \geq d\} \approx \exp\{-md\} \ \ for \ \ d > d^*,$$

where $d^* = d_{N-M}$ is the threshold beyond which we apply the regression to fit $y(d)$ to the data, with m as the slope of the best fit linear function. Note that in arriving at the exponential approximation for the tail distribution we only took into account the slope of the linear regression and not the intercept. In the next section we study the goodness of fit of the exponential distribution to the tail of the empirical distribution by studying *quantile-quantile* plots and performing a Kolmogorov-Smirnov test.

D. Quantile-Quantile Plot

The exponential approximation of the empirical distribution is good when the samples are larger than the threshold d^*. Let $\bar{D}$ denote the distribution of the delay conditioned that it is larger than d^*. Then the CDF of $\bar{D}$ is given by

$$F_{\bar{D}}(d) = \begin{cases} 1 - \exp\{-m(d - d^*)\}, & \text{if } d \geq d^* \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

The q-quantile of this CDF for a given $p \in [0 \ 1]$ can be evaluated as

$$F_{\bar{D}}^{-1}(p) = d^* - \frac{\log(1-p)}{m},$$

where $F_{\hat{D}}^{-1}(p)$ is the inverse of the CDF of the tail. We compare this theoretical quantile function with the quantile of the ECDF obtained from the samples collected. For a fair comparison, we discard all the samples that are smaller that d^* in computing the ECDF. The q-q plot is shown in figure 5. Note that the quantile functions closely follows the 45 degree line, with deviations only at the edges. This is an expected behavior as we are approximating an exponential function with a linear fit. Thus, we overstimate the quantiles at the lower end of the curve, while overstimating them at the upper end. Nevertheless, the correlation coefficient between the two quantile functions is 0.9938. Hence, the exponential approximation of the tail distribution is a good fit.

We next perform the Kolomogrov-Smirnov (KS) test on the data samples. The test statistic for this test is

$$D = \max_d |F_{\bar{D}}(d) - \hat{F}_{\bar{D}}(d)| = 0.0978,$$

where $\hat{F}_{\bar{D}}$ denotes the ECDF of the samples that takes value larger than d^* in the data. At a significance level of 5%, the hypothesis that the samples are drawn from the exponential distribution with $\lambda = m$ is not rejected by the KS test. Thus, it can be believed that the tail is exponentially distributed.

V. Delay-Energy Tradeoff

In this section we study the effect of transmission range T on the mean message delivery time. If nodes can transmit at higher power, then they can transmit the message successfully over a larger distance. However, since the nodes are battery powered, this will require the nodes to recharge often. In Figure 6 we plot the mean message delivery time versus the transmission range T. In generating the plots we set the parameters to the same values given in Section III. Note that if $T < 25$, then the nodes come rarely within the transmission range of each other as they will try to maintain a separation of 25 units ($R_2 = 25$). Thus, we generate the plot for $T \geq 25$. We also obtain the linear regression of the data points and plot it in Figure 6. From the figure we note that there is a linear relation between the mean message delivery time and the transmission range. Thus, to decrease the mean transmission delay by a certain factor, the MS need to increase the transmission range by the same factor.

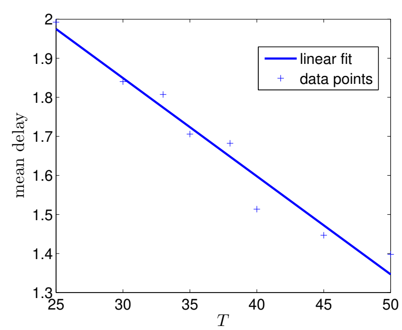

Fig. 6. Transmission range vs mean delay

VI. Conclusion

Mobility modeling is crucial to analyze the performance of wireless networks. To understand the performance of a protocol, simplifying assumptions, like independence of node mobility, are made in wireless networks, which is not realistic. In this paper, we proposed a model to study crowd mobility that captures dependency among the movements of individuals in a group. We implemented a simulator in which the degree of dependency of movements among the individuals can be varied by controlling the parameters. We established through systematic statistical analysis that the distribution of message delivery time is light tailed with exponential distribution.

In this preliminary study, we have not take into account errors in transmissions and minimum contact duration for successful transmission. In future, we like to bring in these aspects into simulations. Also, in the current model, when two nodes come close to each other they sometimes make a complete U-turn, which is not realistic. We would like to make such turns smooth, and study how this impacts the results.

REFERENCES

[1] www.red3d.com/cwr/boids/

[2] http://www.aridolan.com/ofiles/eFloys.html

[3] http://www.aridolan.com/ofiles/Floys2.aspx.

[4] T. Camp, J. Boleng, and V. Davies, A Survey of Mobility Models for Ad Hoc Network Research, in Wireless Communication and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, vol. 2, no. 5, pp. 483-502, 2002.

[5] C. Bettstetter, G. Resta, and P. Santi, The Node Distribution of the Random Waypoint Mobility Model for Wireless Ad Hoc Networks, IEEE Trans. Mobile Computing, vol. 2, no. 3, pp. 257-269, July-Sept. 2003.

[6] E. Hyytia, P. Lassila, and J. Virtamo, Spatial Node Distribution of the Random Waypoint Mobility Model with Applications, IEEE Trans. Mobile Computing, vol. 5, no. 6, pp. 680-694, June 2006.

[7] B. Liang and Z. Haas. Predictive distance-based mobility management for PCS networks. In Proceedings of the Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), March 1999.

[8] X. Hong, M. Gerla, G. Pei, and C. Chiang. A group mobility model for ad hoc wireless networks. In Proceedings of the ACM International Workshop on Modeling and Simulation of Wireless and Mobile Systems (MSWiM), August 1999.

[9] Craig Reynolds, Flocks, Herd, and Schools: A Distributed Behavioral Model, Com- puter Graphics, 21(4), SIGGRAPH'87, pp. 2534.

[10] R. Groenevelt, P. Nain, Message delay in MANETs, in: Proc. of ACM SIGMETRICS, Banff, Canada, June 6, 2005, pp. 412-423; see also R. Groenevelt, Stochastic Models for Mobile Ad Hoc Networks. PhD Thesis, University of Nice-Sophia Antipolis, April 2005.

[11] E. Altman, T. Basar, and F. De Pellegrini, Optimal monotone forwarding policies in delay tolerant mobile ad-hoc networks, in Proc. of ACM/ICST Inter-Perf. Athens, Greece: ACM, October 24 2008.

[12] M. Greenwood, "The natural duration of cancer." Reports on Public Health and Medical Subjects vol. 33, pages 126. Her Majestys Stationery Office, London

[13] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, Spray and wait: an efficient routing scheme for intermittently connected mobile networks, in Proceeding WDTN '05 Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking Pages 252-259.

[14] Radhika Ranjan Roy, Handbook of Mobile Ad Hoc Networks for Mobility Models, Springer, New York Dordrecht Heidelberg London.

[15] T. Small and Z. J. Haas, "Resource and Performance Tradeoffs in Delay-Tolerant Wireless Networks," SIGCOMM05 Workshops, August 2226, 2005, Philadelphia, PA, USA.

[16] R. Agarwal, A. Banerjee, V. Gauthier1, M. Becker1, C. K. Yeo, and B. S. Lee, "Achieving Small-World Properties using Bio-Inspired Techniques in Wireless Networks," The Computer Journal, Vol. 55, no. 8, pages 909-931, Aug 2012.

[17] http://www-sop.inria.fr/members/Manjesh_Kumar.Hanawal/sushma/mobility.html

[18] http://www-sop.inria.fr/members/Manjesh_Kumar.Hanawal/sushma/Draft.pdf