CAD-HOC: A CAD Like Tool For Generating Mobility Benchmarks In Ad-Hoc Networks

Subodh Shah†, Edwin Hernandez and Abdelsalam (Sumi) Helal
Computer and Information Science and Engineering Department
University of Florida
Gainesville, FL 32611-6120, USA
helal@cise.ufl.edu

ABSTRACT

This paper addresses the problem of using unrealistic mobility scenarios for simulation of ad-hoc routing protocols. We present CAD-HOC, a cousin tool to Network Simulator (*ns*), which allows ad-hoc networking experimentation to be performed under visually realistic scenarios such as an airport or a bus terminal, buildings, highways and other facilities. CAD-HOC focuses on capturing the visual scenario and transforming it into mobility and connection benchmarks, which are subsequently fed to *ns* to drive simulation experiments. Generation of mobility and connection benchmarks is facilitated by allowing the users to choose from Brownian, Column, Pursue, and Nomadic models, or to specify user-defined movements. The tool also provides on-line analyses on ad-hoc mobility, ad-hoc connectivity and message complexity, which give an estimate of the overall *ns* simulation complexity, as the mobility scenario is built. Comparative simulation based on CAD-HOC generated scenarios raise several questions about the reliability of ns simulation results of ad-hoc routing protocols based on random, non-realistic scenarios.

KEYWORDS

Mobility scenario generator, mobility benchmarks, ns-simulator, Ad-hoc networks, Ad-hoc routing.

1. INTRODUCTION

Recently, mobile ad-hoc networks have been one of the focus areas of research and standard development. Ad-hoc networks do not require a pre-existing communication infrastructure, and are particularly well suited to applications such as battlefield management, law enforcements and disaster recovery to mention a few. Literature on ad-hoc networks shows that mobile nodes move according to various patterns, often exhibiting group mobility behavior. It also shows that different mobility patterns could affect the overall connectivity of the ad-hoc system in significant ways.

Routing in ad-hoc networks is an emerging area of research that is vital to the operability of such networks. Obviously, routing performance depends heavily on the mobility behavior in a particular scenario. Network Simulator (*ns*) [24] has gradually become the most widely used tool to simulate ad-hoc network environments, especially, ad-hoc routing protocols. Although the capabilities found in *ns* are more than appropriate, the use of random mobility patterns and connection scenarios in setting up experiments is unrealistic and quite limiting. One alternative is for the researcher to use manual *tcl* scripting to generate specific scenarios, which is very tedious and time consuming. The difficulty to generate such scenarios increases with the size and complexity of the scenario. Another alternative is to develop and then use mathematical mobility models that capture the target

† For download and source code questions about CAD-HOC, contact Subodh Shah at cadhoc2001@yahoo.com

application. Such models, however better they are when compared to purely random scenarios, are inflexible to change and are difficult to develop for complex applications.

In this paper, we introduce CAD-HOC, a powerful tool that addresses the aforementioned limitations and that revolutionize the specification of movement and connection scenarios for simulation in *ns*. CAD-HOC is similar in spirit to CAD/CAM tools, in using an extensive GUI to aid the *ns* user to generate realistic mobility scenarios and benchmarks. It provides a great deal of realism in producing mobility scenarios, by allowing users to base their design on particular spatial layouts, which can be produced by using any image that represents a real-life situation. These layouts could be an airport terminal layout, a highway map, or a layout of a section in a shopping mall. CAD-HOC provides immense flexibility to *ns* users by allowing them to specify movement constraints (e.g. cant not move off the highway), movement parameters (e.g. speed, renewal process), and connection modes. CAD-HOC also provides a high degree of productivity by allowing group operations, in which several or tens of nodes and their movement behavior can be specified in one single command. CAD-HOC generated benchmarks can be saved, edited, and ultimately fed to ns to drive fully controlled simulation experiments. One of the tool's valuable assets is an on-line estimation analysis of the characteristics of the scenario being generated. This includes a quantification of the degree of ad-hoc mobility, the average achieved ad-hoc connectivity, and the *ns* message complexity (and hence simulation cost). Such analyses are believed to will aid researchers to mold the generation of mobility scenario and benchmarks to the desired level of complexity.

By semi-automating the specification and generation of non-trivial and realistic mobility benchmarks, CAD-HOC will enable researchers to dimension and reliably measure the performance of various ad-hoc routing protocols (existing and newly proposed) under various real-life scenarios.

1.1. Related work

Analysis and simulation of ad-hoc network protocols has been investigated using pseudo random mobility patterns. Literature shows that quite a bit of work has been done on mobility behavior in ad-hoc networks. Holland and Vaidya [5] observed that the TCP throughput behavior in ad-hoc networks using mobility patterns, profiles of mobile units at different speeds. Notwithstanding their results lacked of a realistic scenario to match the mobility patterns. Hong and Gerla [25] concluded that purely random mobility models are not enough to characterize ad-hoc networks, additionally mentioning that these models are application dependent. Group mobility models and the Reference Point Group Mobility (RPGM) have the mobile hosts organized by groups according to their logical relationships.

The group mobility models are extended in the CAD-HOC tool by constraining the movement of mobile nodes to a certain area defined by the user. In addition to RPGM, scenario-based analysis can be found in [12], where the mobility metric is introduced in the analysis of three realistic environments. The metric consists of a geometric average of the speed of each node in relation to other nodes measured. Johansson made use of three mobility scenarios: a) Conference, with low mobility; b) Event coverage, with fairly high mobility; c) Disaster area, with high mobility. Our tool can map any of these scenarios to a geographical picture where the user could incorporate nodes or groups of nodes constrained movement into a shape, a "beam" between two shapes, and Brownian movement around a whole area. Similar to Johansson, in CAD-HOC, the user can manage the parameters of transmission range, simulation time, number of nodes, and environment size. As opposed to Johansson, CAD-HOC provides a visual tool that not only generates the *ns* script but also subsequently modify the simulation scenario without touching the scripts required by the network simulator.

CAD-HOC provides the user with the message complexity graph, which represents the data traffic complexity of the UDP and TCP communication links among the mobile nodes. Moreover, CAD-HOC allows the user to use within an area, the mobility models specified in Sanchez, Hu and Johnson [3],[28]. These models are: Brownian motion, Column motion, Nomadic motion and Pursue motion. These additional features provide richer ad-hoc mobility scenarios easily drawn, simulated, and changed. Ad-hoc protocols such as DSR ([10], [11], [13]), DSDV ([10], [11], [20]), AODV ([10], [20], [26]), and any implementation ported to *ns* can be tested in similar conditions.

CAD-HOC can be used together with trace modulation [21], [22]. The latter approaches the problem of developing realistic models, which can be reproduced and validated. CAD-HOC can generate its own traces of network characteristics, which can easily drive the emulation of a target network. Mobility and connection patterns could be assigned to imaginary small vehicles attached to smart phones, executing the emulation in a LAN based upon the results and values provided by the CAD-HOC tool and the *ns* traces of data traffic and latency values.

1.2. Conducting Simulations In Network Simulator (*ns*)

Figure 1 shows the normal process of conducting simulation in *ns*. The user has to imagine of a scenario, the number of nodes to be placed in the scenario, and then write the TCL scripts (*.tcl* file) specifying the node configurations parameters and some other *ns* commands required to start and stop *ns*. The user has also to create the movement and connection files that together represent the scenario. These files can be generated using the *cbrgen* or *setdest* utilities, or the user can write TCL scripts to generate the movement and connection files (which is very tedious). Once these files are produced they are fed as input to *ns*, using which *ns* creates the scenario during simulation. The output of the simulation is a trace file (.tr), which is logged with each and every event that took place during the simulation. This file can than be used for obtaining measures such as mobility, throughput, end-to-end delay, and packet loss measurement. An optional output is the NAM supported file (.nam) that logs the necessary events to help visualize the scenario using the NAM. The NAM is a post simulation process that shows how the nodes moved and how they were connected during the simulation.

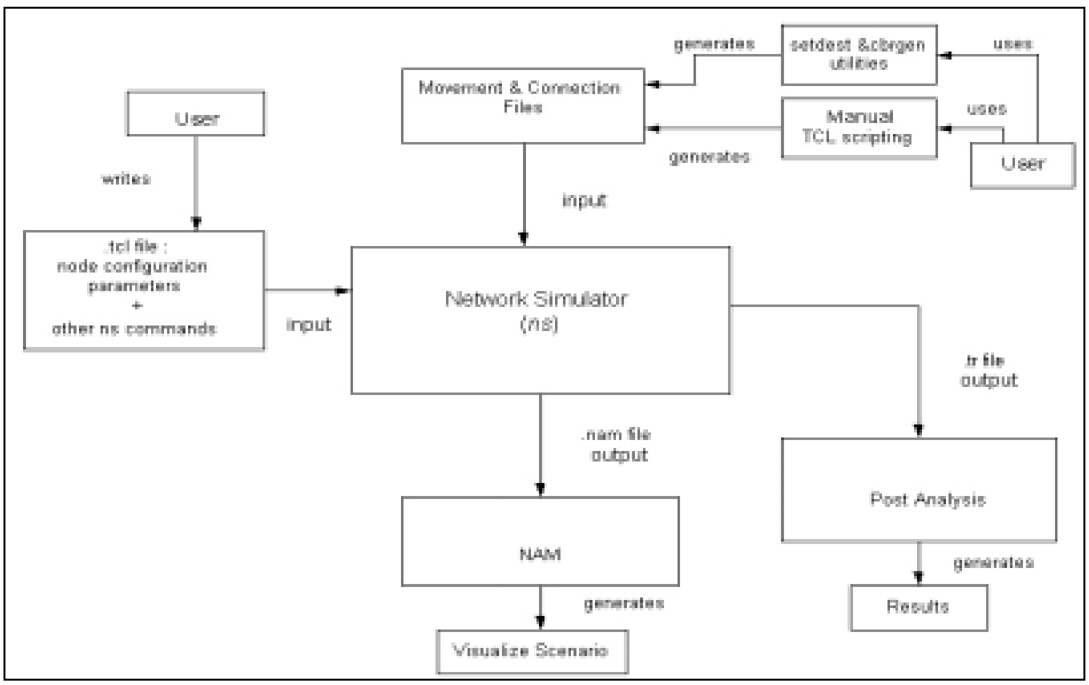


Figure 1 – The diagrammatic representation of conducting simulation in ns

1.3. The Problem

Unfortunately, *ns* lacks the capability to specify or capture realistic and complex mobility scenarios. Currently, researchers can use the *setdest* and *cbrgen* utilities, which allow for only simple, randomly generated movement and connection specifications. Designing real-life scenarios using these utilities is a complex and tiresome process. An alternative is to use mathematical mobility models to create scenarios, but this approach lacks flexibility to change and is difficult to use for complex mobility scenarios. For example, most mobility models do not support curvature motion or mobility-restricted motion. To better study and understand the behavior of ad-hoc routing protocols, there is a need for a powerful tool that can automate or semi-automate the specification and generation of realistic mobility scenarios.

2. THE CAD-HOC TOOL

The goal of the CAD-HOC tool is to generate mobility scenarios and benchmarks capturing them into *ns* movement and connection files. Figure 2 depicts the look and feel of CAD-HOC. The main components of the interface are the main menu, diagram canvas (area where the scenario is prepared), status panel (for displaying error and usage messages), scenario characterization gauges and graphs. The menu bar contains features, which allow the user to create mobility supported and restricted areas. It also contains features to resize the mobility-supported areas, create mobile nodes, specify movements, specify connections, produce movement and connection code files, and continuously (on-line) estimate the degree of ad-hoc mobility, degree of ad-hoc connectivity and the expected *ns* simulation complexity. The mobility-supported areas include squares, triangles, hexagons, lines, beams, and moveable areas (free shape areas).

The ad-hoc mobility gauge estimates the average relative speed of the mobile nodes. This gauge measures the mobility dynamism of the ad-hoc network and not the absolute speeds of its nodes. For example, a group of tightly grouped mobile nodes moving at the same high velocity exhibit low average relative speed. The ad-hoc connectivity gauge depicts the largest percentage of nodes that are "connected" at a particular instance of time. Connectivity here refers to being within transmission range of other nodes. The rationale behind this estimate is to allow the researcher to control the scenario design to the desired level of connectivity. For example, a new ad-hoc routing protocol may at a first glance seem to outperform others; by examining the performance of the protocol under low and high degrees of connectivity different conclusions may be reached. The message complexity estimates the volume of traffic that will occur during *ns* simulation. This analysis gives the user an instantaneous indicator of the simulation cost of the scenario he/she is building.

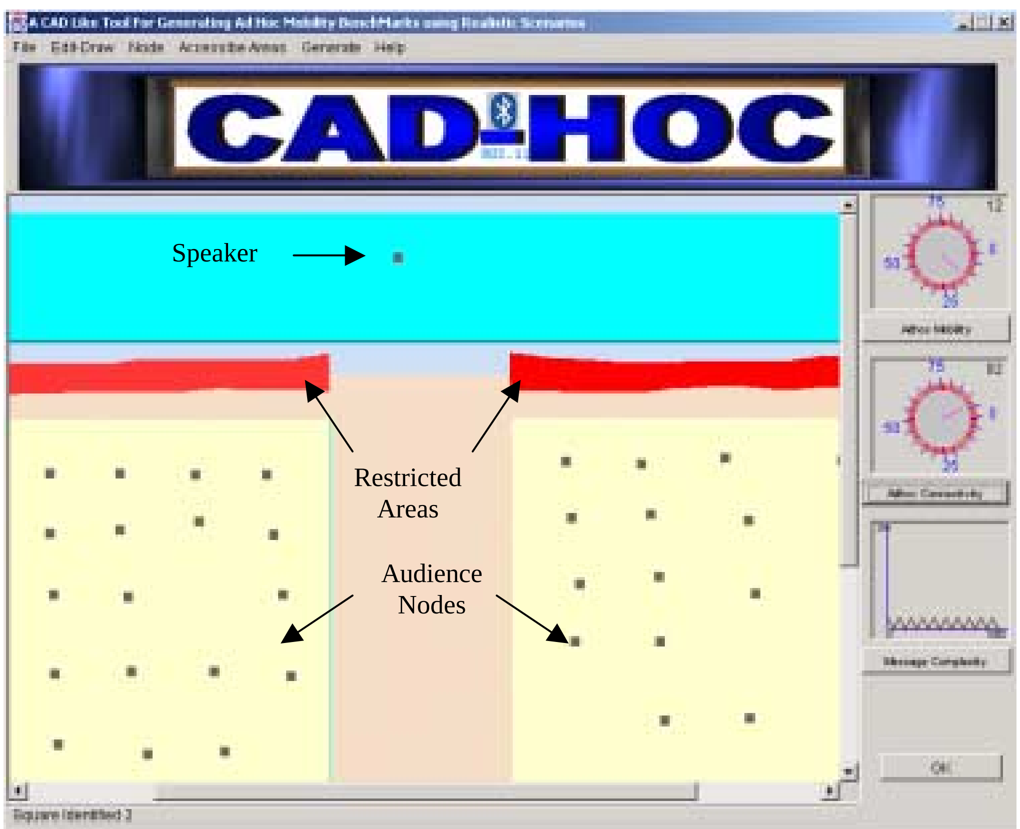


Figure 2 – The main screen of CAD-HOC

Movement (mobility) can be specified using either mobility models such as Brownian, Column, Pursue and Nomadic or as user-defined movements (curvature motion).

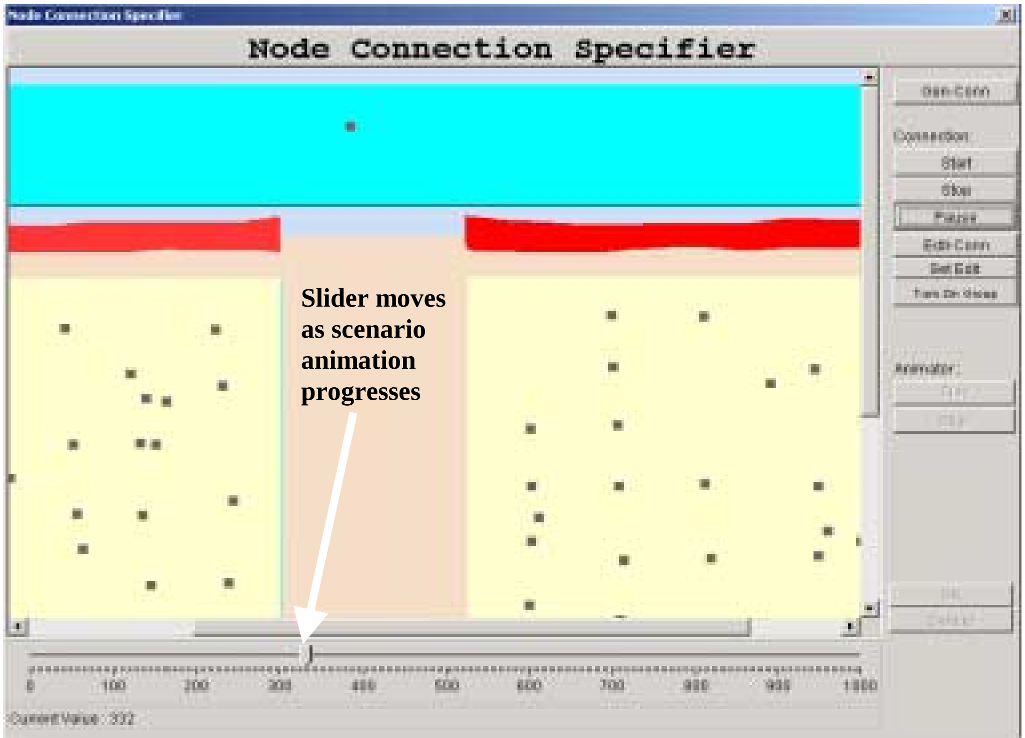


Figure 3 –Connection specifier interface of CAD-HOC

Figure 3 shows the interface where the connection can be specified. This interface is timer based. Once the start button is pressed the timer starts and the nodes are seen to be moving as the movement was specified. This helps to visualize the scenario before committing to generating the benchmark for *ns* simulation. This feature is similar to NAM but has the advantage over NAM of being a pre-simulation (cheaper) visualization tool. The user can specify/modify a connection by pressing the pause button and then click on one node making it the source node, and then click on another making it the destination node; a dialog box will popup, which will ask the user to enter a few parameters and thus specify a connection record between the selected source and destination nodes. This feature is important as it gives an idea of where the source and destination nodes will be while the communication takes place, which help yielding a realistic effect.

2.1. CAD-HOC Architecture

The architecture of CAD-HOC is very simple and easy to understand. It is open-ended to allow for insertion of additional components and functionality by other researchers in this community. Figure 4 shows the overview of the architecture and its components. Researchers use the tool by first laying out the mobility supported and mobility restricted areas. Optionally, an appropriate image can be loaded into the tool prior to specifying the layout. CAD-HOC comes with a library of images, but any *jpg* or *gif* image can be loaded into the tool.

The user does not always have to start from scratch, and can instead load a preexisting scenario for further editing. This allows the user to create new scenarios out of pre-existing ones. The scenario generator engine allows the creation of mobile nodes at desired locations. Users then use the movement specification engine to produce mobility scenarios using supported models or using user-defined birth/death mobility specification. Like movement specification, the connection specification engine is used to parameterize the communication details between any two nodes (TCP and UDP traffic).

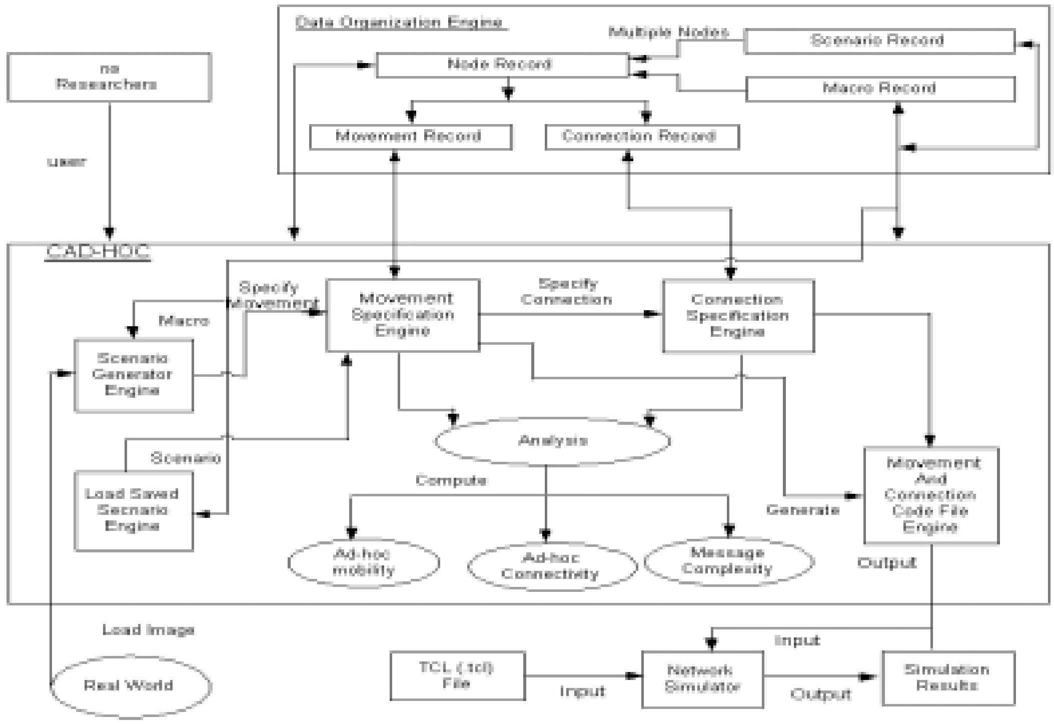


Figure 4 –Overview of CAD-HOC architecture

The scenario generator or load scenario engines in conjunction with movement and connection specification engines collectively set up and visualize a scenario in preparation for experimentation with ad-hoc routing protocols. The end result is benchmark files in a format compatible with ns input files. A data organization engine maintains node records as well as records for all events occurring during the scenario generation. These records are used to generate the benchmark files in addition to being continuously used by the on-line analysis engine. Data organization is discussed in detail in the next section.

2.2. Data Organization

Figure 5 depicts the detailed description of data organization in CAD-HOC. Each node has a record with the parameters like index, color, (x, y, z) coordinates and transmission range. Each node also maintains vectors of movement records and connection records. Each movement record has parameters like start position, end position, speed, start and end time.

Each connection record stores the index of node to which the connection is made, speed of the connection, maximum number of connections allowed, packet size, rate, interval, type of traffic, type of application and the time when the connection should start in the simulation. The mobility supported areas like square, triangle, hexagon, beam and moveable areas store the index, shape parameters, list of nodes added to that areas, the type of movement model to be used for generating mobility. The Macro and Scenario records shown in Figure 5 are actually files created that store all the required details to reproduce the macro or scenario again, these files include information about environment like image loaded, nodes, the mobility restricted and supported areas, and node details.

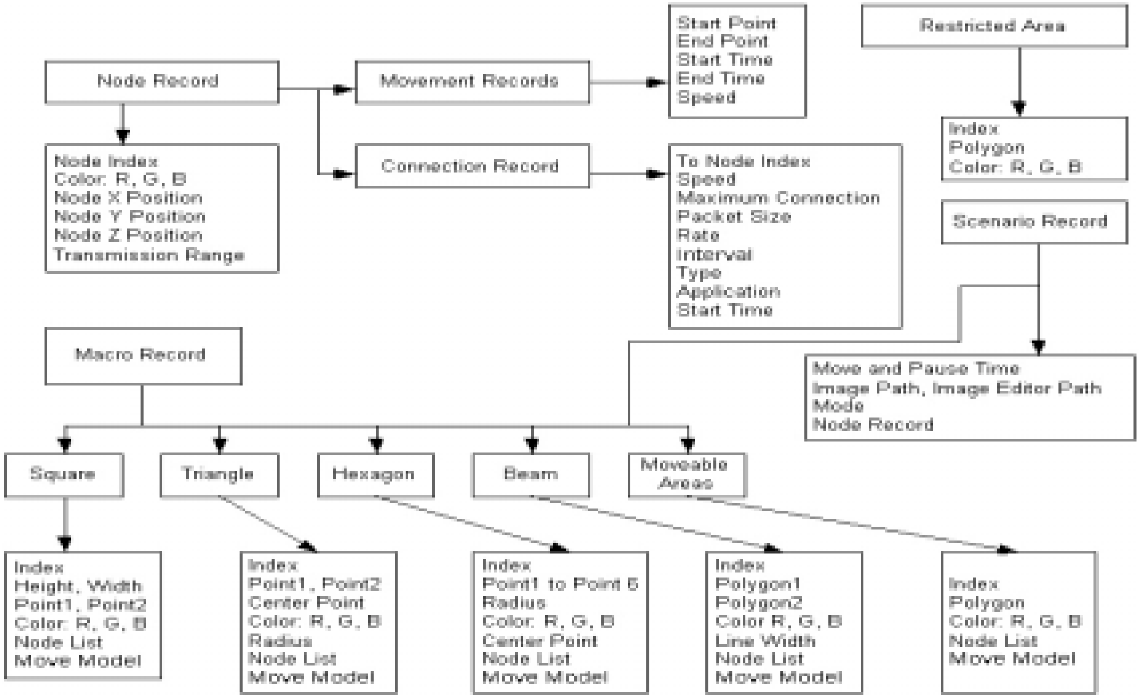


Figure 5 –Overview of data organization in CAD-HOC

2.3. Performance Metrics

Mobility metric represents ad-hoc mobility, ad-hoc connectivity metric represents ad-hoc mobility and message complexity metric represents message complexity.

Mobility Metric

Mobility metric captures and quantifies the kind of node motion relevant to an ad-hoc routing protocol. The mobility metric used follow the geometric mobility model proposed by Johansson et al [11] in which the speed of a node is measured in relation to the speed of other nodes. We include their model in this paper (Appendix A) only as a service to the reader. Full credit goes to Johansson et al.

Ad-hoc Connectivity Metric

Ad-hoc connectivity metric is a measure of average connectivity of the nodes during the simulation. The Ad-hoc connectivity metric is based upon *setdest* algorithm of *ns* to compute the total number of routes possible between the nodes, and the destinations unreachable. The shortest path Dijkstra algorithm is also used in calculating ad-hoc connectivity in the tool. The number of nodes reachable during the simulation is computed by subtracting the destination unreachable from the total possible connections.

Message Complexity Metric

Message complexity throughout the course of simulation depends upon the TCP/UDP traffic as well as the reach ability and movement patterns, which is very difficult to determine as an average. The message complexity along with ad-hoc mobility and connectivity give a better estimate of the total simulation complexity. We propose a message complexity metric, which uses the node connection data recorded during the specification of connection between nodes.

2.4. Prime Features Implemented

Currently implemented features of CAD-HOC are summarized below:

- Load any JPG or GIF image

- Load previously created and saved scenarios
- Save a loaded scenario after modifications
- Unload the image loaded in the design canvas.
- Set any image editor path and then open that image editor from the CAD-HOC main window to modify the image
- Set CAD-HOC in ad-hoc or mobile IP mode. Mobile IP is not yet supported though functionalities for it will be added later.
- Set the movement and pause time to be used to produce movement patterns.
- Place beams that can be used to form a track or a road.
- Place moveable areas allowing creation of mobility supported areas to overlap of any arbitrary shapes on image.
- Place restricted areas that are mobility restricted and are very important to achieve realistic scenario.
- Add nodes to scenarios; there are three ways to do that: (1) add a node randomly, (2) add a node at a specific location or (3) add multiple nodes in a selected area (group operator).
- Delete nodes by selecting nodes and then selecting the delete option.
- An interface that allows to set color and transmission range for each node
- View/Edit/Delete node movements in a table like interface, by clicking on a particular node and then select the appropriate option.
- Place mobility supported areas like square, hexagons, triangles, and lines, by selecting the appropriate menu options.
- Create and save a scenario section (square, triangle, hexagon, or beam moveable area) as macro on right click. Also, insert macros loading previously saved ones.
- Moving, resizing or deleting of mobility supported and restricted areas such as square, hexagon, triangle, beam, and moveable area.
- An interface to specify connection and play a visualization of the scenario before actual simulation in *n*s. It also allows replaying after mobility or connection modifications (trying what-if scenarios).
- Alternative ways to specify connections using group connection operator or by selecting source and destination nodes.
- Change color of nodes and areas
- Produce movement and connection code files to be fed to *ns* for simulation.
- Pre-simulation analysis of the generated scenario by measuring ad-hoc mobility, ad-hoc connectivity, and message complexity.
- Java Doc help of the classes and functions

2.5. Experimental Scenarios And Performance Results

CAD-HOC was used to generate three realistic scenarios, where the movement and connection files produced were fed into *ns* for simulation. There were two goals to simulate the scenarios, firstly to show that the movement and connection files thus produced are *ns* compatible, secondly to get an idea of the results obtained by simulation based on this generated scenario as compared to the simulations done using random movements and connection specifications.

The three scenarios are: (1) an Airport terminal with estimated 76% ad-hoc mobility and 93% ad-hoc connectivity, (2) a highway scenario with estimated 73% mobility and 91% ad-hoc connectivity and (3) a conference room scenario with estimated 12% mobility and 92% ad-hoc connectivity. The parameters used in simulation for all the three scenarios are summarized in Table 1 below.

The transmission range was kept as 250.0 m, which is the default transmission range in *ns* and hence the default value in CAD-HOC. A bandwidth of 1 Mbps and a simulation time of 1000 seconds were used and are the default values in the tool. The environment size could be of any dimension with the constraint that it should be at-least the size of maximum x and y direction movement. The packet size, pause time and traffic type was varied.

Table 1: Parameters used during realistic scenario simulation

Parameter Name	Highway	Airport Terminal	Conference
Transmission Radius (m)	250.0	250.0	250.0
Bandwidth (Mbps)	1	1	1
Simulation Time (s)	1000	1000	1000
Number of Nodes	23	63	73
Environment size (m x m)	1300 x 1040	1300 x 1040	1300 x 1040
Traffic Type	CBR	CBR	CBR
Packet Size (byte)	512	512	512
Pause Time (s)	5	5	5

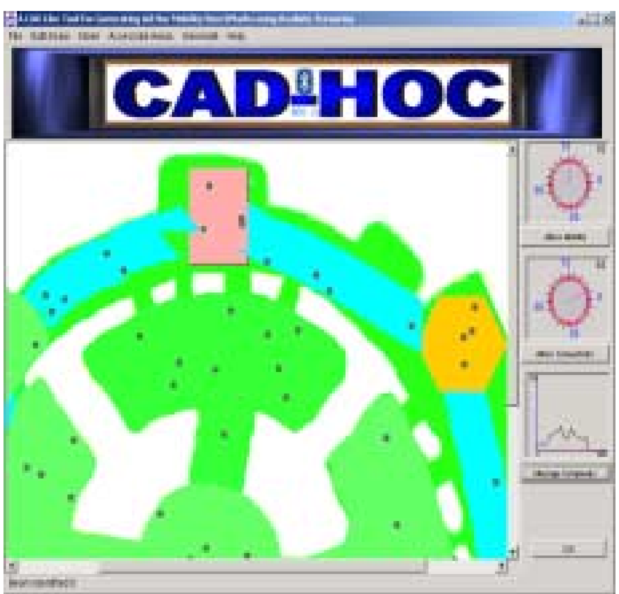


Figure 6 –CAD-HOC generated airport terminal

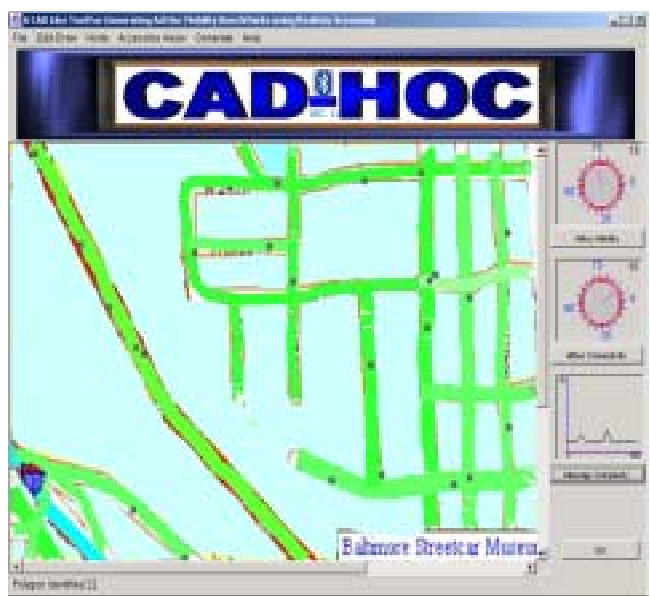


Figure 7 –CAD-HOC generated highway scenario

2.5.1. *Airport Terminal Scenario*

The airport terminal scenario was designed using the Brownian model, to achieve the above stated 76% ad-hoc mobility and 93% ad-hoc connectivity. Figure 6 shows the airport terminal scenario generated using the tool. Dallas (DFW) Airport's image is used to create the base scenario. The scenario is created with the use of two squares, one hexagon, a few moveable areas (that can be of any arbitrary shape) and mobility restricted areas. This scenario had 63 nodes moving around in an airport terminal and had 100 CBR sources. The performance of DSR, AODV and DSDV was measured and compared. The performance results obtained in this airport terminal scenario are shown in the Table 2.

Table 2: Simulation results for airport *terminal scenario*

	DSR	AODV	DSDV
Throughput (bytes/sec)	153839.98	174681.15	155988.86
Delay (ms)	0.016156	0.003682	0.004253
Total Loss (%)	0.78	0.201106	2.9600

The results demonstrate that AODV performs better that DSR and DSDV. It has the lowest loss, minimum delay and highest throughput. This depicts that in this scenario the protocol that is a combination of table-driven and source initiated routing protocol (AODV) works better.

2.5.2. *Highway Scenario*

The highway scenario was designed with the Pursue model to achieve 73% ad-hoc mobility and 91% ad-hoc connectivity. Figure 7 shows the highway scenario generated by CAD-HOC. The scenario is created with the use of beams, which function as the mobility supported areas, and restricted areas where the movement is restricted. The image used in the background is of the Baltimore Streetcar Museum. A total of 23 nodes were placed on the beams, and there were 150 CBR sources used throughout the simulation. The performance result obtained by simulating this highway scenario is shown in Table 3.

The results show that the total loss was less in DSR as compared to AODV and DSDV. Highest throughput and minimum delay were obtained by DSDV. In this case, as the total loss is less, source initiated routing protocol DSR works better than AODV and DSDV.

Table 3: Simulation results for the highway scenario

	DSR	AODV	DSDV
Throughput (bytes/sec)	149619.34	131547.49	162446.381
Delay (ms)	0.15181	0.075677	0.009139
Total Loss (%)	0.397	2.5977	8.732

2.5.3. *Conference Room Scenario*

The conference room scenario was designed by moving the nodes using the mouse and then using the repeat movement option (user-defined movements) to achieve 12% ad-hoc mobility and 92% ad-hoc connectivity. Figure 2 (in section 2 above) depicts the conference scenario generated using the tool.

The scenario is generated with three squares and few restricted areas. This scenario had 73 nodes and 100 CBR sources. The idea of this generated scenario was obtained from Johansson [12]. The performance results obtained by the simulation of the conference scenario are shown in Table 4.

Table 4: Simulation results for conference room scenario

	DSR	AODV	DSDV
Throughput (bytes/sec)	74054.407	68966.731	79310.141
Delay (ms)	0.55690	0.677241	0.494291
Total Loss (%)	8.868	11.23	28.64

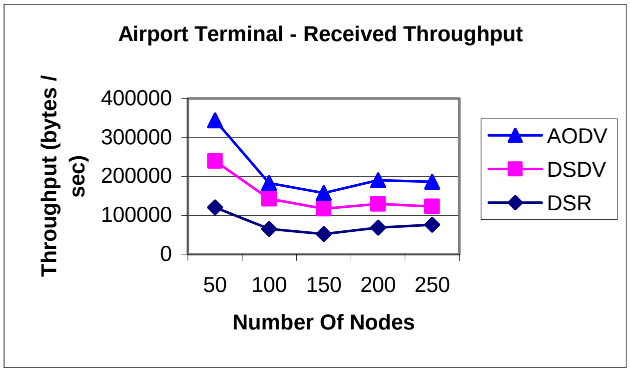
The results show that there was huge loss of about 28% of the packets in the scenario. AODV and DSR had almost the same loss. DSDV showed the highest throughput and minimum delay, while AODV showed the lowest throughput and the maximum delay. This result depicts that DSR, a source initiated routing protocol, works better in this scenario. The results obtained in this simulation match the results obtained by Johannson [12], which showed that there was a 25.4% loss for DSDV, and losses of about 2% and 6% for DSR and AODV, respectively. This very much matches the patterns of results obtained in our simulation, thus proving the usefulness of using realistic scenarios, which can now be designed and generated effortlessly using CAD-HOC.

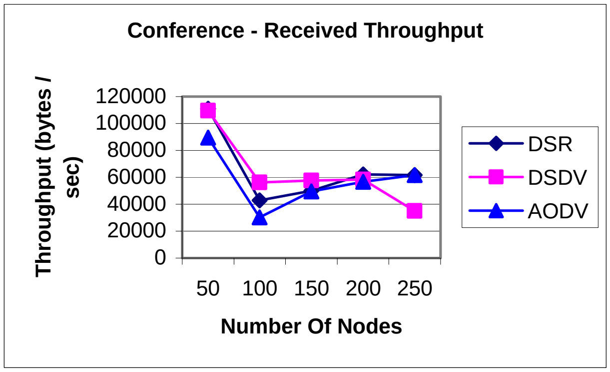
2.6. Effect of Number of Nodes on the Performance of Routing Protocols in Realistic scenarios

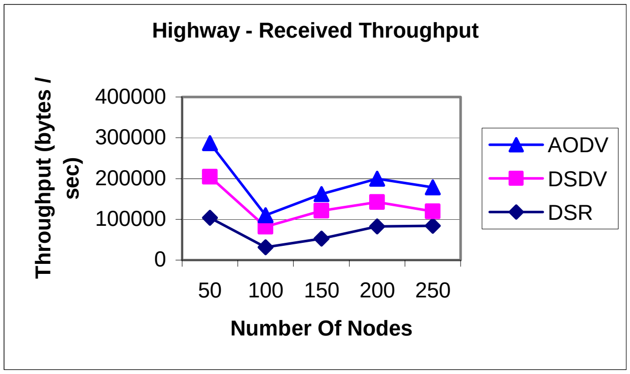
In all three scenarios that we have experimented with, 50 nodes (with 40 CBR sources), 100 nodes (with 100 CBR sources), 150 nodes (with 90 CBR sources), 200 nodes (with 140 CBR sources) and 250 nodes (with 180 CBR sources) were simulated, to understand how the received throughput, delay and loss of packets behave with a change in the number of nodes.

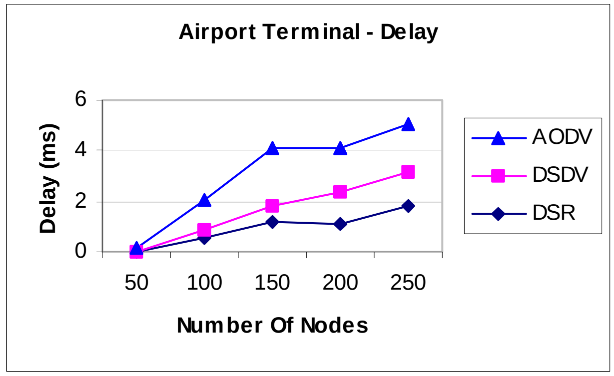
2.6.1. Received Throughput

Graphs depicting throughput vs. number of nodes, for all three scenarios, are shown below. Throughput here represents the received throughput, a combination of receiving data and control packets. The AODV protocol shows higher throughput but it should be due to the more number of control packets being received. This is logical as the graphs for delay and percentage loss show that AODV performed worse than the other two protocols.



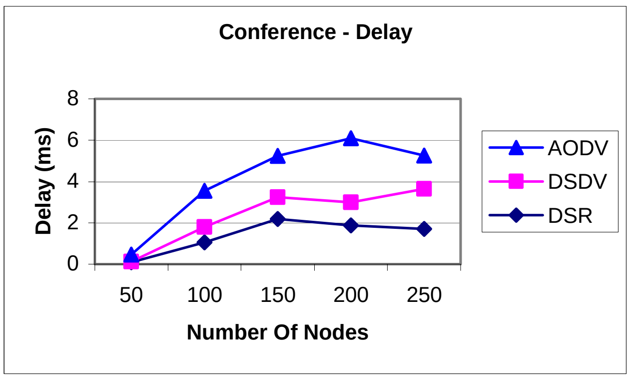


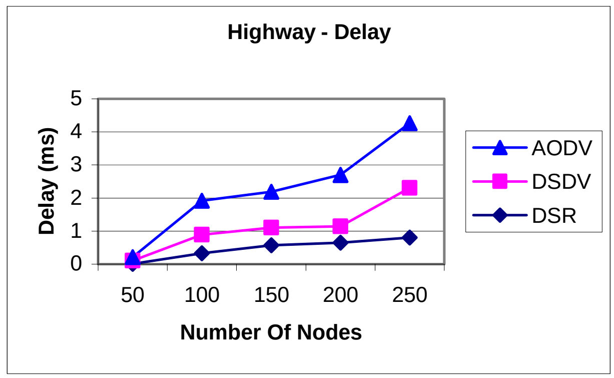




2.6.2. Delay

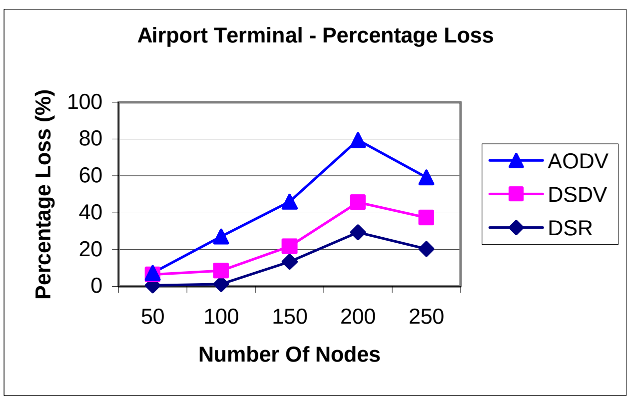
The graphs for delay vs. number of nodes are shown for all of the three scenarios. The graphs for delay show that DSR has the least delay for all the scenarios while AODV had the highest delay. This shows that for low or high mobility scenarios DSR has the lowest delay, while DSDV and AODV have higher delay. This is due to the fact that source initiated routing protocols incur less communication overhead and traffic.

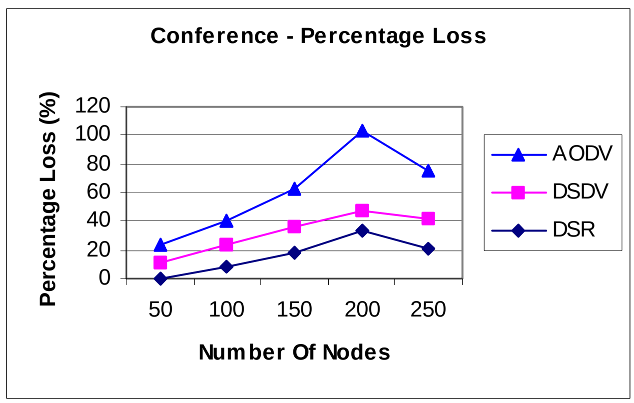


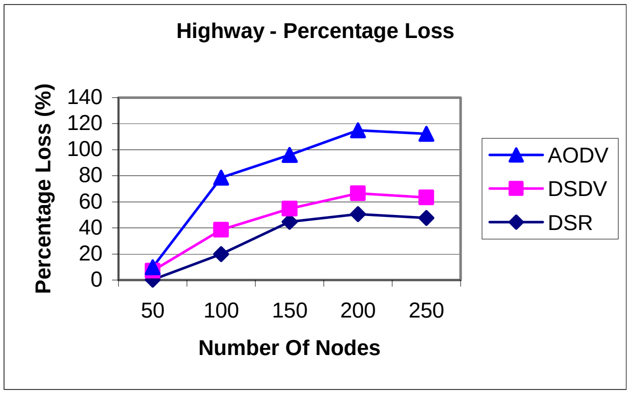


2.6.3. Percentage Loss

The graphs for percentage loss vs. number of nodes are shown for all of the three scenarios. The graphs for percentage loss show that DSR has the lowest loss for all the three scenarios and AODV has the highest loss as compared to other two protocols.







If we consider that DSR had less loss and less delay and assume that the throughtput is shown to be high for DSDV and AODV (due to control packets being received), we can conclude that DSR is a better protocol for the airport and highway scenarios, which represent moderate to high mobility. In the conference scenario, however, the throughput for DSR and DSDV was nearly equal. Even though the emphasis of our work was placed on creating the CAD-HOC tool and not in conducting comprehensive simulation studies, the little realistic simulations we have done raise a question on the generally accepted fact that the choice of ad-hoc routing protocol is mainly application dependent [6].

3. CONCLUSIONS

We presented CAD-HOC, a cousin tool to Network Simulator (*ns*), which allows ad-hoc networking experimentation to be performed under visually realistic scenarios such as an airport or a bus terminal, buildings, highways and other facilities. CAD-HOC focuses on capturing the visual scenario and transforming it into mobility and connection benchmarks, which are subsequently fed to *ns* to drive simulation experiments. Generation of mobility and connection benchmarks is facilitated by allowing the users to choose from Brownian, Column, Pursue, and Nomadic models, or to specify user-defined movements. The tool also provides on-line analyses on ad-hoc mobility, ad-hoc connectivity and message complexity, which give an estimate of the overall *ns* simulation

The tool allows the users to view in advance, the way the nodes will be moving in a scenario and specify the connection based on the movement. A user can always specify a connection between two nodes that are not in the direct range of each other and also specify a connection such that nodes are always in the transmission range of each other. The *ns* simulation results prove that the movement and connection files produced by CAD-HOC are *ns* compatible. The performance results prove the fact that the evaluation of routing protocols in ad-hoc networks should be application- and mobility- dependent.

Its is our belief that CAD-HOC, presented as a support function to ns, will aid the ns user community a great deal in understanding and studying the behavior of ad-hoc routing protocols in different realistic scenarios. This will help us learn what improvements are needed in the ad-hoc routing protocols. It is also our belief that CAD-HOC will prove to be one of the best tools in benchmarking mobility and play an important role in the evolution of new protocols.

The CAD-HOC tool has been developed at the Harris Networking and Communication Lab at the University of FLorida, where research is being done in the field of Mobile Computing and Networking. CAD-HOC can be downloaded from www.harris.cise.ufl.edu/projects/cadhoc.htm.

4. FUTURE WORK

There is room for feature improvement to the CAD-HOC tool. Such improvements would enhance the ease of using the tool and also allow it to generate even more realistic scenarios. The foremost thing to do is to semi-automate the generation of *tcl* script files that will contain the node configuration information and other information like the protocol to be used, link type, addressing type and additional data. This file will reduce the burden of the user to produce the .tcl files based on the scenario created by CAD-HOC.

Addition of functionalities like undo, resize and zoom in/zoom out will aid the user in better visualizing the scenarios. The undo operation allows the user to recover the previous state without actually deleting objects. Flexible canvas size aids the user to see the image as it is without actually scaling the image. Zoom in/zoom out is a feature of any ideal application, thus enhancing the visualization.

More group mobility models better suited for ad-hoc networks can be added, enhancing the ability of generation of more realistic scenarios.

In the current tool, while visualizing the scenario using the node connection specifier option, a thread is created for each node. Thus when the number of nodes increases, the ability of the node connection specifier to produce efficient visualization diminishes. A feature can be added that allows the user to select from the option of keeping speed or time constant to specify movement record, thus increasing the flexibility in generating mobility scenario.

CAD-HOC in its current scope provides extensive features that increase the ease of generating mobility scenarios and benchmarks, thus increasing the ease of performing simulations using *ns*. Future work can lead CAD-HOC to a full scale front-end to *ns*.

5. REFERENCES

[1] "CMU Monarch Project web site." http://www.monarch.cs.cmu.edu/.

[2] Das, Samir; Castenda, Robert; Yan, Jiangtao, "Comparative Performance Evaluation of Routing Protocols for Mobile, Ad-Hoc Networks," In Proceedings of IEEE International Conference on Computer Communications and Networks, San Antonio, Oct. 1998, pp. 153-161.

[3] "Explanation/Applets of Mobility Models." http://www.disca.upv.es/misan/mobmodel.htm.

[4] Gerla, M.; Pei, G.; Lee, S. J., "Wireless Mobile Ad-Hoc Network Routing," In Proceedings of ACM/IEEE WINLAB/Berkeley FOCUS, New Brunswick, NJ, May 1999.

[5] Holland, G.; Vaidya, N. H., "Analysis of TCP Performance over Mobile Ad-Hoc Networks," 5th Annual International Conference Mobile Computing and Networking, August 1999.

[6] Hong, X.; Gerla, M.; Pei, G.; Chiang C.-C., "A Group Mobility Model for Ad-Hoc Wireless Networks," In Proceedings of ACM/IEEE MSWiM'99, Seattle, WA, Aug. 1999, pp.53-60.

[7] Hu, Yih-Chun; Johnson, David B.; "Caching Strategies in On-Demand Routing Protocols for Wireless Ad-Hoc Networks," In Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking, ACM, Boston, MA, August 2000.

[8] IETF web site. http://www.ietf.org.

[9] Iwata, Atsuushi; Chiang, Ching-chuan; Pei, Guangyu; Gerla, Mario; Chen, Tsu-wei, "Scalable Routing Strategies For Ad-Hoc Wireless Networks," IEEE Personal Communications, Seattle, WA, 17(8): pp. 1369-1379, August 1999.

[10] Johnson, D. B., "Routing of Ad-Hoc Networks of Mobile Hosts," In Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, pp. 158-163, IEEE Computer Society, Dec. 1994.

[11] Johnson, David B.; Maltz, David A., "Protocols for Adaptive Wireless and Mobile Networking," IEEE Personal Communications, 3(1): 34-42, February 1996.

[12] Johansson, P.; Larsson, T.; Hedman, N.; Mielczarek, B.; Degermark M., "Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-Hoc Networks," Proceedings of ACM/IEEE MIBOCIM'99, Seattle, WA, Aug. 1999, pp. 195-206.

[13] Lee, S. J.; Toh, C. K.; "Performance Evaluation of Table-Driven and On-Demand Ad-Hoc Routing Protocols," In Proceedings of IEEE Personal Indoor and Mobile Radio Communications (PIMRC) Conference, Sept 1999, pp. 297-301.

[14] Lee S. J.; Gerla M., "A Simulation Study of Table-Driven and On-Demand Routing Protocols for Mobile Ad-Hoc Networks," IEEE Network, 13(4): pp. 48-54, July 1999.

[15] Lesiuk, B. Cameron, "Routing in Ad-Hoc Networks of Mobile Hosts," Mech590 Report, University of Victoria, British Columbia, December 2, 1998.

[16] Maltz, David A.; Broch, Josh; Johnson, David B., "Experiences Designing and Building a Multi-Hop Wireless Ad-Hoc Network Testbed." Technical Report CMU-CS-99-116, School of Computer Science, Carnegie Mellon University, Pittsburgh, March 1999.

[17] MANET Chapter. http://www.ietf.org/html.charters/manet-charter.html.

[18] Broch, Josh.; Maltz, David A.; Johnson, David D.; Hu, Yih-chun; Jetcheva, Jorjeta, " Performance Comparison of Multihop Wireless Ad-Hoc Networks Routing Protocols," In Proceedings of Mobicom' 98, Tel-Aviv, Israel, pp. 85-98.

[19] Marc Greis'Tutorial to NS. http://www.isi.edu/nsnam/ns/tutorial/index.html.

[20] Misra, Padmini, "Routing Protocols for Ad-Hoc Mobile Wireless Networks." ftp://ftp.netlab.ohio-state.edu/pub/jain/courses/cis788-99/adhoc_routing.pdf.

[21] Noble, B. D; Satyanarayan, M.; Nguyen, G. T; Katz, R. H, "Trace-based Mobile Networks Emulation," In Proceedings of the 1997 ACM SIGCOMM Conference, Cannes, France, 14-18 Sept., 1997, pp. 51-61.

[22] Nguyen, G. T.; Katz, R. H.; Noble, B.; Satyanarayan, M., "A Trace-based Approach For Modeling Wireless Channel Behavior," In Proceedings of the 1996 Winter Simulation Conference, Coronado, CA, 8-11 Dec., 1996, pp. 597-604.

[23] Java Developer Connection, NS Documentation (Manual). http://www.isi.edu/nsnam/ns/ns-documentation.html.

[24] NS Home Page. http://www.isi.edu/nsnam/ns/.

[25] Pei, G.; Gerla, M.; Hong, X., "A Wireless Hierarchical Routing Protocol with Group Mobility," In Proceedings of IEEE WCNC '99, Boston, MA, Sep 1999, pp. 1536-1537.

[26] Perkins, Charles E., "Ad-Hoc OnDemand Distance Vector Routing," In 2nd IEEE Workshop, Mobile Computing Systems and Applications, February 1999, Los Angeles, CA, pp. 90-100.

[27] Royer, Elizabeth M., "A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks", IEEE Personal Communications, April 1999, pp. 46-55.

[28] Sanchez, Miguel; Manzoni, Pietro, "Determination of Critical Transmission Range in Ad-Hoc Networks," IEEE Workshop on Multiaccess Mobility and Teletraffic for Wireless Communications 1999, Venice, Italy, 6-8 October 1999.

APPENDIX A

Geometric Mobility Model

The mobility of a scenario is described with single value M using the mobility metric, which is a function of relative motion of the nodes involved in the scenario. If *l(p,t)* is the position of node *p* at time *t*, the relative velocity *v(x,y,t)* between the nodes x and y at time t is

$$v(x, y, t) = \frac{d}{dt}(l(x,t) - l(y,t)) \qquad (2)$$

The mobility measure M_{xy}, between any pair (x, y) of nodes is defined as their absolute relative speed taken as an average over the time T. The formula of obtaining M_{xy} is given below.

$$M_{xy} = \frac{1}{T} \int_{t_0 \leq t \leq t_0 + T} | v(x, y, t) | \, dt \qquad (3)$$

To obtain the total mobility metric M, the mobility measure in (3) is averaged over all pairs of nodes leading to the following formula

$$M = \frac{1}{|x, y|} \sum_{x,y} M_{xy} = \frac{2}{n(n-1)} \sum_{x=1}^{n} \sum_{y=x+1}^{n} M_{xy} \quad (4)$$

where $|x,y|$ is the number the number of distinct node pairs *(x,y)* and n is the number of nodes in the scenario. The mobility metric expresses the average relative speed between all nodes in the network as seen in equation (4). The speed of a node standing still or moving in parallel at the same speed is zero. Mobility metric measures the mobility in meters per second.