Design and Evaluation of an Individually Simulated Mobility Model in Wireless Ad Hoc Networks

Desney S Tan[1], Shuheng Zhou[2], Jiann-Min Ho[3], Janak S Mehta[3], Hideaki Tanabe[3]
{School of Computer Science[1], Electrical and Computer Engineering Department[2], Information Networking Institute[3]}
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA 15213 USA
desney@cs.cmu.edu, {szhou, jiannmin, jsmehta, htanabe}@andrew.cmu.edu

Keywords: Wireless ad hoc networks, individually simulated behavioral model, mobility metric, Boids.

Abstract
Since there has been little focus on systematically creating user mobility and communication pattern traces, researchers in the mobile networking community often propose personal models with which to validate their routing algorithms. This approach has two problems: (1) invalid conclusions may be drawn from overly simplistic or unrealistic models, and (2) it is difficult to compare performance results of different algorithms due to the variety of models used. We describe a novel approach to generate user mobility patterns with an individually simulated behavioral model. In this model, the overall mobility pattern is the result of the interaction between the behaviors of individual nodes. We show that this general-purpose framework can reliably reproduce existing mobility patterns as well as create useful new ones. The model is validated with the statistical characteristics of representative scenarios and through the simulation with the Dynamic Source Routing (DSR) protocol.

1. INTRODUCTION
We believe that computer usage is moving towards a model of mobility. Users are migrating from their traditional desktop personal computers to laptops and portable computers with wireless communication capabilities. For efficiency and convenience, these users would naturally like to be able to share information with minimal mobility constraints and administration costs. To this end, much work has been done in the construction of wireless ad hoc networks – self-organizing wireless networks that require no infrastructure and no administrative intervention. Many protocols have been developed to solve the numerous problems encountered in ad hoc networks. Simulating these networks during the evaluation of a protocol requires careful consideration and design of mobility and communication patterns, which may drastically affect the performance results of a given algorithm.

Currently, designers of protocols provide performance analyses based on simulations run with "custom-made" user mobility and communication patterns. Existing mobility models employ a method of global simulation, in which movement of individual nodes is coordinated by a central control system. Each of these user models is based on certain intuitions and assumptions and might not correctly model realistic use of the system. Also, because designers of each protocol implement different models to demonstrate performance of their algorithms, it is difficult to compare the performance of any two protocols.

There is no indication that there is a single-most "correct" mobility pattern that will accurately capture all the behavior of an ad hoc networking scheme. In fact, such a model, by definition of the use of the system, would be relatively useless and unrealistic at best. There are two possible approaches to tackle the problem of designing realistic and reproducible mobility and communication models. In the first approach, we study and conceive actual use of the systems and create models that most closely represent the real world. Although this approach guarantees that we do not stray from realistic and relevant models, it could potentially lead to overly complex models that are hard to analyze. In the second approach, we view the creation of such models much like the creation of benchmark programs for computer systems. In this method, we would systematically isolate the factors that affect performance of different protocols and design mobility and communication models that specifically stress certain subsets of these factors. Although this approach would allow us to define a standard suite of benchmark patterns that may be used to isolate strengths and weaknesses of different protocols, it may also lead us to produce models that are unrealistic and model characteristics that are of little concern.

In our work, we use a combination of the two approaches and present a general-purpose method that may be used to reliably generate realistic mobility patterns with different characteristics. Rather than using a global control mechanism, we utilize a model of individually simulated nodes. Section 2 examines previous work done in this area.

Section 3 describes our model, in which the overall movement pattern is a result of the close interaction of relatively simple behaviors of each of the individual nodes. Section 4 presents a naïve difficulty metric that expresses the expected difficulty of each of the scenarios. Section 5 documents the implementation and process of generating mobility patterns. Section 6 presents results showing that the model does indeed produce stable reproducible patterns that stress the DSR protocol in different ways. Sections 7 and 8 discuss further directions and conclude.

2. RELATED WORK

Many researchers use the random mobility model [10,12]. According to this model, the speed and direction of motion in a new time interval have no relation to past values. This model can generate unrealistic mobile behavior such as sharp turning or sudden stopping. In an attempt to remedy this discontinuous randomness, variations of this model have been created [1,2]. These models vary the amount of randomness that is displayed in the speed, direction, and pauses between movements.

Broch et al. present the results of a detailed packet-level simulation comparing four multi-hop wireless ad hoc network routing protocols: DSDV, TORA, DSR, and AODV, using a modified "random waypoint" movement model and a communication model based on constant bit rate (CBR) traffic sources [2]. An extension made to these set of models considers the relationship between a mobile host's previous behavior and the current movement speed and direction. Haas presents an incremental model in which speed and direction randomly diverge from the previous speed and direction [4]. Hong et al. recognize the group-based communications that are typical of real use and propose a group mobility model that organizes mobile hosts into groups according to their logical relationships [5]. Noble et al. use real user traces to collect mobility profiles [9]. Using this method, data although dependable, is difficult and tedious to collect.

Although most protocols are designed to be adaptive to the mobility and activity of the nodes, few researchers present very comprehensive sets of mobility models to test against their protocols. Simulation results have shown that the choice of mobility model makes a difference in the physical link dynamics and performance [5]. Little research is being done to create a set of models that can be easily used to evaluate protocols. Of this work, Sanchez considers different human or robotic moving behavior in different situations, formulating such models as Brownian motion, Column model (Scanning, Searching), Pursue model (Target Tracking), and Nomadic community [12].

Separate research has been done in the computer graphics and animation realms to simulate the aggregate motion of natural occurring phenomena such as flocks of birds, herds of land animals, or schools of fish. Work done in this arena has built upon the principles of particle systems [10], which have been used to model such phenomena as fire, smoke, clouds, and the spray and foam of ocean waves. These systems are collections of large numbers of individual particles, each having its own behavior. During their life they have certain behaviors that can alter the each particle's own state, usually consisting of color, opacity, location, and velocity. In particular, evolving from this work, the Boids system explores an approach based on simulation as an alternative to scripting the paths of each bird, animal, or fish (hereafter referred to as nodes) individually [11]. The aggregate motion of the simulated flock is created by a distributed behavioral model much like that at work in a natural flock, in which each of the nodes chooses its own course. We build upon these models to design our framework for creating realistic human group motion.

3. THEORY OF INDIVIDUALLY SIMULATED BEHAVIORAL MODEL

In the design of our model, we assume that group movement is the result of interaction between behaviors of individual nodes. While each node acts separately from all other entities around it, its decisions are somehow affected by the context it is in and by the movement surrounding it. For example, in an event hall, even though each individual exercises freedom of movement independent of any other individual in the hall, it may follow general trends of movement. In this case, individuals might tend to move in small groups and stop together at certain points to view exhibits; later, an individual might decide to join another group as it passes by. If this simulated node model has the correct group-member behavior, all that should be required to create the desired simulated movement is to create some instances of the simulated nodes and allow them to interact.

There are three sub-models that define our system: the perception sub-model (what each node "sees"), the behavioral sub-model (what each node infers from the information available), and the movement sub-model (how each node actually moves). The perception model tries to make available to the behavioral model approximately the same information that is available to a real human as the end result of its perceptual and cognitive processes. Not only is it unrealistic to give each simulated node perfect and complete information about the world, it is just plain wrong and leads to obvious failures of the model. The behavioral model takes this information and decides how it should move based on current state, user parameters, and system defined parameters. The movement model takes input from the behavioral model and calculates the actual movement of the nodes so as to keep a consistent state of the world.

3.1. Perception Sub-model

We do not directly simulate the perception mechanism of humans in this model. Rather, the perception model makes available to each node roughly the same information that would be available to the human as the end result of its perceptual and cognitive processes. Each node has access only to a localized segment of the global geometric database that stores the position and velocity of all nodes. The localized segment is defined as a circular zone of sensitivity centered at the node's local origin. An additional advantage of this method is that since each node "sees" only a limited number of nodes regardless of the total number in the world, we can use a roughly constant time algorithm with few scalability constraints.

3.2. Behavioral Sub-model

In this model we make the assumption that human movement is composed mainly of three somewhat opposing behaviors: a desire to stay close to the group (which corresponds to having certain motivations or goals in the environment), to avoid collisions within the group (an obvious constraint of human movement), and to maintain current velocity (to avoid unrealistic human behavior) [11]. To build a simulated group, we start with a node model that supports constrained motion. We add behaviors that correspond to the opposing forces of grouping as well as collision avoidance and inertia. Stated briefly as rules, and in order of decreasing precedence, the behaviors that lead to simulated grouping are:

- Group Centering: the urge to stay close to nearby nodes
- Collision Avoidance: the attempt to avoid collisions with nearby nodes and with obstacles (currently only implemented as the environmental boundaries)
- Velocity Matching: the desire to match velocity with nearby nodes
- Inertia: the tendency to maintain current velocity

Group centering makes a node want to be near the center of the group. Because each node has a localized perception of the world, "center of the group" actually means the center of nearby nodes. Group centering causes the node to move in a direction that places it closer to the centroid of nearby nodes. If a node is deep inside a group, the population density in its neighborhood is roughly homogeneous, or the same in all directions. In this case, the centroid of the neighborhood nodes is approximately at the center of the neighborhood, so the group centering urge is small. But if a node is on the boundary of the group, its neighboring nodes are on one side. The centroid of the neighborhood nodes is displaced from the center of the neighborhood toward the body of the group. Here the group centering urge is stronger and the movement will be deflected somewhat toward the local group center.

Collision avoidance and velocity matching are complementary behaviors. Together they ensure that the nodes of a simulated flock are free to move within the environment without running into one another. Collision avoidance is the urge to steer away from an imminent impact. It is based on solely the relative position of the nodes and ignores their velocity. Conversely, velocity matching is based only on velocity and ignores position. It is a predictive version of collision avoidance: if the node does a good job of matching velocity with its neighbors, it is unlikely that it will collide with any of them any time soon. With velocity matching, separations between nodes remain approximately invariant with respect to ongoing movement. Collision avoidance serves to establish the minimum required separation distance; velocity matching tends to maintain it.

Inertia is the tendency of a node to keep its current velocity and direction. It is unnatural of human movement to rapidly oscillate back and forth, or to make excessive sudden movements in random directions. This behavior prevents singularities in the data (for example, a node exactly in between two groups would otherwise tend to oscillate back and forth) from causing unrealistic movements. Unlike the other three behaviors, inertia is based solely on the internal state of a node and needs no external information.

Each of the above behaviors produces an isolated suggestion about which way to steer the node, expressed as an acceleration request. These prioritized requests must then be weighted using user constraints and combined to form a single acceleration request that is sent to the movement model. In order to make the constraints intuitive, we implement them as a series of probability (or weighted strength) values. The user merely has to decide how probable (or strong) each of the behaviors is to be when it is combined. Each of these probabilities is defined as a value between 0 and 1. We use a simple additive process, sending in priority order each weighted behavioral value into an accumulator. This process continues until the sum of the accumulated magnitudes gets larger than the maximum acceleration value, which is a parameter of each node. This ensures that the more pressing acceleration requests are immediately satisfied.

3.3. Movement Sub-model

The movement sub-model is the simplest of the three sub-models. At each time step, it uses each node's current velocity and position, as well as the acceleration request sent by the behavioral model, to calculate the new velocity and position for each of the nodes. It then updates the necessary geometric data structures that fully represent the state of the world. After each time step, the state of the world is logged in order to create the mobility pattern trace that will be used for simulation of protocols. Each of the entries in the log includes a timestamp, a node number, and the position (and potentially, orientation) of that node.

4. MOBILITY METRIC AND COMMUNICATION MODEL

A mobility metric is the approximation of how difficult a particular mobility pattern would be to any given network routing protocol. The mobility metric has been shown to be proportional to the amount of movement of nodes relative to surrounding nodes, which might break and change links [7]. In the random mobility model, the metric exhibits a strong linear relation to number of link changes. In our work, we use average number of link changes to evaluate each mobility scenario. In order to count average number of link changes, we count the number of changes in a connectivity matrix that is calculated by Dijkstra's algorithm. This connectivity matrix is updated at every time step and calculated from the point of view of each node. We will show that due to more complex interactions (for example, group centering tendencies) between nodes in our model, this simple metric is insufficient as a predictor of difficulty.

In evaluating the proposed mobility model, we chose to implement a simple and invariant communication model. As implemented, all traffic sources are constant bit rate (CBR) sources. In all our simulations, we use 30 sources with packet size fixed at 256 bytes.

5. IMPLEMENTATION

The aim of our research was to implement and validate our novel method of generating mobility patterns. We had no desire to implement a new simulation environment. As such, we utilized a modified version of Network Simulator (NS), a discrete event simulator targeted at networking research, developed and maintained at the University of California at Berkeley [13]. NS provides substantial support for simulation of TCP, routing, and multicast protocols. NS was an ideal testing platform because it was designed so that scenario files could be generated independently of the simulation environment. We use our algorithms, written mainly in C/C++ to work on UNIX/LINUX platforms, to generate these scenario files, as well as to analyze them to assign each scenario a mobility metric. We then run each of the scenarios with the DSR protocol using NS and the CMU wireless extension package to NS [3]. It should be noted that the generation of the scenarios is applied as a preprocessing stage and does not add any overhead to the actual simulation.

5.1. Scenario File Generation

The user has to provide the system with information about the mobility pattern to be generated. Because we want users to be able to easily create scenarios with somewhat determinate characteristics, we considered the parameters that a user might use to specify a desired scenario. Although the user might conceive of the scenario, the input parameters only guide the system to generating the correct conditions. Because of the nature of the probabilities, emergent behavior (within user constraints) is intrinsically a part of the system. The parameters that the user must specify are: size of the environment, number of nodes, maximum velocity and acceleration of nodes, cruise distance (the ideal distance each node will attempt to stay from neighboring nodes), and the maximum range of radio waves. In addition, the user must specify the four probabilities that govern the behavioral model (for group centering, collision avoidance, velocity matching, and inertia).

As an example, we define three scenarios: Art Gallery, Event Hall, and Battlefield (see Table 1). In the Art Gallery, movement is close to random movement (people are wandering around quite aimlessly). In the Event Hall, we model a strong tendency to stay within the group and to avoid collisions (as individuals latch on to roving groups of viewers). In the Battlefield, the emphasis is on group centering and inertia (as soldiers keep advancing as a tight coordinated group).

Table 1. Example parameters used in simple scenarios.

Parameters	**Art Gallery**	**Event Hall**	**Battle Field**
Field Size [m]	X=50 Y=30	X=150 Y=90	X=1500 Y=1500
Number of nodes	30	50	50
Max velocity [m/s]	0.2	1.0	4.0
Max acceleration [m/s^2]	0.1	0.5	2.0
Cruise distance [m]	3	5	50
Comm. range [m]	10	25	200
Group centering	0.1	0.7	0.7
Collision avoidance	0.4	0.7	0.1
Velocity matching	0.4	0.7	0.1
Inertia	0.1	0.4	0.7

Recall that the probability parameter is used to indicate how probable each of the nodes in the group would be influenced by each individual behavior as well as the weight on the acceleration produced on affected nodes. For example, a 0.1 "group centering" probability means that each node has a 10 percent probability of being affected, or alternatively, that 10 percent of all nodes will be affected by this behavior. Each of these nodes will be affected by a weight of 0.1 on the final acceleration.

The internal state of each node is implemented as an object. Each instance of these objects has a computation process to apply the behavioral model to the internal data. Movement information is generated at small, but discrete, time steps to approximate continuous movement. Since the movement model interpolates between these discrete points, we eventually have a model of continuous movement. Information passed to the simulator includes node identification, time stamp, as well as new node position.

In addition to the movement information, the scenario file includes 'God' information, which encapsulates the optimal routes and patterns to be later used for calculating routing overhead as well as efficiency. The 'God' code is based on an algorithm from Carnegie Mellon University's wireless extension package. This information is generated at each time step using Dijkstra's algorithm to calculate the shortest paths between packet senders and receivers based on the position of each node.

Scenario files were visualized using the Ad-hockey visualization tool (part of the wireless extension package). Subjective observation and evaluation showed that the scenarios generated did indeed represent what seemed to be

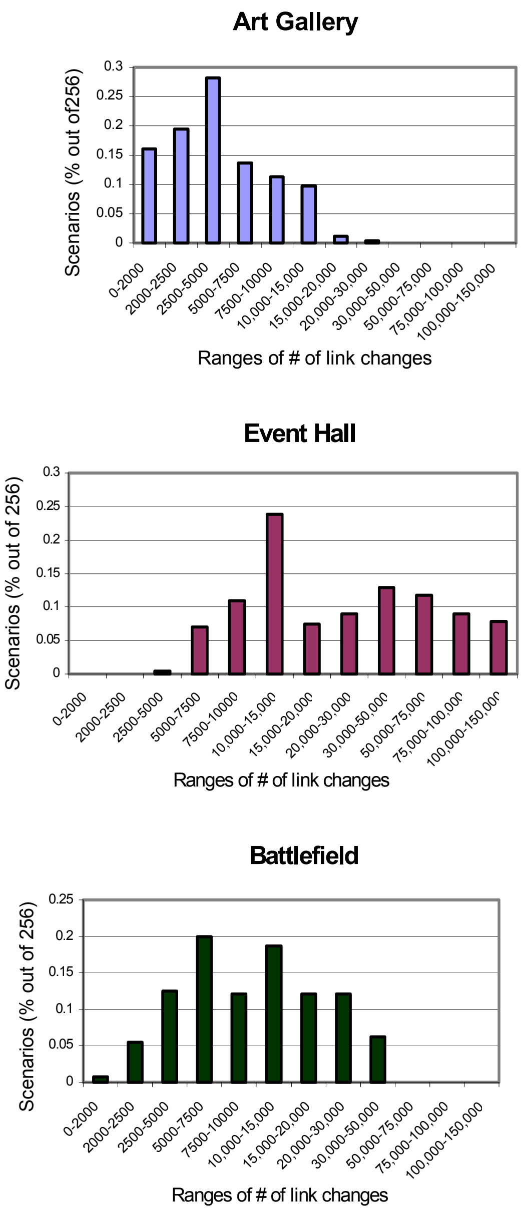


Figure 1. Distribution of number of scenario files according to number of topology changes during 900-second simulation.

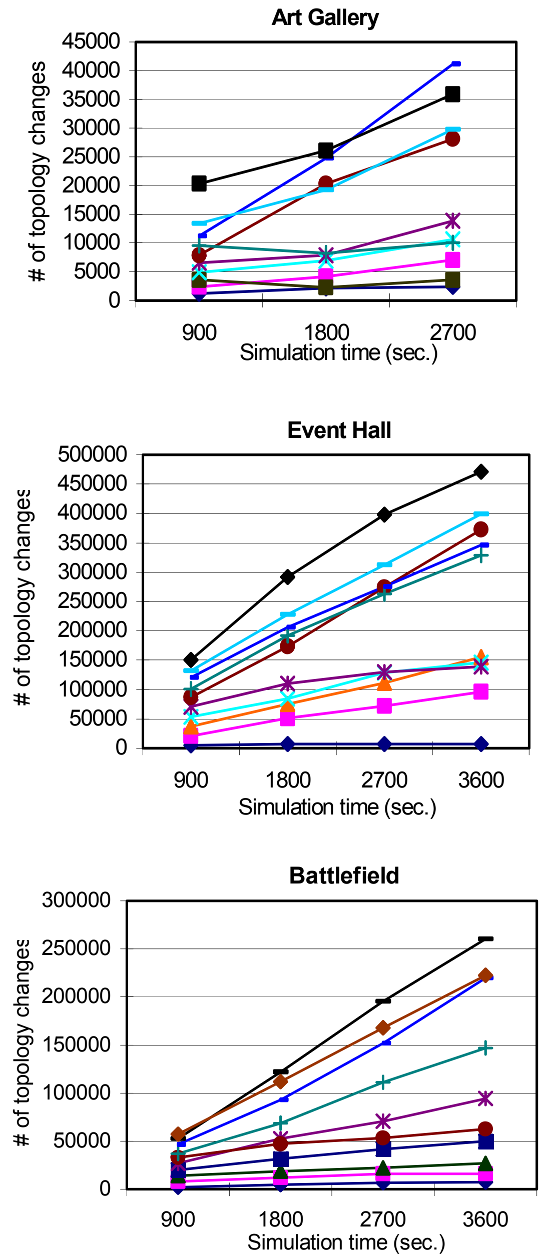


Figure 2. Linear relation between simulation time and number of topology change.

realistic and reasonable human mobility patterns. We present more rigorous results in the next section.

6. EVALUATION

Objective evaluation of a system like this is difficult, if not impossible. We know of no existing work done to objectively evaluate mobility models. Nevertheless, we present certain measures that we use to validate the utility of using our method for generating mobility patterns. First, we show that the patterns generated are reproducible as well as stable (ie. behavior does not fluctuate significantly with small change in parameters). Second, we use a representative set of mobility patterns to measure a particular routing protocol's performance. For this purpose, we chose the Dynamic Source Routing (DSR) protocol, which is claimed to be a high performance protocol that is relatively impervious to small fluctuations in the scenario [2]. We show that each of these mobility patterns does indeed challenge the protocol in different respects and produces very different performance as a result.

6.1. Stability of Mobility Patterns

A potential concern when generating mobility patterns is that the system would be insensitive to input parameters, making it difficult to produce a wide variety of scenarios. Another concern is that the system would be overly sensitive to small changes. Such a system would cause fluctuations in the mobility patterns that would make it difficult for the user to specify the parameters and generate the desired scenarios. Here, we dispel these concern and show that the user can dependably produce a wide variety of stable mobility patterns. Also, since the algorithms used are deterministic, mobility patterns are completely specified by the input parameters and are easily reproducible.

In order to gain a statistical understanding of the model's behavior and to prove stability, we used the above three scenarios (Art Gallery, Event Hall, and Battlefield) to generate 256 mobility pattern files in each case with a simulation time of 900 seconds. Each of these 256 files was generated with a unique permutation of probabilities from the set [0.0, 0.4, 0.7, 1.0] for group centering, collision avoidance, velocity matching, and inertia. Given that the system behavior is a result of complex interactions driven by the input parameters, we chose to interpret the output, or scenario files, rather than to isolate the effect of each parameter. With the mobility metric, we make estimations of its effects on the protocol and simulate to verify that.

We first sorted these files for each scenario by number of topology changes. The order of these changes could be grouped into several subsets, each with several similar parameters (see Figure 1). As evident from the graph, there is a wide range of difficulty levels for each scenario (due to the interaction of the four different behavioral input probabilities). This suggests that the user may fix the basic scenario conditions and only vary the four behavioral probabilities to produce mobility patterns of various difficulties.

We then randomly chose 10 out of the 256 files so that the topology changes in these 10 files were equally distributed over the range of the overall changes without considering the input parameters. With each of the corresponding input parameters from these 10 files, we generated additional files with simulation times of 1800, 2700, and 3600 seconds. We confirmed the linear relation between simulation time and the number of topology changes, implying that the simulations were stable by 900 seconds (see Figure 2). We used these 10 scenario files from each condition (900 seconds) in our DSR simulations.

6.2. Performance Results with DSR

We ran each of the 10 mobility patterns from each of the three different scenarios through NS with the DSR protocol. Two metrics, data packet delivery ratio and routing overhead, were used to evaluate DSR performance with these mobility patterns. Data packet delivery ratio is defined as the ratio between the number of packets received by CBR sinks and the number of packets actually sent by the CBR sources. Routing overhead is calculated to be the total number of routing packets transmitted during the simulation. For packets sent over multiple hops, each hop counts as one transmission.

Figures 3 and 4 show the results in the three scenarios. The interesting thing about these results is the very different performance that DSR exhibits both between the scenarios but also within them. As can be seen in the case of the Battlefield scenario, link/topology change is not an ideal predictor of routing difficulty. In fact, the data packet delivery ratio, in this case, grows with number of link changes. This unintuitive correlation is a result of other factors such as segmentation of the propagation environment and group centering tendencies in the scenario affecting the performance of DSR more than does the number of link changes. This large variety of performance results shows that our mobility model can and does generate very different and interesting types of mobility patterns that can be used to intensively evaluate network routing protocols.

The fact that the three scenarios had influenced the routing overhead and packet delivery ratio in different ways, i.e., the trend-lines are different, suggests that our mobility model has some potentially interesting metrics beyond link change rate and relative mobility. Although we have not analyzed these metrics from the experiments so far, we would like to explore them in our future work.

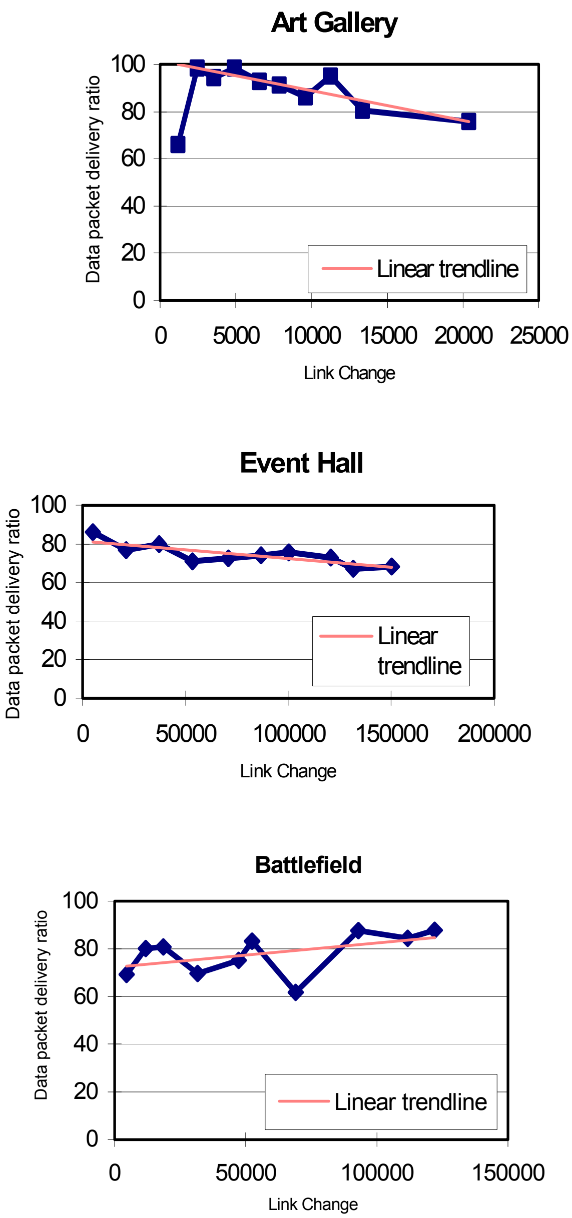


Figure 3. Data packet delivery ratio (%) vs link change.

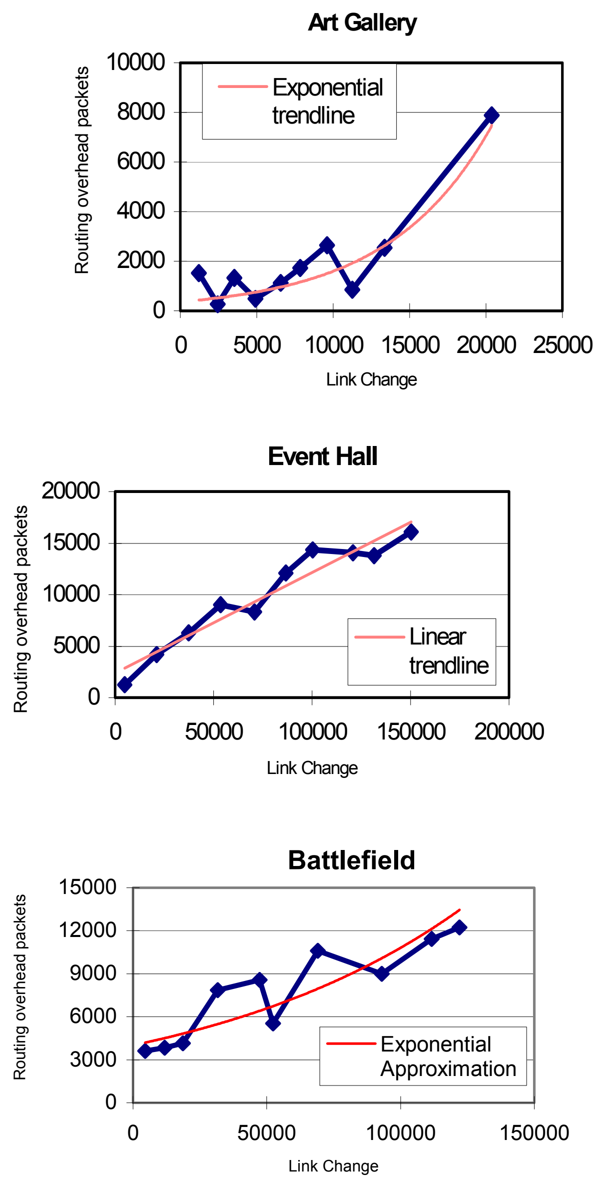


Figure 4. Routing overhead vs link change.

7. FUTURE WORK

Currently, behaviors such as collision avoidance and velocity matching are focused on those that lead to simulated grouping. We would like to consider more constraints and behaviors. An example is the behavior of car alignment on a highway, in which there exists two groups of nodes moving in opposite directions on approximately parallel paths. Given this extension, we could provide the user not only the ability to change probabilities of each behavior, but also the flexibility to choose the subset of behaviors they need.

The scenarios generated so far were based entirely on homogeneous behaviors. In each scenario, every node moved according to the same set of behaviors with the same probabilities. We would like to use heterogeneous behaviors to generate scenarios. This would allow for the creation of scenarios in which multiple 'beings' exist. In such scenarios, we could generate realistic situations in which relatively quick vehicular models are combined with slower moving human models.

We would also like to extend the model to deal with environmental obstacles not only because they influence the movement behavior of nodes, but also because they block radio waves.

Our current communication model is continuous bit rate traffic since our primary focus is on generating mobility patterns. We would like to design a system of communication traffic patterns that are closely correlated to the mobility patterns. For example, when a node is trying to avoid collision, it is unlikely that it would start new connections though it might keep a certain level of current communication traffic going; on the other hand, when it is wandering, it is likely to begin new tasks and generate large amount of traffic.

8. CONCLUSION

In this paper, we have presented a novel method of generating reproducible and realistic mobility patterns for use in the simulation and evaluation of ad hoc networking protocols. In our individually simulated behavioral model, we simulate the behaviors and movement of individual nodes and their relationship to surrounding nodes in order to generate the overall mobility pattern. We propose this system as a general-purpose framework that may be used to reliably reproduce existing mobility patterns as well as to easily generate useful new ones. In order to validate the model, we have presented statistical characteristics of a few mobility patterns generated by our mobility model. We have shown that different mobility patterns generated in this way stress the Dynamic Source Routing (DSR) protocol in different ways and cause it to perform very differently in each case. We believe that the results shown warrant further investigation into the generation of mobility, as well as communication patterns using this method.

ACKNOWLEDGEMENTS

We would like to thank David Johnson, now at Rice University, for his advice and guidance during the early stages of this project.

REFERENCES

[1] Basagni, S., Chlamtac, I., Syrotiuk, V.R. and Woodward, B.A. 1998. A Distance Routing Effect Algorithm for Mobility (DREAM). *In ACM/IEEE MobiCom*, 76-84.

[2] Broch, J., Maltz, D.A., Johnson, D.B., Hu, Y. and Jetcheva, J. 1998. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. *In ACM/IEEE MobiCom*, 25-30.

[3] The CMU Wireless and mobility Extensions to ns: http://www.monarch.cs.cmu.edu/cmu-ns.html

[4] Haas, Z.J. 1997. A New Routing Protocol for Reconfigurable Wireless Networks. *In Proceedings of IEEE 6th International Conference in Universal Personal Communications*, 562-566.

[5] Hong, X., Gerla, M., Pei, G. and Chiang, C. 1999. A Group Mobility Model for Ad Hoc Wireless Networks. *Proceedings of the 2nd ACM International Workshop on Modeling, Analysis, and Simulation of Wireless and Mobile Systems*. August 20, 53-60.

[6] Internet Draft: http://ietf.org/internet-drafts/draft-ietf-manet-zone-zrp-02.txt

[7] Johansson, P., Larsson, T., Hedman, N., Mielczarek, B. and Degermark, M. 1999. Scenario-based performance analysis of routing protocols for mobile ad-hoc networks. *In ACM/IEEE MobiCom*, 195-206.

[8] McDonald, B. and Znati, T. 1999. A Mobility-Based Framework for Adaptive Clustering in Wireless Ad-Hoc Networks. *In IEEE Journal on Selected Areas in Communication*, Vol. 17, No. 8, August 1999 - Special Issue on Wireless Ad-Hoc Networks.

[9] Noble, B.D, Satyanarayanan, M., Nguyen, G.T. and Katz, R.H. 1997. Trace-Based Mobile Network Emulation. *In ACM SigComm Conference*, Cannes, France.

[10] Reeves. W.T. 1983. Particle Systems – A Technique for Modeling a Class of Fuzzy Objects. *In ACM Transactions on Graphics* V2#2.

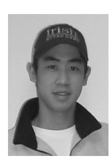
[11] Reynolds, C.W. 1987. Flocks, Herds, and Schools: A Distributed Behavioral Model. *In Siggraph '87* Conference Proceedings, 21(4), 25-34.

[12] Sanchez, M. Mobility Models: http://www.disca.upv.es/misan/mobmodel.htm

[13] UCB/LBNL/VINT Network Simulator - ns (version 2): http://www-mash.CS.Berkeley.EDU/ns/

[14] Zonoozi, M.M. and Dassanayake, P. 1997. User Mobility Modeling and Characterization of Mobility Patterns. *In IEEE Journal on Selected Areas in Communications*, 15(7): 1239-1252.

Author Biography



Desney S Tan is a PhD student working with Dr Randy Pausch on designing systems and human computer interfaces that augment human cognition, specifically human memory. For more information, see http://www.cs.cmu.edu/~desney.