

# Technical Report - TR-BU-CSE001-Oct09

BRAC University, Bangladesh

October, 2009

## ViSim: A User-Friendly Simulation Tool for MANET Routing Protocols

**Nazmus Saquib**

**Md. Sabbir Rahman Sakib**

**Al-Sakib Khan Pathan, Ph.D.**



66 Mohakhali, Dhaka 1212, Bangladesh

Copyright © 2009-2010, BRAC University, Bangladesh. All rights reserved.

## **Prepared By**

**Nazmus Saquib**, Lecturer, EEE Department, BRAC University, Bangladesh

*E-Mail:* [nsaquib@bracu.ac.bd](mailto:nsaquib@bracu.ac.bd)

**Md. Sabbir Rahman Sakib**, LTO, EEE Department, BRAC University, Bangladesh

*E-Mail:* [srsakib@bracu.ac.bd](mailto:srsakib@bracu.ac.bd)

**Al-Sakib Khan Pathan, Ph.D.**, Assistant Professor, CSE Department, BRAC University, Bangladesh

*E-Mail:* [spathan@bracu.ac.bd](mailto:spathan@bracu.ac.bd), [spathan@ieee.org](mailto:spathan@ieee.org)

*For any query, comments, or suggestions, please contact the authors.*

## **ViSim: A Visual Simulation Tool**

### **WHAT IS VISIM?**

ViSim is a new simulation tool that has a user-friendly graphical interface. ViSim could be useful for researchers, students, teachers in their works, and for the demonstration of various wireless network scenarios on the computer screen. It could make the task of simulation more exciting and enhance the interest of the users without going into complex command-only text interface.

### **BUILDING BLOCKS OF VISIM**

We have used two software in Windows environment for our work; ActiveTcl and Microsoft Visual Basic. Before describing ViSim's features and functionalities, in this sub-section we talk about these briefly.

ActiveTcl is an industry-standard Tcl distribution, available for Windows, Linux, Mac OS X, Solaris, AIX and HP-UX. This software creates an environment in Windows to run the ns-2 [1] simulations and .tcl scripts. It is capable of executing the simulation at a rapid pace than cygwin [2]. This package contains ns.exe and nam.exe, two executable files. Once a .tcl script is written referring to a particular scenario with specifications of different simulation parameters such as ad hoc routing protocol name, number of nodes, nodal positions, MAC layer type, simulation area, time, etc; ActiveTcl can run the simulation in Windows environment. For details of ActiveTcl and its latest versions, the readers are encouraged to visit the URL; <http://www.activestate.com/activetcl/>.

Microsoft Visual Basic is a popular software that we have used for developing ViSim

prototype so that it can connect the simulation related tasks with a user-friendly graphical interface. For our work, we have used ActiveTcl8.3.5 and Visual Basic 6.0.

## **OVERVIEW OF VISIM**

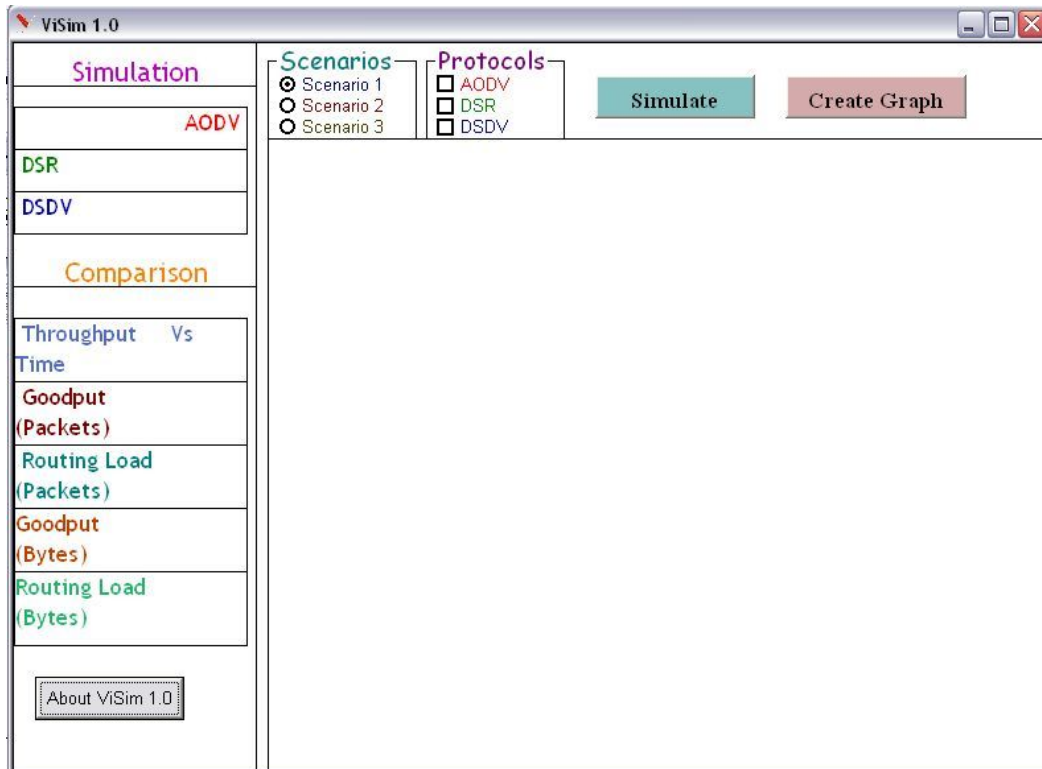
Our graphical simulation tool, ViSim is built using Visual Basic 6.0 in order to make comparisons among various MANET routing protocols since there are very few prototypes available today for performing such type of task. Most of the available tools are somewhat not user-friendly. Hence, keeping that in mind, we built ViSim in such a way that any naïve user can also be able to use this tool to visualize the background simulations done in ns-2 (that is run with the help of ActiveTcl in Windows operating system). ViSim runs a particular .tcl file for all the three mentioned protocols (DSDV, DSR, AODV) and extracts the required information from the trace files that are generated. Eventually the graphs are plotted for different performance indicators such as Throughput, Goodput, and Routing Loads. ViSim can make the task of a network administrator easy to decide which routing protocol would be better for the particular MANET scenario.

## **DIFFERENT WORKING AREAS**

Figure 1 shows the ViSim prototype/tool when it is run in Windows environment for the first time. The graphical interface has some working areas and functionalities that should be known before using it for analysis of various parameters. There are mainly three portions/areas on the ViSim interface:

**(a) Simulation:** In this area, three routing protocols are mentioned. Clicking on the

names of each protocol gives the options of simulating three network scenarios using that particular protocol. The network scenarios could be modified as required in the .tcl scripts that run in the background.

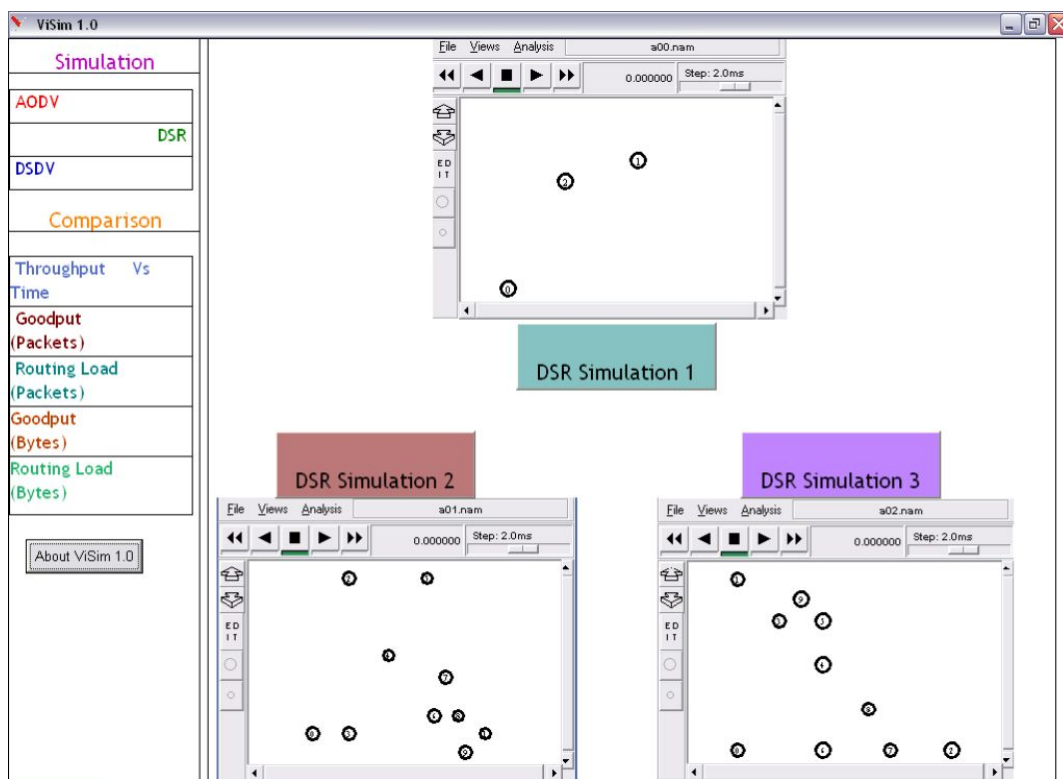


**Figure 1. Visual Simulation Tool Interface, ViSim 1.0 (run in Windows XP).**

**(b) Comparison:** This area has the options; Throughput vs Time, Goodput (Packets), Routing Load (Packets), Goodput (Bytes), and Routing Load (Bytes). All these buttons are used to select the parameters that the user needs for the performance analysis and comparison among the routing protocols.

**(c) Scenarios and Protocols:** This area specifies the options of three network scenarios (radio buttons) and three routing protocols (tick boxes). Also it has two buttons namely; 'Simulate' and 'Create Graph'. 'Simulate' button is used for playing the simulations and 'Create Graph' is used to plot the comparison graphs.

**(d) Output:** Output area is the right-bottom area which is shown as a blank window area when ViSim is run for the first time. Based on the choice of various options, the outputs or further options are shown in this area. The graphs are also plotted on this area when the user chooses the option of creating graphs after performing various simulations and comparisons.



**Figure 2. DSR simulation options.**

## **FUNCTIONALITIES OF VISIM WITH EXAMPLES**

Now, let us see the functionalities of ViSim with some practical examples. Let us suppose that we want to visualize the simulation for DSR for a particular network scenario. For this task, first we have to click DSR button under simulation area. After clicking DSR button, ViSim shows three more options (DSR Simulation 1, DSR

Simulation 2, and DSR Simulation 3) on the output area as shown in Figure 2.

From these three options any one could be chosen. For our task, let us choose DSR Simulation 3. After clicking this button, ViSim calls ns-2 in its background, then reads .tcl file that specifies the simulation scenario 3, generates .nam and .tr files. Once the .nam and .tr files are generated, ViSim calls the NAM (Network Animator tool) in its background and reads the generated .nam files. Consequently, it shows a screen for simulation [see Figure 3].

On the NAM screen, there are few buttons such as play, forward, backward, stop buttons available to control the simulation as these are done usually in Linux based environment with ns-2 and NAM. To see the visual simulation on the screen, the play button should be clicked. Like any other simulation using NAM, we can also change the step size of the simulation.

Now, if we want to make comparisons among three different protocols for performance analysis, we have to choose a specific network scenario. In our case, let us select Scenario 1. Then we have to select three mentioned protocols (or, any two or one) and side by side the performance indicators should be clicked from the five options in the comparison area. Figure 4 shows the output where we selected 'Throughput Vs Time'.

Once the simulations are performed by clicking the 'Simulate' button, we can use the generated results in the background for plotting comparison graphs. Basically, this 'Simulate' button facilitates performing various simulations with three protocols for a

particular network scenario at the same time. This reduces the burden of doing the tasks repeatedly or selecting one protocol at a time under Simulation area. Once all the simulations are completed, the graph can be generated by clicking the ‘Create Graph’ button. By clicking ‘Create Graph’ button, we send the command to read the generated .tr files (trace files) and extract the required information/values from those. These values are used to plot the graphs for different protocols for a specific scenario and for different performance indicators. Figure 5 shows a sample output of what we have done so far (as an important note it should be mentioned that each simulation and plotting of graph takes a bit time as required by ns-2; for example in our case, it took about 25 seconds to plot the graph on the ViSim output area).

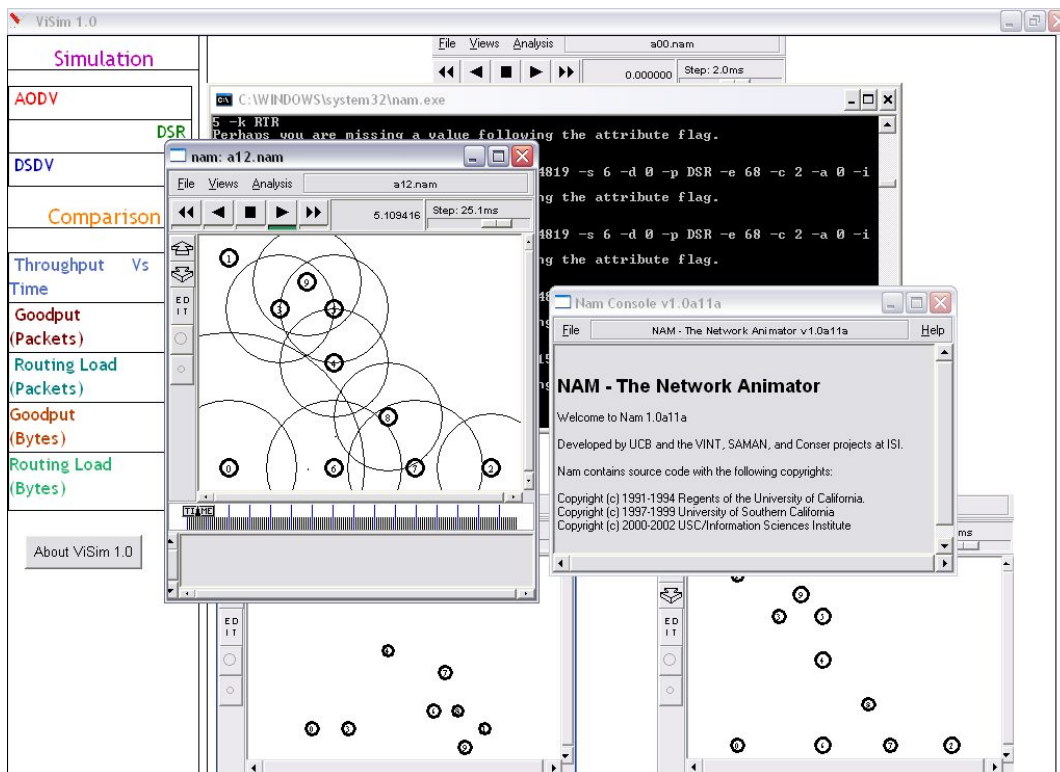


Figure 3. The output after choosing DSR Simulation 3.



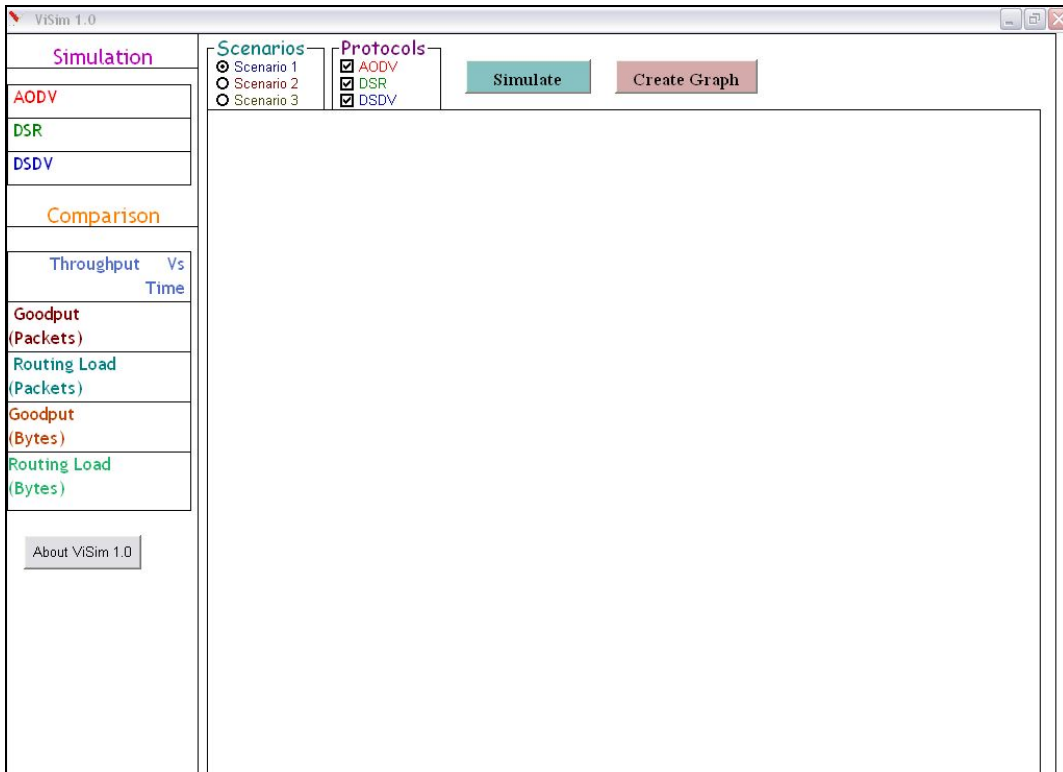


Figure 4. An example where 'Throughput vs Time', Scenario 1, and three protocols are selected for running the simulations.

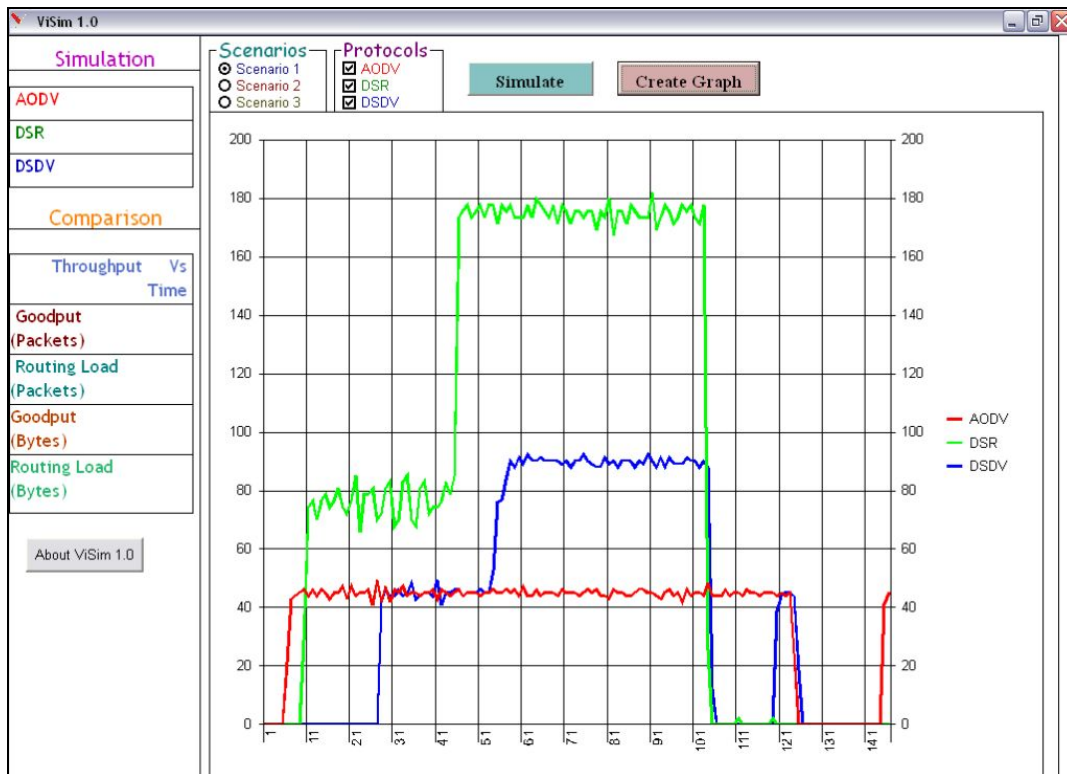


Figure 5. A sample output graph (Throughput vs Time) using network scenario 1, all three protocols; DSDV, DSR, and AODV are compared.

Let us talk about the working mechanism of ViSim buttons a bit. When the user selects the simulation option in order to view the simulation for a particular scenario corresponding to the selected ad hoc network protocol, ViSim calls up a .bat file which contains shell script. This shell script calls the ns-2 and feeds files or file having extension .tcl, according to the choice of simulation. Then ns-2 generates trace file (extension .tr) and nam file (extension .nam). After that NAM is called via shell script and using NAM the script feeds .nam file into NAM which gives a GUI (Graphical User Interface) popup and using it, a user can actually observe the simulation. Again, when the user selects the Comparison option and clicks Create Graph after performing simulations, ViSim gathers the .tr files according to the choice of protocol, reads those and according to the performance indicators, it filters the data and picks up important information to generate the graph. Details about the features of our simulation tool; ViSim, its installation, and user's manual can be found in this URL: <http://faculty.bracu.ac.bd/~spathan/research/visim.html>

For ViSim, we have used some given network specifications. Note that any specification can be modified in the .tcl files according to the requirements to simulate another network setting. Also, various parameters used in ViSim code could be given new values. We are planning to make ViSim an open-source tool so that it could be customized to fit a particular wireless network (MANET, Wireless Sensor Networks, etc.) scenario.

## PERFORMANCE EVALUATIONS AND RESULTS: EXAMPLE OF USING VISIM

In this section, we present the obtained results from our performed experiments using ViSim tool.

### *Simulation Parameters and Specifications*

We used the following specifications and parameters for our experiments:

Simulation Parameter	Value
Channel Type	Wireless Channel
Radio-propagation model	Two Ray Ground Model
Network interface type	Wireless Physical
MAC type	802_11 b
Interface Queue Type	Drop Tail Primary Queue
Antenna model	Omni Direction
Number of Mobile nodes	3-10
Ad Hoc Routing Protocol	DSDV, DSR, AODV
Simulation Area	500m x 400m
Simulation Time	150 ms
Traffic Type	TCP
Nodal speed	3-10 m/s
Packet size	1040 Byte (Data Packets) 40 bytes(Acknowledgement Packets) 60 Bytes (Routing Packets)
Total Number of different Scenarios	15

Comparisons among different protocols were based on the aggregate of the performance metrics resulting from the simulations of 15 different scenarios that were being performed for each protocol separately. To measure the performances, we used the following metrics:

**Throughput:** The total bytes received by the destination node per second (Data packets and Overhead).

**Goodput:**

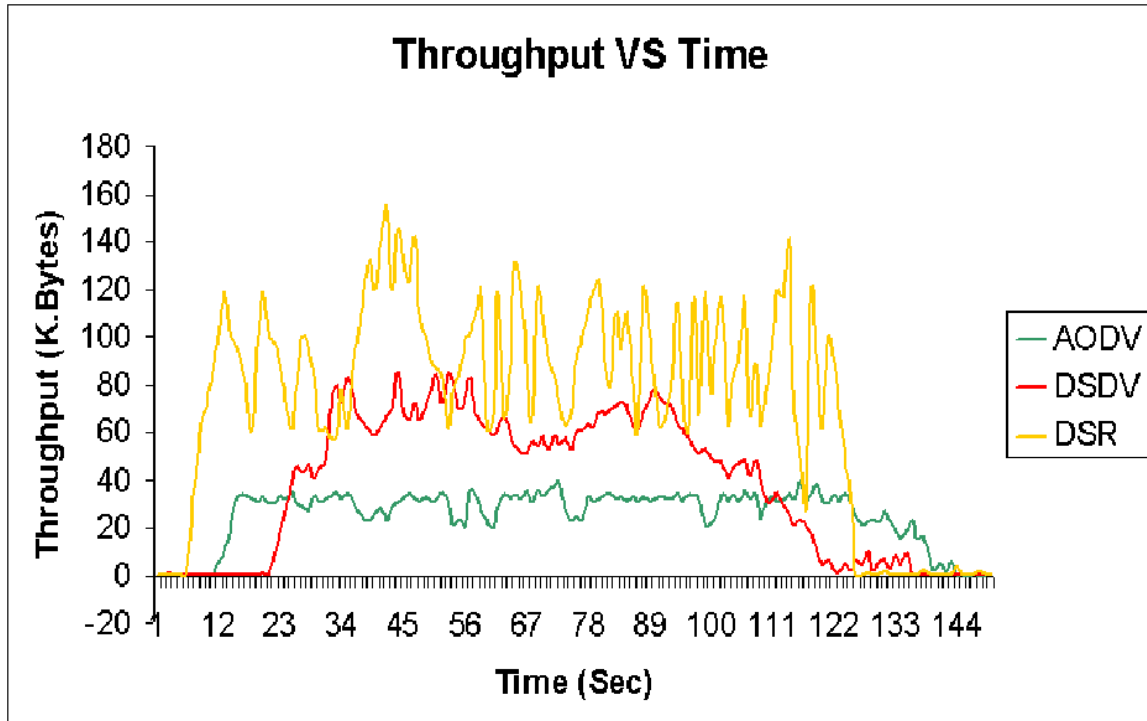
*Goodput (In terms of Number of packets):*

The ratio of the total number of data packets that are sent from the source to the total

number of packets that are transmitted within the network to reach the destination.

***Goodput (In terms of Packet Size in Bytes):***

The ratio of the total bytes of data that are sent from the source to the total bytes that are transmitted within the network to reach the destination.



**Figure 6. Throughput vs Time.**

**Routing Load:**

***Routing Load (In terms of Number of packets):***

The ratio of the total number of routing packets that are sent within the network to the total number packets that are transmitted within the network to reach the destination.

***Routing Load ( In terms of Packet Size in Bytes):***

The ratio of the total bytes of routing packets that are sent within the network to the

total number of bytes that are transmitted within the network to reach the destination.

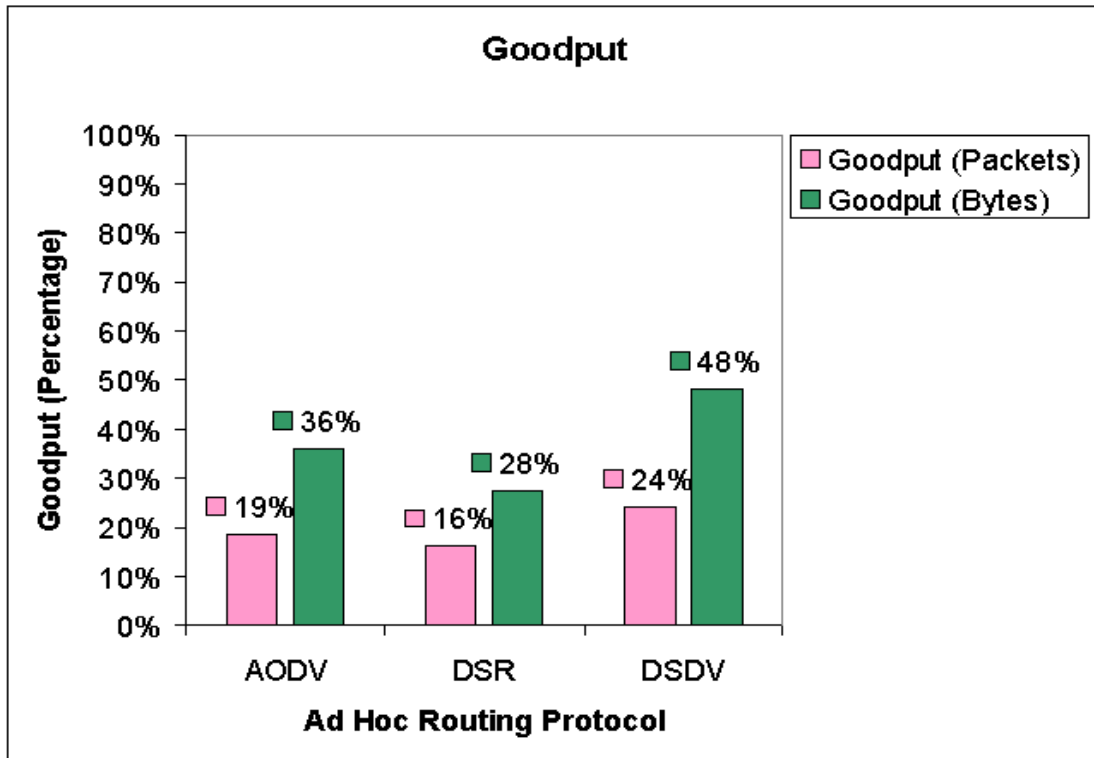


Figure 7. Goodput for three MANET routing protocols.

### *Simulation Results*

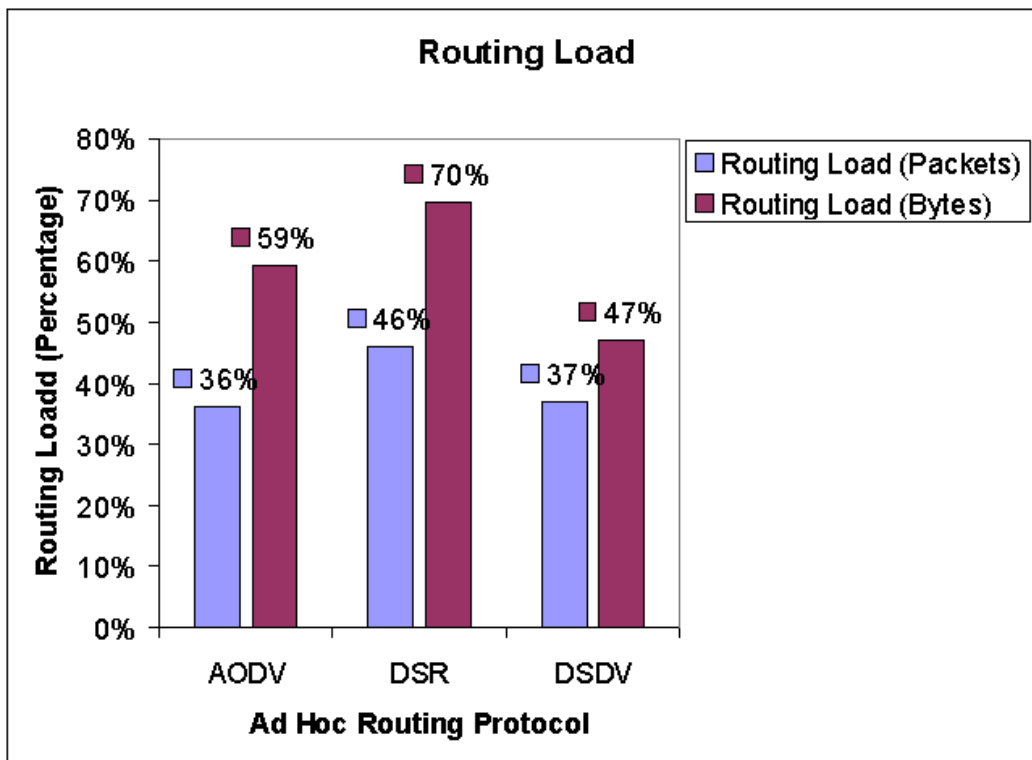
Figure 6 shows 'Throughput vs. Time' where we analyzed the total bytes received by the destination node per second (Data packet and Overhead). Based on the results that we see here, the following comments could be made:

**AODV:** starts off quickly and the data rate is more stable.

**DSR:** starts off quickly however we can see that there are lots of fluctuations in the data rate.

**DSDV:** takes time to start off but the data rate has lesser fluctuations.

We calculated Goodput in terms of number of packets and the packet size in bytes. Now, if we analyze the graph presented in Figure 7, we can see that on an average, if 100 packets are transmitted in the network, 19 packets would be data packets for AODV, 16 for DSR, and 24 for DSDV. In term of bytes, on an average; if 100 bytes of packets are transmitted through the network, 36 bytes would be data packets for AODV, 28 bytes for DSR, and 48 bytes for DSDV. From these data, we could deduce that; though DSDV takes time to converge, it actually is sending more data packets in number as well as in bytes than that of AODV and DSR. Now, the rest of the percentage of each individual graph will be the overheads that contain routing packets and acknowledgements.



**Figure 8. Routing loads for different experimented MANET protocols.**

We again calculated routing loads in terms of number of packets and the packets size in bytes. The results are presented in Figure 8. Again we can see that; though DSR has a

better Throughput, it actually contains more overhead for routing packets. However, DSDV has a relatively lower routing load than AODV and DSR.

## **CONCLUSIONS**

In this technical report, we have presented our user-friendly simulation tool/prototype which can ease the task of simulation of MANET routing protocols even in Windows based environments. Many users dealing with ns-2 simulations face troubles in setting up Linux or other systems and environment. The use of ActiveTcl with graphical ViSim interface could really be beneficial for the research community in general. Using our simulation tool, we obtained different graphs and analyzed the results for different scenarios which lead us to the following conclusions:

1. For AODV, we can see that it adapts quickly to the change of the network and has a relatively stable throughput with a moderate goodput. So, in an application where there is a fast change in the network topology and a requirement of stable data rate, AODV is more preferable.

2. DSDV turns out to have the best goodput and lesser routing load; however, it takes time to converge. So if there is relatively lesser number of nodes in the network and the mobility is somewhat steady or slow, DSDV will work more efficiently.

3. DSR, though has a very high throughput, it actually contains less data packets and we can see that there are lots of fluctuations on the throughput curve which are not preferred in a wireless network.

As our future works, we would like to add more functionalities to ViSim with easy access to the programming codes and parameter changes for various network scenarios.

We'd also like to investigate other established routing protocols to make a full-scale comparison using our visual simulation tool, ViSim. For the information about the official release of ViSim version 1.0, the readers are encouraged to visit:

<http://faculty.bracu.ac.bd/~spathan/research/visim.html>

## **References**

[1] The Network Simulator - ns-2; available at <http://www.isi.edu/nsnam/ns/>

[2] <http://www.cygwin.com/>