

Report for CS641 Project

Yi Lu

Department of Computer Sciences

Purdue University

yilu@cs.purdue.edu

December 9, 2002

Visualization and Analysis Toolkit for Simulating Mobile Ad Hoc Networks in ns2

1 Introduction

1.1 Network Simulator ns2

Network simulator **ns2** is an event-driven network simulator used for networking research [2]. It is a widely used tool for simulating internetwork topologies to test and evaluate various networking protocols. There is a substantial support and flexibility in **ns2** to simulate various traffic generation patterns, routing and multicast protocols.

In order to study different networking issues like protocol interaction, congestion control, effect of network dynamics, scalability etc. it is necessary to simulate various scenarios that include different topology sizes, density distribution, traffic generation, membership distribution, real-time variance of membership, network dynamics etc. The **ns2** scenario generator can be used to create different random scenarios for simulation.

In **ns2**, characteristics of physical media of communication like delay, bandwidth, error rate, antennas and wireless physical interface parameters etc. can be defined. This helps in making the simulation studies as close to realistic scenarios as possible. **ns2** provides the flexibility to add and experiment new protocols or ideas. Recently, much support has been added for simulating wireless networks and interconnecting wired and wireless networks. Trace support in **ns2** may be used to trace packets for wireless and wired scenarios.

ns2 currently supports two mobile networking models. The basic wireless model was ported from CMU/Monarch group. It essentially consists of mobile nodes, which are movable and are able to transmit and receive on channels, with additional supporting features that allow simulations of multi-hop Ad-Hoc networks and wireless LANs. Four Ad-Hoc routing protocols that are currently supported, they are Destination Sequence Distance Vector (DSDV) [6], Dynamic Source Routing (DSR) [4], Temporally Ordered Routing Algorithm (TORA) and Ad-Hoc On-demand Distance Vector (AODV) [5].

1.2 The Problem

The following problems arise in the simulation of mobile ad hoc networks in ns2.

1.2.1 Visualizing Movements of Mobile Hosts

Protocols designed for mobile ad hoc networks must be adaptable the highly dynamic network topologies. To validate or evaluate a protocol, it will be investigated with various scenarios with different mobile host movements. The investigation of a protocol should

answer questions such as, Does the protocol correctly capture the change of network topology? Does it choose an appropriate approach according to current topology? etc.

The network topology is mainly determined by the positions of mobile host, which are determined by the movement of mobile hosts. The *random waypoint* model is the most commonly used model to generate movements for mobile hosts. According to this model, mobile hosts move independently. The speed and direction of a movement have no relation to those of the previous movement. Because of the randomness of this model, the generated movements can hardly be studied analytically. A visualization tool is needed to demonstrate the movements. The requirements of this tool include:

1. **Modelless:** It should not be model-specific. It must be able to visualize movements generated by any model.
2. **Wireless Link Visualization:** The network topology is also determined by the communication ranges of mobile hosts. This tool should be capable of visualizing wireless links if two mobile hosts are within each other's communication range.
3. **Positioning:** Sometimes, we are interested in a specific time point or range. The tool should be able to demonstrate the network topology at any time.

ns2 itself provides a visualization component called *nam*, but it is not dedicated to mobile ad hoc networks. Thus it can not show the wireless links.

1.2.2 Analyzing Experiment Data

ns2 uses trace file to keep track of network activities at different layers. A trace file provides the most detailed information, but it has severe drawbacks.

- **File Size:** The size of a trace file can be incredibly large. For instance, a trace file for a 1000 seconds simulation of 30 mobile hosts with 30 connections can be as large as 500M (only trace routing and agent layer). Usually, a simulation of mobile ad hoc networks is repeat several times to compensate the randomness of scenarios. Thus, keeping all trace files requires a huge amount of storage space.
- **Comprehensive Information:** Trace files do not provide any comprehensive information. It only contains millions of records that capture the network activities. However, most of the time, researchers are only interested in comprehensive information such as throughput, delay, etc.

Thus a tool is needed to analyze trace files. The desired properties of the tool include:

1. **Generating Intermediate Data:** It should be able to generate intermediate data with reasonably detailed information that can reflect the most important facts of the simulation. The size of the file should be less than 1M.

2. **Illustration:** It should be able to illustrate the results of the simulation using graphics.
3. **Compatibility:** It should be able to export data that can be understood by other generic purpose analysis software, such as Matlab, SAS, etc.

Other ns2 users are also aware of this problem. Several analysis programs have been developed. Most of them do not provide the functionality to illustrate the results. One Matlab-base program called *tracegraph* [3] has the desired properties, but it is terribly slow. It took 30 minutes to read a 20M trace file.

In this work, we design and implement a toolkit to support visualizing movement and analyzing trace data for simulating mobile ad hoc networks in **ns2**.

2 Visual Movement

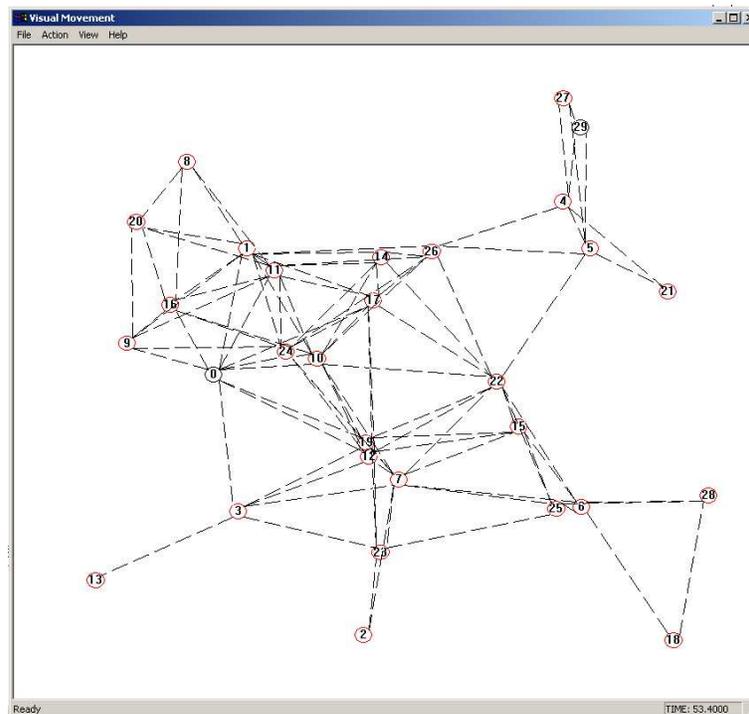


Figure 1: Visual Movement

This component that visualizes movements of mobile host is called *Visual Movement*. It is implemented using Visual C++. This component reads a movement file generated by

the **ns2** scenario generator and illustrate the movement and network topology. The format of the movement file is described in [2].

In *Visual Movement*, a user can specify how fast he/she wants the movement to be demonstrated (available options are as 1, 5, 10, and 20 times fast as the simulation). A user can also let the demonstration jump to anytime point. The demonstration can also be stopped and resumed at anytime. Currently, the communication range of each mobile host is set to 250m. It can be changed easily.

Figure 1 shows a visualized movement. The circles represent mobile hosts. The number in a circle represents the ID of that mobile host. The wireless links are represented by the dotted lines. The counter at the right left bottom of the window indicates the time in simulation. For instance, figure 1 shows the network topology at simulation time 53.4 seconds.

3 Analyzer

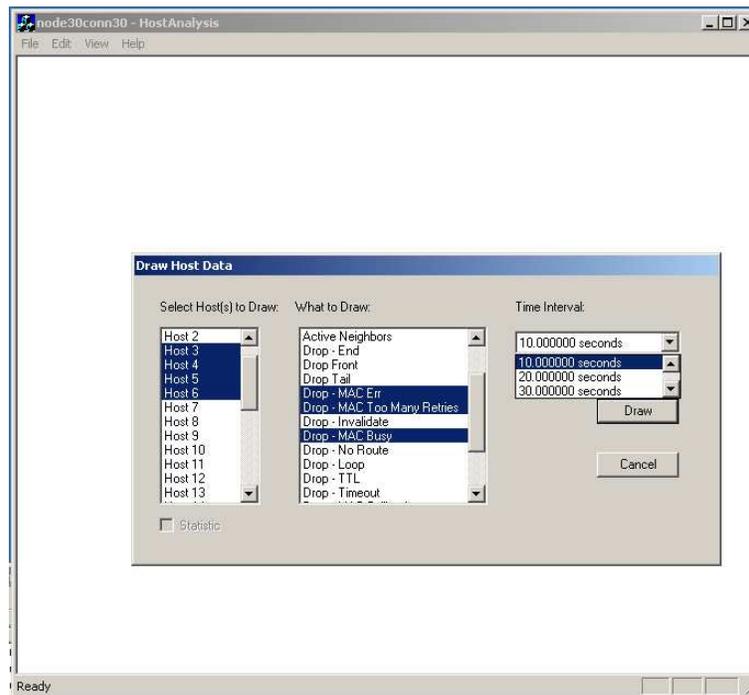


Figure 2: Draw Dialog

The component that analyzes trace files is called *Analyzer*. *Analyzer* can extract three

kinds of information.

- **System-wide information** includes how many packets (or bytes) have been sent? How many have been received? How many protocol packets have been sent or received? How many packets have been forwarded? What is the average delay?
- **Connection-oriented information** is specific to each connection, including how many packets have been sent for this connection? How many been dropped? What is the latency? What is the average delay?
- **Host-oriented information** is specific to each host, including how many packets this host has sent? How many this host has received? How many this host has forwarded? How many this host has dropped? What are the reasons for dropping these packets?

All the extracted information is stored in an intermediate file. This part is implemented using C++.

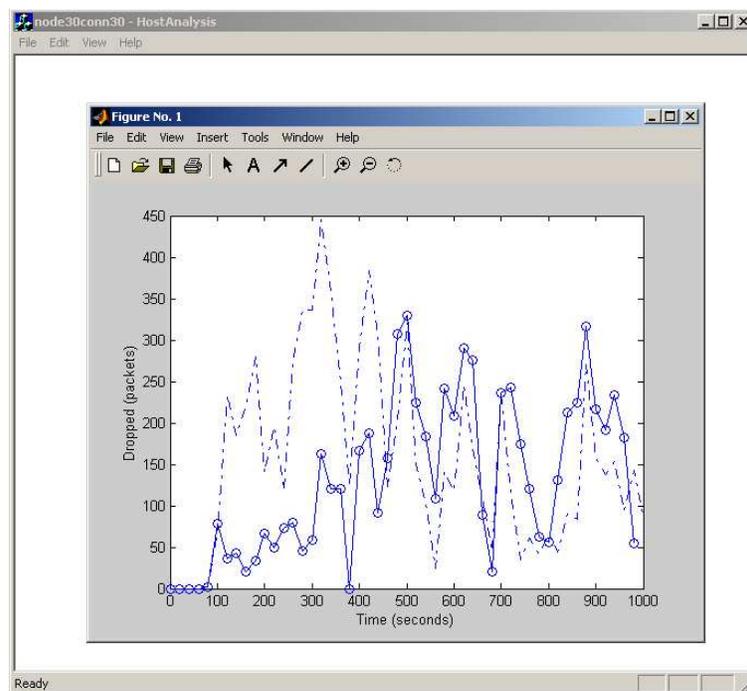


Figure 3: An Example

Currently, a graphic tool is implemented to illustrate the host-oriented information. It reads a generated intermediate file and uses 2-D graph to visualize the information.

Matlab [1] provides a full set of functions to manipulate 2-D graph such as plot. But it is not suitable for low level file I/O. To make this component efficient and flexible, we use C++ to operate the files and use the graph APIs provide by Matlab to plot the data.

Figure 2 shows the Draw Dialog of the program. The title bar of the windows shows the name of the intermediate file(i.e., node30conn30). A user can choose to draw the data of a single host, the sum of data of multiple hosts, or draw multiple hosts separately. He/she can also specify what data to draw. It could be a single item such as *received packets*, *dropped packets*, *active neighbors*, etc. It also can be the combination of multiple data items. For instance, dropped packets due to MAC-related reasons (i.e., MAC error, MAC too many retried, and MAC busy), as shown in the figure. A user can also specify the time interval to plot the data.

Figure 3 shows an example. The two curves represent the dropped packet at host 4 and host 5 respectively.

4 Experiment

The program discussed above is a visualization and analysis toolkit for simulating mobile ad hoc in **ns2**. We do use it to help validating, debugging, and evaluating protocols.

Figure 4 shows the results we got for the study of DSDV and AODV protocols by using this toolkit.

References

- [1] Matlab. <http://www.matlab.com>.
- [2] The network simulator - ns-2. <http://www.isi.edu/nsnam/ns/>.
- [3] Trace graph. <http://www.geocities.com/tracegraph>.
- [4] D. B. Johnson, D. A. Maltz, Y. Hu, and J. G. Jetcheva. The dynamic source routing protocol for mobile ad hoc networks (DSR). *IETF Internet Draft*. <http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-07.txt>, February 2002.
- [5] C. E. Perkins, E. M. Belding-Royer, and S. R. Das. Ad hoc on-demand distance vector (AODV) routing. *IETF Internet Draft*. <http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-11.txt>, June 2002.
- [6] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings of the SIGCOMM 94 Conference on Communications Architectures, Protocols and Applications*, pages 234–244, August 1994.

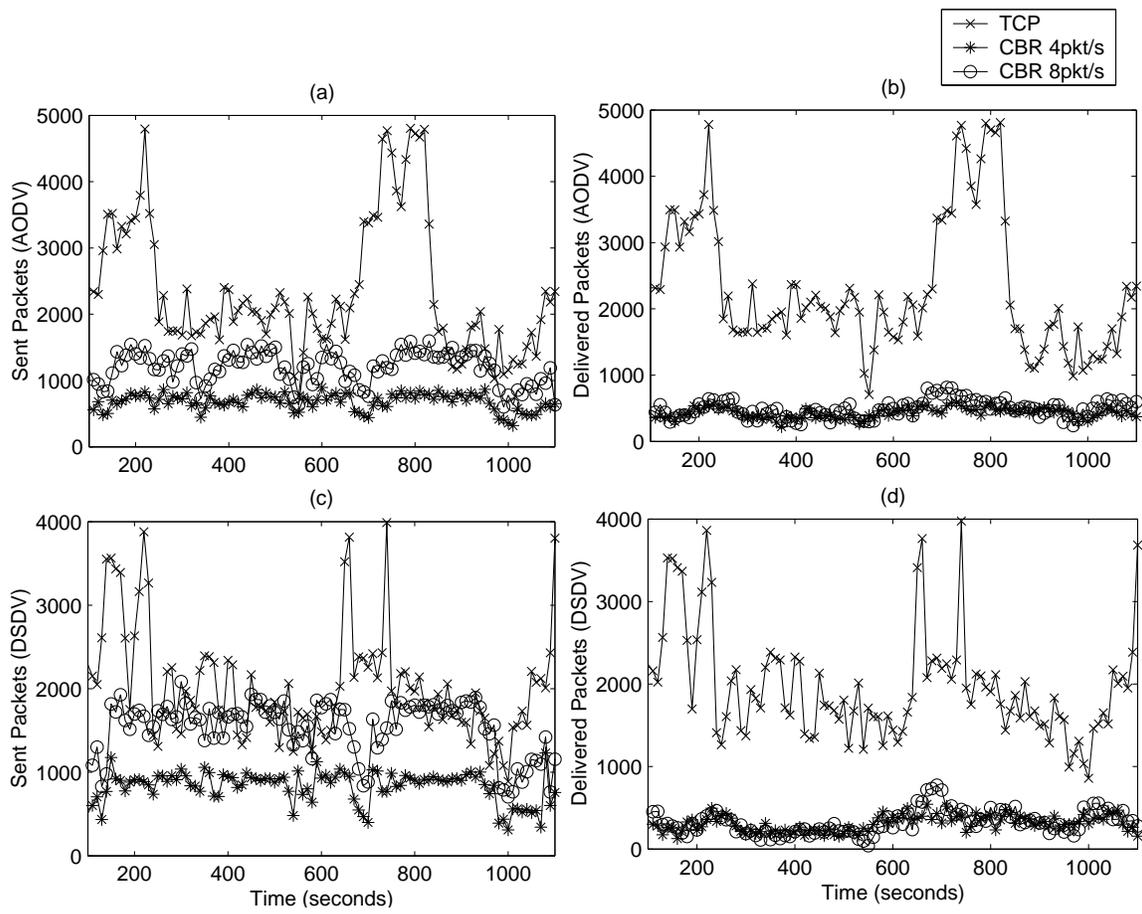


Figure 4: Experiment Results